

Gwent

2.1.1

Generated by Doxygen 1.9.8

1 Gwent - A Calculator	1
1.1 Overview	1
1.2 Printing analog cards	1
1.3 Documentation	1
2 Hierarchical Index	3
2.1 Class Hierarchy	3
3 Class Index	5
3.1 Class List	5
4 File Index	9
4.1 File List	9
5 Class Documentation	13
5.1 Ability Enum Reference	13
5.1.1 Detailed Description	13
5.1.2 Member Data Documentation	13
5.1.2.1 BINDING	13
5.1.2.2 HORN	14
5.1.2.3 MORAL_BOOST	14
5.1.2.4 NONE	14
5.1.2.5 REVENGE	14
5.2 AddCardDialog Class Reference	14
5.2.1 Detailed Description	16
5.2.2 Constructor & Destructor Documentation	16
5.2.2.1 AddCardDialog() [1/2]	16
5.2.2.2 AddCardDialog() [2/2]	16
5.2.3 Member Function Documentation	16
5.2.3.1 onCreate()	16
5.2.4 Member Data Documentation	17
5.2.4.1 disposables	17
5.2.4.2 pickerAdapter	17
5.2.4.3 row	17
5.2.4.4 soundManager	17
5.3 AppDatabase Class Reference	18
5.3.1 Detailed Description	18
5.3.2 Member Function Documentation	19
5.3.2.1 rows()	19
5.3.2.2 units()	19
5.4 BondDamageCalculatorDecorator Class Reference	19
5.4.1 Detailed Description	20
5.4.2 Constructor & Destructor Documentation	21
5.4.2.1 BondDamageCalculatorDecorator()	21

5.4.3 Member Function Documentation	21
5.4.3.1 calculateDamage()	21
5.4.3.2 isBuffed()	22
5.4.4 Member Data Documentation	22
5.4.4.1 idToSquadSize	22
5.5 BurnDialogUseCase Class Reference	22
5.5.1 Detailed Description	23
5.5.2 Member Function Documentation	23
5.5.2.1 burn() [1/2]	23
5.5.2.2 burn() [2/2]	23
5.5.2.3 getBurnUnits()	24
5.6 ResetAlertDialogBuilderAdapter.Callback Interface Reference	24
5.6.1 Detailed Description	25
5.6.2 Member Function Documentation	25
5.6.2.1 reset() [1/2]	25
5.6.2.2 reset() [2/2]	25
5.7 ChangeFactionDialog.Callback Interface Reference	25
5.7.1 Detailed Description	26
5.7.2 Member Function Documentation	26
5.7.2.1 onThemeSelect()	26
5.8 CardListAdapter Class Reference	26
5.8.1 Detailed Description	28
5.8.2 Constructor & Destructor Documentation	28
5.8.2.1 CardListAdapter()	28
5.8.3 Member Function Documentation	28
5.8.3.1 getItemId()	28
5.8.3.2 onBindViewHolder()	29
5.8.3.3 onCreateViewHolder()	29
5.8.4 Member Data Documentation	29
5.8.4.1 onCopy	29
5.8.4.2 onRemove	30
5.9 CardNumberPickerAdapter Class Reference	30
5.9.1 Detailed Description	31
5.9.2 Constructor & Destructor Documentation	31
5.9.2.1 CardNumberPickerAdapter()	31
5.9.3 Member Function Documentation	32
5.9.3.1 addSelectedUnits()	32
5.9.3.2 getDelayedOnValueChangeListener()	32
5.9.4 Member Data Documentation	32
5.9.4.1 abilityPicker	32
5.9.4.2 damagePicker	32
5.9.4.3 EPIC_UNIT_ABILITIES	33

5.9.4.4 epicPicker	33
5.9.4.5 numberPicker	33
5.9.4.6 squadManager	33
5.9.4.7 squadPicker	33
5.10 CardUiState Class Reference	33
5.10.1 Detailed Description	35
5.10.2 Constructor & Destructor Documentation	35
5.10.2.1 CardUiState()	35
5.10.3 Member Function Documentation	35
5.10.3.1 equals()	35
5.10.3.2 getAbilityImageId()	36
5.10.3.3 getDamageBackgroundImageId()	36
5.10.3.4 getDamageString()	36
5.10.3.5 getDamageTextColor()	37
5.10.3.6 getSquadString()	37
5.10.3.7 getUnitId()	37
5.10.3.8 showAbility()	37
5.10.3.9 showSquad()	38
5.10.4 Member Data Documentation	38
5.10.4.1 abilityImageId	38
5.10.4.2 damageBackgroundImageId	38
5.10.4.3 damageString	38
5.10.4.4 damageTextColor	38
5.10.4.5 DIFF_CALLBACK	39
5.10.4.6 squadString	39
5.10.4.7 unitId	39
5.10.4.8 UNUSED	39
5.11 CardUiStateFactory Class Reference	40
5.11.1 Detailed Description	40
5.11.2 Constructor & Destructor Documentation	40
5.11.2.1 CardUiStateFactory()	40
5.11.3 Member Function Documentation	41
5.11.3.1 createCardUiState() [1/2]	41
5.11.3.2 createCardUiState() [2/2]	41
5.11.4 Member Data Documentation	41
5.11.4.1 damageTextColors	41
5.11.4.2 horn	42
5.11.4.3 weather	42
5.12 CardListAdapter.CardViewHolder Class Reference	42
5.12.1 Detailed Description	43
5.12.2 Constructor & Destructor Documentation	44
5.12.2.1 CardViewHolder()	44

5.12.3 Member Function Documentation	44
5.12.3.1 getAbilityView()	44
5.12.3.2 getBindingView()	44
5.12.3.3 getDamageView()	44
5.12.3.4 setItemId()	45
5.12.4 Member Data Documentation	45
5.12.4.1 abilityView	45
5.12.4.2 bindingView	45
5.12.4.3 damageView	46
5.12.4.4 itemId	46
5.13 ChangeFactionDialog Class Reference	46
5.13.1 Detailed Description	48
5.13.2 Constructor & Destructor Documentation	48
5.13.2.1 ChangeFactionDialog()	48
5.13.3 Member Function Documentation	48
5.13.3.1 getOnThemeClickListener()	48
5.13.3.2 onCreate()	48
5.13.4 Member Data Documentation	49
5.13.4.1 callback	49
5.14 CoinFlipDialog Class Reference	49
5.14.1 Detailed Description	50
5.14.2 Constructor & Destructor Documentation	51
5.14.2.1 CoinFlipDialog()	51
5.14.3 Member Function Documentation	51
5.14.3.1 onCreate()	51
5.14.4 Member Data Documentation	51
5.14.4.1 RANDOM	51
5.15 DamageCalculator.Color Enum Reference	51
5.15.1 Detailed Description	52
5.15.2 Member Data Documentation	52
5.15.2.1 BUFFED	52
5.15.2.2 DEBUFFED	52
5.15.2.3 DEFAULT	52
5.16 DamageCalculator Interface Reference	52
5.16.1 Detailed Description	53
5.16.2 Member Function Documentation	53
5.16.2.1 calculateDamage()	53
5.16.2.2 isBuffered()	54
5.17 DamageCalculatorBuildDirector Class Reference	54
5.17.1 Detailed Description	55
5.17.2 Member Function Documentation	55
5.17.2.1 getCalculator()	55

5.17.2.2 setHorns()	56
5.17.2.3 setMoralBoosts()	56
5.17.2.4 setSquads()	56
5.18 DamageCalculatorBuilder Class Reference	57
5.18.1 Detailed Description	58
5.18.2 Member Function Documentation	58
5.18.2.1 getResult()	58
5.18.2.2 setBond()	58
5.18.2.3 setHorn()	58
5.18.2.4 setMoral()	59
5.18.2.5 setWeather()	59
5.18.3 Member Data Documentation	59
5.18.3.1 calculator	59
5.19 DamageCalculatorDecorator Class Reference	59
5.19.1 Detailed Description	60
5.19.2 Constructor & Destructor Documentation	61
5.19.2.1 DamageCalculatorDecorator()	61
5.19.3 Member Data Documentation	61
5.19.3.1 component	61
5.20 DamageCalculatorUseCase Class Reference	61
5.20.1 Detailed Description	62
5.20.2 Member Function Documentation	62
5.20.2.1 getDamageCalculator() [1/2]	62
5.20.2.2 getDamageCalculator() [2/2]	62
5.21 DamageValuePicker Class Reference	63
5.21.1 Detailed Description	64
5.21.2 Constructor & Destructor Documentation	64
5.21.2.1 DamageValuePicker()	64
5.21.3 Member Function Documentation	64
5.21.3.1 getDisplayString()	64
5.21.3.2 getValue()	65
5.21.3.3 setEpicValues()	65
5.21.3.4 setSelectableValues()	65
5.21.3.5 setValue()	65
5.21.4 Member Data Documentation	66
5.21.4.1 epicValues	66
5.22 ResetDialogUseCase.DialogType Enum Reference	66
5.22.1 Detailed Description	66
5.22.2 Member Data Documentation	67
5.22.2.1 DEFAULT	67
5.22.2.2 MONSTER	67
5.22.2.3 NONE	67

5.23 FactionSwitchListener Class Reference	67
5.23.1 Detailed Description	69
5.23.2 Constructor & Destructor Documentation	69
5.23.2.1 FactionSwitchListener()	69
5.23.3 Member Function Documentation	69
5.23.3.1 getListener()	69
5.23.3.2 onSharedPreferenceChanged()	70
5.23.3.3 setTheme()	70
5.23.4 Member Data Documentation	70
5.23.4.1 ballViews	70
5.23.4.2 cardViews	71
5.23.4.3 factionButton	71
5.23.4.4 THEME_MONSTER	71
5.23.4.5 THEME_NILFGAARD	71
5.23.4.6 THEME_NORTHERN_KINGDOMS	71
5.23.4.7 THEME_PREFERENCE_KEY	72
5.23.4.8 THEME_SCOIATAEL	72
5.23.4.9 unitViews	72
5.24 GameBoardViewModel Class Reference	72
5.24.1 Detailed Description	74
5.24.2 Constructor & Destructor Documentation	74
5.24.2.1 GameBoardViewModel()	74
5.24.3 Member Function Documentation	74
5.24.3.1 getMenuUiState()	74
5.24.3.2 getModel()	75
5.24.3.3 getRepository()	75
5.24.3.4 getRowUiState()	75
5.24.3.5 getSoundManager()	76
5.24.3.6 onBurnButtonPressed()	76
5.24.3.7 onFactionSwitchReset()	76
5.24.3.8 onHornViewPressed()	77
5.24.3.9 onResetButtonPressed()	77
5.24.3.10 onWeatherButtonPressed()	78
5.24.3.11 onWeatherViewPressed()	78
5.24.3.12 reset()	78
5.24.4 Member Data Documentation	79
5.24.4.1 INITIALIZER	79
5.24.4.2 menuUiState	79
5.24.4.3 rowUiStates	79
5.24.4.4 soundManager	80
5.25 GwentApplication Class Reference	80
5.25.1 Detailed Description	81

5.25.2 Member Function Documentation	81
5.25.2.1 getRepository() [1/2]	81
5.25.2.2 getRepository() [2/2]	81
5.25.2.3 onCreate()	82
5.25.3 Member Data Documentation	82
5.25.3.1 database	82
5.25.3.2 repository	82
5.26 HornDamageCalculatorDecorator Class Reference	82
5.26.1 Detailed Description	84
5.26.2 Constructor & Destructor Documentation	84
5.26.2.1 HornDamageCalculatorDecorator()	84
5.26.3 Member Function Documentation	85
5.26.3.1 calculateDamage()	85
5.26.3.2 doubleDamage()	85
5.26.3.3 isBuffed()	85
5.26.4 Member Data Documentation	86
5.26.4.1 unitIds	86
5.27 ImageViewSwitchAnimator Class Reference	86
5.27.1 Detailed Description	87
5.27.2 Member Function Documentation	87
5.27.2.1 animatedSwitch()	87
5.27.2.2 getOverlayView()	87
5.28 IndicatorManager Class Reference	88
5.28.1 Detailed Description	88
5.28.2 Constructor & Destructor Documentation	88
5.28.2.1 IndicatorManager()	88
5.28.3 Member Function Documentation	89
5.28.3.1 updateIndicators()	89
5.28.4 Member Data Documentation	89
5.28.4.1 currentView	89
5.28.4.2 indicators	89
5.29 IntroductionActivity Class Reference	90
5.29.1 Detailed Description	90
5.29.2 Member Function Documentation	91
5.29.2.1 onCreate()	91
5.29.3 Member Data Documentation	91
5.29.3.1 indicatorManager	91
5.30 IntroductionFragment Class Reference	91
5.30.1 Detailed Description	93
5.30.2 Member Function Documentation	93
5.30.2.1 newInstance()	93
5.30.2.2 onCreateView()	93

5.30.3 Member Data Documentation	93
5.30.3.1 ARG_SECTION_NUMBER	93
5.30.3.2 imagelds	94
5.30.3.3 PAGES_COUNT	94
5.30.3.4 stringIds	94
5.31 MainActivity Class Reference	94
5.31.1 Detailed Description	96
5.31.2 Member Function Documentation	96
5.31.2.1 inflateCoinFlipPopup()	96
5.31.2.2 inflateFactionPopup()	96
5.31.2.3 initializeViewModel()	96
5.31.2.4 onCreate()	96
5.31.2.5 onDestroy()	97
5.31.2.6 onResume()	97
5.31.2.7 onWindowFocusChanged()	97
5.31.3 Member Data Documentation	97
5.31.3.1 disposables	97
5.31.3.2 factionSwitchListener	97
5.31.3.3 gameBoardViewModel	97
5.32 MenuUiState Class Reference	98
5.32.1 Detailed Description	98
5.32.2 Constructor & Destructor Documentation	99
5.32.2.1 MenuUiState()	99
5.32.3 Member Function Documentation	99
5.32.3.1 equals()	99
5.32.3.2 getDamage()	99
5.32.3.3 isBurn()	99
5.32.3.4 isReset()	99
5.32.3.5 isWeather()	100
5.32.4 Member Data Documentation	100
5.32.4.1 burn	100
5.32.4.2 damage	100
5.32.4.3 reset	100
5.32.4.4 weather	100
5.33 MenuUiStateObserver Class Reference	100
5.33.1 Detailed Description	102
5.33.2 Constructor & Destructor Documentation	102
5.33.2.1 MenuUiStateObserver()	102
5.33.3 Member Function Documentation	102
5.33.3.1 accept()	102
5.33.4 Member Data Documentation	102
5.33.4.1 burnButton	102

5.33.4.2 damageView	103
5.33.4.3 resetButton	103
5.33.4.4 weatherButton	103
5.34 MoralDamageCalculatorDecorator Class Reference	103
5.34.1 Detailed Description	104
5.34.2 Constructor & Destructor Documentation	105
5.34.2.1 MoralDamageCalculatorDecorator()	105
5.34.3 Member Function Documentation	105
5.34.3.1 calculateDamage()	105
5.34.3.2 isBuffed()	106
5.34.4 Member Data Documentation	106
5.34.4.1 unitIds	106
5.35 OnValueChangedListener< T extends Comparable Interface Template Reference	106
5.35.1 Detailed Description	107
5.36 OverlayDialog Class Reference	107
5.36.1 Detailed Description	108
5.36.2 Constructor & Destructor Documentation	108
5.36.2.1 OverlayDialog() [1/2]	108
5.36.2.2 OverlayDialog() [2/2]	108
5.36.3 Member Function Documentation	109
5.36.3.1 onCreate()	109
5.36.4 Member Data Documentation	109
5.36.4.1 BACKGROUND	109
5.36.4.2 dismissViewId	109
5.36.4.3 layout	109
5.36.4.4 NO_DISMISS_VIEW	109
5.37 RemoveUnitsUseCase Class Reference	110
5.37.1 Detailed Description	110
5.37.2 Member Function Documentation	110
5.37.2.1 getRevengeDialog()	110
5.37.2.2 remove() [1/2]	111
5.37.2.3 remove() [2/2]	112
5.38 ResetAlertDialogBuilderAdapter Class Reference	112
5.38.1 Detailed Description	113
5.38.2 Constructor & Destructor Documentation	113
5.38.2.1 ResetAlertDialogBuilderAdapter()	113
5.38.3 Member Function Documentation	113
5.38.3.1 create()	113
5.38.3.2 setMonsterDialog()	114
5.38.3.3 setTrigger()	114
5.38.4 Member Data Documentation	114
5.38.4.1 adapteeBuilder	114

5.38.4.2 dialogCallback	114
5.39 ResetDialogUseCase Class Reference	115
5.39.1 Detailed Description	115
5.39.2 Member Function Documentation	116
5.39.2.1 getDialogType()	116
5.39.2.2 reset() [1/2]	116
5.39.2.3 reset() [2/2]	117
5.40 ResetRepositoryUseCase Class Reference	117
5.40.1 Detailed Description	118
5.40.2 Member Function Documentation	118
5.40.2.1 getRandomUnit()	118
5.40.2.2 getRevengeDialog()	118
5.40.2.3 reset() [1/2]	119
5.40.2.4 reset() [2/2]	119
5.41 RevengeAlertDialogBuilderAdapter Class Reference	120
5.41.1 Detailed Description	121
5.41.2 Constructor & Destructor Documentation	121
5.41.2.1 RevengeAlertDialogBuilderAdapter()	121
5.41.3 Member Function Documentation	121
5.41.3.1 create()	121
5.41.3.2 insertAvengers()	121
5.41.3.3 setNegativeCallback()	122
5.41.3.4 setPositiveCallback()	122
5.41.4 Member Data Documentation	123
5.41.4.1 adapteeBuilder	123
5.41.4.2 AVENGER_ABILITY	123
5.41.4.3 AVENGER_DAMAGE	123
5.41.4.4 AVENGER_EPIC	123
5.41.4.5 AVENGER_ROW	123
5.41.4.6 AVENGER_SQUAD	123
5.42 RowDao Interface Reference	124
5.42.1 Detailed Description	124
5.42.2 Member Function Documentation	125
5.42.2.1 clearRows()	125
5.42.2.2 clearWeather()	125
5.42.2.3 insertRow()	125
5.42.2.4 isHorn()	125
5.42.2.5 isHornFlowable()	126
5.42.2.6 isWeather()	126
5.42.2.7 isWeatherFlowable()	126
5.42.2.8 updateHorn()	127
5.42.2.9 updateWeather()	127

5.43 RowEntity Class Reference	127
5.43.1 Detailed Description	128
5.43.2 Constructor & Destructor Documentation	128
5.43.2.1 RowEntity()	128
5.43.3 Member Function Documentation	130
5.43.3.1 getId()	130
5.43.3.2 isHorn()	130
5.43.3.3 isWeather()	130
5.43.3.4 setHorn()	130
5.43.3.5 setWeather()	130
5.43.4 Member Data Documentation	131
5.43.4.1 horn	131
5.43.4.2 id	131
5.43.4.3 weather	131
5.44 RowType Enum Reference	131
5.44.1 Detailed Description	132
5.44.2 Member Data Documentation	132
5.44.2.1 MELEE	132
5.44.2.2 RANGE	132
5.44.2.3 SIEGE	132
5.45 RowUiState Class Reference	132
5.45.1 Detailed Description	133
5.45.2 Constructor & Destructor Documentation	133
5.45.2.1 RowUiState()	133
5.45.3 Member Function Documentation	133
5.45.3.1 equals()	133
5.45.3.2 getDamage()	133
5.45.3.3 getUnits()	134
5.45.3.4 isHorn()	134
5.45.3.5 isWeather()	134
5.45.4 Member Data Documentation	134
5.45.4.1 damage	134
5.45.4.2 horn	134
5.45.4.3 units	135
5.45.4.4 weather	135
5.46 RowUiStateObserver Class Reference	135
5.46.1 Detailed Description	136
5.46.2 Constructor & Destructor Documentation	136
5.46.2.1 RowUiStateObserver()	136
5.46.3 Member Function Documentation	137
5.46.3.1 accept()	137
5.46.3.2 getObserver()	137

5.46.4 Member Data Documentation	138
5.46.4.1 damageView	138
5.46.4.2 hornView	138
5.46.4.3 unitView	138
5.46.4.4 weatherResource	138
5.46.4.5 weatherView	138
5.47 RuleActivity Class Reference	138
5.47.1 Detailed Description	139
5.47.2 Member Function Documentation	139
5.47.2.1 onCreate()	139
5.47.2.2 onOptionsItemSelected()	140
5.47.3 Member Data Documentation	140
5.47.3.1 INTENT_EXTRA_TAG	140
5.48 RuleSection Enum Reference	140
5.48.1 Detailed Description	141
5.48.2 Member Data Documentation	141
5.48.2.1 CARD_ABILITIES	141
5.48.2.2 CARDS	141
5.48.2.3 COMMANDER	141
5.48.2.4 COURSE	141
5.48.2.5 FACTIONS	141
5.48.2.6 GENERAL	141
5.48.2.7 SPECIAL_CARDS	142
5.49 SectionsPagerAdapter Class Reference	142
5.49.1 Detailed Description	143
5.49.2 Constructor & Destructor Documentation	143
5.49.2.1 SectionsPagerAdapter()	143
5.49.3 Member Function Documentation	143
5.49.3.1 createFragment()	143
5.49.3.2 getItemCount()	143
5.50 SettingsActivity Class Reference	143
5.50.1 Detailed Description	145
5.50.2 Member Function Documentation	145
5.50.2.1 onCreate()	145
5.50.2.2 onOptionsItemSelected()	145
5.50.2.3 onPreferenceStartFragment()	145
5.50.2.4 onSaveInstanceState()	146
5.50.3 Member Data Documentation	146
5.50.3.1 CURRENT_FRAGMENT_KEY	146
5.51 SettingsHeaderFragment Class Reference	146
5.51.1 Detailed Description	147
5.51.2 Member Function Documentation	147

5.51.2.1 onCreatePreferences()	147
5.52 SettingsRuleFragment Class Reference	147
5.52.1 Detailed Description	148
5.52.2 Member Function Documentation	148
5.52.2.1 getSectionClickListener()	148
5.52.2.2 onCreatePreferences()	149
5.53 SettingsSoundFragment Class Reference	149
5.53.1 Detailed Description	150
5.53.2 Member Function Documentation	150
5.53.2.1 onCreatePreferences()	150
5.54 ShowUnitsDialog Class Reference	151
5.54.1 Detailed Description	152
5.54.2 Constructor & Destructor Documentation	153
5.54.2.1 ShowUnitsDialog()	153
5.54.3 Member Function Documentation	153
5.54.3.1 getDialog()	153
5.54.3.2 getRow()	153
5.54.3.3 onCreate()	154
5.54.4 Member Data Documentation	154
5.54.4.1 cardListAdapter	154
5.54.4.2 disposables	154
5.54.4.3 row	154
5.54.4.4 soundManager	155
5.54.4.5 unitRecyclerView	155
5.55 Sound Class Reference	155
5.55.1 Detailed Description	156
5.55.2 Constructor & Destructor Documentation	156
5.55.2.1 Sound()	156
5.55.3 Member Function Documentation	156
5.55.3.1 createSound()	156
5.55.3.2 getPreferenceKey()	157
5.55.3.3 getSoundId()	157
5.55.3.4 isActivated()	157
5.55.3.5 setActivated() [1/2]	157
5.55.3.6 setActivated() [2/2]	158
5.55.4 Member Data Documentation	158
5.55.4.1 activated	158
5.55.4.2 preferenceKey	158
5.55.4.3 soundId	158
5.56 SoundManager Class Reference	158
5.56.1 Detailed Description	160
5.56.2 Constructor & Destructor Documentation	160

5.56.2.1 SoundManager()	160
5.56.3 Member Function Documentation	161
5.56.3.1 playBurnSound()	161
5.56.3.2 playCardAddSound()	161
5.56.3.3 playCardRemovedSound()	161
5.56.3.4 playClearWeatherSound()	161
5.56.3.5 playCoinSound()	162
5.56.3.6 playHornSound()	162
5.56.3.7 playResetSound()	162
5.56.3.8 playSound()	162
5.56.3.9 playWeatherSound()	163
5.56.4 Member Data Documentation	163
5.56.4.1 changeListener	163
5.56.4.2 SOUND_BURN	163
5.56.4.3 SOUND_CARDS_EPIC	163
5.56.4.4 SOUND_CARDS_MELEE	164
5.56.4.5 SOUND_CARDS_RANGE	164
5.56.4.6 SOUND_CARDS_SIEGE	164
5.56.4.7 SOUND_COIN	164
5.56.4.8 SOUND_HORN	164
5.56.4.9 SOUND_RESET	165
5.56.4.10 SOUND_WEATHER_FOG	165
5.56.4.11 SOUND_WEATHER_FROST	165
5.56.4.12 SOUND_WEATHER_GOOD	165
5.56.4.13 SOUND_WEATHER_RAIN	165
5.56.4.14 soundPool	166
5.56.4.15 sounds	166
5.57 SquadManager Class Reference	166
5.57.1 Detailed Description	167
5.57.2 Constructor & Destructor Documentation	167
5.57.2.1 SquadManager()	167
5.57.3 Member Function Documentation	167
5.57.3.1 getFirstSquadWithMembers()	167
5.57.3.2 onSquadChanged()	167
5.57.4 Member Data Documentation	168
5.57.4.1 MAX_NR_SQUADS	168
5.57.4.2 states	168
5.58 SquadState Class Reference	168
5.58.1 Detailed Description	169
5.58.2 Constructor & Destructor Documentation	169
5.58.2.1 SquadState()	169
5.58.3 Member Function Documentation	170

5.58.3.1 getMemberBaseDamage()	170
5.58.3.2 getSquadMembers()	170
5.58.3.3 getSquadNumber()	170
5.58.3.4 getState()	170
5.58.3.5 hasMembers()	171
5.58.4 Member Data Documentation	171
5.58.4.1 memberBaseDamage	171
5.58.4.2 squadMembers	171
5.58.4.3 squadNumber	172
5.59 StringValuePicker< T extends Comparable Class Template Reference	172
5.59.1 Detailed Description	172
5.60 ResetDialogUseCase.Trigger Enum Reference	172
5.60.1 Detailed Description	173
5.60.2 Member Data Documentation	173
5.60.2.1 BUTTON_CLICK	173
5.60.2.2 FACTION_SWITCH	173
5.61 UnitDao Interface Reference	173
5.61.1 Detailed Description	174
5.61.2 Member Function Documentation	174
5.61.2.1 countUnits() [1/2]	174
5.61.2.2 countUnits() [2/2]	174
5.61.2.3 deleteUnit()	174
5.61.2.4 deleteUnits()	175
5.61.2.5 getUnit()	175
5.61.2.6 getUnits() [1/2]	175
5.61.2.7 getUnits() [2/2]	175
5.61.2.8 getUnitsFlowable()	176
5.61.2.9 hasNonEpicUnitsFlowable()	176
5.61.2.10 insertUnit() [1/2]	176
5.61.2.11 insertUnit() [2/2]	176
5.62 UnitEntity Class Reference	177
5.62.1 Detailed Description	179
5.62.2 Constructor & Destructor Documentation	179
5.62.2.1 UnitEntity()	179
5.62.3 Member Function Documentation	179
5.62.3.1 calculateDamage()	179
5.62.3.2 collectionToString()	180
5.62.3.3 getAbility()	180
5.62.3.4 getDamage()	180
5.62.3.5 getId()	180
5.62.3.6 getRow()	181
5.62.3.7 getSquad()	181

5.62.3.8 isBuffed()	181
5.62.3.9 isEpic()	181
5.62.3.10 setAbility()	182
5.62.3.11 setDamage()	182
5.62.3.12 setEpic()	182
5.62.3.13 setId()	182
5.62.3.14 setRow()	183
5.62.3.15 setSquad()	183
5.62.3.16 toString()	183
5.62.4 Member Data Documentation	183
5.62.4.1 ability	183
5.62.4.2 damage	184
5.62.4.3 epic	184
5.62.4.4 EPIC_DAMAGE_VALUES	184
5.62.4.5 id	184
5.62.4.6 NON_EPIC_DAMAGE_VALUES_UPPER_BOUND	184
5.62.4.7 row	184
5.62.4.8 squad	184
5.63 UnitRepository Class Reference	184
5.63.1 Detailed Description	186
5.63.2 Constructor & Destructor Documentation	186
5.63.2.1 UnitRepository()	186
5.63.3 Member Function Documentation	187
5.63.3.1 clearWeather()	187
5.63.3.2 copy()	187
5.63.3.3 countUnits() [1/2]	187
5.63.3.4 countUnits() [2/2]	187
5.63.3.5 delete() [1/2]	188
5.63.3.6 delete() [2/2]	188
5.63.3.7 getRepository()	188
5.63.3.8 getUnit()	189
5.63.3.9 getUnits() [1/2]	189
5.63.3.10 getUnits() [2/2]	189
5.63.3.11 getUnitsFlowable()	190
5.63.3.12 hasNonEpicUnitsFlowable()	190
5.63.3.13 initializeRows()	190
5.63.3.14 insertUnit() [1/2]	190
5.63.3.15 insertUnit() [2/2]	191
5.63.3.16 isHorn()	192
5.63.3.17 isHornFlowable()	192
5.63.3.18 isWeather()	192
5.63.3.19 isWeatherFlowable()	193

5.63.3.20 reset() [1/2]	193
5.63.3.21 reset() [2/2]	193
5.63.3.22 switchHorn()	194
5.63.3.23 switchWeather()	194
5.63.4 Member Data Documentation	194
5.63.4.1 database	194
5.64 ValuePicker< T extends Comparable Class Template Reference	195
5.64.1 Detailed Description	195
5.65 WeatherDamageCalculator Class Reference	195
5.65.1 Detailed Description	196
5.65.2 Constructor & Destructor Documentation	197
5.65.2.1 WeatherDamageCalculator()	197
5.65.3 Member Function Documentation	197
5.65.3.1 calculateDamage()	197
5.65.3.2 isBuffed()	197
5.65.4 Member Data Documentation	198
5.65.4.1 weather	198
6 File Documentation	199
6.1 /home/runner/work/Gwent/Gwent/app/src/main/AndroidManifest.xml	199
6.2 Ability.java	199
6.3 AppDatabase.java	200
6.4 RowDao.java	200
6.5 RowEntity.java	200
6.6 RowType.java	201
6.7 UnitDao.java	201
6.8 UnitEntity.java	202
6.9 UnitRepository.java	204
6.10 BurnDialogUseCase.java	207
6.11 DamageCalculatorUseCase.java	208
6.12 RemoveUnitsUseCase.java	209
6.13 ResetAlertDialogBuilderAdapter.java	210
6.14 ResetDialogUseCase.java	210
6.15 ResetRepositoryUseCase.java	212
6.16 RevengeAlertDialogBuilderAdapter.java	213
6.17 BondDamageCalculatorDecorator.java	214
6.18 DamageCalculator.java	214
6.19 DamageCalculatorBuildDirector.java	214
6.20 DamageCalculatorBuilder.java	215
6.21 DamageCalculatorDecorator.java	216
6.22 HornDamageCalculatorDecorator.java	216
6.23 MoralDamageCalculatorDecorator.java	216

6.24 WeatherDamageCalculator.java	217
6.25 GwentApplication.java	217
6.26 AddCardDialog.java	218
6.27 CardNumberPickerAdapter.java	219
6.28 DamageValuePicker.java	220
6.29 SquadManager.java	221
6.30 SquadState.java	222
6.31 StringValuePicker.java	223
6.32 ValuePicker.java	224
6.33 CardListAdapter.java	225
6.34 CardUiState.java	227
6.35 CardUiStateFactory.java	228
6.36 ShowUnitsDialog.java	229
6.37 ChangeFactionDialog.java	231
6.38 CoinFlipDialog.java	232
6.39 OverlayDialog.java	232
6.40 IndicatorManager.java	233
6.41 IntroductionActivity.java	233
6.42 IntroductionFragment.java	234
6.43 SectionsPagerAdapter.java	235
6.44 FactionSwitchListener.java	235
6.45 GameBoardViewModel.java	237
6.46 ImageViewSwitchAnimator.java	239
6.47 MainActivity.java	240
6.48 MenuUiState.java	243
6.49 MenuUiStateObserver.java	243
6.50 RowUiState.java	244
6.51 RowUiStateObserver.java	245
6.52 RuleActivity.java	246
6.53 RuleSection.java	247
6.54 SettingsActivity.java	247
6.55 SettingsHeaderFragment.java	248
6.56 SettingsRuleFragment.java	248
6.57 SettingsSoundFragment.java	249
6.58 Sound.java	250
6.59 SoundManager.java	250
6.60 background_drawable.xml	252
6.61 icon_copy.xml	253
6.62 icon_delete.xml	253
6.63 icon_design.xml	253
6.64 icon_explore.xml	254
6.65 icon_factionreset.xml	254

6.66 icon_next.xml	254
6.67 icon_rules.xml	254
6.68 icon_sounds.xml	254
6.69 icon_warnings.xml	255
6.70 indicator_selected.xml	255
6.71 indicator_unselected.xml	255
6.72 rectangle.xml	255
6.73 rectangle_black_transparent.xml	255
6.74 rectangle_cancel_popup.xml	255
6.75 fragment_introduction.xml	256
6.76 fragment_introduction.xml	256
6.77 fragment_introduction.xml	257
6.78 fragment_introduction.xml	257
6.79 activity_rules.xml	258
6.80 activity_rules.xml	258
6.81 activity_introduction.xml	259
6.82 activity_main.xml	260
6.83 activity_settings.xml	262
6.84 alertdialog_checkbox.xml	263
6.85 card.xml	263
6.86 popup_add_card.xml	264
6.87 popup_cards.xml	266
6.88 popup_coin_normal.xml	267
6.89 popup_coin_stewie.xml	267
6.90 popup_coin_terry.xml	269
6.91 popup_coin_vin.xml	270
6.92 popup_faction.xml	270
6.93 row.xml	272
6.94 icon_launcher.xml	273
6.95 icon_launcher_round.xml	273
6.96 arrays.xml	273
6.97 attr.xml	274
6.98 colors.xml	274
6.99 dims.xml	274
6.100 icon_launcher_background.xml	275
6.101 preferences.xml	275
6.102 strings.xml	275
6.103 strings.xml	279
6.104 styles.xml	283
6.105 header_preferences.xml	284
6.106 rule_preferences.xml	285
6.107 sound_preferences.xml	285

Chapter 1

Gwent - A Calculator

-
1. Overview
 2. Printing analog cards
 3. Documentation
-

1.1 Overview

Gwent - A Calculator is an Android app ([available in Play Store](#)) that enables the user to play the Gwent card game from The Witcher 3: Wild Hunt with analog cards. This is accomplished by providing an easy-to-use interface which allows adding cards with certain attributes that correspond to the cards played analogously. The app then calculates the power score of the player including weather effects and card abilities. To see what this looks like, please visit the [Play Store entry](#)!

1.2 Printing analog cards

To actually play Gwent using *Gwent - A Calculator*, you need to print out analog Gwent cards. The card images can be downloaded from the following links:

Language	zip	tar
English	gwent_cards_en.zip	gwent_cards_en.tar.gz
German	gwent_cards_de.zip	gwent_cards_de.tar.gz

Please be aware that the Skellige faction is only available in the german packages! [This PDF](#) summarizes in which quantity you need to print which card.

When you installed the app, printed out your deck and found a friend, you are ready to go and can start playing the original Gwent game from The Witcher 3: Wild Hunt.

1.3 Documentation

If you are a developer and interested in improving the app by adding new features or fixing bugs, you are always welcome to open pull requests. An extensive reference specification of the whole app can be found on [this website](#) or in [this PDF-file](#).

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Ability	13
AndroidViewModel	
GameBoardViewModel	72
AppCompatActivity	
IntroductionActivity	90
MainActivity	94
RuleActivity	138
SettingsActivity	143
Application	
GwentApplication	80
BurnDialogUseCase	22
ResetAlertDialogBuilderAdapter.Callback	24
ChangeFactionDialog.Callback	25
CardNumberPickerAdapter	30
CardUiState	33
CardUiStateFactory	40
DamageCalculator.Color	51
Consumer	
MenuUiStateObserver	100
RowUiStateObserver	135
DamageCalculator	52
DamageCalculatorDecorator	59
BondDamageCalculatorDecorator	19
HornDamageCalculatorDecorator	82
MoralDamageCalculatorDecorator	103
WeatherDamageCalculator	195
DamageCalculatorBuildDirector	54
DamageCalculatorBuilder	57
DamageCalculatorUseCase	61
Dialog	
OverlayDialog	107
ChangeFactionDialog	46
CoinFlipDialog	49
AddCardDialog	14
ShowUnitsDialog	151
ResetDialogUseCase.DialogType	66
Fragment	
IntroductionFragment	91
FragmentStateAdapter	
SectionsPagerAdapter	142

ImageViewSwitchAnimator	86
IndicatorManager	88
ListAdapter	
CardListAdapter	26
MenuUiState	98
OnPreferenceStartFragmentCallback	
SettingsActivity	143
OnSharedPreferenceChangeListener	
FactionSwitchListener	67
OnValueChangeListener< T extends Comparable	106
PreferenceFragmentCompat	
SettingsHeaderFragment	146
SettingsRuleFragment	147
SettingsSoundFragment	149
RemoveUnitsUseCase	110
ResetAlertDialogBuilderAdapter	112
ResetDialogUseCase	115
ResetRepositoryUseCase	117
RevengeAlertDialogBuilderAdapter	120
RoomDatabase	
AppDatabase	18
RowDao	124
RowEntity	127
RowType	131
RowUiState	132
RuleSection	140
Sound	155
SoundManager	158
SquadManager	166
SquadState	168
StringValuePicker< T extends Comparable	172
ResetDialogUseCase.Trigger	172
UnitDao	173
UnitEntity	177
UnitRepository	184
ValuePicker	
DamageValuePicker	63
ValuePicker< T extends Comparable	195
ViewHolder	
CardListAdapter.CardViewHolder	42

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Ability	An enum listing the different card abilities possible	13
AddCardDialog	An OverlayDialog used to add new UnitEntity objects to a certain #row	14
AppDatabase	An abstract class extending RoomDatabase and representing the SQLite database of the application	18
BondDamageCalculatorDecorator	A DamageCalculator class responsible for calculating the tight bond buff if necessary, i.e., if there are units with the com.paternaggschga.gwent.data.Ability::BINDING ability	19
BurnDialogUseCase	A use case class responsible for dispatching a remove call to RemoveUnitsUseCase	22
ResetAlertDialogBuilderAdapter.Callback	An interface defining functions to propagate the user's decision back to the creator	24
ChangeFactionDialog.Callback	An interface used for propagating a selected theme back to the creator of a ChangeFactionDialog	25
CardListAdapter	A ListAdapter used to provide CardListAdapter.CardViewHolder objects created from CardUi↔State objects to a RecyclerView	26
CardNumberPickerAdapter	A helper class responsible for initializing the NumberPicker views of an AddCardDialog (in #↔CardNumberPickerAdapter()) and adding UnitEntity objects with the selected attributes when #addSelectedUnits() is called	30
CardUiState	A data class encapsulating the visible state of a card in the card list shown by the ShowUnits↔Dialog	33
CardUiStateFactory	A factory class responsible for creating CardUiState objects from UnitEntity objects	40
CardListAdapter.CardViewHolder	A RecyclerView.ViewHolder class managing a card view that shows a representation of the UnitEntity referenced by #itemId	42
ChangeFactionDialog	An OverlayDialog class used to change the faction design	46
CoinFlipDialog	An OverlayDialog class showing the result of a coin-flip	49
DamageCalculator.Color	An enum containing values representing whether a unit is buffed, de-buffed or not affected by status effects	51

DamageCalculator	An interface for a class capable of calculating the (de-)buffed damage of a unit for a given Unit↔ Entity::id and UnitEntity::damage	52
DamageCalculatorBuildDirector	A build director responsible for the creation of DamageCalculator instances from sets of units using DamageCalculatorBuilder	54
DamageCalculatorBuilder	A builder class responsible for creating a decorator structure of DamageCalculator classes with the given parameters	57
DamageCalculatorDecorator	An abstract DamageCalculator that is used as a superclass for the decorator classes implementing the damage calculation	59
DamageCalculatorUseCase	A use case class responsible for creating DamageCalculator objects for rows using Damage↔ CalculatorBuildDirector	61
DamageValuePicker	A ValuePicker used for selection of damage values for UnitEntity::damage	63
ResetDialogUseCase.DialogType	An Enum defining which form of Dialog should be shown	66
FactionSwitchListener	A class implementing SharedPreferences.OnSharedPreferencesChangeListener for the #↔ THEME_PREFERENCE_KEY SharedPreferences	67
GameBoardViewModel	An AndroidViewModel class responsible for encapsulating and offering state of views in activity_main.xml, i.e., that show the overall game board	72
GwentApplication	An Application encapsulating the Gwent app	80
HornDamageCalculatorDecorator	A DamageCalculator class responsible for calculating the horn buff if necessary, i.e., if there are units with the com.peternaggschga.gwent.data.Ability::HORN ability that are not the unit the damage is calculated for or if there is a commander's horn in this row	82
ImageViewSwitchAnimator	A class providing functionality for changing the resource shown by an ImageView using a fading animation	86
IndicatorManager	A class managing the indicator views shown in the bottom bar of the IntroductionActivity 88	
IntroductionActivity	An AppCompatActivity that gives the user an introduction into the usage of the application	90
IntroductionFragment	A Fragment containing an ImageView and a TextView conveying introductory information	91
MainActivity	An AppCompatActivity that is called on startup and that encapsulates the main view onto the game board	94
MenuUiState	A data class encapsulating the visible state of the menu in the main view of the application . .	98
MenuUiStateObserver	An observer class responsible for updating the menu views when notified, i.e., when a new MenuUiState is produced by the subscribed Flowable	100
MoralDamageCalculatorDecorator	A DamageCalculator class responsible for calculating the moral buff if necessary, i.e., if there are units with the com.peternaggschga.gwent.data.Ability::MORAL_BOOST ability that are not the unit the damage is calculated for	103
OnValueChangeListener< T extends Comparable	An interface defining a callback for changes of the current value	106
OverlayDialog	A Dialog class which is used for popups that are shown on top of the calling Activity	107

RemoveUnitsUseCase	A use case class responsible for removing units from a <code>UnitRepository</code>	110
ResetAlertDialogBuilderAdapter	An adapter class adapting <code>AlertDialog.Builder</code> to provide an interface for creating an <code>AlertDialog</code> asking the user whether he really wants to reset	112
ResetDialogUseCase	A use case class responsible for dispatching a reset call to <code>ResetRepositoryUseCase</code> , possibly after a confirmation by the user obtained from a <code>Dialog</code>	115
ResetRepositoryUseCase	A use case class responsible for resetting the <code>UnitRepository</code>	117
RevengeAlertDialogBuilderAdapter	An adapter class adapting <code>AlertDialog.Builder</code> to provide an interface for creating an <code>AlertDialog</code> asking the user whether they want to invoke the <code>Ability::REVENGE</code> ability	120
RowDao	An interface defining data access operations related to <code>rows</code> table	124
RowEntity	A class representing the state of an attack row, i.e., weather and commanders horn	127
RowType	An enum listing the different types of attack rows	131
RowUiState	A data class encapsulating the visible state of a row in the main view of the application	132
RowUiStateObserver	An observer class responsible for updating the views of the row defined in <code>#row</code> when notified, i.e., when a new <code>RowUiState</code> is produced the observed <code>Flowable</code>	135
RuleActivity	An <code>AppCompatActivity</code> used to present the rules of the game	138
RuleSection	An Enum used to discern the different possible sections shown by <code>RuleActivity</code>	140
SectionsPagerAdapter	A <code>FragmentStateAdapter</code> that returns an <code>IntroductionFragment</code> corresponding to one of the sections/tabs/pages	142
SettingsActivity	An <code>AppCompatActivity</code> implementing <code>PreferenceFragmentCompat.OnPreferenceStartFragment</code> that is used by the user to manage the <code>SharedPreferences</code> of the application	143
SettingsHeaderFragment	A <code>PreferenceFragmentCompat</code> class encapsulating the main preference screen, i.e., the Preferences defined in <code>R.xml#header_preferences</code>	146
SettingsRuleFragment	A <code>PreferenceFragmentCompat</code> class encapsulating the rule preference screen, i.e., the rule sections defined in <code>R.xml#rule_preferences</code>	147
SettingsSoundFragment	A <code>PreferenceFragmentCompat</code> class encapsulating the sound preference screen, i.e., the Preferences defined in <code>R.xml#sound_preferences</code>	149
ShowUnitsDialog	An <code>OverlayDialog</code> used to list the units of a certain row and enabling the user to copy, add, or delete new units	151
Sound	A data class encapsulating information on a certain sound effect, e.g., its id in a <code>SoundPool</code> and whether it is activated in settings	155
SoundManager	A class responsible for initializing and playing sounds when they are enabled	158
SquadManager	A class used by the <code>CardNumberPickerAdapter</code> to determine the first squad with members or set the default damage according to the squads base damage	166
SquadState	A data class encapsulating information about the squad defined by <code>#squadNumber</code> , i.e., how many <code>#squadMembers</code> are there and what is the <code>#memberBaseDamage</code> of units in this squad	168

StringValuePicker< T extends Comparable	
A ValuePicker displaying a resource String for each element in selectableValues	172
ResetDialogUseCase.Trigger	
An Enum listing the possible triggers of a reset	172
UnitDao	
An interface defining data access operations related to units table	173
UnitEntity	
A class representing a card on the game board	177
UnitRepository	
A facade class managing public access to the data layer	184
ValuePicker< T extends Comparable	
A generic abstract wrapper class around NumberPicker used to select values of arbitrary types extending Comparable	195
WeatherDamageCalculator	
A DamageCalculator class responsible for calculating the weather debuff if necessary, i.e., if #weather is true	195

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

/home/runner/work/Gwent/Gwent/app/src/main/AndroidManifest.xml	199
gwent/GwentApplication.java	217
gwent/data/Ability.java	199
gwent/data/AppDatabase.java	200
gwent/data/RowDao.java	200
gwent/data/RowEntity.java	200
gwent/data/RowType.java	201
gwent/data/UnitDao.java	201
gwent/data/UnitEntity.java	202
gwent/data/UnitRepository.java	204
gwent/domain/cases/BurnDialogUseCase.java	207
gwent/domain/cases/DamageCalculatorUseCase.java	208
gwent/domain/cases/RemoveUnitsUseCase.java	209
gwent/domain/cases/ResetAlertDialogBuilderAdapter.java	210
gwent/domain/cases/ResetDialogUseCase.java	210
gwent/domain/cases/ResetRepositoryUseCase.java	212
gwent/domain/cases/RevengeAlertDialogBuilderAdapter.java	213
gwent/domain/damage/BondDamageCalculatorDecorator.java	214
gwent/domain/damage/DamageCalculator.java	214
gwent/domain/damage/DamageCalculatorBuildDirector.java	214
gwent/domain/damage/DamageCalculatorBuilder.java	215
gwent/domain/damage/DamageCalculatorDecorator.java	216
gwent/domain/damage/HornDamageCalculatorDecorator.java	216
gwent/domain/damage/MoralDamageCalculatorDecorator.java	216
gwent/domain/damage/WeatherDamageCalculator.java	217
gwent/ui/dialogs/ChangeFactionDialog.java	231
gwent/ui/dialogs/CoinFlipDialog.java	232
gwent/ui/dialogs/OverlayDialog.java	232
gwent/ui/dialogs/addcard/AddCardDialog.java	218
gwent/ui/dialogs/addcard/CardNumberPickerAdapter.java	219
gwent/ui/dialogs/addcard/DamageValuePicker.java	220
gwent/ui/dialogs/addcard/SquadManager.java	221
gwent/ui/dialogs/addcard/SquadState.java	222
gwent/ui/dialogs/addcard/StringValuePicker.java	223
gwent/ui/dialogs/addcard/ValuePicker.java	224
gwent/ui/dialogs/cards/CardListAdapter.java	225
gwent/ui/dialogs/cards/CardUiState.java	227
gwent/ui/dialogs/cards/CardUiStateFactory.java	228
gwent/ui/dialogs/cards/ShowUnitsDialog.java	229
gwent/ui/introduction/IndicatorManager.java	233

gwent/ui/introduction/IntroductionActivity.java	233
gwent/ui/introduction/IntroductionFragment.java	234
gwent/ui/introduction/SectionsPagerAdapter.java	235
gwent/ui/main/FactionSwitchListener.java	235
gwent/ui/main/GameBoardViewModel.java	237
gwent/ui/main/ImageViewSwitchAnimator.java	239
gwent/ui/main/MainActivity.java	240
gwent/ui/main/MenuUiState.java	243
gwent/ui/main/MenuUiStateObserver.java	243
gwent/ui/main/RowUiState.java	244
gwent/ui/main/RowUiStateObserver.java	245
gwent/ui/settings/RuleActivity.java	246
gwent/ui/settings/RuleSection.java	247
gwent/ui/settings/SettingsActivity.java	247
gwent/ui/settings/SettingsHeaderFragment.java	248
gwent/ui/settings/SettingsRuleFragment.java	248
gwent/ui/settings/SettingsSoundFragment.java	249
gwent/ui/sounds/Sound.java	250
gwent/ui/sounds/SoundManager.java	250
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/background_drawable.xml	252
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_copy.xml	253
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_delete.xml	253
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_design.xml	253
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_explore.xml	254
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_factionreset.xml	254
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_next.xml	254
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_rules.xml	254
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_sounds.xml	254
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_warnings.xml	255
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/indicator_selected.xml	255
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/indicator_unselected.xml	255
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/rectangle.xml	255
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/rectangle_black_transparent.xml	255
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/rectangle_cancel_popup.xml	255
/home/runner/work/Gwent/Gwent/app/src/main/res/layout-land-v26/fragment_introduction.xml	256
/home/runner/work/Gwent/Gwent/app/src/main/res/layout-land/fragment_introduction.xml	256
/home/runner/work/Gwent/Gwent/app/src/main/res/layout-v26/activity_rules.xml	258
/home/runner/work/Gwent/Gwent/app/src/main/res/layout-v26/fragment_introduction.xml	257
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/activity_introduction.xml	259
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/activity_main.xml	260
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/activity_rules.xml	258
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/activity_settings.xml	262
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/alertdialog_checkbox.xml	263
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/card.xml	263
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/fragment_introduction.xml	257
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/popup_add_card.xml	264
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/popup_cards.xml	266
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/popup_coin_normal.xml	267
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/popup_coin_stewie.xml	267
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/popup_coin_terry.xml	269
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/popup_coin_vin.xml	270
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/popup_faction.xml	270
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/row.xml	272
/home/runner/work/Gwent/Gwent/app/src/main/res/mipmap-anydpi-v26/icon_launcher.xml	273
/home/runner/work/Gwent/Gwent/app/src/main/res/mipmap-anydpi-v26/icon_launcher_round.xml	273
/home/runner/work/Gwent/Gwent/app/src/main/res/values-de-rDE/strings.xml	275
/home/runner/work/Gwent/Gwent/app/src/main/res/values/arrays.xml	273
/home/runner/work/Gwent/Gwent/app/src/main/res/values/attr.xml	274

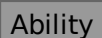
/home/runner/work/Gwent/Gwent/app/src/main/res/values/colors.xml	274
/home/runner/work/Gwent/Gwent/app/src/main/res/values/dimens.xml	274
/home/runner/work/Gwent/Gwent/app/src/main/res/values/icon_launcher_background.xml	275
/home/runner/work/Gwent/Gwent/app/src/main/res/values/preferences.xml	275
/home/runner/work/Gwent/Gwent/app/src/main/res/values/strings.xml	279
/home/runner/work/Gwent/Gwent/app/src/main/res/values/styles.xml	283
/home/runner/work/Gwent/Gwent/app/src/main/res/xml/header_preferences.xml	284
/home/runner/work/Gwent/Gwent/app/src/main/res/xml/rule_preferences.xml	285
/home/runner/work/Gwent/Gwent/app/src/main/res/xml/sound_preferences.xml	285

Chapter 5

Class Documentation

5.1 Ability Enum Reference

An enum listing the different card abilities possible.
Collaboration diagram for Ability:



Public Attributes

- [NONE](#)
Represents a card that has no ability.
- [HORN](#)
Represents a card that has the Commanders Horn ability, e.g.
- [REVENGE](#)
Represents a card that has the Summon Avenger ability.
- [BINDING](#)
Represents a card that has the Tight Bond ability.
- [MORAL_BOOST](#)
Represents a card that has the Morale Boost ability.

5.1.1 Detailed Description

An enum listing the different card abilities possible.
Definition at line 6 of file [Ability.java](#).

5.1.2 Member Data Documentation

5.1.2.1 BINDING

`BINDING`

Represents a card that has the Tight Bond ability.

Definition at line 25 of file [Ability.java](#).

Referenced by [UnitEntity.UnitEntity\(\)](#), [UnitRepository.insertUnit\(\)](#), and [UnitEntity.setSquad\(\)](#).

5.1.2.2 HORN

HORN

Represents a card that has the Commanders Horn ability, e.g. Dandelion.

Definition at line 15 of file [Ability.java](#).

5.1.2.3 MORAL_BOOST

MORAL_BOOST

Represents a card that has the Morale Boost ability.

Definition at line 31 of file [Ability.java](#).

5.1.2.4 NONE

NONE

Represents a card that has no ability.

Definition at line 10 of file [Ability.java](#).

5.1.2.5 REVENGE

REVENGE

Represents a card that has the Summon Avenger ability.

Definition at line 20 of file [Ability.java](#).

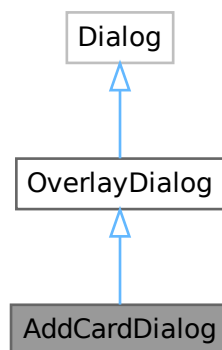
The documentation for this enum was generated from the following file:

- gwent/data/Ability.java

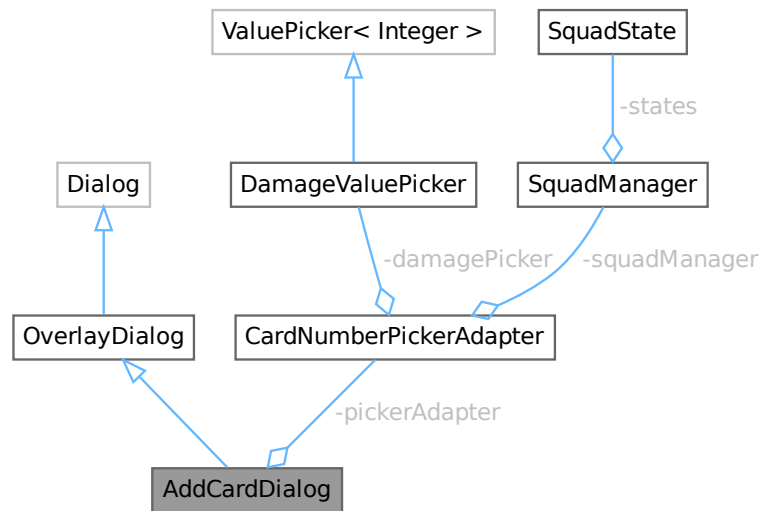
5.2 AddCardDialog Class Reference

An OverlayDialog used to add new UnitEntity objects to a certain #row.

Inheritance diagram for AddCardDialog:



Collaboration diagram for AddCardDialog:



Public Member Functions

- [AddCardDialog](#) (@NonNull ShowUnitsDialog caller, @NonNull SoundManager [soundManager](#))
Constructor of an AddCardDialog called by the given ShowUnitsDialog using the given SoundManager.
- [AddCardDialog](#) (@NonNull Context context, @NonNull RowType [row](#), @NonNull SoundManager [soundManager](#))
Constructor of an AddCardDialog in the given Context and for the given row.

Protected Member Functions

- void [onCreate](#) (Bundle savedInstanceState)
Initializes the #pickerAdapter if not yet done so and sets View.OnClickListener for each button.

Protected Member Functions inherited from [OverlayDialog](#)

- [OverlayDialog](#) (@NonNull Context context, @LayoutRes int [layout](#), @IdRes int [dismissViewId](#))
Constructor of an OverlayDialog in the given Context, with the given layout and dismissViewId.
- [OverlayDialog](#) (@NonNull Context context, @LayoutRes int [layout](#))
Constructor of an OverlayDialog in the given Context and with the given layout.

Private Attributes

- final RowType [row](#)
RowType defining the row the new UnitEntity objects are added to.
- final SoundManager [soundManager](#)
SoundManager used to play a Sound whenever a card is added.
- final CompositeDisposable [disposables](#) = new CompositeDisposable()
CompositeDisposable keeping track of all subscriptions to observables made by this class.
- [CardNumberPickerAdapter](#) [pickerAdapter](#) = null
CardNumberPickerAdapter managing the connection to the NumberPicker views of this Dialog.

Additional Inherited Members

Static Package Attributes inherited from [OverlayDialog](#)

- static final int [NO_DISMISS_VIEW](#) = -1
Integer used as #dismissViewId, when the Dialog shouldn't be cancelable by clicking on a View.

5.2.1 Detailed Description

An OverlayDialog used to add new UnitEntity objects to a certain #row.
Definition at line 22 of file [AddCardDialog.java](#).

5.2.2 Constructor & Destructor Documentation

5.2.2.1 AddCardDialog() [1/2]

```
AddCardDialog (
    @NonNull ShowUnitsDialog caller,
    @NonNull SoundManager soundManager )
```

Constructor of an AddCardDialog called by the given ShowUnitsDialog using the given SoundManager.
Sets a Dialog.OnDismissListener disposing #disposables and showing the caller again. Wrapper of #AddCardDialog(Context, RowType, SoundManager).

See also

[#AddCardDialog\(Context, RowType, SoundManager\)](#)

Parameters

<i>caller</i>	ShowUnitsDialog that called this Dialog.
<i>soundManager</i>	SoundManager used to play a Sound when cards are added.

Definition at line 67 of file [AddCardDialog.java](#).
References [AddCardDialog.disposables](#), and [AddCardDialog.soundManager](#).

5.2.2.2 AddCardDialog() [2/2]

```
AddCardDialog (
    @NonNull Context context,
    @NonNull RowType row,
    @NonNull SoundManager soundManager )
```

Constructor of an AddCardDialog in the given Context and for the given row.
Sets a Dialog.OnDismissListener disposing #disposables and showing the caller again.

Parameters

<i>context</i>	Context this Dialog is shown in.
<i>row</i>	RowType defining which row the new UnitEntity objects are added to.
<i>soundManager</i>	SoundManager used to play a Sound when cards are added.

Definition at line 84 of file [AddCardDialog.java](#).
References [AddCardDialog.disposables](#), [AddCardDialog.row](#), and [AddCardDialog.soundManager](#).

5.2.3 Member Function Documentation

5.2.3.1 onCreate()

```
void onCreate (
```

```
Bundle savedInstanceState ) [protected]
```

Initializes the #pickerAdapter if not yet done so and sets View.OnClickListener for each button.

Parameters

<i>savedInstanceState</i>	If this dialog is being reinitialized after the hosting activity was previously shut down, holds the result from the most recent call to <code>onSaveInstanceState</code> , or null if this is the first time.
---------------------------	--

Reimplemented from [OverlayDialog](#).

Definition at line 102 of file [AddCardDialog.java](#).

References [CardNumberPickerAdapter.addSelectedUnits\(\)](#), [AddCardDialog.disposables](#), [GwentApplication.getRepository\(\)](#), [AddCardDialog.pickerAdapter](#), and [AddCardDialog.row](#).

5.2.4 Member Data Documentation

5.2.4.1 disposables

```
final CompositeDisposable disposables = new CompositeDisposable() [private]
```

CompositeDisposable keeping track of all subscriptions to observables made by this class.

Is being disposed in an `android.content.DialogInterface.OnDismissListener` that is set in `#AddCardDialog(ShowUnitsDialog)` and `#AddCardDialog(Context, RowType)`.

See also

`android.content.DialogInterface.OnDismissListener`
`#AddCardDialog(ShowUnitsDialog, SoundManager)`
`#AddCardDialog(Context, RowType, SoundManager)`

Definition at line 47 of file [AddCardDialog.java](#).

Referenced by [AddCardDialog.AddCardDialog\(\)](#), [AddCardDialog.AddCardDialog\(\)](#), and [AddCardDialog.onCreate\(\)](#).

5.2.4.2 pickerAdapter

```
CardNumberPickerAdapter pickerAdapter = null [private]
```

CardNumberPickerAdapter managing the connection to the NumberPicker views of this Dialog.

Is lazily initialized in `#onCreate()`.

See also

`#onCreate(Bundle)`
`R.id::card_layout`

Definition at line 57 of file [AddCardDialog.java](#).

Referenced by [AddCardDialog.onCreate\(\)](#).

5.2.4.3 row

```
final RowType row [private]
```

RowType defining the row the new UnitEntity objects are added to.

Definition at line 27 of file [AddCardDialog.java](#).

Referenced by [AddCardDialog.AddCardDialog\(\)](#), and [AddCardDialog.onCreate\(\)](#).

5.2.4.4 soundManager

```
final SoundManager soundManager [private]
```

SoundManager used to play a Sound whenever a card is added.

See also

SoundManager::playCardAddSound(RowType, boolean)

Definition at line 35 of file [AddCardDialog.java](#).

Referenced by [AddCardDialog.AddCardDialog\(\)](#), and [AddCardDialog.AddCardDialog\(\)](#).

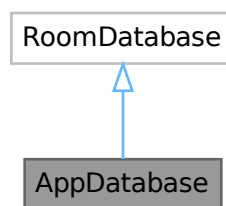
The documentation for this class was generated from the following file:

- gwent/ui/dialogs/addcard/AddCardDialog.java

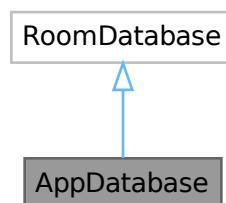
5.3 AppDatabase Class Reference

An abstract class extending RoomDatabase and representing the SQLite database of the application.

Inheritance diagram for AppDatabase:



Collaboration diagram for AppDatabase:



Package Functions

- abstract [UnitDao units](#) ()
Returns an object implementing the operations defined in UnitDao.
- abstract [RowDao rows](#) ()
Returns an object implementing the operations defined in RowDao.

5.3.1 Detailed Description

An abstract class extending RoomDatabase and representing the SQLite database of the application.

The implementation is generated by the Room framework.

Definition at line 11 of file [AppDatabase.java](#).

5.3.2 Member Function Documentation

5.3.2.1 rows()

abstract `RowDao` `rows ()` [abstract], [package]

Returns an object implementing the operations defined in `RowDao`.

Returns

`RowDao` implementing DAO for `rows` table.

Referenced by `UnitRepository.clearWeather()`, `UnitRepository.initializeRows()`, `UnitRepository.isHorn()`, `UnitRepository.isHornFlowable()`, `UnitRepository.isWeather()`, `UnitRepository.isWeatherFlowable()`, `UnitRepository.reset()`, `UnitRepository.switchHorn()`, and `UnitRepository.switchWeather()`.

5.3.2.2 units()

abstract `UnitDao` `units ()` [abstract], [package]

Returns an object implementing the operations defined in `UnitDao`.

Returns

`UnitDao` implementing DAO for `units` table.

Referenced by `UnitRepository.countUnits()`, `UnitRepository.countUnits()`, `UnitRepository.delete()`, `UnitRepository.delete()`, `UnitRepository.getUnit()`, `UnitRepository.getUnits()`, `UnitRepository.getUnits()`, `UnitRepository.getUnitsFlowable()`, `UnitRepository.hasNonEpicUnitsFlowable()`, `UnitRepository.insertUnit()`, and `UnitRepository.reset()`.

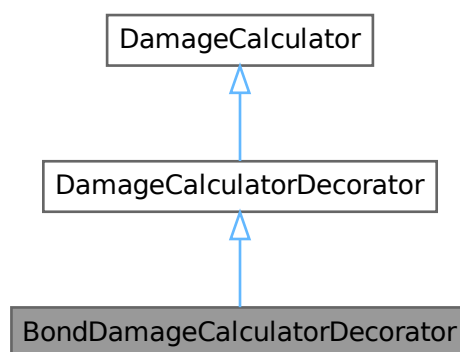
The documentation for this class was generated from the following file:

- `gwent/data/AppDatabase.java`

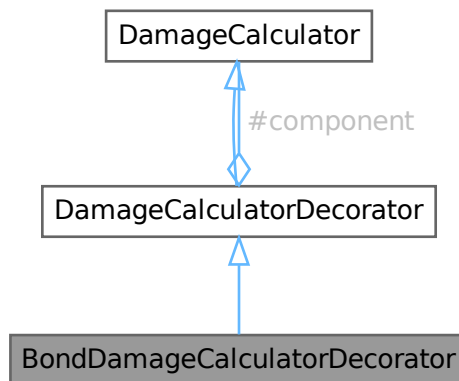
5.4 BondDamageCalculatorDecorator Class Reference

A `DamageCalculator` class responsible for calculating the tight bond buff if necessary, i.e., if there are units with the `com.peternaggschga.gwent.data.Ability::BINDING` ability.

Inheritance diagram for `BondDamageCalculatorDecorator`:



Collaboration diagram for BondDamageCalculatorDecorator:



Public Member Functions

- `int calculateDamage (int id, @IntRange(from=0) int damage)`
Calculates the (de-)buffed damage of unit with the given id and the given base-damage.
- `Color isBuffed (int id)`
Calculates whether the unit with the given id is shown as `Color::BUFFED`, `Color::DEBUFFED`, or `Color::DEFAULT`.

Package Functions

- `BondDamageCalculatorDecorator (@NonNull DamageCalculator component, @NonNull Map< Integer, Integer > idToSquadSize)`
Constructor of a `BondDamageCalculatorDecorator`.

Package Functions inherited from `DamageCalculatorDecorator`

- `DamageCalculatorDecorator (@NonNull DamageCalculator component)`
Constructor of a `DamageCalculatorDecorator`.

Private Attributes

- `final Map< Integer, Integer > idToSquadSize`
A Map mapping the ids of all units with the `com.peternaggschga.gwent.data.Ability::BINDING` ability to the respective squad size.

Additional Inherited Members

Protected Attributes inherited from `DamageCalculatorDecorator`

- `final DamageCalculator component`
A `DamageCalculator` which is decorated by this decorator.

5.4.1 Detailed Description

A `DamageCalculator` class responsible for calculating the tight bond buff if necessary, i.e., if there are units with the `com.peternaggschga.gwent.data.Ability::BINDING` ability.

Acts as a ConcreteDecorator in the implemented decorator pattern and should decorate a WeatherDamageCalculator for correct calculation.

Definition at line 17 of file [BondDamageCalculatorDecorator.java](#).

5.4.2 Constructor & Destructor Documentation

5.4.2.1 BondDamageCalculatorDecorator()

```
BondDamageCalculatorDecorator (
    @NonNull DamageCalculator component,
    @NonNull Map< Integer, Integer > idToSquadSize ) [package]
```

Constructor of a BondDamageCalculatorDecorator.

Should only be called by DamageCalculatorBuilder. The given damage calculator should be of type WeatherDamageCalculator for correct damage calculation. The values of the given Map must contain non-null or positive Integers.

Parameters

<i>component</i>	DamageCalculator that is being decorated by this decorator.
<i>idToSquadSize</i>	Map mapping the ids of all units with the com.paternaggschga.gwent.data.Ability::BINDING ability to the respective squad size.

Exceptions

<i>IllegalArgumentException</i>	When idToSquadSize contains non-positive or null values.
---------------------------------	--

See also

[DamageCalculatorBuilder](#)

Definition at line 35 of file [BondDamageCalculatorDecorator.java](#).

References [DamageCalculatorDecorator.component](#), and [BondDamageCalculatorDecorator.idToSquadSize](#).

5.4.3 Member Function Documentation

5.4.3.1 calculateDamage()

```
int calculateDamage (
    int id,
    @IntRange(from=0) int damage )
```

Calculates the (de-)buffed damage of unit with the given id and the given base-damage.

Returns the given damage times the number of units with the same squad if the unit has the [com.paternaggschga.gwent.data.Ability::BINDING](#) ability.

Parameters

<i>id</i>	Integer representing the UnitEntity::id of the unit whose (de-)buff damage is calculated.
<i>damage</i>	Integer representing the base-damage of the unit whose (de-)buff damage is calculated.

Returns

Integer representing the (de-)buffed damage of the unit.

Exceptions

<i>IllegalArgumentException</i>	When damage is negative.
---------------------------------	--------------------------

Implements [DamageCalculator](#).

Definition at line 54 of file [BondDamageCalculatorDecorator.java](#).

References [DamageCalculator.calculateDamage\(\)](#), [DamageCalculatorDecorator.component](#), and [BondDamageCalculatorDecorator.i](#).

5.4.3.2 isBuffed()

```
Color isBuffed (
    int id )
```

Calculates whether the unit with the given id is shown as Color::BUFFED, Color::DEBUFFED, or Color::DEFAULT. Units are shown as Color::BUFFED when they are in a squad of two or more units, otherwise their Color is defined by #component.

Parameters

<i>id</i>	Integer representing the UnitEntity::id of the unit buff status is calculated.
-----------	--

Returns

Color representing whether the unit is buffed, de-buffed or not affected.

See also

Color

Implements [DamageCalculator](#).

Definition at line 72 of file [BondDamageCalculatorDecorator.java](#).

References [DamageCalculatorDecorator.component](#), [BondDamageCalculatorDecorator.idToSquadSize](#), and [DamageCalculator.isBuffed\(\)](#).

5.4.4 Member Data Documentation

5.4.4.1 idToSquadSize

```
final Map<Integer, Integer> idToSquadSize [private]
```

A Map mapping the ids of all units with the [com.peternaggschga.gwent.data.Ability::BINDING](#) ability to the respective squad size.

Definition at line 22 of file [BondDamageCalculatorDecorator.java](#).

Referenced by [BondDamageCalculatorDecorator.BondDamageCalculatorDecorator\(\)](#), [BondDamageCalculatorDecorator.calculateDa](#) and [BondDamageCalculatorDecorator.isBuffed\(\)](#).

The documentation for this class was generated from the following file:

- gwent/domain/damage/BondDamageCalculatorDecorator.java

5.5 BurnDialogUseCase Class Reference

A use case class responsible for dispatching a remove call to RemoveUnitsUseCase.

Collaboration diagram for BurnDialogUseCase:

BurnDialogUseCase

Static Public Member Functions

- static Single< Boolean > [burn](#) (@NonNull Context context, @NonNull SoundManager soundManager)
Burns the strongest UnitEntity objects in UnitRepository.

Static Protected Member Functions

- static Single< Boolean > [burn](#) (@NonNull Context context, @NonNull UnitRepository repository, @NonNull SoundManager soundManager)
Burns the strongest UnitEntity objects in UnitRepository.

Static Private Member Functions

- static Single< List< UnitEntity > > [getBurnUnits](#) (@NonNull UnitRepository repository)
Returns the list of units that would be affected by a burn operation.

5.5.1 Detailed Description

A use case class responsible for dispatching a remove call to RemoveUnitsUseCase.
Definition at line 28 of file [BurnDialogUseCase.java](#).

5.5.2 Member Function Documentation

5.5.2.1 burn() [1/2]

```
static Single< Boolean > burn (
    @NonNull Context context,
    @NonNull SoundManager soundManager ) [static]
```

Burns the strongest UnitEntity objects in UnitRepository.

Invokes a Dialog asking whether the user really wants to remove those units. ResetRepositoryUseCase is used for resetting. Wrapper for #burn(Context, UnitRepository).

Parameters

<i>context</i>	Context where a Dialog can be inflated.
<i>soundManager</i>	SoundManager used, if an UnitEntity has the Ability::REVENGE ability.

Returns

A Single emitting a Boolean defining whether the units really were burned.

See also

#burn(Context, UnitRepository, SoundManager)

RemoveUnitsUseCase::remove(Context, UnitRepository, Collection, SoundManager)

Definition at line 85 of file [BurnDialogUseCase.java](#).

References [BurnDialogUseCase.burn\(\)](#), and [GwentApplication.getRepository\(\)](#).

Referenced by [BurnDialogUseCase.burn\(\)](#).

5.5.2.2 burn() [2/2]

```
static Single< Boolean > burn (
    @NonNull Context context,
    @NonNull UnitRepository repository,
    @NonNull SoundManager soundManager ) [static], [protected]
```

Burns the strongest UnitEntity objects in UnitRepository.

Invokes a Dialog asking whether the user really wants to remove those units. ResetRepositoryUseCase is used for resetting.

Parameters

<i>context</i>	Context where a Dialog can be inflated.
<i>repository</i>	UnitRepository where units are burned.
<i>soundManager</i>	SoundManager used, if an UnitEntity has the Ability::REVENGE ability.

Returns

A Single emitting a Boolean defining whether the units really were burned.

See also

`RemoveUnitsUseCase::remove(Context, UnitRepository, Collection, SoundManager)`

Definition at line 101 of file [BurnDialogUseCase.java](#).

References [BurnDialogUseCase.getBurnUnits\(\)](#).

5.5.2.3 getBurnUnits()

```
static Single< List< UnitEntity > > getBurnUnits (
    @NonNull UnitRepository repository ) [static], [private]
```

Returns the list of units that would be affected by a burn operation.

The returned list may be empty.

Parameters

<i>repository</i>	UnitRepository where units are fetched.
-------------------	---

Returns

A Single emitting the List of UnitEntity objects that would be affected by the operation.

Definition at line 37 of file [BurnDialogUseCase.java](#).

References [DamageCalculatorUseCase.getDamageCalculator\(\)](#).

Referenced by [BurnDialogUseCase.burn\(\)](#).

The documentation for this class was generated from the following file:

- gwent/domain/cases/BurnDialogUseCase.java

5.6 ResetAlertDialogBuilderAdapter.Callback Interface Reference

An interface defining functions to propagate the user's decision back to the creator.

Collaboration diagram for ResetAlertDialogBuilderAdapter.Callback:

```
ResetAlertDialogBuilderAdapter.
    Callback
```

Public Member Functions

- void [reset](#) (boolean resetDecision, boolean keepUnit)

Called when the user makes a decision.

- default void [reset](#) (boolean resetDecision)

Wrapper for #reset(boolean, boolean).

5.6.1 Detailed Description

An interface defining functions to propagate the user's decision back to the creator.

Definition at line 98 of file [ResetAlertDialogBuilderAdapter.java](#).

5.6.2 Member Function Documentation

5.6.2.1 reset() [1/2]

```
default void reset (
    boolean resetDecision )
```

Wrapper for #reset(boolean, boolean).

Can be used when the AlertDialog is not a monster dialog since it calls #reset(boolean, boolean) without keeping a unit.

Parameters

<i>resetDecision</i>	Boolean defining whether the user has confirmed the reset.
----------------------	--

See also

[#reset\(boolean, boolean\)](#)

Definition at line 117 of file [ResetAlertDialogBuilderAdapter.java](#).

References [ResetAlertDialogBuilderAdapter.Callback.reset\(\)](#).

5.6.2.2 reset() [2/2]

```
void reset (
    boolean resetDecision,
    boolean keepUnit )
```

Called when the user makes a decision.

When the AlertDialog is a monster dialog, #reset(boolean) can be used.

Parameters

<i>resetDecision</i>	Boolean defining whether the user has confirmed the reset.
<i>keepUnit</i>	Boolean defining whether a random unit should be kept when resetting.

See also

[#reset\(boolean\)](#)

Referenced by [ResetAlertDialogBuilderAdapter.ResetAlertDialogBuilderAdapter\(\)](#), [ResetAlertDialogBuilderAdapter.Callback.reset\(\)](#), and [ResetAlertDialogBuilderAdapter.setMonsterDialog\(\)](#).

The documentation for this interface was generated from the following file:

- gwent/domain/cases/ResetAlertDialogBuilderAdapter.java

5.7 ChangeFactionDialog.Callback Interface Reference

An interface used for propagating a selected theme back to the creator of a ChangeFactionDialog.

Collaboration diagram for `ChangeFactionDialog.Callback`:

`ChangeFactionDialog.Callback`

Public Member Functions

- void [onThemeSelect](#) (@IntRange(from=THEME_MONSTER, to=THEME_SCOIATAEL) int theme)
Callback being called when a theme is selected in the respective `ChangeFactionDialog`.

5.7.1 Detailed Description

An interface used for propagating a selected theme back to the creator of a `ChangeFactionDialog`.
 Definition at line 84 of file [ChangeFactionDialog.java](#).

5.7.2 Member Function Documentation

5.7.2.1 onThemeSelect()

```
void onThemeSelect (
    @IntRange(from=THEME_MONSTER, to=THEME_SCOIATAEL) int theme )
```

Callback being called when a theme is selected in the respective `ChangeFactionDialog`.

Parameters

<i>theme</i>	Integer representing the selected theme.
--------------	--

Referenced by [ChangeFactionDialog.getOnThemeClickListener\(\)](#).

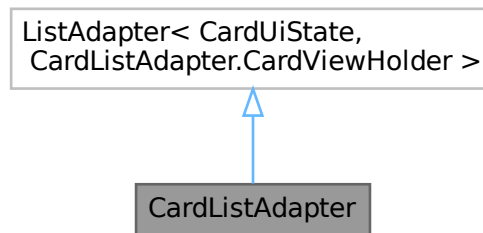
The documentation for this interface was generated from the following file:

- gwent/ui/dialogs/ChangeFactionDialog.java

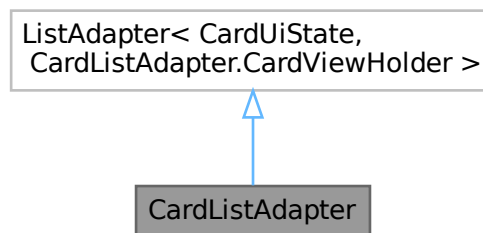
5.8 CardListAdapter Class Reference

A `ListAdapter` used to provide `CardListAdapter.CardViewHolder` objects created from `CardUiState` objects to a `RecyclerView`.

Inheritance diagram for CardListAdapter:



Collaboration diagram for CardListAdapter:



Classes

- class [CardViewHolder](#)

A RecyclerView.ViewHolder class managing a card view that shows a representation of the UnitEntity referenced by #itemId.

Public Member Functions

- [CardViewHolder onCreateViewHolder](#) (@NonNull ViewGroup parent, int viewType)
Creates a new CardViewHolder managing a new card view that is inflated from the given parent.
- void [onBindViewHolder](#) (@NonNull [CardViewHolder](#) holder, @IntRange(from=0) int position)
Changes the views of CardViewHolder to represent the item at the given position.
- long [getItemId](#) (@IntRange(from=0) int position)
Returns the UnitEntity::id of the UnitEntity represented at the given position.

Package Functions

- [CardListAdapter](#) (@NonNull Consumer< Integer > [onCopy](#), @NonNull Consumer< Integer > [onRemove](#))
Constructor of a CardListAdapter with the given #onCopy and #onRemove callbacks.

Private Attributes

- final Consumer< Integer > [onCopy](#)
Consumer that is used in `#onCreateViewHolder(ViewGroup, int)` as `CardListAdapter.CardViewHolder::onCopy`.
- final Consumer< Integer > [onRemove](#)
Consumer that is used in `#onCreateViewHolder(ViewGroup, int)` as `CardListAdapter.CardViewHolder::onRemove`.

5.8.1 Detailed Description

A ListAdapter used to provide CardListAdapter.CardViewHolder objects created from CardUiState objects to a RecyclerView.

See also

CardUiState
CardListAdapter.CardViewHolder
ListAdapter
RecyclerView

Definition at line 28 of file [CardListAdapter.java](#).

5.8.2 Constructor & Destructor Documentation

5.8.2.1 CardListAdapter()

```
CardListAdapter (
    @NonNull Consumer< Integer > onCopy,
    @NonNull Consumer< Integer > onRemove ) [package]
```

Constructor of a CardListAdapter with the given `#onCopy` and `#onRemove` callbacks.

Calls super-constructor `ListAdapter::ListAdapter(DiffUtil.ItemCallback)` with `CardUiState::DIFF_CALLBACK`. Also calls `#setHasStableIds()` since `#getItemId()` returns the stable `UnitEntity::id`.

See also

`ListAdapter::ListAdapter(DiffUtil.ItemCallback)`
`CardUiState::DIFF_CALLBACK`
`#setHasStableIds(boolean)`

Parameters

<i>onCopy</i>	Consumer that is called with the <code>UnitEntity::id</code> , when the copy-button of the representing card is clicked.
<i>onRemove</i>	Consumer that is called with the <code>UnitEntity::id</code> , when the delete-button of the representing card is clicked.

Definition at line 65 of file [CardListAdapter.java](#).

References [CardUiState.DIFF_CALLBACK](#), [CardListAdapter.onCopy](#), and [CardListAdapter.onRemove](#).

5.8.3 Member Function Documentation

5.8.3.1 getItemId()

```
long getItemId (
    @IntRange(from=0) int position )
```

Returns the `UnitEntity::id` of the `UnitEntity` represented at the given position.

Parameters

<i>position</i>	Adapter position to query.
-----------------	----------------------------

Returns

A Long referencing the `UnitEntity::id` of the `UnitEntity` represented at the given position.

Definition at line 122 of file [CardListAdapter.java](#).

5.8.3.2 onBindViewHolder()

```
void onBindViewHolder (
    @NonNull CardViewHolder holder,
    @IntRange(from=0) int position )
```

Changes the views of `CardViewHolder` to represent the item at the given position.

Parameters

<i>holder</i>	The <code>CardViewHolder</code> which should be updated to represent the contents of the item at the given position in the data set.
<i>position</i>	The position of the item within the adapter's data set.

Definition at line 93 of file [CardListAdapter.java](#).

References [CardUiState.getAbilityImageId\(\)](#), [CardUiState.getDamageBackgroundImageId\(\)](#), [CardUiState.getDamageString\(\)](#), [CardUiState.getDamageTextColor\(\)](#), [CardUiState.getSquadString\(\)](#), [CardUiState.getUnitId\(\)](#), [CardUiState.showAbility\(\)](#), and [CardUiState.showSquad\(\)](#).

5.8.3.3 onCreateViewHolder()

```
CardViewHolder onCreateViewHolder (
    @NonNull ViewGroup parent,
    int viewType )
```

Creates a new `CardViewHolder` managing a new card view that is inflated from the given parent.

Parameters

<i>parent</i>	The <code>ViewGroup</code> into which the new View will be added after it is bound to an adapter position.
<i>viewType</i>	The view type of the new View.

Returns

A `CardViewHolder` object for a new card layout and with `#onCopy` and `#onRemove` callbacks.

Definition at line 81 of file [CardListAdapter.java](#).

References [CardListAdapter.onCopy](#), and [CardListAdapter.onRemove](#).

5.8.4 Member Data Documentation

5.8.4.1 onCopy

```
final Consumer<Integer> onCopy [private]
```

Consumer that is used in `#onCreateViewHolder(ViewGroup, int)` as `CardListAdapter.CardViewHolder::onCopy`. Is called when the user clicks on the copy-button of an element with the `UnitEntity::id` of the represented `UnitEntity`.

See also

`#onCreateViewHolder(ViewGroup, int)`

`CardListAdapter.CardViewHolder::onCopy`

Definition at line 39 of file [CardListAdapter.java](#).

Referenced by [CardListAdapter.CardListAdapter\(\)](#), [CardListAdapter.CardViewHolder.CardViewHolder\(\)](#), and [CardListAdapter.onCreateViewHolder\(\)](#).

5.8.4.2 onRemove

```
final Consumer<Integer> onRemove [private]
```

Consumer that is used in `#onCreateViewHolder(ViewGroup, int)` as `CardListAdapter.CardViewHolder::onRemove`. Is called when the user clicks on the delete-button of an element with the `UnitEntity::id` of the represented `UnitEntity`.

See also

`#onCreateViewHolder(ViewGroup, int)`
`CardListAdapter.CardViewHolder::onRemove`

Definition at line 50 of file [CardListAdapter.java](#).

Referenced by [CardListAdapter.CardListAdapter\(\)](#), [CardListAdapter.CardViewHolder.CardViewHolder\(\)](#), and [CardListAdapter.onCreateViewHolder\(\)](#).

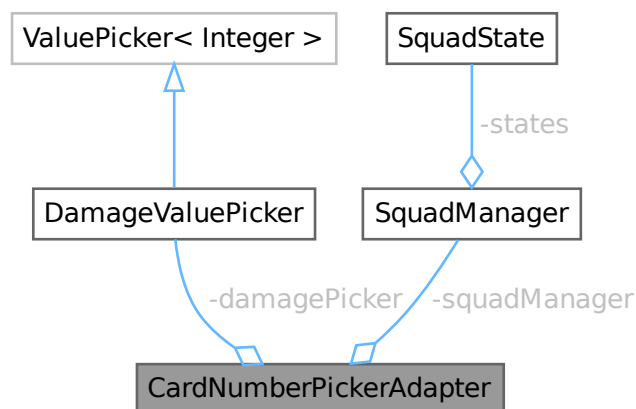
The documentation for this class was generated from the following file:

- `gwent/ui/dialogs/cards/CardListAdapter.java`

5.9 CardNumberPickerAdapter Class Reference

A helper class responsible for initializing the `NumberPicker` views of an `AddCardDialog` (in `#CardNumberPickerAdapter()`) and adding `UnitEntity` objects with the selected attributes when `#addSelectedUnits()` is called.

Collaboration diagram for `CardNumberPickerAdapter`:



Package Functions

- [CardNumberPickerAdapter](#) (@NonNull ViewGroup pickerGroup, @NonNull [SquadManager](#) squadManager)
Constructor of a `CardNumberPickerAdapter` managing the `NumberPicker` views in the given `ViewGroup` using the given `SquadManager`.
- `Single< Boolean > addSelectedUnits` (@NonNull RowType row)
Adds new `UnitEntity` objects the attributes selected by the managed pickers.

Static Package Functions

- static `NumberPicker.OnValueChangeListener` [getDelayedOnValueChangeListener](#) (@NonNull `NumberPicker.OnValueChangeListener` originalListener)
Creates a `NumberPicker.OnValueChangeListener` that only executes the given `NumberPicker.OnValueChangeListener` after 500 ms if the value has not changed.

Private Attributes

- final `ValuePicker< Boolean > epicPicker`
ValuePicker used to decide the value of `UnitEntity::epic`.
- final `DamageValuePicker damagePicker`
DamageValuePicker used to decide the value of `UnitEntity::damage`.
- final `ValuePicker< Ability > abilityPicker`
ValuePicker used to decide the value of `UnitEntity::ability`.
- final `NumberPicker squadPicker`
NumberPicker used to decide the value of `UnitEntity::squad`.
- final `SquadManager squadManager`
SquadManager containing current `SquadState` used to get squad information.
- final `NumberPicker numberPicker`
NumberPicker used to decide the number of `UnitEntity` objects that are inserted.

Static Private Attributes

- static final `List< Ability > EPIC_UNIT_ABILITIES = Arrays.asList(Ability.NONE, Ability.HORN, Ability.MORAL_BOOST)`
List of Ability values that epic units can have.

5.9.1 Detailed Description

A helper class responsible for initializing the `NumberPicker` views of an `AddCardDialog` (in `#CardNumberPickerAdapter()`) and adding `UnitEntity` objects with the selected attributes when `#addSelectedUnits()` is called.

See also

`AddCardDialog`

Definition at line 33 of file `CardNumberPickerAdapter.java`.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 CardNumberPickerAdapter()

```
CardNumberPickerAdapter (
    @NonNull ViewGroup pickerGroup,
    @NonNull SquadManager squadManager ) [package]
```

Constructor of a `CardNumberPickerAdapter` managing the `NumberPicker` views in the given `ViewGroup` using the given `SquadManager`.

Sets value bounds and `NumberPicker.OnValueChangeListener` for the pickers in the `ViewGroup`. The `ViewGroup` must be the `ConstraintLayout` with the id `R.id::card_layout` from `popup_add_card.xml`.

See also

`R.id::card_layout`

Parameters

<i>pickerGroup</i>	<code>ViewGroup</code> containing the managed <code>NumberPicker</code> views.
<i>squadManager</i>	<code>SquadManager</code> containing up-to-date <code>SquadState</code> .

Definition at line 89 of file `CardNumberPickerAdapter.java`.

References `CardNumberPickerAdapter.abilityPicker`, `CardNumberPickerAdapter.damagePicker`, `CardNumberPickerAdapter.EPIC_UNIT_ABILITIES`, `CardNumberPickerAdapter.epicPicker`, `CardNumberPickerAdapter.getDelayedOnValueChangeListener()`, `SquadManager.getFirstSquad()`, `SquadManager.MAX_NR_SQUADS`, `CardNumberPickerAdapter.numberPicker`, `SquadManager.onSquadChanged()`, `DamageValuePicker.setEpicValues()`, `CardNumberPickerAdapter.squadManager`, and `CardNumberPickerAdapter.squadPicker`.

5.9.3 Member Function Documentation

5.9.3.1 addSelectedUnits()

```
Single< Boolean > addSelectedUnits (
    @NonNull RowType row ) [package]
```

Adds new UnitEntity objects the attributes selected by the managed pickers.

Parameters

<i>row</i>	RowType defining to which row the units are added.
------------	--

Returns

A Single tracking operation status and containing whether the added unit is epic or not.

Definition at line 166 of file [CardNumberPickerAdapter.java](#).

References [CardNumberPickerAdapter.abilityPicker](#), [CardNumberPickerAdapter.damagePicker](#), [CardNumberPickerAdapter.epicPicker](#), [SquadManager.getFirstSquadWithMembers\(\)](#), [GwentApplication.getRepository\(\)](#), [DamageValuePicker.getValue\(\)](#), [CardNumberPickerAdapter.numberPicker](#), [CardNumberPickerAdapter.squadManager](#), and [CardNumberPickerAdapter.squadPicker](#).

Referenced by [AddCardDialog.onCreate\(\)](#).

5.9.3.2 getDelayedOnValueChangeListener()

```
static NumberPicker.OnValueChangeListener getDelayedOnValueChangeListener (
    @NonNull NumberPicker.OnValueChangeListener originalListener ) [static], [package]
```

Creates a NumberPicker.OnValueChangeListener that only executes the given NumberPicker.OnValueChangeListener after 500 ms if the value has not changed.

See also

[NumberPicker.OnValueChangeListener](#)

Parameters

<i>originalListener</i>	NumberPicker.OnValueChangeListener that is called when the value does not change.
-------------------------	---

Returns

A NumberPicker.OnValueChangeListener with delayed execution.

Definition at line 152 of file [CardNumberPickerAdapter.java](#).

Referenced by [CardNumberPickerAdapter.CardNumberPickerAdapter\(\)](#).

5.9.4 Member Data Documentation

5.9.4.1 abilityPicker

```
final ValuePicker<Ability> abilityPicker [private]
```

ValuePicker used to decide the value of UnitEntity::ability.

See also

[#squadPicker](#)

Definition at line 58 of file [CardNumberPickerAdapter.java](#).

Referenced by [CardNumberPickerAdapter.CardNumberPickerAdapter\(\)](#), and [CardNumberPickerAdapter.addSelectedUnits\(\)](#).

5.9.4.2 damagePicker

```
final DamageValuePicker damagePicker [private]
```

DamageValuePicker used to decide the value of `UnitEntity::damage`.

Definition at line 50 of file [CardNumberPickerAdapter.java](#).

Referenced by [CardNumberPickerAdapter.CardNumberPickerAdapter\(\)](#), and [CardNumberPickerAdapter.addSelectedUnits\(\)](#).

5.9.4.3 EPIC_UNIT_ABILITIES

```
final List<Ability> EPIC_UNIT_ABILITIES = Arrays.asList(Ability.NONE, Ability.HORN, Ability.MORAL_BOOST) [static], [private]
```

List of Ability values that epic units can have.

Definition at line 38 of file [CardNumberPickerAdapter.java](#).

Referenced by [CardNumberPickerAdapter.CardNumberPickerAdapter\(\)](#).

5.9.4.4 epicPicker

```
final ValuePicker<Boolean> epicPicker [private]
```

ValuePicker used to decide the value of `UnitEntity::epic`.

Definition at line 44 of file [CardNumberPickerAdapter.java](#).

Referenced by [CardNumberPickerAdapter.CardNumberPickerAdapter\(\)](#), and [CardNumberPickerAdapter.addSelectedUnits\(\)](#).

5.9.4.5 numberPicker

```
final NumberPicker numberPicker [private]
```

NumberPicker used to decide the number of `UnitEntity` objects that are inserted.

Definition at line 78 of file [CardNumberPickerAdapter.java](#).

Referenced by [CardNumberPickerAdapter.CardNumberPickerAdapter\(\)](#), and [CardNumberPickerAdapter.addSelectedUnits\(\)](#).

5.9.4.6 squadManager

```
final SquadManager squadManager [private]
```

SquadManager containing current `SquadState` used to get squad information.

Definition at line 72 of file [CardNumberPickerAdapter.java](#).

Referenced by [CardNumberPickerAdapter.CardNumberPickerAdapter\(\)](#), and [CardNumberPickerAdapter.addSelectedUnits\(\)](#).

5.9.4.7 squadPicker

```
final NumberPicker squadPicker [private]
```

NumberPicker used to decide the value of `UnitEntity::squad`.

Only visible if the value of `#abilityPicker` is set to `Ability::BINDING`.

See also

`#abilityPicker`

Definition at line 66 of file [CardNumberPickerAdapter.java](#).

Referenced by [CardNumberPickerAdapter.CardNumberPickerAdapter\(\)](#), and [CardNumberPickerAdapter.addSelectedUnits\(\)](#).

The documentation for this class was generated from the following file:

- [gwent/ui/dialogs/addcard/CardNumberPickerAdapter.java](#)

5.10 CardUiState Class Reference

A data class encapsulating the visible state of a card in the card list shown by the `ShowUnitsDialog`.

Collaboration diagram for CardUiState:

CardUiState

Public Member Functions

- **CardUiState** (int **unitId**, @DrawableRes int **damageBackgroundImageId**, @IntRange(from=UNUSED) int **damage**, @ColorInt int **damageTextColor**, @DrawableRes int **abilityImageId**, @Nullable @IntRange(from=1) Integer **squad**)
Constructor of a CardUiState encapsulating the given data.
- boolean **showAbility** ()
Returns whether the ability view is shown, i.e., if the unit has an ability other than Ability::NONE.
- boolean **showSquad** ()
Returns whether the squad view is shown, i.e., if the unit has the Ability::BINDING ability.
- int **getUnitId** ()
Returns the #unitId of the represented UnitEntity.
- int **getDamageBackgroundImageId** ()
Returns the drawable resource shown as the background of the damage view.
- String **getDamageString** ()
Returns the String shown in the damage view.
- int **getDamageTextColor** ()
Returns the text color of the damage view.
- int **getAbilityImageId** ()
Returns the drawable resource shown in the ability view.
- String **getSquadString** ()
Returns the String shown in the squad view.
- boolean **equals** (Object o)
Checks whether the given Object looks the same as this CardUiState.

Static Public Attributes

- static final DiffUtil.ItemCallback< **CardUiState** > **DIFF_CALLBACK**
DiffUtil.ItemCallback used to compare different CardUiState objects in a ListAdapter, e.g., CardListAdapter.
- static final int **UNUSED** = -1
Integer that is used instead of a drawable resource to indicate the absence of a meaningful #abilityImage.

Private Attributes

- final int **unitId**
Integer referencing the UnitEntity::id of the represented UnitEntity.
- final int **damageBackgroundImageId**
Integer referencing the drawable resource shown as the background of the damage view.
- final String **damageString**
String containing the number shown in the damage view or nothing if the represented UnitEntity is epic.
- final int **damageTextColor**
Integer representing the text color of the damage view.

- final int [abilityImageId](#)
Integer referencing the drawable resource shown in the ability view.
- final String [squadString](#)
String containing the number shown in the squad view.

5.10.1 Detailed Description

A data class encapsulating the visible state of a card in the card list shown by the ShowUnitsDialog. Can be created from the represented UnitEntity using CardUiStateFactory.

See also

CardUiStateFactory
ShowUnitsDialog

Definition at line 24 of file [CardUiState.java](#).

5.10.2 Constructor & Destructor Documentation

5.10.2.1 CardUiState()

```
CardUiState (
    int unitId,
    @DrawableRes int damageBackgroundImageId,
    @IntRange(from=UNUSED) int damage,
    @ColorInt int damageTextColor,
    @DrawableRes int abilityImageId,
    @Nullable @IntRange(from=1) Integer squad )
```

Constructor of a CardUiState encapsulating the given data.

Parameters

<i>unitId</i>	Integer representing the UnitEntity::id of the represented UnitEntity.
<i>damageBackgroundImageId</i>	Integer referencing the drawable resource shown by the damage view.
<i>damage</i>	Integer representing the damage of the represented UnitEntity.
<i>damageTextColor</i>	Integer representing the text color of the damage view.
<i>abilityImageId</i>	Integer referencing the drawable resource shown by the ability image view or #UNUSED.
<i>squad</i>	Integer representing the UnitEntity::squad of the represented UnitEntity.

Exceptions

<i>IllegalArgumentException</i>	When damage is less than #UNUSED or squad is neither null nor greater than zero.
---------------------------------	--

See also

CardUiStateFactory::createCardUiState(UnitEntity, DamageCalculator)

Definition at line 108 of file [CardUiState.java](#).

References [CardUiState.abilityImageId](#), [CardUiState.damageBackgroundImageId](#), [CardUiState.damageTextColor](#), [CardUiState.unitId](#), and [CardUiState.UNUSED](#).

5.10.3 Member Function Documentation

5.10.3.1 equals()

```
boolean equals (
```

```
Object o )
```

Checks whether the given Object looks the same as this CardUiState.

Does not compare #unitId since that field does not influence the visual representation of CardUiState objects.

Parameters

<code>o</code>	Object that is being compared with this CardUiState.
----------------	--

Returns

A Boolean defining whether the objects look the same.

Definition at line 212 of file [CardUiState.java](#).

References [CardUiState.abilityImageId](#), [CardUiState.damageBackgroundImageId](#), [CardUiState.damageString](#), [CardUiState.damageTextColor](#), [CardUiState.equals\(\)](#), and [CardUiState.squadString](#).

Referenced by [CardUiState.equals\(\)](#).

5.10.3.2 getAbilityImageId()

```
int getAbilityImageId ( )
```

Returns the drawable resource shown in the ability view.

Returns

An Integer referencing a drawable resource.

See also

[#abilityImageId](#)

[#showAbility\(\)](#)

Definition at line 189 of file [CardUiState.java](#).

References [CardUiState.abilityImageId](#).

Referenced by [CardListAdapter.onBindViewHolder\(\)](#).

5.10.3.3 getDamageBackgroundImageId()

```
int getDamageBackgroundImageId ( )
```

Returns the drawable resource shown as the background of the damage view.

Returns

An Integer referencing a drawable resource.

See also

[#damageBackgroundImageId](#)

Definition at line 158 of file [CardUiState.java](#).

References [CardUiState.damageBackgroundImageId](#).

Referenced by [CardListAdapter.onBindViewHolder\(\)](#).

5.10.3.4 getDamageString()

```
String getDamageString ( )
```

Returns the String shown in the damage view.

Returns

A String containing the damage of the represented UnitEntity or nothing if the unit is empty.

See also

[#damageString](#)

Definition at line 168 of file [CardUiState.java](#).

References [CardUiState.damageString](#).

Referenced by [CardListAdapter.onBindViewHolder\(\)](#).

5.10.3.5 getDamageTextColor()

```
int getDamageTextColor ( )
```

Returns the text color of the damage view.

Returns

An Integer representing a color.

See also

[#damageTextColor](#)

Definition at line 178 of file [CardUiState.java](#).

References [CardUiState.damageTextColor](#).

Referenced by [CardListAdapter.onBindViewHolder\(\)](#).

5.10.3.6 getSquadString()

```
String getSquadString ( )
```

Returns the String shown in the squad view.

Returns

A String containing the squad of the represented UnitEntity or nothing.

See also

[#squadString](#)

[#showSquad\(\)](#)

Definition at line 200 of file [CardUiState.java](#).

References [CardUiState.squadString](#).

Referenced by [CardListAdapter.onBindViewHolder\(\)](#).

5.10.3.7 getUnitId()

```
int getUnitId ( )
```

Returns the #unitId of the represented UnitEntity.

See also

[#unitId](#)

Returns

An Integer referencing the UnitEntity::id of the represented UnitEntity.

Definition at line 148 of file [CardUiState.java](#).

References [CardUiState.unitId](#).

Referenced by [CardListAdapter.onBindViewHolder\(\)](#).

5.10.3.8 showAbility()

```
boolean showAbility ( )
```

Returns whether the ability view is shown, i.e., if the unit has an ability other than Ability::NONE.

Returns

A Boolean defining whether the ability view is shown.

Definition at line 130 of file [CardUiState.java](#).

References [CardUiState.abilityImageId](#), and [CardUiState.UNUSED](#).

Referenced by [CardListAdapter.onBindViewHolder\(\)](#), and [CardUiState.showSquad\(\)](#).

5.10.3.9 showSquad()

```
boolean showSquad ( )
```

Returns whether the squad view is shown, i.e., if the unit has the Ability::BINDING ability.

Returns

A Boolean defining whether the squad view is shown.

Definition at line 139 of file [CardUiState.java](#).

References [CardUiState.showAbility\(\)](#), and [CardUiState.squadString](#).

Referenced by [CardListAdapter.onBindViewHolder\(\)](#).

5.10.4 Member Data Documentation

5.10.4.1 abilityImageId

```
final int abilityImageId [private]
```

Integer referencing the drawable resource shown in the ability view.

May be #UNUSED if the view is not visible.

See also

[#UNUSED](#)

[#getAbilityImageId\(\)](#)

Definition at line 88 of file [CardUiState.java](#).

Referenced by [CardUiState.CardUiState\(\)](#), [CardUiState.equals\(\)](#), [CardUiState.getAbilityImageId\(\)](#), and [CardUiState.showAbility\(\)](#).

5.10.4.2 damageBackgroundImageId

```
final int damageBackgroundImageId [private]
```

Integer referencing the drawable resource shown as the background of the damage view.

See also

[#getDamageBackgroundImageId\(\)](#)

Definition at line 64 of file [CardUiState.java](#).

Referenced by [CardUiState.CardUiState\(\)](#), [CardUiState.equals\(\)](#), and [CardUiState.getDamageBackgroundImageId\(\)](#).

5.10.4.3 damageString

```
final String damageString [private]
```

String containing the number shown in the damage view or nothing if the represented UnitEntity is epic.

See also

[#getDamageString\(\)](#)

Definition at line 72 of file [CardUiState.java](#).

Referenced by [CardUiState.equals\(\)](#), and [CardUiState.getDamageString\(\)](#).

5.10.4.4 damageTextColor

```
final int damageTextColor [private]
```

Integer representing the text color of the damage view.

See also

[#getDamageTextColor\(\)](#)

Definition at line 79 of file [CardUiState.java](#).

Referenced by [CardUiState.CardUiState\(\)](#), [CardUiState.equals\(\)](#), and [CardUiState.getDamageTextColor\(\)](#).

5.10.4.5 DIFF_CALLBACK

```
final DiffUtil.ItemCallback<CardUiState> DIFF_CALLBACK [static]
```

Initial value:

```
= new DiffUtil.ItemCallback<CardUiState>() {
    @Override
    public boolean areItemsTheSame(@NonNull CardUiState oldItem, @NonNull CardUiState newItem) {
        return oldItem.unitId == newItem.unitId;
    }

    @Override
    public boolean areContentsTheSame(@NonNull CardUiState oldItem, @NonNull CardUiState newItem) {
        return Objects.equals(oldItem, newItem);
    }
}
```

DiffUtil.ItemCallback used to compare different CardUiState objects in a ListAdapter, e.g., CardListAdapter.

See also

#unitId
#equals(Object)

Definition at line 33 of file [CardUiState.java](#).

Referenced by [CardListAdapter.CardListAdapter\(\)](#).

5.10.4.6 squadString

```
final String squadString [private]
```

String containing the number shown in the squad view.

See also

#getSquadString()

Definition at line 95 of file [CardUiState.java](#).

Referenced by [CardUiState.equals\(\)](#), [CardUiState.getSquadString\(\)](#), and [CardUiState.showSquad\(\)](#).

5.10.4.7 unitId

```
final int unitId [private]
```

Integer referencing the UnitEntity::id of the represented UnitEntity.

The only member that is not shown in the UI.

See also

#getUnitId()

Definition at line 57 of file [CardUiState.java](#).

Referenced by [CardUiState.CardUiState\(\)](#), and [CardUiState.getUnitId\(\)](#).

5.10.4.8 UNUSED

```
final int UNUSED = -1 [static]
```

Integer that is used instead of a drawable resource to indicate the absence of a meaningful #abilityImage.

See also

#abilityImageId

Definition at line 50 of file [CardUiState.java](#).

Referenced by [CardUiState.CardUiState\(\)](#), [CardUiStateFactory.createCardUiState\(\)](#), and [CardUiState.showAbility\(\)](#).

The documentation for this class was generated from the following file:

- [gwent/ui/dialogs/cards/CardUiState.java](#)

5.11 CardUiStateFactory Class Reference

A factory class responsible for creating CardUiState objects from UnitEntity objects.

Collaboration diagram for CardUiStateFactory:

CardUiStateFactory

Public Member Functions

- [CardUiStateFactory](#) (@NonNull Context context, boolean [weather](#), boolean [horn](#))
Constructor of a CardUiStateFactory for a row with the given weather and horn (de-)buff values.
- List< [CardUiState](#) > [createCardUiState](#) (@NonNull Collection< UnitEntity > units)
Creates a List of CardUiState objects from the given List of UnitEntity objects.
- [CardUiState](#) [createCardUiState](#) (@NonNull UnitEntity unit, @NonNull DamageCalculator calculator)
Creates a CardUiState from the given UnitEntity.

Private Attributes

- final boolean [weather](#)
Boolean defining the status of the weather debuff in the row for which CardUiState objects are created.
- final boolean [horn](#)
Boolean defining the status of the horn buff in the row for which CardUiState objects are created.
- final Map< Color, Integer > [damageTextColors](#) = new HashMap<> (Color.values().length)
Map containing a color integer for each Color value possible, i.e., Color::DEFAULT, Color::BUFFED, and Color::↔DEBUFFED.

5.11.1 Detailed Description

A factory class responsible for creating CardUiState objects from UnitEntity objects.

See also

[CardUiState](#)

Definition at line 28 of file [CardUiStateFactory.java](#).

5.11.2 Constructor & Destructor Documentation

5.11.2.1 CardUiStateFactory()

```
CardUiStateFactory (
    @NonNull Context context,
    boolean weather,
    boolean horn )
```

Constructor of a CardUiStateFactory for a row with the given weather and horn (de-)buff values. The Context parameter is used to retrieve the color values saved in #damageTextColors.

Parameters

<i>context</i>	Context object used to get colors.
<i>weather</i>	Boolean defining whether the weather debuff is active.
<i>horn</i>	Boolean defining whether the horn buff is active.

See also

`Context::getColor(int)`

Definition at line 58 of file [CardUiStateFactory.java](#).

References [CardUiStateFactory.damageTextColors](#), [CardUiStateFactory.horn](#), and [CardUiStateFactory.weather](#).

5.11.3 Member Function Documentation

5.11.3.1 `createCardUiState()` [1/2]

```
List< CardUiState > createCardUiState (
    @NonNull Collection< UnitEntity > units )
```

Creates a List of CardUiState objects from the given List of UnitEntity objects.

Basically calls `#createCardUiState(UnitEntity, DamageCalculator)` for each given UnitEntity.

See also

`#createCardUiState(UnitEntity, DamageCalculator)`

Parameters

<i>units</i>	List of UnitEntity objects that are converted to CardUiState objects.
--------------	---

Returns

A List of CardUiState objects from the given UnitEntity objects.

Definition at line 74 of file [CardUiStateFactory.java](#).

References [CardUiStateFactory.createCardUiState\(\)](#), [CardUiStateFactory.horn](#), and [CardUiStateFactory.weather](#).

Referenced by [CardUiStateFactory.createCardUiState\(\)](#).

5.11.3.2 `createCardUiState()` [2/2]

```
CardUiState createCardUiState (
    @NonNull UnitEntity unit,
    @NonNull DamageCalculator calculator )
```

Creates a CardUiState from the given UnitEntity.

See also

`#createCardUiState(Collection)`

Parameters

<i>unit</i>	UnitEntity that is converted to a CardUiState.
<i>calculator</i>	DamageCalculator used to calculate damage and Color of the given UnitEntity.

Returns

A CardUiState obtained from the given UnitEntity.

Definition at line 89 of file [CardUiStateFactory.java](#).

References [CardUiStateFactory.damageTextColors](#), and [CardUiState.UNUSED](#).

5.11.4 Member Data Documentation

5.11.4.1 `damageTextColors`

```
final Map<Color, Integer> damageTextColors = new HashMap<>(Color.values().length) [private]
```

Map containing a color integer for each Color value possible, i.e., Color::DEFAULT, Color::BUFFED, and Color::↔DEBUFFED.

Definition at line 46 of file [CardUiStateFactory.java](#).

Referenced by [CardUiStateFactory.CardUiStateFactory\(\)](#), and [CardUiStateFactory.createCardUiState\(\)](#).

5.11.4.2 horn

```
final boolean horn [private]
```

Boolean defining the status of the horn buff in the row for which CardUiState objects are created.

Definition at line 39 of file [CardUiStateFactory.java](#).

Referenced by [CardUiStateFactory.CardUiStateFactory\(\)](#), and [CardUiStateFactory.createCardUiState\(\)](#).

5.11.4.3 weather

```
final boolean weather [private]
```

Boolean defining the status of the weather debuff in the row for which CardUiState objects are created.

Definition at line 33 of file [CardUiStateFactory.java](#).

Referenced by [CardUiStateFactory.CardUiStateFactory\(\)](#), and [CardUiStateFactory.createCardUiState\(\)](#).

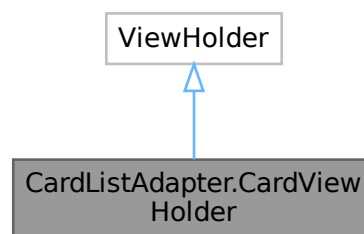
The documentation for this class was generated from the following file:

- gwent/ui/dialogs/cards/CardUiStateFactory.java

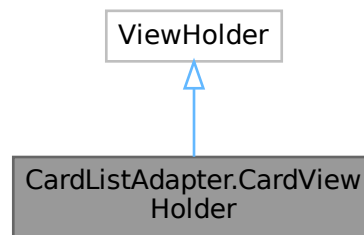
5.12 CardListAdapter.CardViewHolder Class Reference

A RecyclerView.ViewHolder class managing a card view that shows a representation of the UnitEntity referenced by #itemId.

Inheritance diagram for CardListAdapter.CardViewHolder:



Collaboration diagram for CardListAdapter.CardViewHolder:



Package Functions

- `CardViewHolder` (`@NonNull View itemView`, `@NonNull Consumer< Integer > onCopy`, `@NonNull Consumer< Integer > onRemove`)
Constructor of a CardViewHolder for the given View initializing #damageView, #abilityView, and #bindingView.
- `TextView` `getDamageView ()`
Returns the #damageView of the managed View.
- `ImageView` `getAbilityView ()`
Returns the #abilityView of the managed View.
- `TextView` `getBindingView ()`
Returns the #bindingView of the managed View.
- `void` `setItemId (int itemId)`
Sets the #unitId of this ViewHolder.

Private Attributes

- `final TextView` `damageView`
TextView showing the (de-)buffed damage of the represented UnitEntity.
- `final ImageView` `abilityView`
ImageView showing an image of the UnitEntity::ability of the represented UnitEntity.
- `final TextView` `bindingView`
TextView showing the UnitEntity::squad of the represented UnitEntity if UnitEntity::squad is Ability::BINDING.
- `int` `itemId = (int) NO_ID`
Integer referencing the UnitEntity::id of the represented UnitEntity.

5.12.1 Detailed Description

A `RecyclerView.ViewHolder` class managing a card view that shows a representation of the `UnitEntity` referenced by `#itemId`.

See also

`RecyclerView.ViewHolder`

Definition at line 131 of file `CardListAdapter.java`.

5.12.2 Constructor & Destructor Documentation

5.12.2.1 CardViewHolder()

```
CardViewHolder (
    @NonNull View itemView,
    @NonNull Consumer< Integer > onCopy,
    @NonNull Consumer< Integer > onRemove ) [package]
```

Constructor of a CardViewHolder for the given View initializing #damageView, #abilityView, and #bindingView. Sets View.OnClickListener on copy-button and delete-button calling the given onCopy and onRemove callbacks with #unitId.

Parameters

<i>itemView</i>	View with the card layout.
<i>onCopy</i>	Consumer that is called with #unitId, when the copy-button of the given View is clicked.
<i>onRemove</i>	Consumer that is called with #unitId, when the delete-button of the given View is clicked.

Definition at line 173 of file [CardListAdapter.java](#).

References [CardListAdapter.CardViewHolder.abilityView](#), [CardListAdapter.CardViewHolder.bindingView](#), [CardListAdapter.CardViewHolder.itemId](#), [CardListAdapter.onCopy](#), and [CardListAdapter.onRemove](#).

5.12.3 Member Function Documentation

5.12.3.1 getAbilityView()

```
ImageView getAbilityView ( ) [package]
```

Returns the #abilityView of the managed View.

Returns

An ImageView showing the UnitEntity::ability of the represented UnitEntity.

See also

#abilityView

Definition at line 211 of file [CardListAdapter.java](#).

References [CardListAdapter.CardViewHolder.abilityView](#).

5.12.3.2 getBindingView()

```
TextView getBindingView ( ) [package]
```

Returns the #bindingView of the managed View.

Returns

A TextView showing the UnitEntity::squad of the represented UnitEntity.

See also

#bindingView

Definition at line 221 of file [CardListAdapter.java](#).

References [CardListAdapter.CardViewHolder.bindingView](#).

5.12.3.3 getDamageView()

```
TextView getDamageView ( ) [package]
```

Returns the #damageView of the managed View.

Returns

A TextView showing the damage of the represented UnitEntity.

See also

[#damageView](#)

Definition at line 201 of file [CardListAdapter.java](#).

References [CardListAdapter.CardViewHolder.damageView](#).

5.12.3.4 setId()

```
void setId (
    int itemId ) [package]
```

Sets the #unitId of this ViewHolder.

Parameters

<i>itemId</i>	Integer referencing the UnitEntity::id of the represented UnitEntity.
---------------	---

Exceptions

<i>IllegalArgumentException</i>	When itemId is negative.
---------------------------------	--------------------------

See also

[#unitId](#)

Definition at line 232 of file [CardListAdapter.java](#).

References [CardListAdapter.CardViewHolder.unitId](#).

5.12.4 Member Data Documentation**5.12.4.1 abilityView**

```
final ImageView abilityView [private]
```

ImageView showing an image of the UnitEntity::ability of the represented UnitEntity.

If UnitEntity::ability is Ability::NONE, this view's visibility should be View::GONE.

See also

[#getAbilityView\(\)](#)

Definition at line 144 of file [CardListAdapter.java](#).

Referenced by [CardListAdapter.CardViewHolder.CardViewHolder\(\)](#), and [CardListAdapter.CardViewHolder.getAbilityView\(\)](#).

5.12.4.2 bindingView

```
final TextView bindingView [private]
```

TextView showing the UnitEntity::squad of the represented UnitEntity if UnitEntity::squad is Ability::BINDING.

If UnitEntity::ability is not Ability::BINDING, this view's visibility should be View::GONE.

See also

[#getBindingView\(\)](#)

Definition at line 152 of file [CardListAdapter.java](#).

Referenced by [CardListAdapter.CardViewHolder.CardViewHolder\(\)](#), and [CardListAdapter.CardViewHolder.getBindingView\(\)](#).

5.12.4.3 damageView

```
final TextView damageView [private]
```

TextView showing the (de-)buffed damage of the represented UnitEntity.

See also

`#getDamageView()`

Definition at line 137 of file [CardListAdapter.java](#).

Referenced by [CardListAdapter.CardViewHolder.CardViewHolder\(\)](#), and [CardListAdapter.CardViewHolder.getDamageView\(\)](#).

5.12.4.4 itemId

```
int itemId = (int) NO_ID [private]
```

Integer referencing the UnitEntity::id of the represented UnitEntity.

Is initialized with #NO_ID and therefore must be set using `#setItemId()` in `#onBindViewHolder(CardViewHolder, int)`.

See also

`#setItemId(int)`

`#onBindViewHolder(CardViewHolder, int)`

Definition at line 161 of file [CardListAdapter.java](#).

Referenced by [CardListAdapter.CardViewHolder.CardViewHolder\(\)](#), and [CardListAdapter.CardViewHolder.setItemId\(\)](#).

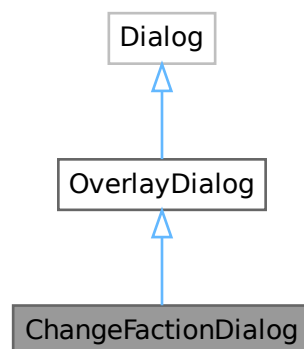
The documentation for this class was generated from the following file:

- `gwent/ui/dialogs/cards/CardListAdapter.java`

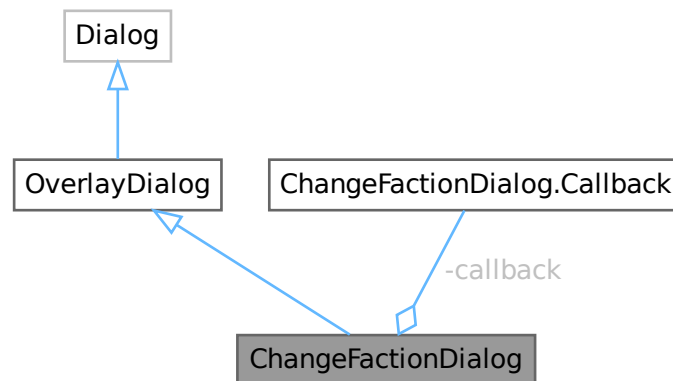
5.13 ChangeFactionDialog Class Reference

An OverlayDialog class used to change the faction design.

Inheritance diagram for ChangeFactionDialog:



Collaboration diagram for ChangeFactionDialog:



Classes

- interface [Callback](#)

An interface used for propagating a selected theme back to the creator of a ChangeFactionDialog.

Public Member Functions

- [ChangeFactionDialog](#) (@NonNull Context context, @NonNull [Callback](#) callback)

Constructor of a ChangeFactionDialog that calls the given Callback when one theme is selected.

Protected Member Functions

- void [onCreate](#) (Bundle savedInstanceState)

Initializes layout and sets listeners for each view.

Protected Member Functions inherited from [OverlayDialog](#)

- [OverlayDialog](#) (@NonNull Context context, @LayoutRes int layout, @IdRes int dismissViewId)

Constructor of an OverlayDialog in the given Context, with the given layout and dismissViewId.

- [OverlayDialog](#) (@NonNull Context context, @LayoutRes int layout)

Constructor of an OverlayDialog in the given Context and with the given layout.

Private Member Functions

- View.OnClickListener [getOnThemeClickListener](#) (@IntRange(from=THEME_MONSTER, to=THEME_SCOIATAEL) int theme)

Returns a View.OnClickListener instance that calls #cancel() and uses #callback to propagate the selected theme.

Private Attributes

- final [Callback](#) callback

Callback that is called when a theme is selected.

Additional Inherited Members

Static Package Attributes inherited from [OverlayDialog](#)

- static final int [NO_DISMISS_VIEW](#) = -1

Integer used as #dismissViewId, when the Dialog shouldn't be cancelable by clicking on a View.

5.13.1 Detailed Description

An OverlayDialog class used to change the faction design.

Definition at line 20 of file [ChangeFactionDialog.java](#).

5.13.2 Constructor & Destructor Documentation

5.13.2.1 ChangeFactionDialog()

```
ChangeFactionDialog (
    @NonNull Context context,
    @NonNull Callback callback )
```

Constructor of a ChangeFactionDialog that calls the given Callback when one theme is selected.

Parameters

<i>context</i>	Context in which this Dialog is run.
<i>callback</i>	Callback that is called when a theme is selected.

Definition at line 33 of file [ChangeFactionDialog.java](#).

References [ChangeFactionDialog.callback](#).

5.13.3 Member Function Documentation

5.13.3.1 getOnThemeClickListener()

```
View.OnClickListener getOnThemeClickListener (
    @IntRange(from=THEME_MONSTER, to=THEME_SCOIATAEL) int theme ) [private]
```

Returns a View.OnClickListener instance that calls #cancel() and uses #callback to propagate the selected theme.

Parameters

<i>theme</i>	Integer representing the selected theme.
--------------	--

Returns

A View.OnClickListener handling theme input.

See also

[#cancel\(\)](#)

[Callback::onThemeSelect\(int\)](#)

Definition at line 73 of file [ChangeFactionDialog.java](#).

References [ChangeFactionDialog.callback](#), and [ChangeFactionDialog.Callback.onThemeSelect\(\)](#).

Referenced by [ChangeFactionDialog.onCreate\(\)](#).

5.13.3.2 onCreate()

```
void onCreate (
    Bundle savedInstanceState ) [protected]
```

Initializes layout and sets listeners for each view.

Parameters

<i>savedInstanceState</i>	If this dialog is being reinitialized after the hosting activity was previously shut down, holds the result from the most recent call to <code>onSaveInstanceState</code> , or null if this is the first time.
---------------------------	--

Reimplemented from [OverlayDialog](#).

Definition at line 47 of file [ChangeFactionDialog.java](#).

References [ChangeFactionDialog.getOnThemeClickListener\(\)](#).

5.13.4 Member Data Documentation

5.13.4.1 callback

```
final Callback callback [private]
```

Callback that is called when a theme is selected.

Definition at line 24 of file [ChangeFactionDialog.java](#).

Referenced by [ChangeFactionDialog.ChangeFactionDialog\(\)](#), and [ChangeFactionDialog.getOnThemeClickListener\(\)](#).

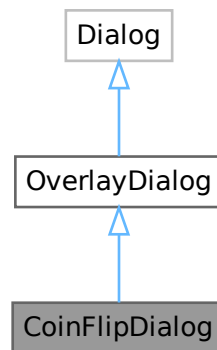
The documentation for this class was generated from the following file:

- [gwent/ui/dialogs/ChangeFactionDialog.java](#)

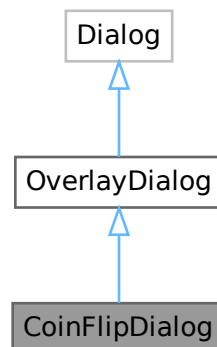
5.14 CoinFlipDialog Class Reference

An [OverlayDialog](#) class showing the result of a coin-flip.

Inheritance diagram for [CoinFlipDialog](#):



Collaboration diagram for CoinFlipDialog:



Public Member Functions

- [CoinFlipDialog](#) (@NonNull Context context)
Constructor of a CoinFlipDialog in the given Context.

Protected Member Functions

- void [onCreate](#) (Bundle savedInstanceState)
Initializes layout and sets listeners for each view.

Protected Member Functions inherited from [OverlayDialog](#)

- [OverlayDialog](#) (@NonNull Context context, @LayoutRes int [layout](#), @IdRes int [dismissViewId](#))
Constructor of an OverlayDialog in the given Context, with the given layout and dismissViewId.
- [OverlayDialog](#) (@NonNull Context context, @LayoutRes int [layout](#))
Constructor of an OverlayDialog in the given Context and with the given layout.

Static Private Attributes

- static final Random [RANDOM](#) = new Random()
Random used to decide the result of the coin-flip.

Additional Inherited Members

Static Package Attributes inherited from [OverlayDialog](#)

- static final int [NO_DISMISS_VIEW](#) = -1
Integer used as #dismissViewId, when the Dialog shouldn't be cancelable by clicking on a View.

5.14.1 Detailed Description

An OverlayDialog class showing the result of a coin-flip. Shows heads or tails with 48.5% probability respectively. The remaining 3% are divided equally between three tie situations.

See also

R.layout::popup_coin_normal
 R.layout::popup_coin_stewie
 R.layout::popup_coin_terry
 R.layout::popup_coin_vin

Definition at line 24 of file [CoinFlipDialog.java](#).

5.14.2 Constructor & Destructor Documentation

5.14.2.1 CoinFlipDialog()

```
CoinFlipDialog (
    @NonNull Context context )
```

Constructor of a CoinFlipDialog in the given Context.

Parameters

<i>context</i>	Context in which the Dialog is run.
----------------	-------------------------------------

Definition at line 34 of file [CoinFlipDialog.java](#).

5.14.3 Member Function Documentation

5.14.3.1 onCreate()

```
void onCreate (
    Bundle savedInstanceState ) [protected]
```

Initializes layout and sets listeners for each view.

The result of the coin-flip is decided here.

Parameters

<i>savedInstanceState</i>	If this dialog is being reinitialized after the hosting activity was previously shut down, holds the result from the most recent call to <code>onSaveInstanceState</code> , or null if this is the first time.
---------------------------	--

Reimplemented from [OverlayDialog](#).

Definition at line 48 of file [CoinFlipDialog.java](#).

References [CoinFlipDialog.RANDOM](#).

5.14.4 Member Data Documentation

5.14.4.1 RANDOM

```
final Random RANDOM = new Random() [static], [private]
```

Random used to decide the result of the coin-flip.

Definition at line 28 of file [CoinFlipDialog.java](#).

Referenced by [CoinFlipDialog.onCreate\(\)](#).

The documentation for this class was generated from the following file:

- gwent/ui/dialogs/CoinFlipDialog.java

5.15 DamageCalculator.Color Enum Reference

An enum containing values representing whether a unit is buffed, de-buffed or not affected by status effects.

Collaboration diagram for DamageCalculator.Color:

DamageCalculator.Color

Public Attributes

- [DEFAULT](#)
The unit is not affected by status effects.
- [BUFFED](#)
The unit is buffed by status effects.
- [DEBUFFED](#)
The unit is de-buffed by weather effects.

5.15.1 Detailed Description

An enum containing values representing whether a unit is buffed, de-buffed or not affected by status effects.
Definition at line 49 of file [DamageCalculator.java](#).

5.15.2 Member Data Documentation

5.15.2.1 BUFFED

`BUFFED`
The unit is buffed by status effects.
Definition at line 57 of file [DamageCalculator.java](#).

5.15.2.2 DEBUFFED

`DEBUFFED`
The unit is de-buffed by weather effects.
Definition at line 62 of file [DamageCalculator.java](#).

5.15.2.3 DEFAULT

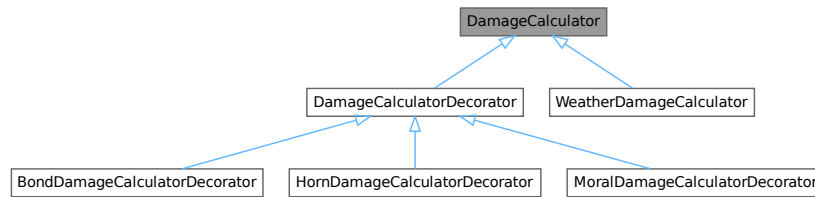
`DEFAULT`
The unit is not affected by status effects.
Definition at line 53 of file [DamageCalculator.java](#).
The documentation for this enum was generated from the following file:

- gwent/domain/damage/DamageCalculator.java

5.16 DamageCalculator Interface Reference

An interface for a class capable of calculating the (de-)buffed damage of a unit for a given `UnitEntity::id` and `UnitEntity::damage`.

Inheritance diagram for DamageCalculator:



Collaboration diagram for DamageCalculator:



Classes

- enum [Color](#)

An enum containing values representing whether a unit is buffed, de-buffed or not affected by status effects.

Public Member Functions

- [Color isBuffed](#) (int id)
Calculates whether the unit with the given id is shown as Color::BUFFED, Color::DEBUFFED, or Color::DEFAULT.
- int [calculateDamage](#) (int id, @IntRange(from=0) int damage)
Calculates the (de-)buffed damage of unit with the given id and the given base-damage.

5.16.1 Detailed Description

An interface for a class capable of calculating the (de-)buffed damage of a unit for a given `UnitEntity::id` and `UnitEntity::damage`.

Is used to implement the visitor design pattern, see [com.peternaggschga.gwent.data.UnitEntity::calculateDamage\(\)](#)

The respective visitors are created as a decorator hierarchy, see `DamageCalculatorBuildDirector::getCalculator()`.

See also

`com.peternaggschga.gwent.data.UnitEntity::calculateDamage(DamageCalculator)`

`DamageCalculatorBuildDirector::getCalculator(boolean, boolean, Collection)`

Definition at line 17 of file [DamageCalculator.java](#).

5.16.2 Member Function Documentation

5.16.2.1 calculateDamage()

```

int calculateDamage (
    int id,
    @IntRange(from=0) int damage )
  
```

Calculates the (de-)buffed damage of unit with the given id and the given base-damage. Calculation is defined by the underlying decorator structure implemented via [DamageCalculatorDecorator](#) and [WeatherDamageCalculator](#).

Parameters

<i>id</i>	Integer representing the <code>UnitEntity::id</code> of the unit whose (de-)buff damage is calculated.
<i>damage</i>	Integer representing the base-damage of the unit whose (de-)buff damage is calculated.

Returns

Integer representing the (de-)buffed damage of the unit.

See also

[DamageCalculatorDecorator](#)
[WeatherDamageCalculator](#)

Implemented in [BondDamageCalculatorDecorator](#), [HornDamageCalculatorDecorator](#), [MoralDamageCalculatorDecorator](#), and [WeatherDamageCalculator](#).

Referenced by [BondDamageCalculatorDecorator.calculateDamage\(\)](#), [HornDamageCalculatorDecorator.calculateDamage\(\)](#), and [MoralDamageCalculatorDecorator.calculateDamage\(\)](#).

5.16.2.2 isBuffed()

```
Color isBuffed (
    int id )
```

Calculates whether the unit with the given id is shown as `Color::BUFFED`, `Color::DEBUFFED`, or `Color::DEFAULT`. Units are shown as `Color::DEFAULT` when they are not affected by any damage changing buffs or de-buffs. Units are shown as `Color::DEBUFFED` when they are only affected by the weather de-buff. Units are shown as `Color::↔ BUFFED` when they are affected by any damage-increasing buff.

Parameters

<i>id</i>	Integer representing the <code>UnitEntity::id</code> of the unit buff status is calculated.
-----------	---

Returns

Color representing whether the unit is buffed, de-buffed or not affected.

See also

[Color](#)

Implemented in [BondDamageCalculatorDecorator](#), [HornDamageCalculatorDecorator](#), [MoralDamageCalculatorDecorator](#), and [WeatherDamageCalculator](#).

Referenced by [BondDamageCalculatorDecorator.isBuffed\(\)](#), [HornDamageCalculatorDecorator.isBuffed\(\)](#), and [MoralDamageCalculatorDecorator.isBuffed\(\)](#).

The documentation for this interface was generated from the following file:

- `gwent/domain/damage/DamageCalculator.java`

5.17 DamageCalculatorBuildDirector Class Reference

A build director responsible for the creation of [DamageCalculator](#) instances from sets of units using [Damage↔ CalculatorBuilder](#).

Collaboration diagram for DamageCalculatorBuildDirector:

DamageCalculatorBuildDirector

Static Public Member Functions

- static [DamageCalculator](#) [getCalculator](#) (boolean weather, boolean horn, @NonNull Collection< UnitEntity > units)

Creates a DamageCalculator for a row with the given weather status, horn status and units.

Static Private Member Functions

- static void [setSquads](#) (@NonNull Collection< UnitEntity > units, @NonNull [DamageCalculatorBuilder](#) builder)

Uses the given units to add a BondDamageCalculatorDecorator to the builder using DamageCalculatorBuilder::setBond().

- static void [setMoralBoosts](#) (@NonNull Collection< UnitEntity > units, @NonNull [DamageCalculatorBuilder](#) builder)

Uses the given units to add a MoralDamageCalculatorDecorator to the builder using DamageCalculatorBuilder::setMoral().

- static void [setHorns](#) (boolean horn, @NonNull Collection< UnitEntity > units, @NonNull [DamageCalculatorBuilder](#) builder)

Uses the given commander's horn status and the given units to add a HornDamageCalculatorDecorator to the builder using DamageCalculatorBuilder::setHorn().

5.17.1 Detailed Description

A build director responsible for the creation of DamageCalculator instances from sets of units using DamageCalculatorBuilder.

Encapsulates the logic for correct order of decorators.

See also

[DamageCalculatorBuilder](#)

Definition at line 21 of file [DamageCalculatorBuildDirector.java](#).

5.17.2 Member Function Documentation

5.17.2.1 getCalculator()

```
static DamageCalculator getCalculator (
    boolean weather,
    boolean horn,
    @NonNull Collection< UnitEntity > units ) [static]
```

Creates a DamageCalculator for a row with the given weather status, horn status and units.

Parameters

<i>weather</i>	Boolean defining whether the calculation encompasses the weather debuff.
<i>horn</i>	Boolean defining whether a commander's horn is in the row.
<i>units</i>	Collection of UnitEntity objects that are in the row.

Returns

A DamageCalculator object capable of calculating damage for the given units.

Definition at line 31 of file [DamageCalculatorBuildDirector.java](#).

References [DamageCalculatorBuilder.getResult\(\)](#), [DamageCalculatorBuildDirector.setHorns\(\)](#), [DamageCalculatorBuildDirector.setMorale\(\)](#), [DamageCalculatorBuildDirector.setSquads\(\)](#), and [DamageCalculatorBuilder.setWeather\(\)](#).

5.17.2.2 setHorns()

```
static void setHorns (
    boolean horn,
    @NonNull Collection< UnitEntity > units,
    @NonNull DamageCalculatorBuilder builder ) [static], [private]
```

Uses the given commander's horn status and the given units to add a HornDamageCalculatorDecorator to the builder using DamageCalculatorBuilder::setHorn().

To accomplish this, the given unit list is converted to a List of the ids of the units with the Ability::HORN Ability (including null if horn is true).

Parameters

<i>horn</i>	Boolean defining whether a commander's horn is in the row.
<i>units</i>	Collection of UnitEntity objects that are in the row.
<i>builder</i>	DamageCalculatorBuilder where the horn buff is added.

See also

DamageCalculatorBuilder::setHorn(List)

Definition at line 106 of file [DamageCalculatorBuildDirector.java](#).

Referenced by [DamageCalculatorBuildDirector.getCalculator\(\)](#).

5.17.2.3 setMoralBoosts()

```
static void setMoralBoosts (
    @NonNull Collection< UnitEntity > units,
    @NonNull DamageCalculatorBuilder builder ) [static], [private]
```

Uses the given units to add a MoralDamageCalculatorDecorator to the builder using DamageCalculatorBuilder::setMoral().

To accomplish this, the given unit list is converted to a List of the ids of the units with the Ability::MORAL_BOOST Ability.

Parameters

<i>units</i>	Collection of UnitEntity objects that are in the row.
<i>builder</i>	DamageCalculatorBuilder where the horn buff is added.

See also

DamageCalculatorBuilder::setMoral(List)

Definition at line 85 of file [DamageCalculatorBuildDirector.java](#).

Referenced by [DamageCalculatorBuildDirector.getCalculator\(\)](#).

5.17.2.4 setSquads()

```
static void setSquads (
    @NonNull Collection< UnitEntity > units,
    @NonNull DamageCalculatorBuilder builder ) [static], [private]
```

Uses the given units to add a BondDamageCalculatorDecorator to the builder using `DamageCalculatorBuilder::setBond()`.

To accomplish this, the given unit list is converted to a Map from ids of units with the `Ability::BINDING` Ability to the respective squad size.

Parameters

<i>units</i>	Collection of UnitEntity objects that are in the row.
<i>builder</i>	DamageCalculatorBuilder where the horn buff is added.

See also

`DamageCalculatorBuilder::setBond(Map)`

Definition at line 61 of file [DamageCalculatorBuildDirector.java](#).

Referenced by [DamageCalculatorBuildDirector.getCalculator\(\)](#).

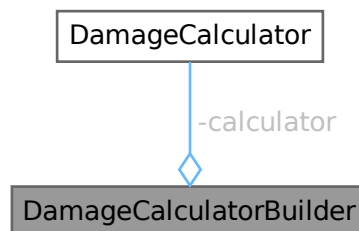
The documentation for this class was generated from the following file:

- `gwent/domain/damage/DamageCalculatorBuildDirector.java`

5.18 DamageCalculatorBuilder Class Reference

A builder class responsible for creating a decorator structure of DamageCalculator classes with the given parameters.

Collaboration diagram for DamageCalculatorBuilder:



Package Functions

- void [setWeather](#) (boolean weather)
Sets the weather debuff for the calculation.
- void [setBond](#) (@NonNull Map< Integer, Integer > idToSquad)
Sets the tight bond buff for the calculation.
- void [setMoral](#) (@NonNull List< Integer > unitIds)
Sets the moral boost buff for the calculation.
- void [setHorn](#) (@NonNull List< Integer > unitIds)
Sets the commander's horn buff for the calculation.
- [DamageCalculator getResult](#) ()
Returns the current state of the built DamageCalculator, i.e.

Private Attributes

- `DamageCalculator calculator = new WeatherDamageCalculator(false)`

The DamageCalculator instance that is being built by this DamageCalculatorBuilder.

5.18.1 Detailed Description

A builder class responsible for creating a decorator structure of DamageCalculator classes with the given parameters.

Should only be used by DamageCalculatorBuildDirector. Calls to `#setWeather()`, `#setBond()`, `#setMoral()` or `#setHorn()` should be in that exact order to create a correct DamageCalculator. Every function call is optional, i.e. every function should be called once or not at all.

See also

DamageCalculatorBuildDirector

DamageCalculator

DamageCalculatorDecorator

Definition at line 19 of file [DamageCalculatorBuilder.java](#).

5.18.2 Member Function Documentation

5.18.2.1 getResult()

```
DamageCalculator getResult ( ) [package]
```

Returns the current state of the built DamageCalculator, i.e. `#calculator`.

Returns

A DamageCalculator that has been built by this builder.

Definition at line 72 of file [DamageCalculatorBuilder.java](#).

References [DamageCalculatorBuilder.calculator](#).

Referenced by [DamageCalculatorBuildDirector.getCalculator\(\)](#).

5.18.2.2 setBond()

```
void setBond (
    @NonNull Map< Integer, Integer > idToSquad ) [package]
```

Sets the tight bond buff for the calculation.

Should be called after `#setWeather()` or never.

Parameters

<i>idToSquad</i>	Map mapping the ids of all units with the com.paternaggschga.gwent.data.Ability::BINDING ability to the respective squad size.
------------------	--

Definition at line 41 of file [DamageCalculatorBuilder.java](#).

References [DamageCalculatorBuilder.calculator](#).

5.18.2.3 setHorn()

```
void setHorn (
    @NonNull List< Integer > unitIds ) [package]
```

Sets the commander's horn buff for the calculation.

Should be called after `#setMoral()` or never.

Parameters

<i>unitIds</i>	List of Integers representing ids of units with the com.peternaggschga.gwent.data.Ability::HORN ability.
----------------	--

Definition at line 62 of file [DamageCalculatorBuilder.java](#).

References [DamageCalculatorBuilder.calculator](#).

5.18.2.4 setMoral()

```
void setMoral (
    @NonNull List< Integer > unitIds ) [package]
```

Sets the moral boost buff for the calculation.

Should be called after #setBond() or never.

Parameters

<i>unitIds</i>	List of Integers representing ids of units with the com.peternaggschga.gwent.data.Ability::MORAL_BOOST ability.
----------------	---

Definition at line 51 of file [DamageCalculatorBuilder.java](#).

References [DamageCalculatorBuilder.calculator](#).

5.18.2.5 setWeather()

```
void setWeather (
    boolean weather ) [package]
```

Sets the weather debuff for the calculation.

Should be called first or never.

Parameters

<i>weather</i>	Boolean defining whether the weather debuff is active.
----------------	--

Definition at line 32 of file [DamageCalculatorBuilder.java](#).

References [DamageCalculatorBuilder.calculator](#).

Referenced by [DamageCalculatorBuildDirector.getCalculator\(\)](#).

5.18.3 Member Data Documentation**5.18.3.1 calculator**

```
DamageCalculator calculator = new WeatherDamageCalculator(false) [private]
```

The DamageCalculator instance that is being built by this DamageCalculatorBuilder.

Starts as a WeatherDamageCalculator, since this is always at the end of the decorator chain.

Definition at line 25 of file [DamageCalculatorBuilder.java](#).

Referenced by [DamageCalculatorBuilder.getResult\(\)](#), [DamageCalculatorBuilder.setBond\(\)](#), [DamageCalculatorBuilder.setHorn\(\)](#), [DamageCalculatorBuilder.setMoral\(\)](#), and [DamageCalculatorBuilder.setWeather\(\)](#).

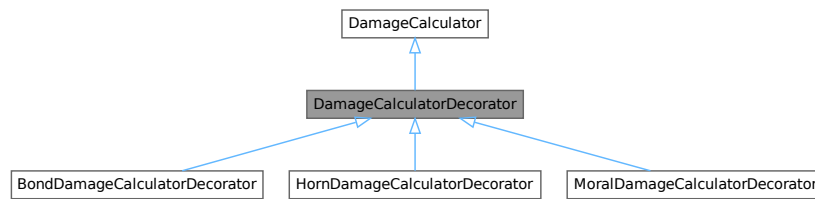
The documentation for this class was generated from the following file:

- gwent/domain/damage/DamageCalculatorBuilder.java

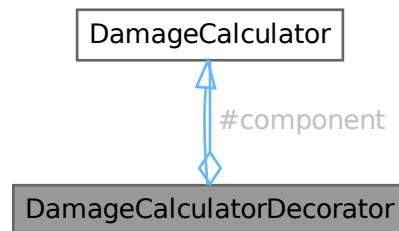
5.19 DamageCalculatorDecorator Class Reference

An abstract DamageCalculator that is used as a superclass for the decorator classes implementing the damage calculation.

Inheritance diagram for DamageCalculatorDecorator:



Collaboration diagram for DamageCalculatorDecorator:



Protected Attributes

- final `DamageCalculator component`
A `DamageCalculator` which is decorated by this decorator.

Package Functions

- `DamageCalculatorDecorator` (@NonNull `DamageCalculator component`)
Constructor of a `DamageCalculatorDecorator`.

Additional Inherited Members

Public Member Functions inherited from `DamageCalculator`

- `Color isBuffed` (int id)
Calculates whether the unit with the given id is shown as `Color::BUFFED`, `Color::DEBUFFED`, or `Color::DEFAULT`.
- int `calculateDamage` (int id, @IntRange(from=0) int damage)
Calculates the (de-)buffed damage of unit with the given id and the given base-damage.

5.19.1 Detailed Description

An abstract `DamageCalculator` that is used as a superclass for the decorator classes implementing the damage calculation.

Definition at line 9 of file `DamageCalculatorDecorator.java`.

5.19.2 Constructor & Destructor Documentation

5.19.2.1 DamageCalculatorDecorator()

```
DamageCalculatorDecorator (
    @NonNull DamageCalculator component ) [package]
```

Constructor of a DamageCalculatorDecorator.

Parameters

<i>component</i>	DamageCalculator that is being decorated by this decorator.
------------------	---

Definition at line 20 of file [DamageCalculatorDecorator.java](#).

References [DamageCalculatorDecorator.component](#).

5.19.3 Member Data Documentation

5.19.3.1 component

```
final DamageCalculator component [protected]
```

A DamageCalculator which is decorated by this decorator.

Definition at line 13 of file [DamageCalculatorDecorator.java](#).

Referenced by [BondDamageCalculatorDecorator.BondDamageCalculatorDecorator\(\)](#), [DamageCalculatorDecorator.DamageCalculatorDecorator\(\)](#), [HornDamageCalculatorDecorator.HornDamageCalculatorDecorator\(\)](#), [MoralDamageCalculatorDecorator.MoralDamageCalculatorDecorator\(\)](#), [BondDamageCalculatorDecorator.calculateDamage\(\)](#), [HornDamageCalculatorDecorator.calculateDamage\(\)](#), [MoralDamageCalculatorDecorator.calculateDamage\(\)](#), [BondDamageCalculatorDecorator.isBuffed\(\)](#), [HornDamageCalculatorDecorator.isBuffed\(\)](#) and [MoralDamageCalculatorDecorator.isBuffed\(\)](#).

The documentation for this class was generated from the following file:

- gwent/domain/damage/DamageCalculatorDecorator.java

5.20 DamageCalculatorUseCase Class Reference

A use case class responsible for creating DamageCalculator objects for rows using DamageCalculatorBuildDirector.

Collaboration diagram for DamageCalculatorUseCase:

DamageCalculatorUseCase

Static Public Member Functions

- static `Single< DamageCalculator >` [getDamageCalculator](#) (@NonNull `UnitRepository` repository, @NonNull `RowType` row)
Returns a Single emitting a DamageCalculator object for the given row with properties retrieved from the given UnitRepository.
- static `DamageCalculator` [getDamageCalculator](#) (boolean weather, boolean horn, @NonNull `Collection< UnitEntity >` units)
Creates a DamageCalculator for a row with the given weather status, horn status and units.

5.20.1 Detailed Description

A use case class responsible for creating DamageCalculator objects for rows using DamageCalculatorBuildDirector.

See also

DamageCalculatorBuildDirector
DamageCalculator

Definition at line 23 of file [DamageCalculatorUseCase.java](#).

5.20.2 Member Function Documentation

5.20.2.1 getDamageCalculator() [1/2]

```
static Single< DamageCalculator > getDamageCalculator (
    @NonNull UnitRepository repository,
    @NonNull RowType row ) [static]
```

Returns a Single emitting a DamageCalculator object for the given row with properties retrieved from the given UnitRepository.

Parameters

<i>repository</i>	UnitRepository used for data collection.
<i>row</i>	RowType defining which row is queried for DamageCalculator generation.

Returns

A Single emitting a DamageCalculator object for the given row.

Definition at line 32 of file [DamageCalculatorUseCase.java](#).

References [DamageCalculatorUseCase.getDamageCalculator\(\)](#).

Referenced by [BurnDialogUseCase.getBurnUnits\(\)](#), and [DamageCalculatorUseCase.getDamageCalculator\(\)](#).

5.20.2.2 getDamageCalculator() [2/2]

```
static DamageCalculator getDamageCalculator (
    boolean weather,
    boolean horn,
    @NonNull Collection< UnitEntity > units ) [static]
```

Creates a DamageCalculator for a row with the given weather status, horn status and units. Basically calls DamageCalculatorBuildDirector::getCalculator with the given parameters.

Parameters

<i>weather</i>	Boolean defining whether the calculation encompasses the weather debuff.
<i>horn</i>	Boolean defining whether a commander's horn is in the row.
<i>units</i>	Collection of UnitEntity objects that are in the row.

Returns

A DamageCalculator object capable of calculating damage for the given units.

See also

DamageCalculatorBuildDirector::getCalculator(boolean, boolean, Collection)

Definition at line 50 of file [DamageCalculatorUseCase.java](#).

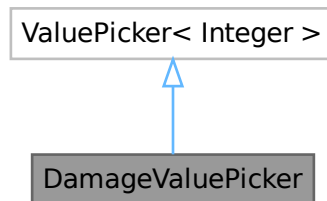
The documentation for this class was generated from the following file:

- gwent/domain/cases/DamageCalculatorUseCase.java

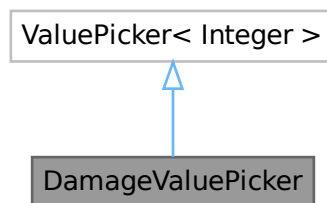
5.21 DamageValuePicker Class Reference

A ValuePicker used for selection of damage values for UnitEntity::damage.

Inheritance diagram for DamageValuePicker:



Collaboration diagram for DamageValuePicker:



Protected Member Functions

- String [getDisplayString](#) (@NonNull Integer value)
Returns a localized String representing the given value when #epicValues is false or otherwise the corresponding epic damage value.

Package Functions

- [DamageValuePicker](#) (@NonNull NumberPicker picker)
Constructor of a DamageValuePicker wrapping the given NumberPicker.
- void [setValue](#) (@NonNull @IntRange(from=0, to=NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) Integer value)
Sets the picker to the given value.
- Integer [getValue](#) ()
Returns the currently selected value.
- void [setEpicValues](#) (boolean [epicValues](#))
Switches the selectableValues between non-epic and epic damage values.
- void [setSelectableValues](#) (@NonNull Collection< Integer > values, @Nullable Integer defaultValue)
Does nothing.

Private Attributes

- boolean `epicValues` = false

Boolean defining whether or not this DamageValuePicker shows epic damage values.

5.21.1 Detailed Description

A ValuePicker used for selection of damage values for UnitEntity::damage.

Allows for switching between epic and normal damage values using `#setEpicValues()`.

Definition at line 25 of file [DamageValuePicker.java](#).

5.21.2 Constructor & Destructor Documentation

5.21.2.1 DamageValuePicker()

```
DamageValuePicker (
    @NonNull NumberPicker picker ) [package]
```

Constructor of a DamageValuePicker wrapping the given NumberPicker.

Calls `#setEpicValues()` to initialize the NumberPicker with non-epic damage values.

See also

`ValuePicker::ValuePicker(NumberPicker, SortedMap)`

`#setEpicValues(boolean)`

Parameters

<i>picker</i>	NumberPicker wrapped by the created DamageValuePicker.
---------------	--

Definition at line 39 of file [DamageValuePicker.java](#).

References [DamageValuePicker.setEpicValues\(\)](#).

5.21.3 Member Function Documentation

5.21.3.1 getDisplayString()

```
String getDisplayString (
    @NonNull Integer value ) [protected]
```

Returns a localized String representing the given value when `#epicValues` is false or otherwise the corresponding epic damage value.

Parameters

<i>value</i>	Value that should be represented as a String.
--------------	---

Returns

A localized String representing the value.

Exceptions

<i>IllegalStateException</i>	When <code>#epicValues</code> is true and <code>displayIntegers</code> does not contain a mapping for the given value.
------------------------------	--

Definition at line 57 of file [DamageValuePicker.java](#).

References [DamageValuePicker.epicValues](#).

5.21.3.2 getValue()

`Integer getValue () [package]`
Returns the currently selected value.

Returns

A value that is selected in picker.

Definition at line 90 of file [DamageValuePicker.java](#).

References [DamageValuePicker.epicValues](#).

Referenced by [CardNumberPickerAdapter.addSelectedUnits\(\)](#).

5.21.3.3 setEpicValues()

`void setEpicValues (
 boolean epicValues) [package]`

Switches the selectableValues between non-epic and epic damage values.
Sets #epicValues.

See also

#epicValues

Parameters

<i>epicValues</i>	Boolean defining whether or not epic values are shown.
-------------------	--

Definition at line 100 of file [DamageValuePicker.java](#).

References [DamageValuePicker.epicValues](#).

Referenced by [CardNumberPickerAdapter.CardNumberPickerAdapter\(\)](#), and [DamageValuePicker.DamageValuePicker\(\)](#).

5.21.3.4 setSelectableValues()

`void setSelectableValues (
 @NonNull Collection< Integer > values,
 @Nullable Integer defaultValue) [package]`

Does nothing.

Overrides parent since selectableValues should only be modified by #setEpicValues().

Parameters

<i>values</i>	Collection of the new selectable values.
<i>defaultValue</i>	Value that is shown in the beginning. If null, then the first value defined by the Comparable interface is used.

Definition at line 121 of file [DamageValuePicker.java](#).

5.21.3.5 setValue()

`void setValue (
 @NonNull @IntRange(from=0, to=NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) Integer value)
 [package]`

Sets the picker to the given value.

If #epicValues is true, ValuePicker::setValue() is called. Otherwise the picker is simply set to the given value.

Parameters

<i>value</i>	Value that the picker is set to.
--------------	----------------------------------

Exceptions

<i>IllegalStateException</i>	When #epicValues is false and the given value is not in [0,20].
------------------------------	---

See also

ValuePicker::setValue(Comparable)

Definition at line 74 of file [DamageValuePicker.java](#).

References [DamageValuePicker.epicValues](#).

5.21.4 Member Data Documentation

5.21.4.1 epicValues

```
boolean epicValues = false [private]
```

Boolean defining whether or not this DamageValuePicker shows epic damage values.

See also

#setEpicValues(boolean)

Definition at line 30 of file [DamageValuePicker.java](#).

Referenced by [DamageValuePicker.getDisplayString\(\)](#), [DamageValuePicker.getValue\(\)](#), [DamageValuePicker.setEpicValues\(\)](#), and [DamageValuePicker.setValue\(\)](#).

The documentation for this class was generated from the following file:

- gwent/ui/dialogs/addcard/DamageValuePicker.java

5.22 ResetDialogUseCase.DialogType Enum Reference

An Enum defining which form of Dialog should be shown.

Collaboration diagram for ResetDialogUseCase.DialogType:

ResetDialogUseCase.DialogType

Public Attributes

- [NONE](#)
No Dialog must be invoked.
- [DEFAULT](#)
A default Dialog asking whether to reset should be invoked.
- [MONSTER](#)
A monster Dialog asking whether to reset and whether to invoke the monster perk, should be invoked.

5.22.1 Detailed Description

An Enum defining which form of Dialog should be shown.

See also

#getDialogType(Context, UnitRepository, Trigger)

Definition at line 148 of file [ResetDialogUseCase.java](#).

5.22.2 Member Data Documentation

5.22.2.1 DEFAULT

DEFAULT

A default Dialog asking whether to reset should be invoked.

Definition at line 156 of file [ResetDialogUseCase.java](#).

Referenced by [ResetDialogUseCase.getDialogType\(\)](#).

5.22.2.2 MONSTER

MONSTER

A monster Dialog asking whether to reset and whether to invoke the monster perk, should be invoked.

Definition at line 162 of file [ResetDialogUseCase.java](#).

Referenced by [ResetDialogUseCase.reset\(\)](#).

5.22.2.3 NONE

NONE

No Dialog must be invoked.

Definition at line 152 of file [ResetDialogUseCase.java](#).

Referenced by [ResetDialogUseCase.getDialogType\(\)](#), and [ResetDialogUseCase.reset\(\)](#).

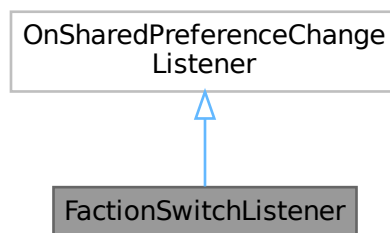
The documentation for this enum was generated from the following file:

- gwent/domain/cases/ResetDialogUseCase.java

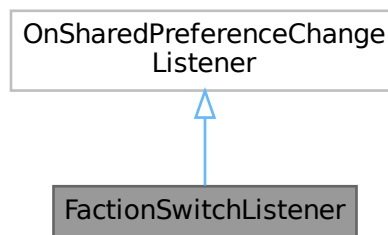
5.23 FactionSwitchListener Class Reference

A class implementing `SharedPreferences.OnSharedPreferencesChangeListener` for the `#THEME_PREFERENCE↵_KEY` `SharedPreferences`.

Inheritance diagram for `FactionSwitchListener`:



Collaboration diagram for FactionSwitchListener:



Public Member Functions

- void `onSharedPreferenceChanged` (SharedPreferences sharedPreferences, @Nullable String key)
Called when a shared preference is changed, added, or removed.

Static Public Member Functions

- static `FactionSwitchListener` `getListener` (@NonNull Window mainWindow)
Returns a new FactionSwitchListener instance for the given Window.
- static void `setTheme` (@NonNull Context context)
Changes the theme of the content referenced by the given Context depending on the preferences obtainable by said Context.

Static Public Attributes

- static final String `THEME_PREFERENCE_KEY` = "theme"
String constant defining the key of the theme SharedPreferences.
- static final int `THEME_MONSTER` = 0
Integer constant representing the Monster theme.
- static final int `THEME_NILFGAARD` = 1
Integer constant representing the Nilfgaard theme.
- static final int `THEME_NORTHERN_KINGDOMS` = 2
Integer constant representing the Northern Kingdoms theme.
- static final int `THEME_SCOIATAEL` = 3
Integer constant representing the Scoia'tael theme.

Private Member Functions

- `FactionSwitchListener` (@NonNull List< ImageView > `ballViews`, @NonNull List< ImageView > `cardViews`, @NonNull List< TextView > `unitViews`, @NonNull ImageButton `factionButton`)
Constructor of a FactionSwitchListener updating the given View objects when #onSharedPreferenceChanged() is called.

Private Attributes

- final List< ImageView > `ballViews`
List of ImageView objects that show a colored ball.
- final List< ImageView > `cardViews`

List of ImageView objects that show a card back.

- final List< TextView > [unitViews](#)

List of TextView objects that show the number of units in a certain color.

- final ImageButton [factionButton](#)

ImageButton showing the logo of the current faction.

5.23.1 Detailed Description

A class implementing SharedPreferences.OnSharedPreferencesChangeListener for the #THEME_PREFERENCE↵_KEY SharedPreferences.

When this Preference is updated, the theme is switched in an animated way using ImageViewSwitchAnimator.

See also

[ImageViewSwitchAnimator](#)

Definition at line 31 of file [FactionSwitchListener.java](#).

5.23.2 Constructor & Destructor Documentation

5.23.2.1 FactionSwitchListener()

```
FactionSwitchListener (
    @NonNull List< ImageView > ballViews,
    @NonNull List< ImageView > cardViews,
    @NonNull List< TextView > unitViews,
    @NonNull ImageButton factionButton ) [private]
```

Constructor of a FactionSwitchListener updating the given View objects when #onSharedPreferencesChanged() is called.

Should only be used by factory method #getListener().

See also

[#getListener\(Window\)](#)

Parameters

<i>ballViews</i>	List of ImageView objects showing a colored ball.
<i>cardViews</i>	List of ImageView objects showing the backside of a card.
<i>unitViews</i>	List of TextView objects showing the number of units.
<i>factionButton</i>	ImageButton that shows the current faction logo.

Definition at line 108 of file [FactionSwitchListener.java](#).

References [FactionSwitchListener.ballViews](#), [FactionSwitchListener.cardViews](#), [FactionSwitchListener.factionButton](#), and [FactionSwitchListener.unitViews](#).

5.23.3 Member Function Documentation

5.23.3.1 getListener()

```
static FactionSwitchListener getListener (
    @NonNull Window mainWindow ) [static]
```

Returns a new FactionSwitchListener instance for the given Window.

Factory method of FactionSwitchListener.

Parameters

<i>mainWindow</i>	Window, that is updated by the new FactionSwitchListener.
-------------------	---

Returns

A FactionSwitchListener instance for the given Window.

Definition at line 124 of file [FactionSwitchListener.java](#).

References [FactionSwitchListener.ballViews](#), [FactionSwitchListener.cardViews](#), and [FactionSwitchListener.unitViews](#).

Referenced by [MainActivity.onCreate\(\)](#).

5.23.3.2 onSharedPreferenceChanged()

```
void onSharedPreferenceChanged (
    SharedPreferences sharedPreferences,
    @Nullable String key )
```

Called when a shared preference is changed, added, or removed.

Only handles changes when the given key is equal to #THEME_PREFERENCE_KEY. Updates the View objects in #ballViews, #cardViews, #unitViews, and #factionButton using ImageViewSwitchAnimator.

See also

[ImageViewSwitchAnimator](#)

Parameters

<i>sharedPreferences</i>	SharedPreferences that received the change.
<i>key</i>	String representing the key of the preference that was changed, added, or removed.

Definition at line 176 of file [FactionSwitchListener.java](#).

References [ImageViewSwitchAnimator.animatedSwitch\(\)](#), [FactionSwitchListener.ballViews](#), [FactionSwitchListener.cardViews](#), [FactionSwitchListener.factionButton](#), [FactionSwitchListener.THEME_MONSTER](#), [FactionSwitchListener.THEME_NILFGAARD](#), [FactionSwitchListener.THEME_NORTHERN_KINGDOMS](#), [FactionSwitchListener.THEME_PREFERENCE_KEY](#), [FactionSwitchListener.THEME_SCOIATAEL](#), and [FactionSwitchListener.unitViews](#).

5.23.3.3 setTheme()

```
static void setTheme (
    @NonNull Context context ) [static]
```

Changes the theme of the content referenced by the given Context depending on the preferences obtainable by said Context.

Parameters

<i>context</i>	Context that is used to query the current theme preference and that gets it theme updated accordingly.
----------------	--

Definition at line 148 of file [FactionSwitchListener.java](#).

References [FactionSwitchListener.THEME_MONSTER](#), [FactionSwitchListener.THEME_NILFGAARD](#), [FactionSwitchListener.THEME_PREFERENCE_KEY](#), and [FactionSwitchListener.THEME_SCOIATAEL](#).

Referenced by [MainActivity.onCreate\(\)](#).

5.23.4 Member Data Documentation**5.23.4.1 ballViews**

```
final List<ImageView> ballViews [private]
```

List of ImageView objects that show a colored ball.

See also

R.drawable::ball_red
R.drawable::ball_grey
R.drawable::ball_blue
R.drawable::ball_green

Definition at line 66 of file [FactionSwitchListener.java](#).

Referenced by [FactionSwitchListener.FactionSwitchListener\(\)](#), [FactionSwitchListener.getListener\(\)](#), and [FactionSwitchListener.onSharedPreferenceChanged\(\)](#).

5.23.4.2 cardViews

```
final List<ImageView> cardViews [private]
```

List of ImageView objects that show a card back.

See also

R.drawable::card_monster_landscape_free
R.drawable::card_nilfgaard_landscape_free
R.drawable::card_northern_kingdoms_landscape_free
R.drawable::card_scoiatael_landscape_free

Definition at line 76 of file [FactionSwitchListener.java](#).

Referenced by [FactionSwitchListener.FactionSwitchListener\(\)](#), [FactionSwitchListener.getListener\(\)](#), and [FactionSwitchListener.onSharedPreferenceChanged\(\)](#).

5.23.4.3 factionButton

```
final ImageButton factionButton [private]
```

ImageButton showing the logo of the current faction.

See also

R.drawable::icon_round_monster
R.drawable::icon_round_nilfgaard
R.drawable::icon_round_northern_kingdoms
R.drawable::icon_round_scoiatael

Definition at line 96 of file [FactionSwitchListener.java](#).

Referenced by [FactionSwitchListener.FactionSwitchListener\(\)](#), and [FactionSwitchListener.onSharedPreferenceChanged\(\)](#).

5.23.4.4 THEME_MONSTER

```
final int THEME_MONSTER = 0 [static]
```

Integer constant representing the Monster theme.

Definition at line 41 of file [FactionSwitchListener.java](#).

Referenced by [FactionSwitchListener.onSharedPreferenceChanged\(\)](#), and [FactionSwitchListener.setTheme\(\)](#).

5.23.4.5 THEME_NILFGAARD

```
final int THEME_NILFGAARD = 1 [static]
```

Integer constant representing the Nilfgaard theme.

Definition at line 46 of file [FactionSwitchListener.java](#).

Referenced by [FactionSwitchListener.onSharedPreferenceChanged\(\)](#), and [FactionSwitchListener.setTheme\(\)](#).

5.23.4.6 THEME_NORTHERN_KINGDOMS

```
final int THEME_NORTHERN_KINGDOMS = 2 [static]
```

Integer constant representing the Northern Kingdoms theme.

Definition at line 51 of file [FactionSwitchListener.java](#).

Referenced by [FactionSwitchListener.onSharedPreferenceChanged\(\)](#), and [FactionSwitchListener.setTheme\(\)](#).

5.23.4.7 THEME_PREFERENCE_KEY

```
final String THEME_PREFERENCE_KEY = "theme" [static]
```

String constant defining the key of the theme SharedPreference.

Definition at line 36 of file [FactionSwitchListener.java](#).

Referenced by [FactionSwitchListener.onSharedPreferenceChanged\(\)](#), and [FactionSwitchListener.setTheme\(\)](#).

5.23.4.8 THEME_SCOIATAEL

```
final int THEME_SCOIATAEL = 3 [static]
```

Integer constant representing the Scoia'tael theme.

Definition at line 56 of file [FactionSwitchListener.java](#).

Referenced by [FactionSwitchListener.onSharedPreferenceChanged\(\)](#), and [FactionSwitchListener.setTheme\(\)](#).

5.23.4.9 unitViews

```
final List<TextView> unitViews [private]
```

List of TextView objects that show the number of units in a certain color.

See also

[R.color::color_text_monster](#)

[R.color::color_text_nilfgaard](#)

[R.color::color_text_northern_kingdoms](#)

[R.color::color_text_scoiatael](#)

Definition at line 86 of file [FactionSwitchListener.java](#).

Referenced by [FactionSwitchListener.FactionSwitchListener\(\)](#), [FactionSwitchListener.getListener\(\)](#), and [FactionSwitchListener.onSharedPreferenceChanged\(\)](#).

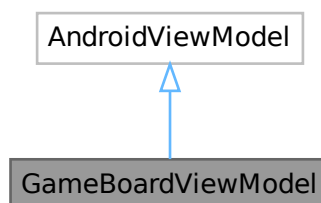
The documentation for this class was generated from the following file:

- gwent/ui/main/FactionSwitchListener.java

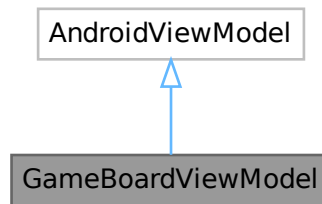
5.24 GameBoardViewModel Class Reference

An AndroidViewModel class responsible for encapsulating and offering state of views in [activity_main.xml](#), i.e., that show the overall game board.

Inheritance diagram for GameBoardViewModel:



Collaboration diagram for GameBoardViewModel:



Public Member Functions

- Flowable< [RowUiState](#) > [getRowUiState](#) (@NonNull RowType row)
Returns a Flowable object emitting RowUiState for the given row.
- Flowable< [MenuUiState](#) > [getMenuUiState](#) ()
Returns a Flowable object emitting MenuUiState.
- SoundManager [getSoundManager](#) ()
Returns the SoundManager managed by this GameBoardViewModel.
- Completable [onWeatherViewPressed](#) (@NonNull RowType row)
Updates the weather debuff of the given row.
- Completable [onHornViewPressed](#) (@NonNull RowType row)
Updates the horn buff of the given row.
- Completable [onResetButtonPressed](#) (@NonNull Context context)
Triggers a reset and possibly an alert dialog, depending on preferences.
- Completable [onFactionSwitchReset](#) (@NonNull Context context)
Triggers a reset and possibly an alert dialog, depending on preferences.
- Completable [onWeatherButtonPressed](#) ()
Clears all weather effects.
- Completable [onBurnButtonPressed](#) (@NonNull Context context)
Clears units with the most damage.

Static Public Member Functions

- static [GameBoardViewModel](#) [getModel](#) (@NonNull ViewModelStoreOwner owner, @NonNull UnitRepository repository, @NonNull SoundManager [soundManager](#))
Factory method of a GameBoardViewModel.

Private Member Functions

- [GameBoardViewModel](#) (@NonNull [GwentApplication](#) application)
Constructor of a GameBoardViewModel object.
- Single< UnitRepository > [getRepository](#) ()
Returns the UnitRepository used by the parent GwentApplication.
- Completable [reset](#) (@NonNull Context context, @NonNull ResetDialogUseCase.Trigger trigger)
Triggers a reset and possibly an alert dialog, depending on preferences.

Private Attributes

- final Map< RowType, Flowable< RowUiState > > rowUiStates = new HashMap<>(RowType.values().length)
A map structure containing the Flowable objects emitting the RowUiState for each row.
- Flowable< MenuUiState > menuUiState = Flowable.empty()
Flowable emitting the MenuUiState for the right-hand side menu.
- SoundManager soundManager
SoundManager used to play Sound effects on click events.

Static Private Attributes

- static final ViewModelInitializer< GameBoardViewModel > INITIALIZER
ViewModelInitializer used by androidx.lifecycle.ViewModelProvider.Factory to instantiate the class.

5.24.1 Detailed Description

An AndroidViewModel class responsible for encapsulating and offering state of views in [activity_main.xml](#), i.e., that show the overall game board.

Click events on the rows and the menu are handled also.

Definition at line 38 of file [GameBoardViewModel.java](#).

5.24.2 Constructor & Destructor Documentation

5.24.2.1 GameBoardViewModel()

```
GameBoardViewModel (
    @NonNull GwentApplication application ) [private]
```

Constructor of a GameBoardViewModel object.

Should only be called in #initializer.

Parameters

<i>application</i>	GwentApplication that uses this AndroidViewModel.
--------------------	---

See also

#INITIALIZER

Definition at line 60 of file [GameBoardViewModel.java](#).

5.24.3 Member Function Documentation

5.24.3.1 getMenuUiState()

```
Flowable< MenuUiState > getMenuUiState ( )
```

Returns a Flowable object emitting MenuUiState.

Returns

A Flowable object for the state of the menu.

See also

MenuUiState

#menuUiState

Definition at line 185 of file [GameBoardViewModel.java](#).

References [GameBoardViewModel.menuUiState](#).

Referenced by [MainActivity.initializeViewModel\(\)](#).

5.24.3.2 getModel()

```
static GameBoardViewModel getModel (
    @NonNull ViewModelStoreOwner owner,
    @NonNull UnitRepository repository,
    @NonNull SoundManager soundManager ) [static]
```

Factory method of a GameBoardViewModel.

Creates a new GameBoardViewModel for the given owner and initializes #rowUiStates and #menuUiState.

Parameters

<i>owner</i>	ViewModelStoreOwner instantiating the GameBoardViewModel.
<i>repository</i>	UnitRepository where Flowables are retrieved.
<i>soundManager</i>	SoundManager used to play Sound effects on click events.

Returns

A new GameBoardViewModel instance.

See also

ViewModelProvider::ViewModelProvider(ViewModelStoreOwner, ViewModelProvider.Factory)

Definition at line 98 of file [GameBoardViewModel.java](#).

References [RowUiState.getDamage\(\)](#), [RowUiState.getUnits\(\)](#), [GameBoardViewModel.INITIALIZER](#), [RowUiState.isHorn\(\)](#), [RowUiState.isWeather\(\)](#), [GameBoardViewModel.menuUiState](#), [GameBoardViewModel.reset\(\)](#), [GameBoardViewModel.rowUiStates](#), and [GameBoardViewModel.soundManager](#).

Referenced by [MainActivity.onCreate\(\)](#).

5.24.3.3 getRepository()

```
Single< UnitRepository > getRepository ( ) [private]
```

Returns the UnitRepository used by the parent GwentApplication.

Basically a wrapper for GwentApplication::getRepository(Context).

Returns

A Single emitting the UnitRepository instance.

See also

GwentApplication::getRepository(Context)

Definition at line 161 of file [GameBoardViewModel.java](#).

References [GwentApplication.getRepository\(\)](#).

Referenced by [GameBoardViewModel.onHornViewPressed\(\)](#), [GameBoardViewModel.onWeatherButtonPressed\(\)](#), and [GameBoardViewModel.onWeatherViewPressed\(\)](#).

5.24.3.4 getRowUiState()

```
Flowable< RowUiState > getRowUiState (
    @NonNull RowType row )
```

Returns a Flowable object emitting RowUiState for the given row.

Parameters

<i>row</i>	RowType defining the row for which the state is queried.
------------	--

Returns

A Flowable object for the state of the given row.

See also

RowUiState
#rowUiStates

Definition at line 173 of file [GameBoardViewModel.java](#).

References [GameBoardViewModel.rowUiStates](#).

Referenced by [MainActivity.initializeViewModel\(\)](#).

5.24.3.5 getSoundManager()

```
SoundManager getSoundManager ( )
```

Returns the SoundManager managed by this GameBoardViewModel.

Returns

A SoundManager used by this GameBoardViewModel.

Definition at line 195 of file [GameBoardViewModel.java](#).

References [GameBoardViewModel.soundManager](#).

Referenced by [MainActivity.inflateCoinFlipPopup\(\)](#), and [MainActivity.initializeViewModel\(\)](#).

5.24.3.6 onBurnButtonPressed()

```
Completable onBurnButtonPressed (
    @NonNull Context context )
```

Clears units with the most damage.

May inflate a warning dialog depending on the user's preferences and a Toast informing the user about the burned units. Should only be called by the View.OnClickListener of the burn button. Plays a matching sound, if units were removed.

Parameters

<i>context</i>	Context
----------------	---------

Returns

A Completable tracking operation status.

See also

BurnDialogUseCase::burn(Context, SoundManager)

Definition at line 310 of file [GameBoardViewModel.java](#).

References [GameBoardViewModel.soundManager](#).

Referenced by [MainActivity.initializeViewModel\(\)](#).

5.24.3.7 onFactionSwitchReset()

```
Completable onFactionSwitchReset (
    @NonNull Context context )
```

Triggers a reset and possibly an alert dialog, depending on preferences.

Should only be called when the faction has been switched. Wrapper for #reset(Context, ResetDialogUseCase.<Trigger>).

Parameters

<i>context</i>	Context object used to acquire SharedPreferences and inflate Dialog views.
----------------	--

Returns

A Completable tracking operation status.

See also

`#reset(Context, ResetDialogUseCase.Trigger)`

Definition at line 263 of file [GameBoardViewModel.java](#).

References [GameBoardViewModel.reset\(\)](#).

Referenced by [MainActivity.inflateFactionPopup\(\)](#).

5.24.3.8 onHornViewPressed()

```
Completable onHornViewPressed (
    @NonNull RowType row )
```

Updates the horn buff of the given row.

Flips between on and off. Plays a matching sound, if the horn is switched on.

Parameters

<i>row</i>	RowType defining the affected row.
------------	------------------------------------

Returns

A Completable tracking operation status.

See also

`UnitRepository::switchHorn(RowType)`

Definition at line 227 of file [GameBoardViewModel.java](#).

References [GameBoardViewModel.getRepository\(\)](#), and [GameBoardViewModel.soundManager](#).

Referenced by [MainActivity.initializeViewModel\(\)](#).

5.24.3.9 onResetButtonPressed()

```
Completable onResetButtonPressed (
    @NonNull Context context )
```

Triggers a reset and possibly an alert dialog, depending on preferences.

Should only be called by the button's View.OnClickListener. Wrapper for `#reset(Context, ResetDialogUseCase.Trigger)`.

Parameters

<i>context</i>	Context object used to acquire SharedPreferences and inflate Dialog views.
----------------	--

Returns

A Completable tracking operation status.

See also

```
#reset(Context, ResetDialogUseCase.Trigger)
ResetDialogUseCase.Trigger::BUTTON_CLICK
```

Definition at line 250 of file [GameBoardViewModel.java](#).

References [GameBoardViewModel.reset\(\)](#).

Referenced by [MainActivity.initializeViewModel\(\)](#).

5.24.3.10 onWeatherButtonPressed()

```
Completable onWeatherButtonPressed ( )
```

Clears all weather effects.

Should only be called by the View.OnClickListener of the weather button. Plays a matching sound on completion.

Returns

A Completable tracking operation status.

See also

```
UnitRepository::clearWeather()
```

Definition at line 295 of file [GameBoardViewModel.java](#).

References [GameBoardViewModel.getRepository\(\)](#).

Referenced by [MainActivity.initializeViewModel\(\)](#).

5.24.3.11 onWeatherViewPressed()

```
Completable onWeatherViewPressed (
    @NonNull RowType row )
```

Updates the weather debuff of the given row.

Flips between good and bad weather. Plays a matching sound, if the weather is switched on.

Parameters

<i>row</i>	RowType defining the affected row.
------------	------------------------------------

Returns

A Completable tracking operation status.

See also

```
UnitRepository::switchWeather(RowType)
```

Definition at line 207 of file [GameBoardViewModel.java](#).

References [GameBoardViewModel.getRepository\(\)](#), and [GameBoardViewModel.soundManager](#).

Referenced by [MainActivity.initializeViewModel\(\)](#).

5.24.3.12 reset()

```
Completable reset (
    @NonNull Context context,
    @NonNull ResetDialogUseCase.Trigger trigger ) [private]
```

Triggers a reset and possibly an alert dialog, depending on preferences.

Plays a matching sound, if units were removed.

Parameters

<i>context</i>	Context object used to acquire SharedPreferences and inflate Dialog views.
----------------	--

Parameters

<i>trigger</i>	com.peternaggschga.gwent.domain.cases.ResetDialogUseCase.Trigger defining which action triggered the reset.
----------------	---

Returns

A Completable tracking operation status.

See also

[#reset\(Context, ResetDialogUseCase.Trigger\)](#)

Definition at line 276 of file [GameBoardViewModel.java](#).

References [GameBoardViewModel.soundManager](#).

Referenced by [GameBoardViewModel.getModel\(\)](#), [GameBoardViewModel.onFactionSwitchReset\(\)](#), and [GameBoardViewModel.onResetButtonPressed\(\)](#).

5.24.4 Member Data Documentation

5.24.4.1 INITIALIZER

```
final ViewModelInitializer<GameBoardViewModel> INITIALIZER [static], [private]
```

Initial value:

```
= new ViewModelInitializer<>({
    GameBoardViewModel.class,
    creationExtras -> {
        GwentApplication app = (GwentApplication) creationExtras.get(APPLICATION_KEY);
        assert app != null;
        return new GameBoardViewModel(app);
    }
})
```

ViewModelInitializer used by [androidx.lifecycle.ViewModelProvider.Factory](#) to instantiate the class.

See also

[androidx.lifecycle.ViewModelProvider.Factory::from\(ViewModelInitializer\[\]\)](#)

Definition at line 45 of file [GameBoardViewModel.java](#).

Referenced by [GameBoardViewModel.getModel\(\)](#).

5.24.4.2 menuUiState

```
Flowable<MenuUiState> menuUiState = Flowable.empty() [private]
```

Flowable emitting the MenuUiState for the right-hand side menu.

Initialized in [#getModel\(\)](#).

See also

[#getMenuUiState\(\)](#)

[MenuUiState](#)

Definition at line 80 of file [GameBoardViewModel.java](#).

Referenced by [GameBoardViewModel.getMenuUiState\(\)](#), and [GameBoardViewModel.getModel\(\)](#).

5.24.4.3 rowUiStates

```
final Map<RowType, Flowable<RowUiState> > rowUiStates = new HashMap<>(RowType.values().length)
[private]
```

A map structure containing the Flowable objects emitting the RowUiState for each row.

Initialized in [#getModel\(\)](#).

See also

[#getRowUiState\(RowType\)](#)

[RowUiState](#)

Definition at line 71 of file [GameBoardViewModel.java](#).

Referenced by [GameBoardViewModel.getModel\(\)](#), and [GameBoardViewModel.getRowUiState\(\)](#).

5.24.4.4 soundManager

`SoundManager soundManager [private]`

SoundManager used to play Sound effects on click events.

Definition at line 85 of file [GameBoardViewModel.java](#).

Referenced by [GameBoardViewModel.getModel\(\)](#), [GameBoardViewModel.getSoundManager\(\)](#), [GameBoardViewModel.onBurnButtonClicked\(\)](#), [GameBoardViewModel.onHornViewPressed\(\)](#), [GameBoardViewModel.onWeatherViewPressed\(\)](#), and [GameBoardViewModel.reset\(\)](#).

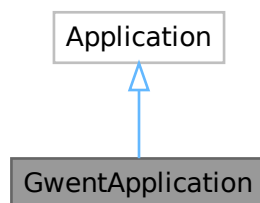
The documentation for this class was generated from the following file:

- `gwent/ui/main/GameBoardViewModel.java`

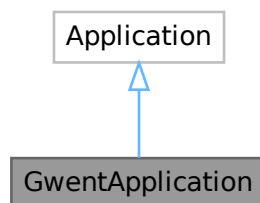
5.25 GwentApplication Class Reference

An `Application` encapsulating the Gwent app.

Inheritance diagram for GwentApplication:



Collaboration diagram for GwentApplication:



Public Member Functions

- void [onCreate](#) ()
Called when the application is starting, before any activity, service, or receiver objects (excluding content providers) have been created.
- `Single< UnitRepository >` [getRepository](#) ()
Returns the `repository` used to create, read, update and delete game state.

Static Public Member Functions

- static `Single< UnitRepository >` [getRepository](#) (@NonNull Context context)

Returns a `UnitRepository` used by the `GwentApplication` referenced by the given `Context`.

Private Attributes

- AppDatabase [database](#)
AppDatabase used by this app.

Static Private Attributes

- static UnitRepository [repository](#) = null
UnitRepository used for communication to database.

5.25.1 Detailed Description

An `Application` encapsulating the Gwent app.

This class is responsible for maintaining a Singleton `UnitRepository` connected to the AppDatabase of this app.

Definition at line 18 of file [GwentApplication.java](#).

5.25.2 Member Function Documentation

5.25.2.1 `getRepository()` [1/2]

```
Single< UnitRepository > getRepository ( )
```

Returns the `repository` used to create, read, update and delete game state.

Lazily initializes `repository`, if not yet done so.

See also

[#getRepository\(Context\)](#)

Returns

A `UnitRepository` used to access game state.

Definition at line 62 of file [GwentApplication.java](#).

References [GwentApplication.database](#), and [GwentApplication.repository](#).

5.25.2.2 `getRepository()` [2/2]

```
static Single< UnitRepository > getRepository (
    @NonNull Context context ) [static]
```

Returns a `UnitRepository` used by the `GwentApplication` referenced by the given `Context`.

Wrapper of `getRepository()`.

See also

[#getRepository\(\)](#)

Parameters

<i>context</i>	Context of a <code>GwentApplication</code> .
----------------	--

Returns

A `UnitRepository` used by the given `GwentApplication` to access game state.

Definition at line 41 of file [GwentApplication.java](#).

Referenced by [CardNumberPickerAdapter.addSelectedUnits\(\)](#), [BurnDialogUseCase.burn\(\)](#), [ShowUnitsDialog.getDialog\(\)](#), [GameBoardViewModel.getRepository\(\)](#), [MainActivity.initializeViewModel\(\)](#), [MainActivity.onCreate\(\)](#), [AddCardDialog.onCreate\(\)](#), and [ResetDialogUseCase.reset\(\)](#).

5.25.2.3 onCreate()

```
void onCreate ( )
```

Called when the application is starting, before any activity, service, or receiver objects (excluding content providers) have been created.

Initializes database.

Definition at line 51 of file [GwentApplication.java](#).

References [GwentApplication.database](#).

5.25.3 Member Data Documentation

5.25.3.1 database

```
AppDatabase database [private]
```

AppDatabase used by this app.

Is initialized in `onCreate()`.

See also

[#onCreate\(\)](#)

Definition at line 32 of file [GwentApplication.java](#).

Referenced by [GwentApplication.getRepository\(\)](#), and [GwentApplication.onCreate\(\)](#).

5.25.3.2 repository

```
UnitRepository repository = null [static], [private]
```

UnitRepository used for communication to database.

Is lazily initialized in `getRepository()` and provided as a Singleton.

See also

[#getRepository\(\)](#)

[#getRepository\(Context\)](#)

Definition at line 25 of file [GwentApplication.java](#).

Referenced by [GwentApplication.getRepository\(\)](#).

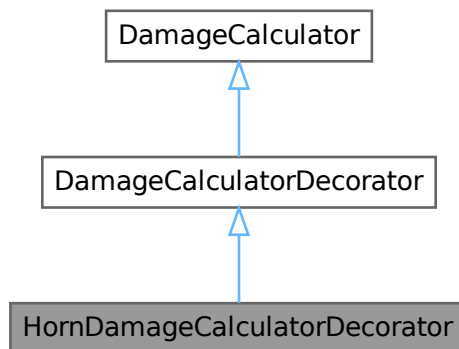
The documentation for this class was generated from the following file:

- `gwent/GwentApplication.java`

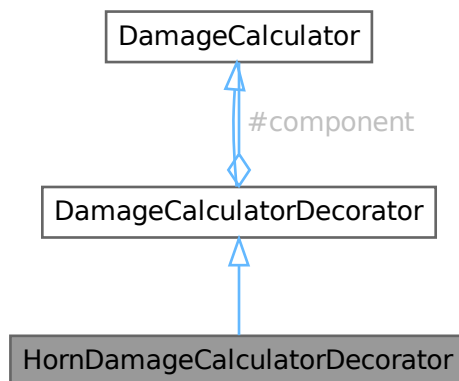
5.26 HornDamageCalculatorDecorator Class Reference

A DamageCalculator class responsible for calculating the horn buff if necessary, i.e., if there are units with the [com.peternaggschga.gwent.data.Ability::HORN](#) ability that are not the unit the damage is calculated for or if there is a commander's horn in this row.

Inheritance diagram for HornDamageCalculatorDecorator:



Collaboration diagram for HornDamageCalculatorDecorator:



Public Member Functions

- `int calculateDamage (int id, @IntRange(from=0) int damage)`
Calculates the (de-)buffed damage of unit with the given id and the given base-damage.
- `Color isBuffed (int id)`
Calculates whether the unit with the given id is shown as Color::BUFFED, Color::DEBUFFED, or Color::DEFAULT.

Package Functions

- `HornDamageCalculatorDecorator (@NonNull DamageCalculator component, @NonNull List< Integer > unitIds)`
Constructor of a HornDamageCalculatorDecorator.

Package Functions inherited from [DamageCalculatorDecorator](#)

- [DamageCalculatorDecorator](#) (@NonNull [DamageCalculator](#) component)

Constructor of a [DamageCalculatorDecorator](#).

Private Member Functions

- boolean [doubleDamage](#) (int id)

Decides whether the unit with the given id is affected by the horn buff.

Private Attributes

- final List< Integer > [unitIds](#)

A List of Integers containing the ids of all units with the [com.peternaggschga.gwent.data.Ability::HORN](#) ability.

Additional Inherited Members

Protected Attributes inherited from [DamageCalculatorDecorator](#)

- final [DamageCalculator](#) component

A [DamageCalculator](#) which is decorated by this decorator.

5.26.1 Detailed Description

A [DamageCalculator](#) class responsible for calculating the horn buff if necessary, i.e., if there are units with the [com.peternaggschga.gwent.data.Ability::HORN](#) ability that are not the unit the damage is calculated for or if there is a commander's horn in this row.

Acts as a [ConcreteDecorator](#) in the implemented decorator pattern and should decorate a [WeatherDamageCalculator](#), a [BondDamageCalculatorDecorator](#) or a [MoralDamageCalculatorDecorator](#).

Definition at line 18 of file [HornDamageCalculatorDecorator.java](#).

5.26.2 Constructor & Destructor Documentation

5.26.2.1 [HornDamageCalculatorDecorator\(\)](#)

```
HornDamageCalculatorDecorator (
    @NonNull DamageCalculator component,
    @NonNull List< Integer > unitIds ) [package]
```

Constructor of a [HornDamageCalculatorDecorator](#).

Should only be called by [DamageCalculatorBuilder](#). The given damage calculator should be of type [WeatherDamageCalculator](#), [BondDamageCalculatorDecorator](#) or [MoralDamageCalculatorDecorator](#) for correct damage calculation. If the respective row has a commander's horn, `null` must be an element of the given List.

Parameters

<i>component</i>	DamageCalculator that is being decorated by this decorator.
<i>unitIds</i>	List of Integers representing ids of units with the com.peternaggschga.gwent.data.Ability::HORN ability.

See also

[DamageCalculatorBuilder](#)

Definition at line 38 of file [HornDamageCalculatorDecorator.java](#).

References [DamageCalculatorDecorator.component](#), and [HornDamageCalculatorDecorator.unitIds](#).

5.26.3 Member Function Documentation

5.26.3.1 calculateDamage()

```
int calculateDamage (
    int id,
    @IntRange(from=0) int damage )
```

Calculates the (de-)buffed damage of unit with the given id and the given base-damage.
Returns given damage times two if the unit is buffed by a commander's horn.

See also

[#doubleDamage\(int\)](#)

Parameters

<i>id</i>	Integer representing the <code>UnitEntity::id</code> of the unit whose (de-)buff damage is calculated.
<i>damage</i>	Integer representing the base-damage of the unit whose (de-)buff damage is calculated.

Returns

Integer representing the (de-)buffed damage of the unit.

Exceptions

<i>IllegalArgumentException</i>	When damage is negative.
---	--------------------------

Implements [DamageCalculator](#).

Definition at line 63 of file [HornDamageCalculatorDecorator.java](#).

References [DamageCalculator.calculateDamage\(\)](#), [DamageCalculatorDecorator.component](#), and [HornDamageCalculatorDecorator.d](#).

5.26.3.2 doubleDamage()

```
boolean doubleDamage (
    int id ) [private]
```

Decides whether the unit with the given id is affected by the horn buff.

Parameters

<i>id</i>	Integer representing the <code>UnitEntity::id</code> of the unit that is being evaluated.
-----------	---

Returns

A Boolean value defining whether the horn buff affects the given unit.

Definition at line 49 of file [HornDamageCalculatorDecorator.java](#).

References [HornDamageCalculatorDecorator.unitIds](#).

Referenced by [HornDamageCalculatorDecorator.calculateDamage\(\)](#), and [HornDamageCalculatorDecorator.isBuffed\(\)](#).

5.26.3.3 isBuffed()

```
Color isBuffed (
```

```
int id )
```

Calculates whether the unit with the given id is shown as Color::BUFFED, Color::DEBUFFED, or Color::DEFAULT. Units are shown as Color::BUFFED when they are affected by a horn buff, otherwise their Color is defined by #component.

Parameters

<i>id</i>	Integer representing the UnitEntity::id of the unit buff status is calculated.
-----------	--

Returns

Color representing whether the unit is buffed, de-buffed or not affected.

See also

Color

#doubleDamage(int)

Implements [DamageCalculator](#).

Definition at line 82 of file [HornDamageCalculatorDecorator.java](#).

References [DamageCalculatorDecorator.component](#), [HornDamageCalculatorDecorator.doubleDamage\(\)](#), and [DamageCalculator.isBuffed\(\)](#).

5.26.4 Member Data Documentation

5.26.4.1 unitIds

```
final List<Integer> unitIds [private]
```

A List of Integers containing the ids of all units with the [com.paternaggschga.gwent.data.Ability::HORN](#) ability.

If a commander's horn is in the respective row, #unitIds contains `null`.

Definition at line 24 of file [HornDamageCalculatorDecorator.java](#).

Referenced by [HornDamageCalculatorDecorator.HornDamageCalculatorDecorator\(\)](#), and [HornDamageCalculatorDecorator.doubleD](#)

The documentation for this class was generated from the following file:

- gwent/domain/damage/HornDamageCalculatorDecorator.java

5.27 ImageViewSwitchAnimator Class Reference

A class providing functionality for changing the resource shown by an ImageView using a fading animation.

Collaboration diagram for ImageViewSwitchAnimator:

ImageViewSwitchAnimator

Static Public Member Functions

- static Completable [animatedSwitch](#) (@NonNull ImageView view, @DrawableRes int resId)
Changes the resource of the given ImageView in an animation to the given resource id.

Static Private Member Functions

- static ImageView [getOverlayView](#) (@NonNull ImageView template)
Returns a copy of the given template which is one layer above it.

5.27.1 Detailed Description

A class providing functionality for changing the resource shown by an `ImageView` using a fading animation.

See also

`#animatedSwitch(ImageView, int)`

Definition at line 20 of file [ImageViewSwitchAnimator.java](#).

5.27.2 Member Function Documentation

5.27.2.1 animatedSwitch()

```
static Completable animatedSwitch (  
    @NonNull ImageView view,  
    @DrawableRes int resId ) [static]
```

Changes the resource of the given `ImageView` in an animation to the given resource id.

Animation is realized by creating a new `ImageView` on top of the given `ImageView` which is then faded out using the `fade_out` animation provided by Android. The operation is complete when the animation ends.

See also

`android.R.anim::fade_out`

Parameters

<i>view</i>	<code>ImageView</code> , whose resource is to be changed.
<i>resId</i>	Integer representing the drawable resource that the given view should show.

Returns

A `Completable` tracking operation status.

Definition at line 53 of file [ImageViewSwitchAnimator.java](#).

References [ImageViewSwitchAnimator.getOverlayView\(\)](#).

Referenced by [MenuUiStateObserver.accept\(\)](#), [RowUiStateObserver.accept\(\)](#), and [FactionSwitchListener.onSharedPreferenceChange\(\)](#).

5.27.2.2 getOverlayView()

```
static ImageView getOverlayView (  
    @NonNull ImageView template ) [static], [private]
```

Returns a copy of the given template which is one layer above it.

Parameters

<i>template</i>	<code>ImageView</code> that is copied.
-----------------	--

Returns

An `ImageView` similar to `template` but one layer in front of it.

Definition at line 27 of file `ImageViewSwitchAnimator.java`.

Referenced by `ImageViewSwitchAnimator.animatedSwitch()`.

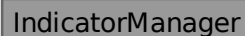
The documentation for this class was generated from the following file:

- `gwent/ui/main/ImageViewSwitchAnimator.java`

5.28 IndicatorManager Class Reference

A class managing the indicator views shown in the bottom bar of the `IntroductionActivity`.

Collaboration diagram for `IndicatorManager`:



```

classDiagram
    class IndicatorManager
  
```

Package Functions

- `IndicatorManager` (`@NonNull Window window`)
Constructor of an `IndicatorManager` for the given `Window`.
- void `updateIndicators` (`@IntRange(from=0, to=4) int position`)
Updates the indicator views according to the given new position.

Private Attributes

- final `ImageView[]` `indicators` = `new ImageView[5]`
Array of `ImageViews` that contains the indicator views.
- int `currentView` = 0
Integer referencing the position, i.e.

5.28.1 Detailed Description

A class managing the indicator views shown in the bottom bar of the `IntroductionActivity`.

The managed views show the user how many pages there are and which one they are on.

Definition at line 15 of file `IndicatorManager.java`.

5.28.2 Constructor & Destructor Documentation

5.28.2.1 IndicatorManager()

```
IndicatorManager (
    @NonNull Window window ) [package]
```

Constructor of an `IndicatorManager` for the given `Window`.

Initializes values in `indicators`.

Parameters

<i>window</i>	<code>Window</code> containing the indicator views.
---------------	---

See also

[Window::findViewById\(int\)](#)
[R.id::introduction_indicator_0](#)
[R.id::introduction_indicator_1](#)
[R.id::introduction_indicator_2](#)
[R.id::introduction_indicator_3](#)
[R.id::introduction_indicator_4](#)

Definition at line 43 of file [IndicatorManager.java](#).

References [IndicatorManager.indicators](#).

5.28.3 Member Function Documentation

5.28.3.1 updateIndicators()

```
void updateIndicators (
    @IntRange(from=0, to=4) int position ) [package]
```

Updates the indicator views according to the given new position.

Sets the `currentView` to `R.drawable#indicator_unselected` and the view at the new position to `R.drawable#indicator_selected`. Saves the given position in `currentView`.

Parameters

<i>position</i>	Integer referencing the page represented by the managed indicators.
-----------------	---

Exceptions

<i>ArrayIndexOutOfBoundsException</i>	When position is not in [0, 4].
---------------------------------------	---------------------------------

Definition at line 59 of file [IndicatorManager.java](#).

References [IndicatorManager.currentView](#), and [IndicatorManager.indicators](#).

5.28.4 Member Data Documentation

5.28.4.1 currentView

```
int currentView = 0 [private]
```

Integer referencing the position, i.e.

the index in `indicators`, of the view that is currently shown as selected. Is set in `updateIndicators(int)`.

See also

[#updateIndicators\(int\)](#)

Definition at line 30 of file [IndicatorManager.java](#).

Referenced by [IndicatorManager.updateIndicators\(\)](#).

5.28.4.2 indicators

```
final ImageView [] indicators = new ImageView[5] [private]
```

Array of `ImageViews` that contains the indicator views.

Definition at line 20 of file [IndicatorManager.java](#).

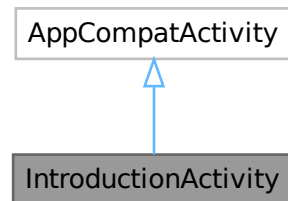
Referenced by [IndicatorManager.IndicatorManager\(\)](#), and [IndicatorManager.updateIndicators\(\)](#).

The documentation for this class was generated from the following file:

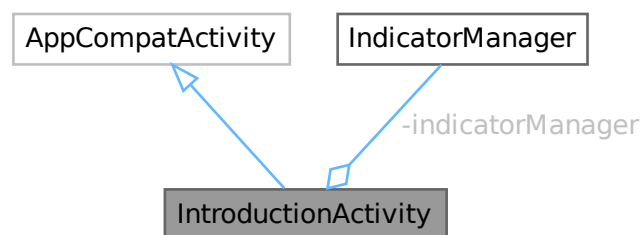
- `gwent/ui/introduction/IndicatorManager.java`

5.29 IntroductionActivity Class Reference

An `AppCompatActivity` that gives the user an introduction into the usage of the application.
Inheritance diagram for `IntroductionActivity`:



Collaboration diagram for `IntroductionActivity`:



Protected Member Functions

- void `onCreate` (Bundle savedInstanceState)
Sets layout to `R.layout#activity_introduction`, initializes `indicatorManager`, sets listeners on the buttons, and creates a new `OnBackPressedCallback` that switches to the previous page, if it is not the first one.

Private Attributes

- `IndicatorManager indicatorManager`
`IndicatorManager` used to update the progress indicators according to the currently shown page.

5.29.1 Detailed Description

An `AppCompatActivity` that gives the user an introduction into the usage of the application.
Is called when the app is first started (as tracked by the `androidx.preference.Preference` at key `R.string#preference_first_use_key`.
Definition at line 20 of file `IntroductionActivity.java`.

5.29.2 Member Function Documentation

5.29.2.1 onCreate()

```
void onCreate (
    Bundle savedInstanceState ) [protected]
```

Sets layout to `R.layout#activity_introduction`, initializes `indicatorManager`, sets listeners on the buttons, and creates a new `OnBackPressedCallback` that switches to the previous page, if it is not the first one.

Parameters

<i>savedInstanceState</i>	If the activity is being re-initialized after previously being shut down then this <code>Bundle</code> contains the data it most recently supplied in <code>onSaveInstanceState</code> . Note: Otherwise it is null.
---------------------------	---

Called when enabled and the user clicks on the back-button. Switches the `ViewPager2` containing the `IntroductionFragments` to the last position.

Updates the `IntroductionActivity#indicatorManager`, changes button-visibility and whether the call-back for `onBackPressed` is active.

See also

`IndicatorManager::updateIndicators(int)`

Parameters

<i>position</i>	Position index of the new selected page.
-----------------	--

Definition at line 37 of file [IntroductionActivity.java](#).

References [IntroductionActivity.indicatorManager](#).

5.29.3 Member Data Documentation

5.29.3.1 indicatorManager

```
IndicatorManager indicatorManager [private]
```

`IndicatorManager` used to update the progress indicators according to the currently shown page.

Is initialized in `onCreate(Bundle)`.

Definition at line 25 of file [IntroductionActivity.java](#).

Referenced by [IntroductionActivity.onCreate\(\)](#).

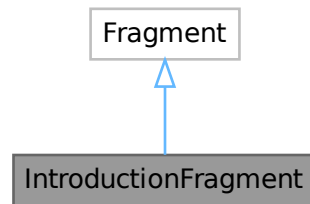
The documentation for this class was generated from the following file:

- `gwent/ui/introduction/IntroductionActivity.java`

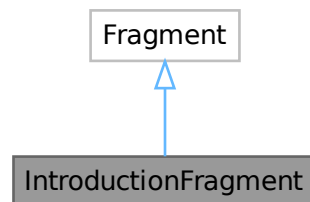
5.30 IntroductionFragment Class Reference

A `Fragment` containing an `ImageView` and a `TextView` conveying introductory information.

Inheritance diagram for IntroductionFragment:



Collaboration diagram for IntroductionFragment:



Public Member Functions

- View [onCreateView](#) (@NonNull LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState)
Inflates the `R.layout#fragment_introduction` layout and sets text and image.

Static Public Member Functions

- static [IntroductionFragment newInstance](#) (@IntRange(from=0, to=[PAGES_COUNT](#) - 1) int index)
Creates a new `IntroductionFragment` showing the page defined by the given index.

Static Public Attributes

- static final int [PAGES_COUNT](#) = 5
Integer constant defining the number of possible pages represented by this fragment.

Static Private Attributes

- static final String [ARG_SECTION_NUMBER](#) = "section_number"
String constant defining the argument key used to communicate the requested page to the fragment.
- static final int[] [imageIds](#)
Integer array constant containing the resource ids for all introductory images.
- static final int[] [stringIds](#)
Integer array constant containing the resource ids for all introductory texts.

5.30.1 Detailed Description

A `Fragment` containing an `ImageView` and a `TextView` conveying introductory information.
Definition at line 21 of file [IntroductionFragment.java](#).

5.30.2 Member Function Documentation

5.30.2.1 newInstance()

```
static IntroductionFragment newInstance (
    @IntRange(from=0, to=PAGES_COUNT - 1) int index ) [static]
```

Creates a new `IntroductionFragment` showing the page defined by the given index.
Factory method of `IntroductionFragment`.

Parameters

<i>index</i>	Integer defining the page that the created fragment shows.
--------------	--

Returns

An `IntroductionFragment` showing the page with the given index.

Exceptions

<i>IllegalArgumentException</i>	When the given index is not in [0, PAGES_COUNT - 1].
---------------------------------	--

Definition at line 50 of file [IntroductionFragment.java](#).

References [IntroductionFragment.ARG_SECTION_NUMBER](#), and [IntroductionFragment.PAGES_COUNT](#).

Referenced by [SectionsPagerAdapter.createFragment\(\)](#).

5.30.2.2 onCreateView()

```
View onCreateView (
    @NonNull LayoutInflater inflater,
    ViewGroup container,
    Bundle savedInstanceState )
```

Inflates the `R.layout#fragment_introduction` layout and sets text and image.

Parameters

<i>inflater</i>	The <code>LayoutInflater</code> object that can be used to inflate any views in the fragment,
<i>container</i>	If non-null, this is the parent view that the fragment's UI should be attached to. The fragment should not add the view itself, but this can be used to generate the <code>LayoutParams</code> of the view.
<i>savedInstanceState</i>	If non-null, this fragment is being re-constructed from a previous saved state as given here.

Returns

A `View` that is the root of the newly inflated layout.

Definition at line 75 of file [IntroductionFragment.java](#).

References [IntroductionFragment.ARG_SECTION_NUMBER](#), [IntroductionFragment.imageIds](#), and [IntroductionFragment.stringIds](#).

5.30.3 Member Data Documentation

5.30.3.1 ARG_SECTION_NUMBER

```
final String ARG_SECTION_NUMBER = "section_number" [static], [private]
```

`String` constant defining the argument key used to communicate the requested page to the fragment.
 Definition at line 30 of file [IntroductionFragment.java](#).
 Referenced by [IntroductionFragment.newInstance\(\)](#), and [IntroductionFragment.onCreateView\(\)](#).

5.30.3.2 imageIds

```
final int [] imageIds [static], [private]
```

Initial value:

```
= {R.drawable.introduction_1, R.drawable.introduction_2,  
   R.drawable.introduction_3, R.drawable.introduction_4, R.drawable.introduction_5}
```

`Integer` array constant containing the resource ids for all introductory images.

Definition at line 34 of file [IntroductionFragment.java](#).

Referenced by [IntroductionFragment.onCreateView\(\)](#).

5.30.3.3 PAGES_COUNT

```
final int PAGES_COUNT = 5 [static]
```

`Integer` constant defining the number of possible pages represented by this fragment.

Is equivalent to the length of `stringIds` and `imageIds`.

Definition at line 26 of file [IntroductionFragment.java](#).

Referenced by [SectionsPagerAdapter.getItemCount\(\)](#), and [IntroductionFragment.newInstance\(\)](#).

5.30.3.4 stringIds

```
final int [] stringIds [static], [private]
```

Initial value:

```
= {R.string.introduction_page1, R.string.introduction_page2,  
   R.string.introduction_page3, R.string.introduction_page4, R.string.introduction_page5}
```

`Integer` array constant containing the resource ids for all introductory texts.

Definition at line 39 of file [IntroductionFragment.java](#).

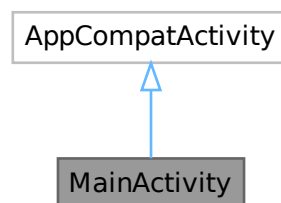
Referenced by [IntroductionFragment.onCreateView\(\)](#).

The documentation for this class was generated from the following file:

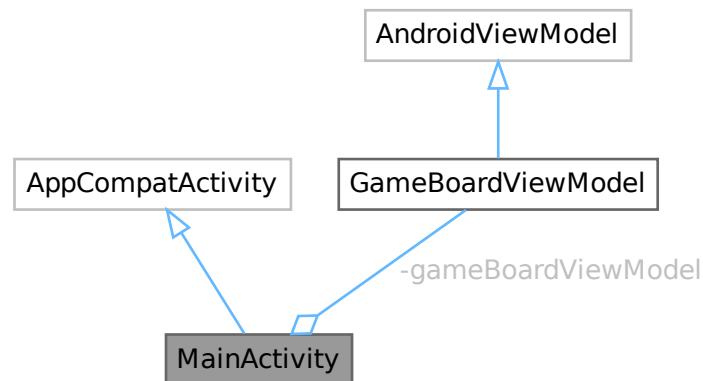
- `gwent/ui/introduction/IntroductionFragment.java`

5.31 MainActivity Class Reference

An `AppCompatActivity` that is called on startup and that encapsulates the main view onto the game board.
 Inheritance diagram for `MainActivity`:



Collaboration diagram for MainActivity:



Public Member Functions

- void [onWindowFocusChanged](#) (boolean hasFocus)
Called when the focus of this activities `android.view.Window` changes.

Protected Member Functions

- void [onCreate](#) (@Nullable Bundle savedInstanceState)
Sets the theme and layout, initializes `gameBoardViewModel` and `factionSwitchListener` and sets listeners for some menu buttons.
- void [onResume](#) ()
Called when the application is resumed after a pause or on startup.
- void [onDestroy](#) ()
Called when the activity is destroyed.

Private Member Functions

- void [initializeViewModel](#) ()
Initializes the Views in this activity to communicate with the `gameBoardViewModel`.
- void [inflateFactionPopup](#) ()
Shows a new `ChangeFactionDialog` enabling the user to choose the preferred theme.
- void [inflateCoinFlipPopup](#) ()
Shows a new `CoinFlipDialog` and plays a coin-flip sound using `SoundManager#playCoinSound()`.

Private Attributes

- final CompositeDisposable [disposables](#) = new CompositeDisposable()
CompositeDisposable used to store all `io.reactivex.rxjava3.disposables.Disposables`, this activity might create to allow for their disposal in `onDestroy()`.
- SharedPreferences.OnSharedPreferenceChangeListener [factionSwitchListener](#)
SharedPreferences.OnSharedPreferenceChangeListener that is called when faction-layout, i.e., the `androidx.preference.Preference` at the key defined by `com.paternaggschga.gwent.ui.main.FactionSwitch` is changed.
- [GameBoardViewModel gameBoardViewModel](#)
GameBoardViewModel holding the ui state of this activity.

5.31.1 Detailed Description

An `AppCompatActivity` that is called on startup and that encapsulates the main view onto the game board.
Definition at line 46 of file [MainActivity.java](#).

5.31.2 Member Function Documentation

5.31.2.1 inflateCoinFlipPopup()

```
void inflateCoinFlipPopup ( ) [private]
```

Shows a new `CoinFlipDialog` and plays a coin-flip sound using `SoundManager#playCoinSound()`.
Definition at line 274 of file [MainActivity.java](#).

References [MainActivity.gameBoardViewModel](#), and [GameBoardViewModel.getSoundManager\(\)](#).

Referenced by [MainActivity.onCreate\(\)](#).

5.31.2.2 inflateFactionPopup()

```
void inflateFactionPopup ( ) [private]
```

Shows a new `ChangeFactionDialog` enabling the user to choose the preferred theme.

The decision is saved in the preference at the key `FactionSwitchListener#THEME_PREFERENCE_KEY` and if the user opted to reset on faction switch (i.e., the preference at `R.string#preference_key_faction_reset` is true), `GameBoardViewModel#onFactionSwitchReset(Context)` is called.

Definition at line 250 of file [MainActivity.java](#).

References [MainActivity.disposables](#), [MainActivity.gameBoardViewModel](#), and [GameBoardViewModel.onFactionSwitchReset\(\)](#).

Referenced by [MainActivity.onCreate\(\)](#).

5.31.2.3 initializeViewModel()

```
void initializeViewModel ( ) [private]
```

Initializes the Views in this activity to communicate with the `gameBoardViewModel`.

Sets `android.widget.Button.OnClickListener` for the weather-, horn-, and card-views of each row as well as listeners for the reset-, weather- and burn-buttons.

Definition at line 191 of file [MainActivity.java](#).

References [MainActivity.disposables](#), [MainActivity.gameBoardViewModel](#), [GameBoardViewModel.getMenuUiState\(\)](#), [RowUiStateObserver.getObserver\(\)](#), [GwentApplication.getRepository\(\)](#), [GameBoardViewModel.getRowUiState\(\)](#), [GameBoardViewModel.getSoundManager\(\)](#), [GameBoardViewModel.onBurnButtonPressed\(\)](#), [GameBoardViewModel.onHornViewPressed\(\)](#), [GameBoardViewModel.onResetButtonPressed\(\)](#), [GameBoardViewModel.onWeatherButtonPressed\(\)](#), and [GameBoardViewModel.onWeatherViewPressed\(\)](#).

Referenced by [MainActivity.onCreate\(\)](#).

5.31.2.4 onCreate()

```
void onCreate (
    @Nullable Bundle savedInstanceState ) [protected]
```

Sets the theme and layout, initializes `gameBoardViewModel` and `factionSwitchListener` and sets listeners for some menu buttons.

If the application is started for the first time (as tracked by the preference at key `R.string#preference_first_use_key`) the `IntroductionActivity` is called first. The theme is set according to the preference at the key specified by `FactionSwitchListener#THEME_PREFERENCE_KEY`. The layout is set to `R.layout#activity_main`. `android.widget.Button.OnClickListener` are set for the buttons referenced by `R.id#factionButton`, `R.id#coinButton`, and `R.id#settingsButton`.

Parameters

<i>savedInstanceState</i>	If the activity is being re-initialized after previously being shut down then this Bundle contains the data it most recently supplied in <code>onSaveInstanceState</code> . Note: Otherwise it is null.
---------------------------	--

Definition at line 83 of file [MainActivity.java](#).

References [MainActivity.disposables](#), [MainActivity.factionSwitchListener](#), [MainActivity.gameBoardViewModel](#),

[FactionSwitchListener.getListener\(\)](#), [GameBoardViewModel.getModel\(\)](#), [GwentApplication.getRepository\(\)](#), [MainActivity.inflateCoinFlipPopup\(\)](#), [MainActivity.inflateFactionPopup\(\)](#), [MainActivity.initializeViewModel\(\)](#), and [FactionSwitchListener.setTheme\(\)](#).

5.31.2.5 onDestroy()

```
void onDestroy ( ) [protected]
```

Called when the activity is destroyed.

Disposes and clears all `io.reactivex.rxjava3.disposables.Disposables` in `disposables`.

Definition at line 180 of file [MainActivity.java](#).

References [MainActivity.disposables](#).

5.31.2.6 onResume()

```
void onResume ( ) [protected]
```

Called when the application is resumed after a pause or on startup.

Sets the background image according to the preference at the key referenced by `R.string#preference_key_design`.

Definition at line 124 of file [MainActivity.java](#).

5.31.2.7 onWindowFocusChanged()

```
void onWindowFocusChanged (
    boolean hasFocus )
```

Called when the focus of this activities `android.view.Window` changes.

Hides system ui and sets flags to keep the screen on when the window is in focus for more than 250 ms.

Parameters

<i>hasFocus</i>	Whether the window of this activity has focus.
-----------------	--

Definition at line 154 of file [MainActivity.java](#).

5.31.3 Member Data Documentation

5.31.3.1 disposables

```
final CompositeDisposable disposables = new CompositeDisposable() [private]
```

`CompositeDisposable` used to store all `io.reactivex.rxjava3.disposables.Disposables`, this activity might create to allow for their disposal in `onDestroy()`.

Definition at line 52 of file [MainActivity.java](#).

Referenced by [MainActivity.inflateFactionPopup\(\)](#), [MainActivity.initializeViewModel\(\)](#), [MainActivity.onCreate\(\)](#), and [MainActivity.onDestroy\(\)](#).

5.31.3.2 factionSwitchListener

```
SharedPreferences.OnSharedPreferenceChangeListener factionSwitchListener [private]
```

`SharedPreferences.OnSharedPreferenceChangeListener` that is called when faction-layout, i.e., the `androidx.preference.Preference` at the key defined by `com.peternaggschga.gwent.ui.main.FactionSwitchListener` is changed.

Reference must be kept (even if not used) to avoid garbage collection of the registered listener (see [here](#) for more information).

Definition at line 61 of file [MainActivity.java](#).

Referenced by [MainActivity.onCreate\(\)](#).

5.31.3.3 gameBoardViewModel

```
GameBoardViewModel gameBoardViewModel [private]
```

`GameBoardViewModel` holding the ui state of this activity.

Definition at line 66 of file [MainActivity.java](#).

Referenced by [MainActivity.inflateCoinFlipPopup\(\)](#), [MainActivity.inflateFactionPopup\(\)](#), [MainActivity.initializeViewModel\(\)](#), and [MainActivity.onCreate\(\)](#).

The documentation for this class was generated from the following file:

- [gwent/ui/main/MainActivity.java](#)

5.32 MenuUiState Class Reference

A data class encapsulating the visible state of the menu in the main view of the application.
Collaboration diagram for MenuUiState:



```
classDiagram
    class MenuUiState
```

Public Member Functions

- [MenuUiState](#) ([@IntRange](#)(from=0) [int](#) damage, [boolean](#) reset, [boolean](#) weather, [boolean](#) burn)
Constructor of a MenuUiState encapsulating the given data.
- [int](#) [getDamage](#) ()
Returns the summed-up damage of all units on the game board.
- [boolean](#) [isReset](#) ()
Returns whether the reset button is clickable.
- [boolean](#) [isWeather](#) ()
Returns whether the weather button is clickable.
- [boolean](#) [isBurn](#) ()
Returns whether the burn button is clickable.
- [boolean](#) [equals](#) ([Object](#) o)

Private Attributes

- final [int](#) [damage](#)
Defines the summed-up damage of all units on the game board.
- final [boolean](#) [reset](#)
Defines whether the reset button is clickable.
- final [boolean](#) [weather](#)
Defines whether the weather button is clickable.
- final [boolean](#) [burn](#)
Defines whether the burn button is clickable.

5.32.1 Detailed Description

A data class encapsulating the visible state of the menu in the main view of the application.
Definition at line 10 of file [MenuUiState.java](#).

5.32.2 Constructor & Destructor Documentation

5.32.2.1 MenuUiState()

```
MenuUiState (
    @IntRange(from=0) int damage,
    boolean reset,
    boolean weather,
    boolean burn )
```

Constructor of a MenuUiState encapsulating the given data.

Parameters

<i>damage</i>	Integer representing the summed-up damage of all units.
<i>reset</i>	Boolean defining whether the reset button is clickable.
<i>weather</i>	Boolean defining whether the weather button is clickable.
<i>burn</i>	Boolean defining whether the burn button is clickable.

Exceptions

<i>IllegalArgumentException</i>	When damage is negative.
---------------------------------	--------------------------

Definition at line 41 of file [MenuUiState.java](#).

References [MenuUiState.burn](#), [MenuUiState.damage](#), [MenuUiState.reset](#), and [MenuUiState.weather](#).

5.32.3 Member Function Documentation

5.32.3.1 equals()

```
boolean equals (
    Object o )
```

Definition at line 89 of file [MenuUiState.java](#).

5.32.3.2 getDamage()

```
int getDamage ( )
```

Returns the summed-up damage of all units on the game board.

Returns

An Integer representing the summed-up damage of all units.

Definition at line 57 of file [MenuUiState.java](#).

References [MenuUiState.damage](#).

5.32.3.3 isBurn()

```
boolean isBurn ( )
```

Returns whether the burn button is clickable.

Returns

A Boolean defining whether the burn button is clickable.

Definition at line 84 of file [MenuUiState.java](#).

References [MenuUiState.burn](#).

5.32.3.4 isReset()

```
boolean isReset ( )
```

Returns whether the reset button is clickable.

Returns

A Boolean defining whether the reset button is clickable.

Definition at line 66 of file [MenuUiState.java](#).

References [MenuUiState.reset](#).

5.32.3.5 isWeather()

```
boolean isWeather ( )
```

Returns whether the weather button is clickable.

Returns

A Boolean defining whether the weather button is clickable.

Definition at line 75 of file [MenuUiState.java](#).

References [MenuUiState.weather](#).

5.32.4 Member Data Documentation

5.32.4.1 burn

```
final boolean burn [private]
```

Defines whether the burn button is clickable.

Definition at line 30 of file [MenuUiState.java](#).

Referenced by [MenuUiState.MenuUiState\(\)](#), and [MenuUiState.isBurn\(\)](#).

5.32.4.2 damage

```
final int damage [private]
```

Defines the summed-up damage of all units on the game board.

Definition at line 15 of file [MenuUiState.java](#).

Referenced by [MenuUiState.MenuUiState\(\)](#), and [MenuUiState.getDamage\(\)](#).

5.32.4.3 reset

```
final boolean reset [private]
```

Defines whether the reset button is clickable.

Definition at line 20 of file [MenuUiState.java](#).

Referenced by [MenuUiState.MenuUiState\(\)](#), and [MenuUiState.isReset\(\)](#).

5.32.4.4 weather

```
final boolean weather [private]
```

Defines whether the weather button is clickable.

Definition at line 25 of file [MenuUiState.java](#).

Referenced by [MenuUiState.MenuUiState\(\)](#), and [MenuUiState.isWeather\(\)](#).

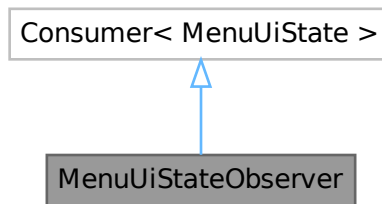
The documentation for this class was generated from the following file:

- gwent/ui/main/MenuUiState.java

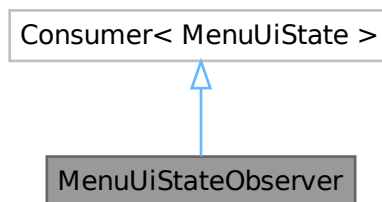
5.33 MenuUiStateObserver Class Reference

An observer class responsible for updating the menu views when notified, i.e., when a new MenuUiState is produced by the subscribed Flowable.

Inheritance diagram for MenuUiStateObserver:



Collaboration diagram for MenuUiStateObserver:



Public Member Functions

- [MenuUiStateObserver](#) (@NonNull TextView [damageView](#), @NonNull ImageButton [resetButton](#), @NonNull ImageButton [weatherButton](#), @NonNull ImageButton [burnButton](#))
Constructor of a MenuUiStateObserver updating the given views when #onChanged() is called.
- void [accept](#) (@NonNull [MenuUiState](#) menuUiState)
Changes value in #damageView as well as appearance and clickable status of #resetButton, #weatherButton, and #burnButton.

Private Attributes

- final TextView [damageView](#)
A TextView showing the user the summed-up damage of all units, i.e., MenuUiState::damage.
- final ImageButton [resetButton](#)
An ImageButton responsible for resetting the whole board, i.e., clearing weather and commander's horn and deleting all units.
- final ImageButton [weatherButton](#)
An ImageButton responsible for clearing all weather effects.
- final ImageButton [burnButton](#)
An ImageButton responsible for deleting the units with the highest damage.

5.33.1 Detailed Description

An observer class responsible for updating the menu views when notified, i.e., when a new `MenuUiState` is produced by the subscribed `Flowable`.

See also

`MenuUiState`

Definition at line 18 of file [MenuUiStateObserver.java](#).

5.33.2 Constructor & Destructor Documentation

5.33.2.1 MenuUiStateObserver()

```
MenuUiStateObserver (
    @NonNull TextView damageView,
    @NonNull ImageButton resetButton,
    @NonNull ImageButton weatherButton,
    @NonNull ImageButton burnButton )
```

Constructor of a `MenuUiStateObserver` updating the given views when `#onChanged()` is called.

Parameters

<i>damageView</i>	TextView showing the summed-up damage of all units.
<i>resetButton</i>	ImageButton used to reset the whole game board.
<i>weatherButton</i>	ImageButton used to reset the weather debuff in all rows.
<i>burnButton</i>	ImageButton used to remove the strongest units from the game board.

Definition at line 55 of file [MenuUiStateObserver.java](#).

References [MenuUiStateObserver.burnButton](#), [MenuUiStateObserver.damageView](#), [MenuUiStateObserver.resetButton](#), and [MenuUiStateObserver.weatherButton](#).

5.33.3 Member Function Documentation

5.33.3.1 accept()

```
void accept (
    @NonNull MenuUiState menuUiState )
```

Changes value in `#damageView` as well as appearance and clickable status of `#resetButton`, `#weatherButton`, and `#burnButton`.

Appearance changes are animated using `ImageViewSwitchAnimator`.

Parameters

<i>menuUiState</i>	MenuUiState representing the updated state of the menu.
--------------------	---

See also

`ImageViewSwitchAnimator`

Definition at line 72 of file [MenuUiStateObserver.java](#).

References [ImageViewSwitchAnimator.animatedSwitch\(\)](#), [MenuUiStateObserver.burnButton](#), [MenuUiStateObserver.damageView](#), [MenuUiStateObserver.resetButton](#), and [MenuUiStateObserver.weatherButton](#).

5.33.4 Member Data Documentation

5.33.4.1 burnButton

```
final ImageButton burnButton [private]
```

An ImageButton responsible for deleting the units with the highest damage.

Is not clickable and gray when MenuUiState::burn is `false`.

Definition at line 45 of file [MenuUiStateObserver.java](#).

Referenced by [MenuUiStateObserver.MenuUiStateObserver\(\)](#), and [MenuUiStateObserver.accept\(\)](#).

5.33.4.2 damageView

```
final TextView damageView [private]
```

A TextView showing the user the summed-up damage of all units, i.e., MenuUiState::damage.

Definition at line 23 of file [MenuUiStateObserver.java](#).

Referenced by [MenuUiStateObserver.MenuUiStateObserver\(\)](#), and [MenuUiStateObserver.accept\(\)](#).

5.33.4.3 resetButton

```
final ImageButton resetButton [private]
```

An ImageButton responsible for resetting the whole board, i.e., clearing weather and commander's horn and deleting all units.

Is not clickable and gray when MenuUiState::reset is `false`.

Definition at line 31 of file [MenuUiStateObserver.java](#).

Referenced by [MenuUiStateObserver.MenuUiStateObserver\(\)](#), and [MenuUiStateObserver.accept\(\)](#).

5.33.4.4 weatherButton

```
final ImageButton weatherButton [private]
```

An ImageButton responsible for clearing all weather effects.

Is not clickable and gray when MenuUiState::weather is `false`.

Definition at line 38 of file [MenuUiStateObserver.java](#).

Referenced by [MenuUiStateObserver.MenuUiStateObserver\(\)](#), and [MenuUiStateObserver.accept\(\)](#).

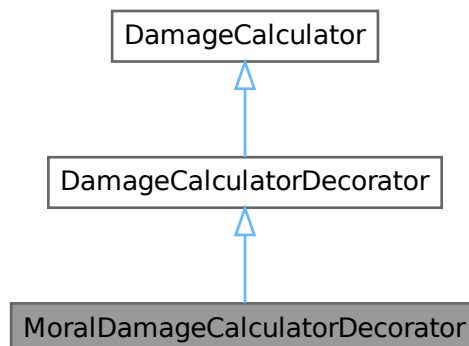
The documentation for this class was generated from the following file:

- gwent/ui/main/MenuUiStateObserver.java

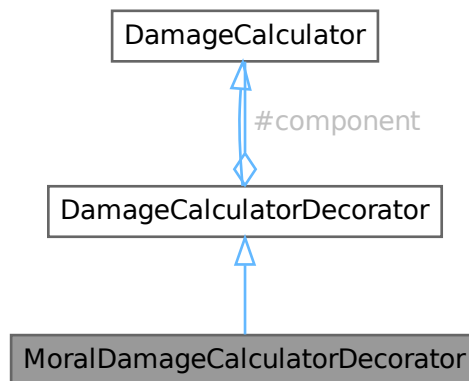
5.34 MoralDamageCalculatorDecorator Class Reference

A DamageCalculator class responsible for calculating the moral buff if necessary, i.e., if there are units with the [com.peternaggschga.gwent.data.Ability::MORAL_BOOST](#) ability that are not the unit the damage is calculated for.

Inheritance diagram for MoralDamageCalculatorDecorator:



Collaboration diagram for MoralDamageCalculatorDecorator:



Public Member Functions

- `int calculateDamage (int id, @IntRange(from=0) int damage)`
Calculates the (de-)buffed damage of unit with the given id and the given base-damage.
- `Color isBuffed (int id)`
Calculates whether the unit with the given id is shown as `Color::BUFFED`, `Color::DEBUFFED`, or `Color::DEFAULT`.

Package Functions

- `MoralDamageCalculatorDecorator (@NonNull DamageCalculator component, @NonNull List< Integer > unitIds)`
Constructor of a MoralDamageCalculatorDecorator.

Package Functions inherited from DamageCalculatorDecorator

- `DamageCalculatorDecorator (@NonNull DamageCalculator component)`
Constructor of a DamageCalculatorDecorator.

Private Attributes

- `final List< Integer > unitIds`
A List of Integers containing the ids of all units with the `com.peternaggschga.gwent.data.Ability::MORAL_BOOST` ability.

Additional Inherited Members

Protected Attributes inherited from DamageCalculatorDecorator

- `final DamageCalculator component`
A DamageCalculator which is decorated by this decorator.

5.34.1 Detailed Description

A DamageCalculator class responsible for calculating the moral buff if necessary, i.e., if there are units with the `com.peternaggschga.gwent.data.Ability::MORAL_BOOST` ability that are not the unit the damage is calculated for.

Acts as a ConcreteDecorator in the implemented decorator pattern and should decorate a WeatherDamage↔ Calculator or a BondDamageCalculatorDecorator for correct damage calculation.
Definition at line 17 of file [MoralDamageCalculatorDecorator.java](#).

5.34.2 Constructor & Destructor Documentation

5.34.2.1 MoralDamageCalculatorDecorator()

```
MoralDamageCalculatorDecorator (
    @NonNull DamageCalculator component,
    @NonNull List< Integer > unitIds ) [package]
```

Constructor of a MoralDamageCalculatorDecorator.

Should only be called by DamageCalculatorBuilder. The given damage calculator should be of type Weather↔ DamageCalculator or BondDamageCalculatorDecorator for correct damage calculation. The given List must not contain null.

Parameters

<i>component</i>	DamageCalculator that is being decorated by this decorator.
<i>unitIds</i>	List of Integers representing ids of units with the com.paternaggschga.gwent.data.Ability::MORAL_BOOST ability.

Exceptions

<i>IllegalArgumentException</i>	When unitIds contains null values.
---------------------------------	------------------------------------

See also

DamageCalculatorBuilder

Definition at line 36 of file [MoralDamageCalculatorDecorator.java](#).

References [DamageCalculatorDecorator.component](#), and [MoralDamageCalculatorDecorator.unitIds](#).

5.34.3 Member Function Documentation

5.34.3.1 calculateDamage()

```
int calculateDamage (
    int id,
    @IntRange(from=0) int damage )
```

Calculates the (de-)buffed damage of unit with the given id and the given base-damage.

Returns the given damage plus the number of moral boosts. If the unit itself has the [com.paternaggschga.gwent.data.Ability::MORAL_](#) ability, it is boosted one time less.

Parameters

<i>id</i>	Integer representing the UnitEntity::id of the unit whose (de-)buff damage is calculated.
<i>damage</i>	Integer representing the base-damage of the unit whose (de-)buff damage is calculated.

Returns

Integer representing the (de-)buffed damage of the unit.

Exceptions

<i>IllegalArgumentException</i>	When damage is negative.
---------------------------------	--------------------------

Implements [DamageCalculator](#).

Definition at line 56 of file [MoralDamageCalculatorDecorator.java](#).

References [DamageCalculator.calculateDamage\(\)](#), [DamageCalculatorDecorator.component](#), and [MoralDamageCalculatorDecorator.i](#).

5.34.3.2 isBuffed()

```
Color isBuffed (
    int id )
```

Calculates whether the unit with the given id is shown as Color::BUFFED, Color::DEBUFFED, or Color::DEFAULT. Units are shown as Color::BUFFED when they are affected by a moral boost buff, otherwise their Color is defined by #component.

Parameters

<i>id</i>	Integer representing the UnitEntity::id of the unit buff status is calculated.
-----------	--

Returns

Color representing whether the unit is buffed, de-buffed or not affected.

See also

Color

Implements [DamageCalculator](#).

Definition at line 75 of file [MoralDamageCalculatorDecorator.java](#).

References [DamageCalculatorDecorator.component](#), [DamageCalculator.isBuffed\(\)](#), and [MoralDamageCalculatorDecorator.unitIds](#).

5.34.4 Member Data Documentation

5.34.4.1 unitIds

```
final List<Integer> unitIds [private]
```

A List of Integers containing the ids of all units with the [com.peternaggschga.gwent.data.Ability::MORAL_BOOST](#) ability.

Definition at line 22 of file [MoralDamageCalculatorDecorator.java](#).

Referenced by [MoralDamageCalculatorDecorator.MoralDamageCalculatorDecorator\(\)](#), [MoralDamageCalculatorDecorator.calculateD](#) and [MoralDamageCalculatorDecorator.isBuffed\(\)](#).

The documentation for this class was generated from the following file:

- [gwent/domain/damage/MoralDamageCalculatorDecorator.java](#)

5.35 OnValueChangeListener< T extends Comparable Interface Template Reference

An interface defining a callback for changes of the current value.

Collaboration diagram for OnValueChangeListener< T extends Comparable:

```
OnValueChangeListener
< T extends Comparable
```


5.35.1 Detailed Description

An interface defining a callback for changes of the current value.

Definition at line 211 of file [ValuePicker.java](#).

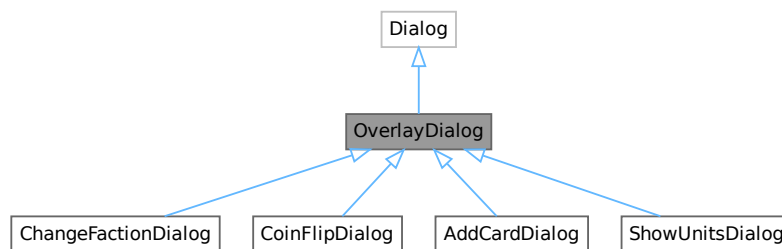
The documentation for this interface was generated from the following file:

- gwent/ui/dialogs/addcard/ValuePicker.java

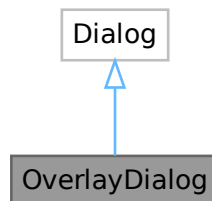
5.36 OverlayDialog Class Reference

A Dialog class which is used for popups that are shown on top of the calling Activity.

Inheritance diagram for OverlayDialog:



Collaboration diagram for OverlayDialog:



Protected Member Functions

- [OverlayDialog](#) (@NonNull Context context, @LayoutRes int [layout](#), @IdRes int [dismissViewId](#))
Constructor of an OverlayDialog in the given Context, with the given layout and dismissViewId.
- [OverlayDialog](#) (@NonNull Context context, @LayoutRes int [layout](#))
Constructor of an OverlayDialog in the given Context and with the given layout.
- void [onCreate](#) (Bundle savedInstanceState)
Initializes #layout using #setContentView(int).

Static Package Attributes

- static final int [NO_DISMISS_VIEW](#) = -1
Integer used as #dismissViewId, when the Dialog shouldn't be cancelable by clicking on a View.

Private Attributes

- final int [layout](#)
Integer referencing the layout shown by the Dialog.
- final int [dismissViewId](#)
Integer referencing a view that can be clicked to dismiss the Dialog.

Static Private Attributes

- static final ColorDrawable [BACKGROUND](#) = new ColorDrawable(Color.TRANSPARENT)
ColorDrawable shown as the background of the Dialog (above the calling Activity).

5.36.1 Detailed Description

A Dialog class which is used for popups that are shown on top of the calling Activity.

The Dialog uses the layout specified in `#layout` with the `#BACKGROUND` color as the background. The view specified by `#dismissViewId` may be clicked to dismiss the dialog.

Definition at line 22 of file [OverlayDialog.java](#).

5.36.2 Constructor & Destructor Documentation

5.36.2.1 OverlayDialog() [1/2]

```
OverlayDialog (
    @NonNull Context context,
    @LayoutRes int layout,
    @IdRes int dismissViewId ) [protected]
```

Constructor of an OverlayDialog in the given Context, with the given layout and dismissViewId.

When dismissViewId is equal to `#NO_CANCEL_VIEW`, the Dialog is not cancelable by clicking on a certain view.

Parameters

<i>context</i>	Context of the created OverlayDialog.
<i>layout</i>	Integer referencing the layout shown by the created OverlayDialog.
<i>dismiss↔ViewId</i>	Integer referencing the cancel view or <code>#NO_CANCEL_VIEW</code> .

Definition at line 54 of file [OverlayDialog.java](#).

References [OverlayDialog.dismissViewId](#), and [OverlayDialog.layout](#).

5.36.2.2 OverlayDialog() [2/2]

```
OverlayDialog (
    @NonNull Context context,
    @LayoutRes int layout ) [protected]
```

Constructor of an OverlayDialog in the given Context and with the given layout.

Wrapper of `#OverlayDialog(Context, int, int)`.

See also

`#OverlayDialog(Context, int, int)`

Parameters

<i>context</i>	Context of the created OverlayDialog.
<i>layout</i>	Integer referencing the layout shown by the created OverlayDialog.

Definition at line 68 of file [OverlayDialog.java](#).

References [OverlayDialog.layout](#), and [OverlayDialog.NO_DISMISS_VIEW](#).

5.36.3 Member Function Documentation

5.36.3.1 onCreate()

```
void onCreate (
    Bundle savedInstanceState ) [protected]
```

Initializes #layout using #setContentView(int).

Sets View.OnClickListener canceling the dialog for the View referenced by #dismissViewId if it is set. Switches whether the Dialog is cancelable using #setCancelable().

Parameters

<i>savedInstanceState</i>	If this dialog is being reinitialized after the hosting activity was previously shut down, holds the result from the most recent call to <code>onSaveInstanceState</code> , or null if this is the first time.
---------------------------	--

See also

[#setContentView\(int\)](#)

[#setCancelable\(boolean\)](#)

Reimplemented in [AddCardDialog](#), [ShowUnitsDialog](#), [ChangeFactionDialog](#), and [CoinFlipDialog](#).

Definition at line 85 of file [OverlayDialog.java](#).

References [OverlayDialog.BACKGROUND](#), [OverlayDialog.dismissViewId](#), [OverlayDialog.layout](#), and [OverlayDialog.NO_DISMISS_VIEW](#).

5.36.4 Member Data Documentation

5.36.4.1 BACKGROUND

```
final ColorDrawable BACKGROUND = new ColorDrawable(Color.TRANSPARENT) [static], [private]
```

ColorDrawable shown as the background of the Dialog (above the calling Activity).

Definition at line 33 of file [OverlayDialog.java](#).

Referenced by [OverlayDialog.onCreate\(\)](#).

5.36.4.2 dismissViewId

```
final int dismissViewId [private]
```

Integer referencing a view that can be clicked to dismiss the Dialog.

Should be equal to #NO_CANCEL_VIEW, when the Dialog is not cancelable.

See also

[#NO_DISMISS_VIEW](#)

Definition at line 45 of file [OverlayDialog.java](#).

Referenced by [OverlayDialog.OverlayDialog\(\)](#), and [OverlayDialog.onCreate\(\)](#).

5.36.4.3 layout

```
final int layout [private]
```

Integer referencing the layout shown by the Dialog.

Definition at line 38 of file [OverlayDialog.java](#).

Referenced by [OverlayDialog.OverlayDialog\(\)](#), [OverlayDialog.OverlayDialog\(\)](#), and [OverlayDialog.onCreate\(\)](#).

5.36.4.4 NO_DISMISS_VIEW

```
final int NO_DISMISS_VIEW = -1 [static], [package]
```

Integer used as #dismissViewId, when the Dialog shouldn't be cancelable by clicking on a View.

See also

`#dismissViewId`

Definition at line 28 of file [OverlayDialog.java](#).

Referenced by [OverlayDialog.OverlayDialog\(\)](#), and [OverlayDialog.onCreate\(\)](#).

The documentation for this class was generated from the following file:

- `gwent/ui/dialogs/OverlayDialog.java`

5.37 RemoveUnitsUseCase Class Reference

A use case class responsible for removing units from a `UnitRepository`.

Collaboration diagram for `RemoveUnitsUseCase`:



```

classDiagram
    class RemoveUnitsUseCase
  
```

Static Public Member Functions

- static Completable [remove](#) (@NonNull Context context, @NonNull UnitRepository repository, @NonNull Collection< UnitEntity > units, @NonNull SoundManager soundManager)
Removes the given UnitEntity objects from the given UnitRepository.
- static Completable [remove](#) (@NonNull Context context, @NonNull UnitRepository repository, int id, @NonNull SoundManager soundManager)
Removes the unit with the given id from the given UnitRepository.

Static Private Member Functions

- static Dialog [getRevengeDialog](#) (@NonNull Context context, @NonNull UnitRepository repository, @NonNull CompletableEmitter emitter, @NonNull Collection< UnitEntity > units, @IntRange(from=1) int revengeUnits, @NonNull SoundManager soundManager)
Creates a Dialog asking whether the Ability::REVENGE ability should be activated.

5.37.1 Detailed Description

A use case class responsible for removing units from a `UnitRepository`.

Capable of invoking a Dialog if a `UnitEntity` with the `Ability::REVENGE` ability is removed. Should not be used directly by the UI layer.

See also

`BurnDialogUseCase`

Definition at line 27 of file [RemoveUnitsUseCase.java](#).

5.37.2 Member Function Documentation

5.37.2.1 getRevengeDialog()

```
static Dialog getRevengeDialog (
    @NonNull Context context,
    @NonNull UnitRepository repository,
```

```

@NonNull CompletableEmitter emitter,
@NonNull Collection< UnitEntity > units,
@IntRange(from=1) int revengeUnits,
@NonNull SoundManager soundManager ) [static], [private]

```

Creates a Dialog asking whether the Ability::REVENGE ability should be activated.
The Dialog is created using an RevengeAlertDialogBuilderAdapter.

Parameters

<i>context</i>	Context of the shown Dialog.
<i>repository</i>	UnitRepository where the UnitEntity objects are removed and avengers are inserted.
<i>emitter</i>	CompletableEmitter where CompletableEmitter::onComplete must be called, when the user makes a decision.
<i>units</i>	Collection of UnitEntity objects that are removed.
<i>revengeUnits</i>	Long representing the number of revenge units.
<i>soundManager</i>	SoundManager used when an Avenger is added.

Returns

A Dialog asking whether the Ability::REVENGE ability should be activated.

See also

RevengeAlertDialogBuilderAdapter::insertAvengers(UnitRepository, int, SoundManager)

Definition at line 89 of file [RemoveUnitsUseCase.java](#).

References [RevengeAlertDialogBuilderAdapter.create\(\)](#), [RevengeAlertDialogBuilderAdapter.insertAvengers\(\)](#), and [RevengeAlertDialogBuilderAdapter.setPositiveCallback\(\)](#).

Referenced by [RemoveUnitsUseCase.remove\(\)](#).

5.37.2.2 remove() [1/2]

```

static Completable remove (
    @NonNull Context context,
    @NonNull UnitRepository repository,
    @NonNull Collection< UnitEntity > units,
    @NonNull SoundManager soundManager ) [static]

```

Removes the given UnitEntity objects from the given UnitRepository.

If a UnitEntity has the Ability::REVENGE ability, a Dialog asking whether the ability should be used is shown.

Parameters

<i>context</i>	Context of the shown Dialog.
<i>repository</i>	UnitRepository where the UnitEntity objects are removed.
<i>units</i>	Collection of UnitEntity objects that are removed.
<i>soundManager</i>	SoundManager used, if an UnitEntity has the Ability::REVENGE ability.

Returns

A Completable tracking operation status.

Exceptions

<i>NullPointerException</i>	When units contains a null value.
-----------------------------	-----------------------------------

See also

`#getRevengeDialog(Context, UnitRepository, CompletableEmitter, Collection, int, SoundManager)`
`UnitRepository::delete(Collection)`

Definition at line 43 of file [RemoveUnitsUseCase.java](#).

References [RemoveUnitsUseCase.getRevengeDialog\(\)](#).

5.37.2.3 remove() [2/2]

```
static Completable remove (
    @NonNull Context context,
    @NonNull UnitRepository repository,
    int id,
    @NonNull SoundManager soundManager ) [static]
```

Removes the unit with the given id from the given UnitRepository.

If the unit has the Ability::REVENGE ability, a Dialog asking whether the ability should be used is shown. Wrapper of `#remove(Context, UnitRepository, Collection)`.

Parameters

<i>context</i>	Context of the shown Dialog.
<i>repository</i>	UnitRepository where the UnitEntity objects are removed.
<i>id</i>	Integer

Returns

A Completable tracking operation status.

See also

`#remove(Context, UnitRepository, Collection, SoundManager)`

Definition at line 68 of file [RemoveUnitsUseCase.java](#).

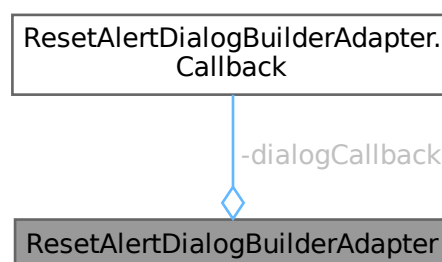
The documentation for this class was generated from the following file:

- `gwent/domain/cases/RemoveUnitsUseCase.java`

5.38 ResetAlertDialogBuilderAdapter Class Reference

An adapter class adapting AlertDialog.Builder to provide an interface for creating an AlertDialog asking the user whether he really wants to reset.

Collaboration diagram for ResetAlertDialogBuilderAdapter:



Classes

- interface [Callback](#)

An interface defining functions to propagate the user's decision back to the creator.

Package Functions

- [ResetAlertDialogBuilderAdapter](#) (@NonNull Context context, @NonNull [Callback](#) dialogCallback)
Constructor of a ResetAlertDialogBuilderAdapter.
- [AlertDialog](#) [create](#) ()
Creates an AlertDialog with the arguments supplied to this builder.
- [ResetAlertDialogBuilderAdapter](#) [setTrigger](#) (@NonNull ResetDialogUseCase.Trigger trigger)
Changes the shown message of the AlertDialog and whether it is cancelable depending on the trigger of the dialog.
- [ResetAlertDialogBuilderAdapter](#) [setMonsterDialog](#) (boolean monsterDialog)
Changes the positive button callback depending on whether it is a monsterDialog or not.

Private Attributes

- final [AlertDialog.Builder](#) [adapteeBuilder](#)
AlertDialog.Builder that is adapted by this class.
- final [Callback](#) [dialogCallback](#)
Callback used to propagate the user's answer to the built AlertDialog back to its creator.

5.38.1 Detailed Description

An adapter class adapting [AlertDialog.Builder](#) to provide an interface for creating an [AlertDialog](#) asking the user whether he really wants to reset.

Definition at line 16 of file [ResetAlertDialogBuilderAdapter.java](#).

5.38.2 Constructor & Destructor Documentation

5.38.2.1 ResetAlertDialogBuilderAdapter()

```
ResetAlertDialogBuilderAdapter (  
    @NonNull Context context,  
    @NonNull Callback dialogCallback ) [package]
```

Constructor of a [ResetAlertDialogBuilderAdapter](#).

The given [Callback](#) is called when the user responds to the built [AlertDialog](#).

Parameters

<i>context</i>	Context of the built AlertDialog .
<i>dialogCallback</i>	Callback used to propagate the user's answer to the creator.

Definition at line 36 of file [ResetAlertDialogBuilderAdapter.java](#).

References [ResetAlertDialogBuilderAdapter.dialogCallback](#), and [ResetAlertDialogBuilderAdapter.Callback.reset\(\)](#).

5.38.3 Member Function Documentation

5.38.3.1 create()

```
AlertDialog create ( ) [package]
```

Creates an [AlertDialog](#) with the arguments supplied to this builder.

Basically just calls [AlertDialog.Builder::create\(\)](#) on [#adapteeBuilder](#).

See also

`AlertDialog.Builder::create()`

Definition at line 52 of file [ResetAlertDialogBuilderAdapter.java](#).

References [ResetAlertDialogBuilderAdapter.adapteeBuilder](#).

Referenced by [ResetDialogUseCase.reset\(\)](#).

5.38.3.2 setMonsterDialog()

```
ResetAlertDialogBuilderAdapter setMonsterDialog (
    boolean monsterDialog ) [package]
```

Changes the positive button callback depending on whether it is a monsterDialog or not.

When it is a monsterDialog, a checkbox is shown defining whether the perk of monster faction should be activated.

Parameters

<i>monsterDialog</i>	Boolean defining whether a monsterDialog should be shown.
----------------------	---

Returns

The `ResetAlertDialogBuilderAdapter` with the changed trigger.

Definition at line 81 of file [ResetAlertDialogBuilderAdapter.java](#).

References [ResetAlertDialogBuilderAdapter.adapteeBuilder](#), [ResetAlertDialogBuilderAdapter.dialogCallback](#), and [ResetAlertDialogBuilderAdapter.Callback.reset\(\)](#).

Referenced by [ResetDialogUseCase.reset\(\)](#).

5.38.3.3 setTrigger()

```
ResetAlertDialogBuilderAdapter setTrigger (
    @NonNull ResetDialogUseCase.Trigger trigger ) [package]
```

Changes the shown message of the `AlertDialog` and whether it is cancelable depending on the trigger of the dialog.

Parameters

<i>trigger</i>	<code>ResetDialogUseCase.Trigger</code> defining what triggered the reset.
----------------	--

Returns

The `ResetAlertDialogBuilderAdapter` with the changed trigger.

Definition at line 64 of file [ResetAlertDialogBuilderAdapter.java](#).

References [ResetAlertDialogBuilderAdapter.adapteeBuilder](#), and [ResetDialogUseCase.Trigger.FACTION_SWITCH](#).

5.38.4 Member Data Documentation

5.38.4.1 adapteeBuilder

```
final AlertDialog.Builder adapteeBuilder [private]
```

`AlertDialog.Builder` that is adapted by this class.

Definition at line 21 of file [ResetAlertDialogBuilderAdapter.java](#).

Referenced by [ResetAlertDialogBuilderAdapter.create\(\)](#), [ResetAlertDialogBuilderAdapter.setMonsterDialog\(\)](#), and [ResetAlertDialogBuilderAdapter.setTrigger\(\)](#).

5.38.4.2 dialogCallback

```
final Callback dialogCallback [private]
```

Callback used to propagate the user's answer to the built `AlertDialog` back to its creator.

Definition at line 27 of file [ResetAlertDialogBuilderAdapter.java](#).

Referenced by [ResetAlertDialogBuilderAdapter.ResetAlertDialogBuilderAdapter\(\)](#), and [ResetAlertDialogBuilderAdapter.setMonsterDialogType\(\)](#).
 The documentation for this class was generated from the following file:

- gwent/domain/cases/ResetAlertDialogBuilderAdapter.java

5.39 ResetDialogUseCase Class Reference

A use case class responsible for dispatching a reset call to `ResetRepositoryUseCase`, possibly after a confirmation by the user obtained from a `Dialog`.

Collaboration diagram for `ResetDialogUseCase`:

ResetDialogUseCase

Classes

- enum [DialogType](#)
An Enum defining which form of Dialog should be shown.
- enum [Trigger](#)
An Enum listing the possible triggers of a reset.

Static Public Member Functions

- static `Single< Boolean > reset` (@NonNull Context context, @NonNull [Trigger](#) trigger, @NonNull SoundManager soundManager)
Resets the given UnitRepository.

Static Protected Member Functions

- static `Single< Boolean > reset` (@NonNull Context context, @NonNull UnitRepository repository, @NonNull [Trigger](#) trigger, @NonNull SoundManager soundManager)
Resets the given UnitRepository.

Static Private Member Functions

- static `Single< DialogType > getDialogType` (@NonNull Context context, @NonNull UnitRepository repository, @NonNull [Trigger](#) trigger)
Returns a DialogType defining which kind of Dialog should be invoked.

5.39.1 Detailed Description

A use case class responsible for dispatching a reset call to `ResetRepositoryUseCase`, possibly after a confirmation by the user obtained from a `Dialog`.

See also

`ResetRepositoryUseCase`

Definition at line 31 of file [ResetDialogUseCase.java](#).

5.39.2 Member Function Documentation

5.39.2.1 `getDialogType()`

```
static Single< DialogType > getDialogType (
    @NonNull Context context,
    @NonNull UnitRepository repository,
    @NonNull Trigger trigger ) [static], [private]
```

Returns a DialogType defining which kind of Dialog should be invoked.

Parameters

<i>context</i>	Context used for retrieval of SharedPreferences.
<i>repository</i>	UnitRepository used to check if a certain DialogType is even necessary.
<i>trigger</i>	Trigger defining what triggered the reset.

Returns

A DialogType defining the kind of Dialog.

See also

DialogType

Definition at line 102 of file [ResetDialogUseCase.java](#).

References [ResetDialogUseCase.DialogType.DEFAULT](#), and [ResetDialogUseCase.DialogType.NONE](#).

Referenced by [ResetDialogUseCase.reset\(\)](#).

5.39.2.2 `reset()` [1/2]

```
static Single< Boolean > reset (
    @NonNull Context context,
    @NonNull Trigger trigger,
    @NonNull SoundManager soundManager ) [static]
```

Resets the given UnitRepository.

May invoke a Dialog asking whether the user really wants to reset depending on the given trigger and warning settings. ResetRepositoryUseCase is used for resetting. Wrapper for `#reset(Context, UnitRepository, Trigger)`.

Parameters

<i>context</i>	Context where a Dialog can be inflated.
<i>trigger</i>	Trigger defining what triggered this reset.
<i>soundManager</i>	SoundManager used, if an UnitEntity has the Ability::REVENGE ability.

Returns

A Single emitting a Boolean defining whether the reset really took place.

See also

`#reset(Context, UnitRepository, Trigger, SoundManager)`

`ResetRepositoryUseCase::reset(Context, UnitRepository, boolean, SoundManager)`

Definition at line 47 of file [ResetDialogUseCase.java](#).

References [GwentApplication.getRepository\(\)](#), and [ResetDialogUseCase.reset\(\)](#).

Referenced by [ResetDialogUseCase.reset\(\)](#).

5.39.2.3 reset() [2/2]

```
static Single< Boolean > reset (
    @NonNull Context context,
    @NonNull UnitRepository repository,
    @NonNull Trigger trigger,
    @NonNull SoundManager soundManager ) [static], [protected]
```

Resets the given UnitRepository.

May invoke a Dialog asking whether the user really wants to reset depending on the given trigger and warning settings. ResetRepositoryUseCase is used for resetting.

Parameters

<i>context</i>	Context where a Dialog can be inflated.
<i>repository</i>	UnitRepository that is reset.
<i>trigger</i>	Trigger defining what triggered this reset.
<i>soundManager</i>	SoundManager used, if an UnitEntity has the Ability::REVENGE ability.

Returns

A Single emitting a Boolean defining whether the reset really took place.

See also

ResetRepositoryUseCase::reset(Context, UnitRepository, boolean, SoundManager)

Definition at line 66 of file [ResetDialogUseCase.java](#).

References [ResetAlertDialogBuilderAdapter.create\(\)](#), [ResetDialogUseCase.getDialogType\(\)](#), [ResetDialogUseCase.DialogType.MONSTER_DIALOG](#), [ResetDialogUseCase.DialogType.NONE](#), [ResetRepositoryUseCase.reset\(\)](#), and [ResetAlertDialogBuilderAdapter.setMonsterDialog\(\)](#).

The documentation for this class was generated from the following file:

- gwent/domain/cases/ResetDialogUseCase.java

5.40 ResetRepositoryUseCase Class Reference

A use case class responsible for resetting the UnitRepository.

Collaboration diagram for ResetRepositoryUseCase:

ResetRepositoryUseCase

Static Public Member Functions

- static Completable [reset](#) (@NonNull Context context, @NonNull UnitRepository repository, @NonNull SoundManager soundManager)
Resets the given UnitRepository.

Static Package Functions

- static Maybe< UnitEntity > [reset](#) (@NonNull Context context, @NonNull UnitRepository repository, boolean keepUnit, @NonNull SoundManager soundManager)
Resets the given UnitRepository and keeps one random unit if keepUnit is true.

Static Private Member Functions

- static Optional< UnitEntity > [getRandomUnit](#) (@NonNull List< UnitEntity > units)
Selects a random unit that is not epic.
- static Dialog [getRevengeDialog](#) (@NonNull Context context, @NonNull UnitRepository repository, @NonNull CompletableEmitter emitter, @Nullable UnitEntity keptUnit, @IntRange(from=1) int revengeUnits, @NonNull SoundManager soundManager)
Creates a Dialog asking whether the Ability::REVENGE ability should be activated.

5.40.1 Detailed Description

A use case class responsible for resetting the UnitRepository.

Capable of invoking a Dialog if a UnitEntity with the Ability::REVENGE ability is removed. Should not be used directly by the UI layer.

See also

[ResetDialogUseCase](#)

Definition at line 31 of file [ResetRepositoryUseCase.java](#).

5.40.2 Member Function Documentation

5.40.2.1 getRandomUnit()

```
static Optional< UnitEntity > getRandomUnit (
    @NonNull List< UnitEntity > units ) [static], [private]
```

Selects a random unit that is not epic.

If all units are epic or if there are no units at all, an empty Optional is returned.

Parameters

<i>units</i>	List of UnitEntity objects, one of which is selected.
--------------	---

Returns

An Optional containing the selected unit or nothing if no unit could be selected.

Definition at line 71 of file [ResetRepositoryUseCase.java](#).

Referenced by [ResetRepositoryUseCase.reset\(\)](#).

5.40.2.2 getRevengeDialog()

```
static Dialog getRevengeDialog (
    @NonNull Context context,
    @NonNull UnitRepository repository,
    @NonNull CompletableEmitter emitter,
    @Nullable UnitEntity keptUnit,
    @IntRange(from=1) int revengeUnits,
    @NonNull SoundManager soundManager ) [static], [private]
```

Creates a Dialog asking whether the Ability::REVENGE ability should be activated.

The Dialog is created using an RevengeAlertDialogBuilderAdapter.

Parameters

<i>context</i>	Context of the shown Dialog.
<i>repository</i>	UnitRepository where the UnitEntity objects are removed and avengers are inserted.
<i>emitter</i>	CompletableEmitter where CompletableEmitter::onComplete must be called, when the user makes a decision.
<i>keptUnit</i>	UnitEntity that should be kept.

Parameters

<i>revengeUnits</i>	Long representing the number of revenge units.
<i>soundManager</i>	SoundManager used, if an Avenger is added.

Returns

A Dialog asking whether the Ability::REVENGE ability should be activated.

See also

RevengeAlertDialogBuilderAdapter::insertAvengers(UnitRepository, int, SoundManager)

Definition at line 93 of file [ResetRepositoryUseCase.java](#).

References [RevengeAlertDialogBuilderAdapter.create\(\)](#), [RevengeAlertDialogBuilderAdapter.insertAvengers\(\)](#), and [RevengeAlertDialogBuilderAdapter.setPositiveCallback\(\)](#).

Referenced by [ResetRepositoryUseCase.reset\(\)](#).

5.40.2.3 reset() [1/2]

```
static Completable reset (
    @NonNull Context context,
    @NonNull UnitRepository repository,
    @NonNull SoundManager soundManager ) [static]
```

Resets the given UnitRepository.

If a removed UnitEntity has the Ability::REVENGE ability, a Dialog asking whether the ability should be used is shown. Wrapper of #reset(Context, UnitRepository, boolean, SoundManager).

Parameters

<i>context</i>	Context of the shown Dialog.
<i>repository</i>	UnitRepository that is being reset.
<i>soundManager</i>	SoundManager used, if an UnitEntity has the Ability::REVENGE ability.

Returns

A Completable tracking operation status.

See also

#reset(Context, UnitRepository, boolean, SoundManager)

Definition at line 123 of file [ResetRepositoryUseCase.java](#).

References [ResetRepositoryUseCase.reset\(\)](#).

5.40.2.4 reset() [2/2]

```
static Maybe< UnitEntity > reset (
    @NonNull Context context,
    @NonNull UnitRepository repository,
    boolean keepUnit,
    @NonNull SoundManager soundManager ) [static], [package]
```

Resets the given UnitRepository and keeps one random unit if keepUnit is true.

If a removed UnitEntity has the Ability::REVENGE ability, a Dialog asking whether the ability should be used is shown.

Parameters

<i>context</i>	Context of the shown Dialog.
----------------	------------------------------

Parameters

<i>repository</i>	UnitRepository that is being reset.
<i>keepUnit</i>	Boolean defining whether a single UnitEntity should be kept.
<i>soundManager</i>	SoundManager used, if an UnitEntity has the Ability::REVENGE ability.

Returns

A Maybe emitting the kept UnitEntity or nothing if keepUnit is false.

See also

`#getRevengeDialog(Context, UnitRepository, CompletableEmitter, UnitEntity, int, SoundManager)`

`UnitRepository::reset(UnitEntity)`

Definition at line 46 of file [ResetRepositoryUseCase.java](#).

References [ResetRepositoryUseCase.getRandomUnit\(\)](#), and [ResetRepositoryUseCase.getRevengeDialog\(\)](#).

Referenced by [ResetRepositoryUseCase.reset\(\)](#), and [ResetDialogUseCase.reset\(\)](#).

The documentation for this class was generated from the following file:

- gwent/domain/cases/ResetRepositoryUseCase.java

5.41 RevengeAlertDialogBuilderAdapter Class Reference

An adapter class adapting AlertDialog.Builder to provide an interface for creating an AlertDialog asking the user whether they want to invoke the Ability::REVENGE ability.

Collaboration diagram for RevengeAlertDialogBuilderAdapter:

RevengeAlertDialogBuilder
Adapter

Static Public Member Functions

- static Completable [insertAvengers](#) (@NonNull UnitRepository repository, @IntRange(from=0) int number↵ OfAvengers, @NonNull SoundManager soundManager)
Inserts numberOfAvengers avenger units into the given UnitRepository.

Package Functions

- [RevengeAlertDialogBuilderAdapter](#) (@NonNull Context context)
Constructor of a RevengeAlertDialogBuilderAdapter.
- AlertDialog [create](#) ()
Creates an AlertDialog with the arguments supplied to this builder.
- [RevengeAlertDialogBuilderAdapter](#) [setPositiveCallback](#) (@NonNull DialogInterface.OnClickListener on↵ PositiveButtonClick)
Adds the given callback to the positive button of the built Dialog.
- [RevengeAlertDialogBuilderAdapter](#) [setNegativeCallback](#) (@NonNull DialogInterface.OnClickListener on↵ NegativeButtonClick)
Adds the given callback to the negative button of the built Dialog.

Private Attributes

- final AlertDialog.Builder [adapteeBuilder](#)
AlertDialog.Builder that is adapted by this class.

Static Private Attributes

- static final boolean [AVENGER_EPIC](#) = false
Boolean constant defining whether the default UnitEntity summoned by the Ability::REVENGE ability is epic.
- static final Ability [AVENGER_ABILITY](#) = Ability.NONE
Ability constant defining the Ability of the default UnitEntity summoned by the Ability::REVENGE ability.
- static final int [AVENGER_DAMAGE](#) = 8
Integer constant defining the damage of the default UnitEntity summoned by the Ability::REVENGE ability.
- static final Integer [AVENGER_SQUAD](#) = null
Integer constant defining the squad of the default UnitEntity summoned by the Ability::REVENGE ability.
- static final RowType [AVENGER_ROW](#) = RowType.MELEE
RowType constant defining the row of the default UnitEntity summoned by the Ability::REVENGE ability.

5.41.1 Detailed Description

An adapter class adapting AlertDialog.Builder to provide an interface for creating an AlertDialog asking the user whether they want to invoke the Ability::REVENGE ability.

Definition at line 23 of file [RevengeAlertDialogBuilderAdapter.java](#).

5.41.2 Constructor & Destructor Documentation

5.41.2.1 RevengeAlertDialogBuilderAdapter()

```
RevengeAlertDialogBuilderAdapter (
    @NonNull Context context ) [package]
```

Constructor of a RevengeAlertDialogBuilderAdapter.

Initializes the buttons with empty callbacks.

Parameters

<i>context</i>	Context of the built AlertDialog.
----------------	-----------------------------------

Definition at line 76 of file [RevengeAlertDialogBuilderAdapter.java](#).

5.41.3 Member Function Documentation

5.41.3.1 create()

```
AlertDialog create ( ) [package]
```

Creates an AlertDialog with the arguments supplied to this builder.

Basically just calls AlertDialog.Builder::create() on #adapteeBuilder.

See also

[AlertDialog.Builder::create\(\)](#)

Definition at line 108 of file [RevengeAlertDialogBuilderAdapter.java](#).

References [RevengeAlertDialogBuilderAdapter.adapteeBuilder](#).

Referenced by [RemoveUnitsUseCase.getRevengeDialog\(\)](#), and [ResetRepositoryUseCase.getRevengeDialog\(\)](#).

5.41.3.2 insertAvengers()

```
static Completable insertAvengers (
    @NonNull UnitRepository repository,
```

```
@IntRange(from=0) int numberOfAvengers,
@NonNull SoundManager soundManager ) [static]
```

Inserts numberOfAvengers avenger units into the given UnitRepository.

The inserted UnitEntity objects have the attributes defined in #AVENGER_EPIC, #AVENGER_DAMAGE, #AVENGER_ABILITY, #AVENGER_SQUAD, and #AVENGER_ROW.

Parameters

<i>repository</i>	UnitRepository where avengers are inserted.
<i>numberOfAvengers</i>	Integer defining how many avengers are inserted.

Returns

A Completable tracking operation status.

Definition at line 96 of file [RevengeAlertDialogBuilderAdapter.java](#).

References [RevengeAlertDialogBuilderAdapter.AVENGER_ABILITY](#), [RevengeAlertDialogBuilderAdapter.AVENGER_DAMAGE](#), [RevengeAlertDialogBuilderAdapter.AVENGER_EPIC](#), [RevengeAlertDialogBuilderAdapter.AVENGER_ROW](#), and [RevengeAlertDialogBuilderAdapter.AVENGER_SQUAD](#).

Referenced by [RemoveUnitsUseCase.getRevengeDialog\(\)](#), and [ResetRepositoryUseCase.getRevengeDialog\(\)](#).

5.41.3.3 setNegativeCallback()

```
RevengeAlertDialogBuilderAdapter setNegativeCallback (
    @NonNull DialogInterface.OnClickListener onNegativeButtonClick ) [package]
```

Adds the given callback to the negative button of the built Dialog.

Parameters

<i>onNegativeButtonClick</i>	DialogInterface.OnClickListener that is called, when the negative button is clicked.
------------------------------	--

Returns

The RevengeAlertDialogBuilder with the updated negative callback.

Definition at line 133 of file [RevengeAlertDialogBuilderAdapter.java](#).

References [RevengeAlertDialogBuilderAdapter.adapteeBuilder](#).

5.41.3.4 setPositiveCallback()

```
RevengeAlertDialogBuilderAdapter setPositiveCallback (
    @NonNull DialogInterface.OnClickListener onPositiveButtonClick ) [package]
```

Adds the given callback to the positive button of the built Dialog.

Callback should call insertAvengers().

Parameters

<i>onPositiveButtonClick</i>	DialogInterface.OnClickListener that is called, when the positive button is clicked.
------------------------------	--

Returns

The RevengeAlertDialogBuilder with the updated positive callback.

See also

[#insertAvengers\(UnitRepository, int, SoundManager\)](#)

Definition at line 121 of file [RevengeAlertDialogBuilderAdapter.java](#).

References [RevengeAlertDialogBuilderAdapter.adapteeBuilder](#).

Referenced by [RemoveUnitsUseCase.getRevengeDialog\(\)](#), and [ResetRepositoryUseCase.getRevengeDialog\(\)](#).

5.41.4 Member Data Documentation

5.41.4.1 adapteeBuilder

```
final AlertDialog.Builder adapteeBuilder [private]
```

AlertDialog.Builder that is adapted by this class.

Definition at line 68 of file [RevengeAlertDialogBuilderAdapter.java](#).

Referenced by [RevengeAlertDialogBuilderAdapter.create\(\)](#), [RevengeAlertDialogBuilderAdapter.setNegativeCallback\(\)](#), and [RevengeAlertDialogBuilderAdapter.setPositiveCallback\(\)](#).

5.41.4.2 AVENGER_ABILITY

```
final Ability AVENGER_ABILITY = Ability.NONE [static], [private]
```

Ability constant defining the Ability of the default UnitEntity summoned by the Ability::REVENGE ability.

See also

[Ability::REVENGE](#)

Definition at line 38 of file [RevengeAlertDialogBuilderAdapter.java](#).

Referenced by [RevengeAlertDialogBuilderAdapter.insertAvengers\(\)](#).

5.41.4.3 AVENGER_DAMAGE

```
final int AVENGER_DAMAGE = 8 [static], [private]
```

Integer constant defining the damage of the default UnitEntity summoned by the Ability::REVENGE ability.

See also

[Ability::REVENGE](#)

Definition at line 46 of file [RevengeAlertDialogBuilderAdapter.java](#).

Referenced by [RevengeAlertDialogBuilderAdapter.insertAvengers\(\)](#).

5.41.4.4 AVENGER_EPIC

```
final boolean AVENGER_EPIC = false [static], [private]
```

Boolean constant defining whether the default UnitEntity summoned by the Ability::REVENGE ability is epic.

See also

[Ability::REVENGE](#)

Definition at line 30 of file [RevengeAlertDialogBuilderAdapter.java](#).

Referenced by [RevengeAlertDialogBuilderAdapter.insertAvengers\(\)](#).

5.41.4.5 AVENGER_ROW

```
final RowType AVENGER_ROW = RowType.MELEE [static], [private]
```

RowType constant defining the row of the default UnitEntity summoned by the Ability::REVENGE ability.

See also

[Ability::REVENGE](#)

Definition at line 62 of file [RevengeAlertDialogBuilderAdapter.java](#).

Referenced by [RevengeAlertDialogBuilderAdapter.insertAvengers\(\)](#).

5.41.4.6 AVENGER_SQUAD

```
final Integer AVENGER_SQUAD = null [static], [private]
```

Integer constant defining the squad of the default UnitEntity summoned by the Ability::REVENGE ability.

See also

Ability::REVENGE

Definition at line 54 of file [RevengeAlertDialogBuilderAdapter.java](#).

Referenced by [RevengeAlertDialogBuilderAdapter.insertAvengers\(\)](#).

The documentation for this class was generated from the following file:

- gwent/domain/cases/RevengeAlertDialogBuilderAdapter.java

5.42 RowDao Interface Reference

An interface defining data access operations related to `rows` table.

Collaboration diagram for RowDao:



Public Member Functions

- Completable [insertRow](#) (@NonNull [RowEntity](#) row)
Inserts the given RowEntity into rows.
- Completable [clearRows](#) ()
Deletes all RowEntity objects from rows.
- Completable [updateWeather](#) (@NonNull [RowType](#) row)
Flips RowEntity::weather for the given RowType in rows.
- Completable [clearWeather](#) ()
Sets RowEntity::weather to false for all RowEntity elements in rows.
- Completable [updateHorn](#) (@NonNull [RowType](#) row)
Flips RowEntity::horn for the given RowType in rows.
- Single< Boolean > [isWeather](#) (@NonNull [RowType](#) row)
Fetches RowEntity::weather for the given RowType in rows.
- Flowable< Boolean > [isWeatherFlowable](#) (@NonNull [RowType](#) row)
Fetches a Flowable of RowEntity::weather for the given RowType in rows.
- Single< Boolean > [isHorn](#) (@NonNull [RowType](#) row)
Fetches RowEntity::horn for the given RowType in rows.
- Flowable< Boolean > [isHornFlowable](#) (@NonNull [RowType](#) row)
Fetches a Flowable of RowEntity::horn for the given RowType in rows.

5.42.1 Detailed Description

An interface defining data access operations related to `rows` table.

The implementation is generated by the Room framework.

Definition at line 19 of file [RowDao.java](#).

5.42.2 Member Function Documentation

5.42.2.1 clearRows()

`Completable clearRows ()`

Deletes all `RowEntity` objects from `rows`.

Returns

A `Completable` tracking operation status.

Referenced by [UnitRepository.reset\(\)](#).

5.42.2.2 clearWeather()

`Completable clearWeather ()`

Sets `RowEntity::weather` to `false` for all `RowEntity` elements in `rows`.

Returns

A `Completable` tracking operation status.

Referenced by [UnitRepository.clearWeather\(\)](#).

5.42.2.3 insertRow()

`Completable insertRow (`

`@NonNull RowEntity row)`

Inserts the given `RowEntity` into `rows`.

If the same `RowType` is already in the table, the new insert is ignored.

Parameters

<code>row</code>	<code>RowEntity</code> that is being inserted.
------------------	--

Returns

A `Completable` tracking operation status.

Referenced by [UnitRepository.initializeRows\(\)](#).

5.42.2.4 isHorn()

`Single< Boolean > isHorn (`

`@NonNull RowType row)`

Fetches `RowEntity::horn` for the given `RowType` in `rows`.

See also

[#isHornFlowable\(RowType\)](#)

Parameters

<code>row</code>	<code>RowType</code> of the queried <code>RowEntity</code> .
------------------	--

Returns

A Single tracking operation status and returning the value.

Referenced by [UnitRepository.isHorn\(\)](#).

5.42.2.5 isHornFlowable()

```
Flowable< Boolean > isHornFlowable (
    @NonNull RowType row )
```

Fetches a Flowable of RowEntity::horn for the given RowType in rows.

Parameters

<i>row</i>	RowType of the queried RowEntity.
------------	-----------------------------------

Returns

A Flowable emitting the values.

See also

[#isHorn\(RowType\)](#)

Referenced by [UnitRepository.isHornFlowable\(\)](#).

5.42.2.6 isWeather()

```
Single< Boolean > isWeather (
    @NonNull RowType row )
```

Fetches RowEntity::weather for the given RowType in rows.

See also

[#isWeatherFlowable\(RowType\)](#)

Parameters

<i>row</i>	RowType of the queried RowEntity.
------------	-----------------------------------

Returns

A Single tracking operation status and returning the value.

Referenced by [UnitRepository.isWeather\(\)](#).

5.42.2.7 isWeatherFlowable()

```
Flowable< Boolean > isWeatherFlowable (
    @NonNull RowType row )
```

Fetches a Flowable of RowEntity::weather for the given RowType in rows.

Parameters

<i>row</i>	RowType of the queried RowEntity.
------------	-----------------------------------

Returns

A Flowable emitting the values.

See also

`#isWeather(RowType)`

Referenced by [UnitRepository.isWeatherFlowable\(\)](#).

5.42.2.8 updateHorn()

```
Completable updateHorn (
    @NonNull RowType row )
```

Flips RowEntity::horn for the given RowType in rows.

Parameters

<i>row</i>	RowType of the updated RowEntity.
------------	-----------------------------------

Returns

A Completable tracking operation status.

Referenced by [UnitRepository.switchHorn\(\)](#).

5.42.2.9 updateWeather()

```
Completable updateWeather (
    @NonNull RowType row )
```

Flips RowEntity::weather for the given RowType in rows.

Parameters

<i>row</i>	RowType of the updated RowEntity.
------------	-----------------------------------

Returns

A Completable tracking operation status.

Referenced by [UnitRepository.switchWeather\(\)](#).

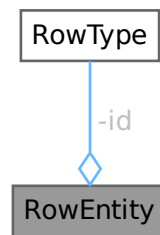
The documentation for this interface was generated from the following file:

- gwent/data/RowDao.java

5.43 RowEntity Class Reference

A class representing the state of an attack row, i.e., weather and commanders horn.

Collaboration diagram for RowEntity:



Package Functions

- [RowEntity](#) (@NonNull [RowType](#) id)
Constructor of a RowEntity.
- [RowType](#) getId ()
Getter for #id.
- boolean isWeather ()
Getter for #weather.
- void setWeather (boolean weather)
Setter for #weather.
- boolean isHorn ()
Getter for #horn.
- void setHorn (boolean horn)
Setter for #horn.

Private Attributes

- final [RowType](#) id
Defines the type of the row as one of the values in RowType, thereby limiting the number of different rows to three.
- boolean weather
Defines whether the weather debuff is present in the represented row, i.e., if set to true, the damage value of units in the represented row is set to 1.
- boolean horn
Defines whether the commanders horn buff is present in the represented row, i.e., if set to true, the damage value of units in the represented row is doubled.

5.43.1 Detailed Description

A class representing the state of an attack row, i.e., weather and commanders horn.
Is a persistent Entity and is therefore saved in a database table named `rows`.
Definition at line 14 of file [RowEntity.java](#).

5.43.2 Constructor & Destructor Documentation

5.43.2.1 RowEntity()

```
RowEntity (
    @NonNull RowType id ) [package]
```

Constructor of a RowEntity.

#weather and #horn cannot be set here since a new row must start with the respective default values.

Parameters

<i>id</i>	RowType of the represented row.
-----------	---------------------------------

Definition at line 47 of file [RowEntity.java](#).

5.43.3 Member Function Documentation

5.43.3.1 getId()

```
RowType getId ( ) [package]
```

Getter for #id.

Only used by Room extension.

Returns

RowType of the represented row.

Definition at line 58 of file [RowEntity.java](#).

5.43.3.2 isHorn()

```
boolean isHorn ( ) [package]
```

Getter for #horn.

Only used by Room extension.

Returns

Boolean representing the current status of the horn buff.

Definition at line 88 of file [RowEntity.java](#).

5.43.3.3 isWeather()

```
boolean isWeather ( ) [package]
```

Getter for #weather.

Only used by Room extension.

Returns

Boolean representing the current status of the weather debuff.

Definition at line 68 of file [RowEntity.java](#).

5.43.3.4 setHorn()

```
void setHorn (
    boolean horn ) [package]
```

Setter for #horn.

Only used by Room extension.

Parameters

<i>horn</i>	Boolean representing the new status of the horn buff.
-------------	---

Definition at line 98 of file [RowEntity.java](#).

5.43.3.5 setWeather()

```
void setWeather (
    boolean weather ) [package]
```

Setter for #weather.

Only used by Room extension.

Parameters

<i>weather</i>	Boolean representing the new status of the weather debuff.
----------------	--

Definition at line 78 of file [RowEntity.java](#).

5.43.4 Member Data Documentation

5.43.4.1 horn

```
boolean horn [private]
```

Defines whether the commanders horn buff is present in the represented row, i.e., if set to true, the damage value of units in the represented row is doubled.

Is set to `false` by default.

Definition at line 38 of file [RowEntity.java](#).

5.43.4.2 id

```
final RowType id [private]
```

Defines the type of the row as one of the values in RowType, thereby limiting the number of different rows to three.

Primary key of the represented row.

Definition at line 22 of file [RowEntity.java](#).

5.43.4.3 weather

```
boolean weather [private]
```

Defines whether the weather debuff is present in the represented row, i.e., if set to true, the damage value of units in the represented row is set to 1.

Is set to `false` by default.

Definition at line 30 of file [RowEntity.java](#).

The documentation for this class was generated from the following file:

- gwent/data/RowEntity.java

5.44 RowType Enum Reference

An enum listing the different types of attack rows.

Collaboration diagram for RowType:

RowType

Public Attributes

- [MELEE](#)
Represents the row containing melee combat units.
- [RANGE](#)
Represents the row containing ranged combat units.
- [SIEGE](#)

Represents the row containing siege combat units.

5.44.1 Detailed Description

An enum listing the different types of attack rows.
Definition at line 6 of file [RowType.java](#).

5.44.2 Member Data Documentation

5.44.2.1 MELEE

MELEE

Represents the row containing melee combat units.
Definition at line 10 of file [RowType.java](#).

5.44.2.2 RANGE

RANGE

Represents the row containing ranged combat units.
Definition at line 15 of file [RowType.java](#).

5.44.2.3 SIEGE

SIEGE

Represents the row containing siege combat units.
Definition at line 21 of file [RowType.java](#).

The documentation for this enum was generated from the following file:

- gwent/data/RowType.java

5.45 RowUiState Class Reference

A data class encapsulating the visible state of a row in the main view of the application.
Collaboration diagram for RowUiState:

RowUiState

Public Member Functions

- [RowUiState](#) ([@IntRange](#)(from=0) [int](#) damage, [boolean](#) weather, [boolean](#) horn, [@IntRange](#)(from=0) [int](#) units)
Constructor of a RowUiState encapsulating the given data.
- [int](#) [getDamage](#) ()
Returns the summed-up damage of all units in this row.
- [boolean](#) [isWeather](#) ()
Returns whether the weather debuff is active in this row.
- [boolean](#) [isHorn](#) ()
Returns whether the commander's horn buff is active in this row.
- [int](#) [getUnits](#) ()
Returns the number of units in this row.
- [boolean](#) [equals](#) ([Object](#) o)

Private Attributes

- final int [damage](#)
Defines the summed-up damage of all units in this row.
- final boolean [weather](#)
Defines whether the weather debuff is active in this row.
- final boolean [horn](#)
Defines whether the commander's horn buff is active in this row.
- final int [units](#)
Defines the number of units in this row.

5.45.1 Detailed Description

A data class encapsulating the visible state of a row in the main view of the application.
Definition at line 10 of file [RowUiState.java](#).

5.45.2 Constructor & Destructor Documentation**5.45.2.1 RowUiState()**

```
RowUiState (
    @IntRange(from=0) int damage,
    boolean weather,
    boolean horn,
    @IntRange(from=0) int units )
```

Constructor of a RowUiState encapsulating the given data.

Parameters

<i>damage</i>	Integer representing the summed-up damage of all units.
<i>weather</i>	Boolean defining whether the weather debuff is active.
<i>horn</i>	Boolean defining whether the commander's horn buff is active.
<i>units</i>	Integer representing the number of units.

Exceptions

<i>IllegalArgumentException</i>	When damage or units is negative.
---------------------------------	-----------------------------------

Definition at line 41 of file [RowUiState.java](#).

References [RowUiState.damage](#), [RowUiState.horn](#), [RowUiState.units](#), and [RowUiState.weather](#).

5.45.3 Member Function Documentation**5.45.3.1 equals()**

```
boolean equals (
    Object o )
```

Definition at line 89 of file [RowUiState.java](#).

5.45.3.2 getDamage()

```
int getDamage ( )
```

Returns the summed-up damage of all units in this row.

Returns

An Integer representing the summed-up damage of all units.

Definition at line 59 of file [RowUiState.java](#).

References [RowUiState.damage](#).

Referenced by [GameBoardViewModel.getModel\(\)](#).

5.45.3.3 getUnits()

```
int getUnits ( )
```

Returns the number of units in this row.

Returns

An Integer representing the number of units in this row.

Definition at line 84 of file [RowUiState.java](#).

References [RowUiState.units](#).

Referenced by [GameBoardViewModel.getModel\(\)](#).

5.45.3.4 isHorn()

```
boolean isHorn ( )
```

Returns whether the commander's horn buff is active in this row.

Returns

A Boolean whether the commander's horn buff is active.

Definition at line 75 of file [RowUiState.java](#).

References [RowUiState.horn](#).

Referenced by [GameBoardViewModel.getModel\(\)](#).

5.45.3.5 isWeather()

```
boolean isWeather ( )
```

Returns whether the weather debuff is active in this row.

Returns

A Boolean defining whether the weather debuff is active.

Definition at line 67 of file [RowUiState.java](#).

References [RowUiState.weather](#).

Referenced by [GameBoardViewModel.getModel\(\)](#).

5.45.4 Member Data Documentation

5.45.4.1 damage

```
final int damage [private]
```

Defines the summed-up damage of all units in this row.

Definition at line 15 of file [RowUiState.java](#).

Referenced by [RowUiState.RowUiState\(\)](#), and [RowUiState.getDamage\(\)](#).

5.45.4.2 horn

```
final boolean horn [private]
```

Defines whether the commander's horn buff is active in this row.

Definition at line 25 of file [RowUiState.java](#).

Referenced by [RowUiState.RowUiState\(\)](#), and [RowUiState.isHorn\(\)](#).

5.45.4.3 units

```
final int units [private]
```

Defines the number of units in this row.

Definition at line 31 of file [RowUiState.java](#).

Referenced by [RowUiState.RowUiState\(\)](#), and [RowUiState.getUnits\(\)](#).

5.45.4.4 weather

```
final boolean weather [private]
```

Defines whether the weather debuff is active in this row.

Definition at line 20 of file [RowUiState.java](#).

Referenced by [RowUiState.RowUiState\(\)](#), and [RowUiState.isWeather\(\)](#).

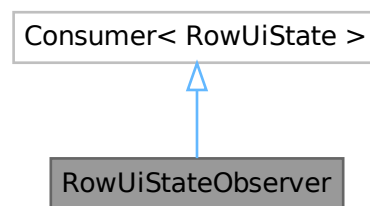
The documentation for this class was generated from the following file:

- gwent/ui/main/RowUiState.java

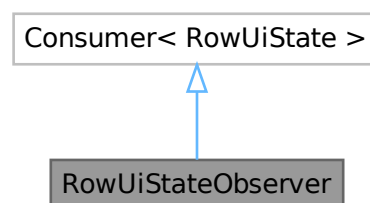
5.46 RowUiStateObserver Class Reference

An observer class responsible for updating the views of the row defined in #row when notified, i.e., when a new RowUiState is produced the observed Flowable.

Inheritance diagram for RowUiStateObserver:



Collaboration diagram for RowUiStateObserver:



Public Member Functions

- void [accept](#) (@NonNull [RowUiState](#) rowUiState)

Changes values in #damageView and #unitView as well as the image in #weatherView and #hornView.

Static Public Member Functions

- static [RowUiStateObserver](#) [getObserver](#) (@NonNull [RowType](#) row, @NonNull [TextView](#) [damageView](#), @NonNull [ImageView](#) [weatherView](#), @NonNull [ImageView](#) [hornView](#), @NonNull [TextView](#) [unitView](#))
Returns a new [RowUiStateObserver](#) for the given row updating the given views.

Private Member Functions

- [RowUiStateObserver](#) (@NonNull [TextView](#) [damageView](#), @NonNull [ImageView](#) [weatherView](#), @DrawableRes int [weatherResource](#), @NonNull [ImageView](#) [hornView](#), @NonNull [TextView](#) [unitView](#))
Constructor of a [RowUiStateObserver](#) updating the given views when [#onChanged\(\)](#) is called.

Private Attributes

- final [TextView](#) [damageView](#)
A [TextView](#) showing the user the summed-up damage of all units in this row, i.e., [RowUiState::damage](#).
- final [ImageView](#) [weatherView](#)
An [ImageView](#) responsible for showing the current state of the weather debuff in this row.
- final int [weatherResource](#)
An Integer representing the drawable resource shown by [#weatherView](#) when the weather debuff is active, i.e., when [RowUiState::weather](#) is `true`.
- final [ImageView](#) [hornView](#)
An [ImageView](#) responsible for showing the current state of the commander's horn buff in this row.
- final [TextView](#) [unitView](#)
A [TextView](#) showing the number of units in this row, i.e., [RowUiState::units](#).

5.46.1 Detailed Description

An observer class responsible for updating the views of the row defined in [#row](#) when notified, i.e., when a new [RowUiState](#) is produced the observed [Flowable](#).

See also

[RowUiState](#)

Definition at line 22 of file [RowUiStateObserver.java](#).

5.46.2 Constructor & Destructor Documentation

5.46.2.1 RowUiStateObserver()

```
RowUiStateObserver (
    @NonNull TextView damageView,
    @NonNull ImageView weatherView,
    @DrawableRes int weatherResource,
    @NonNull ImageView hornView,
    @NonNull TextView unitView ) [private]
```

Constructor of a [RowUiStateObserver](#) updating the given views when [#onChanged\(\)](#) is called.
Should only be called by [#getObserver\(\)](#).

Parameters

damageView	TextView showing the summed-up damage of all units in the observed row.
weatherView	ImageView showing the current state of the weather debuff of the observed row.
weatherResource	Integer representing the drawable resource shown when the weather debuff is active.
hornView	ImageView showing the current state of the commander's horn buff of the observed row.
unitView	TextView showing the number of units in the observed row.

See also

`#getObserver(RowType, TextView, ImageView, ImageView, TextView)`

Definition at line 74 of file [RowUiStateObserver.java](#).

References [RowUiStateObserver.damageView](#), [RowUiStateObserver.hornView](#), [RowUiStateObserver.unitView](#), [RowUiStateObserver.weatherResource](#), and [RowUiStateObserver.weatherView](#).

5.46.3 Member Function Documentation

5.46.3.1 accept()

```
void accept (
    @NonNull RowUiState rowUiState )
```

Changes values in `#damageView` and `#unitView` as well as the image in `#weatherView` and `#hornView`. Image switches are animated using `ImageViewSwitchAnimator`.

Parameters

<i>rowUiState</i>	RowUiState representing the updated state of the row.
-------------------	---

See also

`ImageViewSwitchAnimator`

Definition at line 127 of file [RowUiStateObserver.java](#).

References [ImageViewSwitchAnimator.animatedSwitch\(\)](#), [RowUiStateObserver.damageView](#), [RowUiStateObserver.hornView](#), [RowUiStateObserver.unitView](#), [RowUiStateObserver.weatherResource](#), and [RowUiStateObserver.weatherView](#).

5.46.3.2 getObserver()

```
static RowUiStateObserver getObserver (
    @NonNull RowType row,
    @NonNull TextView damageView,
    @NonNull ImageView weatherView,
    @NonNull ImageView hornView,
    @NonNull TextView unitView ) [static]
```

Returns a new `RowUiStateObserver` for the given row updating the given views.

Factory method for `RowUiStateObserver`. `#weatherResource` is defined according to the given `RowType`.

Parameters

<i>row</i>	RowType defining which row is being observed.
<i>damageView</i>	TextView showing the summed-up damage of all units in the observed row.
<i>weatherView</i>	ImageView showing the current state of the weather debuff of the observed row.
<i>hornView</i>	ImageView showing the current state of the commander's horn buff of the observed row.
<i>unitView</i>	TextView showing the number of units in the observed row.

Returns

A `RowUiStateObserver` updating the given views.

See also

`#RowUiStateObserver(TextView, ImageView, int, ImageView, TextView)`

Definition at line 99 of file [RowUiStateObserver.java](#).

References [RowUiStateObserver.damageView](#), [RowUiStateObserver.hornView](#), [RowUiStateObserver.unitView](#), [RowUiStateObserver.weatherResource](#), and [RowUiStateObserver.weatherView](#).

Referenced by [MainActivity.initializeViewModel\(\)](#).

5.46.4 Member Data Documentation

5.46.4.1 damageView

```
final TextView damageView [private]
```

A `TextView` showing the user the summed-up damage of all units in this row, i.e., `RowUiState::damage`.

Definition at line 28 of file [RowUiStateObserver.java](#).

Referenced by [RowUiStateObserver.RowUiStateObserver\(\)](#), [RowUiStateObserver.accept\(\)](#), and [RowUiStateObserver.getObserver\(\)](#).

5.46.4.2 hornView

```
final ImageView hornView [private]
```

An `ImageView` responsible for showing the current state of the commander's horn buff in this row.

Is gray when `RowUiState::horn` is `false`.

See also

`#weatherResource`

Definition at line 55 of file [RowUiStateObserver.java](#).

Referenced by [RowUiStateObserver.RowUiStateObserver\(\)](#), [RowUiStateObserver.accept\(\)](#), and [RowUiStateObserver.getObserver\(\)](#).

5.46.4.3 unitView

```
final TextView unitView [private]
```

A `TextView` showing the number of units in this row, i.e., `RowUiState::units`.

Definition at line 61 of file [RowUiStateObserver.java](#).

Referenced by [RowUiStateObserver.RowUiStateObserver\(\)](#), [RowUiStateObserver.accept\(\)](#), and [RowUiStateObserver.getObserver\(\)](#).

5.46.4.4 weatherResource

```
final int weatherResource [private]
```

An `Integer` representing the drawable resource shown by `#weatherView` when the weather debuff is active, i.e., when `RowUiState::weather` is `true`.

See also

`#weatherView`

Definition at line 46 of file [RowUiStateObserver.java](#).

Referenced by [RowUiStateObserver.RowUiStateObserver\(\)](#), [RowUiStateObserver.accept\(\)](#), and [RowUiStateObserver.getObserver\(\)](#).

5.46.4.5 weatherView

```
final ImageView weatherView [private]
```

An `ImageView` responsible for showing the current state of the weather debuff in this row.

Displays the resource in `#weatherResource` when `RowUiState::weather` is `true`.

See also

`#weatherResource`

Definition at line 37 of file [RowUiStateObserver.java](#).

Referenced by [RowUiStateObserver.RowUiStateObserver\(\)](#), [RowUiStateObserver.accept\(\)](#), and [RowUiStateObserver.getObserver\(\)](#).

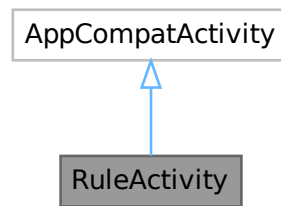
The documentation for this class was generated from the following file:

- `gwent/ui/main/RowUiStateObserver.java`

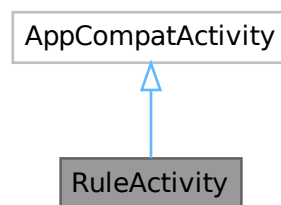
5.47 RuleActivity Class Reference

An `AppCompatActivity` used to present the rules of the game.

Inheritance diagram for RuleActivity:



Collaboration diagram for RuleActivity:



Public Member Functions

- void [onCreate](#) (Bundle savedInstanceState)
Initializes the content of the selected rule.
- boolean [onOptionsItemSelected](#) (@NonNull MenuItem item)
Called whenever a MenuItem in the options menu is selected.

Static Public Attributes

- static final String [INTENT_EXTRA_TAG](#) = "rule_section"
String constant defining the identifier where the requested RuleSection is provided in the calling android.content.Intent.

5.47.1 Detailed Description

An AppCompatActivity used to present the rules of the game.

Can be accessed from the SettingsActivity. When called, a selected RuleSection must always be given through the calling android.content.Intent at the key defined in INTENT_EXTRA_TAG.

Definition at line 22 of file [RuleActivity.java](#).

5.47.2 Member Function Documentation

5.47.2.1 onCreate()

```
void onCreate (
```

```
Bundle savedInstanceState )
```

Initializes the content of the selected rule.

Parameters

<i>savedInstanceState</i>	If the activity is being re-initialized after previously being shut down then this Bundle contains the data it most recently supplied in <code>onSaveInstanceState</code> . Note: Otherwise it is null.
---------------------------	--

Definition at line 37 of file [RuleActivity.java](#).

References [RuleActivity.INTENT_EXTRA_TAG](#).

5.47.2.2 onOptionsItemSelected()

```
boolean onOptionsItemSelected (
    @NonNull MenuItem item )
```

Called whenever a MenuItem in the options menu is selected.

Returns to the calling `android.app.Activity` when the `android.R.id#home` item was selected.

Parameters

<i>item</i>	MenuItem that was selected.
-------------	-----------------------------

Returns

Boolean defining whether the call has been handled.

Definition at line 97 of file [RuleActivity.java](#).

5.47.3 Member Data Documentation

5.47.3.1 INTENT_EXTRA_TAG

```
final String INTENT_EXTRA_TAG = "rule_section" [static]
```

String constant defining the identifier where the requested RuleSection is provided in the calling `android.content.Intent`.

Definition at line 27 of file [RuleActivity.java](#).

Referenced by [SettingsRuleFragment.getSectionClickListener\(\)](#), and [RuleActivity.onCreate\(\)](#).

The documentation for this class was generated from the following file:

- `gwent/ui/settings/RuleActivity.java`

5.48 RuleSection Enum Reference

An Enum used to discern the different possible sections shown by RuleActivity.

Collaboration diagram for RuleSection:

RuleSection

Public Attributes

- [GENERAL](#)
Represents a section containing general background-information on the game Gwent.
- [COURSE](#)
Represents a section containing information on the course of a game of Gwent.
- [FACTIONS](#)
Represents a section containing information on the factions playable in Gwent.
- [COMMANDER](#)
Represents a section containing information on the available commanders of the factions and their abilities.
- [CARDS](#)
Represents a section containing information on the different types of cards in Gwent.
- [CARD_ABILITIES](#)
Represents a section containing information on the possible abilities of cards in Gwent.
- [SPECIAL_CARDS](#)
Represents a section containing information on special cards available in Gwent.

5.48.1 Detailed Description

An Enum used to discern the different possible sections shown by `RuleActivity`.
Definition at line 6 of file [RuleSection.java](#).

5.48.2 Member Data Documentation

5.48.2.1 CARD_ABILITIES

`CARD_ABILITIES`

Represents a section containing information on the possible abilities of cards in Gwent.
Definition at line 35 of file [RuleSection.java](#).

5.48.2.2 CARDS

`CARDS`

Represents a section containing information on the different types of cards in Gwent.
Definition at line 30 of file [RuleSection.java](#).

5.48.2.3 COMMANDER

`COMMANDER`

Represents a section containing information on the available commanders of the factions and their abilities.
Definition at line 25 of file [RuleSection.java](#).

5.48.2.4 COURSE

`COURSE`

Represents a section containing information on the course of a game of Gwent.
Definition at line 15 of file [RuleSection.java](#).

5.48.2.5 FACTIONS

`FACTIONS`

Represents a section containing information on the factions playable in Gwent.
Definition at line 20 of file [RuleSection.java](#).

5.48.2.6 GENERAL

`GENERAL`

Represents a section containing general background-information on the game Gwent.
Definition at line 10 of file [RuleSection.java](#).

5.48.2.7 SPECIAL_CARDS

SPECIAL_CARDS

Represents a section containing information on special cards available in Gwent.

Definition at line 41 of file [RuleSection.java](#).

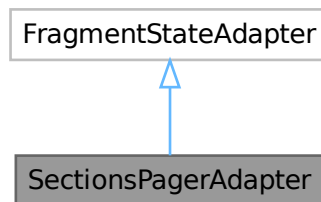
The documentation for this enum was generated from the following file:

- gwent/ui/settings/RuleSection.java

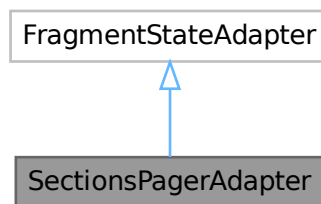
5.49 SectionsPagerAdapter Class Reference

A `FragmentManager` that returns an `IntroductionFragment` corresponding to one of the sections/tabs/pages.

Inheritance diagram for `SectionsPagerAdapter`:



Collaboration diagram for `SectionsPagerAdapter`:



Public Member Functions

- [SectionsPagerAdapter](#) (`@NonNull` `FragmentManager` `fragmentManager`)
Constructor of a `SectionsPagerAdapter` for the given `FragmentManager`.
- `int` [getItemCount](#) ()
Returns the total number of items in the data set held by the adapter.
- `FragmentManager` [createFragment](#) (`int` `position`)
Provides a new `IntroductionFragment` associated with the specified position.

5.49.1 Detailed Description

A `FragmentManager` that returns an `IntroductionFragment` corresponding to one of the sections/tabs/pages.

Definition at line 12 of file [SectionsPagerAdapter.java](#).

5.49.2 Constructor & Destructor Documentation

5.49.2.1 SectionsPagerAdapter()

```
SectionsPagerAdapter (
    @NonNull FragmentActivity fragmentActivity )
```

Constructor of a `SectionsPagerAdapter` for the given `FragmentActivity`.

Parameters

<i>fragmentActivity</i>	<code>FragmentActivity</code> that uses this adapter.
-------------------------	---

Definition at line 18 of file [SectionsPagerAdapter.java](#).

5.49.3 Member Function Documentation

5.49.3.1 createFragment()

```
Fragment createFragment (
    int position )
```

Provides a new `IntroductionFragment` associated with the specified position.

Parameters

<i>position</i>	Integer defining the page that is queried.
-----------------	--

Returns

A `Fragment` that will be shown at the specified position.

Definition at line 39 of file [SectionsPagerAdapter.java](#).

References [IntroductionFragment.newInstance\(\)](#).

5.49.3.2 getItemCount()

```
int getItemCount ( )
```

Returns the total number of items in the data set held by the adapter.

Essentially queries `IntroductionFragment#PAGES_COUNT`.

Returns

An `Integer` defining the number of pages.

Definition at line 28 of file [SectionsPagerAdapter.java](#).

References [IntroductionFragment.PAGES_COUNT](#).

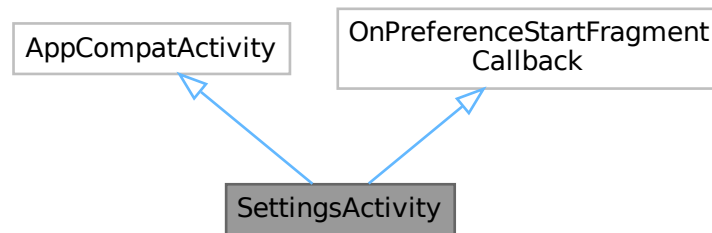
The documentation for this class was generated from the following file:

- [gwent/ui/introduction/SectionsPagerAdapter.java](#)

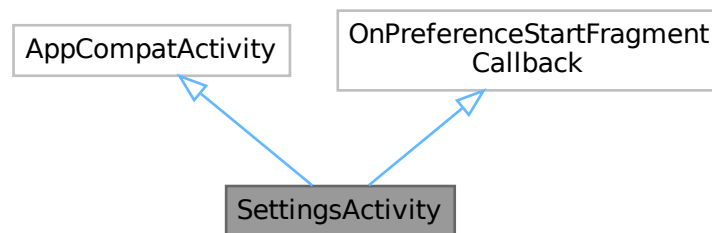
5.50 SettingsActivity Class Reference

An `AppCompatActivity` implementing `PreferenceFragmentCompat.OnPreferenceStartFragmentCallback` that is used by the user to manage the `SharedPreferences` of the application.

Inheritance diagram for `SettingsActivity`:



Collaboration diagram for `SettingsActivity`:



Public Member Functions

- boolean `onOptionsItemSelected` (@NonNull MenuItem item)
Called whenever a MenuItem in the options menu is selected.
- boolean `onPreferenceStartFragment` (@NonNull PreferenceFragmentCompat caller, @NonNull Preference pref)
Called when the user has clicked on a Preference that has a Fragment class name associated with it.

Protected Member Functions

- void `onCreate` (@Nullable Bundle savedInstanceState)
Initializes layout and ActionBar as well as creates and displays a new SettingsHeaderFragment.
- void `onSaveInstanceState` (@Nullable Bundle outState)
Saves the currently visible Fragment into the given Bundle.

Static Private Attributes

- static final String `CURRENT_FRAGMENT_KEY` = "currentFragment"
String constant defining the key in savedInstanceState Bundles where the last shown Fragment is saved.

5.50.1 Detailed Description

An `AppCompatActivity` implementing `PreferenceFragmentCompat.OnPreferenceStartFragmentCallback` that is used by the user to manage the `SharedPreferences` of the application.

Definition at line 24 of file [SettingsActivity.java](#).

5.50.2 Member Function Documentation

5.50.2.1 onCreate()

```
void onCreate (
    @Nullable Bundle savedInstanceState ) [protected]
```

Initializes layout and `ActionBar` as well as creates and displays a new `SettingsHeaderFragment`.

Parameters

<i>savedInstanceState</i>	If the activity is being re-initialized after previously being shut down then this <code>Bundle</code> contains the data it most recently supplied in <code>onSaveInstanceState</code> . Note: Otherwise it is null.
---------------------------	---

Definition at line 38 of file [SettingsActivity.java](#).

References [SettingsActivity.CURRENT_FRAGMENT_KEY](#).

5.50.2.2 onOptionsItemSelected()

```
boolean onOptionsItemSelected (
    @NonNull MenuItem item )
```

Called whenever a `MenuItem` in the options menu is selected.

Returns to the calling `android.app.Activity` when the `android.R.id#home` item was selected.

Parameters

<i>item</i>	<code>MenuItem</code> that was selected.
-------------	--

Returns

`Boolean` defining whether the call has been handled.

Definition at line 85 of file [SettingsActivity.java](#).

5.50.2.3 onPreferenceStartFragment()

```
boolean onPreferenceStartFragment (
    @NonNull PreferenceFragmentCompat caller,
    @NonNull Preference pref )
```

Called when the user has clicked on a `Preference` that has a `Fragment` class name associated with it.

Switches to an instance of the given `Fragment`.

Parameters

<i>caller</i>	<code>PreferenceFragmentCompat</code> requesting navigation.
<i>pref</i>	<code>Preference</code> requesting the <code>Fragment</code> .

Returns

`Boolean` defining whether the `Fragment` creation has been handled.

Definition at line 102 of file [SettingsActivity.java](#).

5.50.2.4 onSaveInstanceState()

```
void onSaveInstanceState (
    @NonNull Bundle outState ) [protected]
```

Saves the currently visible `Fragment` into the given `Bundle`.

Parameters

<i>outState</i>	<code>Bundle</code> in which the currently used <code>Fragment</code> is saved.
-----------------	---

Definition at line 66 of file [SettingsActivity.java](#).

References [SettingsActivity.CURRENT_FRAGMENT_KEY](#).

5.50.3 Member Data Documentation

5.50.3.1 CURRENT_FRAGMENT_KEY

```
final String CURRENT_FRAGMENT_KEY = "currentFragment" [static], [private]
```

`String` constant defining the key in savedInstanceState `Bundles` where the last shown `Fragment` is saved.

Definition at line 29 of file [SettingsActivity.java](#).

Referenced by [SettingsActivity.onCreate\(\)](#), and [SettingsActivity.onSaveInstanceState\(\)](#).

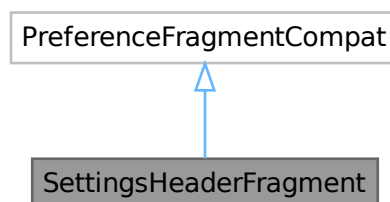
The documentation for this class was generated from the following file:

- `gwent/ui/settings/SettingsActivity.java`

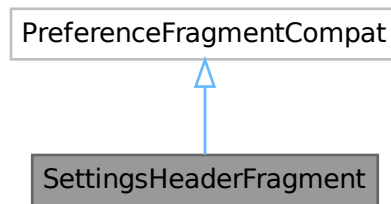
5.51 SettingsHeaderFragment Class Reference

A `PreferenceFragmentCompat` class encapsulating the main preference screen, i.e., the `Preferences` defined in `R.xml#header_preferences`.

Inheritance diagram for `SettingsHeaderFragment`:



Collaboration diagram for SettingsHeaderFragment:



Public Member Functions

- void `onCreatePreferences` (Bundle savedInstanceState, String rootKey)
Called during onCreate (Bundle) to supply the preferences for this fragment.

5.51.1 Detailed Description

A `PreferenceFragmentCompat` class encapsulating the main preference screen, i.e., the Preferences defined in `R.xml#header_preferences`.

Definition at line 18 of file [SettingsHeaderFragment.java](#).

5.51.2 Member Function Documentation

5.51.2.1 onCreatePreferences()

```
void onCreatePreferences (
    Bundle savedInstanceState,
    String rootKey )
```

Called during `onCreate (Bundle)` to supply the preferences for this fragment.

Sets shown Preferences from `R.xml#header_preferences` and registers an `androidx.preference.Preference` on the Preference at `R.string#preference_key_introduction` to start a new `IntroductionActivity`.

Parameters

<i>savedInstanceState</i>	If the fragment is being re-created from a previous saved state, this is the state.
<i>rootKey</i>	If non-null, this preference fragment should be rooted at the <code>androidx.preference.PreferenceScreen</code> with this key.

Definition at line 31 of file [SettingsHeaderFragment.java](#).

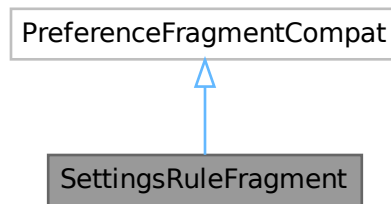
The documentation for this class was generated from the following file:

- `gwent/ui/settings/SettingsHeaderFragment.java`

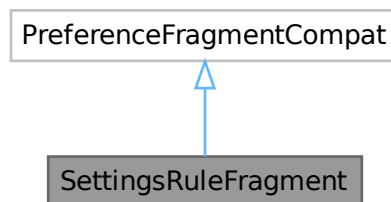
5.52 SettingsRuleFragment Class Reference

A `PreferenceFragmentCompat` class encapsulating the rule preference screen, i.e., the rule sections defined in `R.xml#rule_preferences`.

Inheritance diagram for SettingsRuleFragment:



Collaboration diagram for SettingsRuleFragment:



Public Member Functions

- void [onCreatePreferences](#) (Bundle savedInstanceState, String rootKey)
Called during onCreate(Bundle) to supply the preferences for this fragment.

Private Member Functions

- Preference.OnPreferenceClickListener [getSectionClickListener](#) (@NonNull RuleSection section)
Creates an Preference.OnPreferenceClickListener that starts a new RuleActivity for the given RuleSection using an Intent.

5.52.1 Detailed Description

A `PreferenceFragmentCompat` class encapsulating the rule preference screen, i.e., the rule sections defined in `R.xml#rule_preferences`.
 Definition at line 31 of file [SettingsRuleFragment.java](#).

5.52.2 Member Function Documentation

5.52.2.1 getSectionClickListener()

```

Preference.OnPreferenceClickListener getSectionClickListener (
    @NonNull RuleSection section ) [private]

```

Creates an `Preference.OnPreferenceClickListener` that starts a new `RuleActivity` for the given `RuleSection` using an `Intent`.

The `Intent` provides the requested `RuleSection` to the `RuleActivity` using `Intent#putExtra(String, Serializable)` with `RuleActivity#INTENT_EXTRA_TAG` as a tag.

Parameters

<i>section</i>	<code>RuleSection</code> that is requested.
----------------	---

Returns

An `Preference.OnPreferenceClickListener` calling a `RuleActivity`.

Definition at line 43 of file [SettingsRuleFragment.java](#).

References [RuleActivity.INTENT_EXTRA_TAG](#).

Referenced by [SettingsRuleFragment.onCreatePreferences\(\)](#).

5.52.2.2 onCreatePreferences()

```
void onCreatePreferences (
    Bundle savedInstanceState,
    String rootKey )
```

Called during `onCreate(Bundle)` to supply the preferences for this fragment.

Sets shown Preferences from `R.xml#rule_preferences`. Also provides each element with an `Preference.OnPreferenceClickListener` that starts a new `RuleActivity` for the respective `RuleSection`.

Parameters

<i>savedInstanceState</i>	If the fragment is being re-created from a previous saved state, this is the state.
<i>rootKey</i>	If non-null, this preference fragment should be rooted at the <code>androidx.preference.PreferenceScreen</code> with this key.

See also

[#getSectionClickListener\(RuleSection\)](#)

Definition at line 66 of file [SettingsRuleFragment.java](#).

References [SettingsRuleFragment.getSectionClickListener\(\)](#).

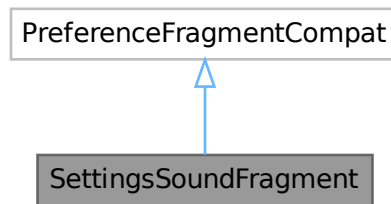
The documentation for this class was generated from the following file:

- [gwent/ui/settings/SettingsRuleFragment.java](#)

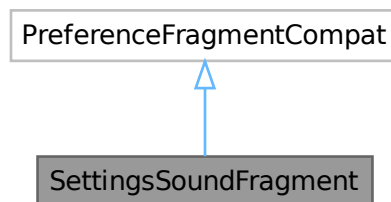
5.53 SettingsSoundFragment Class Reference

A `PreferenceFragmentCompat` class encapsulating the sound preference screen, i.e., the Preferences defined in `R.xml#sound_preferences`.

Inheritance diagram for `SettingsSoundFragment`:



Collaboration diagram for `SettingsSoundFragment`:



Public Member Functions

- void [onCreatePreferences](#) (Bundle savedInstanceState, String rootKey)
Called during `onCreate (Bundle)` to supply the preferences for this fragment.

5.53.1 Detailed Description

A `PreferenceFragmentCompat` class encapsulating the sound preference screen, i.e., the `Preferences` defined in `R.xml#sound_preferences`.

Definition at line 16 of file [SettingsSoundFragment.java](#).

5.53.2 Member Function Documentation

5.53.2.1 onCreatePreferences()

```
void onCreatePreferences (
    Bundle savedInstanceState,
    String rootKey )
```

Called during `onCreate (Bundle)` to supply the preferences for this fragment.

Sets shown `Preferences` from `R.xml#sound_preferences`.

Parameters

<i>savedInstanceState</i>	If the fragment is being re-created from a previous saved state, this is the state.
<i>rootKey</i>	If non-null, this preference fragment should be rooted at the <code>androidx.preference.PreferenceScreen</code> with this key.

Definition at line 27 of file [SettingsSoundFragment.java](#).

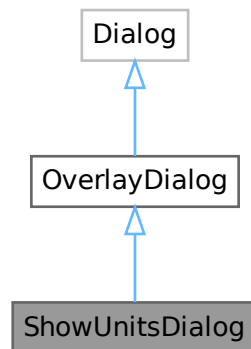
The documentation for this class was generated from the following file:

- gwent/ui/settings/SettingsSoundFragment.java

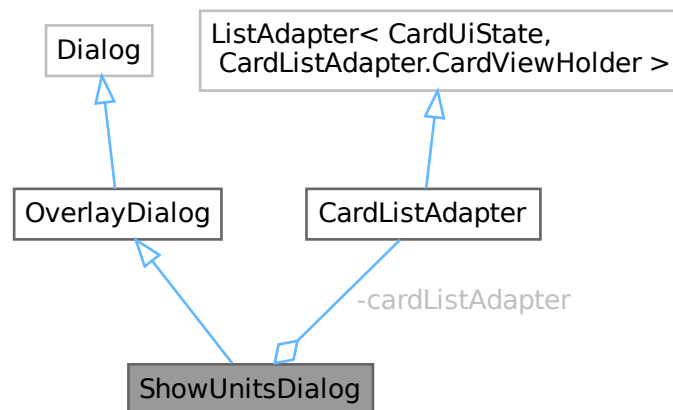
5.54 ShowUnitsDialog Class Reference

An OverlayDialog used to list the units of a certain row and enabling the user to copy, add, or delete new units.

Inheritance diagram for ShowUnitsDialog:



Collaboration diagram for ShowUnitsDialog:



Public Member Functions

- RowType [getRow](#) ()
Returns the row this ShowUnitsDialog is representing.

Static Public Member Functions

- static Single< [ShowUnitsDialog](#) > [getDialog](#) (@NonNull Context context, @NonNull RowType [row](#), @NonNull SoundManager [soundManager](#))

Creates a new ShowUnitsDialog in the given Context and for the given row.

Protected Member Functions

- void [onCreate](#) (Bundle savedInstanceState)

Initializes the RecyclerView by connecting it to the #cardListAdapter and sets View.OnClickListener for each button.

Protected Member Functions inherited from [OverlayDialog](#)

- [OverlayDialog](#) (@NonNull Context context, @LayoutRes int [layout](#), @IdRes int [dismissViewId](#))
- [OverlayDialog](#) (@NonNull Context context, @LayoutRes int [layout](#))

Constructor of an OverlayDialog in the given Context, with the given layout and dismissViewId.

Constructor of an OverlayDialog in the given Context and with the given layout.

Private Member Functions

- [ShowUnitsDialog](#) (@NonNull Context context, @NonNull RowType [row](#), @NonNull [CardListAdapter](#) [cardListAdapter](#), @NonNull SoundManager [soundManager](#))

Constructor of a ShowUnitsDialog shown in the given Context and with the given CardListAdapter.

Private Attributes

- final RowType [row](#)
RowType defining which row is represented by this Dialog.
- final [CardListAdapter](#) [cardListAdapter](#)
CardListAdapter offering a list of CardUiState objects to the UI.
- final SoundManager [soundManager](#)
SoundManager used to play a Sound whenever a card is added or removed.
- final CompositeDisposable [disposables](#) = new CompositeDisposable()
CompositeDisposable keeping track of all subscriptions to observables made by this class.
- RecyclerView [unitRecyclerView](#)
RecyclerView presenting the units provided by the cardListAdapter.

Additional Inherited Members

Static Package Attributes inherited from [OverlayDialog](#)

- static final int [NO_DISMISS_VIEW](#) = -1

Integer used as #dismissViewId, when the Dialog shouldn't be cancelable by clicking on a View.

5.54.1 Detailed Description

An OverlayDialog used to list the units of a certain row and enabling the user to copy, add, or delete new units.

See also

[CardListAdapter](#)

Definition at line 31 of file [ShowUnitsDialog.java](#).

5.54.2 Constructor & Destructor Documentation

5.54.2.1 ShowUnitsDialog()

```
ShowUnitsDialog (
    @NonNull Context context,
    @NonNull RowType row,
    @NonNull CardListAdapter cardListAdapter,
    @NonNull SoundManager soundManager ) [private]
```

Constructor of a ShowUnitsDialog shown in the given Context and with the given CardListAdapter.

Parameters

<i>context</i>	Context this Dialog is shown in.
<i>row</i>	RowType defining which row all shown units belong to.
<i>cardListAdapter</i>	CardListAdapter providing an always up-to-date list of CardUiState objects for a certain row.
<i>soundManager</i>	SoundManager used to play sounds when cards are added or removed.

Definition at line 78 of file [ShowUnitsDialog.java](#).

References [ShowUnitsDialog.cardListAdapter](#), [ShowUnitsDialog.row](#), and [ShowUnitsDialog.soundManager](#).

5.54.3 Member Function Documentation

5.54.3.1 getDialog()

```
static Single< ShowUnitsDialog > getDialog (
    @NonNull Context context,
    @NonNull RowType row,
    @NonNull SoundManager soundManager ) [static]
```

Creates a new ShowUnitsDialog in the given Context and for the given row.

Parameters

<i>context</i>	Context the Dialog is shown in.
<i>row</i>	RowType defining the row that is represented by this Dialog.
<i>soundManager</i>	SoundManager used to play sounds when cards are added or removed.

Returns

A Single emitting the created ShowUnitsDialog.

Definition at line 94 of file [ShowUnitsDialog.java](#).

References [ShowUnitsDialog.disposables](#), [GwentApplication.getRepository\(\)](#), [ShowUnitsDialog.row](#), and [ShowUnitsDialog.soundManager](#).

5.54.3.2 getRow()

```
RowType getRow ( )
```

Returns the row this ShowUnitsDialog is representing.

Returns

A RowType defining the represented row.

See also

[#row](#)

Definition at line 188 of file [ShowUnitsDialog.java](#).

References [ShowUnitsDialog.row](#).

5.54.3.3 onCreate()

```
void onCreate (
    Bundle savedInstanceState ) [protected]
```

Initializes the RecyclerView by connecting it to the #cardListAdapter and sets View.OnClickListener for each button. Also registers a RecyclerView.AdapterDataObserver responsible for scrolling to the end of the RecyclerView whenever an item is being inserted.

See also

`CardListAdapter::registerAdapterDataObserver(RecyclerView.AdapterDataObserver)`

Parameters

<i>savedInstanceState</i>	If this dialog is being reinitialized after the hosting activity was previously shut down, holds the result from the most recent call to <code>onSaveInstanceState</code> , or null if this is the first time.
---------------------------	--

Smoothly scrolls to the last item in the RecyclerView to show the user that an item has been inserted.

Parameters

<i>positionStart</i>	Integer defining the first position from where new items are inserted.
<i>itemCount</i>	Integer defining how many items have been inserted.

See also

`RecyclerView::smoothScrollToPosition(int)`

Reimplemented from [OverlayDialog](#).

Definition at line 142 of file [ShowUnitsDialog.java](#).

References [ShowUnitsDialog.cardListAdapter](#), [ShowUnitsDialog.disposables](#), [ShowUnitsDialog.soundManager](#), and [ShowUnitsDialog.unitRecyclerView](#).

5.54.4 Member Data Documentation

5.54.4.1 cardListAdapter

```
final CardListAdapter cardListAdapter [private]
```

CardListAdapter offering a list of CardUiState objects to the UI.

Definition at line 44 of file [ShowUnitsDialog.java](#).

Referenced by [ShowUnitsDialog.ShowUnitsDialog\(\)](#), and [ShowUnitsDialog.onCreate\(\)](#).

5.54.4.2 disposables

```
final CompositeDisposable disposables = new CompositeDisposable() [private]
```

CompositeDisposable keeping track of all subscriptions to observables made by this class.

Is being disposed in an `android.content.DialogInterface.OnDismissListener` that is set in `#onCreate()`.

See also

`android.content.DialogInterface.OnDismissListener`

`#onCreate(Bundle)`

Definition at line 63 of file [ShowUnitsDialog.java](#).

Referenced by [ShowUnitsDialog.getDialog\(\)](#), and [ShowUnitsDialog.onCreate\(\)](#).

5.54.4.3 row

```
final RowType row [private]
```

RowType defining which row is represented by this Dialog.

See also

`#getRow()`

Definition at line 38 of file [ShowUnitsDialog.java](#).

Referenced by [ShowUnitsDialog.ShowUnitsDialog\(\)](#), [ShowUnitsDialog.getDialog\(\)](#), and [ShowUnitsDialog.getRow\(\)](#).

5.54.4.4 soundManager

```
final SoundManager soundManager [private]
```

SoundManager used to play a Sound whenever a card is added or removed.

See also

`SoundManager::playCardAddSound(RowType, boolean)`

`SoundManager::playCardRemovedSound()`

Definition at line 53 of file [ShowUnitsDialog.java](#).

Referenced by [ShowUnitsDialog.ShowUnitsDialog\(\)](#), [ShowUnitsDialog.getDialog\(\)](#), and [ShowUnitsDialog.onCreate\(\)](#).

5.54.4.5 unitRecyclerView

```
RecyclerView unitRecyclerView [private]
```

RecyclerView presenting the units provided by the cardListAdapter.

Definition at line 68 of file [ShowUnitsDialog.java](#).

Referenced by [ShowUnitsDialog.onCreate\(\)](#).

The documentation for this class was generated from the following file:

- `gwent/ui/dialogs/cards/ShowUnitsDialog.java`

5.55 Sound Class Reference

A data class encapsulating information on a certain sound effect, e.g., its id in a SoundPool and whether it is activated in settings.

Collaboration diagram for Sound:



```

classDiagram
    class Sound
  
```

Package Functions

- String [getPreferenceKey](#) ()
Returns the key where this Sound is activated in the SharedPreferences.
- int [getSoundId](#) ()
Returns the id of this sound effect at the SoundPool.
- boolean [isActivated](#) ()
Returns whether the sound is activated in the settings.
- void [setActivated](#) (boolean [activated](#))
Changes the value of #activated to the given Boolean.
- void [setActivated](#) (@NonNull SharedPreferences [sharedPreferences](#))
Changes the value of #activated based on the given SharedPreferences.

Static Package Functions

- static [Sound createSound](#) (@NonNull Context context, @StringRes int preferenceRes, @NonNull SoundPool soundPool, @RawRes int resId)

Returns a new Sound object encapsulating data for a newly registered sound effect.

Private Member Functions

- [Sound](#) (@NonNull String preferenceKey, int soundId, boolean activated)

Constructor of a Sound encapsulating the given values.

Private Attributes

- final String [preferenceKey](#)

String representing a key of the default SharedPreferences.

- final int [soundId](#)

Integer representing the id of the registered sound at a SoundPool.

- boolean [activated](#)

Boolean defining whether the sound is muted in the settings.

5.55.1 Detailed Description

A data class encapsulating information on a certain sound effect, e.g., its id in a SoundPool and whether it is activated in settings.

Definition at line 20 of file [Sound.java](#).

5.55.2 Constructor & Destructor Documentation

5.55.2.1 Sound()

```
Sound (
    @NonNull String preferenceKey,
    int soundId,
    boolean activated ) [private]
```

Constructor of a Sound encapsulating the given values.

Should only be called by factory method #createSound()

Parameters

<i>preferenceKey</i>	String representing the key where the activated status is defined in SharedPreferences.
<i>soundId</i>	Integer representing the id of the registered sound at a SoundPool.
<i>activated</i>	Boolean defining whether the sound is activated in the settings.

See also

[#createSound\(Context, int, SoundPool, int\)](#)

Definition at line 48 of file [Sound.java](#).

References [Sound.activated](#), [Sound.preferenceKey](#), and [Sound.soundId](#).

5.55.3 Member Function Documentation

5.55.3.1 createSound()

```
static Sound createSound (
    @NonNull Context context,
    @StringRes int preferenceRes,
    @NonNull SoundPool soundPool,
    @RawRes int resId ) [static], [package]
```

Returns a new Sound object encapsulating data for a newly registered sound effect.

Factory method for the Sound class. Gets #`preferenceKey` from the given Context. Registers the sound with the given `resId` at the given SoundPool. Decides the value of #`activated` from SharedPreferences obtained using the given Context.

Parameters

<i>context</i>	Context of the application where this Sound is used.
<i>preferenceRes</i>	Integer referencing the String resource of the preference key for this Sound.
<i>soundPool</i>	SoundPool where the Sound is registered.
<i>resId</i>	Integer referencing the raw resource of the Sound.

Returns

A Sound object referencing the newly registered sound effect.

Definition at line 69 of file [Sound.java](#).

References [Sound.activated](#), [Sound.preferenceKey](#), and [Sound.soundId](#).

Referenced by [SoundManager.SoundManager\(\)](#).

5.55.3.2 getPreferenceKey()

```
String getPreferenceKey ( ) [package]
```

Returns the key where this Sound is activated in the SharedPreferences.

Returns

A String referencing the activation option.

Definition at line 86 of file [Sound.java](#).

References [Sound.preferenceKey](#).

5.55.3.3 getSoundId()

```
int getSoundId ( ) [package]
```

Returns the id of this sound effect at the SoundPool.

Returns

An Integer referencing the sound effect at the SoundPool.

Definition at line 95 of file [Sound.java](#).

References [Sound.soundId](#).

5.55.3.4 isActivated()

```
boolean isActivated ( ) [package]
```

Returns whether the sound is activated in the settings.

Returns

A Boolean defining whether the Sound is activated.

Definition at line 104 of file [Sound.java](#).

References [Sound.activated](#).

5.55.3.5 setActivated() [1/2]

```
void setActivated (
```

```
    @NonNull SharedPreferences sharedPreferences ) [package]
```

Changes the value of #`activated` based on the given SharedPreferences.

Only queries the boolean at key #`preferenceKey`, not whether all sounds are deactivated!

Parameters

<i>sharedPreferences</i>	SharedPreferences containing information on the activated sounds.
--------------------------	---

Definition at line 124 of file [Sound.java](#).

References [Sound.activated](#), [Sound.preferenceKey](#), and [Sound.setActivated\(\)](#).

5.55.3.6 setActivated() [2/2]

```
void setActivated (
    boolean activated ) [package]
```

Changes the value of #activated to the given Boolean.

Parameters

<i>activated</i>	Boolean defining whether the Sound is activated.
------------------	--

See also

[#setActivated\(SharedPreferences\)](#)

Definition at line 114 of file [Sound.java](#).

References [Sound.activated](#).

Referenced by [Sound.setActivated\(\)](#).

5.55.4 Member Data Documentation**5.55.4.1 activated**

```
boolean activated [private]
```

Boolean defining whether the sound is muted in the settings.

Definition at line 37 of file [Sound.java](#).

Referenced by [Sound.Sound\(\)](#), [Sound.createSound\(\)](#), [Sound.isActivated\(\)](#), [Sound.setActivated\(\)](#), and [Sound.setActivated\(\)](#).

5.55.4.2 preferenceKey

```
final String preferenceKey [private]
```

String representing a key of the default SharedPreferences.

The referenced preference decides whether the sound is activated (assuming that sounds are not completely disabled).

Definition at line 27 of file [Sound.java](#).

Referenced by [Sound.Sound\(\)](#), [Sound.createSound\(\)](#), [Sound.getPreferenceKey\(\)](#), and [Sound.setActivated\(\)](#).

5.55.4.3 soundId

```
final int soundId [private]
```

Integer representing the id of the registered sound at a SoundPool.

Definition at line 32 of file [Sound.java](#).

Referenced by [Sound.Sound\(\)](#), [Sound.createSound\(\)](#), and [Sound.getSoundId\(\)](#).

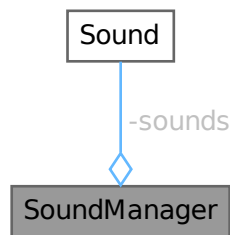
The documentation for this class was generated from the following file:

- [gwent/ui/sounds/Sound.java](#)

5.56 SoundManager Class Reference

A class responsible for initializing and playing sounds when they are enabled.

Collaboration diagram for SoundManager:



Public Member Functions

- `SoundManager` (@NonNull Context context)
Constructor of a SoundManager in the given Context.
- void `playSound` (@IntRange(from=`SOUND_WEATHER_GOOD`, to=`SOUND_COIN`) int soundId)
Plays the given Sound if it is #activated.
- void `playClearWeatherSound` ()
Plays the clear weather sound.
- void `playWeatherSound` (@NonNull RowType row)
Plays the weather sound of the given row.
- void `playHornSound` ()
Plays the horn sound.
- void `playCardAddSound` (@NonNull RowType row, boolean epic)
Plays the card add sound for the given row.
- void `playCardRemovedSound` ()
Plays the reset sound.
- void `playResetSound` ()
Plays the reset sound.
- void `playBurnSound` ()
Plays the burn sound.
- void `playCoinSound` ()
Plays the coin sound.

Static Public Attributes

- static final int `SOUND_WEATHER_GOOD` = 0
Constant Integer representing the clear weather sound.
- static final int `SOUND_WEATHER_FROST` = 1
Constant Integer representing the frost weather sound.
- static final int `SOUND_WEATHER_FOG` = 2
Constant Integer representing the fog weather sound.
- static final int `SOUND_WEATHER_RAIN` = 3
Constant Integer representing the rain weather sound.
- static final int `SOUND_HORN` = 4
Constant Integer representing the horn sound.
- static final int `SOUND_CARDS_EPIC` = 5

- *Constant Integer representing the epic unit sound.*
static final int [SOUND_CARDS_MELEE](#) = 6
- *Constant Integer representing the melee unit sound.*
static final int [SOUND_CARDS_RANGE](#) = 7
- *Constant Integer representing the range unit sound.*
static final int [SOUND_CARDS_SIEGE](#) = 8
- *Constant Integer representing the siege unit sound.*
static final int [SOUND_RESET](#) = 9
- *Constant Integer representing the reset sound.*
static final int [SOUND_BURN](#) = 10
- *Constant Integer representing the scorch sound.*
static final int [SOUND_COIN](#) = 11
- *Constant Integer representing the coin-flip sound.*

Private Attributes

- final [Sound](#)[] [sounds](#) = new [Sound](#)[[SOUND_COIN](#) + 1]
Array of Sound objects representing the different sound effects.
- final [SoundPool](#) [soundPool](#)
SoundPool where sound effects are registered.
- final [SharedPreferences.OnSharedPreferenceChangeListener](#) [changeListener](#)
OnSharedPreferenceChangeListener that is registered for the default SharedPreferences and updates the Sound↔::active attributes of all #sounds when a preference change occurs.

5.56.1 Detailed Description

A class responsible for initializing and playing sounds when they are enabled.

See also

[Sound](#)

Definition at line 23 of file [SoundManager.java](#).

5.56.2 Constructor & Destructor Documentation

5.56.2.1 SoundManager()

```
SoundManager (
    @NonNull Context context )
```

Constructor of a SoundManager in the given Context.

Creates a new #soundPool and registers #sounds using [Sound::createSound\(\)](#). Also registers a new #change↔Listener that updates the [Sound::activated](#) status when sound settings are updated.

Parameters

<i>context</i>	Context, this SoundManager is used in.
----------------	--

See also

[Sound::createSound\(Context, int, SoundPool, int\)](#)

Definition at line 156 of file [SoundManager.java](#).

References [SoundManager.changeListener](#), [Sound.createSound\(\)](#), [SoundManager.SOUND_BURN](#), [SoundManager.SOUND_CARDS_MELEE](#), [SoundManager.SOUND_CARDS_RANGE](#), [SoundManager.SOUND_CARDS_SIEGE](#), [SoundManager.SOUND_COIN](#), [SoundManager.SOUND_HORN](#), [SoundManager.SOUND_RESET](#), [SoundManager.SOUND_WEATHER_FROST](#), [SoundManager.SOUND_WEATHER_GOOD](#), [SoundManager.SOUND_WEATHER_RA](#), [SoundManager.soundPool](#), and [SoundManager.sounds](#).

5.56.3 Member Function Documentation

5.56.3.1 playBurnSound()

```
void playBurnSound ( )
```

Plays the burn sound.

Wrapper for #playSound().

See also

#playSound(int)

Definition at line 311 of file [SoundManager.java](#).

References [SoundManager.playSound\(\)](#), and [SoundManager.SOUND_BURN](#).

5.56.3.2 playCardAddSound()

```
void playCardAddSound (
    @NonNull RowType row,
    boolean epic )
```

Plays the card add sound for the given row.

If the unit is epic, the epic sound is played. Wrapper for #playSound().

Parameters

<i>row</i>	RowType referencing the row for which the sound should be played.
<i>epic</i>	Boolean defining whether the added unit is epic.

See also

#playSound(int)

Definition at line 268 of file [SoundManager.java](#).

References [SoundManager.playSound\(\)](#), [SoundManager.SOUND_CARDS_EPIC](#), [SoundManager.SOUND_CARDS_MELEE](#), [SoundManager.SOUND_CARDS_RANGE](#), and [SoundManager.SOUND_CARDS_SIEGE](#).

5.56.3.3 playCardRemovedSound()

```
void playCardRemovedSound ( )
```

Plays the reset sound.

Wrapper for #playSound().

See also

#playSound(int)

Definition at line 291 of file [SoundManager.java](#).

References [SoundManager.playSound\(\)](#), and [SoundManager.SOUND_RESET](#).

5.56.3.4 playClearWeatherSound()

```
void playClearWeatherSound ( )
```

Plays the clear weather sound.

Wrapper for #playSound().

See also

#playSound(int)

Definition at line 225 of file [SoundManager.java](#).

References [SoundManager.playSound\(\)](#), and [SoundManager.SOUND_WEATHER_GOOD](#).

5.56.3.5 playCoinSound()

```
void playCoinSound ( )
```

Plays the coin sound.

Wrapper for #playSound().

See also

#playSound(int)

Definition at line 320 of file [SoundManager.java](#).

References [SoundManager.playSound\(\)](#), and [SoundManager.SOUND_COIN](#).

5.56.3.6 playHornSound()

```
void playHornSound ( )
```

Plays the horn sound.

Wrapper for #playSound().

See also

#playSound(int)

Definition at line 255 of file [SoundManager.java](#).

References [SoundManager.playSound\(\)](#), and [SoundManager.SOUND_HORN](#).

5.56.3.7 playResetSound()

```
void playResetSound ( )
```

Plays the reset sound.

Wrapper for #playSound().

See also

#playSound(int)

Definition at line 301 of file [SoundManager.java](#).

References [SoundManager.playSound\(\)](#), and [SoundManager.SOUND_RESET](#).

5.56.3.8 playSound()

```
void playSound (
```

```
    @IntRange(from=SOUND\_WEATHER\_GOOD, to=SOUND\_COIN) int soundId )
```

Plays the given Sound if it is #activated.

Given soundId must be either #SOUND_WEATHER_GOOD, #SOUND_WEATHER_FROST, #SOUND_WEATHER_FOG, #SOUND_WEATHER_RAIN, #SOUND_HORN, #SOUND_CARDS_EPIC, #SOUND_CARDS_MELEE, #SOUND_CARDS_RANGE, #SOUND_CARDS_SIEGE, #SOUND_RESET, #SOUND_BURN, or #SOUND_COIN, i.e., $0 \leq \text{soundId} \leq 11$ must be true. Alternately, #playClearWeatherSound(), #playWeatherSound(), #playHornSound(), #playCardAddSound(), #playCardRemovedSound(), #playResetSound(), #playBurnSound(), or #playCoinSound() may be used.

Parameters

<i>soundId</i>	Integer representing the Sound that should be played.
----------------	---

Exceptions

<i>IndexOutOfBoundsException</i>	When $0 \leq \text{soundId} \leq 11$ is not true.
----------------------------------	---

Definition at line 213 of file [SoundManager.java](#).

References [SoundManager.soundPool](#), and [SoundManager.sounds](#).

Referenced by [SoundManager.playBurnSound\(\)](#), [SoundManager.playCardAddSound\(\)](#), [SoundManager.playCardRemovedSound\(\)](#), [SoundManager.playClearWeatherSound\(\)](#), [SoundManager.playCoinSound\(\)](#), [SoundManager.playHornSound\(\)](#), [SoundManager.playResetSound\(\)](#), and [SoundManager.playWeatherSound\(\)](#).

5.56.3.9 playWeatherSound()

```
void playWeatherSound (
    @NonNull RowType row )
```

Plays the weather sound of the given row.

Wrapper for [#playSound\(\)](#).

Parameters

<i>row</i>	RowType referencing the row for which the sound should be played.
------------	---

See also

[#playSound\(int\)](#)

Definition at line 236 of file [SoundManager.java](#).

References [SoundManager.playSound\(\)](#), [SoundManager.SOUND_WEATHER_FOG](#), [SoundManager.SOUND_WEATHER_FROST](#), and [SoundManager.SOUND_WEATHER_RAIN](#).

5.56.4 Member Data Documentation

5.56.4.1 changeListener

```
final SharedPreferences.OnSharedPreferenceChangeListener changeListener [private]
```

OnSharedPreferenceChangeListener that is registered for the default SharedPreferences and updates the Sound↔::active attributes of all #sounds when a preference change occurs.

Reference must be kept (even if not used) to avoid garbage collection of the registered listener (see [here](#) for more information).

Definition at line 145 of file [SoundManager.java](#).

Referenced by [SoundManager.SoundManager\(\)](#).

5.56.4.2 SOUND_BURN

```
final int SOUND_BURN = 10 [static]
```

Constant Integer representing the scorch sound.

Index of the respective Sound object in #sounds.

See also

[#playBurnSound\(\)](#)

Definition at line 110 of file [SoundManager.java](#).

Referenced by [SoundManager.SoundManager\(\)](#), and [SoundManager.playBurnSound\(\)](#).

5.56.4.3 SOUND_CARDS_EPIC

```
final int SOUND_CARDS_EPIC = 5 [static]
```

Constant Integer representing the epic unit sound.

Index of the respective Sound object in #sounds.

See also

[#playCardAddSound\(RowType, boolean\)](#)

Definition at line 70 of file [SoundManager.java](#).

Referenced by [SoundManager.SoundManager\(\)](#), and [SoundManager.playCardAddSound\(\)](#).

5.56.4.4 SOUND_CARDS_MELEE

```
final int SOUND_CARDS_MELEE = 6 [static]
```

Constant Integer representing the melee unit sound.

Index of the respective Sound object in #sounds.

See also

`#playCardAddSound(RowType, boolean)`

Definition at line 78 of file [SoundManager.java](#).

Referenced by [SoundManager.SoundManager\(\)](#), and [SoundManager.playCardAddSound\(\)](#).

5.56.4.5 SOUND_CARDS_RANGE

```
final int SOUND_CARDS_RANGE = 7 [static]
```

Constant Integer representing the range unit sound.

Index of the respective Sound object in #sounds.

See also

`#playCardAddSound(RowType, boolean)`

Definition at line 86 of file [SoundManager.java](#).

Referenced by [SoundManager.SoundManager\(\)](#), and [SoundManager.playCardAddSound\(\)](#).

5.56.4.6 SOUND_CARDS_SIEGE

```
final int SOUND_CARDS_SIEGE = 8 [static]
```

Constant Integer representing the siege unit sound.

Index of the respective Sound object in #sounds.

See also

`#playCardAddSound(RowType, boolean)`

Definition at line 94 of file [SoundManager.java](#).

Referenced by [SoundManager.SoundManager\(\)](#), and [SoundManager.playCardAddSound\(\)](#).

5.56.4.7 SOUND_COIN

```
final int SOUND_COIN = 11 [static]
```

Constant Integer representing the coin-flip sound.

Index of the respective Sound object in #sounds.

See also

`#playCoinSound()`

Definition at line 118 of file [SoundManager.java](#).

Referenced by [SoundManager.SoundManager\(\)](#), and [SoundManager.playCoinSound\(\)](#).

5.56.4.8 SOUND_HORN

```
final int SOUND_HORN = 4 [static]
```

Constant Integer representing the horn sound.

Index of the respective Sound object in #sounds.

See also

`#playHornSound()`

Definition at line 62 of file [SoundManager.java](#).

Referenced by [SoundManager.SoundManager\(\)](#), and [SoundManager.playHornSound\(\)](#).

5.56.4.9 SOUND_RESET

```
final int SOUND_RESET = 9 [static]
```

Constant Integer representing the reset sound.

Index of the respective Sound object in #sounds.

See also

`#playResetSound()`

Definition at line 102 of file [SoundManager.java](#).

Referenced by [SoundManager.SoundManager\(\)](#), [SoundManager.playCardRemovedSound\(\)](#), and [SoundManager.playResetSound\(\)](#).

5.56.4.10 SOUND_WEATHER_FOG

```
final int SOUND_WEATHER_FOG = 2 [static]
```

Constant Integer representing the fog weather sound.

Index of the respective Sound object in #sounds.

See also

`#playCardAddSound(RowType, boolean)`

Definition at line 46 of file [SoundManager.java](#).

Referenced by [SoundManager.SoundManager\(\)](#), and [SoundManager.playWeatherSound\(\)](#).

5.56.4.11 SOUND_WEATHER_FROST

```
final int SOUND_WEATHER_FROST = 1 [static]
```

Constant Integer representing the frost weather sound.

Index of the respective Sound object in #sounds.

See also

`#playCardAddSound(RowType, boolean)`

Definition at line 38 of file [SoundManager.java](#).

Referenced by [SoundManager.SoundManager\(\)](#), and [SoundManager.playWeatherSound\(\)](#).

5.56.4.12 SOUND_WEATHER_GOOD

```
final int SOUND_WEATHER_GOOD = 0 [static]
```

Constant Integer representing the clear weather sound.

Index of the respective Sound object in #sounds.

See also

`#playClearWeatherSound()`

Definition at line 30 of file [SoundManager.java](#).

Referenced by [SoundManager.SoundManager\(\)](#), and [SoundManager.playClearWeatherSound\(\)](#).

5.56.4.13 SOUND_WEATHER_RAIN

```
final int SOUND_WEATHER_RAIN = 3 [static]
```

Constant Integer representing the rain weather sound.

Index of the respective Sound object in #sounds.

See also

`#playCardAddSound(RowType, boolean)`

Definition at line 54 of file [SoundManager.java](#).

Referenced by [SoundManager.SoundManager\(\)](#), and [SoundManager.playWeatherSound\(\)](#).

5.56.4.14 soundPool

```
final SoundPool soundPool [private]
```

SoundPool where sound effects are registered.

Definition at line 134 of file [SoundManager.java](#).

Referenced by [SoundManager.SoundManager\(\)](#), and [SoundManager.playSound\(\)](#).

5.56.4.15 sounds

```
final Sound [] sounds = new Sound[SOUND_COIN + 1] [private]
```

Array of Sound objects representing the different sound effects.

Indices are the public constants defined here, i.e., #SOUND_WEATHER_GOOD, #SOUND_WEATHER_FROST, #SOUND_WEATHER_FOG, #SOUND_WEATHER_RAIN, #SOUND_HORN, #SOUND_CARDS_EPIC, #SOUND_CARDS_MELEE, #SOUND_CARDS_RANGE, #SOUND_CARDS_SIEGE, #SOUND_RESET, #SOUND_BURN, and #SOUND_COIN.

Definition at line 128 of file [SoundManager.java](#).

Referenced by [SoundManager.SoundManager\(\)](#), and [SoundManager.playSound\(\)](#).

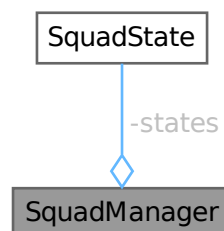
The documentation for this class was generated from the following file:

- gwent/ui/sounds/SoundManager.java

5.57 SquadManager Class Reference

A class used by the CardNumberPickerAdapter to determine the first squad with members or set the default damage according to the squads base damage.

Collaboration diagram for SquadManager:



Static Public Attributes

- static final int [MAX_NR_SQUADS](#) = 3
Integer constant defining how many different squads there may be per row.

Package Functions

- [SquadManager](#) (@NonNull List< UnitEntity > units)
Constructor of a new SquadManager storing information on the squads of the given units.
- int [getFirstSquadWithMembers](#) ()
Returns the lowest squad number referring to a squad that has members.
- void [onSquadChanged](#) (@IntRange(from=1, to=[MAX_NR_SQUADS](#)) int newVal, @NonNull [DamageValuePicker](#) picker)
Shows a Toast with information about the selected squad and sets the given picker to the SquadState::memberBaseDamage of said squad.

Private Attributes

- final [SquadState](#)[] `states` = new [SquadState](#)[[MAX_NR_SQUADS](#)]
Array of SquadState objects containing one SquadState for each possible squad.

5.57.1 Detailed Description

A class used by the [CardNumberPickerAdapter](#) to determine the first squad with members or set the default damage according to the squads base damage.

See also

[CardNumberPickerAdapter](#)

Definition at line 24 of file [SquadManager.java](#).

5.57.2 Constructor & Destructor Documentation**5.57.2.1 SquadManager()**

```
SquadManager (
    @NonNull List< UnitEntity > units ) [package]
```

Constructor of a new [SquadManager](#) storing information on the squads of the given units.

Parameters

<i>units</i>	List of UnitEntity objects that is used to fetch squad information.
--------------	---

Definition at line 41 of file [SquadManager.java](#).

References [SquadState.getState\(\)](#), [SquadManager.MAX_NR_SQUADS](#), and [SquadManager.states](#).

5.57.3 Member Function Documentation**5.57.3.1 getFirstSquadWithMembers()**

```
int getFirstSquadWithMembers ( ) [package]
```

Returns the lowest squad number referring to a squad that has members.

If no squad has members, i.e., there are no units with the [Ability::BINDING](#) ability, 1 is returned.

Returns

An Integer referencing to the first squad with members.

Definition at line 55 of file [SquadManager.java](#).

References [SquadState.getSquadNumber\(\)](#), [SquadState.hasMembers\(\)](#), and [SquadManager.states](#).

Referenced by [CardNumberPickerAdapter.CardNumberPickerAdapter\(\)](#), and [CardNumberPickerAdapter.addSelectedUnits\(\)](#).

5.57.3.2 onSquadChanged()

```
void onSquadChanged (
    @IntRange(from=1, to=MAX\_NR\_SQUADS) int newVal,
    @NonNull DamageValuePicker picker ) [package]
```

Shows a Toast with information about the selected squad and sets the given picker to the [SquadState::member↔BaseDamage](#) of said squad.

Parameters

<i>newVal</i>	Integer representing the newly selected squad.
<i>picker</i>	DamageValuePicker that is updated.

Exceptions

<code>IllegalArgumentException</code>	When newVal is not between 1 and #MAX_NR_SQUADS.
---------------------------------------	--

See also

`SquadState::getMemberBaseDamage()`

Definition at line 72 of file [SquadManager.java](#).

References [SquadState.getMemberBaseDamage\(\)](#), [SquadState.getSquadMembers\(\)](#), [SquadState.getSquadNumber\(\)](#), [SquadState.hasMembers\(\)](#), [SquadManager.MAX_NR_SQUADS](#), and [SquadManager.states](#).

Referenced by [CardNumberPickerAdapter.CardNumberPickerAdapter\(\)](#).

5.57.4 Member Data Documentation

5.57.4.1 MAX_NR_SQUADS

```
final int MAX_NR_SQUADS = 3 [static]
```

Integer constant defining how many different squads there may be per row.

Definition at line 28 of file [SquadManager.java](#).

Referenced by [CardNumberPickerAdapter.CardNumberPickerAdapter\(\)](#), [SquadManager.SquadManager\(\)](#), and [SquadManager.onSquadChanged\(\)](#).

5.57.4.2 states

```
final SquadState [] states = new SquadState[MAX_NR_SQUADS] [private]
```

Array of SquadState objects containing one SquadState for each possible squad.

Definition at line 34 of file [SquadManager.java](#).

Referenced by [SquadManager.SquadManager\(\)](#), [SquadManager.getFirstSquadWithMembers\(\)](#), and [SquadManager.onSquadChange](#)

The documentation for this class was generated from the following file:

- [gwent/ui/dialogs/addcard/SquadManager.java](#)

5.58 SquadState Class Reference

A data class encapsulating information about the squad defined by #squadNumber, i.e., how many #squadMembers are there and what is the #memberBaseDamage of units in this squad.

Collaboration diagram for SquadState:

SquadState

Package Functions

- boolean [hasMembers](#) ()
Returns whether the represented squad has members, i.e., whether #squadMembers is greater than 0.
- int [getSquadNumber](#) ()
Returns the number of the represented squad.
- int [getSquadMembers](#) ()
Returns the number of members in the represented squad.
- int [getMemberBaseDamage](#) ()
Returns the base damage of members in the represented squad.

Static Package Functions

- static [SquadState getState](#) (@IntRange(from=1, to=MAX_NR_SQUADS) int [squadNumber](#), @NonNull List<UnitEntity> units)

Creates a new SquadState representing the squad with the given #squadNumber.

Private Member Functions

- [SquadState](#) (@IntRange(from=1, to=MAX_NR_SQUADS) int [squadNumber](#), @IntRange(from=0) int [squadMembers](#), @IntRange(from=0, to=NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) int [memberBaseDamage](#))

Constructor of a SquadState with the given #squadNumber, #squadMembers, and #memberBaseDamage.

Private Attributes

- final int [squadNumber](#)
Integer containing the number of the represented squad.
- final int [squadMembers](#)
Integer representing the number of members in the squad.
- final int [memberBaseDamage](#)
Integer containing the base damage of members of this squad.

5.58.1 Detailed Description

A data class encapsulating information about the squad defined by #squadNumber, i.e., how many #squadMembers are there and what is the #memberBaseDamage of units in this squad.

See also

[SquadManager](#)

Definition at line 19 of file [SquadState.java](#).

5.58.2 Constructor & Destructor Documentation

5.58.2.1 SquadState()

```
SquadState (
    @IntRange(from=1, to=MAX_NR_SQUADS) int squadNumber,
    @IntRange(from=0) int squadMembers,
    @IntRange(from=0, to=NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) int memberBaseDamage )
[private]
```

Constructor of a SquadState with the given #squadNumber, #squadMembers, and #memberBaseDamage. Should only be used by #getState().

Parameters

<i>squadNumber</i>	Integer containing the number of the represented squad.
<i>squadMembers</i>	Integer representing the number of members in the squad.
<i>memberBaseDamage</i>	Integer containing the base damage of members of this squad.

See also

[#getState\(int, List\)](#)

Exceptions

<i>IllegalArgumentException</i>	When one of the parameters doesn't meet its IntRange constraint.
---------------------------------	--

Definition at line 55 of file [SquadState.java](#).

References [SquadState.memberBaseDamage](#), [SquadState.squadMembers](#), and [SquadState.squadNumber](#).

5.58.3 Member Function Documentation

5.58.3.1 `getMemberBaseDamage()`

```
int getMemberBaseDamage ( ) [package]
```

Returns the base damage of members in the represented squad.

See also

`#memberBaseDamage`

Returns

An Integer defining the base damage of units in the represented squad.

Definition at line 127 of file [SquadState.java](#).

References [SquadState.memberBaseDamage](#).

Referenced by [SquadManager.onSquadChanged\(\)](#).

5.58.3.2 `getSquadMembers()`

```
int getSquadMembers ( ) [package]
```

Returns the number of members in the represented squad.

See also

`#squadMembers`

Returns

An Integer defining how many members are in the represented squad.

Definition at line 117 of file [SquadState.java](#).

References [SquadState.squadMembers](#).

Referenced by [SquadManager.onSquadChanged\(\)](#).

5.58.3.3 `getSquadNumber()`

```
int getSquadNumber ( ) [package]
```

Returns the number of the represented squad.

See also

`#squadNumber`

Returns

An Integer defining which squad is represented.

Definition at line 107 of file [SquadState.java](#).

References [SquadState.squadNumber](#).

Referenced by [SquadManager.getFirstSquadWithMembers\(\)](#), and [SquadManager.onSquadChanged\(\)](#).

5.58.3.4 `getState()`

```
static SquadState getState (
    @IntRange(from=1, to=MAX_NR_SQUADS) int squadNumber,
    @NonNull List< UnitEntity > units ) [static], [package]
```

Creates a new [SquadState](#) representing the squad with the given `#squadNumber`.

Information for `#squadMembers` and `#memberBaseDamage` is retrieved from the given List of [UnitEntity](#) objects.

Parameters

<i>squadNumber</i>	Integer containing the number of the represented squad.
<i>units</i>	List of UnitEntity objects used to count squad-members.

Returns

A SquadState object that is newly created from the given List of units.

Exceptions

<i>IllegalArgumentException</i>	When the given squad number is not between 1 and #MAX_NR_SQUADS.
---------------------------------	--

Definition at line 80 of file [SquadState.java](#).

References [SquadState.squadNumber](#).

Referenced by [SquadManager.SquadManager\(\)](#).

5.58.3.5 hasMembers()

```
boolean hasMembers ( ) [package]
```

Returns whether the represented squad has members, i.e., whether #squadMembers is greater than 0.

Returns

A Boolean defining whether the represented squad has members or not.

Definition at line 97 of file [SquadState.java](#).

References [SquadState.squadMembers](#).

Referenced by [SquadManager.getFirstSquadWithMembers\(\)](#), and [SquadManager.onSquadChanged\(\)](#).

5.58.4 Member Data Documentation

5.58.4.1 memberBaseDamage

```
final int memberBaseDamage [private]
```

Integer containing the base damage of members of this squad.

If members have different UnitEntity::damage values, it is undefined which of them is picked. If #squadMembers is 0, #memberBaseDamage defaults to 5.

See also

[#getMemberBaseDamage\(\)](#)

Definition at line 43 of file [SquadState.java](#).

Referenced by [SquadState.SquadState\(\)](#), and [SquadState.getMemberBaseDamage\(\)](#).

5.58.4.2 squadMembers

```
final int squadMembers [private]
```

Integer representing the number of members in the squad.

See also

[#getSquadMembers\(\)](#)

Definition at line 34 of file [SquadState.java](#).

Referenced by [SquadState.SquadState\(\)](#), [SquadState.getSquadMembers\(\)](#), and [SquadState.hasMembers\(\)](#).

5.58.4.3 squadNumber

```
final int squadNumber [private]
```

Integer containing the number of the represented squad.

Equivalent to `UnitEntity::squad` of the units in the squad.

See also

`#getSquadNumber()`

Definition at line 27 of file [SquadState.java](#).

Referenced by [SquadState.SquadState\(\)](#), [SquadState.getSquadNumber\(\)](#), and [SquadState.getState\(\)](#).

The documentation for this class was generated from the following file:

- `gwent/ui/dialogs/addcard/SquadState.java`

5.59 StringValuePicker< T extends Comparable Class Template Reference

A `ValuePicker` displaying a resource `String` for each element in `selectableValues`.

Collaboration diagram for `StringValuePicker< T extends Comparable`:

```
StringValuePicker<
T extends Comparable
```

5.59.1 Detailed Description

A `ValuePicker` displaying a resource `String` for each element in `selectableValues`.

The resource `String` is retrieved from `displayIntegers` by using the associated `Integer` as the `String` id in `#get↔DisplayString()`.

See also

`#getDisplayString(Comparable)`

Definition at line 17 of file [StringValuePicker.java](#).

The documentation for this class was generated from the following file:

- `gwent/ui/dialogs/addcard/StringValuePicker.java`

5.60 ResetDialogUseCase.Trigger Enum Reference

An `Enum` listing the possible triggers of a reset.

Collaboration diagram for `ResetDialogUseCase.Trigger`:

```
ResetDialogUseCase.Trigger
```

Public Attributes

- [BUTTON_CLICK](#)
Represents, that a reset was triggered by a click on the reset button.
- [FACTION_SWITCH](#)
Represents that a reset was triggered by a faction switch.

5.60.1 Detailed Description

An `Enum` listing the possible triggers of a reset.

Definition at line 130 of file [ResetDialogUseCase.java](#).

5.60.2 Member Data Documentation**5.60.2.1 BUTTON_CLICK**

`BUTTON_CLICK`

Represents, that a reset was triggered by a click on the reset button.

Definition at line 134 of file [ResetDialogUseCase.java](#).

5.60.2.2 FACTION_SWITCH

`FACTION_SWITCH`

Represents that a reset was triggered by a faction switch.

Only relevant if faction reset is activated, i.e., the preference at the key referenced by `R.string#preference_key_faction_r` is true.

Definition at line 141 of file [ResetDialogUseCase.java](#).

Referenced by [ResetAlertDialogBuilderAdapter.setTrigger\(\)](#).

The documentation for this enum was generated from the following file:

- gwent/domain/cases/ResetDialogUseCase.java

5.61 UnitDao Interface Reference

An interface defining data access operations related to `units` table.

Collaboration diagram for UnitDao:



```

classDiagram
    class UnitDao
  
```

Public Member Functions

- Completable [insertUnit](#) (@NonNull [UnitEntity](#) unit)
Inserts the given UnitEntity into units.
- Completable [insertUnit](#) (boolean epic, @IntRange(from=0, to=[UnitEntity.NON_EPIC_DAMAGE_VALUES_UPPER_BOUND](#)) int damage, @NonNull [Ability](#) ability, @IntRange(from=1) @Nullable Integer squad, @NonNull [RowType](#) row)
Inserts a UnitEntity with the given values into units.
- Completable [deleteUnit](#) (int id)
Deletes the UnitEntity object with the given id from units.
- Completable [deleteUnits](#) (@NonNull Collection< [UnitEntity](#) > units)

- Deletes the given UnitEntity objects from units.*
- `Single< UnitEntity > getUnit (int id)`
Fetches the UnitEntity object with the given id from units.
- `Single< List< UnitEntity > > getUnits (@NonNull RowType row)`
Fetches all UnitEntity objects from units in the given row.
- `Flowable< List< UnitEntity > > getUnitsFlowable (@NonNull RowType row)`
Fetches a Flowable for all UnitEntity objects from units in the given row.
- `Single< List< UnitEntity > > getUnits ()`
Fetches all UnitEntity objects from units.
- `Single< Integer > countUnits (@NonNull RowType row)`
Counts UnitEntity objects in units in the given row.
- `Single< Integer > countUnits ()`
Counts UnitEntity objects in units.
- `Flowable< Boolean > hasNonEpicUnitsFlowable ()`
Fetches a Flowable of Booleans defining whether units contains units that are not epic.

5.61.1 Detailed Description

An interface defining data access operations related to `units` table.
The implementation is generated by the Room framework.
Definition at line 25 of file `UnitDao.java`.

5.61.2 Member Function Documentation

5.61.2.1 `countUnits()` [1/2]

```
Single< Integer > countUnits ( )
```

Counts UnitEntity objects in units.

Returns

A Single tracking operation status and returning the value.

5.61.2.2 `countUnits()` [2/2]

```
Single< Integer > countUnits (
    @NonNull RowType row )
```

Counts UnitEntity objects in units in the given row.

Parameters

<i>row</i>	RowType defining the UnitEntity::row foreign key.
------------	---

Returns

A Single tracking operation status and returning the value.

Referenced by `UnitRepository.countUnits()`, and `UnitRepository.countUnits()`.

5.61.2.3 `deleteUnit()`

```
Completable deleteUnit (
    int id )
```

Deletes the UnitEntity object with the given id from units.

Parameters

<i>id</i>	Integer representing the <code>UnitEntity::id</code> of the deleted <code>UnitEntity</code> .
-----------	---

Returns

A `Completable` tracking operation status.

Referenced by [UnitRepository.delete\(\)](#).

5.61.2.4 deleteUnits()

```
Completable deleteUnits (
    @NonNull Collection< UnitEntity > units )
```

Deletes the given `UnitEntity` objects from `units`.

Parameters

<i>units</i>	List of <code>UnitEntity</code> that are being inserted.
--------------	--

Returns

A `Completable` tracking operation status.

Referenced by [UnitRepository.delete\(\)](#).

5.61.2.5 getUnit()

```
Single< UnitEntity > getUnit (
    int id )
```

Fetches the `UnitEntity` object with the given `id` from `units`.

Parameters

<i>id</i>	Integer representing the <code>UnitEntity::id</code> of the requested <code>UnitEntity</code> .
-----------	---

Returns

A `Single` tracking operation status and returning the value.

Referenced by [UnitRepository.getUnit\(\)](#).

5.61.2.6 getUnits() [1/2]

```
Single< List< UnitEntity > > getUnits ( )
```

Fetches all `UnitEntity` objects from `units`.

Returns

A `Single` tracking operation status and returning the value.

5.61.2.7 getUnits() [2/2]

```
Single< List< UnitEntity > > getUnits (
    @NonNull RowType row )
```

Fetches all `UnitEntity` objects from `units` in the given row.

See also

[#getUnitsFlowable\(RowType\)](#)

Parameters

<i>row</i>	RowType defining the UnitEntity::row foreign key.
------------	---

Returns

A Single tracking operation status and returning the value.

Referenced by [UnitRepository.getUnits\(\)](#), and [UnitRepository.getUnits\(\)](#).

5.61.2.8 getUnitsFlowable()

```
Flowable< List< UnitEntity > > getUnitsFlowable (
    @NonNull RowType row )
```

Fetches a Flowable for all UnitEntity objects from units in the given row.

Parameters

<i>row</i>	RowType defining the UnitEntity::row foreign key.
------------	---

Returns

A Flowable emitting the values.

See also

[#getUnits\(RowType\)](#)

Referenced by [UnitRepository.getUnitsFlowable\(\)](#).

5.61.2.9 hasNonEpicUnitsFlowable()

```
Flowable< Boolean > hasNonEpicUnitsFlowable ( )
```

Fetches a Flowable of Booleans defining whether units contains units that are not epic.

Returns

A Flowable emitting the values.

Referenced by [UnitRepository.hasNonEpicUnitsFlowable\(\)](#).

5.61.2.10 insertUnit() [1/2]

```
Completable insertUnit (
    @NonNull UnitEntity unit )
```

Inserts the given UnitEntity into units.

If the same UnitEntity::id is already in the table, the new insert is ignored.

Parameters

<i>unit</i>	UnitEntity that is being inserted.
-------------	------------------------------------

Returns

A Completable tracking operation status.

Referenced by [UnitRepository.insertUnit\(\)](#), and [UnitRepository.reset\(\)](#).

5.61.2.11 insertUnit() [2/2]

```
Completable insertUnit (
```

```

    boolean epic,
    @IntRange(from=0, to=UnitEntity.NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) int damage,
    @NonNull Ability ability,
    @IntRange(from=1) @Nullable Integer squad,
    @NonNull RowType row )

```

Inserts a UnitEntity with the given values into units.

Parameters

<i>epic</i>	Boolean defining whether the unit is epic.
<i>damage</i>	Non-Negative value defining the base-damage of the unit.
<i>ability</i>	Ability defining the ability of the unit.
<i>squad</i>	Non-Negative value defining the squad of units, if UnitEntity::ability is Ability::BINDING, or null.
<i>row</i>	RowType defining the UnitEntity::row foreign key.

Returns

A Completable tracking operation status.

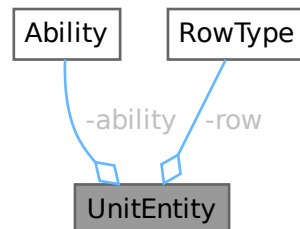
The documentation for this interface was generated from the following file:

- gwent/data/UnitDao.java

5.62 UnitEntity Class Reference

A class representing a card on the game board.

Collaboration diagram for UnitEntity:



Public Member Functions

- **int calculateDamage** (@NonNull DamageCalculator calculator)
Calculates the damage of this unit when (de-)buffed.
- **Color isBuffed** (@NonNull DamageCalculator calculator)
Returns whether this unit is (de-)buffed.
- **String toString** (@NonNull Context context)
Returns a string representation of this unit.
- **int getId** ()
Getter for #id.
- **boolean isEpic** ()
Getter for #epic.

- `int getDamage ()`
Getter for #damage.
- `Ability getAbility ()`
Getter for #ability.
- `Integer getSquad ()`
Getter for #squad.
- `RowType getRow ()`
Getter for #row.

Static Public Member Functions

- `static String collectionToString (@NonNull Context context, @NonNull Collection< UnitEntity > units)`
Creates a String containing the descriptions of all units in the given collection, separated by commas.

Static Public Attributes

- `static final Integer[] EPIC_DAMAGE_VALUES = new Integer[] {0, 7, 8, 10, 11, 15}`
Integer array containing every damage value an epic unit may have.
- `static final int NON_EPIC_DAMAGE_VALUES_UPPER_BOUND = 20`
Integer defining what the maximum damage of a non-epic UnitEntity can be.

Package Functions

- `UnitEntity (boolean epic, @IntRange(from=0) int damage, @NonNull Ability ability, @IntRange(from=1) @Nullable Integer squad, @NonNull RowType row)`
Constructor of a UnitEntity.
- `void setId (int id)`
Setter for #id.
- `void setEpic (boolean epic)`
Setter for #epic.
- `void setDamage (@IntRange(from=0) int damage)`
Setter for #damage.
- `void setAbility (@NonNull Ability ability)`
Setter for #ability.
- `void setSquad (@IntRange(from=1) @Nullable Integer squad)`
Setter for #squad.
- `void setRow (@NonNull RowType row)`
Setter for #row.

Private Attributes

- `int id`
Defines the primary key of the represented unit.
- `boolean epic`
Defines whether the represented card is epic.
- `int damage`
Defines the base-damage of the represented card.
- `Ability ability`
Defines the Ability of the represented card.
- `Integer squad`
Defines the squad the unit belongs to if #ability is Ability::BINDING.
- `RowType row`
Defines the attack row the card lies in.

5.62.1 Detailed Description

A class representing a card on the game board.

Is a persistent Entity and is therefore saved in a database table named `units`.

Definition at line 36 of file [UnitEntity.java](#).

5.62.2 Constructor & Destructor Documentation

5.62.2.1 UnitEntity()

```
UnitEntity (
    boolean epic,
    @IntRange(from=0) int damage,
    @NonNull Ability ability,
    @IntRange(from=1) @Nullable Integer squad,
    @NonNull RowType row ) [package]
```

Constructor of a UnitEntity.

#id may not be set here since the value is generated automatically.

Parameters

<i>epic</i>	Boolean representing whether card is #epic.
<i>damage</i>	Non-negative value representing the #damage of the card.
<i>ability</i>	Ability representing the #ability of the card.
<i>squad</i>	Integer representing the #squad of a card that has the Ability::BINDING #ability.
<i>row</i>	RowType representing the combat type of the card.

Exceptions

<i>IllegalArgumentException</i>	When damage is less than zero or if ability is Ability::BINDING and squad is null or less than one or if ability is not Ability::BINDING and squad is not null.
---------------------------------	---

Definition at line 105 of file [UnitEntity.java](#).

References [Ability.BINDING](#).

5.62.3 Member Function Documentation

5.62.3.1 calculateDamage()

```
int calculateDamage (
    @NonNull DamageCalculator calculator )
```

Calculates the damage of this unit when (de-)buffed.

Returns #damage if #epic is true. Otherwise, the damage is calculated through the given DamageCalculator, which follows the visitor pattern.

Parameters

<i>calculator</i>	DamageCalculator visitor used for damage calculation.
-------------------	---

Returns

An Integer representing the units (de-)buffed damage.

See also

[#getDamage\(\)](#)

Definition at line 132 of file [UnitEntity.java](#).

5.62.3.2 collectionToString()

```
static String collectionToString (
    @NonNull Context context,
    @NonNull Collection< UnitEntity > units ) [static]
```

Creates a String containing the descriptions of all units in the given collection, separated by commas. If n units have the same description, instead of printing the same description n times, "n×" is printed in front of the description. Unit descriptions are created using #toString(Context). Ordering of units in the given Collection is not guaranteed to be kept.

Parameters

<i>context</i>	Context used to acquire String resources.
<i>units</i>	Collection of UnitEntity objects that should be in the created String.

Returns

A String containing the description of all units.

Exceptions

<i>IllegalArgumentException</i>	When units collection is empty.
---------------------------------	---------------------------------

See also

#toString(Context)

Definition at line 166 of file [UnitEntity.java](#).

5.62.3.3 getAbility()

```
Ability getAbility ( )
Getter for #ability.
```

Returns

Ability representing the units' ability.

Definition at line 315 of file [UnitEntity.java](#).

5.62.3.4 getDamage()

```
int getDamage ( )
Getter for #damage.
```

Returns

Integer representing the card's base-damage.

See also

#calculateDamage(DamageCalculator)

Definition at line 291 of file [UnitEntity.java](#).

5.62.3.5 getId()

```
int getId ( )
Getter for #id.
```

Returns

Integer representing the units' id.

Definition at line 252 of file [UnitEntity.java](#).
Referenced by [UnitRepository.getUnitsFlowable\(\)](#).

5.62.3.6 getRow()

```
RowType getRow ( )
```

Getter for #row.

Returns

RowType representing the units combat row.

Definition at line 362 of file [UnitEntity.java](#).

5.62.3.7 getSquad()

```
Integer getSquad ( )
```

Getter for #squad.

Returns

Integer representing the units' squad if #ability is Ability::BINDING or null.

Definition at line 335 of file [UnitEntity.java](#).

5.62.3.8 isBuffed()

```
Color isBuffed (
    @NonNull DamageCalculator calculator )
```

Returns whether this unit is (de-)buffed.

Returns Color::DEFAULT if #epic is true. Otherwise, whether the unit is (de-)buffed is calculated through the given DamageCalculator, which follows the visitor pattern.

Parameters

<i>calculator</i>	DamageCalculator visitor used for damage calculation.
-------------------	---

Returns

A Color representing the units (de-)buffed damage.

See also

[#getDamage\(\)](#)

Definition at line 147 of file [UnitEntity.java](#).

5.62.3.9 isEpic()

```
boolean isEpic ( )
```

Getter for #epic.

Returns

Boolean representing whether the card is epic.

Definition at line 271 of file [UnitEntity.java](#).

5.62.3.10 setAbility()

```
void setAbility (
    @NonNull Ability ability ) [package]
```

Setter for #ability.

Only used by Room extension.

Parameters

<i>ability</i>	Ability representing the units' ability.
----------------	--

Definition at line 325 of file [UnitEntity.java](#).

5.62.3.11 setDamage()

```
void setDamage (
    @IntRange(from=0) int damage ) [package]
```

Setter for #damage.

Only used by Room extension.

Parameters

<i>damage</i>	Integer representing the card's base-damage.
---------------	--

Exceptions

<i>IllegalArgumentException</i>	When damage is less than zero.
---------------------------------	--------------------------------

Definition at line 302 of file [UnitEntity.java](#).

5.62.3.12 setEpic()

```
void setEpic (
    boolean epic ) [package]
```

Setter for #epic.

Only used by Room extension.

Parameters

<i>epic</i>	Boolean representing whether the card is epic.
-------------	--

Definition at line 281 of file [UnitEntity.java](#).

5.62.3.13 setId()

```
void setId (
    int id ) [package]
```

Setter for #id.

Only used by Room extension.

Parameters

<i>id</i>	Integer representing the units' id.
-----------	-------------------------------------

Definition at line 262 of file [UnitEntity.java](#).

5.62.3.14 setRow()

```
void setRow (
    @NonNull RowType row ) [package]
```

Setter for #row.

Only used by Room extension.

Parameters

<i>row</i>	RowType representing the units combat row.
------------	--

Definition at line 372 of file [UnitEntity.java](#).

5.62.3.15 setSquad()

```
void setSquad (
    @IntRange(from=1) @Nullable Integer squad ) [package]
```

Setter for #squad.

Only used by Room extension.

Parameters

<i>squad</i>	Integer representing the units' squad if #ability is Ability::BINDING or null.
--------------	--

Exceptions

<i>IllegalArgumentException</i>	When #ability is Ability::BINDING and squad is null or less than 1 or if #ability is not Ability::BINDING and squad is not null.
---------------------------------	--

Definition at line 346 of file [UnitEntity.java](#).

References [Ability.BINDING](#).

5.62.3.16 toString()

```
String toString (
    @NonNull Context context )
```

Returns a string representation of this unit.

The representation contains information on each field of this class, i.e., #row, #epic, #damage, #ability, and #squad.

Parameters

<i>context</i>	Context used to acquire String resources.
----------------	---

Returns

A string representing the unit.

Definition at line 210 of file [UnitEntity.java](#).

5.62.4 Member Data Documentation**5.62.4.1 ability**

```
Ability ability [private]
```

Defines the Ability of the represented card.

Is set to Ability::NONE by default. Must not be null.

Definition at line 75 of file [UnitEntity.java](#).

5.62.4.2 damage

```
int damage [private]
```

Defines the base-damage of the represented card.

Must be non-negative.

Definition at line 55 of file [UnitEntity.java](#).

5.62.4.3 epic

```
boolean epic [private]
```

Defines whether the represented card is epic.

Is set to `false` by default.

Definition at line 49 of file [UnitEntity.java](#).

5.62.4.4 EPIC_DAMAGE_VALUES

```
final Integer [] EPIC_DAMAGE_VALUES = new Integer[] {0, 7, 8, 10, 11, 15} [static]
```

Integer array containing every damage value an epic unit may have.

Definition at line 61 of file [UnitEntity.java](#).

Referenced by [UnitRepository.insertUnit\(\)](#).

5.62.4.5 id

```
int id [private]
```

Defines the primary key of the represented unit.

Is generated automatically on insert.

Definition at line 42 of file [UnitEntity.java](#).

5.62.4.6 NON_EPIC_DAMAGE_VALUES_UPPER_BOUND

```
final int NON_EPIC_DAMAGE_VALUES_UPPER_BOUND = 20 [static]
```

Integer defining what the maximum damage of a non-epic UnitEntity can be.

Definition at line 66 of file [UnitEntity.java](#).

Referenced by [UnitRepository.insertUnit\(\)](#).

5.62.4.7 row

```
RowType row [private]
```

Defines the attack row the card lies in.

Must not be `null`.

Definition at line 92 of file [UnitEntity.java](#).

5.62.4.8 squad

```
Integer squad [private]
```

Defines the squad the unit belongs to if #ability is Ability::BINDING.

If #ability is anything else, this value is `null`.

Definition at line 84 of file [UnitEntity.java](#).

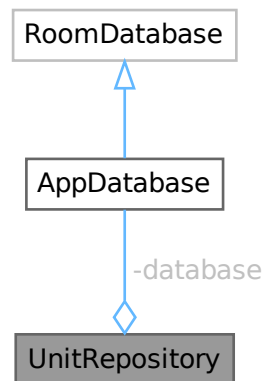
The documentation for this class was generated from the following file:

- gwent/data/UnitEntity.java

5.63 UnitRepository Class Reference

A facade class managing public access to the data layer.

Collaboration diagram for UnitRepository:



Public Member Functions

- Completable `reset ()`
Resets the board by removing all units and resetting row status.
- Completable `reset (@Nullable UnitEntity keptUnit)`
Resets the board by removing all units but the given one and resetting row status.
- Completable `insertUnit (boolean epic, @IntRange(from=0, to=UnitEntity.NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) int damage, @NonNull Ability ability, @IntRange(from=1) @Nullable Integer squad, @NonNull RowType row, @IntRange(from=0) int number)`
Adds a number of units with the given stats to the given row.
- Completable `switchWeather (@NonNull RowType row)`
Flips RowEntity::weather of the given attack row.
- Single< Boolean > `isWeather (@NonNull RowType row)`
Returns the value of RowEntity::weather for the given attack row.
- Flowable< Boolean > `isWeatherFlowable (@NonNull RowType row)`
Returns a Flowable emitting the latest value of RowEntity::weather for the given attack row.
- Completable `clearWeather ()`
Sets RowEntity::weather to false for all attack rows.
- Completable `switchHorn (@NonNull RowType row)`
Flips RowEntity::horn of the given attack row.
- Single< Boolean > `isHorn (@NonNull RowType row)`
Returns the value of RowEntity::horn for the given attack row.
- Flowable< Boolean > `isHornFlowable (@NonNull RowType row)`
Returns a Flowable emitting the values of RowEntity::horn for the given attack row.
- Completable `delete (@NonNull Collection< UnitEntity > units)`
Removes the given units from the game.
- Completable `delete (int id)`
Removes the unit with the given id from the game.
- Completable `copy (int id)`
Copies the unit with the given id.
- Single< Integer > `countUnits (@NonNull RowType row)`
Counts the units in the given attack row.

- `Single< Integer > countUnits ()`
Counts the units in all attack rows.
- `Single< UnitEntity > getUnit (int id)`
Returns the unit with the given id.
- `Single< List< UnitEntity > > getUnits (@NonNull RowType row)`
Returns the units in the given attack row.
- `Flowable< List< UnitEntity > > getUnitsFlowable (@NonNull RowType row)`
Returns a Flowable emitting the units in the given attack row.
- `Single< List< UnitEntity > > getUnits ()`
Returns the units in the given attack row.
- `Flowable< Boolean > hasNonEpicUnitsFlowable ()`
Returns a Flowable emitting Boolean values defining whether there are units in the game that are not epic.

Static Public Member Functions

- `static Single< UnitRepository > getRepository (@NonNull AppDatabase database)`
Factory method for the UnitRepository class.

Private Member Functions

- `UnitRepository (@NonNull AppDatabase database)`
Constructor of a UnitRepository.
- `Completable initializeRows ()`
Adds one attack row for each RowType.
- `Completable insertUnit (boolean epic, @IntRange(from=0, to=UnitEntity.NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) int damage, @NonNull Ability ability, @IntRange(from=1) @Nullable Integer squad, @NonNull RowType row)`
Adds a unit with the given stats to the given row.

Private Attributes

- `final AppDatabase database`
Defines the AppDatabase that is used as a data source by this repository.

5.63.1 Detailed Description

A facade class managing public access to the data layer.

The contained functions mostly redirect requests to package-private DAO methods in RowDao and UnitDao. Some functions implement slightly more complex behavior by chaining multiple DAO calls, e.g., `#reset()`.

Definition at line 25 of file [UnitRepository.java](#).

5.63.2 Constructor & Destructor Documentation

5.63.2.1 UnitRepository()

```
UnitRepository (
    @NonNull AppDatabase database ) [private]
```

Constructor of a UnitRepository.

Depends on the given AppDatabase as a data source. Should only be called by `#getRepository()`.

Parameters

<i>database</i>	AppDatabase that is injected for the repository.
-----------------	--

See also

`#getRepository(AppDatabase)`

Definition at line 41 of file [UnitRepository.java](#).

References [UnitRepository.database](#).

5.63.3 Member Function Documentation

5.63.3.1 clearWeather()

```
Completable clearWeather ( )
```

Sets `RowEntity::weather` to `false` for all attack rows.

Returns

A Completable tracking operation status.

Definition at line 208 of file [UnitRepository.java](#).

References [RowDao.clearWeather\(\)](#), [UnitRepository.database](#), and [AppDatabase.rows\(\)](#).

5.63.3.2 copy()

```
Completable copy (
    int id )
```

Copies the unit with the given id.

Parameters

<i>id</i>	Integer representing the unit that should be copied.
-----------	--

Returns

A Completable tracking operation status.

Definition at line 277 of file [UnitRepository.java](#).

References [UnitRepository.getUnit\(\)](#), and [UnitRepository.insertUnit\(\)](#).

5.63.3.3 countUnits() [1/2]

```
Single< Integer > countUnits ( )
```

Counts the units in all attack rows.

Returns

A Single tracking operation status and returning the value.

See also

`#countUnits(RowType)`

Definition at line 302 of file [UnitRepository.java](#).

References [UnitDao.countUnits\(\)](#), [UnitRepository.database](#), and [AppDatabase.units\(\)](#).

5.63.3.4 countUnits() [2/2]

```
Single< Integer > countUnits (
    @NonNull RowType row )
```

Counts the units in the given attack row.

Parameters

<i>row</i>	RowEntity::id where the units are counted.
------------	--

Returns

A Single tracking operation status and returning the value.

See also

`#countUnits()`

Definition at line 291 of file [UnitRepository.java](#).

References [UnitDao.countUnits\(\)](#), [UnitRepository.database](#), and [AppDatabase.units\(\)](#).

5.63.3.5 delete() [1/2]

```
Completable delete (
    @NonNull Collection< UnitEntity > units )
```

Removes the given units from the game.

Parameters

<i>units</i>	List of units to be removed.
--------------	------------------------------

Returns

A Completable tracking operation status.

Definition at line 256 of file [UnitRepository.java](#).

References [UnitRepository.database](#), [UnitDao.deleteUnits\(\)](#), and [AppDatabase.units\(\)](#).

5.63.3.6 delete() [2/2]

```
Completable delete (
    int id )
```

Removes the unit with the given id from the game.

Parameters

<i>id</i>	Integer representing the unit that should be deleted.
-----------	---

Returns

A Completable tracking operation status.

Definition at line 267 of file [UnitRepository.java](#).

References [UnitRepository.database](#), [UnitDao.deleteUnit\(\)](#), and [AppDatabase.units\(\)](#).

5.63.3.7 getRepository()

```
static Single< UnitRepository > getRepository (
    @NonNull AppDatabase database ) [static]
```

Factory method for the UnitRepository class.

Creates a new UnitRepository managing the given AppDatabase. Also initializes one RowEntity per RowType using `#initializeRows()`.

See also

`#initializeRows()`

Parameters

<i>database</i>	AppDatabase managed and initialized by the returned UnitRepository.
-----------------	---

Returns

A Single emitting the created UnitRepository when initialization is finished.

Definition at line 54 of file [UnitRepository.java](#).

References [UnitRepository.database](#), and [UnitRepository.initializeRows\(\)](#).

5.63.3.8 getUnit()

```
Single< UnitEntity > getUnit (
    int id )
```

Returns the unit with the given id.

Parameters

<i>id</i>	Integer representing the queried unit.
-----------	--

Returns

A Single tracking operation status and returning the value.

Definition at line 313 of file [UnitRepository.java](#).

References [UnitRepository.database](#), [UnitDao.getUnit\(\)](#), and [AppDatabase.units\(\)](#).

Referenced by [UnitRepository.copy\(\)](#).

5.63.3.9 getUnits() [1/2]

```
Single< List< UnitEntity > > getUnits ( )
```

Returns the units in the given attack row.

Returns

A Single tracking operation status and returning the value.

See also

[#getUnits\(RowType\)](#)

Definition at line 361 of file [UnitRepository.java](#).

References [UnitRepository.database](#), [UnitDao.getUnits\(\)](#), and [AppDatabase.units\(\)](#).

5.63.3.10 getUnits() [2/2]

```
Single< List< UnitEntity > > getUnits (
    @NonNull RowType row )
```

Returns the units in the given attack row.

See also

[#getUnitsFlowable\(RowType\)](#)

Parameters

<i>row</i>	RowEntity::id where the units have been placed.
------------	---

Returns

A Single tracking operation status and returning the value.

See also

`#getUnits()`

Definition at line 325 of file [UnitRepository.java](#).

References [UnitRepository.database](#), [UnitDao.getUnits\(\)](#), and [AppDatabase.units\(\)](#).

5.63.3.11 getUnitsFlowable()

```
Flowable< List< UnitEntity > > getUnitsFlowable (
    @NonNull RowType row )
```

Returns a Flowable emitting the units in the given attack row.

Parameters

<i>row</i>	RowEntity::id where the units have been placed.
------------	---

Returns

A Flowable emitting the values.

See also

`#getUnits(RowType)`

`#getUnits()`

Definition at line 338 of file [UnitRepository.java](#).

References [UnitRepository.database](#), [UnitEntity.getId\(\)](#), [UnitDao.getUnitsFlowable\(\)](#), and [AppDatabase.units\(\)](#).

5.63.3.12 hasNonEpicUnitsFlowable()

```
Flowable< Boolean > hasNonEpicUnitsFlowable ( )
```

Returns a Flowable emitting Boolean values defining whether there are units in the game that are not epic.

Returns

A Flowable emitting the values.

Definition at line 370 of file [UnitRepository.java](#).

References [UnitRepository.database](#), [UnitDao.hasNonEpicUnitsFlowable\(\)](#), and [AppDatabase.units\(\)](#).

5.63.3.13 initializeRows()

```
Completable initializeRows ( ) [private]
```

Adds one attack row for each RowType.

If an attack row already exists, it is not inserted again.

Returns

A Completable tracking operation status.

Definition at line 69 of file [UnitRepository.java](#).

References [UnitRepository.database](#), [RowDao.insertRow\(\)](#), and [AppDatabase.rows\(\)](#).

Referenced by [UnitRepository.getRepository\(\)](#), and [UnitRepository.reset\(\)](#).

5.63.3.14 insertUnit() [1/2]

```
Completable insertUnit (
    boolean epic,
    @IntRange(from=0, to=UnitEntity.NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) int damage,
    @NonNull Ability ability,
    @IntRange(from=1) @Nullable Integer squad,
    @NonNull RowType row ) [private]
```

Adds a unit with the given stats to the given row.

Parameters

<i>epic</i>	Boolean representing whether card is #epic.
<i>damage</i>	Non-negative value representing the #damage of the card.
<i>ability</i>	Ability representing the #ability of the card.
<i>squad</i>	Integer representing the #squad of a card that has the Ability::BINDING #ability.
<i>row</i>	RowType representing the combat type of the card.

Returns

A Completable tracking operation status.

Exceptions

<i>IllegalArgumentException</i>	When damage is less than zero or if ability is Ability::BINDING and squad is null or less than zero or if ability is not Ability::BINDING and squad is not null.
---------------------------------	--

Definition at line 119 of file [UnitRepository.java](#).

References [Ability.BINDING](#), [UnitRepository.database](#), [UnitEntity.EPIC_DAMAGE_VALUES](#), [UnitDao.insertUnit\(\)](#), [UnitEntity.NON_EPIC_DAMAGE_VALUES_UPPER_BOUND](#), and [AppDatabase.units\(\)](#).

Referenced by [UnitRepository.copy\(\)](#), and [UnitRepository.insertUnit\(\)](#).

5.63.3.15 insertUnit() [2/2]

```
Completable insertUnit (
    boolean epic,
    @IntRange(from=0, to=UnitEntity.NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) int damage,
    @NonNull Ability ability,
    @IntRange(from=1) @Nullable Integer squad,
    @NonNull RowType row,
    @IntRange(from=0) int number )
```

Adds a number of units with the given stats to the given row.

Essentially calls #insertUnit(boolean, int, Ability, Integer, RowType) multiple times.

Parameters

<i>epic</i>	Boolean representing whether card is #epic.
<i>damage</i>	Non-negative value representing the #damage of the card.
<i>ability</i>	Ability representing the #ability of the card.
<i>squad</i>	Integer representing the #squad of a card that has the Ability::BINDING #ability.
<i>row</i>	RowType representing the combat type of the card.
<i>number</i>	Integer representing the number of units to be added.

Returns

A Completable tracking operation status.

See also

[#insertUnit\(boolean, int, Ability, Integer, RowType\)](#)

Exceptions

<i>IllegalArgumentException</i>	When damage is less than zero or if ability is Ability::BINDING and squad is null or less than zero or if ability is not Ability::BINDING and squad is not null.
---------------------------------	--

Definition at line 153 of file [UnitRepository.java](#).

References [UnitRepository.insertUnit\(\)](#).

5.63.3.16 isHorn()

```
Single< Boolean > isHorn (
    @NonNull RowType row )
```

Returns the value of RowEntity::horn for the given attack row.

See also

[#isHornFlowable\(RowType\)](#)

Parameters

<i>row</i>	RowEntity::id where the horn status is queried.
------------	---

Returns

A Single tracking operation status and returning the value.

Definition at line 229 of file [UnitRepository.java](#).

References [UnitRepository.database](#), [RowDao.isHorn\(\)](#), and [AppDatabase.rows\(\)](#).

5.63.3.17 isHornFlowable()

```
Flowable< Boolean > isHornFlowable (
    @NonNull RowType row )
```

Returns a Flowable emitting the values of RowEntity::horn for the given attack row.

Parameters

<i>row</i>	RowEntity::id where the horn status is queried.
------------	---

Returns

A Flowable emitting the values.

See also

[#isHorn\(RowType\)](#)

Definition at line 241 of file [UnitRepository.java](#).

References [UnitRepository.database](#), [RowDao.isHornFlowable\(\)](#), and [AppDatabase.rows\(\)](#).

5.63.3.18 isWeather()

```
Single< Boolean > isWeather (
    @NonNull RowType row )
```

Returns the value of RowEntity::weather for the given attack row.

See also

[#isWeatherFlowable\(RowType\)](#)

Parameters

<i>row</i>	RowEntity::id where the weather is queried.
------------	---

Returns

A Single tracking operation status and returning the value.

Definition at line 181 of file [UnitRepository.java](#).

References [UnitRepository.database](#), [RowDao.isWeather\(\)](#), and [AppDatabase.rows\(\)](#).

5.63.3.19 isWeatherFlowable()

```
Flowable< Boolean > isWeatherFlowable (
    @NonNull RowType row )
```

Returns a Flowable emitting the latest value of RowEntity::weather for the given attack row.

Parameters

<i>row</i>	RowEntity::id where the weather is queried.
------------	---

Returns

A Flowable emitting the values.

See also

[#isWeather\(RowType\)](#)

Definition at line 193 of file [UnitRepository.java](#).

References [UnitRepository.database](#), [RowDao.isWeatherFlowable\(\)](#), and [AppDatabase.rows\(\)](#).

5.63.3.20 reset() [1/2]

```
Completable reset ( )
```

Resets the board by removing all units and resetting row status.

Resetting row status is equivalent to removing the old rows and calling [#initializeRows\(\)](#). Method is a wrapper for [#reset\(UnitEntity\)](#).

Returns

A Completable tracking operation status.

See also

[#reset\(UnitEntity\)](#)

Definition at line 86 of file [UnitRepository.java](#).

References [UnitRepository.reset\(\)](#).

Referenced by [UnitRepository.reset\(\)](#).

5.63.3.21 reset() [2/2]

```
Completable reset (
    @Nullable UnitEntity keptUnit )
```

Resets the board by removing all units but the given one and resetting row status.

Resetting row status is equivalent to removing the old rows and calling [#initializeRows\(\)](#).

Parameters

<i>keptUnit</i>	UnitEntity that should be kept.
-----------------	---------------------------------

Returns

A Completable tracking operation status.

See also

[#initializeRows\(\)](#)

Definition at line 99 of file [UnitRepository.java](#).

References [RowDao.clearRows\(\)](#), [UnitRepository.database](#), [UnitRepository.initializeRows\(\)](#), [UnitDao.insertUnit\(\)](#), [AppDatabase.rows\(\)](#), and [AppDatabase.units\(\)](#).

5.63.3.22 switchHorn()

```
Completable switchHorn (
    @NonNull RowType row )
```

Flips RowEntity::horn of the given attack row.

Parameters

<i>row</i>	RowEntity::id where the horn status should be updated.
------------	--

Returns

A Completable tracking operation status.

Definition at line 218 of file [UnitRepository.java](#).

References [UnitRepository.database](#), [AppDatabase.rows\(\)](#), and [RowDao.updateHorn\(\)](#).

5.63.3.23 switchWeather()

```
Completable switchWeather (
    @NonNull RowType row )
```

Flips RowEntity::weather of the given attack row.

Parameters

<i>row</i>	RowEntity::id where the weather should be updated.
------------	--

Returns

A Completable tracking operation status.

Definition at line 170 of file [UnitRepository.java](#).

References [UnitRepository.database](#), [AppDatabase.rows\(\)](#), and [RowDao.updateWeather\(\)](#).

5.63.4 Member Data Documentation**5.63.4.1 database**

```
final AppDatabase database [private]
```

Defines the AppDatabase that is used as a data source by this repository.

Is provided by dependency injection in [#UnitRepository\(\)](#).

Definition at line 31 of file [UnitRepository.java](#).

Referenced by [UnitRepository.UnitRepository\(\)](#), [UnitRepository.clearWeather\(\)](#), [UnitRepository.countUnits\(\)](#), [UnitRepository.countUnits\(\)](#), [UnitRepository.delete\(\)](#), [UnitRepository.delete\(\)](#), [UnitRepository.getRepository\(\)](#), [UnitRepository.getUnit\(\)](#), [UnitRepository.getUnits\(\)](#), [UnitRepository.getUnits\(\)](#), [UnitRepository.getUnitsFlowable\(\)](#), [UnitRepository.hasNonEpicUnitsFlowable\(\)](#), [UnitRepository.initializeRows\(\)](#), [UnitRepository.insertUnit\(\)](#), [UnitRepository.isHorn\(\)](#), [UnitRepository.isHornFlowable\(\)](#), [UnitRepository.isWeather\(\)](#), [UnitRepository.isWeatherFlowable\(\)](#), [UnitRepository.reset\(\)](#), [UnitRepository.switchHorn\(\)](#), and [UnitRepository.switchWeather\(\)](#).

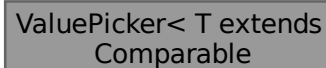
The documentation for this class was generated from the following file:

- gwent/data/UnitRepository.java

5.64 ValuePicker< T extends Comparable Class Template Reference

A generic abstract wrapper class around `NumberPicker` used to select values of arbitrary types extending `Comparable`.

Collaboration diagram for `ValuePicker< T extends Comparable`:



A diagram showing a box labeled "ValuePicker< T extends Comparable".

5.64.1 Detailed Description

A generic abstract wrapper class around `NumberPicker` used to select values of arbitrary types extending `Comparable`.

Must be subclassed to provide the displayed `String` values using `#getDisplayString()`. The default case, which is implemented in `StringValuePicker`, is that `displayIntegers` maps the `selectableValues` to a `String` resource id.

See also

`NumberPicker`
`StringValuePicker`

Definition at line 29 of file [ValuePicker.java](#).

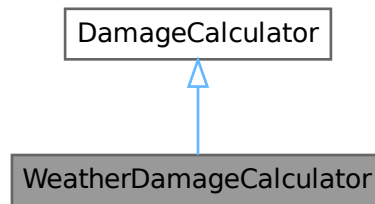
The documentation for this class was generated from the following file:

- gwent/ui/dialogs/addcard/ValuePicker.java

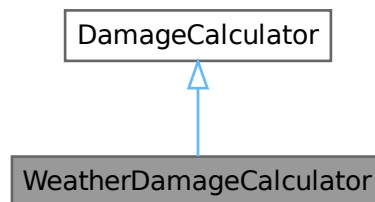
5.65 WeatherDamageCalculator Class Reference

A `DamageCalculator` class responsible for calculating the weather debuff if necessary, i.e., if `#weather` is true.

Inheritance diagram for WeatherDamageCalculator:



Collaboration diagram for WeatherDamageCalculator:



Public Member Functions

- `int calculateDamage (int id, @IntRange(from=0) int damage)`
Calculates the (de-)buffed damage of unit with the given id and the given base-damage.
- `Color isBuffed (int id)`
Calculates whether the unit with the given id is shown as Color::BUFFED, Color::DEBUFFED, or Color::DEFAULT.

Package Functions

- `WeatherDamageCalculator (boolean weather)`
Constructor of a WeatherDamageCalculator.

Private Attributes

- final boolean `weather`
Defines whether the weather debuff should be applied when calling #calculateDamage().

5.65.1 Detailed Description

A DamageCalculator class responsible for calculating the weather debuff if necessary, i.e., if #weather is true. Acts as the ConcreteComponent in the implemented decorator pattern and is therefore always at the end of the decorator chain.

Definition at line 14 of file [WeatherDamageCalculator.java](#).

5.65.2 Constructor & Destructor Documentation

5.65.2.1 WeatherDamageCalculator()

```
WeatherDamageCalculator (
    boolean weather ) [package]
```

Constructor of a WeatherDamageCalculator.

Should only be called by DamageCalculatorBuilder.

Parameters

<i>weather</i>	Boolean defining whether the weather debuff should be applied.
----------------	--

See also

DamageCalculatorBuilder

Definition at line 27 of file [WeatherDamageCalculator.java](#).

References [WeatherDamageCalculator.weather](#).

5.65.3 Member Function Documentation

5.65.3.1 calculateDamage()

```
int calculateDamage (
    int id,
    @IntRange(from=0) int damage )
```

Calculates the (de-)buffed damage of unit with the given id and the given base-damage.

Returns the given damage if #weather is false or 1.

Parameters

<i>id</i>	Integer representing the UnitEntity::id of the unit whose (de-)buff damage is calculated.
<i>damage</i>	Integer representing the base-damage of the unit whose (de-)buff damage is calculated.

Returns

Integer representing the (de-)buffed damage of the unit.

Exceptions

<i>IllegalArgumentException</i>	When damage is negative.
---------------------------------	--------------------------

Implements [DamageCalculator](#).

Definition at line 41 of file [WeatherDamageCalculator.java](#).

References [WeatherDamageCalculator.weather](#).

5.65.3.2 isBuffed()

```
Color isBuffed (
    int id )
```

Calculates whether the unit with the given id is shown as Color::BUFFED, Color::DEBUFFED, or Color::DEFAULT.

Returns Color::DEBUFFED if #weather is true or else Color::DEFAULT.

Parameters

<i>id</i>	Integer representing the UnitEntity::id of the unit buff status is calculated.
-----------	--

Returns

Color representing whether the unit is buffed, de-buffed or not affected.

See also

Color

Implements [DamageCalculator](#).

Definition at line 61 of file [WeatherDamageCalculator.java](#).

References [WeatherDamageCalculator.weather](#).

5.65.4 Member Data Documentation

5.65.4.1 weather

```
final boolean weather [private]
```

Defines whether the weather debuff should be applied when calling `#calculateDamage()`.

Definition at line 18 of file [WeatherDamageCalculator.java](#).

Referenced by [WeatherDamageCalculator.WeatherDamageCalculator\(\)](#), [WeatherDamageCalculator.calculateDamage\(\)](#), and [WeatherDamageCalculator.isBuffed\(\)](#).

The documentation for this class was generated from the following file:

- `gwent/domain/damage/WeatherDamageCalculator.java`

Chapter 6

File Documentation

6.1 /home/runner/work/Gwent/Gwent/app/src/main/AndroidManifest.xml

```
00001 <?xml version="1.0" encoding="utf-8"?>
00002 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
00003     xmlns:tools="http://schemas.android.com/tools">
00004
00005     <uses-permission
00006         android:name="com.google.android.gms.permission.AD_ID"
00007         tools:node="remove" />
00008     <application
00009         android:name=".GwentApplication"
00010         android:appCategory="game"
00011         android:description="@string/app_description"
00012         android:icon="@mipmap/icon_launcher"
00013         android:label="@string/app_name"
00014         android:resizeableActivity="false"
00015         android:roundIcon="@mipmap/icon_launcher_round"
00016         android:supportsRtl="true"
00017         android:theme="@style/ScoiataelTheme"
00018         tools:targetApi="o">
00019         <activity android:name=".ui.introduction.IntroductionActivity" />
00020         <activity
00021             android:name=".ui.main.MainActivity"
00022             android:exported="true"
00023             android:screenOrientation="userLandscape">
00024             <intent-filter>
00025                 <action android:name="android.intent.action.MAIN" />
00026                 <category android:name="android.intent.category.LAUNCHER" />
00027             </intent-filter>
00028         </activity>
00029         <activity
00030             android:name=".ui.settings.SettingsActivity"
00031             android:exported="true"
00032             android:label="@string/settings_title">
00033             <intent-filter>
00034                 <action android:name="android.intent.action.APPLICATION_PREFERENCES" />
00035             </intent-filter>
00036         </activity>
00037         <activity
00038             android:name=".ui.settings.RuleActivity"
00039             android:label="@string/preference_rules_header" />
00040     </application>
00041 </manifest>
```

6.2 Ability.java

```
00001 package com.paternaggschga.gwent.data;
00002
00006 public enum Ability {
00010     NONE,
00011
00015     HORN,
00016
00020     REVENGE,
00021
00025     BINDING,
00026
00030     MORAL_BOOST
00031 }
```

6.3 AppDatabase.java

```

00001 package com.paternaggschga.gwent.data;
00002
00003 import androidx.room.Database;
00004 import androidx.room.RoomDatabase;
00005
00010 @Database(entities = {UnitEntity.class, RowEntity.class}, version = 1)
00011 public abstract class AppDatabase extends RoomDatabase {
00017     abstract UnitDao units();
00018
00024     abstract RowDao rows();
00025 }

```

6.4 RowDao.java

```

00001 package com.paternaggschga.gwent.data;
00002
00003 import androidx.annotation.NonNull;
00004 import androidx.room.Dao;
00005 import androidx.room.Insert;
00006 import androidx.room.OnConflictStrategy;
00007 import androidx.room.Query;
00008
00009 import io.reactivex.rxjava3.core.Completable;
00010 import io.reactivex.rxjava3.core.Flowable;
00011 import io.reactivex.rxjava3.core.Single;
00012
00017 @Dao
00018 @SuppressWarnings("NullableProblems")
00019 interface RowDao {
00027     @Insert(onConflict = OnConflictStrategy.IGNORE)
00028     Completable insertRow(@NonNull RowEntity row);
00029
00035     @Query("DELETE FROM rows")
00036     Completable clearRows();
00037
00044     @Query("UPDATE rows SET weather = NOT weather WHERE id = :row")
00045     Completable updateWeather(@NonNull RowType row);
00046
00052     @Query("UPDATE rows SET weather = 0")
00053     Completable clearWeather();
00054
00061     @Query("UPDATE rows SET horn = NOT horn WHERE id = :row")
00062     Completable updateHorn(@NonNull RowType row);
00063
00070     @Query("SELECT weather FROM rows WHERE id = :row")
00071     Single<Boolean> isWeather(@NonNull RowType row);
00072
00080     @Query("SELECT weather FROM rows WHERE id = :row")
00081     Flowable<Boolean> isWeatherFlowable(@NonNull RowType row);
00082
00089     @Query("SELECT horn FROM rows WHERE id = :row")
00090     Single<Boolean> isHorn(@NonNull RowType row);
00091
00099     @Query("SELECT horn FROM rows WHERE id = :row")
00100     Flowable<Boolean> isHornFlowable(@NonNull RowType row);
00101 }

```

6.5 RowEntity.java

```

00001 package com.paternaggschga.gwent.data;
00002
00003 import androidx.annotation.NonNull;
00004 import androidx.room.ColumnInfo;
00005 import androidx.room.Entity;
00006 import androidx.room.PrimaryKey;
00007
00012 @Entity(tableName = "rows")
00013 @SuppressWarnings("unused")
00014 class RowEntity {
00020     @PrimaryKey
00021     @NonNull
00022     private final RowType id;
00023
00029     @ColumnInfo(defaultValue = "false")
00030     private boolean weather;
00031
00037     @ColumnInfo(defaultValue = "false")
00038     private boolean horn;
00039
00047     RowEntity(@NonNull RowType id) {

```

```

00048         this.id = id;
00049     }
00050
00057     @NonNull
00058     RowType getId() {
00059         return id;
00060     }
00061
00068     boolean isWeather() {
00069         return weather;
00070     }
00071
00078     void setWeather(boolean weather) {
00079         this.weather = weather;
00080     }
00081
00088     boolean isHorn() {
00089         return horn;
00090     }
00091
00098     void setHorn(boolean horn) {
00099         this.horn = horn;
00100     }
00101 }

```

6.6 RowType.java

```

00001 package com.paternaggschga.gwent.data;
00002
00006 public enum RowType {
00010     MELEE,
00011
00015     RANGE,
00016
00020     SIEGE
00021 }

```

6.7 UnitDao.java

```

00001 package com.paternaggschga.gwent.data;
00002
00003 import androidx.annotation.IntRange;
00004 import androidx.annotation.NonNull;
00005 import androidx.annotation.Nullable;
00006 import androidx.room.Dao;
00007 import androidx.room.Delete;
00008 import androidx.room.Insert;
00009 import androidx.room.OnConflictStrategy;
00010 import androidx.room.Query;
00011
00012 import java.util.Collection;
00013 import java.util.List;
00014
00015 import io.reactivex.rxjava3.core.Completable;
00016 import io.reactivex.rxjava3.core.Flowable;
00017 import io.reactivex.rxjava3.core.Single;
00018
00023 @Dao
00024 @SuppressWarnings("NullableProblems")
00025 interface UnitDao {
00033     @Insert(onConflict = OnConflictStrategy.IGNORE)
00034     Completable insertUnit(@NonNull UnitEntity unit);
00035
00047     @Query("INSERT INTO units (epic, damage, ability, squad, `row`) VALUES (:epic, :damage, :ability, :squad, :row)")
00048     Completable insertUnit(boolean epic, @IntRange(from = 0, to = UnitEntity.NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) int damage, @NonNull Ability ability, @IntRange(from = 1) @Nullable Integer squad, @NonNull RowType row);
00049
00055     @Query("DELETE FROM units WHERE `id` = :id")
00056     Completable deleteUnit(int id);
00057
00064     @Delete
00065     Completable deleteUnits(@NonNull Collection<UnitEntity> units);
00066
00072     @Query("SELECT * FROM units WHERE `id` = :id")
00073     Single<UnitEntity> getUnit(int id);
00074
00081     @Query("SELECT * FROM units WHERE `row` = :row")
00082     Single<List<UnitEntity>> getUnits(@NonNull RowType row);
00083
00091     @Query("SELECT * FROM units WHERE `row` = :row")

```

```

00092     Flowable<List<UnitEntity>> getUnitsFlowable(@NonNull RowType row);
00093
00098     @Query("SELECT * FROM units")
00099     Single<List<UnitEntity>> getUnits();
00100
00106     @Query("SELECT COUNT(*) FROM units WHERE `row` = :row")
00107     Single<Integer> countUnits(@NonNull RowType row);
00108
00114     @Query("SELECT COUNT(*) FROM units")
00115     Single<Integer> countUnits();
00116
00122     @Query("SELECT COUNT(*) > 0 FROM units WHERE epic = 0")
00123     Flowable<Boolean> hasNonEpicUnitsFlowable();
00124 }

```

6.8 UnitEntity.java

```

00001 package com.paternaggschga.gwent.data;
00002
00003 import static com.paternaggschga.gwent.domain.damage.DamageCalculator.Color;
00004 import static com.paternaggschga.gwent.domain.damage.DamageCalculator.Color.DEFAULT;
00005
00006 import android.content.Context;
00007
00008 import androidx.annotation.IntRange;
00009 import androidx.annotation.NonNull;
00010 import androidx.annotation.Nullable;
00011 import androidx.room.ColumnInfo;
00012 import androidx.room.Entity;
00013 import androidx.room.ForeignKey;
00014 import androidx.room.PrimaryKey;
00015
00016 import com.paternaggschga.gwent.R;
00017 import com.paternaggschga.gwent.domain.damage.DamageCalculator;
00018
00019 import java.util.ArrayList;
00020 import java.util.Collection;
00021 import java.util.HashMap;
00022 import java.util.Iterator;
00023 import java.util.List;
00024 import java.util.Map;
00025
00030 @Entity(tableName = "units", foreignKeys = {
00031     @ForeignKey(entity = RowEntity.class,
00032         parentColumns = "id",
00033         childColumns = "row",
00034         onDelete = ForeignKey.CASCADE)})
00035 @SuppressWarnings("unused")
00036 public class UnitEntity {
00041     @PrimaryKey(autoGenerate = true)
00042     private int id;
00043
00048     @ColumnInfo(defaultValue = "false")
00049     private boolean epic;
00050
00055     private int damage;
00056
00060     @NonNull
00061     public static final Integer[] EPIC_DAMAGE_VALUES = new Integer[]{0, 7, 8, 10, 11, 15};
00062
00066     public static final int NON_EPIC_DAMAGE_VALUES_UPPER_BOUND = 20;
00067
00073     @ColumnInfo(defaultValue = "NONE")
00074     @NonNull
00075     private Ability ability;
00076
00081     @ColumnInfo(defaultValue = "NULL")
00082     @IntRange(from = 1)
00083     @Nullable
00084     private Integer squad;
00085
00090     @ColumnInfo(index = true)
00091     @NonNull
00092     private RowType row;
00093
00105     UnitEntity(boolean epic, @IntRange(from = 0) int damage, @NonNull Ability ability, @IntRange(from
00106 = 1) @Nullable Integer squad, @NonNull RowType row) {
00107         if (damage < 0) {
00108             throw new IllegalArgumentException("Damage must be greater or equal to 0 but is " + damage
00109 + ".");
00110         }
00111         if (ability != Ability.BINDING && squad != null) {
00112             throw new IllegalArgumentException("Squad must be null or ability must be BINDING but
00113 squad is " + squad + " and ability is " + ability + ".");
00114         }
00115     }

```



```

00112         if (ability == Ability.BINDING && (squad == null || squad < 1)) {
00113             throw new IllegalArgumentException("Squad must not be null or less than one but squad is "
+ squad + ".");
00114         }
00115         this.epic = epic;
00116         this.damage = damage;
00117         this.ability = ability;
00118         this.squad = squad;
00119         this.row = row;
00120     }
00121
00132     public int calculateDamage(@NonNull DamageCalculator calculator) {
00133         return epic ? damage : calculator.calculateDamage(id, damage);
00134     }
00135
00146     @NonNull
00147     public Color isBuffed(@NonNull DamageCalculator calculator) {
00148         return epic ? DEFAULT : calculator.isBuffed(id);
00149     }
00150
00165     @NonNull
00166     public static String collectionToString(@NonNull Context context, @NonNull Collection<UnitEntity>
units) {
00167         if (units.isEmpty()) {
00168             throw new IllegalArgumentException("Units collection must not be empty.");
00169         }
00170
00171         Map<String, Integer> descriptionStrings = new HashMap<>(units.size());
00172         for (UnitEntity unit : units) {
00173             String description = unit.toString(context);
00174             descriptionStrings.merge(description, 1, Integer::sum);
00175         }
00176
00177         List<String> descriptions = new ArrayList<>(descriptionStrings.size());
00178         if (descriptionStrings.size() < units.size()) {
00179             descriptionStrings.forEach((key, value) ->
00180                 descriptions.add(context.getString(R.string.unit_toString_multiplicity, value,
key)));
00181         } else {
00182             descriptions.addAll(descriptionStrings.keySet());
00183         }
00184
00185         Iterator<String> descriptionIterator = descriptions.iterator();
00186         String[] result = {descriptionIterator.next()};
00187         if (descriptionIterator.hasNext()) {
00188             result[0] = context.getString(R.string.unit_collection_toString_accumulation_word,
descriptionIterator.next(),
00189                 result[0])
00190                 .trim();
00191         }
00192         descriptionIterator.forEachRemaining(description ->
00193             result[0] = context.getString(R.string.unit_collection_toString_accumulation_symbol,
description,
00194                 result[0])
00195                 .trim());
00196         return result[0];
00197     }
00198 }
00199
00200
00209     @NonNull
00210     public String toString(@NonNull Context context) {
00211         String row;
00212         switch (getRow()) {
00213             case RANGE:
00214                 row = context.getString(R.string.unit_toString_range);
00215                 break;
00216             case SIEGE:
00217                 row = context.getString(R.string.unit_toString_siege);
00218                 break;
00219             case MELEE:
00220                 default:
00221                 row = context.getString(R.string.unit_toString_melee);
00222         }
00223         String epic = isEpic() ? context.getString(R.string.unit_toString_epic) :
context.getString(R.string.unit_toString_unit);
00224         String ability;
00225         String squad = "";
00226         switch (getAbility()) {
00227             case HORN:
00228                 ability = context.getString(R.string.add_picker_ability_horn);
00229                 break;
00230             case BINDING:
00231                 ability = context.getString(R.string.add_picker_ability_binding);
00232                 squad = context.getString(R.string.unit_toString_squad, getSquad());
00233                 break;
00234             case MORAL_BOOST:
00235                 ability = context.getString(R.string.add_picker_ability_moralBoost);
00236                 break;

```

```

00237         case REVENGE:
00238             ability = context.getString(R.string.add_picker_ability_revenge);
00239             break;
00240         case NONE:
00241             default:
00242                 ability = context.getString(R.string.unit_toString_ability_none);
00243     }
00244     return context.getString(R.string.unit_toString, row, epic, getDamage(), ability,
squad).trim();
00245 }
00246
00252     public int getId() {
00253         return id;
00254     }
00255
00262     void setId(int id) {
00263         this.id = id;
00264     }
00265
00271     public boolean isEpic() {
00272         return epic;
00273     }
00274
00281     void setEpic(boolean epic) {
00282         this.epic = epic;
00283     }
00284
00291     public int getDamage() {
00292         return damage;
00293     }
00294
00302     void setDamage(@IntRange(from = 0) int damage) {
00303         if (damage < 0) {
00304             throw new IllegalArgumentException("Damage must not be less than zero but is " + damage +
".");
00305         }
00306         this.damage = damage;
00307     }
00308
00314     @NonNull
00315     public Ability getAbility() {
00316         return ability;
00317     }
00318
00325     void setAbility(@NonNull Ability ability) {
00326         this.ability = ability;
00327     }
00328
00334     @Nullable
00335     public Integer getSquad() {
00336         return squad;
00337     }
00338
00346     void setSquad(@IntRange(from = 1) @Nullable Integer squad) {
00347         if (ability != Ability.BINDING && squad != null) {
00348             throw new IllegalArgumentException("Squad must be null or ability must be BINDING but
squad is " + squad + " and ability is " + ability + ".");
00349         }
00350         if (ability == Ability.BINDING && (squad == null || squad < 1)) {
00351             throw new IllegalArgumentException("Squad must not be null or less than one but squad is "
+ squad + ".");
00352         }
00353         this.squad = squad;
00354     }
00355
00361     @NonNull
00362     public RowType getRow() {
00363         return row;
00364     }
00365
00372     void setRow(@NonNull RowType row) {
00373         this.row = row;
00374     }
00375 }

```

6.9 UnitRepository.java

```

00001 package com.peternaggschga.gwent.data;
00002
00003 import androidx.annotation.IntRange;
00004 import androidx.annotation.NonNull;
00005 import androidx.annotation.Nullable;
00006
00007 import java.util.Arrays;
00008 import java.util.Collection;

```

```

00009 import java.util.Iterator;
00010 import java.util.List;
00011 import java.util.concurrent.TimeUnit;
00012
00013 import io.reactivex.rxjava3.android.schedulers.AndroidSchedulers;
00014 import io.reactivex.rxjava3.core.Completable;
00015 import io.reactivex.rxjava3.core.Flowable;
00016 import io.reactivex.rxjava3.core.Single;
00017 import io.reactivex.rxjava3.schedulers.Schedulers;
00018
00025 public class UnitRepository {
00030     @NonNull
00031     private final AppDatabase database;
00032
00041     private UnitRepository(@NonNull AppDatabase database) {
00042         this.database = database;
00043     }
00044
00053     @NonNull
00054     public static Single<UnitRepository> getRepository(@NonNull AppDatabase database) {
00055         UnitRepository repository = new UnitRepository(database);
00056         return repository.initializeRows()
00057             .andThen(Single.just(repository))
00058             .subscribeOn(Schedulers.io())
00059             .observeOn(AndroidSchedulers.mainThread());
00060     }
00061
00068     @NonNull
00069     private Completable initializeRows() {
00070         Completable result = Completable.complete();
00071         for (RowType row : RowType.values()) {
00072             result = result.andThen(database.rows().insertRow(new RowEntity(row)));
00073         }
00074         return result;
00075     }
00076
00085     @NonNull
00086     public Completable reset() {
00087         return reset(null);
00088     }
00089
00098     @NonNull
00099     public Completable reset(@Nullable UnitEntity keptUnit) {
00100         Completable result = database.rows().clearRows().andThen(initializeRows());
00101         if (keptUnit != null) {
00102             result = result.andThen(database.units().insertUnit(keptUnit));
00103         }
00104         return result.subscribeOn(Schedulers.io()).observeOn(AndroidSchedulers.mainThread());
00105     }
00106
00118     @NonNull
00119     private Completable insertUnit(boolean epic, @IntRange(from = 0, to =
UnitEntity.NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) int damage, @NonNull Ability ability,
00120                                     @IntRange(from = 1) @Nullable Integer squad, @NonNull RowType row)
00121     {
00122         if (damage < 0 || damage > UnitEntity.NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) {
00123             throw new IllegalArgumentException("Damage be between 0 and " +
UnitEntity.NON_EPIC_DAMAGE_VALUES_UPPER_BOUND + " but is " + damage + ".");
00124         }
00125         if (epic && Arrays.stream(UnitEntity.EPIC_DAMAGE_VALUES).noneMatch(integer -> integer ==
damage)) {
00126             throw new IllegalArgumentException("Damage of epic units must be in " +
Arrays.toString(UnitEntity.EPIC_DAMAGE_VALUES) + " but is " + damage + ".");
00127         }
00128         if (ability != Ability.BINDING && squad != null) {
00129             throw new IllegalArgumentException("Squad must be null or ability must be BINDING but
squad is " + squad + " and ability is " + ability + ".");
00130         }
00131         if (ability == Ability.BINDING && (squad == null || squad < 1)) {
00132             throw new IllegalArgumentException("Squad must not be null or less than one but squad is "
+ squad + ".");
00133         }
00134         return database.units().insertUnit(epic, damage, ability, squad, row)
00135             .subscribeOn(Schedulers.io())
00136             .observeOn(AndroidSchedulers.mainThread());
00137     }
00152     @NonNull
00153     public Completable insertUnit(boolean epic, @IntRange(from = 0, to =
UnitEntity.NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) int damage, @NonNull Ability ability,
00154                                     @IntRange(from = 1) @Nullable Integer squad, @NonNull RowType row,
00155                                     @IntRange(from = 0) int number) {
00156         Completable result = Completable.complete();
00157         for (int i = 0; i < number; i++) {
00158             result = result.andThen(insertUnit(epic, damage, ability, squad, row));
00159         }
00160         return result.subscribeOn(Schedulers.io()).observeOn(AndroidSchedulers.mainThread());

```

```

00161     }
00162
00169     @NonNull
00170     public Completable switchWeather(@NonNull RowType row) {
00171         return
00172         database.rows().updateWeather(row).subscribeOn(Schedulers.io()).observeOn(AndroidSchedulers.mainThread());
00173     }
00180     @NonNull
00181     public Single<Boolean> isWeather(@NonNull RowType row) {
00182         return
00183         database.rows().isWeather(row).subscribeOn(Schedulers.io()).observeOn(AndroidSchedulers.mainThread());
00184     }
00192     @NonNull
00193     public Flowable<Boolean> isWeatherFlowable(@NonNull RowType row) {
00194         return database.rows()
00195             .isWeatherFlowable(row)
00196             .onBackpressureLatest()
00197             .distinctUntilChanged()
00198             .subscribeOn(Schedulers.io())
00199             .observeOn(AndroidSchedulers.mainThread());
00200     }
00201
00207     @NonNull
00208     public Completable clearWeather() {
00209         return
00210         database.rows().clearWeather().subscribeOn(Schedulers.io()).observeOn(AndroidSchedulers.mainThread());
00211     }
00217     @NonNull
00218     public Completable switchHorn(@NonNull RowType row) {
00219         return
00220         database.rows().updateHorn(row).subscribeOn(Schedulers.io()).observeOn(AndroidSchedulers.mainThread());
00221     }
00228     @NonNull
00229     public Single<Boolean> isHorn(@NonNull RowType row) {
00230         return
00231         database.rows().isHorn(row).subscribeOn(Schedulers.io()).observeOn(AndroidSchedulers.mainThread());
00232     }
00240     @NonNull
00241     public Flowable<Boolean> isHornFlowable(@NonNull RowType row) {
00242         return database.rows()
00243             .isHornFlowable(row)
00244             .onBackpressureLatest()
00245             .distinctUntilChanged()
00246             .subscribeOn(Schedulers.io())
00247             .observeOn(AndroidSchedulers.mainThread());
00248     }
00249
00255     @NonNull
00256     public Completable delete(@NonNull Collection<UnitEntity> units) {
00257         return
00258         database.units().deleteUnits(units).subscribeOn(Schedulers.io()).observeOn(AndroidSchedulers.mainThread());
00259     }
00266     @NonNull
00267     public Completable delete(int id) {
00268         return
00269         database.units().deleteUnit(id).subscribeOn(Schedulers.io()).observeOn(AndroidSchedulers.mainThread());
00270     }
00276     @NonNull
00277     public Completable copy(int id) {
00278         return getUnit(id).flatMapCompletable(unit ->
00279             insertUnit(unit.isEpic(), unit.getDamage(), unit.getAbility(),
00280                 unit.getSquad(), unit.getRow())
00281             .subscribeOn(Schedulers.io())
00282             .observeOn(AndroidSchedulers.mainThread()));
00283     }
00290     @NonNull
00291     public Single<Integer> countUnits(@NonNull RowType row) {
00292         return
00293         database.units().countUnits(row).subscribeOn(Schedulers.io()).observeOn(AndroidSchedulers.mainThread());
00294     }
00301     @NonNull
00302     public Single<Integer> countUnits() {
00303         return
00304         database.units().countUnits().subscribeOn(Schedulers.io()).observeOn(AndroidSchedulers.mainThread());
00305     }
00312     @NonNull
00313     public Single<UnitEntity> getUnit(int id) {

```

```

00314         return
database.units().getUnit(id).subscribeOn(Schedulers.io()).observeOn(AndroidSchedulers.mainThread());
00315     }
00316
00324     @NonNull
00325     public Single<List<UnitEntity>> getUnits(@NonNull RowType row) {
00326         return
database.units().getUnits(row).subscribeOn(Schedulers.io()).observeOn(AndroidSchedulers.mainThread());
00327     }
00328
00337     @NonNull
00338     public Flowable<List<UnitEntity>> getUnitsFlowable(@NonNull RowType row) {
00339         return database.units()
00340             .getUnitsFlowable(row)
00341             .onBackpressureLatest()
00342             .debounce(10, TimeUnit.MILLISECONDS)
00343             .distinctUntilChanged((list1, list2) -> {
00344                 if (list1.size() != list2.size()) {
00345                     return false;
00346                 }
00347                 Iterator<UnitEntity> list1Iterator = list1.iterator();
00348                 return list2.stream().map(UnitEntity::getId).allMatch(id ->
id.equals(list1Iterator.next().getId()));
00349             })
00350             .subscribeOn(Schedulers.io())
00351             .observeOn(AndroidSchedulers.mainThread());
00352     }
00353
00360     @NonNull
00361     public Single<List<UnitEntity>> getUnits() {
00362         return
database.units().getUnits().subscribeOn(Schedulers.io()).observeOn(AndroidSchedulers.mainThread());
00363     }
00364
00369     @NonNull
00370     public Flowable<Boolean> hasNonEpicUnitsFlowable() {
00371         return database.units()
00372             .hasNonEpicUnitsFlowable()
00373             .onBackpressureLatest()
00374             .debounce(10, TimeUnit.MILLISECONDS)
00375             .distinctUntilChanged()
00376             .subscribeOn(Schedulers.io())
00377             .observeOn(AndroidSchedulers.mainThread());
00378     }
00379 }

```

6.10 BurnDialogUseCase.java

```

00001 package com.paternaggschga.gwent.domain.cases;
00002
00003 import android.content.Context;
00004
00005 import androidx.annotation.NonNull;
00006 import androidx.appcompat.app.AlertDialog;
00007
00008 import com.paternaggschga.gwent.GwentApplication;
00009 import com.paternaggschga.gwent.R;
00010 import com.paternaggschga.gwent.data.RowType;
00011 import com.paternaggschga.gwent.data.UnitEntity;
00012 import com.paternaggschga.gwent.data.UnitRepository;
00013 import com.paternaggschga.gwent.domain.damage.DamageCalculator;
00014 import com.paternaggschga.gwent.ui.sounds.SoundManager;
00015
00016 import java.util.ArrayList;
00017 import java.util.Collection;
00018 import java.util.HashMap;
00019 import java.util.List;
00020 import java.util.Map;
00021 import java.util.Objects;
00022
00023 import io.reactivex.rxjava3.core.Single;
00024
00028 public class BurnDialogUseCase {
00036     @NonNull
00037     private static Single<List<UnitEntity>> getBurnUnits(@NonNull UnitRepository repository) {
00038         return repository.getUnits()
00039             .flatMap(units -> {
00040                 if (units.isEmpty()) {
00041                     return Single.just(units);
00042                 }
00043
00044                 Single<Map<RowType, DamageCalculator>> calculators = Single.just(new
HashMap<>(RowType.values().length));
00045                 for (RowType row : RowType.values()) {
00046                     calculators =

```

```

        calculators.zipWith(DamageCalculatorUseCase.getDamageCalculator(repository, row), (calculatorMap,
damageCalculator) -> {
00047             calculatorMap.put(row, damageCalculator);
00048             return calculatorMap;
00049         });
00050     }
00051
00052     return calculators.map(damageCalculators -> {
00053         List<UnitEntity> maxDamageUnits = new ArrayList<>(units.size());
00054         final int[] maxDamage = {0};
00055         units.stream()
00056             .filter(unit -> !unit.isEpic())
00057             .forEach(unit -> {
00058                 int damage =
unit.calculateDamage(Objects.requireNonNull(damageCalculators.get(unit.getRow())));
00059                 if (damage > maxDamage[0]) {
00060                     maxDamage[0] = damage;
00061                     maxDamageUnits.clear();
00062                     maxDamageUnits.add(unit);
00063                 } else if (damage == maxDamage[0]) {
00064                     maxDamageUnits.add(unit);
00065                 }
00066             });
00067             return maxDamageUnits;
00068         });
00069     });
00070 }
00071
00084 @NonNull
00085 public static Single<Boolean> burn(@NonNull Context context, @NonNull SoundManager soundManager) {
00086     return GwentApplication.getRepository(context).flatMap(repository -> burn(context, repository,
soundManager));
00087 }
00088
00100 @NonNull
00101 protected static Single<Boolean> burn(@NonNull Context context, @NonNull UnitRepository
repository, @NonNull SoundManager soundManager) {
00102     return getBurnUnits(repository).flatMap(units -> {
00103         if (units.isEmpty()) {
00104             return Single.just(false);
00105         }
00106         return Single.create(emitter -> new AlertDialog.Builder(context)
00107             .setIconAttribute(android.R.attr.alertDialogIcon)
00108             .setTitle(R.string.alertDialog_burn_title)
00109             .setMessage(context.getString(R.string.alertDialog_burn_msg,
UnitEntity.collectionToString(context, units)))
00110             .setNegativeButton(R.string.alertDialog_burn_negative, (dialog, which) ->
dialog.cancel())
00111             .setPositiveButton(R.string.alertDialog_burn_positive, (dialog, which) -> {
00112                 // noinspection CheckResult, ResultOfMethodCallIgnored
00113                 RemoveUnitsUseCase.remove(context, repository, units,
soundManager).subscribe(() -> emitter.onSuccess(true));
00114             })
00115             .setCancelable(true)
00116             .setOnCancelListener(dialog -> emitter.onSuccess(false))
00117             .create()
00118             .show());
00119         });
00120     }
00121 }

```

6.11 DamageCalculatorUseCase.java

```

00001 package com.paternaggschga.gwent.domain.cases;
00002
00003 import android.util.Pair;
00004
00005 import androidx.annotation.NonNull;
00006
00007 import com.paternaggschga.gwent.data.RowType;
00008 import com.paternaggschga.gwent.data.UnitEntity;
00009 import com.paternaggschga.gwent.data.UnitRepository;
00010 import com.paternaggschga.gwent.domain.damage.DamageCalculator;
00011 import com.paternaggschga.gwent.domain.damage.DamageCalculatorBuildDirector;
00012
00013 import java.util.Collection;
00014
00015 import io.reactivex.rxjava3.core.Single;
00016
00023 public class DamageCalculatorUseCase {
00031     @NonNull
00032     public static Single<DamageCalculator> getDamageCalculator(@NonNull UnitRepository repository,
@NonNull RowType row) {
00033         return repository.isWeather(row)
00034             .zipWith(repository.isHorn(row), Pair::create)

```

```

00035         .zipWith(repository.getUnits(row), (weatherHorn, units) ->
00036             getDamageCalculator(weatherHorn.first, weatherHorn.second, units));
00037     }
00038
00039     @NonNull
00040     public static DamageCalculator getDamageCalculator(boolean weather, boolean horn, @NonNull
00041         Collection<UnitEntity> units) {
00042         return DamageCalculatorBuildDirector.getCalculator(weather, horn, units);
00043     }
00044 }

```

6.12 RemoveUnitsUseCase.java

```

00001 package com.paternaggschga.gwent.domain.cases;
00002
00003 import android.app.Dialog;
00004 import android.content.Context;
00005
00006 import androidx.annotation.IntRange;
00007 import androidx.annotation.NonNull;
00008
00009 import com.paternaggschga.gwent.data.Ability;
00010 import com.paternaggschga.gwent.data.UnitEntity;
00011 import com.paternaggschga.gwent.data.UnitRepository;
00012 import com.paternaggschga.gwent.ui.sounds.SoundManager;
00013
00014 import java.util.Collection;
00015 import java.util.Collections;
00016
00017 import io.reactivex.rxjava3.core.Completable;
00018 import io.reactivex.rxjava3.core.CompletableEmitter;
00019
00020 public class RemoveUnitsUseCase {
00021     @NonNull
00022     public static Completable remove(@NonNull Context context, @NonNull UnitRepository repository,
00023         @NonNull Collection<UnitEntity> units, @NonNull SoundManager
00024         soundManager) {
00025         long revengeUnits = units.stream()
00026             .filter(unit -> unit.getAbility() == Ability.REVENGE)
00027             .count();
00028         if (revengeUnits == 0) {
00029             return repository.delete(units);
00030         }
00031         return Completable.create(emitter ->
00032             getRevengeDialog(context, repository, emitter, units, (int) revengeUnits,
00033                 soundManager).show()
00034         );
00035     }
00036
00037     public static Completable remove(@NonNull Context context, @NonNull UnitRepository repository, int
00038     id, @NonNull SoundManager soundManager) {
00039         return repository.getUnit(id)
00040             .flatMapCompletable(unitEntity ->
00041                 remove(context, repository, Collections.singletonList(unitEntity),
00042                     soundManager));
00043     }
00044
00045     @NonNull
00046     private static Dialog getRevengeDialog(@NonNull Context context, @NonNull UnitRepository
00047     repository,
00048         @NonNull CompletableEmitter emitter, @NonNull
00049         Collection<UnitEntity> units,
00050         @IntRange(from = 1) int revengeUnits, @NonNull SoundManager
00051         soundManager) {
00052         return new RevengeAlertDialogBuilderAdapter(context)
00053             .setPositiveCallback((dialog, which) -> {
00054                 // noinspection CheckResult, ResultOfMethodCallIgnored
00055                 repository.delete(units)
00056                     .andThen(RevengeAlertDialogBuilderAdapter.insertAvengers(repository,
00057                         revengeUnits, soundManager))
00058                     .subscribe(emitter::onComplete);
00059             })
00060             .setNegativeCallback((dialog, which) -> {
00061                 // noinspection CheckResult, ResultOfMethodCallIgnored
00062                 repository.delete(units).subscribe(emitter::onComplete);
00063             })
00064             .create();
00065     }
00066 }

```

6.13 ResetAlertDialogBuilderAdapter.java

```

00001 package com.peternaggschga.gwent.domain.cases;
00002
00003 import android.content.Context;
00004 import android.view.View;
00005 import android.widget.CheckBox;
00006
00007 import androidx.annotation.NonNull;
00008 import androidx.appcompat.app.AlertDialog;
00009
00010 import com.peternaggschga.gwent.R;
00011
00016 class ResetAlertDialogBuilderAdapter {
00020     @NonNull
00021     private final AlertDialog.Builder adapteeBuilder;
00022
00026     @NonNull
00027     private final Callback dialogCallback;
00028
00036     ResetAlertDialogBuilderAdapter(@NonNull Context context, @NonNull Callback dialogCallback) {
00037         this.adapteeBuilder = new AlertDialog.Builder(context)
00038             .setIconAttribute(android.R.attr.alertDialogIcon)
00039             .setTitle(R.string.alertDialog_reset_title)
00040             .setOnCancelListener(dialog -> dialogCallback.reset(false))
00041             .setNegativeButton(R.string.alertDialog_reset_negative, (dialog, which) ->
00042                 dialog.cancel());
00043         this.dialogCallback = dialogCallback;
00044     }
00051     @NonNull
00052     AlertDialog create() {
00053         return adapteeBuilder.create();
00054     }
00055
00063     @NonNull
00064     ResetAlertDialogBuilderAdapter setTrigger(@NonNull ResetDialogUseCase.Trigger trigger) {
00065         adapteeBuilder.setMessage((trigger != ResetDialogUseCase.Trigger.FACTION_SWITCH) ?
00066             R.string.alertDialog_reset_msg_default :
00067             R.string.alertDialog_reset_msg_faction_switch)
00068             .setCancelable(trigger != ResetDialogUseCase.Trigger.FACTION_SWITCH);
00069         return this;
00070     }
00071
00080     @NonNull
00081     ResetAlertDialogBuilderAdapter setMonsterDialog(boolean monsterDialog) {
00082         if (monsterDialog) {
00083             View checkBoxView = View.inflate(adapteeBuilder.getContext(),
00084                 R.layout.alertdialog_checkbox, null);
00085             adapteeBuilder.setView(checkBoxView)
00086                 .setPositiveButton(R.string.alertDialog_reset_positive, (dialog, which) -> {
00087                     CheckBox checkBox = checkBoxView.findViewById(R.id.alertDialog_checkbox);
00088                     dialogCallback.reset(true, checkBox.isChecked());
00089                 });
00089         } else {
00090             adapteeBuilder.setPositiveButton(R.string.alertDialog_reset_positive, ((dialog, which) ->
00091                 dialogCallback.reset(true)));
00092         }
00093         return this;
00094     }
00098     interface Callback {
00107         void reset(boolean resetDecision, boolean keepUnit);
00108
00117         default void reset(boolean resetDecision) {
00118             reset(resetDecision, false);
00119         }
00120     }
00121 }

```

6.14 ResetDialogUseCase.java

```

00001 package com.peternaggschga.gwent.domain.cases;
00002
00003 import static com.peternaggschga.gwent.ui.main.FactionSwitchListener.THEME_MONSTER;
00004 import static com.peternaggschga.gwent.ui.main.FactionSwitchListener.THEME_PREFERENCE_KEY;
00005 import static com.peternaggschga.gwent.ui.main.FactionSwitchListener.THEME_SCOIATAEL;
00006
00007 import android.content.Context;
00008 import android.content.SharedPreferences;
00009 import android.widget.Toast;
00010
00011 import androidx.annotation.NonNull;
00012 import androidx.preference.PreferenceManager;

```



```

00013
00014 import com.paternaggschga.gwent.GwentApplication;
00015 import com.paternaggschga.gwent.R;
00016 import com.paternaggschga.gwent.data.RowType;
00017 import com.paternaggschga.gwent.data.UnitRepository;
00018 import com.paternaggschga.gwent.ui.sounds.SoundManager;
00019
00020 import java.util.Arrays;
00021 import java.util.stream.Collectors;
00022
00023 import io.reactivex.rxjava3.core.Single;
00024
00031 public class ResetDialogUseCase {
00046     @NonNull
00047     public static Single<Boolean> reset(@NonNull Context context, @NonNull Trigger trigger, @NonNull
SoundManager soundManager) {
00048         return GwentApplication.getRepository(context)
00049             .flatMap(repository -> reset(context, repository, trigger, soundManager));
00050     }
00051
00065     @NonNull
00066     protected static Single<Boolean> reset(@NonNull Context context, @NonNull UnitRepository
repository,
00067                                         @NonNull Trigger trigger, @NonNull SoundManager
soundManager) {
00068         return getDialogType(context, repository, trigger).flatMap(dialogType -> {
00069             if (dialogType == DialogType.NONE) {
00070                 return ResetRepositoryUseCase.reset(context, repository,
soundManager).andThen(Single.just(true));
00071             }
00072             return Single.create(emitter -> new ResetAlertDialogBuilderAdapter(context,
(resetDecision, keepUnit) -> {
00073                 if (!resetDecision) {
00074                     emitter.onSuccess(false);
00075                     return;
00076                 }
00077                 // noinspection CheckResult, ResultOfMethodCallIgnored
00078                 ResetRepositoryUseCase.reset(context, repository, keepUnit, soundManager)
00079                     .doAfterTerminate(() -> emitter.onSuccess(true))
00080                     .subscribe(unit ->
00081                         Toast.makeText(context,
00082
context.getString(R.string.alertDialog_factionreset_monster_toast_keep, unit.toString(context)),
00083                             Toast.LENGTH_LONG)
00084                             .show());
00085                 }).setTrigger(trigger)
00086                     .setMonsterDialog(dialogType == DialogType.MONSTER)
00087                     .create()
00088                     .show());
00089             });
00090     }
00091
00101     @NonNull
00102     private static Single<DialogType> getDialogType(@NonNull Context context, @NonNull UnitRepository
repository,
00103                                                     @NonNull Trigger trigger) {
00104         return Single.concat(Arrays.stream(RowType.values()).map(row ->
00105             repository.isWeather(row)
00106                 .concatWith(repository.isHorn(row))
00107                 .any(state -> state)
00108             ).collect(Collectors.toList()).any(state -> state)
00109             .zipWith(repository.getUnits(), (statusEffects, units) -> {
00110                 SharedPreferences preferences =
00111                 PreferenceManager.getDefaultSharedPreferences(context);
00112                 boolean monsterDialog = trigger != Trigger.FACTION_SWITCH;
00113                 monsterDialog &= preferences.getInt(THEME_PREFERENCE_KEY, THEME_SCOIATAEL) ==
THEME_MONSTER;
00114                 monsterDialog &= units.stream().anyMatch(unit -> !unit.isEpic());
00115                 if (monsterDialog) {
00116                     return DialogType.MONSTER;
00117                 }
00118                 boolean defaultWarning = statusEffects || !units.isEmpty();
00119                 defaultWarning &=
00120                 preferences.getBoolean(context.getString(R.string.preference_key_warning),
context.getResources().getBoolean(R.bool.warning_preference_default));
00121                 if (defaultWarning) {
00122                     return DialogType.DEFAULT;
00123                 }
00124                 return DialogType.NONE;
00125             });
00126     }
00130     public enum Trigger {
00134         BUTTON_CLICK,
00140         FACTION_SWITCH
00141     }
00142

```

```

00148     private enum DialogType {
00152         NONE,
00156         DEFAULT,
00161         MONSTER
00162     }
00163 }

```

6.15 ResetRepositoryUseCase.java

```

00001 package com.paternaggschga.gwent.domain.cases;
00002
00003 import android.app.Dialog;
00004 import android.content.Context;
00005
00006 import androidx.annotation.IntRange;
00007 import androidx.annotation.NonNull;
00008 import androidx.annotation.Nullable;
00009
00010 import com.paternaggschga.gwent.data.Ability;
00011 import com.paternaggschga.gwent.data.UnitEntity;
00012 import com.paternaggschga.gwent.data.UnitRepository;
00013 import com.paternaggschga.gwent.ui.sounds.SoundManager;
00014
00015 import java.util.List;
00016 import java.util.Optional;
00017 import java.util.Random;
00018 import java.util.stream.Collectors;
00019
00020 import io.reactivex.rxjava3.core.Completable;
00021 import io.reactivex.rxjava3.core.CompletableEmitter;
00022 import io.reactivex.rxjava3.core.Maybe;
00023
00031 class ResetRepositoryUseCase {
00045     @NonNull
00046     static Maybe<UnitEntity> reset(@NonNull Context context, @NonNull UnitRepository repository,
00047                                   boolean keepUnit, @NonNull SoundManager soundManager) {
00048         return repository.getUnits()
00049             .flatMapMaybe(units -> {
00050                 Optional<UnitEntity> keptUnit = keepUnit ? getRandomUnit(units) :
00051                 Optional.empty();
00052                 long revengeUnits = units.stream()
00053                     .filter(unit -> unit.getAbility() == Ability.REVENGE)
00054                     .count() - (keptUnit.isPresent() && keptUnit.get().getAbility() ==
00055                     Ability.REVENGE ? 1 : 0);
00056                 Completable resultAction = (revengeUnits == 0) ?
00057                     repository.reset(keptUnit.orElse(null)) :
00058                     Completable.create(emitter ->
00059                         getRandomUnit(context, repository, emitter,
00060                         keptUnit.orElse(null), (int) revengeUnits, soundManager).show()
00061                         );
00062                 return resultAction.andThen(Maybe.fromOptional(keptUnit));
00063             });
00064     }
00065
00066     @NonNull
00067     private static Optional<UnitEntity> getRandomUnit(@NonNull List<UnitEntity> units) {
00068         units = units.stream()
00069             .filter(unit -> !unit.isEpic())
00070             .collect(Collectors.toList());
00071         return units.isEmpty() ? Optional.empty() : Optional.of(units.get(new
00072         Random().nextInt(units.size())));
00073     }
00074
00075     @NonNull
00076     private static Dialog getRevengeDialog(@NonNull Context context, @NonNull UnitRepository
00077     repository,
00078     @NonNull CompletableEmitter emitter, @Nullable UnitEntity
00079     keptUnit,
00080     @IntRange(from = 1) int revengeUnits, @NonNull SoundManager
00081     soundManager) {
00082         return new RevengeAlertDialogBuilderAdapter(context)
00083             .setPositiveCallback((dialogInterface, which) -> {
00084                 // noinspection CheckResult, ResultOfMethodCallIgnored
00085                 repository.reset(keptUnit)
00086                     .andThen(RevengeAlertDialogBuilderAdapter.insertAvengers(repository,
00087                     revengeUnits, soundManager))
00088                     .subscribe(emitter::onComplete);
00089             })
00090             .setNegativeCallback(((dialog, which) -> {
00091                 // noinspection CheckResult, ResultOfMethodCallIgnored
00092                 repository.reset(keptUnit).subscribe(emitter::onComplete);
00093             }))
00094             .create();
00095     }
00096 }
00097

```

```

00122     @NonNull
00123     public static CompletableFuture reset(@NonNull Context context, @NonNull UnitRepository repository,
@NonNull SoundManager soundManager) {
00124         return CompletableFuture.fromMaybe(reset(context, repository, false, soundManager));
00125     }
00126 }

```

6.16 RevengeAlertDialogBuilderAdapter.java

```

00001 package com.paternaggschga.gwent.domain.cases;
00002
00003 import android.content.Context;
00004 import android.content.DialogInterface;
00005
00006 import androidx.annotation.IntRange;
00007 import androidx.annotation.NonNull;
00008 import androidx.appcompat.app.AlertDialog;
00009
00010 import com.paternaggschga.gwent.R;
00011 import com.paternaggschga.gwent.data.Ability;
00012 import com.paternaggschga.gwent.data.RowType;
00013 import com.paternaggschga.gwent.data.UnitRepository;
00014 import com.paternaggschga.gwent.ui.sounds.SoundManager;
00015
00016 import io.reactivex.rxjava3.core.CompletableFuture;
00017
00023 class RevengeAlertDialogBuilderAdapter {
00030     private static final boolean AVENGER_EPIC = false;
00031
00038     private static final Ability AVENGER_ABILITY = Ability.NONE;
00039
00046     private static final int AVENGER_DAMAGE = 8;
00047
00054     private static final Integer AVENGER_SQUAD = null;
00055
00062     private static final RowType AVENGER_ROW = RowType.MELEE;
00063
00067     @NonNull
00068     private final AlertDialog.Builder adapteeBuilder;
00069
00076     RevengeAlertDialogBuilderAdapter(@NonNull Context context) {
00077         this.adapteeBuilder = new AlertDialog.Builder(context)
00078             .setIconAttribute(android.R.attr.alertDialogIcon)
00079             .setTitle(R.string.alertDialog_revenge_title)
00080             .setMessage(R.string.alertDialog_revenge_msg)
00081             .setCancelable(false)
00082             .setPositiveButton(R.string.alertDialog_revenge_positive, (dialog, which) ->
dialog.cancel())
00083             .setNegativeButton(R.string.alertDialog_revenge_negative, (dialog, which) ->
dialog.cancel());
00084     }
00085
00095     @NonNull
00096     public static CompletableFuture insertAvengers(@NonNull UnitRepository repository, @IntRange(from = 0)
int numberOfAvengers, @NonNull SoundManager soundManager) {
00097         return repository.insertUnit(AVENGER_EPIC, AVENGER_DAMAGE, AVENGER_ABILITY, AVENGER_SQUAD,
AVENGER_ROW, numberOfAvengers)
00098             .doOnComplete(() -> soundManager.playCardAddSound(AVENGER_ROW, AVENGER_EPIC));
00099     }
00100
00107     @NonNull
00108     AlertDialog create() {
00109         return adapteeBuilder.create();
00110     }
00111
00120     @NonNull
00121     RevengeAlertDialogBuilderAdapter setPositiveCallback(@NonNull DialogInterface.OnClickListener
onPositiveButtonClick) {
00122         adapteeBuilder.setPositiveButton(R.string.alertDialog_revenge_positive,
onPositiveButtonClick);
00123         return this;
00124     }
00125
00132     @NonNull
00133     RevengeAlertDialogBuilderAdapter setNegativeCallback(@NonNull DialogInterface.OnClickListener
onNegativeButtonClick) {
00134         adapteeBuilder.setNegativeButton(R.string.alertDialog_revenge_negative,
onNegativeButtonClick);
00135         return this;
00136     }
00137 }

```

6.17 BondDamageCalculatorDecorator.java

```

00001 package com.paternaggschga.gwent.domain.damage;
00002
00003 import static com.paternaggschga.gwent.domain.damage.DamageCalculator.Color.BUFFED;
00004
00005 import androidx.annotation.IntRange;
00006 import androidx.annotation.NonNull;
00007
00008 import java.util.Map;
00009 import java.util.Objects;
00010
00017 class BondDamageCalculatorDecorator extends DamageCalculatorDecorator {
00022     private final Map<Integer, Integer> idToSquadSize;
00023
00035     BondDamageCalculatorDecorator(@NonNull DamageCalculator component, @NonNull Map<Integer, Integer>
idToSquadSize) {
00036         super(component);
00037         if (idToSquadSize.values().stream().anyMatch(integer -> integer == null || integer < 1)) {
00038             throw new IllegalArgumentException("Map idToSquadSize must not contain non-positive or
null values.");
00039         }
00040         this.idToSquadSize = idToSquadSize;
00041     }
00042
00053     @Override
00054     public int calculateDamage(int id, @IntRange(from = 0) int damage) {
00055         if (damage < 0) {
00056             throw new IllegalArgumentException("Damage must be greater or equal to 0.");
00057         }
00058         return Objects.requireNonNull(idToSquadSize.getDefault(id, 1)) *
component.calculateDamage(id, damage);
00059     }
00060
00071     @Override
00072     public Color isBuffed(int id) {
00073         return (Objects.requireNonNull(idToSquadSize.getDefault(id, 0)) > 1)
? BUFFED
: component.isBuffed(id);
00074     }
00075 }
00076
00077 }

```

6.18 DamageCalculator.java

```

00001 package com.paternaggschga.gwent.domain.damage;
00002
00003 import androidx.annotation.IntRange;
00004
00005 import java.util.Collection;
00006
00017 public interface DamageCalculator {
00030     Color isBuffed(int id);
00031
00043     int calculateDamage(int id, @IntRange(from = 0) int damage);
00044
00049     enum Color {
00053         DEFAULT,
00057         BUFFED,
00061         DEBUFFED
00062     }
00063 }

```

6.19 DamageCalculatorBuildDirector.java

```

00001 package com.paternaggschga.gwent.domain.damage;
00002
00003 import androidx.annotation.NonNull;
00004
00005 import com.paternaggschga.gwent.data.Ability;
00006 import com.paternaggschga.gwent.data.UnitEntity;
00007
00008 import java.util.Collection;
00009 import java.util.HashMap;
00010 import java.util.List;
00011 import java.util.Map;
00012 import java.util.Objects;
00013 import java.util.stream.Collectors;
00014
00021 public class DamageCalculatorBuildDirector {
00030     @NonNull
00031     public static DamageCalculator getCalculator(boolean weather, boolean horn, @NonNull
Collection<UnitEntity> units) {

```

```

00032         DamageCalculatorBuilder builder = new DamageCalculatorBuilder();
00033         builder.setWeather(weather);
00034
00035         if (units.stream().anyMatch(unit -> unit.getAbility() == Ability.BINDING)) {
00036             setSquads(units, builder);
00037         }
00038
00039         if (units.stream().anyMatch(unit -> unit.getAbility() == Ability.MORAL_BOOST)) {
00040             setMoralBoosts(units, builder);
00041         }
00042
00043         if (horn || units.stream().anyMatch(unit -> unit.getAbility() == Ability.HORN)) {
00044             setHorns(horn, units, builder);
00045         }
00046
00047         return builder.getResult();
00048     }
00049
00061     private static void setSquads(@NonNull Collection<UnitEntity> units, @NonNull
DamageCalculatorBuilder builder) {
00062         List<UnitEntity> bindingUnits = units.stream().filter(unit -> unit.getAbility() ==
Ability.BINDING).collect(Collectors.toList());
00063         Map<Integer, Integer> squadToSquadSize = new HashMap<>();
00064         for (UnitEntity unit : bindingUnits) {
00065             squadToSquadSize.putIfAbsent(unit.getSquad(), 0);
00066             squadToSquadSize.put(unit.getSquad(),
Objects.requireNonNull(squadToSquadSize.get(unit.getSquad())) + 1);
00067         }
00068         Map<Integer, Integer> idToSquadSize = new HashMap<>();
00069         for (UnitEntity unit : bindingUnits) {
00070             idToSquadSize.put(unit.getId(), squadToSquadSize.get(unit.getSquad()));
00071         }
00072         builder.setBond(idToSquadSize);
00073     }
00074
00085     private static void setMoralBoosts(@NonNull Collection<UnitEntity> units, @NonNull
DamageCalculatorBuilder builder) {
00086         List<Integer> unitIds = units.stream()
00087             .filter(unit -> unit.getAbility() == Ability.MORAL_BOOST)
00088             .map(UnitEntity::getId)
00089             .collect(Collectors.toList());
00090         builder.setMoral(unitIds);
00091     }
00092
00106     private static void setHorns(boolean horn, @NonNull Collection<UnitEntity> units, @NonNull
DamageCalculatorBuilder builder) {
00107         List<Integer> unitIds = units.stream()
00108             .filter(unit -> unit.getAbility() == Ability.HORN)
00109             .map(UnitEntity::getId)
00110             .collect(Collectors.toList());
00111         if (horn) {
00112             unitIds.add(null);
00113         }
00114         builder.setHorn(unitIds);
00115     }
00116 }

```

6.20 DamageCalculatorBuilder.java

```

00001 package com.paternaggschga.gwent.domain.damage;
00002
00003 import androidx.annotation.NonNull;
00004
00005 import java.util.List;
00006 import java.util.Map;
00007
00019 class DamageCalculatorBuilder {
00024     @NonNull
00025     private DamageCalculator calculator = new WeatherDamageCalculator(false);
00026
00032     void setWeather(boolean weather) {
00033         calculator = new WeatherDamageCalculator(weather);
00034     }
00035
00041     void setBond(@NonNull Map<Integer, Integer> idToSquad) {
00042         calculator = new BondDamageCalculatorDecorator(calculator, idToSquad);
00043     }
00044
00051     void setMoral(@NonNull List<Integer> unitIds) {
00052         calculator = new MoralDamageCalculatorDecorator(calculator, unitIds);
00053     }
00054
00062     void setHorn(@NonNull List<Integer> unitIds) {
00063         calculator = new HornDamageCalculatorDecorator(calculator, unitIds);
00064     }

```

```

00065
00071     @NonNull
00072     DamageCalculator getResult() {
00073         return calculator;
00074     }
00075 }

```

6.21 DamageCalculatorDecorator.java

```

00001 package com.paternaggschga.gwent.domain.damage;
00002
00003 import androidx.annotation.NonNull;
00004
00009 abstract class DamageCalculatorDecorator implements DamageCalculator {
00013     protected final DamageCalculator component;
00014
00020     DamageCalculatorDecorator(@NonNull DamageCalculator component) {
00021         this.component = component;
00022     }
00023 }

```

6.22 HornDamageCalculatorDecorator.java

```

00001 package com.paternaggschga.gwent.domain.damage;
00002
00003 import static com.paternaggschga.gwent.domain.damage.DamageCalculator.Color.BUFFED;
00004
00005 import androidx.annotation.IntRange;
00006 import androidx.annotation.NonNull;
00007
00008 import java.util.List;
00009
00018 class HornDamageCalculatorDecorator extends DamageCalculatorDecorator {
00024     private final List<Integer> unitIds;
00025
00038     HornDamageCalculatorDecorator(@NonNull DamageCalculator component, @NonNull List<Integer> unitIds)
00039     {
00040         super(component);
00041         this.unitIds = unitIds;
00042     }
00049     private boolean doubleDamage(int id) {
00050         return unitIds.contains(null) || !(unitIds.isEmpty() || unitIds.contains(id)) ||
unitIds.size() > 1;
00051     }
00052
00062     @Override
00063     public int calculateDamage(int id, @IntRange(from = 0) int damage) {
00064         if (damage < 0) {
00065             throw new IllegalArgumentException("Damage must be greater or equal to 0 but is " + damage
+ ".");
00066         }
00067         return (doubleDamage(id) ? 2 : 1) * component.calculateDamage(id, damage);
00068     }
00069
00081     @Override
00082     public Color isBuffed(int id) {
00083         return doubleDamage(id) ? BUFFED : component.isBuffed(id);
00084     }
00085 }

```

6.23 MoralDamageCalculatorDecorator.java

```

00001 package com.paternaggschga.gwent.domain.damage;
00002
00003 import static com.paternaggschga.gwent.domain.damage.DamageCalculator.Color.BUFFED;
00004
00005 import androidx.annotation.IntRange;
00006 import androidx.annotation.NonNull;
00007
00008 import java.util.List;
00009
00017 class MoralDamageCalculatorDecorator extends DamageCalculatorDecorator {
00022     private final List<Integer> unitIds;
00023
00036     MoralDamageCalculatorDecorator(@NonNull DamageCalculator component, @NonNull List<Integer>
unitIds) {
00037         super(component);
00038         if (unitIds.contains(null)) {

```

```

00039         throw new IllegalArgumentException("List<Integer> unitIds must not contain null values.");
00040     }
00041     this.unitIds = unitIds;
00042 }
00043
00055 @Override
00056 public int calculateDamage(int id, @IntRange(from = 0) int damage) {
00057     if (damage < 0) {
00058         throw new IllegalArgumentException("Damage must be greater or equal to 0 but is " + damage
+ ".");
00059     }
00060     int componentDamage = component.calculateDamage(id, damage) + unitIds.size();
00061     return unitIds.contains(id) ? componentDamage - 1 : componentDamage;
00062 }
00063
00074 @Override
00075 public Color isBuffed(int id) {
00076     return (!unitIds.isEmpty() && (!unitIds.contains(id) || (unitIds.size() > 1)))
? BUFFED
: component.isBuffed(id);
00079 }
00080 }

```

6.24 WeatherDamageCalculator.java

```

00001 package com.paternaggschga.gwent.domain.damage;
00002
00003 import static com.paternaggschga.gwent.domain.damage.DamageCalculator.Color.DEBUFFED;
00004 import static com.paternaggschga.gwent.domain.damage.DamageCalculator.Color.DEFAULT;
00005
00006 import androidx.annotation.IntRange;
00007
00014 class WeatherDamageCalculator implements DamageCalculator {
00018     private final boolean weather;
00019
00027     WeatherDamageCalculator(boolean weather) {
00028         this.weather = weather;
00029     }
00030
00040 @Override
00041 public int calculateDamage(int id, @IntRange(from = 0) int damage) {
00042     if (damage < 0) {
00043         throw new IllegalArgumentException("Damage must not be less than zero but is " + damage +
".");
00044     }
00045     if (damage == 0) {
00046         return damage;
00047     }
00048     return weather ? 1 : damage;
00049 }
00050
00060 @Override
00061 public Color isBuffed(int id) {
00062     return weather ? DEBUFFED : DEFAULT;
00063 }
00064 }

```

6.25 GwentApplication.java

```

00001 package com.paternaggschga.gwent;
00002
00003 import android.app.Application;
00004 import android.content.Context;
00005
00006 import androidx.annotation.NonNull;
00007 import androidx.room.Room;
00008
00009 import com.paternaggschga.gwent.data.AppDatabase;
00010 import com.paternaggschga.gwent.data.UnitRepository;
00011
00012 import io.reactivex.rxjava3.core.Single;
00013
00018 public class GwentApplication extends Application {
00025     private static UnitRepository repository = null;
00032     private AppDatabase database;
00033
00041     public static Single<UnitRepository> getRepository(@NonNull Context context) {
00042         return ((GwentApplication) context.getApplicationContext()).getRepository();
00043     }
00044
00050 @Override
00051 public void onCreate() {

```

```

00052         super.onCreate();
00053         database = Room.databaseBuilder(this, AppDatabase.class, "database").build();
00054     }
00055
00062     public Single<UnitRepository> getRepository() {
00063         if (repository != null) {
00064             return Single.just(repository);
00065         }
00066         return UnitRepository.getRepository(database).doOnSuccess(unitRepository -> repository =
unitRepository);
00067     }
00068 }

```

6.26 AddCardDialog.java

```

00001 package com.paternaggschga.gwent.ui.dialogs.addcard;
00002
00003 import android.content.Context;
00004 import android.os.Bundle;
00005
00006 import androidx.annotation.NonNull;
00007 import androidx.annotation.Nullable;
00008
00009 import com.paternaggschga.gwent.GwentApplication;
00010 import com.paternaggschga.gwent.R;
00011 import com.paternaggschga.gwent.data.RowType;
00012 import com.paternaggschga.gwent.data.UnitRepository;
00013 import com.paternaggschga.gwent.ui.dialogs.OverlayDialog;
00014 import com.paternaggschga.gwent.ui.dialogs.cards.ShowUnitsDialog;
00015 import com.paternaggschga.gwent.ui.sounds.SoundManager;
00016
00017 import io.reactivex.rxjava3.disposables.CompositeDisposable;
00018
00022 public class AddCardDialog extends OverlayDialog {
00026     @NonNull
00027     private final RowType row;
00028
00034     @NonNull
00035     private final SoundManager soundManager;
00036
00046     @NonNull
00047     private final CompositeDisposable disposables = new CompositeDisposable();
00048
00056     @Nullable
00057     private CardNumberPickerAdapter pickerAdapter = null;
00058
00067     public AddCardDialog(@NonNull ShowUnitsDialog caller, @NonNull SoundManager soundManager) {
00068         this(caller.getContext(), caller.getRow(), soundManager);
00069
00070         setOnDismissListener(dialog -> {
00071             disposables.dispose();
00072             caller.show();
00073         });
00074     }
00075
00084     public AddCardDialog(@NonNull Context context, @NonNull RowType row, @NonNull SoundManager
soundManager) {
00085         super(context, R.layout.popup_add_card, R.id.popup_add_card_cancel_button);
00086         this.row = row;
00087         this.soundManager = soundManager;
00088
00089         setOnDismissListener(dialog -> disposables.dispose());
00090     }
00091
00101     @Override
00102     protected void onCreate(Bundle savedInstanceState) {
00103         super.onCreate(savedInstanceState);
00104
00105         if (pickerAdapter == null) {
00106             disposables.add(
00107                 GwentApplication.getRepository(getContext())
00108                     .flatMap(UnitRepository::getUnits)
00109                     .map(SquadManager::new)
00110                     .map(squadManager ->
00111                         new CardNumberPickerAdapter(findViewById(R.id.card_layout),
squadManager))
00112                     .subscribe(cardNumberPickerAdapter -> pickerAdapter =
cardNumberPickerAdapter)
00113             );
00114         }
00115
00116         findViewById(R.id.popup_add_card_save_button).setOnClickListener(v -> {
00117             if (pickerAdapter != null) {
00118                 disposables.add(pickerAdapter.addSelectedUnits(row).subscribe(epic -> {
00119                     soundManager.playCardAddSound(row, epic);

```



```

00120             dismiss();
00121         });
00122     }
00123 });
00124 }
00125 }

```

6.27 CardNumberPickerAdapter.java

```

00001 package com.peternaggschga.gwent.ui.dialogs.addcard;
00002
00003 import android.os.Handler;
00004 import android.os.Looper;
00005 import android.view.View;
00006 import android.view.ViewGroup;
00007 import android.widget.NumberPicker;
00008
00009 import androidx.annotation.NonNull;
00010
00011 import com.peternaggschga.gwent.GwentApplication;
00012 import com.peternaggschga.gwent.R;
00013 import com.peternaggschga.gwent.data.Ability;
00014 import com.peternaggschga.gwent.data.RowType;
00015 import com.peternaggschga.gwent.data.UnitEntity;
00016
00017 import org.jetbrains.annotations.Contract;
00018
00019 import java.util.Arrays;
00020 import java.util.List;
00021 import java.util.Objects;
00022 import java.util.SortedMap;
00023 import java.util.TreeMap;
00024
00025 import io.reactivex.rxjava3.core.Single;
00026
00033 class CardNumberPickerAdapter {
00037     @NonNull
00038     private static final List<Ability> EPIC_UNIT_ABILITIES = Arrays.asList(Ability.NONE, Ability.HORN,
Ability.MORAL_BOOST);
00039
00043     @NonNull
00044     private final ValuePicker<Boolean> epicPicker;
00045
00049     @NonNull
00050     private final DamageValuePicker damagePicker;
00051
00057     @NonNull
00058     private final ValuePicker<Ability> abilityPicker;
00059
00065     @NonNull
00066     private final NumberPicker squadPicker;
00067
00071     @NonNull
00072     private final SquadManager squadManager;
00073
00077     @NonNull
00078     private final NumberPicker numberPicker;
00079
00089     CardNumberPickerAdapter(@NonNull ViewGroup pickerGroup, @NonNull SquadManager squadManager) {
00090         this.squadManager = squadManager;
00091
00092         SortedMap<Boolean, Integer> epicStringResources = new TreeMap<>();
00093         epicStringResources.put(false, R.string.add_picker_epic_normal);
00094         epicStringResources.put(true, R.string.add_picker_epic_epic);
00095         epicPicker = new
StringValuePicker<>(pickerGroup.findViewById(R.id.popup_add_card_epic_picker),
epicStringResources,
false);
00096
00097         damagePicker = new
DamageValuePicker(pickerGroup.findViewById(R.id.popup_add_card_dmg_picker));
00098
00099         SortedMap<Ability, Integer> abilityStringResources = new TreeMap<>();
00100         abilityStringResources.put(Ability.NONE, R.string.add_picker_ability_default);
00101         abilityStringResources.put(Ability.HORN, R.string.add_picker_ability_horn);
00102         abilityStringResources.put(Ability.REVENGE, R.string.add_picker_ability_revenge);
00103         abilityStringResources.put(Ability.BINDING, R.string.add_picker_ability_binding);
00104         abilityStringResources.put(Ability.MORAL_BOOST, R.string.add_picker_ability_moralBoost);
00105         abilityPicker = new
StringValuePicker<>(pickerGroup.findViewById(R.id.popup_add_card_ability_picker),
abilityStringResources,
Ability.NONE);
00106
00107         squadPicker = pickerGroup.findViewById(R.id.popup_add_card_binding_picker);
00108         squadPicker.setMinValue(1);
00109
00110
00111
00112

```

```

00113         squadPicker.setMaxValue(SquadManager.MAX_NR_SQUADS);
00114
00115         numberPicker = pickerGroup.findViewById(R.id.popup_add_card_number_picker);
00116         numberPicker.setMinValue(1);
00117         numberPicker.setMaxValue(10);
00118
00119         epicPicker.setOnValueChangedListener((picker, oldVal, newVal) -> {
00120             damagePicker.setEpicValues(newVal);
00121             abilityPicker.setSelectableValues(newVal ? EPIC_UNIT_ABILITIES :
Arrays.asList(Ability.values()));
00122         });
00123
00124         abilityPicker.setOnValueChangedListener((picker, oldVal, newVal) -> {
00125             if (newVal == Ability.BINDING) {
00126                 squadPicker.setVisibility(View.VISIBLE);
00127                 squadPicker.setValue(squadManager.getFirstSquadWithMembers());
00128                 squadManager.onSquadChanged(squadManager.getFirstSquadWithMembers(), damagePicker);
00129             } else {
00130                 squadPicker.setVisibility(View.GONE);
00131             }
00132         });
00133
00134         squadPicker.setOnValueChangedListener(
00135             getDelayedOnValueChangeListener((picker, oldVal, newVal) -> {
00136                 if (newVal == picker.getValue()) {
00137                     squadManager.onSquadChanged(newVal, damagePicker);
00138                 }
00139             })
00140         );
00141     }
00142
00150     @NonNull
00151     @Contract(pure = true)
00152     static NumberPicker.OnValueChangeListener getDelayedOnValueChangeListener(@NonNull
NumberPicker.OnValueChangeListener originalListener) {
00153         return (picker, oldVal, newVal) -> new Handler(Looper.getMainLooper()).postDelayed(() -> {
00154             if (newVal == picker.getValue()) {
00155                 originalListener.onValueChange(picker, oldVal, newVal);
00156             }
00157         }, 500);
00158     }
00159
00165     @NonNull
00166     Single<Boolean> addSelectedUnits(@NonNull RowType row) {
00167         // handle delayed events due to delayed OnValueChangeListeners
00168         // delayed changes in damagePicker after hero selection
00169         int damage;
00170         if (epicPicker.getValue()) {
00171             damage = Arrays.stream(UnitEntity.EPIC_DAMAGE_VALUES).anyMatch(integer ->
Objects.equals(integer, damagePicker.getValue())) ? damagePicker.getValue() :
UnitEntity.EPIC_DAMAGE_VALUES[3];
00172         } else {
00173             damage = damagePicker.getValue();
00174         }
00175
00176         // delayed changes in squadPicker after squad selection
00177         Integer squad;
00178         if (abilityPicker.getValue() == Ability.BINDING) {
00179             squad = squadPicker.getVisibility() == View.VISIBLE ? squadPicker.getValue() :
squadManager.getFirstSquadWithMembers();
00180         } else {
00181             squad = null;
00182         }
00183         return GwentApplication.getRepository(numberPicker.getContext())
00184             .flatMapCompletable(repository ->
00185                 repository.insertUnit(epicPicker.getValue(),
00186                     damage,
00187                     abilityPicker.getValue(),
00188                     squad,
00189                     row,
00190                     numberPicker.getValue())
00191             .andThen(Single.just(epicPicker.getValue())));
00192     }
00193 }

```

6.28 DamageValuePicker.java

```

00001 package com.paternaggschga.gwent.ui.dialogs.addcard;
00002
00003
00004 import static com.paternaggschga.gwent.data.UnitEntity.EPIC_DAMAGE_VALUES;
00005 import static com.paternaggschga.gwent.data.UnitEntity.NON_EPIC_DAMAGE_VALUES_UPPER_BOUND;
00006
00007 import android.widget.NumberPicker;
00008

```

```

00009 import androidx.annotation.IntRange;
00010 import androidx.annotation.NonNull;
00011 import androidx.annotation.Nullable;
00012
00013 import java.util.Arrays;
00014 import java.util.Collection;
00015 import java.util.Map;
00016 import java.util.SortedMap;
00017 import java.util.TreeMap;
00018 import java.util.stream.Collectors;
00019 import java.util.stream.IntStream;
00020
00025 class DamageValuePicker extends ValuePicker<Integer> {
00030     private boolean epicValues = false;
00031
00039     DamageValuePicker(@NonNull NumberPicker picker) {
00040         super(picker, Arrays.stream(EPIC_DAMAGE_VALUES).collect(
00041             TreeMap::new,
00042             (map, integer) -> map.put(integer, integer),
00043             Map::putAll)
00044         );
00045         setEpicValues(false);
00046     }
00047
00055     @Override
00056     @NonNull
00057     protected String getDisplayString(@NonNull Integer value) {
00058         if (epicValues && !getDisplayIntegers().containsKey(value)) {
00059             throw new IllegalStateException("When epicValues is true, displayIntegers must contain the
given key but does not contain " + value + ".");
00060         }
00061         return epicValues ? String.valueOf(getDisplayIntegers().get(value)) : String.valueOf(value);
00062     }
00063
00073     @Override
00074     void setValue(@NonNull @IntRange(from = 0, to = NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) Integer value)
    {
00075         if (epicValues) {
00076             super.setValue(value);
00077         } else if (value < 0 || value > NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) {
00078             throw new IllegalStateException("Value must be in [0, " +
NON_EPIC_DAMAGE_VALUES_UPPER_BOUND + "] but is " + value + ".");
00079         } else {
00080             getPicker().setValue(value);
00081         }
00082     }
00083
00088     @NonNull
00089     @Override
00090     Integer getValue() {
00091         return epicValues ? super.getValue() : getPicker().getValue();
00092     }
00093
00100     void setEpicValues(boolean epicValues) {
00101         this.epicValues = epicValues;
00102
00103         if (epicValues) {
00104             super.setSelectableValues(Arrays.asList(EPIC_DAMAGE_VALUES), EPIC_DAMAGE_VALUES[3]);
00105             return;
00106         }
00107         getSelectableValues().clear();
00108         getSelectableValues().addAll(IntStream.rangeClosed(0,
NON_EPIC_DAMAGE_VALUES_UPPER_BOUND).boxed().collect(Collectors.toList()));
00109         getPicker().setDisplayedValues(null);
00110         getPicker().setMaxValue(NON_EPIC_DAMAGE_VALUES_UPPER_BOUND);
00111         getPicker().setValue(5);
00112     }
00113
00120     @Override
00121     void setSelectableValues(@NonNull Collection<Integer> values, @Nullable Integer defaultValue) {
00122     }
00123 }

```

6.29 SquadManager.java

```

00001 package com.peternaggschga.gwent.ui.dialogs.addcard;
00002
00003
00004
00005 import android.content.Context;
00006 import android.widget.Toast;
00007
00008 import androidx.annotation.IntRange;
00009 import androidx.annotation.NonNull;
00010

```

```

00011 import com.paternaggschga.gwent.R;
00012 import com.paternaggschga.gwent.data.Ability;
00013 import com.paternaggschga.gwent.data.UnitEntity;
00014
00015 import java.util.Arrays;
00016 import java.util.List;
00017 import java.util.stream.Collectors;
00018
00024 class SquadManager {
00028     public static final int MAX_NR_SQUADS = 3;
00029
00033     @NonNull
00034     private final SquadState[] states = new SquadState[MAX_NR_SQUADS];
00035
00041     SquadManager(@NonNull List<UnitEntity> units) {
00042         units = units.stream()
00043             .filter(unit -> unit.getAbility() == Ability.BINDING)
00044             .collect(Collectors.toList());
00045         for (int i = 0; i < MAX_NR_SQUADS; i++) {
00046             states[i] = SquadState.getState(i + 1, units);
00047         }
00048     }
00049
00055     int getFirstSquadWithMembers() {
00056         return Arrays.stream(states)
00057             .filter(SquadState::hasMembers)
00058             .findFirst()
00059             .map(SquadState::getSquadNumber)
00060             .orElse(1);
00061     }
00062
00072     void onSquadChanged(@IntRange(from = 1, to = MAX_NR_SQUADS) int newVal, @NonNull DamageValuePicker
picker) {
00073         if (newVal < 1 || newVal > MAX_NR_SQUADS) {
00074             throw new IllegalArgumentException("NewVal must be in [1, " + MAX_NR_SQUADS + "] but is "
+ newVal + ".");
00075         }
00076         SquadState squad = states[newVal - 1];
00077
00078         Context context = picker.getContext();
00079         Toast.makeText(context,
context.getString(R.string.popUp_add_card_binding_count, squad.getSquadNumber(),
squad.getSquadMembers()),
00080             Toast.LENGTH_SHORT).show();
00081
00082         if (squad.hasMembers()) {
00083             picker.setValue(squad.getMemberBaseDamage());
00084         }
00085     }
00086 }
00087 }

```

6.30 SquadState.java

```

00001 package com.paternaggschga.gwent.ui.dialogs.addcard;
00002
00003 import static com.paternaggschga.gwent.data.UnitEntity.NON_EPIC_DAMAGE_VALUES_UPPER_BOUND;
00004 import static com.paternaggschga.gwent.ui.dialogs.addcard.SquadManager.MAX_NR_SQUADS;
00005
00006 import androidx.annotation.IntRange;
00007 import androidx.annotation.NonNull;
00008
00009 import com.paternaggschga.gwent.data.UnitEntity;
00010
00011 import java.util.List;
00012 import java.util.stream.Collectors;
00013
00019 class SquadState {
00026     @IntRange(from = 1, to = MAX_NR_SQUADS)
00027     private final int squadNumber;
00028
00033     @IntRange(from = 0)
00034     private final int squadMembers;
00035
00042     @IntRange(from = 0, to = NON_EPIC_DAMAGE_VALUES_UPPER_BOUND)
00043     private final int memberBaseDamage;
00044
00055     private SquadState(@IntRange(from = 1, to = MAX_NR_SQUADS) int squadNumber, @IntRange(from = 0)
int squadMembers,
00056         @IntRange(from = 0, to = NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) int
memberBaseDamage) {
00057         if (squadNumber < 1 || squadNumber > MAX_NR_SQUADS) {
00058             throw new IllegalArgumentException("SquadNumber must be in [1, " + MAX_NR_SQUADS + "] but
is " + squadNumber + ".");
00059         }
00060         if (squadMembers < 0) {

```

```

00061         throw new IllegalArgumentException("SquadMembers must be non-negative but is " +
squadMembers + ".");
00062     }
00063     if (memberBaseDamage < 0 || memberBaseDamage > NON_EPIC_DAMAGE_VALUES_UPPER_BOUND) {
00064         throw new IllegalArgumentException("MemberBaseDamage must be in [0, " +
NON_EPIC_DAMAGE_VALUES_UPPER_BOUND + "] but is " + memberBaseDamage + ".");
00065     }
00066     this.squadNumber = squadNumber;
00067     this.squadMembers = squadMembers;
00068     this.memberBaseDamage = memberBaseDamage;
00069 }
00070
00071 @NonNull
00072 static SquadState getState(@IntRange(from = 1, to = MAX_NR_SQUADS) int squadNumber,
00073 @NonNull List<UnitEntity> units) {
00074     if (squadNumber < 1 || squadNumber > MAX_NR_SQUADS) {
00075         throw new IllegalArgumentException("SquadNumber must be in [1, " + MAX_NR_SQUADS + "] but
is " + squadNumber + ".");
00076     }
00077     units = units.stream()
00078         .filter(unit -> unit.getSquad() != null && unit.getSquad() == squadNumber)
00079         .collect(Collectors.toList());
00080     return new SquadState(squadNumber,
00081         units.size(),
00082         units.stream().findAny().map(UnitEntity::getDamage).orElse(5));
00083 }
00084
00085 boolean hasMembers() {
00086     return squadMembers > 0;
00087 }
00088
00089 @IntRange(from = 1, to = MAX_NR_SQUADS)
00090 int getSquadNumber() {
00091     return squadNumber;
00092 }
00093
00094 @IntRange(from = 0)
00095 int getSquadMembers() {
00096     return squadMembers;
00097 }
00098
00099 @IntRange(from = 0, to = NON_EPIC_DAMAGE_VALUES_UPPER_BOUND)
00100 int getMemberBaseDamage() {
00101     return memberBaseDamage;
00102 }
00103 }

```

6.31 StringValuePicker.java

```

00001 package com.paternaggschga.gwent.ui.dialogs.addcard;
00002
00003
00004
00005 import android.widget.NumberPicker;
00006
00007 import androidx.annotation.NonNull;
00008 import androidx.annotation.Nullable;
00009
00010 import java.util.SortedMap;
00011
00012 class StringValuePicker<T extends Comparable<T>> extends ValuePicker<T> {
00013     @SuppressWarnings("unused")
00014     StringValuePicker(@NonNull NumberPicker picker, @NonNull SortedMap<T, Integer> valueToStringRes) {
00015         this(picker, valueToStringRes, null);
00016     }
00017
00018     StringValuePicker(@NonNull NumberPicker picker, @NonNull SortedMap<T, Integer> valueToStringRes,
00019 @Nullable T defaultValue) {
00020         super(picker, valueToStringRes, defaultValue);
00021     }
00022
00023     @Override
00024     @NonNull
00025     protected String getDisplayString(@NonNull T value) {
00026         Integer resId;
00027         if ((resId = getDisplayIntegers().get(value)) == null) {
00028             throw new IllegalStateException("Value must be key in displayIntegers but is " + value +
".");
00029         }
00030         return getContext().getString(resId);
00031     }
00032 }

```

6.32 ValuePicker.java

```

00001 package com.peternaggschga.gwent.ui.dialogs.addcard;
00002
00003
00004
00005 import android.content.Context;
00006 import android.widget.NumberPicker;
00007
00008 import androidx.annotation.NonNull;
00009 import androidx.annotation.Nullable;
00010 import androidx.annotation.UiContext;
00011
00012 import java.util.ArrayList;
00013 import java.util.Collection;
00014 import java.util.HashMap;
00015 import java.util.List;
00016 import java.util.Map;
00017 import java.util.SortedMap;
00018
00029 abstract class ValuePicker<T extends Comparable<T> {
00037     @NonNull
00038     private final NumberPicker picker;
00039
00045     @NonNull
00046     private final Map<T, Integer> displayIntegers;
00047
00054     @NonNull
00055     private final List<T> selectableValues;
00056
00065     ValuePicker(@NonNull NumberPicker picker, @NonNull SortedMap<T, Integer> valueToStringRes) {
00066         this(picker, valueToStringRes, null);
00067     }
00068
00081     ValuePicker(@NonNull NumberPicker picker, @NonNull SortedMap<T, Integer> valueToStringRes,
00082                 @Nullable T defaultValue) {
00083         if (valueToStringRes.isEmpty()) {
00084             throw new IllegalArgumentException("SortedMap<T, Integer> valueToStringRes must not be
00085 empty.");
00086         }
00086         if (defaultValue != null && !valueToStringRes.containsKey(defaultValue)) {
00087             throw new IllegalArgumentException("DefaultValue must be null or a key in the given
00088 SortedMap.");
00088         }
00089         this.picker = picker;
00090         displayIntegers = new HashMap<>(valueToStringRes);
00091         selectableValues = new ArrayList<>(valueToStringRes.size());
00092
00093         picker.setMinValue(0);
00094         setSelectableValues(valueToStringRes.keySet(), defaultValue);
00095     }
00096
00107     void setSelectableValues(@NonNull Collection<T> values, @Nullable T defaultValue) {
00108         if (values.isEmpty()) {
00109             throw new IllegalArgumentException("Collection<T> values must not be empty.");
00110         }
00111         if (!displayIntegers.keySet().containsAll(values)) {
00112             throw new IllegalStateException("DisplayIntegers must contain all given values.");
00113         }
00114         if (defaultValue != null && !displayIntegers.containsKey(defaultValue)) {
00115             throw new IllegalArgumentException("DefaultValue must be null or a key in the given
00116 SortedMap.");
00116         }
00117
00118         picker.setDisplayedValues(null);
00119         picker.setValue(0);
00120         picker.setMaxValue(values.size() - 1);
00121
00122         selectableValues.clear();
00123         String[] displayValues = new String[values.size()];
00124         int i = 0;
00125         for (T value : values) {
00126             selectableValues.add(value);
00127             displayValues[i++] = getDisplayString(value);
00128         }
00129
00130         if (defaultValue != null) {
00131             setValue(defaultValue);
00132         }
00133         picker.setDisplayedValues(displayValues);
00134     }
00135
00142     @NonNull
00143     protected NumberPicker getPicker() {
00144         return picker;
00145     }
00146

```

```

00157     void setOnValueChangedListener(@NonNull OnValueChangeListener<T> onValueChangeListener) {
00158         picker.setOnValueChangedListener(
00159             CardNumberPickerAdapter.getDelayedOnValueChangeListener((picker, oldVal, newVal) ->
00160                 onValueChangeListener.onValueChange(ValuePicker.this,
00161                     selectableValues.get(oldVal),
00162                     selectableValues.get(newVal)))
00163         );
00164     }
00165
00170     @NonNull
00171     T getValue() {
00172         return selectableValues.get(picker.getValue());
00173     }
00174
00180     void setValue(@NonNull T value) {
00181         int position;
00182         if ((position = selectableValues.indexOf(value)) < 0) {
00183             throw new IllegalStateException("SelectableValues must contain the given value.");
00184         }
00185         picker.setValue(position);
00186     }
00187
00194     @NonNull
00195     protected abstract String getDisplayString(@NonNull T value);
00196
00203     @UiThread
00204     public Context getContext() {
00205         return picker.getContext();
00206     }
00207
00211     interface OnValueChangeListener<T extends Comparable<T>> {
00219         void onValueChange(@NonNull ValuePicker<T> picker, T oldVal, T newVal);
00220     }
00221
00228     @NonNull
00229     protected Map<T, Integer> getDisplayIntegers() {
00230         return displayIntegers;
00231     }
00232
00239     @NonNull
00240     protected List<T> getSelectableValues() {
00241         return selectableValues;
00242     }
00243
00254     void setSelectableValues(@NonNull Collection<T> values) {
00255         setSelectableValues(values, null);
00256     }
00257 }

```

6.33 CardListAdapter.java

```

00001 package com.paternaggschga.gwent.ui.dialogs.cards;
00002
00003 import static androidx.recyclerview.widget.RecyclerView.NO_ID;
00004
00005 import android.view.LayoutInflater;
00006 import android.view.View;
00007 import android.view.ViewGroup;
00008 import android.widget.ImageView;
00009 import android.widget.TextView;
00010
00011 import androidx.annotation.IntRange;
00012 import androidx.annotation.NonNull;
00013 import androidx.core.util.Consumer;
00014 import androidx.recyclerview.widget.DiffUtil;
00015 import androidx.recyclerview.widget.ListAdapter;
00016 import androidx.recyclerview.widget.RecyclerView;
00017
00018 import com.paternaggschga.gwent.R;
00019
00028 class CardListAdapter extends ListAdapter<CardUiState, CardListAdapter.CardViewHolder> {
00038     @NonNull
00039     private final Consumer<Integer> onCopy;
00040
00049     @NonNull
00050     private final Consumer<Integer> onRemove;
00051
00065     CardListAdapter(@NonNull Consumer<Integer> onCopy, @NonNull Consumer<Integer> onRemove) {
00066         super(CardUiState.DIFF_CALLBACK);
00067         this.onCopy = onCopy;
00068         this.onRemove = onRemove;
00069         setHasStableIds(true);
00070     }
00071
00079     @NonNull

```

```

00080     @Override
00081     public CardViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
00082         View cardView = LayoutInflater.from(parent.getContext()).inflate(R.layout.card, parent,
false);
00083         return new CardViewHolder(cardView, onCopy, onRemove);
00084     }
00085
00092     @Override
00093     public void onBindViewHolder(@NonNull CardViewHolder holder, @IntRange(from = 0) int position) {
00094         CardUiState item = getItem(position);
00095         holder.getDamageView().setText(item.getDamageString());
00096         holder.getDamageView().setBackgroundResource(item.getDamageBackgroundImageId());
00097         holder.getDamageView().setTextColor(item.getDamageTextColor());
00098
00099         if (item.showAbility()) {
00100             holder.getAbilityView().setImageResource(item.getAbilityImageId());
00101             holder.getAbilityView().setVisibility(View.VISIBLE);
00102         } else {
00103             holder.getAbilityView().setVisibility(View.GONE);
00104         }
00105
00106         if (item.showSquad()) {
00107             holder.getBindingView().setText(item.getSquadString());
00108             holder.getBindingView().setVisibility(View.VISIBLE);
00109         } else {
00110             holder.getBindingView().setVisibility(View.GONE);
00111         }
00112
00113         holder.setItemId(item.getUnitId());
00114     }
00121
00122     @Override
00123     public long getItemId(@IntRange(from = 0) int position) {
00124         return getItem(position).getUnitId();
00125     }
00131
00132     static class CardViewHolder extends RecyclerView.ViewHolder {
00133         @NonNull
00134         private final TextView damageView;
00143         @NonNull
00144         private final ImageView abilityView;
00151         @NonNull
00152         private final TextView bindingView;
00161         private int itemId = (int) NO_ID;
00162
00173         CardViewHolder(@NonNull View itemView,
00174             @NonNull Consumer<Integer> onCopy, @NonNull Consumer<Integer> onRemove) {
00175             super(itemView);
00176
00177             damageView = itemView.findViewById(R.id.damageView);
00178             abilityView = itemView.findViewById(R.id.abilityView);
00179             bindingView = itemView.findViewById(R.id.bindingView);
00180
00181             itemView.findViewById(R.id.copyButton).setOnClickListener(v -> {
00182                 if (itemId <= NO_ID) {
00183                     throw new IllegalArgumentException("ItemId must be greater than NO_ID (" + NO_ID +
") but is " + itemId + ".");
00184                 }
00185                 onCopy.accept(itemId);
00186             });
00187             itemView.findViewById(R.id.deleteButton).setOnClickListener(v -> {
00188                 if (itemId <= NO_ID) {
00189                     throw new IllegalArgumentException("ItemId must be greater than NO_ID (" + NO_ID +
") but is " + itemId + ".");
00190                 }
00191                 onRemove.accept(itemId);
00192             });
00193         }
00194
00200         @NonNull
00201         TextView getDamageView() {
00202             return damageView;
00203         }
00204
00210         @NonNull
00211         ImageView getAbilityView() {
00212             return abilityView;
00213         }
00214
00220         @NonNull
00221         TextView getBindingView() {
00222             return bindingView;
00223         }
00224
00232         void setItemId(int itemId) {
00233             if (itemId <= NO_ID) {
00234                 throw new IllegalArgumentException("ItemId must be greater than NO_ID (-1) but is " +

```



```

        itemId + ".");
00235     }
00236     this.itemId = itemId;
00237 }
00238 }
00239 }

```

6.34 CardUiState.java

```

00001 package com.paternaggschga.gwent.ui.dialogs.cards;
00002
00003
00004
00005 import androidx.annotation.ColorInt;
00006 import androidx.annotation.DrawableRes;
00007 import androidx.annotation.IntRange;
00008 import androidx.annotation.NonNull;
00009 import androidx.annotation.Nullable;
00010 import androidx.recyclerview.widget.DiffUtil;
00011
00012 import com.paternaggschga.gwent.data.UnitEntity;
00013 import com.paternaggschga.gwent.domain.damage.DamageCalculator;
00014
00015 import java.util.Objects;
00016
00024 public class CardUiState {
00032     @NonNull
00033     public static final DiffUtil.ItemCallback<CardUiState> DIFF_CALLBACK = new
DiffUtil.ItemCallback<CardUiState>() {
00034         @Override
00035         public boolean areItemsTheSame(@NonNull CardUiState oldItem, @NonNull CardUiState newItem) {
00036             return oldItem.unitId == newItem.unitId;
00037         }
00038
00039         @Override
00040         public boolean areContentsTheSame(@NonNull CardUiState oldItem, @NonNull CardUiState newItem)
{
00041             return Objects.equals(oldItem, newItem);
00042         }
00043     };
00044
00050     public static final int UNUSED = -1;
00051
00057     private final int unitId;
00058
00063     @DrawableRes
00064     private final int damageBackgroundImageId;
00065
00071     @NonNull
00072     private final String damageString;
00073
00078     @ColorInt
00079     private final int damageTextColor;
00080
00087     @DrawableRes
00088     private final int abilityImageId;
00089
00094     @NonNull
00095     private final String squadString;
00096
00108     public CardUiState(int unitId, @DrawableRes int damageBackgroundImageId, @IntRange(from = UNUSED)
int damage,
00109         @ColorInt int damageTextColor, @DrawableRes int abilityImageId,
00110         @Nullable @IntRange(from = 1) Integer squad) {
00111         if (damage < UNUSED) {
00112             throw new IllegalArgumentException("Damage must be greater or equal to UNUSED (" + UNUSED
+ ") but is " + damage + ".");
00113         }
00114         if (squad != null && squad < 1) {
00115             throw new IllegalArgumentException("Squad must not be null or less than 1 but is " + squad
+ ".");
00116         }
00117         this.unitId = unitId;
00118         this.damageBackgroundImageId = damageBackgroundImageId;
00119         this.damageString = damage == UNUSED ? "" : String.valueOf(damage);
00120         this.damageTextColor = damageTextColor;
00121         this.abilityImageId = abilityImageId;
00122         this.squadString = squad == null ? "" : String.valueOf(squad);
00123     }
00124
00130     public boolean showAbility() {
00131         return abilityImageId != UNUSED;
00132     }
00133
00139     public boolean showSquad() {

```

```

00140         return showAbility() && !squadString.isEmpty();
00141     }
00142
00148     public int getUnitId() {
00149         return unitId;
00150     }
00151
00157     @DrawableRes
00158     public int getDamageBackgroundImageId() {
00159         return damageBackgroundImageId;
00160     }
00161
00167     @NonNull
00168     public String getDamageString() {
00169         return damageString;
00170     }
00171
00177     @ColorInt
00178     public int getDamageTextColor() {
00179         return damageTextColor;
00180     }
00181
00188     @DrawableRes
00189     public int getAbilityImageId() {
00190         return abilityImageId;
00191     }
00192
00199     @NonNull
00200     public String getSquadString() {
00201         return squadString;
00202     }
00203
00211     @Override
00212     public boolean equals(Object o) {
00213         if (this == o) return true;
00214         if (!(o instanceof CardUiState)) return false;
00215         CardUiState that = (CardUiState) o;
00216         return damageBackgroundImageId == that.damageBackgroundImageId
00217             && damageTextColor == that.damageTextColor
00218             && abilityImageId == that.abilityImageId
00219             && Objects.equals(damageString, that.damageString)
00220             && Objects.equals(squadString, that.squadString);
00221     }
00222 }

```

6.35 CardUiStateFactory.java

```

00001 package com.paternaggschga.gwent.ui.dialogs.cards;
00002
00003 import static com.paternaggschga.gwent.domain.damage.DamageCalculator.Color;
00004 import static com.paternaggschga.gwent.domain.damage.DamageCalculator.Color.BUFFED;
00005 import static com.paternaggschga.gwent.domain.damage.DamageCalculator.Color.DEBUFFED;
00006 import static com.paternaggschga.gwent.domain.damage.DamageCalculator.Color.DEFAULT;
00007
00008 import android.content.Context;
00009
00010 import androidx.annotation.NonNull;
00011
00012 import com.paternaggschga.gwent.R;
00013 import com.paternaggschga.gwent.data.UnitEntity;
00014 import com.paternaggschga.gwent.domain.cases.DamageCalculatorUseCase;
00015 import com.paternaggschga.gwent.domain.damage.DamageCalculator;
00016
00017 import java.util.Collection;
00018 import java.util.HashMap;
00019 import java.util.List;
00020 import java.util.Map;
00021 import java.util.Objects;
00022 import java.util.stream.Collectors;
00023
00028 public class CardUiStateFactory {
00033     private final boolean weather;
00034
00039     private final boolean horn;
00040
00045     @NonNull
00046     private final Map<Color, Integer> damageTextColors = new HashMap<> (Color.values().length);
00047
00058     public CardUiStateFactory(@NonNull Context context, boolean weather, boolean horn) {
00059         this.weather = weather;
00060         this.horn = horn;
00061         damageTextColors.put(DEFAULT, context.getColor(R.color.color_damage_textColor));
00062         damageTextColors.put(BUFFED, context.getColor(R.color.color_damage_textColor_buffed));
00063         damageTextColors.put(DEBUFFED, context.getColor(R.color.color_damage_textColor_debuffed));
00064     }

```

```

00065
00073     @NonNull
00074     public List<CardUiState> createCardUiState(@NonNull Collection<UnitEntity> units) {
00075         DamageCalculator calculator = DamageCalculatorUseCase.getDamageCalculator(weather, horn,
units);
00076         return units.stream()
00077             .map(unit -> createCardUiState(unit, calculator))
00078             .collect(Collectors.toList());
00079     }
00080
00088     @NonNull
00089     public CardUiState createCardUiState(@NonNull UnitEntity unit, @NonNull DamageCalculator
calculator) {
00090         int damageBackgroundId = R.drawable.icon_damage_background;
00091         int damage = unit.calculateDamage(calculator);
00092
00093         if (unit.isEpic()) {
00094             switch (damage) {
00095                 case 7:
00096                     damageBackgroundId = R.drawable.icon_epic_damage_7;
00097                     break;
00098                 case 8:
00099                     damageBackgroundId = R.drawable.icon_epic_damage_8;
00100                     break;
00101                 case 10:
00102                     damageBackgroundId = R.drawable.icon_epic_damage_10;
00103                     break;
00104                 case 11:
00105                     damageBackgroundId = R.drawable.icon_epic_damage_11;
00106                     break;
00107                 case 15:
00108                     damageBackgroundId = R.drawable.icon_epic_damage_15;
00109                     break;
00110                 case 0:
00111                 default:
00112                     damageBackgroundId = R.drawable.icon_epic_damage_0;
00113             }
00114             damage = CardUiState.UNUSED;
00115         }
00116
00117         int damageColor = Objects.requireNonNull(damageTextColors.get(unit.isBuffed(calculator)));
00118
00119         int abilityImage;
00120         switch (unit.getAbility()) {
00121             case HORN:
00122                 abilityImage = R.drawable.icon_horn;
00123                 break;
00124             case REVENGE:
00125                 abilityImage = R.drawable.icon_revenge;
00126                 break;
00127             case BINDING:
00128                 abilityImage = R.drawable.icon_binding;
00129                 break;
00130             case MORAL_BOOST:
00131                 abilityImage = R.drawable.icon_moral_boost;
00132                 break;
00133             case NONE:
00134             default:
00135                 abilityImage = CardUiState.UNUSED;
00136         }
00137
00138         return new CardUiState(unit.getId(), damageBackgroundId, damage, damageColor, abilityImage,
unit.getSquad());
00139     }
00140 }

```

6.36 ShowUnitsDialog.java

```

00001 package com.peternaggschga.gwent.ui.dialogs.cards;
00002
00003 import android.content.Context;
00004 import android.os.Bundle;
00005 import android.util.Log;
00006
00007 import androidx.annotation.NonNull;
00008 import androidx.recyclerview.widget.DefaultItemAnimator;
00009 import androidx.recyclerview.widget.LinearLayoutManager;
00010 import androidx.recyclerview.widget.RecyclerView;
00011
00012 import com.peternaggschga.gwent.GwentApplication;
00013 import com.peternaggschga.gwent.R;
00014 import com.peternaggschga.gwent.data.RowType;
00015 import com.peternaggschga.gwent.data.UnitEntity;
00016 import com.peternaggschga.gwent.domain.cases.RemoveUnitsUseCase;
00017 import com.peternaggschga.gwent.ui.dialogs.OverlayDialog;

```

```

00018 import com.paternaggschga.gwent.ui.dialogs.addcard.AddCardDialog;
00019 import com.paternaggschga.gwent.ui.sounds.SoundManager;
00020
00021 import java.util.Objects;
00022
00023 import io.reactivex.rxjava3.core.Single;
00024 import io.reactivex.rxjava3.disposables.CompositeDisposable;
00025
00031 public class ShowUnitsDialog extends OverlayDialog {
00037     @NonNull
00038     private final RowType row;
00039
00043     @NonNull
00044     private final CardListAdapter cardListAdapter;
00045
00052     @NonNull
00053     private final SoundManager soundManager;
00054
00062     @NonNull
00063     private final CompositeDisposable disposables = new CompositeDisposable();
00064
00068     private RecyclerView unitRecyclerView;
00069
00078     private ShowUnitsDialog(@NonNull Context context, @NonNull RowType row,
00079                             @NonNull CardListAdapter cardListAdapter, @NonNull SoundManager
00080                             soundManager) {
00081         super(context, R.layout.popup_cards, R.id.popup_cards_cancel_button);
00082         this.row = row;
00083         this.cardListAdapter = cardListAdapter;
00084         this.soundManager = soundManager;
00085     }
00093     @NonNull
00094     public static Single<ShowUnitsDialog> getDialog(@NonNull Context context, @NonNull RowType row,
00095     @NonNull SoundManager soundManager) {
00096         return GwentApplication.getRepository(context)
00097             .flatMap(repository -> repository.isWeather(row)
00098                 .zipWith(repository.isHorn(row), (weather, horn) ->
00099                     new CardUiStateFactory(context, weather, horn))
00100                 .map(factory -> {
00101                     CompositeDisposable initialDisposables = new CompositeDisposable();
00102                     CardListAdapter adapter = new CardListAdapter(
00103                         id -> initialDisposables.add(
00104                             repository
00105                                 .copy(id)
00106                                 .andThen(repository.getUnit(id))
00107                                 .map(UnitEntity::isEpic)
00108                                 .subscribe(epic ->
00109                                     soundManager.playCardAddSound(row, epic))
00110                                     ),
00111                         id -> initialDisposables.add(
00112                             RemoveUnitsUseCase.remove(context, repository, id,
00113                                 soundManager)
00114                                     .subscribe(soundManager::playCardRemovedSound,
00115                                         throwable ->
00116                                             Log.e(ShowUnitsDialog.class.getSimpleName(),
00117                                                 "There has been an error with the
00118                                                 removal of a unit. " +
00119                                                 "A reason might be tapping
00120                                                 delete buttons too fast!",
00121                                                 throwable))
00122                                     );
00123                     initialDisposables.add(
00124                         repository.getUnitsFlowable(row)
00125                             .map(factory::createCardUiState)
00126                             .subscribe(adapter::submitList)
00127                     );
00128                     ShowUnitsDialog result = new ShowUnitsDialog(context, row, adapter,
00129                         soundManager);
00130                     result.disposables.add(initialDisposables);
00131                     return result;
00132                 })
00133             );
00134     }
00141     @Override
00142     protected void onCreate(Bundle savedInstanceState) {
00143         super.onCreate(savedInstanceState);
00144
00145         if (unitRecyclerView == null) {
00146             LinearLayoutManager llm = new LinearLayoutManager(getContext());
00147             llm.setOrientation(RecyclerView.HORIZONTAL);
00148             unitRecyclerView = findViewById(R.id.cards_list);
00149             (DefaultItemAnimator)
00150                 Objects.requireNonNull(unitRecyclerView.getItemAnimator()).setSupportsChangeAnimations(false);
00151             unitRecyclerView.setLayoutManager(llm);

```

```

00151         unitRecyclerView.setAdapter(cardListAdapter);
00152
00153         cardListAdapter.registerAdapterDataObserver(new RecyclerView.AdapterDataObserver() {
00162             @Override
00163             public void onItemRangeInserted(int positionStart, int itemCount) {
00164                 super.onItemRangeInserted(positionStart, itemCount);
00165                 int scrollPosition = positionStart + itemCount - 1;
00166                 if (scrollPosition >= 0) {
00167                     unitRecyclerView.smoothScrollToPosition(scrollPosition);
00168                 }
00169             }
00170         });
00171     }
00172
00173     findViewById(R.id.popup_cards_add_button).setOnClickListener(v -> {
00174         hide();
00175         new AddCardDialog(ShowUnitsDialog.this, soundManager).show();
00176     });
00177
00178     setOnDismissListener(dialog -> disposables.dispose());
00179 }
00180
00181 @NonNull
00182 public RowType getRow() {
00183     return row;
00184 }
00185 }
00186 }
00187 }
00188 }
00189 }
00190 }
00191 }

```

6.37 ChangeFactionDialog.java

```

00001 package com.paternaggschga.gwent.ui.dialogs;
00002
00003 import static com.paternaggschga.gwent.ui.main.FactionSwitchListener.THEME_MONSTER;
00004 import static com.paternaggschga.gwent.ui.main.FactionSwitchListener.THEME_NILFGAARD;
00005 import static com.paternaggschga.gwent.ui.main.FactionSwitchListener.THEME_NORTHERN_KINGDOMS;
00006 import static com.paternaggschga.gwent.ui.main.FactionSwitchListener.THEME_SCOIATAEL;
00007
00008 import android.content.Context;
00009 import android.os.Bundle;
00010 import android.view.View;
00011
00012 import androidx.annotation.IntRange;
00013 import androidx.annotation.NonNull;
00014
00015 import com.paternaggschga.gwent.R;
00016
00020 public class ChangeFactionDialog extends OverlayDialog {
00024     private final Callback callback;
00025
00033     public ChangeFactionDialog(@NonNull Context context, @NonNull Callback callback) {
00034         super(context, R.layout.popup_faction, R.id.factionBackground);
00035         this.callback = callback;
00036     }
00037
00046     @Override
00047     protected void onCreate(Bundle savedInstanceState) {
00048         super.onCreate(savedInstanceState);
00049
00050         findViewById(R.id.monsterCardView).setOnClickListener(getOnThemeClickListener(THEME_MONSTER));
00051         findViewById(R.id.monsterButton).setOnClickListener(getOnThemeClickListener(THEME_MONSTER));
00052
00053         findViewById(R.id.nilfgaardCardView).setOnClickListener(getOnThemeClickListener(THEME_NILFGAARD));
00054         findViewById(R.id.nilfgaardButton).setOnClickListener(getOnThemeClickListener(THEME_NILFGAARD));
00055
00056         findViewById(R.id.northernKingdomsCardView).setOnClickListener(getOnThemeClickListener(THEME_NORTHERN_KINGDOMS));
00057         findViewById(R.id.northernKingdomsButton).setOnClickListener(getOnThemeClickListener(THEME_NORTHERN_KINGDOMS));
00058
00059         findViewById(R.id.scoiataelCardView).setOnClickListener(getOnThemeClickListener(THEME_SCOIATAEL));
00060         findViewById(R.id.scoiataelButton).setOnClickListener(getOnThemeClickListener(THEME_SCOIATAEL));
00061     }
00062
00072     @NonNull
00073     private View.OnClickListener getOnThemeClickListener(@IntRange(from = THEME_MONSTER, to =
00074     THEME_SCOIATAEL) int theme) {
00074         return v -> {
00075             cancel();
00076             callback.onThemeSelect(theme);
00077         };
00078     }
00079 }

```

```

00079
00084     public interface Callback {
00090         void onThemeSelect(@IntRange(from = THEME_MONSTER, to = THEME_SCOIATAEL) int theme);
00091     }
00092 }

```

6.38 CoinFlipDialog.java

```

00001 package com.paternaggschga.gwent.ui.dialogs;
00002
00003 import android.content.Context;
00004 import android.os.Bundle;
00005 import android.widget.ImageView;
00006 import android.widget.TextView;
00007
00008 import androidx.annotation.NonNull;
00009
00010 import com.paternaggschga.gwent.R;
00011
00012 import java.util.Random;
00013
00024 public class CoinFlipDialog extends OverlayDialog {
00028     private static final Random RANDOM = new Random();
00029
00034     public CoinFlipDialog(@NonNull Context context) {
00035         super(context, R.layout.popup_coin_normal, R.id.coinflipBackground);
00036     }
00037
00047     @Override
00048     protected void onCreate(Bundle savedInstanceState) {
00049         super.onCreate(savedInstanceState);
00050
00051         switch (RANDOM.nextInt(100)) {
00052             case 0:
00053                 setContentView(R.layout.popup_coin_stewie);
00054                 break;
00055             case 1:
00056                 setContentView(R.layout.popup_coin_terry);
00057                 break;
00058             case 2:
00059                 setContentView(R.layout.popup_coin_vin);
00060                 break;
00061             default:
00062                 if (RANDOM.nextBoolean()) {
00063                     ((ImageView)
00064 findViewById(R.id.popup_coin_normal_coinView)).setImageResource(R.drawable.coin_lose);
00064                     ((TextView)
00065 findViewById(R.id.popup_coin_normal_textView)).setText(R.string.popUp_coin_normal_lose);
00065                 }
00066             }
00067         }
00068     }

```

6.39 OverlayDialog.java

```

00001 package com.paternaggschga.gwent.ui.dialogs;
00002
00003 import android.app.Dialog;
00004 import android.content.Context;
00005 import android.graphics.Color;
00006 import android.graphics.drawable.ColorDrawable;
00007 import android.os.Bundle;
00008 import android.view.ViewGroup;
00009 import android.view.Window;
00010
00011 import androidx.annotation.IdRes;
00012 import androidx.annotation.LayoutRes;
00013 import androidx.annotation.NonNull;
00014
00015 import java.util.Objects;
00016
00022 public class OverlayDialog extends Dialog {
00028     static final int NO_DISMISS_VIEW = -1;
00032     @NonNull
00033     private static final ColorDrawable BACKGROUND = new ColorDrawable(Color.TRANSPARENT);
00037     @LayoutRes
00038     private final int layout;
00044     @IdRes
00045     private final int dismissViewId;
00046
00054     protected OverlayDialog(@NonNull Context context, @LayoutRes int layout, @IdRes int dismissViewId)
00055     {

```

```

00055         super(context);
00056         this.layout = layout;
00057         this.dismissViewId = dismissViewId;
00058     }
00059
00067     @SuppressWarnings("unused")
00068     protected OverlayDialog(@NonNull Context context, @LayoutRes int layout) {
00069         this(context, layout, NO_DISMISS_VIEW);
00070     }
00071
00084     @Override
00085     protected void onCreate(Bundle savedInstanceState) {
00086         super.onCreate(savedInstanceState);
00087
00088         setContentView(layout);
00089
00090         Window window = Objects.requireNonNull(getWindow());
00091         window.setLayout(ViewGroup.LayoutParams.MATCH_PARENT, ViewGroup.LayoutParams.MATCH_PARENT);
00092         window.setBackgroundDrawable(BACKGROUND);
00093
00094         if (dismissViewId != NO_DISMISS_VIEW) {
00095             findViewById(dismissViewId).setOnClickListener(v -> dismiss());
00096         }
00097         setCancelable(true);
00098     }
00099 }

```

6.40 IndicatorManager.java

```

00001 package com.paternaggschga.gwent.ui.introduction;
00002
00003 import android.view.Window;
00004 import android.widget.ImageView;
00005
00006 import androidx.annotation.IntRange;
00007 import androidx.annotation.NonNull;
00008
00009 import com.paternaggschga.gwent.R;
00010
00015 class IndicatorManager {
00019     @NonNull
00020     private final ImageView[] indicators = new ImageView[5];
00021
00029     @IntRange(from = 0, to = 4)
00030     private int currentView = 0;
00031
00043     IndicatorManager(@NonNull Window window) {
00044         indicators[0] = window.findViewById(R.id.introduction_indicator_0);
00045         indicators[1] = window.findViewById(R.id.introduction_indicator_1);
00046         indicators[2] = window.findViewById(R.id.introduction_indicator_2);
00047         indicators[3] = window.findViewById(R.id.introduction_indicator_3);
00048         indicators[4] = window.findViewById(R.id.introduction_indicator_4);
00049     }
00050
00059     void updateIndicators(@IntRange(from = 0, to = 4) int position) {
00060         indicators[currentView].setImageResource(R.drawable.indicator_unselected);
00061         indicators[position].setImageResource(R.drawable.indicator_selected);
00062         currentView = position;
00063     }
00064 }

```

6.41 IntroductionActivity.java

```

00001 package com.paternaggschga.gwent.ui.introduction;
00002
00003 import android.os.Bundle;
00004 import android.view.View;
00005 import android.widget.Button;
00006 import android.widget.ImageButton;
00007
00008 import androidx.activity.OnBackPressedCallback;
00009 import androidx.appcompat.app.AppCompatActivity;
00010 import androidx.preference.PreferenceManager;
00011 import androidx.viewpager2.widget.ViewPager2;
00012
00013 import com.paternaggschga.gwent.R;
00014
00020 public class IntroductionActivity extends AppCompatActivity {
00025     private IndicatorManager indicatorManager;
00026
00036     @Override
00037     protected void onCreate(Bundle savedInstanceState) {

```

```

00038         super.onCreate(savedInstanceState);
00039
00040         setContentView(R.layout.activity_introduction);
00041
00042         if (indicatorManager == null) {
00043             indicatorManager = new IndicatorManager(getWindow());
00044         }
00045
00046         final ViewPager2 viewPager = findViewById(R.id.introduction_viewPager);
00047         OnBackPressedCallback callback = new OnBackPressedCallback(false) {
00052             @Override
00053             public void handleOnBackPressed() {
00054                 viewPager.setCurrentItem(viewPager.getCurrentItem() - 1);
00055             }
00056         };
00057         getOnBackPressedDispatcher().addCallback(callback);
00058
00059         View.OnClickListener onFinish = view -> {
00060             PreferenceManager.getDefaultSharedPreferences(getApplicationContext())
00061                 .edit()
00062                 .putBoolean(getString(R.string.preference_first_use_key), false)
00063                 .apply();
00064             callback.setEnabled(false);
00065             getOnBackPressedDispatcher().onBackPressed();
00066         };
00067
00068         final Button skipButton = findViewById(R.id.introduction_button_skip);
00069         final ImageButton nextButton = findViewById(R.id.introduction_button_next);
00070         final Button finishButton = findViewById(R.id.introduction_button_finish);
00071
00072         skipButton.setOnClickListener(onFinish);
00073         nextButton.setOnClickListener(v -> viewPager.setCurrentItem(viewPager.getCurrentItem() + 1));
00074         finishButton.setOnClickListener(onFinish);
00075
00076         viewPager.setAdapter(new SectionsPagerAdapter(this));
00077         viewPager.registerOnPageChangeCallback(new ViewPager2.OnPageChangeCallback() {
00084             @Override
00085             public void onPageSelected(int position) {
00086                 indicatorManager.updateIndicators(position);
00087                 skipButton.setVisibility(position == IntroductionFragment.PAGES_COUNT - 1 ? View.GONE
: View.VISIBLE);
00088                 nextButton.setVisibility(position == IntroductionFragment.PAGES_COUNT - 1 ? View.GONE
: View.VISIBLE);
00089                 finishButton.setVisibility(position == IntroductionFragment.PAGES_COUNT - 1 ?
View.VISIBLE : View.GONE);
00090                 callback.setEnabled(position != 0);
00091             }
00092         });
00093     }
00094 }

```

6.42 IntroductionFragment.java

```

00001 package com.paternaggschga.gwent.ui.introduction;
00002
00003 import android.os.Bundle;
00004 import android.text.Html;
00005 import android.text.method.LinkMovementMethod;
00006 import android.view.LayoutInflater;
00007 import android.view.View;
00008 import android.view.ViewGroup;
00009 import android.widget.ImageView;
00010 import android.widget.TextView;
00011
00012 import androidx.annotation.IntRange;
00013 import androidx.annotation.NonNull;
00014 import androidx.fragment.app.Fragment;
00015
00016 import com.paternaggschga.gwent.R;
00017
00021 public class IntroductionFragment extends Fragment {
00026     public static final int PAGES_COUNT = 5;
00030     private static final String ARG_SECTION_NUMBER = "section_number";
00034     private static final int[] imageIds = {R.drawable.introduction_1, R.drawable.introduction_2,
R.drawable.introduction_3, R.drawable.introduction_4, R.drawable.introduction_5};
00035
00039     private static final int[] stringIds = {R.string.introduction_page1, R.string.introduction_page2,
R.string.introduction_page3, R.string.introduction_page4, R.string.introduction_page5};
00040
00041
00049     @NonNull
00050     public static IntroductionFragment newInstance(@IntRange(from = 0, to = PAGES_COUNT - 1) int
index) {
00051         if (index < 0 || index >= PAGES_COUNT) {
00052             throw new IllegalArgumentException("Index must be in [0, PAGES_COUNT - 1] but is " + index
+ ".");
00053         }

```



```

00054         Bundle arguments = new Bundle();
00055         arguments.putInt(ARG_SECTION_NUMBER, index);
00056         IntroductionFragment fragment = new IntroductionFragment();
00057         fragment.setArguments(arguments);
00058         return fragment;
00059     }
00060
00074     @Override
00075     public View onCreateView(@NonNull LayoutInflater inflater, ViewGroup container, Bundle
savedInstanceState) {
00076         int page = getArguments() != null ? getArguments().getInt(ARG_SECTION_NUMBER) : 0;
00077
00078         View root = inflater.inflate(R.layout.fragment_introduction, container, false);
00079
00080         TextView textView = root.findViewById(R.id.introduction_textView);
00081         textView.setText(Html.fromHtml(getString(stringIds[page]), Html.FROM_HTML_MODE_LEGACY));
00082         textView.setMovementMethod(LinkMovementMethod.getInstance());
00083
00084         ((ImageView) root.findViewById(R.id.introduction_imageView)).setImageResource(imageIds[page]);
00085         return root;
00086     }
00087 }

```

6.43 SectionsPagerAdapter.java

```

00001 package com.paternaggschga.gwent.ui.introduction;
00002
00003 import androidx.annotation.NonNull;
00004 import androidx.fragment.app.Fragment;
00005 import androidx.fragment.app.FragmentActivity;
00006 import androidx.viewpager2.adapter.FragmentStateAdapter;
00007
00012 public class SectionsPagerAdapter extends FragmentStateAdapter {
00018     public SectionsPagerAdapter(@NonNull FragmentActivity fragmentActivity) {
00019         super(fragmentActivity);
00020     }
00021
00027     @Override
00028     public int getItemCount() {
00029         return IntroductionFragment.PAGES_COUNT;
00030     }
00031
00037     @NonNull
00038     @Override
00039     public Fragment createFragment(int position) {
00040         return IntroductionFragment.newInstance(position);
00041     }
00042 }

```

6.44 FactionSwitchListener.java

```

00001 package com.paternaggschga.gwent.ui.main;
00002
00003 import android.content.Context;
00004 import android.content.SharedPreferences;
00005 import android.content.res.TypedArray;
00006 import android.os.Build;
00007 import android.view.Window;
00008 import android.widget.ImageButton;
00009 import android.widget.ImageView;
00010 import android.widget.TextView;
00011
00012 import androidx.annotation.NonNull;
00013 import androidx.annotation.Nullable;
00014 import androidx.constraintlayout.widget.ConstraintLayout;
00015 import androidx.preference.PreferenceManager;
00016
00017 import com.paternaggschga.gwent.R;
00018
00019 import org.jetbrains.annotations.Contract;
00020
00021 import java.util.ArrayList;
00022 import java.util.List;
00023
00031 public class FactionSwitchListener implements SharedPreferences.OnSharedPreferenceChangeListener {
00035     @NonNull
00036     public static final String THEME_PREFERENCE_KEY = "theme";
00037
00041     public static final int THEME_MONSTER = 0;
00042
00046     public static final int THEME_NILFGAARD = 1;
00047 }

```

```

00051     public static final int THEME_NORTHERN_KINGDOMS = 2;
00052
00056     public static final int THEME_SCOIATAEL = 3;
00057
00065     @NonNull
00066     private final List<ImageView> ballViews;
00067
00075     @NonNull
00076     private final List<ImageView> cardViews;
00077
00085     @NonNull
00086     private final List<TextView> unitViews;
00087
00095     @NonNull
00096     private final ImageButton factionButton;
00097
00108     private FactionSwitchListener(@NonNull List<ImageView> ballViews, @NonNull List<ImageView>
cardViews,
00109                                     @NonNull List<TextView> unitViews, @NonNull ImageButton
factionButton) {
00110         this.ballViews = ballViews;
00111         this.cardViews = cardViews;
00112         this.unitViews = unitViews;
00113         this.factionButton = factionButton;
00114     }
00115
00122     @NonNull
00123     @Contract ("_" -> new")
00124     public static FactionSwitchListener getListener(@NonNull Window mainWindow) {
00125         List<ImageView> ballViews = new ArrayList<> (4);
00126         List<ImageView> cardViews = new ArrayList<> (3);
00127         List<TextView> unitViews = new ArrayList<> (3);
00128
00129         for (int id : new int[] {R.id.firstRow, R.id.secondRow, R.id.thirdRow}) {
00130             ConstraintLayout rowLayout = mainWindow.findViewById(id);
00131
00132             ballViews.add(rowLayout.findViewById(R.id.pointBall));
00133             cardViews.add(rowLayout.findViewById(R.id.cardsImage));
00134             unitViews.add(rowLayout.findViewById(R.id.cardCountView));
00135         }
00136
00137         ballViews.add(mainWindow.findViewById(R.id.overallPointBall));
00138
00139         return new FactionSwitchListener(ballViews, cardViews, unitViews,
mainWindow.findViewById(R.id.factionButton));
00140     }
00141
00148     public static void setTheme(@NonNull Context context) {
00149         switch (PreferenceManager.getDefaultSharedPreferences(context)
00150             .getInt (THEME_PREFERENCE_KEY, THEME_SCOIATAEL)) {
00151             case THEME_MONSTER:
00152                 context.setTheme(R.style.MonsterTheme);
00153                 break;
00154             case THEME_NILFGAARD:
00155                 context.setTheme(R.style.NilfgaardTheme);
00156                 break;
00157             case THEME_NORTHERN_KINGDOMS:
00158                 context.setTheme(R.style.NorthernKingdomsTheme);
00159                 break;
00160             case THEME_SCOIATAEL:
00161             default:
00162                 context.setTheme(R.style.ScoiataelTheme);
00163         }
00164     }
00165
00175     @Override
00176     public void onSharedPreferenceChanged(SharedPreferences sharedPreferences, @Nullable String key) {
00177         if (!THEME_PREFERENCE_KEY.equals(key)) {
00178             return;
00179         }
00180
00181         Context context = factionButton.getContext();
00182
00183         switch (sharedPreferences.getInt(key, THEME_SCOIATAEL)) {
00184             case THEME_MONSTER:
00185                 context.setTheme(R.style.MonsterTheme);
00186                 break;
00187             case THEME_NILFGAARD:
00188                 context.setTheme(R.style.NilfgaardTheme);
00189                 break;
00190             case THEME_NORTHERN_KINGDOMS:
00191                 context.setTheme(R.style.NorthernKingdomsTheme);
00192                 break;
00193             case THEME_SCOIATAEL:
00194             default:
00195                 context.setTheme(R.style.ScoiataelTheme);
00196         }

```

```

00197
00198     int ballImageRes;
00199     int cardImageRes;
00200     int unitNumberTextColor;
00201     int factionButtonImageRes;
00202     if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
00203         try (TypedArray attributeValues =
00204             context.getTheme().obtainStyledAttributes(R.styleable.theme)) {
00205             ballImageRes = attributeValues.getResourceId(R.styleable.theme_point_ball_mipmap,
00206                 R.drawable.ball_green);
00207             cardImageRes = attributeValues.getResourceId(R.styleable.theme_card_view_mipmap,
00208                 R.drawable.card_scoiatael_landscape_free);
00209             unitNumberTextColor = attributeValues.getColor(R.styleable.theme_colorPrimary,
00210                 context.getColor(R.color.color_text_scoiatael));
00211             factionButtonImageRes =
00212                 attributeValues.getResourceId(R.styleable.theme_android_alertDialogIcon,
00213                     R.drawable.icon_round_scoiatael);
00214         } else {
00215             //noinspection resource
00216             TypedArray attributeValues = context.getTheme().obtainStyledAttributes(R.styleable.theme);
00217             ballImageRes = attributeValues.getResourceId(R.styleable.theme_point_ball_mipmap,
00218                 R.drawable.ball_green);
00219             cardImageRes = attributeValues.getResourceId(R.styleable.theme_card_view_mipmap,
00220                 R.drawable.card_scoiatael_landscape_free);
00221             unitNumberTextColor = attributeValues.getColor(R.styleable.theme_colorPrimary,
00222                 context.getColor(R.color.color_text_scoiatael));
00223             factionButtonImageRes =
00224                 attributeValues.getResourceId(R.styleable.theme_android_alertDialogIcon,
00225                     R.drawable.icon_round_scoiatael);
00226             attributeValues.recycle();
00227         }
00228     }
00229     ballViews.forEach(view -> ImageViewSwitchAnimator.animatedSwitch(view,
00230         ballImageRes).subscribe());
00231     cardViews.forEach(view -> ImageViewSwitchAnimator.animatedSwitch(view,
00232         cardImageRes).subscribe());
00233     unitViews.forEach(textView -> textView.setTextColor(unitNumberTextColor));
00234     ImageViewSwitchAnimator.animatedSwitch(factionButton, factionButtonImageRes).subscribe();
00235 }
00236 }

```

6.45 GameBoardViewModel.java

```

00001 package com.paternaggschga.gwent.ui.main;
00002
00003 import static androidx.lifecycle.ViewModelProvider.AndroidViewModelFactory.APPLICATION_KEY;
00004
00005 import android.content.Context;
00006
00007 import androidx.annotation.NonNull;
00008 import androidx.lifecycle.AndroidViewModel;
00009 import androidx.lifecycle.ViewModelProvider;
00010 import androidx.lifecycle.ViewModelStoreOwner;
00011 import androidx.lifecycle.viewmodel.ViewModelInitializer;
00012
00013 import com.paternaggschga.gwent.GwentApplication;
00014 import com.paternaggschga.gwent.data.RowType;
00015 import com.paternaggschga.gwent.data.UnitRepository;
00016 import com.paternaggschga.gwent.domain.cases.BurnDialogUseCase;
00017 import com.paternaggschga.gwent.domain.cases.DamageCalculatorUseCase;
00018 import com.paternaggschga.gwent.domain.cases.ResetDialogUseCase;
00019 import com.paternaggschga.gwent.domain.damage.DamageCalculator;
00020 import com.paternaggschga.gwent.ui.sounds.SoundManager;
00021
00022 import java.util.HashMap;
00023 import java.util.Map;
00024 import java.util.Objects;
00025 import java.util.concurrent.TimeUnit;
00026
00027 import io.reactivex.rxjava3.android.schedulers.AndroidSchedulers;
00028 import io.reactivex.rxjava3.core.Completable;
00029 import io.reactivex.rxjava3.core.Flowable;
00030 import io.reactivex.rxjava3.core.Single;
00031 import io.reactivex.rxjava3.schedulers.Schedulers;
00032
00033 public class GameBoardViewModel extends AndroidViewModel {
00034     @NonNull
00035     private static final ViewModelInitializer<GameBoardViewModel> INITIALIZER = new
00036         ViewModelInitializer<>() {
00037         GameBoardViewModel class,
00038         creationExtras -> {
00039             GwentApplication app = (GwentApplication) creationExtras.get(APPLICATION_KEY);
00040             assert app != null;
00041             return new GameBoardViewModel(app);
00042         }
00043     };

```

```

00051         });
00052
00060     private GameBoardViewModel(@NonNull GwentApplication application) {
00061         super(application);
00062     }
00063
00070     @NonNull
00071     private final Map<RowType, Flowable<RowUiState>> rowUiStates = new
HashMap<>(RowType.values().length);
00072
00079     @NonNull
00080     private Flowable<MenuUiState> menuUiState = Flowable.empty();
00081
00085     private SoundManager soundManager;
00086
00097     @NonNull
00098     public static GameBoardViewModel getModel(@NonNull ViewModelStoreOwner owner,
00099                                             @NonNull UnitRepository repository,
00100                                             @NonNull SoundManager soundManager) {
00101         GameBoardViewModel result = new ViewModelProvider(owner,
ViewModelProvider.Factory.from(INIALIZER))
00102             .get(GameBoardViewModel.class);
00103
00104         for (RowType row : RowType.values()) {
00105             result.rowUiStates.put(row,
00106                 Flowable.combineLatest(repository.isWeatherFlowable(row),
00107                     repository.isHornFlowable(row),
00108                     repository.getUnitsFlowable(row),
00109                     (weather, horn, units) -> {
00110                         DamageCalculator calculator =
DamageCalculatorUseCase.getDamageCalculator(weather, horn, units);
00111                         int damage = units.stream()
00112                             .map(unitEntity -> unitEntity.calculateDamage(calculator))
00113                             .reduce(0, Integer::sum);
00114                         return new RowUiState(damage, weather, horn, units.size());
00115                     })
00116                 .distinctUntilChanged()
00117                 .onBackpressureLatest()
00118                 .debounce(10, TimeUnit.MILLISECONDS)
00119                 .subscribeOn(Schedulers.io())
00120                 .observeOn(AndroidSchedulers.mainThread())
00121             );
00122         }
00123
00124         Flowable<MenuUiState> combinedRowStates = Flowable.combineLatest(result.rowUiStates.values(),
(Object[] rowUiStates) -> {
00125             int damage = 0;
00126             boolean reset = false;
00127             boolean weather = false;
00128             for (Object state : rowUiStates) {
00129                 RowUiState rowUiState = (RowUiState) state;
00130                 damage += rowUiState.getDamage();
00131                 reset |= rowUiState.isHorn() || rowUiState.getUnits() != 0;
00132                 weather |= rowUiState.isWeather();
00133             }
00134             reset |= weather;
00135             return new MenuUiState(damage, reset, weather, false);
00136         }).distinctUntilChanged().onBackpressureLatest();
00137
00138         result.menuUiState = Flowable.combineLatest(combinedRowStates,
00139             repository.hasNonEpicUnitsFlowable(),
00140             (menuUiState, hasNonEpicUnits) -> new MenuUiState(menuUiState.getDamage(),
00141                 menuUiState.isReset(),
00142                 menuUiState.isWeather(),
00143                 hasNonEpicUnits)
00144             .distinctUntilChanged()
00145             .onBackpressureLatest()
00146             .debounce(10, TimeUnit.MILLISECONDS)
00147             .subscribeOn(Schedulers.io())
00148             .observeOn(AndroidSchedulers.mainThread());
00149
00150         result.soundManager = soundManager;
00151         return result;
00152     }
00153
00161     private Single<UnitRepository> getRepository() {
00162         return GwentApplication.getRepository(getApplication());
00163     }
00164
00172     @NonNull
00173     public Flowable<RowUiState> getRowUiState(@NonNull RowType row) {
00174         return Objects.requireNonNull(rowUiStates.get(row));
00175     }
00176
00184     @NonNull
00185     public Flowable<MenuUiState> getMenuUiState() {
00186         return menuUiState;

```

```

00187     }
00188
00194     @NonNull
00195     public SoundManager getSoundManager() {
00196         return soundManager;
00197     }
00198
00206     @NonNull
00207     public Completable onWeatherViewPressed(@NonNull RowType row) {
00208         return getRepository()
00209             .flatMap(repository -> repository
00210                 .switchWeather(row)
00211                 .andThen(repository.isWeather(row)))
00212             .doOnSuccess(weather -> {
00213                 if (weather) {
00214                     soundManager.playWeatherSound(row);
00215                 }
00216             })
00217             .ignoreElement();
00218     }
00219
00227     public Completable onHornViewPressed(@NonNull RowType row) {
00228         return getRepository()
00229             .flatMap(repository -> repository
00230                 .switchHorn(row)
00231                 .andThen(repository.isHorn(row)))
00232             .doOnSuccess(horn -> {
00233                 if (horn) {
00234                     soundManager.playHornSound();
00235                 }
00236             })
00237             .ignoreElement();
00238     }
00239
00249     @NonNull
00250     public Completable onResetButtonPressed(@NonNull Context context) {
00251         return reset(context, ResetDialogUseCase.Trigger.BUTTON_CLICK);
00252     }
00253
00262     @NonNull
00263     public Completable onFactionSwitchReset(@NonNull Context context) {
00264         return reset(context, ResetDialogUseCase.Trigger.FACTION_SWITCH);
00265     }
00266
00275     @NonNull
00276     private Completable reset(@NonNull Context context, @NonNull ResetDialogUseCase.Trigger trigger) {
00277         return ResetDialogUseCase
00278             .reset(context, trigger, soundManager)
00279             .doOnSuccess(playSound -> {
00280                 if (playSound) {
00281                     soundManager.playResetSound();
00282                 }
00283             })
00284             .ignoreElement();
00285     }
00286
00294     @NonNull
00295     public Completable onWeatherButtonPressed() {
00296         return
00297         getRepository().flatMapCompletable(UnitRepository::clearWeather).doOnComplete(soundManager::playClearWeatherSound);
00298     }
00299
00309     @NonNull
00310     public Completable onBurnButtonPressed(@NonNull Context context) {
00311         return BurnDialogUseCase
00312             .burn(context, soundManager)
00313             .doOnSuccess(playSound -> {
00314                 if (playSound) {
00315                     soundManager.playBurnSound();
00316                 }
00317             })
00318             .ignoreElement();
00319     }
00320 }

```

6.46 ImageViewSwitchAnimator.java

```

00001 package com.peternaggschga.gwent.ui.main;
00002
00003 import android.view.ViewGroup;
00004 import android.view.animation.Animation;
00005 import android.view.animation.AnimationUtils;
00006 import android.widget.ImageView;
00007
00008 import androidx.annotation.DrawableRes;

```

```

00009 import androidx.annotation.NonNull;
00010 import androidx.constraintlayout.widget.ConstraintLayout;
00011
00012 import io.reactivex.rxjava3.core.Completable;
00013
00020 public class ImageViewSwitchAnimator {
00026     @NonNull
00027     private static ImageView getOverlayView(@NonNull ImageView template) {
00028         ImageView result = new ImageView(template.getContext());
00029         ViewGroup.LayoutParams oldParams = template.getLayoutParams();
00030         ConstraintLayout.LayoutParams newParams = new ConstraintLayout.LayoutParams(oldParams.width,
00031             oldParams.height);
00032         newParams.topToTop = template.getId();
00033         newParams.endToEnd = template.getId();
00034         newParams.startToStart = template.getId();
00035         newParams.bottomToBottom = template.getId();
00036         result.setLayoutParams(newParams);
00037         result.setImageDrawable(template.getDrawable());
00038         result.setScaleType(template.getScaleType());
00039         result.setTranslationZ(template.getTranslationZ() + 1);
00040         return result;
00041     }
00042
00052     @NonNull
00053     public static Completable animatedSwitch(@NonNull ImageView view, @DrawableRes int resId) {
00054         return Completable.create(emitter -> {
00055             Animation fadeAnimation = AnimationUtils.loadAnimation(view.getContext(),
00056                 android.R.anim.fade_out);
00057             ImageView foregroundView = getOverlayView(view);
00058             ViewGroup parent = (ViewGroup) view.getParent();
00059             fadeAnimation.setAnimationListener(new Animation.AnimationListener() {
00060                 @Override
00061                 public void onAnimationStart(Animation animation) {
00062                     view.setImageResource(resId);
00063                 }
00064
00065                 @Override
00066                 public void onAnimationEnd(Animation animation) {
00067                     parent.removeView(foregroundView);
00068                     emitter.onComplete();
00069                 }
00070
00071                 @Override
00072                 public void onAnimationRepeat(Animation animation) {
00073                 }
00074             });
00075             parent.addView(foregroundView);
00076             foregroundView.startAnimation(fadeAnimation);
00077         });
00078     }
00079 }
00080 }

```

6.47 MainActivity.java

```

00001 package com.peternaggschga.gwent.ui.main;
00002
00003 import static com.peternaggschga.gwent.ui.main.FactionSwitchListener.THEME_PREFERENCE_KEY;
00004 import static com.peternaggschga.gwent.ui.main.FactionSwitchListener.THEME_SCOIATAEL;
00005
00006 import android.app.Dialog;
00007 import android.content.Context;
00008 import android.content.Intent;
00009 import android.content.SharedPreferences;
00010 import android.os.Bundle;
00011 import android.os.Handler;
00012 import android.os.Looper;
00013 import android.view.View;
00014 import android.view.WindowManager;
00015 import android.widget.ImageButton;
00016 import android.widget.ImageView;
00017
00018 import androidx.annotation.NonNull;
00019 import androidx.annotation.Nullable;
00020 import androidx.appcompat.app.AppCompatActivity;
00021 import androidx.constraintlayout.widget.ConstraintLayout;
00022 import androidx.core.view.WindowCompat;
00023 import androidx.core.view.WindowInsetsCompat;
00024 import androidx.core.view.WindowInsetsControllerCompat;
00025 import androidx.preference.PreferenceManager;
00026
00027 import com.peternaggschga.gwent.GwentApplication;
00028 import com.peternaggschga.gwent.R;
00029 import com.peternaggschga.gwent.data.RowType;

```

```

00030 import com.paternaggschga.gwent.ui.dialogs.ChangeFactionDialog;
00031 import com.paternaggschga.gwent.ui.dialogs.CoinFlipDialog;
00032 import com.paternaggschga.gwent.ui.dialogs.OverlayDialog;
00033 import com.paternaggschga.gwent.ui.dialogs.addcard.AddCardDialog;
00034 import com.paternaggschga.gwent.ui.dialogs.cards.ShowUnitsDialog;
00035 import com.paternaggschga.gwent.ui.introduction.IntroductionActivity;
00036 import com.paternaggschga.gwent.ui.settings.SettingsActivity;
00037 import com.paternaggschga.gwent.ui.sounds.SoundManager;
00038
00039 import io.reactivex.rxjava3.core.Single;
00040 import io.reactivex.rxjava3.disposables.CompositeDisposable;
00041 import io.reactivex.rxjava3.functions.Function;
00042
00046 public class MainActivity extends AppCompatActivity {
00051     @NonNull
00052     private final CompositeDisposable disposables = new CompositeDisposable();
00053
00061     private SharedPreferences.OnSharedPreferenceChangeListener factionSwitchListener;
00062
00066     private GameBoardViewModel gameBoardViewModel;
00067
00082     @Override
00083     protected void onCreate(@Nullable Bundle savedInstanceState) {
00084         super.onCreate(savedInstanceState);
00085
00086         SharedPreferences preferences = PreferenceManager.getDefaultSharedPreferences(this);
00087
00088         if (preferences.getBoolean(getString(R.string.preference_first_use_key), true)) {
00089             startActivity(new Intent(this, IntroductionActivity.class));
00090         }
00091
00092         FactionSwitchListener.setTheme(this);
00093         setContentView(R.layout.activity_main);
00094
00095         if (gameBoardViewModel == null) {
00096             disposables.add(
00097                 GwentApplication.getRepository(this)
00098                     .map(repository ->
00099                         GameBoardViewModel.getModel(MainActivity.this,
00100                             repository,
00101                             new SoundManager(this)))
00102                     .subscribe(gameBoardViewModel -> {
00103                         this.gameBoardViewModel = gameBoardViewModel;
00104                         initializeViewModel();
00105                     })
00106             );
00107         }
00108
00109         if (factionSwitchListener == null) {
00110             factionSwitchListener = FactionSwitchListener.getListener(getWindow());
00111             preferences.registerOnSharedPreferenceChangeListener(factionSwitchListener);
00112         }
00113
00114         findViewById(R.id.factionButton).setOnClickListener(v -> inflateFactionPopup());
00115         findViewById(R.id.coinButton).setOnClickListener(v -> inflateCoinFlipPopup());
00116         findViewById(R.id.settingsButton).setOnClickListener(v -> startActivity(new
Intent(MainActivity.this, SettingsActivity.class)));
00117     }
00118
00123     @Override
00124     protected void onResume() {
00125         super.onResume();
00126         // set background image according to preferences
00127         int backgroundImageKey = Integer.parseInt(
00128             PreferenceManager.getDefaultSharedPreferences(this)
00129                 .getString(getString(R.string.preference_key_design),
00130                     getString(R.string.design_preference_default))
00131         );
00132         ImageView backgroundImage = findViewById(R.id.backgroundImageView);
00133         if (backgroundImageKey != 0) {
00134             backgroundImage.setVisibility(View.VISIBLE);
00135             backgroundImage.setImageResource(new int[]{
00136                 R.drawable.background_geralt,
00137                 R.drawable.background_ciri,
00138                 R.drawable.background_jaskier,
00139                 R.drawable.background_yennefer,
00140                 R.drawable.background_eredin
00141             }[backgroundImageKey - 1]);
00142         } else {
00143             backgroundImage.setVisibility(View.INVISIBLE);
00144         }
00145     }
00146
00153     @Override
00154     public void onWindowFocusChanged(boolean hasFocus) {
00155         super.onWindowFocusChanged(hasFocus);
00156

```

```

00157         if (hasFocus) {
00158             new Handler(Looper.getMainLooper()).postDelayed(() -> {
00159                 if (!hasWindowFocus()) {
00160                     return;
00161                 }
00162             }, 250);
00163             // hide system UI
00164             WindowInsetsControllerCompat windowController =
00165                 WindowCompat.getInsetsController(getWindow(), getWindow().getDecorView());
00166             windowController.hide(WindowInsetsCompat.Type.systemBars());
00167             windowController.hide(WindowInsetsCompat.Type.tappableElement());
00168             windowController.setSystemBarsBehavior(WindowInsetsControllerCompat.BEHAVIOR_SHOW_TRANSIENT_BARS_BY_SWIPE);
00169             // keep screen on
00170             getWindow().addFlags(WindowManager.LayoutParams.FLAG_KEEP_SCREEN_ON);
00171         }, 250);
00172     }
00173 }
00174
00175 @Override
00176 protected void onDestroy() {
00177     super.onDestroy();
00178     disposables.dispose();
00179     disposables.clear();
00180 }
00181
00182 private void initializeViewModel() {
00183     int[] rowIds = {R.id.firstRow, R.id.secondRow, R.id.thirdRow};
00184     for (int rowId = 0; rowId < rowIds.length; rowId++) {
00185         RowType row = RowType.values()[rowId];
00186         ConstraintLayout rowLayout = findViewById(rowIds[rowId]);
00187
00188         ImageView weather = rowLayout.findViewById(R.id.weatherView);
00189         ImageView horn = rowLayout.findViewById(R.id.hornView);
00190         ConstraintLayout cards = rowLayout.findViewById(R.id.cardView);
00191
00192         weather.setOnClickListener(v ->
00193             disposables.add(gameBoardViewModel.onWeatherViewPressed(row).subscribe()));
00194         horn.setOnClickListener(v ->
00195             disposables.add(gameBoardViewModel.onHornViewPressed(row).subscribe()));
00196
00197         cards.setOnClickListener(v -> disposables.add(
00198             GwentApplication.getRepository(this)
00199                 .flatMap(repository -> repository.countUnits(row))
00200                 .map(count -> count == 0)
00201                 .flatMap((Function<Boolean, Single<? extends OverlayDialog>) rowEmpty ->
00202                     rowEmpty
00203                         ? Single.just(new AddCardDialog(MainActivity.this, row,
00204                             gameBoardViewModel.getSoundManager()))
00205                         : ShowUnitsDialog.getDialog(MainActivity.this, row,
00206                             gameBoardViewModel.getSoundManager()))
00207                 .subscribe(Dialog::show)
00208         ));
00209         final RowUiStateObserver observer = RowUiStateObserver.getObserver(row,
00210             rowLayout.findViewById(R.id.pointView),
00211             weather,
00212             horn,
00213             rowLayout.findViewById(R.id.cardCountView));
00214         disposables.add(gameBoardViewModel.getRowUiState(row).subscribe(observer));
00215     }
00216
00217     ImageButton reset = findViewById(R.id.resetButton);
00218     ImageButton weather = findViewById(R.id.weatherButton);
00219     ImageButton burn = findViewById(R.id.burnButton);
00220
00221     final MenuUiStateObserver observer = new
00222     MenuUiStateObserver(findViewById(R.id.overallPointView),
00223         reset,
00224         weather,
00225         burn);
00226     disposables.add(gameBoardViewModel.getMenuUiState().subscribe(observer));
00227
00228     reset.setOnClickListener(v ->
00229         disposables.add(gameBoardViewModel.onResetButtonPressed(this).subscribe()));
00230     weather.setOnClickListener(v ->
00231         disposables.add(gameBoardViewModel.onWeatherButtonPressed().subscribe()));
00232     burn.setOnClickListener(v ->
00233         disposables.add(gameBoardViewModel.onBurnButtonPressed(MainActivity.this).subscribe()));
00234 }
00235
00236 private void inflateFactionPopup() {
00237     new ChangeFactionDialog(this, theme -> {
00238         SharedPreferences preferences = PreferenceManager.getDefaultSharedPreferences(this);
00239     });
00240 }

```



```

00254         if (preferences.getInt(THEME_PREFERENCE_KEY, THEME_SCOIATAEL) == theme) {
00255             return;
00256         }
00257
00258         boolean resetOnFactionSwitch = preferences.getBoolean(
00259             getString(R.string.preference_key_faction_reset),
00260             getResources().getBoolean(R.bool.faction_reset_preference_default)
00261         );
00262         if (resetOnFactionSwitch) {
00263             disposables.add(gameBoardViewModel.onFactionSwitchReset(this).subscribe());
00264         }
00265         preferences.edit()
00266             .putInt(THEME_PREFERENCE_KEY, theme)
00267             .apply();
00268     }).show();
00269 }
00270
00274 private void inflateCoinFlipPopup() {
00275     new CoinFlipDialog(this).show();
00276     gameBoardViewModel.getSoundManager().playCoinSound();
00277 }
00278 }

```

6.48 MenuUiState.java

```

00001 package com.paternaggschga.gwent.ui.main;
00002
00003
00004
00005 import androidx.annotation.IntRange;
00006
00010 public class MenuUiState {
00014     @IntRange(from = 0)
00015     private final int damage;
00016
00020     private final boolean reset;
00021
00025     private final boolean weather;
00026
00030     private final boolean burn;
00031
00041     public MenuUiState(@IntRange(from = 0) int damage, boolean reset, boolean weather, boolean burn) {
00042         if (damage < 0) {
00043             throw new IllegalArgumentException("Damage must be greater or equal to 0 but is " + damage
00044                 + ".");
00045         }
00046         this.damage = damage;
00047         this.reset = reset;
00048         this.weather = weather;
00049         this.burn = burn;
00050
00056     @IntRange(from = 0)
00057     public int getDamage() {
00058         return damage;
00059     }
00060
00066     public boolean isReset() {
00067         return reset;
00068     }
00069
00075     public boolean isWeather() {
00076         return weather;
00077     }
00078
00084     public boolean isBurn() {
00085         return burn;
00086     }
00087
00088     @Override
00089     public boolean equals(Object o) {
00090         if (this == o) return true;
00091         if (!(o instanceof MenuUiState)) return false;
00092         MenuUiState that = (MenuUiState) o;
00093         return damage == that.damage && reset == that.reset && weather == that.weather && burn ==
00094             that.burn;
00095     }
00096 }

```

6.49 MenuUiStateObserver.java

```

00001 package com.paternaggschga.gwent.ui.main;

```

```

00002
00003 import android.widget.ImageButton;
00004 import android.widget.TextView;
00005
00006 import androidx.annotation.NonNull;
00007
00008 import com.paternaggschga.gwent.R;
00009
00010 import io.reactivex.rxjava3.functions.Consumer;
00011
00018 public class MenuUiStateObserver implements Consumer<MenuUiState> {
00022     @NonNull
00023     private final TextView damageView;
00024
00030     @NonNull
00031     private final ImageButton resetButton;
00032
00037     @NonNull
00038     private final ImageButton weatherButton;
00039
00044     @NonNull
00045     private final ImageButton burnButton;
00046
00055     public MenuUiStateObserver(@NonNull TextView damageView, @NonNull ImageButton resetButton,
00056                               @NonNull ImageButton weatherButton, @NonNull ImageButton burnButton) {
00057         this.damageView = damageView;
00058         this.resetButton = resetButton;
00059         this.weatherButton = weatherButton;
00060         this.burnButton = burnButton;
00061     }
00062
00071     @Override
00072     public void accept(@NonNull MenuUiState menuUiState) {
00073         damageView.setText(String.valueOf(menuUiState.getDamage()));
00074         resetButton.setClickable(menuUiState.isReset());
00075         ImageViewSwitchAnimator.animatedSwitch(resetButton, menuUiState.isReset() ?
R.drawable.icon_reset : R.drawable.icon_reset_grey)
00076             .subscribe();
00077         weatherButton.setClickable(menuUiState.isWeather());
00078         ImageViewSwitchAnimator.animatedSwitch(weatherButton, menuUiState.isWeather() ?
R.drawable.icon_weather : R.drawable.icon_weather_grey)
00079             .subscribe();
00080         burnButton.setClickable(menuUiState.isBurn());
00081         ImageViewSwitchAnimator.animatedSwitch(burnButton, menuUiState.isBurn() ? R.drawable.icon_burn
: R.drawable.icon_burn_grey)
00082             .subscribe();
00083     }
00084 }

```

6.50 RowUiState.java

```

00001 package com.paternaggschga.gwent.ui.main;
00002
00003
00004
00005 import androidx.annotation.IntRange;
00006
00010 public class RowUiState {
00014     @IntRange(from = 0)
00015     private final int damage;
00016
00020     private final boolean weather;
00021
00025     private final boolean horn;
00026
00030     @IntRange(from = 0)
00031     private final int units;
00032
00041     public RowUiState(@IntRange(from = 0) int damage, boolean weather, boolean horn, @IntRange(from =
0) int units) {
00042         if (damage < 0) {
00043             throw new IllegalArgumentException("Damage must be greater or equal to 0 but is " + damage
+ ".");
00044         }
00045         if (units < 0) {
00046             throw new IllegalArgumentException("Units must be greater or equal to 0 but is " + units +
".");
00047         }
00048         this.damage = damage;
00049         this.weather = weather;
00050         this.horn = horn;
00051         this.units = units;
00052     }
00053
00058     @IntRange(from = 0)

```

```

00059     public int getDamage() {
00060         return damage;
00061     }
00062
00067     public boolean isWeather() {
00068         return weather;
00069     }
00070
00075     public boolean isHorn() {
00076         return horn;
00077     }
00078
00083     @IntRange(from = 0)
00084     public int getUnits() {
00085         return units;
00086     }
00087
00088     @Override
00089     public boolean equals(Object o) {
00090         if (this == o) return true;
00091         if (!(o instanceof RowUiState)) return false;
00092         RowUiState state = (RowUiState) o;
00093         return damage == state.damage && weather == state.weather && horn == state.horn && units ==
state.units;
00094     }
00095 }

```

6.51 RowUiStateObserver.java

```

00001 package com.paternaggschga.gwent.ui.main;
00002
00003 import android.widget.ImageView;
00004 import android.widget.TextView;
00005
00006 import androidx.annotation.DrawableRes;
00007 import androidx.annotation.NonNull;
00008
00009 import com.paternaggschga.gwent.R;
00010 import com.paternaggschga.gwent.data.RowType;
00011
00012 import org.jetbrains.annotations.Contract;
00013
00014 import io.reactivex.rxjava3.functions.Consumer;
00015
00022 public class RowUiStateObserver implements Consumer<RowUiState> {
00027     @NonNull
00028     private final TextView damageView;
00029
00036     @NonNull
00037     private final ImageView weatherView;
00038
00045     @DrawableRes
00046     private final int weatherResource;
00047
00054     @NonNull
00055     private final ImageView hornView;
00056
00060     @NonNull
00061     private final TextView unitView;
00062
00074     private RowUiStateObserver(@NonNull TextView damageView, @NonNull ImageView weatherView,
00075                               @DrawableRes int weatherResource, @NonNull ImageView hornView,
00076                               @NonNull TextView unitView) {
00077         this.damageView = damageView;
00078         this.weatherView = weatherView;
00079         this.weatherResource = weatherResource;
00080         this.hornView = hornView;
00081         this.unitView = unitView;
00082     }
00083
00097     @NonNull
00098     @Contract("_, _, _, _ -> new")
00099     public static RowUiStateObserver getObserver(@NonNull RowType row, @NonNull TextView damageView,
00100                                                 @NonNull ImageView weatherView, @NonNull ImageView
00101                                                 hornView,
00102                                                 @NonNull TextView unitView) {
00103         int weatherResource;
00104         switch (row) {
00105             case MELEE:
00106                 weatherResource = R.drawable.frost_weather;
00107                 break;
00108             case RANGE:
00109                 weatherResource = R.drawable.fog_weather;
00110                 break;
00111             case SIEGE:

```

```

00111         weatherResource = R.drawable.rain_weather;
00112         break;
00113     default:
00114         weatherResource = R.drawable.good_weather;
00115     }
00116     return new RowUiStateObserver(damageView, weatherView, weatherResource, hornView, unitView);
00117 }
00118
00126 @Override
00127 public void accept(@NonNull RowUiState rowUiState) {
00128     damageView.setText(String.valueOf(rowUiState.getDamage()));
00129     ImageViewSwitchAnimator.animatedSwitch(weatherView, rowUiState.isWeather() ? weatherResource :
R.drawable.good_weather)
00130         .subscribe();
00131     ImageViewSwitchAnimator.animatedSwitch(hornView, rowUiState.isHorn() ? R.drawable.horn :
R.drawable.horn_grey)
00132         .subscribe();
00133     unitView.setText(String.valueOf(rowUiState.getUnits()));
00134 }
00135 }

```

6.52 RuleActivity.java

```

00001 package com.paternaggschga.gwent.ui.settings;
00002
00003 import android.os.Bundle;
00004 import android.text.Html;
00005 import android.view.MenuItem;
00006 import android.widget.TextView;
00007
00008 import androidx.annotation.NonNull;
00009 import androidx.appcompat.app.ActionBar;
00010 import androidx.appcompat.app.AppCompatActivity;
00011
00012 import com.paternaggschga.gwent.R;
00013
00014 import java.util.Objects;
00015
00022 public class RuleActivity extends AppCompatActivity {
00027     public static final String INTENT_EXTRA_TAG = "rule_section";
00028
00036     @Override
00037     public void onCreate(Bundle savedInstanceState) {
00038         super.onCreate(savedInstanceState);
00039
00040         setContentView(R.layout.activity_rules);
00041
00042         RuleSection section = Objects.requireNonNull((RuleSection)
getIntent().getSerializableExtra(INTENT_EXTRA_TAG));
00043
00044         setSupportActionBar(findViewById(R.id.rules_toolbar));
00045
00046         TextView textView = findViewById(R.id.rules_textview);
00047
00048         ActionBar actionBar = Objects.requireNonNull(getSupportActionBar());
00049         actionBar.setDisplayHomeAsUpEnabled(true);
00050
00051         switch (section) {
00052             case GENERAL:
00053                 textView.setText(Html.fromHtml(getString(R.string.rules_general_text),
Html.FROM_HTML_MODE_LEGACY));
00054                 actionBar.setTitle(R.string.preference_rules_general_title);
00055                 break;
00056             case COURSE:
00057                 textView.setText(Html.fromHtml(getString(R.string.rules_course_text),
Html.FROM_HTML_MODE_LEGACY));
00058                 actionBar.setTitle(R.string.preference_rules_course_title);
00059                 break;
00060             case FACTIONS:
00061                 textView.setText(Html.fromHtml(getString(R.string.rules_factions_text),
Html.FROM_HTML_MODE_LEGACY));
00062                 actionBar.setTitle(R.string.preference_rules_factions_title);
00063                 break;
00064             case COMMANDER:
00065                 textView.setText(Html.fromHtml(getString(R.string.rules_commander_text),
Html.FROM_HTML_MODE_LEGACY));
00066                 actionBar.setTitle(R.string.preference_rules_commander_title);
00067                 break;
00068             case CARDS:
00069                 textView.setText(Html.fromHtml(getString(R.string.rules_cards_text),
Html.FROM_HTML_MODE_LEGACY));
00070                 actionBar.setTitle(R.string.preference_rules_cards_title);
00071                 break;
00072             case CARD_ABILITIES:
00073                 textView.setText(Html.fromHtml(getString(R.string.rules_card_abilities_text),

```

```

00079             Html.FROM_HTML_MODE_LEGACY));
00080         actionBar.setTitle(R.string.preference_rules_card_abilities_title);
00081         break;
00082     case SPECIAL_CARDS:
00083         textView.setText(Html.fromHtml(getString(R.string.rules_special_cards_text),
00084             Html.FROM_HTML_MODE_LEGACY));
00085         actionBar.setTitle(R.string.preference_rules_special_cards_title);
00086     }
00087 }
00088
00096 @Override
00097 public boolean onOptionsItemSelected(@NonNull MenuItem item) {
00098     if (item.getItemId() == android.R.id.home) {
00099         getOnBackPressedDispatcher().onBackPressed();
00100         return true;
00101     }
00102     return super.onOptionsItemSelected(item);
00103 }
00104 }

```

6.53 RuleSection.java

```

00001 package com.paternaggschga.gwent.ui.settings;
00002
00006 enum RuleSection {
00010     GENERAL,
00011
00015     COURSE,
00016
00020     FACTIONS,
00021
00025     COMMANDER,
00026
00030     CARDS,
00031
00035     CARD_ABILITIES,
00036
00040     SPECIAL_CARDS
00041 }

```

6.54 SettingsActivity.java

```

00001 package com.paternaggschga.gwent.ui.settings;
00002
00003 import android.content.SharedPreferences;
00004 import android.os.Bundle;
00005 import android.view.MenuItem;
00006
00007 import androidx.annotation.NonNull;
00008 import androidx.annotation.Nullable;
00009 import androidx.appcompat.app.ActionBar;
00010 import androidx.appcompat.app.AppCompatActivity;
00011 import androidx.fragment.app.Fragment;
00012 import androidx.preference.Preference;
00013 import androidx.preference.PreferenceFragmentCompat;
00014
00015 import com.paternaggschga.gwent.R;
00016 import com.paternaggschga.gwent.ui.main.FactionSwitchListener;
00017
00018 import java.util.Objects;
00019
00024 public class SettingsActivity extends AppCompatActivity implements
    PreferenceFragmentCompat.OnPreferenceStartFragmentCallback {
00029     private static final String CURRENT_FRAGMENT_KEY = "currentFragment";
00030
00037     @Override
00038     protected void onCreate(@Nullable Bundle savedInstanceState) {
00039         super.onCreate(savedInstanceState);
00040         FactionSwitchListener.setTheme(this);
00041
00042         setContentView(R.layout.activity_settings);
00043
00044         setSupportActionBar(findViewById(R.id.settings_toolbar));
00045         ActionBar actionBar = getSupportActionBar();
00046         if (actionBar != null) {
00047             actionBar.setDisplayHomeAsUpEnabled(true);
00048         }
00049
00050         Fragment currentFragment = null;
00051         if (savedInstanceState != null) {
00052             currentFragment = getSupportFragmentManager().getFragment(savedInstanceState,
                CURRENT_FRAGMENT_KEY);

```

```

00053     }
00054     currentFragment = currentFragment == null ? new SettingsHeaderFragment() : currentFragment;
00055     getSupportFragmentManager().beginTransaction()
00056         .replace(R.id.settingsFrameLayout, currentFragment)
00057         .commit();
00058 }
00059
00065 @Override
00066 protected void onSaveInstanceState(@NonNull Bundle outState) {
00067     super.onSaveInstanceState(outState);
00068
00069     getSupportFragmentManager().getFragments()
00070         .stream()
00071         .filter(Fragment::isVisible)
00072         .findAny()
00073         .ifPresent(fragment ->
00074             getSupportFragmentManager().putFragment(outState, CURRENT_FRAGMENT_KEY,
fragment));
00075     }
00076
00084 @Override
00085 public boolean onOptionsItemSelected(@NonNull MenuItem item) {
00086     if (item.getItemId() == android.R.id.home) {
00087         getOnBackPressedDispatcher().onBackPressed();
00088         return true;
00089     }
00090     return super.onOptionsItemSelected(item);
00091 }
00092
00101 @Override
00102 public boolean onPreferenceStartFragment(@NonNull PreferenceFragmentCompat caller, @NonNull
Preference pref) {
00103     // Instantiate the new Fragment
00104     final Bundle args = pref.getExtras();
00105
00106     Fragment fragment = getSupportFragmentManager()
00107         .getFragmentManager()
00108         .instantiate(getClassLoader(), Objects.requireNonNull(pref.getFragment()));
00109     fragment.setArguments(args);
00110
00111     // Replace the existing Fragment with the new Fragment
00112     getSupportFragmentManager().beginTransaction()
00113         .replace(R.id.settingsFrameLayout, fragment)
00114         .addToBackStack(null)
00115         .commit();
00116     setTitle(pref.getTitle());
00117     return true;
00118 }
00119 }

```

6.55 SettingsHeaderFragment.java

```

00001 package com.paternaggschga.gwent.ui.settings;
00002
00003 import android.content.Intent;
00004 import android.os.Bundle;
00005
00006 import androidx.preference.Preference;
00007 import androidx.preference.PreferenceFragmentCompat;
00008
00009 import com.paternaggschga.gwent.R;
00010 import com.paternaggschga.gwent.ui.introduction.IntroductionActivity;
00011
00012 import java.util.Objects;
00013
00018 public class SettingsHeaderFragment extends PreferenceFragmentCompat {
00019     @Override
00020     public void onCreatePreferences(Bundle savedInstanceState, String rootKey) {
00021         setPreferencesFromResource(R.xml.header_preferences, rootKey);
00022
00023         Preference introductionPreference =
00024             Objects.requireNonNull(findPreference(getString(R.string.preference_key_introduction)));
00025         introductionPreference.setOnPreferenceClickListener(preference -> {
00026             startActivity(new Intent(getContext(), IntroductionActivity.class));
00027             return true;
00028         });
00029     }
00030 }

```

6.56 SettingsRuleFragment.java

```

00001 package com.paternaggschga.gwent.ui.settings;

```

```

00002
00003 import static com.paternaggschga.gwent.ui.settings.RuleSection.CARDS;
00004 import static com.paternaggschga.gwent.ui.settings.RuleSection.CARD_ABILITIES;
00005 import static com.paternaggschga.gwent.ui.settings.RuleSection.COMMANDER;
00006 import static com.paternaggschga.gwent.ui.settings.RuleSection.COURSE;
00007 import static com.paternaggschga.gwent.ui.settings.RuleSection.FACTIONS;
00008 import static com.paternaggschga.gwent.ui.settings.RuleSection.GENERAL;
00009 import static com.paternaggschga.gwent.ui.settings.RuleSection.SPECIAL_CARDS;
00010
00011 import android.content.Intent;
00012 import android.os.Bundle;
00013
00014 import androidx.annotation.Keep;
00015 import androidx.annotation.NonNull;
00016 import androidx.preference.Preference;
00017 import androidx.preference.PreferenceFragmentCompat;
00018
00019 import com.paternaggschga.gwent.R;
00020
00021 import org.jetbrains.annotations.Contract;
00022
00023 import java.io.Serializable;
00024 import java.util.Objects;
00025
00030 @Keep
00031 public class SettingsRuleFragment extends PreferenceFragmentCompat {
00041     @NonNull
00042     @Contract(pure = true)
00043     private Preference.OnPreferenceClickListener getSectionClickListener(@NonNull RuleSection section)
00044     {
00045         return preference -> {
00046             startActivity(
00047                 new Intent(getContext(), RuleActivity.class)
00048                     .putExtra(RuleActivity.INTENT_EXTRA_TAG, section)
00049             );
00050             return true;
00051         };
00052     }
00053
00065     @Override
00066     public void onCreatePreferences(Bundle savedInstanceState, String rootKey) {
00067         setPreferencesFromResource(R.xml.rule_preferences, rootKey);
00068
00069         Preference rulesGeneral =
00070             Objects.requireNonNull(findPreference(getString(R.string.preference_rules_general_key)));
00071         rulesGeneral.setOnPreferenceClickListener(getSectionClickListener(GENERAL));
00072
00073         Preference rulesCourse =
00074             Objects.requireNonNull(findPreference(getString(R.string.preference_rules_course_key)));
00075         rulesCourse.setOnPreferenceClickListener(getSectionClickListener(COURSE));
00076
00077         Preference ruleFactions =
00078             Objects.requireNonNull(findPreference(getString(R.string.preference_rules_factions_key)));
00079         ruleFactions.setOnPreferenceClickListener(getSectionClickListener(FACTIONS));
00080
00081         Preference rulesCommander =
00082             Objects.requireNonNull(findPreference(getString(R.string.preference_rules_commander_key)));
00083         rulesCommander.setOnPreferenceClickListener(getSectionClickListener(COMMANDER));
00084
00085         Preference rulesCards =
00086             Objects.requireNonNull(findPreference(getString(R.string.preference_rules_cards_key)));
00087         rulesCards.setOnPreferenceClickListener(getSectionClickListener(CARDS));
00088
00089         Preference ruleCardAbilities =
00090             Objects.requireNonNull(findPreference(getString(R.string.preference_rules_card_abilities_key)));
00091         ruleCardAbilities.setOnPreferenceClickListener(getSectionClickListener(CARD_ABILITIES));
00092
00093         Preference rulesSpecialCards =
00094             Objects.requireNonNull(findPreference(getString(R.string.preference_rules_special_cards_key)));
00095         rulesSpecialCards.setOnPreferenceClickListener(getSectionClickListener(SPECIAL_CARDS));
00096     }
00097 }

```

6.57 SettingsSoundFragment.java

```

00001 package com.paternaggschga.gwent.ui.settings;
00002
00003 import android.os.Bundle;
00004
00005 import androidx.annotation.Keep;
00006 import androidx.preference.Preference;
00007 import androidx.preference.PreferenceFragmentCompat;
00008
00009 import com.paternaggschga.gwent.R;
00010

```

```

00015 @Keep
00016 public class SettingsSoundFragment extends PreferenceFragmentCompat {
00026     @Override
00027     public void onCreatePreferences(Bundle savedInstanceState, String rootKey) {
00028         setPreferencesFromResource(R.xml.sound_preferences, rootKey);
00029     }
00030 }

```

6.58 Sound.java

```

00001 package com.paternaggschga.gwent.ui.sounds;
00002
00003 import android.content.Context;
00004 import android.content.SharedPreferences;
00005 import android.media.SoundPool;
00006
00007 import androidx.annotation.NonNull;
00008 import androidx.annotation.RawRes;
00009 import androidx.annotation.StringRes;
00010 import androidx.preference.PreferenceManager;
00011
00012 import com.paternaggschga.gwent.R;
00013
00014 import org.jetbrains.annotations.Contract;
00015
00020 class Sound {
00026     @NonNull
00027     private final String preferenceKey;
00028
00032     private final int soundId;
00033
00037     private boolean activated;
00038
00048     private Sound(@NonNull String preferenceKey, int soundId, boolean activated) {
00049         this.preferenceKey = preferenceKey;
00050         this.soundId = soundId;
00051         this.activated = activated;
00052     }
00053
00067     @NonNull
00068     @Contract("_, _, _, _ -> new")
00069     static Sound createSound(@NonNull Context context, @StringRes int preferenceRes,
00070                             @NonNull SoundPool soundPool, @RawRes int resId) {
00071         String preferenceKey = context.getString(preferenceRes);
00072         int soundId = soundPool.load(context, resId, 1);
00073         SharedPreferences pref = PreferenceManager.getDefaultSharedPreferences(context);
00074         boolean activated = context.getResources().getBoolean(R.bool.sound_preference_default);
00075         activated = pref.getBoolean(context.getString(R.string.preference_key_sounds_all), activated)
00076             && pref.getBoolean(preferenceKey, activated);
00077         return new Sound(preferenceKey, soundId, activated);
00078     }
00079
00085     @NonNull
00086     String getPreferenceKey() {
00087         return preferenceKey;
00088     }
00089
00095     int getSoundId() {
00096         return soundId;
00097     }
00098
00104     boolean isActivated() {
00105         return activated;
00106     }
00107
00114     void setActivated(boolean activated) {
00115         this.activated = activated;
00116     }
00117
00124     void setActivated(@NonNull SharedPreferences sharedPreferences) {
00125         setActivated(sharedPreferences.getBoolean(preferenceKey, activated));
00126     }
00127 }

```

6.59 SoundManager.java

```

00001 package com.paternaggschga.gwent.ui.sounds;
00002
00003 import android.content.Context;
00004 import android.content.SharedPreferences;
00005 import android.media.AudioAttributes;
00006 import android.media.SoundPool;

```



```

00007 import android.os.Build;
00008
00009 import androidx.annotation.IntRange;
00010 import androidx.annotation.NonNull;
00011 import androidx.preference.PreferenceManager;
00012
00013 import com.peternaggschga.gwent.R;
00014 import com.peternaggschga.gwent.data.RowType;
00015
00016 import java.util.Arrays;
00017
00023 public class SoundManager {
00030     public static final int SOUND_WEATHER_GOOD = 0;
00031
00038     public static final int SOUND_WEATHER_FROST = 1;
00039
00046     public static final int SOUND_WEATHER_FOG = 2;
00047
00054     public static final int SOUND_WEATHER_RAIN = 3;
00055
00062     public static final int SOUND_HORN = 4;
00063
00070     public static final int SOUND_CARDS_EPIC = 5;
00071
00078     public static final int SOUND_CARDS_MELEE = 6;
00079
00086     public static final int SOUND_CARDS_RANGE = 7;
00087
00094     public static final int SOUND_CARDS_SIEGE = 8;
00095
00102     public static final int SOUND_RESET = 9;
00103
00110     public static final int SOUND_BURN = 10;
00111
00118     public static final int SOUND_COIN = 11;
00119
00127     @NonNull
00128     private final Sound[] sounds = new Sound[SOUND_COIN + 1];
00129
00133     @NonNull
00134     private final SoundPool soundPool;
00135
00143     @NonNull
00144     @SuppressWarnings("FieldCanBeLocal")
00145     private final SharedPreferences.OnSharedPreferenceChangeListener changeListener;
00146
00156     public SoundManager(@NonNull Context context) {
00157         AudioAttributes attributes = new AudioAttributes.Builder()
00158             .setUsage(AudioAttributes.USAGE_GAME)
00159             .setContentType(AudioAttributes.CONTENT_TYPE_SONIFICATION)
00160             .build();
00161
00162         if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.UPSIDE_DOWN_CAKE) {
00163             soundPool = new
00164 SoundPool.Builder().setMaxStreams(5).setAudioAttributes(attributes).setContext(context).build();
00165         } else {
00166             soundPool = new
00167 SoundPool.Builder().setMaxStreams(5).setAudioAttributes(attributes).build();
00168         }
00169
00170         sounds[SOUND_WEATHER_GOOD] = Sound.createSound(context,
00171 R.string.preference_key_sounds_weather, soundPool, R.raw.weather_good);
00172         sounds[SOUND_WEATHER_FROST] = Sound.createSound(context,
00173 R.string.preference_key_sounds_weather, soundPool, R.raw.weather_frost);
00174         sounds[SOUND_WEATHER_FOG] = Sound.createSound(context, R.string.preference_key_sounds_weather,
00175 soundPool, R.raw.weather_fog);
00176         sounds[SOUND_WEATHER_RAIN] = Sound.createSound(context,
00177 R.string.preference_key_sounds_weather, soundPool, R.raw.weather_rain);
00178         sounds[SOUND_HORN] = Sound.createSound(context, R.string.preference_key_sounds_horn,
00179 soundPool, R.raw.horn);
00180         sounds[SOUND_CARDS_EPIC] = Sound.createSound(context, R.string.preference_key_sounds_cards,
00181 soundPool, R.raw.card_epic);
00182         sounds[SOUND_CARDS_MELEE] = Sound.createSound(context, R.string.preference_key_sounds_cards,
00183 soundPool, R.raw.card_melee);
00184         sounds[SOUND_CARDS_RANGE] = Sound.createSound(context, R.string.preference_key_sounds_cards,
00185 soundPool, R.raw.card_range);
00186         sounds[SOUND_CARDS_SIEGE] = Sound.createSound(context, R.string.preference_key_sounds_cards,
00187 soundPool, R.raw.card_siege);
00188         sounds[SOUND_RESET] = Sound.createSound(context, R.string.preference_key_sounds_reset,
00189 soundPool, R.raw.reset);
00190         sounds[SOUND_BURN] = Sound.createSound(context, R.string.preference_key_sounds_burn,
00191 soundPool, R.raw.burn);
00192         sounds[SOUND_COIN] = Sound.createSound(context, R.string.preference_key_sounds_coin,
00193 soundPool, R.raw.coin);
00194
00195         SharedPreferences pref = PreferenceManager.getDefaultSharedPreferences(context);
00196         changeListener = (sharedPreferences, key) -> {

```

```

00183         if (context.getResources().getString(R.string.preference_key_sounds_all).equals(key)) {
00184             if (sharedPreferences.getBoolean(key,
00185                 context.getResources().getBoolean(R.bool.sound_preference_default))) {
00186                 Arrays.stream(sounds).forEach(sound -> sound.setActivated(sharedPreferences));
00187             } else {
00188                 Arrays.stream(sounds).forEach(sound -> sound.setActivated(false));
00189             }
00190         } else {
00191             Arrays.stream(sounds)
00192                 .filter(sound -> sound.getPreferenceKey().equals(key))
00193                 .forEach(sound -> sound.setActivated(sharedPreferences.getBoolean(key,
00194                     context.getResources().getBoolean(R.bool.sound_preference_default))));
00195         }
00196     };
00197     pref.registerOnSharedPreferenceChangeListener(changeListener);
00198 }
00199
00213 public void playSound(@IntRange(from = SOUND_WEATHER_GOOD, to = SOUND_COIN) int soundId) {
00214     if (sounds[soundId].isActivated()) {
00215         soundPool.play(sounds[soundId].getSoundId(), 1, 1, 0, 0, 1);
00216     }
00217 }
00218
00225 public void playClearWeatherSound() {
00226     playSound(SOUND_WEATHER_GOOD);
00227 }
00228
00236 public void playWeatherSound(@NonNull RowType row) {
00237     switch (row) {
00238         case MELEE:
00239             playSound(SOUND_WEATHER_FROST);
00240             break;
00241         case RANGE:
00242             playSound(SOUND_WEATHER_FOG);
00243             break;
00244         case SIEGE:
00245             playSound(SOUND_WEATHER_RAIN);
00246     }
00247 }
00248
00255 public void playHornSound() {
00256     playSound(SOUND_HORN);
00257 }
00258
00268 public void playCardAddSound(@NonNull RowType row, boolean epic) {
00269     if (epic) {
00270         playSound(SOUND_CARDS_EPIC);
00271         return;
00272     }
00273     switch (row) {
00274         case MELEE:
00275             playSound(SOUND_CARDS_MELEE);
00276             break;
00277         case RANGE:
00278             playSound(SOUND_CARDS_RANGE);
00279             break;
00280         case SIEGE:
00281             playSound(SOUND_CARDS_SIEGE);
00282     }
00283 }
00284
00291 public void playCardRemovedSound() {
00292     playSound(SOUND_RESET);
00293 }
00294
00301 public void playResetSound() {
00302     playSound(SOUND_RESET);
00303 }
00304
00311 public void playBurnSound() {
00312     playSound(SOUND_BURN);
00313 }
00314
00320 public void playCoinSound() {
00321     playSound(SOUND_COIN);
00322 }
00323 }

```

6.60 background_drawable.xml

```

00001 <vector xmlns:android="http://schemas.android.com/apk/res/android"
00002     xmlns:tools="http://schemas.android.com/tools"
00003     android:width="200dp"
00004     android:height="117.63864dp"
00005     android:viewportWidth="510.03275"

```

```

00006     android:viewportHeight="299.9978">
00007
00008     <path
00009         android:fillColor="#5f4f2d"
00010         android:pathData="M-0,-0h510.033v299.998h-510.033z"
00011         android:strokeWidth="0.264583"
00012         android:strokeColor="#00000000" />
00013
00014     <path
00015         android:fillColor="#35270a"
00016         android:pathData="m495.074,15h15v-14.997zM15.004,15L0,15v-15zM465.07,15h30.004l-15,-15zM435.065,15h30.004l-15,-15z"
00017         android:strokeWidth="0.264583"
00018         android:strokeColor="#00000000"
00019         android:strokeLineCap="butt"
00020         android:strokeLineJoin="miter"
00021         tools:ignore="VectorPath" />
00022 </vector>

```

6.61 icon_copy.xml

```

00001 <vector xmlns:android="http://schemas.android.com/apk/res/android"
00002     android:width="50dp"
00003     android:height="50dp"
00004     android:autoMirrored="true"
00005     android:viewportWidth="112.7125"
00006     android:viewportHeight="112.7125">
00007
00008     <path
00009         android:fillColor="?attr/card_view_button_background"
00010         android:pathData="M56.356,56.356m-56.356,0a56.356,56.356 0,1 1,112.713 0a56.356,56.356 0,1
00011 1,-112.713 0"
00012         android:strokeWidth="0.907308" />
00013
00014     <path
00015         android:fillColor="#ffffff"
00016         android:pathData="M63.681,91.414L38.472,91.414c-5.822,0 -10.559,-4.914
-10.559,-10.955L27.914,43.347c0,-6.041 4.736,-10.955 10.559,-10.955h25.209c5.822,0 10.559,4.914
10.559,10.955v37.111c0,6.041 -4.736,10.955 -10.559,10.955zM38.472,37.869c-2.911,0 -5.279,2.457
-5.279,5.478v37.111c0,3.02 2.369,5.478 5.279,5.478h25.209c2.911,0 5.279,-2.457
5.279,-5.478L68.961,43.347c0,-3.02 -2.369,-5.478 -5.279,-5.478zM84.799,73.611L84.799,32.254c0,-6.041
-4.736,-10.955 -10.559,-10.955L44.94,21.299c-1.458,0 -2.64,1.226 -2.64,2.739 0,1.513 1.182,2.739
2.64,2.739h29.301c2.911,0 5.279,2.457 5.279,5.478v41.357c0,1.513 1.182,2.739 2.64,2.739 1.458,0
2.64,-1.226 2.64,-2.739zM84.799,73.611"
00016         android:strokeWidth="0.13444" />
00017 </vector>

```

6.62 icon_delete.xml

```

00001 <vector xmlns:android="http://schemas.android.com/apk/res/android"
00002     android:width="50dp"
00003     android:height="50dp"
00004     android:viewportWidth="150.28333"
00005     android:viewportHeight="150.28333">
00006
00007     <path
00008         android:fillColor="?attr/card_view_button_background"
00009         android:pathData="M75.142,75.142m-75.142,0a75.142,75.142 0,1 1,150.283 0a75.142,75.142 0,1
00010 1,-150.283 0"
00011         android:strokeWidth="0.264583" />
00012
00013     <path
00014         android:fillColor="#ffffff"
00015         android:pathData="m112.311,48.066l-63.967,63.847l-10.372,-10.393l63.967,-63.847zM48.414,37.623
00016 L112.261,101.59 101.868,111.962 38.021,47.996z"
00015         android:strokeWidth="0.264583" />
00016 </vector>

```

6.63 icon_design.xml

```

00001 <vector xmlns:android="http://schemas.android.com/apk/res/android"
00002     android:width="24dp"
00003     android:height="24dp"
00004     android:tint="?android:attr/colorControlNormal"
00005     android:viewportWidth="24"
00006     android:viewportHeight="24">
00007
00008     <path
00009         android:fillColor="@android:color/white"

```

```

00010         android:pathData="M21,19V5c0,-1.1 -0.9,-2 -2,-2H5c-1.1,0 -2,0.9 -2,2v14c0,1.1 0.9,2
2,2h14c1.1,0 2,-0.9 2,-2zM8.5,13.5l2.5,3.0lL14.5,12l4.5,6H5l3.5,-4.5z" />
00011 </vector>

```

6.64 icon_explore.xml

```

00001 <vector xmlns:android="http://schemas.android.com/apk/res/android"
00002     android:width="24dp"
00003     android:height="24dp"
00004     android:autoMirrored="true"
00005     android:tint="?android:attr/colorControlNormal"
00006     android:viewportWidth="24"
00007     android:viewportHeight="24">
00008     <path
00009         android:fillColor="@android:color/white"
00010         android:pathData="M12,10.9c-0.61,0 -1.1,0.49 -1.1,1.1s0.49,1.1 1.1,1.1c0.61,0 1.1,-0.49
1.1,-1.1s-0.49,-1.1 -1.1,-1.1zM12,2C6.48,2 2,6.48 2,12s4.48,10 10,10 -4.48 10,-10S17.52,2
12,2zM14.19,14.19L6,18l3.81,-8.19L18,6l-3.81,8.19z" />
00011 </vector>

```

6.65 icon_factionreset.xml

```

00001 <vector xmlns:android="http://schemas.android.com/apk/res/android"
00002     android:width="24dp"
00003     android:height="24dp"
00004     android:tint="?android:attr/colorControlNormal"
00005     android:viewportWidth="24"
00006     android:viewportHeight="24">
00007
00008     <path
00009         android:fillColor="@android:color/white"
00010         android:pathData="M12,6v3l4,-4 -4,-4v3c-4.42,0 -8,3.58 -8,8 0,1.57 0.46,3.03
1.24,4.26L6.7,14.8c-0.45,-0.83 -0.7,-1.79 -0.7,-2.8 0,-3.31 2.69,-6
6,-6zM18.76,7.74L17.3,9.2c0.44,0.84 0.7,1.79 0.7,2.8 0,3.31 -2.69,6 -6,6v-3l-4,4 4v-3c4.42,0 8,-3.58
8,-8 0,-1.57 -0.46,-3.03 -1.24,-4.26z" />
00011 </vector>

```

6.66 icon_next.xml

```

00001 <vector xmlns:android="http://schemas.android.com/apk/res/android"
00002     android:width="24dp"
00003     android:height="24dp"
00004     android:autoMirrored="true"
00005     android:tint="#FFFFFF"
00006     android:viewportWidth="24"
00007     android:viewportHeight="24">
00008     <path
00009         android:fillColor="@android:color/white"
00010         android:pathData="M10,6L8.59,7.41 13.17,12l-4.58,4.59L10,18l6,-6z" />
00011 </vector>

```

6.67 icon_rules.xml

```

00001 <vector xmlns:android="http://schemas.android.com/apk/res/android"
00002     android:width="24dp"
00003     android:height="24dp"
00004     android:tint="?android:attr/colorControlNormal"
00005     android:viewportWidth="24"
00006     android:viewportHeight="24">
00007
00008     <path
00009         android:fillColor="@android:color/white"
00010         android:pathData="M18,2H6c-1.1,0 -2,0.9 -2,2v16c0,1.1 0.9,2 2,2h12c1.1,0 2,-0.9 2,-2V4c0,-1.1
-0.9,-2 -2,-2zM6,4h5v8l-2.5,-1.5L6,12V4z" />
00011 </vector>

```

6.68 icon_sounds.xml

```

00001 <vector xmlns:android="http://schemas.android.com/apk/res/android"
00002     android:width="24dp"
00003     android:height="24dp"
00004     android:tint="?android:attr/colorControlNormal"
00005     android:viewportWidth="24"
00006     android:viewportHeight="24">

```

```

00007
00008     <path
00009         android:fillColor="@android:color/white"
00010         android:pathData="M12,3v10.55c-0.59,-0.34 -1.27,-0.55 -2,-0.55 -2.21,0 -4,1.79 -4,4s1.79,4 4,4
00011         4,-1.79 4,-4V7h4V3h-6z" />
00011 </vector>

```

6.69 icon_warnings.xml

```

00001 <vector xmlns:android="http://schemas.android.com/apk/res/android"
00002     android:width="24dp"
00003     android:height="24dp"
00004     android:tint="?android:attr/colorControlNormal"
00005     android:viewportWidth="24"
00006     android:viewportHeight="24">
00007
00008     <path
00009         android:fillColor="@android:color/white"
00010         android:pathData="M1,21h22L12,2 1,21zM13,18h-2v-2h2v2zM13,14h-2v-4h2v4z" />
00011 </vector>

```

6.70 indicator_selected.xml

```

00001 <shape xmlns:android="http://schemas.android.com/apk/res/android"
00002     android:shape="oval">
00003     <corners android:radius="100dp" />
00004     <solid android:color="@android:color/primary_text_dark" />
00005 </shape>

```

6.71 indicator_unselected.xml

```

00001 <shape xmlns:android="http://schemas.android.com/apk/res/android"
00002     android:shape="oval">
00003     <corners android:radius="100dp" />
00004     <solid android:color="@android:color/tab_indicator_text" />
00005 </shape>

```

6.72 rectangle.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <shape xmlns:android="http://schemas.android.com/apk/res/android"
00003     android:shape="rectangle">
00004
00005     <stroke
00006         android:width="@dimen/rectangle_stroke_width"
00007         android:color="?attr/colorPrimary" />
00008
00009     <corners android:radius="@dimen/rectangle_corner_radius" />
00010
00011     <gradient
00012         android:angle="90"
00013         android:centerColor="?attr/card_background_color"
00014         android:endColor="?attr/card_background_gradient"
00015         android:startColor="?attr/card_background_gradient" />
00016 </shape>

```

6.73 rectangle_black_transparent.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <shape xmlns:android="http://schemas.android.com/apk/res/android"
00003     android:shape="rectangle">
00004
00005     <solid android:color="@color/color_popup_background" />
00006
00007     <corners android:radius="@dimen/rectangle_corner_radius" />
00008 </shape>

```

6.74 rectangle_cancel_popup.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <shape xmlns:android="http://schemas.android.com/apk/res/android"

```

```

00003     android:shape="rectangle">
00004
00005     <gradient
00006         android:angle="90"
00007         android:centerColor="@color/color_cancel_popup_background"
00008         android:endColor="@color/color_cancel_popup_background_gradient"
00009         android:startColor="@color/color_cancel_popup_background_gradient" />
00010
00011     <corners android:radius="@dimen/rectangle_corner_radius" />
00012
00013     <stroke
00014         android:width="@dimen/rectangle_stroke_width"
00015         android:color="@color/color_popup_cancel_text" />
00016 </shape>

```

6.75 fragment_introduction.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     xmlns:tools="http://schemas.android.com/tools"
00006     android:layout_width="match_parent"
00007     android:layout_height="match_parent"
00008     tools:context=".ui.introduction.IntroductionFragment">
00009
00010     <ImageView
00011         android:id="@+id/introduction_imageView"
00012         android:layout_width="0dp"
00013         android:layout_height="match_parent"
00014         android:layout_margin="@dimen/activity_horizontal_margin"
00015         android:contentDescription="@string/content_description_introduction_image"
00016         app:layout_constraintBottom_toBottomOf="parent"
00017         app:layout_constraintEnd_toStartOf="@id/introduction_scrollView"
00018         app:layout_constraintStart_toStartOf="parent"
00019         app:layout_constraintTop_toTopOf="parent"
00020         app:layout_constraintWidth_percent=".5" />
00021
00022     <ScrollView
00023         android:id="@+id/introduction_scrollView"
00024         android:layout_width="0dp"
00025         android:layout_height="wrap_content"
00026         app:layout_constraintBottom_toBottomOf="parent"
00027         app:layout_constraintEnd_toEndOf="parent"
00028         app:layout_constraintStart_toEndOf="@id/introduction_imageView"
00029         app:layout_constraintTop_toTopOf="parent">
00030
00031         <TextView
00032             android:id="@+id/introduction_textView"
00033             android:layout_width="match_parent"
00034             android:layout_height="wrap_content"
00035             android:layout_marginHorizontal="@dimen/activity_horizontal_margin"
00036             android:justificationMode="inter_word"
00037             android:textSize="@dimen/rules_textSize" />
00038     </ScrollView>
00039 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.76 fragment_introduction.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     xmlns:tools="http://schemas.android.com/tools"
00006     android:layout_width="match_parent"
00007     android:layout_height="match_parent"
00008     tools:context=".ui.introduction.IntroductionFragment">
00009
00010     <ImageView
00011         android:id="@+id/introduction_imageView"
00012         android:layout_width="0dp"
00013         android:layout_height="match_parent"
00014         android:layout_margin="@dimen/activity_horizontal_margin"
00015         android:contentDescription="@string/content_description_introduction_image"
00016         app:layout_constraintBottom_toBottomOf="parent"
00017         app:layout_constraintEnd_toStartOf="@id/introduction_scrollView"
00018         app:layout_constraintStart_toStartOf="parent"
00019         app:layout_constraintTop_toTopOf="parent"
00020         app:layout_constraintWidth_percent=".5" />
00021
00022     <ScrollView

```

```

00022         android:id="@+id/introduction_scrollView"
00023         android:layout_width="0dp"
00024         android:layout_height="wrap_content"
00025         app:layout_constraintBottom_toBottomOf="parent"
00026         app:layout_constraintEnd_toEndOf="parent"
00027         app:layout_constraintStart_toEndOf="@id/introduction_imageView"
00028         app:layout_constraintTop_toTopOf="parent">
00029
00030         <TextView
00031             android:id="@+id/introduction_textView"
00032             android:layout_width="match_parent"
00033             android:layout_height="wrap_content"
00034             android:layout_marginHorizontal="@dimen/activity_horizontal_margin"
00035             android:textSize="@dimen/rules_textSize" />
00036     </ScrollView>
00037
00038 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.77 fragment_introduction.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     xmlns:tools="http://schemas.android.com/tools"
00006     android:layout_width="match_parent"
00007     android:layout_height="match_parent"
00008     tools:context=".ui.introduction.IntroductionFragment">
00009
00010     <ImageView
00011         android:id="@+id/introduction_imageView"
00012         android:layout_width="match_parent"
00013         android:layout_height="0dp"
00014         android:layout_margin="@dimen/activity_horizontal_margin"
00015         android:contentDescription="@string/content_description_introduction_image"
00016         app:layout_constraintBottom_toTopOf="@id/introduction_scrollView"
00017         app:layout_constraintEnd_toEndOf="parent"
00018         app:layout_constraintStart_toStartOf="parent"
00019         app:layout_constraintTop_toTopOf="parent"
00020         app:layout_constraintVertical_weight="1" />
00021
00022     <ScrollView
00023         android:id="@+id/introduction_scrollView"
00024         android:layout_width="match_parent"
00025         android:layout_height="0dp"
00026         app:layout_constraintBottom_toBottomOf="parent"
00027         app:layout_constraintEnd_toEndOf="parent"
00028         app:layout_constraintStart_toStartOf="parent"
00029         app:layout_constraintTop_toBottomOf="@id/introduction_imageView"
00030         app:layout_constraintVertical_weight="2">
00031
00032         <TextView
00033             android:id="@+id/introduction_textView"
00034             android:layout_width="match_parent"
00035             android:layout_height="wrap_content"
00036             android:layout_gravity="center"
00037             android:layout_marginHorizontal="@dimen/activity_horizontal_margin"
00038             android:justificationMode="inter_word"
00039             android:textSize="@dimen/rules_textSize" />
00040     </ScrollView>
00041 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.78 fragment_introduction.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     xmlns:tools="http://schemas.android.com/tools"
00006     android:layout_width="match_parent"
00007     android:layout_height="match_parent"
00008     tools:context=".ui.introduction.IntroductionFragment">
00009
00010     <ImageView
00011         android:id="@+id/introduction_imageView"
00012         android:layout_width="match_parent"
00013         android:layout_height="0dp"
00014         android:layout_margin="@dimen/activity_horizontal_margin"
00015         android:contentDescription="@string/content_description_introduction_image"
00016         app:layout_constraintBottom_toTopOf="@id/introduction_scrollView"
00017         app:layout_constraintEnd_toEndOf="parent"

```

```

00017         app:layout_constraintStart_toStartOf="parent"
00018         app:layout_constraintTop_toTopOf="parent"
00019         app:layout_constraintVertical_weight="1" />
00020
00021     <ScrollView
00022         android:id="@+id/introduction_scrollView"
00023         android:layout_width="match_parent"
00024         android:layout_height="0dp"
00025         app:layout_constraintBottom_toBottomOf="parent"
00026         app:layout_constraintEnd_toEndOf="parent"
00027         app:layout_constraintStart_toStartOf="parent"
00028         app:layout_constraintTop_toBottomOf="@id/introduction_imageView"
00029         app:layout_constraintVertical_weight="2">
00030
00031         <TextView
00032             android:id="@+id/introduction_textView"
00033             android:layout_width="match_parent"
00034             android:layout_height="wrap_content"
00035             android:layout_gravity="center"
00036             android:layout_marginHorizontal="@dimen/activity_horizontal_margin"
00037             android:textSize="@dimen/rules_textSize" />
00038     </ScrollView>
00039
00040 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.79 activity_rules.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     android:layout_width="match_parent"
00006     android:layout_height="match_parent">
00007
00008     <androidx.appcompat.widget.Toolbar
00009         android:id="@+id/rulesToolbar"
00010         android:layout_width="match_parent"
00011         android:layout_height="?attr/actionBarSize"
00012         app:layout_constraintTop_toTopOf="parent" />
00013
00014     <ScrollView
00015         android:layout_width="match_parent"
00016         android:layout_height="0dp"
00017         android:layout_marginBottom="@dimen/margin_row_buttons"
00018         app:layout_constraintBottom_toBottomOf="parent"
00019         app:layout_constraintTop_toBottomOf="@id/rulesToolbar">
00020
00021         <TextView
00022             android:id="@+id/rulesTextView"
00023             android:layout_width="match_parent"
00024             android:layout_height="wrap_content"
00025             android:layout_marginHorizontal="@dimen/activity_horizontal_margin"
00026             android:justificationMode="inter_word"
00027             android:textColor="@android:color/secondary_text_dark"
00028             android:textSize="@dimen/rules_textSize" />
00029     </ScrollView>
00030 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.80 activity_rules.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     android:layout_width="match_parent"
00006     android:layout_height="match_parent">
00007
00008     <androidx.appcompat.widget.Toolbar
00009         android:id="@+id/rulesToolbar"
00010         android:layout_width="match_parent"
00011         android:layout_height="?attr/actionBarSize"
00012         app:layout_constraintTop_toTopOf="parent" />
00013
00014     <ScrollView
00015         android:layout_width="match_parent"
00016         android:layout_height="0dp"
00017         android:layout_marginBottom="@dimen/margin_row_buttons"
00018         app:layout_constraintBottom_toBottomOf="parent"
00019         app:layout_constraintTop_toBottomOf="@id/rulesToolbar">
00020
00021         <TextView
00022             android:id="@+id/rulesTextView"

```



```

00022         android:layout_width="match_parent"
00023         android:layout_height="wrap_content"
00024         android:layout_marginHorizontal="@dimen/activity_horizontal_margin"
00025         android:textColor="@android:color/secondary_text_dark"
00026         android:textSize="@dimen/rules_textSize" />
00027     </ScrollView>
00028 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.81 activity_introduction.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     xmlns:tools="http://schemas.android.com/tools"
00006     android:layout_width="match_parent"
00007     android:layout_height="match_parent"
00008     android:fitsSystemWindows="true"
00009     tools:context=".ui.introduction.IntroductionActivity">
00010     <androidx.viewpager2.widget.ViewPager2
00011         android:id="@+id/introduction_viewPager"
00012         android:layout_width="match_parent"
00013         android:layout_height="0dp"
00014         android:layout_marginTop="@dimen/activity_vertical_margin"
00015         app:layout_behavior="@string/appbar_scrolling_view_behavior"
00016         app:layout_constraintBottom_toTopOf="@id/introduction_bottomLayout"
00017         app:layout_constraintTop_toTopOf="parent" />
00018
00019     <View
00020         android:layout_width="match_parent"
00021         android:layout_height="@dimen/divider_strength"
00022         android:alpha="0.12"
00023         android:background="@android:color/white"
00024         app:layout_constraintBottom_toTopOf="@id/introduction_bottomLayout" />
00025
00026     <FrameLayout
00027         android:id="@+id/introduction_bottomLayout"
00028         android:layout_width="match_parent"
00029         android:layout_height="?attr/actionBarSize"
00030         android:layout_marginStart="@dimen/activity_horizontal_margin"
00031         android:layout_marginEnd="@dimen/activity_horizontal_margin"
00032         app:layout_constraintBottom_toBottomOf="parent">
00033
00034         <Button
00035             android:id="@+id/introduction_button_skip"
00036             style="@style/Widget.AppCompat.Button.Borderless"
00037             android:layout_width="wrap_content"
00038             android:layout_height="wrap_content"
00039             android:layout_gravity="start|center_vertical"
00040             android:text="@string/introduction_button_skip" />
00041
00042         <LinearLayout
00043             android:layout_width="wrap_content"
00044             android:layout_height="wrap_content"
00045             android:layout_gravity="center"
00046             android:orientation="horizontal">
00047
00048             <ImageView
00049                 android:id="@+id/introduction_indicator_0"
00050                 android:layout_width="@dimen/indicator_radius"
00051                 android:layout_height="@dimen/indicator_radius"
00052                 android:layout_marginEnd="@dimen/border_margin"
00053                 app:srcCompat="@drawable/indicator_selected"
00054                 android:contentDescription="@string/content_description_introduction_indicator" />
00055
00056             <ImageView
00057                 android:id="@+id/introduction_indicator_1"
00058                 android:layout_width="@dimen/indicator_radius"
00059                 android:layout_height="@dimen/indicator_radius"
00060                 android:layout_marginEnd="@dimen/border_margin"
00061                 app:srcCompat="@drawable/indicator_unselected"
00062                 android:contentDescription="@string/content_description_introduction_indicator" />
00063
00064             <ImageView
00065                 android:id="@+id/introduction_indicator_2"
00066                 android:layout_width="@dimen/indicator_radius"
00067                 android:layout_height="@dimen/indicator_radius"
00068                 android:layout_marginEnd="@dimen/border_margin"
00069                 app:srcCompat="@drawable/indicator_unselected"
00070                 android:contentDescription="@string/content_description_introduction_indicator" />
00071
00072             <ImageView
00073                 android:id="@+id/introduction_indicator_3"
00074                 android:layout_width="@dimen/indicator_radius"

```

```

00075         android:layout_height="@dimen/indicator_radius"
00076         android:layout_marginEnd="@dimen/border_margin"
00077         app:srcCompat="@drawable/indicator_unselected"
00078         android:contentDescription="@string/content_description_introduction_indicator" />
00079
00080     <ImageView
00081         android:id="@+id/introduction_indicator_4"
00082         android:layout_width="@dimen/indicator_radius"
00083         android:layout_height="@dimen/indicator_radius"
00084         app:srcCompat="@drawable/indicator_unselected"
00085         android:contentDescription="@string/content_description_introduction_indicator" />
00086 </LinearLayout>
00087
00088 <Button
00089     android:id="@+id/introduction_button_finish"
00090     style="@style/Widget.AppCompat.Button.Borderless"
00091     android:layout_width="wrap_content"
00092     android:layout_height="wrap_content"
00093     android:layout_gravity="end|center_vertical"
00094     android:text="@string/introduction_button_finish"
00095     android:visibility="gone" />
00096
00097 <ImageButton
00098     android:id="@+id/introduction_button_next"
00099     style="@style/Widget.AppCompat.Button.Borderless"
00100     android:layout_width="wrap_content"
00101     android:layout_height="wrap_content"
00102     android:layout_gravity="end|center_vertical"
00103     android:contentDescription="@string/content_description_introduction_next"
00104     android:padding="@dimen/activity_horizontal_margin"
00105     app:srcCompat="@drawable/icon_next" />
00106 </FrameLayout>
00107
00108 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.82 activity_main.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     xmlns:tools="http://schemas.android.com/tools"
00006     android:layout_width="match_parent"
00007     android:layout_height="match_parent"
00008     android:background="@drawable/background_drawable"
00009     tools:context=".ui.main.MainActivity">
00010
00011     <ImageView
00012         android:id="@+id/backgroundImageView"
00013         android:layout_width="match_parent"
00014         android:layout_height="match_parent"
00015         android:contentDescription="@string/content_description_activity_main_background"
00016         android:scaleType="centerCrop"
00017         android:visibility="gone" />
00018
00019     <LinearLayout
00020         android:layout_width="match_parent"
00021         android:layout_height="wrap_content"
00022         android:baselineAligned="false"
00023         android:orientation="horizontal"
00024         app:layout_constraintBottom_toBottomOf="parent"
00025         app:layout_constraintTop_toTopOf="parent">
00026
00027         <include
00028             android:id="@+id/firstRow"
00029             layout="@layout/row"
00030             android:layout_width="0dp"
00031             android:layout_height="wrap_content"
00032             android:layout_gravity="center"
00033             android:layout_marginHorizontal="@dimen/border_margin"
00034             android:layout_marginVertical="@dimen/activity_vertical_margin"
00035             android:layout_weight="1" />
00036
00037         <include
00038             android:id="@+id/secondRow"
00039             layout="@layout/row"
00040             android:layout_width="0dp"
00041             android:layout_height="wrap_content"
00042             android:layout_gravity="center"
00043             android:layout_marginHorizontal="@dimen/border_margin"
00044             android:layout_marginVertical="@dimen/activity_vertical_margin"
00045             android:layout_weight="1" />
00046
00047         <include
00048             android:id="@+id/thirdRow"

```

```

00048         layout="@layout/row"
00049         android:layout_width="0dp"
00050         android:layout_height="wrap_content"
00051         android:layout_gravity="center"
00052         android:layout_marginVertical="@dimen/activity_vertical_margin"
00053         android:layout_marginStart="@dimen/border_margin"
00054         android:layout_marginEnd="@dimen/margin_row_buttons"
00055         android:layout_weight="1" />
00056
00057     <androidx.constraintlayout.widget.ConstraintLayout
00058         android:layout_width="0dp"
00059         android:layout_height="match_parent"
00060         android:layout_gravity="center"
00061         android:layout_marginVertical="@dimen/activity_vertical_margin"
00062         android:layout_marginStart="@dimen/margin_row_buttons"
00063         android:layout_marginEnd="@dimen/border_margin"
00064         android:layout_weight="1">
00065
00066         <androidx.constraintlayout.widget.ConstraintLayout
00067             android:id="@+id/overallPointLayout"
00068             android:layout_width="match_parent"
00069             android:layout_height="0dp"
00070             android:layout_margin="@dimen/margin_menu_buttons"
00071             app:layout_constraintBottom_toTopOf="@id/menuView"
00072             app:layout_constraintTop_toTopOf="parent"
00073             app:layout_constraintVertical_weight="1">
00074
00075             <ImageView
00076                 android:id="@+id/overallPointBall"
00077                 android:layout_width="match_parent"
00078                 android:layout_height="match_parent"
00079                 android:background="@null"
00080                 android:contentDescription="@string/content_description_activity_main_point_ball"
00081                 android:scaleType="fitCenter"
00082                 android:src="@attr/point_ball_mipmap"
00083                 android:translationZ="5dp" />
00084
00085             <TextView
00086                 android:id="@+id/overallPointView"
00087                 android:layout_width="wrap_content"
00088                 android:layout_height="wrap_content"
00089                 android:textColor="@color/circle_text"
00090                 android:textSize="@dimen/overall_point_circle_textSize"
00091                 android:translationZ="10dp"
00092                 app:layout_constraintBottom_toBottomOf="parent"
00093                 app:layout_constraintEnd_toEndOf="parent"
00094                 app:layout_constraintStart_toStartOf="parent"
00095                 app:layout_constraintTop_toTopOf="parent" />
00096         </androidx.constraintlayout.widget.ConstraintLayout>
00097
00098         <androidx.constraintlayout.widget.ConstraintLayout
00099             android:id="@+id/menuView"
00100             android:layout_width="match_parent"
00101             android:layout_height="0dp"
00102             app:layout_constraintBottom_toBottomOf="parent"
00103             app:layout_constraintTop_toBottomOf="@id/overallPointLayout"
00104             app:layout_constraintVertical_weight="3">
00105
00106             <ImageButton
00107                 android:id="@+id/factionButton"
00108                 android:layout_width="0dp"
00109                 android:layout_height="0dp"
00110                 android:layout_margin="@dimen/margin_menu_buttons"
00111                 android:background="@null"
00112
00113                 android:contentDescription="@string/content_description_activity_main_faction_button"
00114                 android:scaleType="fitCenter"
00115                 android:src="@android:attr/alertDialogIcon"
00116                 app:layout_constraintBottom_toTopOf="@id/weatherButton"
00117                 app:layout_constraintEnd_toStartOf="@id/resetButton"
00118                 app:layout_constraintStart_toStartOf="parent"
00119                 app:layout_constraintTop_toTopOf="parent" />
00120
00121             <ImageButton
00122                 android:id="@+id/resetButton"
00123                 android:layout_width="0dp"
00124                 android:layout_height="0dp"
00125                 android:layout_margin="@dimen/margin_menu_buttons"
00126                 android:background="@null"
00127                 android:clickable="false"
00128
00129                 android:contentDescription="@string/content_description_activity_main_reset_button"
00130                 android:scaleType="fitCenter"
00131                 android:src="@drawable/icon_reset_grey"
00132                 app:layout_constraintBottom_toTopOf="@id/burnButton"
00133                 app:layout_constraintEnd_toEndOf="parent"
00134                 app:layout_constraintStart_toEndOf="@id/factionButton"

```

```

00133         app:layout_constraintTop_toTopOf="parent" />
00134
00135     <ImageButton
00136         android:id="@+id/weatherButton"
00137         android:layout_width="0dp"
00138         android:layout_height="0dp"
00139         android:layout_margin="@dimen/margin_menu_buttons"
00140         android:background="@null"
00141         android:clickable="false"
00142
00143         android:contentDescription="@string/content_description_activity_main_weather_button"
00144         android:scaleType="fitCenter"
00145         android:src="@drawable/icon_weather_grey"
00146         app:layout_constraintBottom_toTopOf="@id/coinButton"
00147         app:layout_constraintEnd_toStartOf="@id/burnButton"
00148         app:layout_constraintStart_toStartOf="parent"
00149         app:layout_constraintTop_toBottomOf="@id/factionButton" />
00150
00151     <ImageButton
00152         android:id="@+id/burnButton"
00153         android:layout_width="0dp"
00154         android:layout_height="0dp"
00155         android:layout_margin="@dimen/margin_menu_buttons"
00156         android:background="@null"
00157         android:clickable="false"
00158         android:contentDescription="@string/content_description_activity_main_burn_button"
00159         android:scaleType="fitCenter"
00160         android:src="@drawable/icon_burn_grey"
00161         app:layout_constraintBottom_toTopOf="@id/settingsButton"
00162         app:layout_constraintEnd_toEndOf="parent"
00163         app:layout_constraintStart_toEndOf="@+id/weatherButton"
00164         app:layout_constraintTop_toBottomOf="@id/resetButton" />
00165
00166     <ImageButton
00167         android:id="@+id/coinButton"
00168         android:layout_width="0dp"
00169         android:layout_height="0dp"
00170         android:layout_margin="@dimen/margin_menu_buttons"
00171         android:background="@null"
00172         android:contentDescription="@string/content_description_activity_main_coin_button"
00173         android:scaleType="fitCenter"
00174         android:src="@drawable/icon_coin"
00175         app:layout_constraintBottom_toBottomOf="parent"
00176         app:layout_constraintEnd_toStartOf="@id/settingsButton"
00177         app:layout_constraintStart_toStartOf="parent"
00178         app:layout_constraintTop_toBottomOf="@id/weatherButton" />
00179
00180     <ImageButton
00181         android:id="@+id/settingsButton"
00182         android:layout_width="0dp"
00183         android:layout_height="0dp"
00184         android:layout_margin="@dimen/margin_menu_buttons"
00185         android:background="@null"
00186
00187         android:contentDescription="@string/content_description_activity_main_settings_button"
00188         android:scaleType="fitCenter"
00189         android:src="@drawable/icon_settings"
00190         app:layout_constraintBottom_toBottomOf="parent"
00191         app:layout_constraintEnd_toEndOf="parent"
00192         app:layout_constraintStart_toEndOf="@+id/coinButton"
00193         app:layout_constraintTop_toBottomOf="@id/burnButton" />
00194     </androidx.constraintlayout.widget.ConstraintLayout>
00195 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.83 activity_settings.xml

```

00001 <androidx.constraintlayout.widget.ConstraintLayout
00002     xmlns:android="http://schemas.android.com/apk/res/android"
00003     xmlns:app="http://schemas.android.com/apk/res-auto"
00004     android:layout_width="match_parent"
00005     android:layout_height="match_parent"
00006     android:fitsSystemWindows="true">
00007
00008     <androidx.appcompat.widget.Toolbar
00009         android:id="@+id/settingsToolbar"
00010         android:layout_width="match_parent"
00011         android:layout_height="?attr/actionBarSize"
00012         app:layout_constraintBottom_toTopOf="@id/settingsFrameLayout"
00013         app:layout_constraintTop_toTopOf="parent"
00014         app:title="@string/settings_title" />
00015
00016     <FrameLayout
00017         android:id="@+id/settingsFrameLayout"

```

```

00017         android:layout_width="match_parent"
00018         android:layout_height="0dp"
00019         app:layout_constraintBottom_toBottomOf="parent"
00020         app:layout_constraintTop_toBottomOf="@id/settingsToolbar" />
00021 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.84 alertdialog_checkbox.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
00003     android:layout_width="match_parent"
00004     android:layout_height="match_parent">
00005
00006     <CheckBox
00007         android:id="@+id/alertDialog_checkbox"
00008         android:layout_width="match_parent"
00009         android:layout_height="wrap_content"
00010         android:layout_marginStart="@dimen/alertDialog_checkBox_marginStart"
00011         android:layout_marginEnd="@dimen/alertDialog_checkBox_marginEnd"
00012         android:checked="true"
00013         android:layoutDirection="rtl"
00014         android:minHeight="@dimen/alertDialog_checkBox_minHeight"
00015         android:text="@string/alertDialog_reset_checkbox" />
00016 </FrameLayout>

```

6.85 card.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     android:layout_width="@dimen/popUp_card_width"
00006     android:layout_height="match_parent"
00007     android:layout_margin="@dimen/border_margin"
00008     android:background="@drawable/rectangle">
00009
00010     <TextView
00011         android:id="@+id/damageView"
00012         android:layout_width="@dimen/popUp_card_image_diameter"
00013         android:layout_height="@dimen/popUp_card_image_diameter"
00014         android:background="@drawable/icon_damage_background"
00015         android:gravity="center"
00016         android:textColor="@color/color_damage_textColor"
00017         android:textSize="@dimen/popUp_card_textSize"
00018         android:textStyle="bold"
00019         app:layout_constraintBottom_toTopOf="@+id/abilityView"
00020         app:layout_constraintEnd_toEndOf="parent"
00021         app:layout_constraintStart_toStartOf="parent"
00022         app:layout_constraintTop_toTopOf="parent" />
00023
00024     <ImageView
00025         android:id="@+id/abilityView"
00026         android:layout_width="@dimen/popUp_card_image_diameter"
00027         android:layout_height="@dimen/popUp_card_image_diameter"
00028         android:contentDescription="@string/content_description_card_ability"
00029         android:visibility="gone"
00030         app:layout_constraintBottom_toTopOf="@id/copyButton"
00031         app:layout_constraintEnd_toStartOf="@id/bindingView"
00032         app:layout_constraintStart_toStartOf="parent"
00033         app:layout_constraintTop_toBottomOf="@+id/damageView"
00034         app:srcCompat="@drawable/icon_binding" />
00035
00036     <TextView
00037         android:id="@+id/bindingView"
00038         android:layout_width="@dimen/popUp_card_image_diameter"
00039         android:layout_height="@dimen/popUp_card_image_diameter"
00040         android:background="@drawable/icon_damage_background"
00041         android:gravity="center"
00042         android:textColor="@color/color_damage_textColor"
00043         android:textSize="@dimen/popUp_card_textSize"
00044         android:textStyle="bold"
00045         android:visibility="gone"
00046         app:layout_constraintBottom_toBottomOf="@id/abilityView"
00047         app:layout_constraintEnd_toEndOf="parent"
00048         app:layout_constraintStart_toEndOf="@id/abilityView"
00049         app:layout_constraintTop_toTopOf="@+id/abilityView" />
00050
00051     <ImageButton
00052         android:id="@+id/copyButton"
00053         android:layout_width="@dimen/popUp_card_image_diameter"
00054         android:layout_height="@dimen/popUp_card_image_diameter"
00055         android:background="@null"

```

```

00055         android:contentDescription="@string/content_description_card_copy"
00056         app:layout_constraintBottom_toBottomOf="parent"
00057         app:layout_constraintEnd_toStartOf="@id/deleteButton"
00058         app:layout_constraintStart_toStartOf="parent"
00059         app:layout_constraintTop_toBottomOf="@id/abilityView"
00060         app:srcCompat="@drawable/icon_copy" />
00061
00062     <ImageButton
00063         android:id="@+id/deleteButton"
00064         android:layout_width="@dimen/popUp_card_image_diameter"
00065         android:layout_height="@dimen/popUp_card_image_diameter"
00066         android:background="@null"
00067         android:contentDescription="@string/content_description_card_delete"
00068         app:layout_constraintBottom_toBottomOf="@id/copyButton"
00069         app:layout_constraintEnd_toEndOf="parent"
00070         app:layout_constraintStart_toEndOf="@id/copyButton"
00071         app:layout_constraintTop_toTopOf="@id/copyButton"
00072         app:srcCompat="@drawable/icon_delete" />
00073 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.86 popup_add_card.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     android:layout_width="match_parent"
00006     android:layout_height="match_parent"
00007     android:fitsSystemWindows="true">
00008
00009     <androidx.constraintlayout.widget.ConstraintLayout
00010         android:id="@+id/card_layout"
00011         android:layout_width="0dp"
00012         android:layout_height="wrap_content"
00013         android:layout_marginHorizontal="@dimen/border_margin"
00014         android:layout_marginTop="@dimen/border_margin"
00015         android:background="@drawable/rectangle"
00016         app:layout_constraintBottom_toTopOf="@id/button_layout"
00017         app:layout_constraintEnd_toEndOf="@id/button_layout"
00018         app:layout_constraintHeight_max="@dimen/popUp_add_card_max_card_height"
00019         app:layout_constraintStart_toStartOf="@id/button_layout"
00020         app:layout_constraintTop_toTopOf="parent">
00021
00022         <TextView
00023             android:id="@+id/popup_add_card_epic_picker_label"
00024             android:layout_width="0dp"
00025             android:layout_height="wrap_content"
00026             android:layout_marginBottom="@dimen/popUp_add_card_label_marginBottom"
00027             android:gravity="center"
00028             android:labelFor="@id/popup_add_card_epic_picker"
00029             android:text="@string/popUp_add_card_label_epic"
00030             android:textColor="@attr/card_count_text_color"
00031             android:textSize="@dimen/popUp_add_card_label_textSize"
00032             app:layout_constraintBottom_toBottomOf="parent"
00033             app:layout_constraintEnd_toEndOf="@id/popup_add_card_epic_picker"
00034             app:layout_constraintStart_toStartOf="@id/popup_add_card_epic_picker"
00035             app:layout_constraintTop_toBottomOf="@id/popup_add_card_epic_picker" />
00036
00037         <NumberPicker
00038             android:id="@+id/popup_add_card_epic_picker"
00039             android:layout_width="0dp"
00040             android:layout_height="wrap_content"
00041             android:layout_marginHorizontal="@dimen/border_margin"
00042             android:layout_marginTop="@dimen/border_margin"
00043             android:descendantFocusability="blocksDescendants"
00044             app:layout_constraintBottom_toTopOf="@id/popup_add_card_epic_picker_label"
00045             app:layout_constraintEnd_toStartOf="@id/popup_add_card_dmg_picker"
00046             app:layout_constraintStart_toStartOf="parent"
00047             app:layout_constraintTop_toTopOf="parent" />
00048
00049         <TextView
00050             android:id="@+id/popup_add_card_dmg_picker_label"
00051             android:layout_width="0dp"
00052             android:layout_height="wrap_content"
00053             android:layout_marginBottom="@dimen/popUp_add_card_label_marginBottom"
00054             android:gravity="center"
00055             android:labelFor="@id/popup_add_card_dmg_picker"
00056             android:text="@string/popUp_add_card_label_dmg"
00057             android:textColor="@attr/card_count_text_color"
00058             android:textSize="@dimen/popUp_add_card_label_textSize"
00059             app:layout_constraintBottom_toBottomOf="parent"
00060             app:layout_constraintEnd_toEndOf="@id/popup_add_card_dmg_picker"
00061             app:layout_constraintStart_toStartOf="@id/popup_add_card_dmg_picker"
00062             app:layout_constraintTop_toBottomOf="@id/popup_add_card_dmg_picker" />

```



```
00063     <NumberPicker
00064         android:id="@+id/popup_add_card_dmg_picker"
00065         android:layout_width="0dp"
00066         android:layout_height="wrap_content"
00067         android:layout_marginHorizontal="@dimen/border_margin"
00068         android:layout_marginTop="@dimen/border_margin"
00069         android:descendantFocusability="blocksDescendants"
00070         app:layout_constraintBottom_toTopOf="@id/popup_add_card_dmg_picker_label"
00071         app:layout_constraintEnd_toStartOf="@id/popup_add_card_ability_picker"
00072         app:layout_constraintStart_toEndOf="@id/popup_add_card_epic_picker"
00073         app:layout_constraintTop_toTopOf="parent" />
00074
00075     <TextView
00076         android:id="@+id/popup_add_card_ability_picker_label"
00077         android:layout_width="0dp"
00078         android:layout_height="wrap_content"
00079         android:layout_marginBottom="@dimen/popUp_add_card_label_marginBottom"
00080         android:gravity="center"
00081         android:labelFor="@id/popup_add_card_ability_picker"
00082         android:text="@string/popUp_add_card_label_ability"
00083         android:textColor="?attr/card_count_text_color"
00084         android:textSize="@dimen/popUp_add_card_label_textSize"
00085         app:layout_constraintBottom_toBottomOf="parent"
00086         app:layout_constraintEnd_toEndOf="@id/popup_add_card_ability_picker"
00087         app:layout_constraintStart_toStartOf="@id/popup_add_card_ability_picker"
00088         app:layout_constraintTop_toBottomOf="@id/popup_add_card_ability_picker" />
00089
00090     <NumberPicker
00091         android:id="@+id/popup_add_card_ability_picker"
00092         android:layout_width="0dp"
00093         android:layout_height="wrap_content"
00094         android:layout_marginHorizontal="@dimen/border_margin"
00095         android:layout_marginTop="@dimen/border_margin"
00096         android:descendantFocusability="blocksDescendants"
00097         app:layout_constraintBottom_toTopOf="@id/popup_add_card_ability_picker_label"
00098         app:layout_constraintEnd_toStartOf="@id/popup_add_card_binding_picker"
00099         app:layout_constraintStart_toEndOf="@id/popup_add_card_dmg_picker"
00100         app:layout_constraintTop_toTopOf="parent" />
00101
00102     <TextView
00103         android:id="@+id/popup_add_card_binding_picker_label"
00104         android:layout_width="0dp"
00105         android:layout_height="wrap_content"
00106         android:layout_marginBottom="@dimen/popUp_add_card_label_marginBottom"
00107         android:gravity="center"
00108         android:labelFor="@id/popup_add_card_binding_picker"
00109         android:text="@string/popUp_add_card_label_binding"
00110         android:textColor="?attr/card_count_text_color"
00111         android:textSize="@dimen/popUp_add_card_label_textSize"
00112         app:layout_constraintBottom_toBottomOf="parent"
00113         app:layout_constraintEnd_toEndOf="@id/popup_add_card_binding_picker"
00114         app:layout_constraintStart_toStartOf="@id/popup_add_card_binding_picker"
00115         app:layout_constraintTop_toBottomOf="@id/popup_add_card_binding_picker" />
00116
00117     <NumberPicker
00118         android:id="@+id/popup_add_card_binding_picker"
00119         android:layout_width="0dp"
00120         android:layout_height="wrap_content"
00121         android:layout_marginHorizontal="@dimen/border_margin"
00122         android:layout_marginTop="@dimen/border_margin"
00123         android:descendantFocusability="blocksDescendants"
00124         android:visibility="gone"
00125         app:layout_constraintBottom_toTopOf="@id/popup_add_card_binding_picker_label"
00126         app:layout_constraintEnd_toStartOf="@id/popup_add_card_number_picker"
00127         app:layout_constraintStart_toEndOf="@id/popup_add_card_ability_picker"
00128         app:layout_constraintTop_toTopOf="parent" />
00129
00130     <TextView
00131         android:id="@+id/popup_add_card_number_picker_label"
00132         android:layout_width="0dp"
00133         android:layout_height="wrap_content"
00134         android:layout_marginBottom="@dimen/popUp_add_card_label_marginBottom"
00135         android:gravity="center"
00136         android:labelFor="@id/popup_add_card_number_picker"
00137         android:text="@string/popUp_add_card_label_number"
00138         android:textColor="?attr/card_count_text_color"
00139         android:textSize="@dimen/popUp_add_card_label_textSize"
00140         app:layout_constraintBottom_toBottomOf="parent"
00141         app:layout_constraintEnd_toEndOf="@id/popup_add_card_number_picker"
00142         app:layout_constraintStart_toStartOf="@id/popup_add_card_number_picker"
00143         app:layout_constraintTop_toBottomOf="@id/popup_add_card_number_picker" />
00144
00145     <NumberPicker
00146         android:id="@+id/popup_add_card_number_picker"
00147         android:layout_width="0dp"
00148         android:layout_height="wrap_content"
00149         android:layout_marginHorizontal="@dimen/border_margin"
```

```

00150         android:layout_marginTop="@dimen/border_margin"
00151         android:descendantFocusability="blocksDescendants"
00152         app:layout_constraintBottom_toTopOf="@id/popup_add_card_number_picker_label"
00153         app:layout_constraintEnd_toEndOf="parent"
00154         app:layout_constraintStart_toEndOf="@id/popup_add_card_binding_picker"
00155         app:layout_constraintTop_toTopOf="parent" />
00156     </androidx.constraintlayout.widget.ConstraintLayout>
00157
00158     <androidx.constraintlayout.widget.ConstraintLayout
00159         android:id="@+id/button_layout"
00160         android:layout_width="wrap_content"
00161         android:layout_height="0dp"
00162         android:layout_margin="@dimen/border_margin"
00163         app:layout_constraintBottom_toBottomOf="parent"
00164         app:layout_constraintEnd_toEndOf="parent"
00165         app:layout_constraintHeight_max="@dimen/popUp_card_button_height"
00166         app:layout_constraintStart_toStartOf="parent"
00167         app:layout_constraintTop_toBottomOf="@id/card_layout">
00168
00169         <Button
00170             android:id="@+id/popup_add_card_cancel_button"
00171             android:layout_width="@dimen/popUp_card_button_width"
00172             android:layout_height="match_parent"
00173             android:layout_marginEnd="@dimen/popUp_card_button_margin"
00174             android:background="@drawable/rectangle_cancel_popup"
00175             android:text="@string/alertDialog_cancel"
00176             android:textAllCaps="false"
00177             android:textColor="@color/color_popup_cancel_text"
00178             android:textSize="@dimen/popUp_card_textSize"
00179             app:layout_constraintEnd_toStartOf="@id/popup_add_card_save_button"
00180             app:layout_constraintHeight_max="@dimen/popUp_card_button_height"
00181             app:layout_constraintStart_toStartOf="parent" />
00182
00183         <Button
00184             android:id="@+id/popup_add_card_save_button"
00185             android:layout_width="@dimen/popUp_card_button_width"
00186             android:layout_height="match_parent"
00187             android:layout_marginStart="@dimen/popUp_card_button_margin"
00188             android:background="@drawable/rectangle"
00189             android:text="@string/popUp_cards_save"
00190             android:textAllCaps="false"
00191             android:textColor="?attr/card_count_text_color"
00192             android:textSize="@dimen/popUp_card_textSize"
00193             app:layout_constraintEnd_toEndOf="parent"
00194             app:layout_constraintHeight_max="@dimen/popUp_card_button_height"
00195             app:layout_constraintStart_toEndOf="@id/popup_add_card_cancel_button" />
00196     </androidx.constraintlayout.widget.ConstraintLayout>
00197 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.87 popup_cards.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     android:layout_width="match_parent"
00006     android:layout_height="match_parent"
00007     android:fitsSystemWindows="true"
00008     app:layout_constraintBottom_toBottomOf="parent"
00009     app:layout_constraintTop_toTopOf="parent">
00010
00011     <androidx.recyclerview.widget.RecyclerView
00012         android:id="@+id/cards_list"
00013         android:layout_width="0dp"
00014         android:layout_height="@dimen/popUp_card_list_height"
00015         android:layout_margin="@dimen/activity_horizontal_margin"
00016         app:layout_constraintBottom_toTopOf="@id/button_layout"
00017         app:layout_constraintEnd_toEndOf="parent"
00018         app:layout_constraintStart_toStartOf="parent"
00019         app:layout_constraintTop_toTopOf="parent" />
00020
00021     <androidx.constraintlayout.widget.ConstraintLayout
00022         android:id="@+id/button_layout"
00023         android:layout_width="wrap_content"
00024         android:layout_height="0dp"
00025         android:layout_margin="@dimen/border_margin"
00026         app:layout_constraintBottom_toBottomOf="parent"
00027         app:layout_constraintEnd_toEndOf="parent"
00028         app:layout_constraintHeight_max="@dimen/popUp_card_button_height"
00029         app:layout_constraintStart_toStartOf="parent"
00030         app:layout_constraintTop_toBottomOf="@id/cards_list">
00031
00032         <Button
00033             android:id="@+id/popup_cards_cancel_button"
00034             android:layout_width="@dimen/popUp_card_button_width"

```



```

00034         android:layout_height="match_parent"
00035         android:layout_marginEnd="@dimen/popUp_card_button_margin"
00036         android:background="@drawable/rectangle_cancel_popup"
00037         android:text="@string/alertDialog_cancel"
00038         android:textAllCaps="false"
00039         android:textColor="@color/color_popup_cancel_text"
00040         android:textSize="@dimen/popUp_card_textSize"
00041         app:layout_constraintEnd_toStartOf="@id/popup_cards_add_button"
00042         app:layout_constraintStart_toStartOf="parent" />
00043
00044     <Button
00045         android:id="@+id/popup_cards_add_button"
00046         android:layout_width="@dimen/popUp_card_button_width"
00047         android:layout_height="match_parent"
00048         android:layout_marginStart="@dimen/popUp_card_button_margin"
00049         android:background="@drawable/rectangle"
00050         android:text="@string/popUp_cards_add"
00051         android:textAllCaps="false"
00052         android:textColor="?attr/card_count_text_color"
00053         android:textSize="@dimen/popUp_card_textSize"
00054         app:layout_constraintEnd_toEndOf="parent"
00055         app:layout_constraintStart_toEndOf="@id/popup_cards_cancel_button" />
00056 </androidx.constraintlayout.widget.ConstraintLayout>
00057 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.88 popup_coin_normal.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     android:id="@+id/coinflipBackground"
00006     android:layout_width="match_parent"
00007     android:layout_height="match_parent"
00008     android:fitsSystemWindows="true">
00009
00010     <LinearLayout
00011         android:layout_width="wrap_content"
00012         android:layout_height="wrap_content"
00013         android:background="@drawable/rectangle_black_transparent"
00014         android:orientation="vertical"
00015         app:layout_constraintBottom_toBottomOf="parent"
00016         app:layout_constraintEnd_toEndOf="parent"
00017         app:layout_constraintStart_toStartOf="parent"
00018         app:layout_constraintTop_toTopOf="parent">
00019
00020         <ImageView
00021             android:id="@+id/popup_coin_normal_coinView"
00022             android:layout_width="@dimen/popUp_coin_diameter"
00023             android:layout_height="@dimen/popUp_coin_diameter"
00024             android:layout_gravity="center"
00025             android:layout_margin="@dimen/activity_horizontal_margin"
00026             android:contentDescription="@string/content_description_coinpopup_coin"
00027             android:src="@drawable/coin_win" />
00028
00029         <TextView
00030             android:id="@+id/popup_coin_normal_textView"
00031             android:layout_width="wrap_content"
00032             android:layout_height="wrap_content"
00033             android:layout_gravity="center"
00034             android:layout_margin="@dimen/activity_horizontal_margin"
00035             android:text="@string/popUp_coin_normal_win"
00036             android:textColor="@color/color_coin_text"
00037             android:textSize="@dimen/popUp_coin_title_textSize" />
00038     </LinearLayout>
00039 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.89 popup_coin_stewie.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     android:id="@+id/coinflipBackground"
00006     android:layout_width="match_parent"
00007     android:layout_height="match_parent"
00008     android:fitsSystemWindows="true">
00009
00010     <androidx.constraintlayout.widget.ConstraintLayout
00011         android:layout_width="wrap_content"
00012         android:layout_height="0dp"
00013         android:layout_margin="@dimen/activity_vertical_margin"

```

```

00013     android:background="@drawable/rectangle_black_transparent"
00014     app:layout_constraintBottom_toBottomOf="parent"
00015     app:layout_constraintEnd_toEndOf="parent"
00016     app:layout_constraintHeight_max="@dimen/popUp_coin_window_max_height"
00017     app:layout_constraintStart_toStartOf="parent"
00018     app:layout_constraintTop_toTopOf="parent">
00019
00020     <LinearLayout
00021         android:id="@+id/imageLayout"
00022         android:layout_width="wrap_content"
00023         android:layout_height="match_parent"
00024         android:layout_margin="@dimen/activity_horizontal_margin"
00025         android:orientation="vertical"
00026         app:layout_constraintEnd_toStartOf="@id/textLayout"
00027         app:layout_constraintStart_toStartOf="parent">
00028
00029         <ImageView
00030             android:layout_width="wrap_content"
00031             android:layout_height="0dp"
00032             android:layout_weight="1"
00033             android:contentDescription="@string/content_description_coinpopup_stewie_1"
00034             android:src="@drawable/stewie_1" />
00035
00036         <ImageView
00037             android:layout_width="wrap_content"
00038             android:layout_height="0dp"
00039             android:layout_weight="1"
00040             android:contentDescription="@string/content_description_coinpopup_stewie_2"
00041             android:src="@drawable/stewie_2" />
00042
00043         <ImageView
00044             android:layout_width="wrap_content"
00045             android:layout_height="0dp"
00046             android:layout_weight="1"
00047             android:contentDescription="@string/content_description_coinpopup_stewie_3"
00048             android:src="@drawable/stewie_3" />
00049     </LinearLayout>
00050
00051     <androidx.constraintlayout.widget.ConstraintLayout
00052         android:id="@+id/textLayout"
00053         android:layout_width="wrap_content"
00054         android:layout_height="match_parent"
00055         android:layout_margin="@dimen/activity_horizontal_margin"
00056         app:layout_constraintEnd_toEndOf="parent"
00057         app:layout_constraintStart_toStartOf="@id/imageLayout"
00058         app:layout_constraintWidth_max="@dimen/popUp_coin_textview_max_width">
00059
00060         <TextView
00061             android:id="@+id/text1"
00062             android:layout_width="match_parent"
00063             android:layout_height="wrap_content"
00064             android:layout_margin="@dimen/border_margin"
00065             android:gravity="center"
00066             android:text="@string/popUp_coin_stewie_1"
00067             android:textColor="@color/color_coin_text"
00068             android:textSize="@dimen/popUp_coin_title_textSize"
00069             app:layout_constraintTop_toTopOf="parent" />
00070
00071         <TextView
00072             android:id="@+id/text2"
00073             android:layout_width="match_parent"
00074             android:layout_height="wrap_content"
00075             android:layout_margin="@dimen/border_margin"
00076             android:gravity="center"
00077             android:text="@string/popUp_coin_stewie_2"
00078             android:textColor="@color/color_coin_text"
00079             android:textSize="@dimen/popUp_coin_dialog_textSize"
00080             app:layout_constraintTop_toBottomOf="@id/text1" />
00081
00082         <TextView
00083             android:id="@+id/text3"
00084             android:layout_width="match_parent"
00085             android:layout_height="wrap_content"
00086             android:layout_margin="@dimen/border_margin"
00087             android:gravity="center"
00088             android:text="@string/popUp_coin_stewie_3"
00089             android:textColor="@color/color_coin_text"
00090             android:textSize="@dimen/popUp_coin_dialog_textSize"
00091             app:layout_constraintBottom_toTopOf="@id/text4"
00092             app:layout_constraintTop_toBottomOf="@id/text2" />
00093
00094         <TextView
00095             android:id="@+id/text4"
00096             android:layout_width="match_parent"
00097             android:layout_height="wrap_content"
00098             android:layout_margin="@dimen/border_margin"
00099             android:gravity="center"

```

```

00100         android:text="@string/popUp_coin_stewie_4"
00101         android:textColor="@color/color_coin_text"
00102         android:textSize="@dimen/popUp_coin_dialog_textSize"
00103         app:layout_constraintBottom_toBottomOf="parent"
00104         app:layout_constraintTop_toBottomOf="@id/text3" />
00105     </androidx.constraintlayout.widget.ConstraintLayout>
00106 </androidx.constraintlayout.widget.ConstraintLayout>
00107 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.90 popup_coin_terry.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     android:id="@+id/coinflipBackground"
00006     android:layout_width="match_parent"
00007     android:layout_height="match_parent"
00008     android:fitsSystemWindows="true">
00009     <androidx.constraintlayout.widget.ConstraintLayout
00010         android:layout_width="wrap_content"
00011         android:layout_height="0dp"
00012         android:layout_margin="@dimen/activity_vertical_margin"
00013         android:background="@drawable/rectangle_black_transparent"
00014         app:layout_constraintBottom_toBottomOf="parent"
00015         app:layout_constraintEnd_toEndOf="parent"
00016         app:layout_constraintHeight_max="@dimen/popUp_coin_window_max_height"
00017         app:layout_constraintStart_toStartOf="parent"
00018         app:layout_constraintTop_toTopOf="parent">
00019         <LinearLayout
00020             android:id="@+id/imageLayout"
00021             android:layout_width="wrap_content"
00022             android:layout_height="match_parent"
00023             android:layout_margin="@dimen/activity_horizontal_margin"
00024             android:orientation="vertical"
00025             app:layout_constraintEnd_toStartOf="@id/textLayout"
00026             app:layout_constraintStart_toStartOf="parent">
00027             <ImageView
00028                 android:layout_width="wrap_content"
00029                 android:layout_height="0dp"
00030                 android:layout_weight="1"
00031                 android:contentDescription="@string/content_description_coinpopup_terry_1"
00032                 android:src="@drawable/terry_1" />
00033             <ImageView
00034                 android:layout_width="wrap_content"
00035                 android:layout_height="0dp"
00036                 android:layout_weight="1"
00037                 android:contentDescription="@string/content_description_coinpopup_terry_2"
00038                 android:src="@drawable/terry_2" />
00039             <ImageView
00040                 android:layout_width="wrap_content"
00041                 android:layout_height="0dp"
00042                 android:layout_weight="1"
00043                 android:contentDescription="@string/content_description_coinpopup_terry_3"
00044                 android:src="@drawable/terry_3" />
00045             </LinearLayout>
00046         <androidx.constraintlayout.widget.ConstraintLayout
00047             android:id="@+id/textLayout"
00048             android:layout_width="wrap_content"
00049             android:layout_height="match_parent"
00050             android:layout_margin="@dimen/activity_horizontal_margin"
00051             app:layout_constraintEnd_toEndOf="parent"
00052             app:layout_constraintStart_toEndOf="@id/imageLayout"
00053             app:layout_constraintWidth_max="@dimen/popUp_coin_textview_max_width">
00054             <TextView
00055                 android:id="@+id/text1"
00056                 android:layout_width="match_parent"
00057                 android:layout_height="wrap_content"
00058                 android:layout_margin="@dimen/border_margin"
00059                 android:gravity="center"
00060                 android:text="@string/popUp_coin_terry_1"
00061                 android:textColor="@color/color_coin_text"
00062                 android:textSize="@dimen/popUp_coin_title_textSize"
00063                 app:layout_constraintBottom_toTopOf="@id/text2"
00064                 app:layout_constraintTop_toTopOf="parent" />
00065             <TextView
00066                 android:id="@+id/text2"

```

```

00074         android:layout_width="match_parent"
00075         android:layout_height="wrap_content"
00076         android:layout_margin="@dimen/border_margin"
00077         android:gravity="center"
00078         android:text="@string/popUp_coin_tery_2"
00079         android:textColor="@color/color_coin_text"
00080         android:textSize="@dimen/popUp_coin_dialog_textSize"
00081         app:layout_constraintBottom_toTopOf="@id/text3"
00082         app:layout_constraintTop_toBottomOf="@id/text1" />
00083
00084     <TextView
00085         android:id="@+id/text3"
00086         android:layout_width="match_parent"
00087         android:layout_height="wrap_content"
00088         android:layout_margin="@dimen/border_margin"
00089         android:gravity="center"
00090         android:text="@string/popUp_coin_tery_3"
00091         android:textColor="@color/color_coin_text"
00092         android:textSize="@dimen/popUp_coin_dialog_textSize"
00093         app:layout_constraintBottom_toBottomOf="parent"
00094         app:layout_constraintTop_toBottomOf="@id/text2" />
00095     </androidx.constraintlayout.widget.ConstraintLayout>
00096 </androidx.constraintlayout.widget.ConstraintLayout>
00097 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.91 popup_coin_vin.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     android:id="@+id/coinflipBackground"
00006     android:layout_width="match_parent"
00007     android:layout_height="match_parent"
00008     android:fitsSystemWindows="true">
00009
00010     <androidx.constraintlayout.widget.ConstraintLayout
00011         android:layout_width="wrap_content"
00012         android:layout_height="0dp"
00013         android:layout_margin="@dimen/activity_vertical_margin"
00014         android:background="@drawable/rectangle_black_transparent"
00015         app:layout_constraintBottom_toBottomOf="parent"
00016         app:layout_constraintEnd_toEndOf="parent"
00017         app:layout_constraintHeight_max="@dimen/popUp_coin_window_max_height"
00018         app:layout_constraintStart_toStartOf="parent"
00019         app:layout_constraintTop_toTopOf="parent">
00020
00021         <ImageView
00022             android:id="@+id/imageView"
00023             android:layout_width="wrap_content"
00024             android:layout_height="match_parent"
00025             android:layout_margin="@dimen/activity_vertical_margin"
00026             android:contentDescription="@string/content_description_coinpopup_vin"
00027             android:src="@drawable/vin"
00028             app:layout_constraintEnd_toStartOf="@id/textView"
00029             app:layout_constraintStart_toStartOf="parent" />
00030
00031         <TextView
00032             android:id="@+id/textView"
00033             android:layout_width="wrap_content"
00034             android:layout_height="match_parent"
00035             android:layout_margin="@dimen/activity_horizontal_margin"
00036             android:gravity="center"
00037             android:text="@string/popUp_coin_vin"
00038             android:textColor="@color/color_coin_text"
00039             android:textSize="@dimen/popUp_coin_title_textSize"
00040             app:layout_constraintEnd_toEndOf="parent"
00041             app:layout_constraintStart_toEndOf="@id/imageView"
00042             app:layout_constraintWidth_max="@dimen/popUp_coin_textview_max_width" />
00043     </androidx.constraintlayout.widget.ConstraintLayout>

```

6.92 popup_faction.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     android:id="@+id/factionBackground"
00006     android:layout_width="match_parent"
00007     android:layout_height="match_parent"
00008     android:fitsSystemWindows="true">

```

```
00008
00009 <ImageView
00010     android:id="@+id/monsterCardView"
00011     android:layout_width="0dp"
00012     android:layout_height="wrap_content"
00013     android:layout_margin="@dimen/border_margin"
00014     android:contentDescription="@string/content_description_factionpopup_monster"
00015     android:src="@drawable/card_monster"
00016     app:layout_constraintTop_toTopOf="parent"
00017     app:layout_constraintBottom_toBottomOf="parent"
00018     app:layout_constraintEnd_toEndOf="@id/monsterButton"
00019     app:layout_constraintStart_toStartOf="@id/monsterButton" />
00020
00021 <Button
00022     android:id="@+id/monsterButton"
00023     android:layout_width="0dp"
00024     android:layout_height="wrap_content"
00025     android:layout_margin="@dimen/popUp_faction_margin"
00026     android:background="@drawable/rectangle"
00027     android:text="@string/faction_monster"
00028     android:textAllCaps="false"
00029     android:textColor="@color/color_text_monster"
00030     android:textSize="@dimen/rectangle_textSize"
00031     android:theme="@style/MonsterTheme"
00032     app:layout_constraintBottom_toBottomOf="parent"
00033     app:layout_constraintEnd_toStartOf="@id/niflgaardButton"
00034     app:layout_constraintHeight_min="@dimen/rectangle_height"
00035     app:layout_constraintStart_toStartOf="parent"
00036     app:layout_constraintTop_toBottomOf="@+id/monsterCardView"
00037     app:layout_constraintWidth_max="@dimen/popUp_faction_button_max_width" />
00038
00039 <ImageView
00040     android:id="@+id/niflgaardCardView"
00041     android:layout_width="0dp"
00042     android:layout_height="wrap_content"
00043     android:layout_margin="@dimen/border_margin"
00044     android:contentDescription="@string/content_description_factionpopup_niflgaard"
00045     android:src="@drawable/card_niflgaard"
00046     app:layout_constraintBottom_toBottomOf="parent"
00047     app:layout_constraintTop_toTopOf="parent"
00048     app:layout_constraintEnd_toEndOf="@id/niflgaardButton"
00049     app:layout_constraintStart_toStartOf="@id/niflgaardButton" />
00050
00051 <Button
00052     android:id="@+id/niflgaardButton"
00053     android:layout_width="0dp"
00054     android:layout_height="wrap_content"
00055     android:layout_margin="@dimen/popUp_faction_margin"
00056     android:background="@drawable/rectangle"
00057     android:text="@string/faction_niflgaard"
00058     android:textAllCaps="false"
00059     android:textColor="@color/color_text_niflgaard"
00060     android:textSize="@dimen/rectangle_textSize"
00061     android:theme="@style/NiflgaardTheme"
00062     app:layout_constraintBottom_toBottomOf="parent"
00063     app:layout_constraintEnd_toStartOf="@id/northernKingdomsButton"
00064     app:layout_constraintHeight_min="@dimen/rectangle_height"
00065     app:layout_constraintStart_toEndOf="@id/monsterButton"
00066     app:layout_constraintTop_toBottomOf="@id/niflgaardCardView"
00067     app:layout_constraintWidth_max="@dimen/popUp_faction_button_max_width" />
00068
00069 <ImageView
00070     android:id="@+id/northernKingdomsCardView"
00071     android:layout_width="0dp"
00072     android:layout_height="wrap_content"
00073     android:layout_margin="@dimen/border_margin"
00074     android:contentDescription="@string/content_description_factionpopup_northern_kingdoms"
00075     android:src="@drawable/card_northern_kingdoms"
00076     app:layout_constraintTop_toTopOf="parent"
00077     app:layout_constraintBottom_toBottomOf="parent"
00078     app:layout_constraintEnd_toEndOf="@id/northernKingdomsButton"
00079     app:layout_constraintStart_toStartOf="@id/northernKingdomsButton" />
00080
00081 <Button
00082     android:id="@+id/northernKingdomsButton"
00083     android:layout_width="0dp"
00084     android:layout_height="wrap_content"
00085     android:layout_margin="@dimen/popUp_faction_margin"
00086     android:background="@drawable/rectangle"
00087     android:text="@string/faction_northern_kingdoms"
00088     android:textAllCaps="false"
00089     android:textColor="@color/color_text_northern_kingdoms"
00090     android:textSize="@dimen/rectangle_textSize"
00091     android:theme="@style/NorthernKingdomsTheme"
00092     app:layout_constraintBottom_toBottomOf="parent"
00093     app:layout_constraintEnd_toStartOf="@id/scoiataelButton"
00094     app:layout_constraintHeight_min="@dimen/rectangle_height"
```

```

00095     app:layout_constraintStart_toEndOf="@id/nilfgaardButton"
00096     app:layout_constraintTop_toBottomOf="@id/northernKingdomsCardView"
00097     app:layout_constraintWidth_max="@dimen/popUp_faction_button_max_width" />
00098
00099     <ImageView
00100         android:id="@+id/scoiataelCardView"
00101         android:layout_width="0dp"
00102         android:layout_height="wrap_content"
00103         android:layout_margin="@dimen/border_margin"
00104         android:contentDescription="@string/content_description_factionpopup_scoiatael"
00105         android:src="@drawable/card_scoiatael"
00106         app:layout_constraintTop_toTopOf="parent"
00107         app:layout_constraintBottom_toBottomOf="parent"
00108         app:layout_constraintEnd_toEndOf="@id/scoiataelButton"
00109         app:layout_constraintStart_toStartOf="@id/scoiataelButton" />
00110
00111     <Button
00112         android:id="@+id/scoiataelButton"
00113         android:layout_width="0dp"
00114         android:layout_height="wrap_content"
00115         android:layout_margin="@dimen/popUp_faction_margin"
00116         android:background="@drawable/rectangle"
00117         android:text="@string/faction_scoiatael"
00118         android:textAllCaps="false"
00119         android:textColor="@color/color_text_scoiatael"
00120         android:textSize="@dimen/rectangle_text_size"
00121         android:theme="@style/ScoiataelTheme"
00122         app:layout_constraintBottom_toBottomOf="parent"
00123         app:layout_constraintEnd_toEndOf="parent"
00124         app:layout_constraintHeight_min="@dimen/rectangle_height"
00125         app:layout_constraintStart_toEndOf="@id/northernKingdomsButton"
00126         app:layout_constraintTop_toBottomOf="@id/scoiataelCardView"
00127         app:layout_constraintWidth_max="@dimen/popUp_faction_button_max_width" />
00128 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.93 row.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <androidx.constraintlayout.widget.ConstraintLayout
00003     xmlns:android="http://schemas.android.com/apk/res/android"
00004     xmlns:app="http://schemas.android.com/apk/res-auto"
00005     android:layout_width="wrap_content"
00006     android:layout_height="wrap_content">
00007
00008     <androidx.constraintlayout.widget.ConstraintLayout
00009         android:layout_width="@dimen/point_circle_diameter"
00010         android:layout_height="@dimen/point_circle_diameter"
00011         android:translationZ="5dp"
00012         app:layout_constraintEnd_toEndOf="parent"
00013         app:layout_constraintStart_toStartOf="parent"
00014         app:layout_constraintTop_toTopOf="parent">
00015
00016         <ImageView
00017             android:id="@+id/pointBall"
00018             android:layout_width="match_parent"
00019             android:layout_height="match_parent"
00020             android:background="@null"
00021             android:contentDescription="@string/content_description_row_pointball"
00022             android:scaleType="fitXY"
00023             android:src="@attr/point_ball_mipmap"
00024             android:translationZ="5dp" />
00025
00026         <TextView
00027             android:id="@+id/pointView"
00028             android:layout_width="match_parent"
00029             android:layout_height="match_parent"
00030             android:gravity="center"
00031             android:textColor="@color/circle_text"
00032             android:textSize="@dimen/point_circle_text_size"
00033             android:translationZ="10dp" />
00034     </androidx.constraintlayout.widget.ConstraintLayout>
00035
00036     <ImageView
00037         android:id="@+id/weatherView"
00038         android:layout_width="0dp"
00039         android:layout_height="0dp"
00040         android:layout_marginTop="@dimen/point_circle_diameter_div2"
00041         android:contentDescription="@string/content_description_row_weather"
00042         android:scaleType="centerCrop"
00043         app:layout_constraintBottom_toTopOf="@id/hornView"
00044         app:layout_constraintDimensionRatio="1048:674"
00045         app:layout_constraintEnd_toEndOf="@id/cardView"
00046         app:layout_constraintStart_toStartOf="@id/cardView"
00047         app:layout_constraintTop_toTopOf="parent"
00048         app:srcCompat="@drawable/good_weather" />

```

```

00048
00049     <ImageView
00050         android:id="@+id/hornView"
00051         android:layout_width="0dp"
00052         android:layout_height="0dp"
00053         android:contentDescription="@string/content_description_row_horn"
00054         android:scaleType="centerCrop"
00055         app:layout_constraintBottom_toTopOf="@id/cardView"
00056         app:layout_constraintDimensionRatio="1048:674"
00057         app:layout_constraintEnd_toEndOf="@id/cardView"
00058         app:layout_constraintStart_toStartOf="@id/cardView"
00059         app:layout_constraintTop_toBottomOf="@id/weatherView"
00060         app:srcCompat="@drawable/horn_grey" />
00061
00062     <androidx.constraintlayout.widget.ConstraintLayout
00063         android:id="@+id/cardView"
00064         android:layout_width="wrap_content"
00065         android:layout_height="0dp"
00066         app:layout_constraintBottom_toBottomOf="parent"
00067         app:layout_constraintEnd_toEndOf="parent"
00068         app:layout_constraintStart_toStartOf="parent"
00069         app:layout_constraintTop_toBottomOf="@id/hornView">
00070
00071         <ImageView
00072             android:id="@+id/cardsImage"
00073             android:layout_width="match_parent"
00074             android:layout_height="match_parent"
00075             android:contentDescription="@string/content_description_row_card"
00076             android:src="@attr/card_view_mipmap"
00077             android:translationZ="5dp" />
00078
00079         <TextView
00080             android:id="@+id/cardCountView"
00081             android:layout_width="wrap_content"
00082             android:layout_height="wrap_content"
00083             android:textColor="@attr/card_count_text_color"
00084             android:textSize="@dimen/card_count_textSize"
00085             android:translationZ="10dp"
00086             app:layout_constraintBottom_toBottomOf="parent"
00087             app:layout_constraintEnd_toEndOf="parent"
00088             app:layout_constraintStart_toStartOf="parent"
00089             app:layout_constraintTop_toTopOf="parent" />
00090     </androidx.constraintlayout.widget.ConstraintLayout>
00091 </androidx.constraintlayout.widget.ConstraintLayout>

```

6.94 icon_launcher.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">
00003     <background android:drawable="@color/icon_launcher_background" />
00004     <foreground android:drawable="@mipmap/icon_launcher_foreground" />
00005     <monochrome android:drawable="@mipmap/icon_launcher_monochrome" />
00006 </adaptive-icon>

```

6.95 icon_launcher_round.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">
00003     <background android:drawable="@color/icon_launcher_background" />
00004     <foreground android:drawable="@mipmap/icon_launcher_foreground" />
00005     <monochrome android:drawable="@mipmap/icon_launcher_monochrome" />
00006 </adaptive-icon>

```

6.96 arrays.xml

```

00001 <resources>
00002     <string-array name="design_names">
00003         <item>@string/array_design_default</item>
00004         <item>@string/array_design_geralt</item>
00005         <item>@string/array_design_ciri</item>
00006         <item>@string/array_design_jaskier</item>
00007         <item>@string/array_design_yennefer</item>
00008         <item>@string/array_design_eredin</item>
00009     </string-array>
00010
00011     <string-array name="design_values">
00012         <item>0</item>
00013         <item>1</item>
00014         <item>2</item>

```



```

00015         <item>3</item>
00016         <item>4</item>
00017         <item>5</item>
00018     </string-array>
00019 </resources>

```

6.97 attr.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <resources>
00003     <attr name="card_count_text_color" format="color" />
00004
00005     <attr name="card_view_mipmap" format="reference" />
00006
00007     <attr name="point_ball_mipmap" format="reference" />
00008
00009     <attr name="card_background_color" format="reference" />
00010
00011     <attr name="card_background_gradient" format="reference" />
00012
00013     <attr name="card_view_button_background" format="reference" />
00014
00015     <declare-styleable name="theme">
00016         <attr name="point_ball_mipmap" />
00017         <attr name="card_view_mipmap" />
00018         <attr name="colorPrimary" />
00019         <attr name="android:alertDialogIcon" />
00020     </declare-styleable>
00021 </resources>

```

6.98 colors.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <resources>
00003     <color name="circle_text">#000000</color>
00004
00005     <color name="color_popup_background">#BF000000</color>
00006     <color name="color_popup_cancel_text">#FFFFFF</color>
00007     <color name="color_coin_text">#FFFFFF</color>
00008     <color name="color_cancel_popup_background">#505050</color>
00009     <color name="color_cancel_popup_background_gradient">#000000</color>
00010     <color name="color_damage_textColor">#2F1700</color>
00011     <color name="color_damage_textColor_buffed">#10A010</color>
00012     <color name="color_damage_textColor_debuffed">#E42020</color>
00013
00014     <color name="color_text_scoiatael">#CAC559</color>
00015     <color name="color_background_scoiatael">#339300</color>
00016     <color name="color_gradient_scoiatael">#164000</color>
00017     <color name="color_accent_scoiatael">#52EC00</color>
00018     <color name="color_card_view_button_background_scoiatael">#40B900</color>
00019
00020     <color name="color_text_monster">#B6D1DB</color>
00021     <color name="color_background_monster">#99030E</color>
00022     <color name="color_gradient_monster">#430008</color>
00023     <color name="color_accent_monster">#FF1F30</color>
00024     <color name="color_card_view_button_background_monster">#C80917</color>
00025
00026     <color name="color_text_northern_kingdoms">#D9D9D9</color>
00027     <color name="color_background_northern_kingdoms">#005ADF</color>
00028     <color name="color_gradient_northern_kingdoms">#002E71</color>
00029     <color name="color_accent_northern_kingdoms">#3285FF</color>
00030     <color name="color_card_view_button_background_northern_kingdoms">#003B92</color>
00031
00032     <color name="color_text_nilfgaard">#EFC561</color>
00033     <color name="color_background_nilfgaard">#424242</color>
00034     <color name="color_gradient_nilfgaard">#0C0C0C</color>
00035     <color name="color_accent_nilfgaard">#CCCCCC</color>
00036     <color name="color_card_view_button_background_nilfgaard">#787878</color>
00037 </resources>

```

6.99 dimens.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <resources>
00003     <dimen name="point_circle_diameter">80dp</dimen>
00004     <dimen name="point_circle_diameter_div2">40dp</dimen>
00005     <dimen name="point_circle_textSize">40sp</dimen>
00006     <dimen name="overall_point_circle_textSize">45sp</dimen>
00007     <dimen name="card_count_textSize">40sp</dimen>

```



```

00008
00009 <dimen name="border_margin">8dp</dimen>
00010 <dimen name="margin_row_buttons">16dp</dimen>
00011 <dimen name="margin_menu_buttons">12dp</dimen>
00012
00013 <dimen name="rectangle_height">65dp</dimen>
00014 <dimen name="rectangle_stroke_width">4dp</dimen>
00015 <dimen name="rectangle_corner_radius">15dp</dimen>
00016 <dimen name="rectangle_textSize">20sp</dimen>
00017
00018 <dimen name="popUp_faction_button_max_width">200dp</dimen>
00019 <dimen name="popUp_faction_margin">24dp</dimen>
00020
00021 <dimen name="popUp_card_list_height">240dp</dimen>
00022 <dimen name="popUp_card_width">160dp</dimen>
00023 <dimen name="popUp_card_textSize">22sp</dimen>
00024 <dimen name="popUp_card_button_width">200dp</dimen>
00025 <dimen name="popUp_card_button_height">80dp</dimen>
00026 <dimen name="popUp_card_image_diameter">50dp</dimen>
00027 <dimen name="popUp_card_button_margin">75dp</dimen>
00028
00029 <dimen name="popUp_add_card_label_textSize">21sp</dimen>
00030 <dimen name="popUp_add_card_label_marginBottom">24dp</dimen>
00031 <dimen name="popUp_add_card_max_card_height">300dp</dimen>
00032
00033 <dimen name="popUp_coin_textview_max_width">275dp</dimen>
00034 <dimen name="popUp_coin_diameter">90dp</dimen>
00035 <dimen name="popUp_coin_window_max_height">400dp</dimen>
00036 <dimen name="popUp_coin_title_textSize">24sp</dimen>
00037 <dimen name="popUp_coin_dialog_textSize">18sp</dimen>
00038
00039 <dimen name="alertDialog_checkBox_marginStart">40dp</dimen>
00040 <dimen name="alertDialog_checkBox_marginEnd">25dp</dimen>
00041 <dimen name="alertDialog_checkBox_minHeight">48dp</dimen>
00042
00043 <dimen name="rules_textSize">20sp</dimen>
00044
00045 <dimen name="divider_strength">2dp</dimen>
00046 <dimen name="indicator_radius">8dp</dimen>
00047
00048 <!-- Default screen margins, per the Android Design guidelines. -->
00049 <dimen name="activity_horizontal_margin">16dp</dimen>
00050 <dimen name="activity_vertical_margin">16dp</dimen>
00051 </resources>

```

6.100 icon_launcher_background.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <resources>
00003 <color name="icon_launcher_background">#FFFFFF</color>
00004 </resources>

```

6.101 preferences.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>
00002 <resources>
00003 <string name="preference_key_design" translatable="false">design</string>
00004 <string name="design_preference_default" translatable="false">0</string>
00005 <bool name="warning_preference_default">true</bool>
00006 <string name="preference_key_warning" translatable="false">warnings</string>
00007 <bool name="faction_reset_preference_default">false</bool>
00008 <string name="preference_key_faction_reset" translatable="false">factionReset</string>
00009 <string name="preference_key_introduction" translatable="false">introduction</string>
00010
00011 <!-- Sound preferences -->
00012 <bool name="sound_preference_default">true</bool>
00013
00014 <string name="preference_key_sounds_all" translatable="false">sound_all</string>
00015 <string name="preference_key_sounds_cards" translatable="false">sound_cards</string>
00016 <string name="preference_key_sounds_reset" translatable="false">sound_reset</string>
00017 <string name="preference_key_sounds_weather" translatable="false">sound_weather</string>
00018 <string name="preference_key_sounds_horn" translatable="false">sound_horn</string>
00019 <string name="preference_key_sounds_burn" translatable="false">sound_burn</string>
00020 <string name="preference_key_sounds_coin" translatable="false">sound_coin</string>
00021 </resources>

```

6.102 strings.xml

```

00001 <?xml version="1.0" encoding="utf-8"?>

```

```

00002 <resources>
00003     <string name="app_name">Gwint</string>
00004     <string name="app_description">Eine App, die es ermöglicht, das Kartenspiel Gwint aus The Witcher
3 mit analogen Karten zu spielen.</string>
00005     <string name="settings_title">Einstellungen</string>
00006
00007     <string name="faction_nilfgaard">Kaiserreich Nilfgaard</string>
00008     <string name="faction_northern_kingdoms">Nördliche Königreiche</string>
00009
00010     <string name="popUp_cards_add">Karte hinzufügen</string>
00011     <string name="popUp_cards_save">Speichern</string>
00012     <string name="popUp_add_card_label_dmg">Stärke</string>
00013     <string name="popUp_add_card_label_ability">Fähigkeit</string>
00014     <string name="popUp_add_card_label_binding">Gruppe</string>
00015     <string name="popUp_add_card_label_number">Anzahl</string>
00016
00017     <string name="popUp_coin_normal_win">Du beginnst.</string>
00018     <string name="popUp_coin_normal_lose">Dein Gegner beginnt.</string>
00019     <string name="popUp_coin_terry_1">Terry Jeffords hat die Münze verbogen!</string>
00020     <string name="popUp_coin_stewie_1">Chris Griffin hat die Münze gegessen!</string>
00021     <string name="popUp_coin_vin">Vin hat Stahl verbrannt und die Münze weggedrückt!</string>
00022
00023     <string name="unit_toString">%1$s%2$s (%3$d) - Fähigkeit: %4$s %5$s</string>
00024     <string name="unit_toString_multiplicity">%1$d× %2$s</string>
00025     <string name="unit_toString_melee">Nahkampf</string>
00026     <string name="unit_toString_range">Fernkampf</string>
00027     <string name="unit_toString_siege">Belagerungs</string>
00028     <string name="unit_toString_epic">held</string>
00029     <string name="unit_toString_unit">einheit</string>
00030     <string name="unit_toString_ability_none">Keine</string>
00031     <string name="unit_toString_squad">(%1$d)</string>
00032     <string name="unit_collection_toString_accumulation_word">%1$s und %2$s</string>
00033
00034     <string name="alertDialog_cancel">Abbrechen</string>
00035
00036     <string name="alertDialog_burn_title">Verbrennen</string>
00037     <string name="alertDialog_burn_positive">Verbrennen</string>
00038     <string name="alertDialog_burn_msg">Möchtest du wirklich %1$s verbrennen?</string>
00039     <string name="alertDialog_burn_negative">Nicht verbrennen</string>
00040
00041     <string name="alertDialog_reset_title">Spielfeld zurücksetzen</string>
00042     <string name="alertDialog_reset_msg_default">Willst du wirklich das gesamte Spielfeld
zurücksetzen?</string>
00043     <string name="alertDialog_reset_msg_faction_switch">Du hast \"Zurücksetzen bei Fraktionswechsel\"
aktiviert. Soll das Spielfeld wirklich zurückgesetzt werden?</string>
00044     <string name="alertDialog_reset_checkbox">Die Fähigkeit der Monsterfraktion auslösen (du behältst
eine zufällige Einheit)</string>
00045     <string name="alertDialog_factionreset_monster_toast_keep">Du behältst: %1$s</string>
00046     <string name="alertDialog_reset_positive">Zurücksetzen</string>
00047     <string name="alertDialog_reset_negative">Nicht zurücksetzen</string>
00048
00049     <string name="alertDialog_revenge_title">Rächerfähigkeit</string>
00050     <string name="alertDialog_revenge_msg">Du hast eine Einheit vom Spielfeld entfernt, die einen
Rächer herbeirufen kann. Möchtest du diese Fähigkeit auslösen?</string>
00051     <string name="alertDialog_revenge_positive">Rächer rufen</string>
00052
00053     <string name="preference_design">Hintergrundbild</string>
00054     <string name="preference_warnings_title">Warnungen</string>
00055     <string name="preference_warnings_desc">Legt fest, ob beim Zurücksetzen eine Warnung erscheinen
soll.</string>
00056     <string name="preference_faction_reset_title">Zurücksetzen bei Fraktionswechsel</string>
00057     <string name="preference_faction_reset_desc">Legt fest, ob das Spielfeld beim Fraktionswechsel
automatisch zurückgesetzt werden soll.</string>
00058
00059     <string name="preference_sounds_header">Tonausgabe</string>
00060     <string name="preference_sounds_category">Ereignisse</string>
00061     <string name="preference_sounds_cards_new">Neue Einheit</string>
00062     <string name="preference_sounds_cards_delete">Einheit entfernt</string>
00063     <string name="preference_sounds_weather">Wetteränderung</string>
00064     <string name="preference_sounds_horn">Horn des Kommandanten</string>
00065     <string name="preference_sounds_burn">Verbrennen</string>
00066     <string name="preference_sounds_coin">Münzwurf</string>
00067
00068     <string name="preference_info_introduction">Einführung</string>
00069
00070     <string name="preference_rules_header">Spielregeln</string>
00071     <string name="preference_rules_general_title">Allgemein</string>
00072     <string name="preference_rules_course_title">Spielablauf</string>
00073     <string name="preference_rules_factions_title">Fraktionen</string>
00074     <string name="preference_rules_commander_title">Anführer</string>
00075     <string name="preference_rules_cards_title">Kartensatz und Spielfeld</string>
00076     <string name="preference_rules_card_abilities_title">Kartenfähigkeiten</string>
00077     <string name="preference_rules_special_cards_title">Spezialkarten</string>
00078
00079     <string name="introduction_button_skip">Überspringen</string>
00080     <string name="introduction_button_finish">Fertig</string>
00081

```

```

00082     <string name="content_description_activity_main_background">Hintergrundbild</string>
00083     <string name="content_description_activity_main_point_ball">Gesamtpunktzahl</string>
00084     <string name="content_description_activity_main_faction_button">Fraktionslogo</string>
00085     <string name="content_description_activity_main_weather_button">Gutes-Wetter-Icon</string>
00086     <string name="content_description_activity_main_burn_button">Verbrennen-Icon</string>
00087     <string name="content_description_activity_main_coin_button">Münzen-Icon</string>
00088     <string name="content_description_activity_main_settings_button">Einstellungs-Icon</string>
00089     <string name="content_description_card_ability">Fähigkeits-Icon</string>
00090     <string name="content_description_card_delete">Löschen-Icon</string>
00091     <string name="content_description_card_copy">Kopieren-Icon</string>
00092     <string name="content_description_coinpopup_coin">Münze</string>
00093     <string name="content_description_coinpopup_stewie_1">Chris Griffin, der betroffen die Arme
hebt</string>
00094     <string name="content_description_coinpopup_stewie_2">Stewie Griffin, der etwas erzählt</string>
00095     <string name="content_description_coinpopup_stewie_3">Stewie Griffin, der etwas schreit</string>
00096     <string name="content_description_coinpopup_terry_1">Eine Münze, die verbogen auf einem Tisch
liegt</string>
00097     <string name="content_description_coinpopup_terry_2">Jake Peralta (Andy Samberg), der erstaunt
etwas sagt</string>
00098     <string name="content_description_coinpopup_terry_3">Terry Jeffords (Terry Crews), der wütend, mit
erhobenem Finger etwas sagt und dabei von Charles Boyle (Joe Lo Truglio) erschrocken angesehen
wird</string>
00099     <string name="content_description_coinpopup_vin">Ein Bild von Vin aus der Mistborn-Reihe von
Brandon Sanderson</string>
00100     <string name="content_description_factionpopup_monster">Karte mit Symbol der
Monsterfraktion</string>
00101     <string name="content_description_factionpopup_nilfgaard">Karte mit Symbol der
Kaiserreich-Nilfgaard-Fraktion</string>
00102     <string name="content_description_factionpopup_northern_kingdoms">Karte mit Symbol der
Nördliche-Königreiche-Fraktion</string>
00103     <string name="content_description_factionpopup_scoiaael">Karte mit Symbol der
Scoia\taelfraktion</string>
00104     <string name="content_description_row_pointball">Punktanzeige der Angriffsreihe</string>
00105     <string name="content_description_row_weather">Wetteranzeige der Angriffsreihe</string>
00106     <string name="content_description_row_horn">Hornanzeige der Angriffsreihe</string>
00107     <string name="content_description_row_card">Kartenanzeige der Angriffsreihe</string>
00108     <string name="content_description_introduction_image">Screenshots der App</string>
00109     <string name="content_description_introduction_indicator">Punkt, der den Fortschritt der
Einführung zeigt</string>
00110     <string name="content_description_introduction_next">Pfeil, der zum nächsten Schritt
führt</string>
00111
00112     <string name="rules_general_text"><![CDATA[<p>Gwint ist ein altes Kartenspiel für zwei Spieler,
das vor allem bei Zwergen beliebt ist. Es stellt die Konfrontation zweier Armeen in einer Schlacht
nach. Die Spieler sind die Generäle und die Karten ihre Streitkräfte. Jeder Gwintspieler verwendet
seinen eigenen, individuell zusammengestellten Kartensatz. Für jede der vier Fraktionen lässt sich ein
eigenes Deck erstellen: Monster, Kaiserreich Nilfgaard, Nördliche Königreiche und Scoia\tael. Jede
Fraktion enthält einige einzigartige Karten, die eine unterschiedliche Spielweise ermöglichen. Des
Weiteren gibt es eine Anzahl neutraler Karten, die in allen vier Fraktionen eingesetzt werden können.
Ziel des Spiels ist es, den Gegner zu besiegen, indem man in zwei von drei Runden eine höhere
Punktzahl als der Gegner erreicht.</p>]]></string>
00113     <string name="rules_course_text"><![CDATA[<p>Zu Beginn der ersten Runde ziehen beide Spieler
jeweils 10 Karten aus ihrem Deck. Danach dürfen zwei Karten abgelegt und neu gezogen werden. Dieser
Schritt ist optional, die abgelegten Karten werden wieder in den Nachziehstapel eingemischt. Der
Spieler, der den ersten Zug macht, wird per Münzwurf ermittelt.<br/>
00114     Die Spieler legen abwechselnd Karten. Es besteht Legezwang! Wenn ein Spieler keine Karte
ausspielen möchte, muss er passen und darf den Rest der Runde keinen Spielzug mehr durchführen.
Daraufhin kann der Gegner noch weitere Karten ausspielen, bis er selbst passt. Sollte ein Spieler
keine Handkarten mehr haben, muss er ebenfalls passen.<br />
00115     Wenn beide Spieler gepasst haben, ist die Runde beendet und der Gewinner wird anhand der
Gesamtstärke der Einheiten ermittelt. Sollte die Gesamtstärke gleich sein, zählt die Runde als Sieg
für beide Spieler. Nun werden alle Karten vom Spielfeld auf den Ablagestapel gelegt. Die Handkarten
werden in der nächsten Runde weiter verwendet, es werden keine neuen Karten gezogen. Der Gewinner hat
in der nächsten Runde den ersten Zug. Bei Unentschieden beginnt derjenige, der in der vergangenen
Runde den zweiten Zug hatte. Ziel des Spiels ist es, zwei Runden zu gewinnen.</p>
00116     ]]></string>
00117     <string name="rules_factions_text"><![CDATA[<p>
00118     Bei Gwint unterscheidet man vier Fraktionen: Monster, Kaiserreich Nilfgaard, Nördliche Königreiche
und die Scoia\tael. Jede dieser Fraktionen besitzt einzigartige Karten, Anführer und jeweils einen
bestimmten Fraktionsvorteil. Zusätzlich gibt es noch neutrale Einheiten, welche keiner Fraktion
zugehörig sind und somit in jedem Deck auftauchen können.</p>
00119     Die Monster werden vom König der Wilden Jagd Eredin Bréacc Glas, auch Sperber genannt, angeführt.
Ihr Vorteil ist die Eisenhaut, welche es ermöglicht, dass nach jeder Runde eine zufällige Einheit auf
dem Feld verbleibt.<br />
00120     Emhyr var Emreis, der sich auch „Die Weiße Flamme, Die Auf den Grabhügeln Der Feinde tanzt“ nennt,
ist Kaiser von Nilfgaard, dessen Stärke der Verrat ist. Dieser ermöglicht es, dass Spieler mit einem
Nilfgaard-Deck jede Runde gewinnen, die sonst als Unentschieden enden würde.<br />
00121     Den expandierenden Nilfgaardern stellen sich die Nördlichen Königreiche unter Foltest, König
Temeriens, entgegen. Diese erhalten nach einer gewonnenen Runde eine Verstärkung, dürfen also eine
Karte vom Nachziehstapel ziehen.<br />
00122     In ihrem Kampf versuchen die Königreiche, Anderlinge zu dezimieren, darunter die Scoia\tael,
welche als nicht-menschliche Rebellen gegen die Repressionen aufbegehren und sich dafür den Hinterhalt
zunutzen machen: Die Scoia\tael können in der ersten Runde entscheiden, wer den ersten Zug macht. Sie
werden von der Elfin Francesca Findabair angeführt, die auch als Enid an Gleanna (Gänseblümchen des
Tals) bekannt ist.
00123     ]]></string>
00124     <string name="rules_commander_text"><![CDATA[<p>

```

```

00125     <p>Jede Fraktion besitzt bestimmte Anführer mit unterschiedlichen Fähigkeiten. Diese werden hier
aufgezählt.</p>
00126     <h1>Monster: Eredin Bréacc Glas</h1>
00127     <p>Der Trügerische: <br />Passiv - Verdoppelt die Stärke aller Karten mit der Fähigkeit "Spion"
(wirkt auf beide Spieler).</p>
00128     <p>Bringer des Todes: <br />Wirf 2 Karten ab und ziehe 1 Karte deiner Wahl aus deinem
Kartensatz.</p>
00129     <p>Führer der Roten Reiter: <br />Durchsuche deinen Nachziehstapel nach einer Wetterkarte und
spiele sie sofort aus.</p>
00130     <p>König der Wilden Jagd: <br />Verdopple die Stärke deiner Nahkampfeinheiten, sofern kein "Horn
des Kommandanten" in der Reihe liegt.</p>
00131     <p>Zerstörer der Welten: <br />Nimm eine Karte von deinem Ablagestapel zurück auf die Hand.</p>
00132     <h1>Kaiserreich Nilfgaard: Emhyr var Emreis</h1>
00133     <p>Der Unbarmherzige: <br />Zieh eine Karte vom Ablagestapel deines Gegners.</p>
00134     <p>Die Weiße Flamme: <br />Lass die Anführer-Fähigkeit deines Gegners unwirksam werden.</p>
00135     <p>Invasor des Nordens: <br />Fähigkeiten, die Einheiten auf dem Schlachtfeld wiederherstellen,
stellen zufällig ausgewählte Einheiten wieder her. Wirkt auf beide Spieler.</p>
00136     <p>Kaiser von Nilfgaard: <br />Schau dir drei zufällig gewählte Karten aus der Hand deines Gegners
an.</p>
00137     <p>Kaiserliche Majestät: <br />Durchsuche deinen Nachziehstapel nach "Strömender Regen" und spiele
die Karte sofort aus.</p>
00138     <h1>Nördliche Königreiche: Foltest</h1>
00139     <p>Der Stählerne: <br />Zerstöre die stärkste(n) Belagerungseinheit(en) des Gegners, sofern deren
Gesamtstärke mindestens 10 beträgt.</p>
00140     <p>Der Anführer des Nordens: <br />Entferne durch "Klirrende Kälte", "Strömender Regen" und
"Extrem dichter Nebel" hervorgerufene Wettereffekte.</p>
00141     <p>Der Belagerer: <br />Verdopple die Stärke deiner Belagerungseinheiten, sofern kein "Horn des
Kommandanten" in der Reihe liegt.</p>
00142     <p>König Temeriens: <br />Durchsuche deinen Nachziehstapel nach "Extrem dichter Nebel" und spiele
die Karte sofort aus.</p>
00143     <p>Sohn des Medell: <br />Zerstöre die stärkste(n) Fernkampfeinheit(en) des Gegners, wenn die
Gesamtstärke seiner Fernkampfeinheiten mindestens 10 beträgt.</p>
00144     <h1>Scoia'tael: Francesca Findabair</h1>
00145     <p>Gänseblümchen des Tals: <br />Zieh zu Beginn der Schlacht eine Extrakarte.</p>
00146     <p>Hoffnung der Aen Seidhe: <br />Verschiebt Einheiten mit der Fähigkeit „Gewandtheit“ auf die für
sie optimale Reihe, sofern sie nicht bereits dort befinden.</p>
00147     <p>Königin Dol Blathannas: <br />Zerstöre die stärkste(n) Nahkampfeinheit(en) des Gegners, wenn
die Gesamtstärke seiner Nahkampfeinheiten mindestens 10 beträgt.</p>
00148     <p>Die Schöne: <br />Verdopple die Stärke deiner Fernkampfeinheiten, sofern kein „Horn des
Kommandanten“ in der Reihe liegt.</p>
00149     <p>Reinblütige Elfin: <br />Durchsuche deinen Nachziehstapel nach „Klirrende Kälte“ und spiele die
Karte sofort aus.</p>
00150     ]]></string>
00151     <string name="rules_cards_text"><![CDATA[<p>
00152     Man unterscheidet bei den Kartentypen in drei Hauptkategorien: Anführer-, Einheiten- und
Sonderkarten. In jedem Kartensatz gibt es genau eine Anführerkarte (je Fraktion kann der Spieler
zwischen vier unterschiedlichen wählen). Diese hat eine bestimmte Fähigkeit, die nur einmal im ganzen
Spiel während eines Zuges aktiviert werden kann, dafür aber unabhängig von den Handkarten immer
verfügbar ist.<br />
00153     Dazu kommen mindestens 22 Einheitenkarten, zu denen auch die Heldenkarten zählen. Letztere sind
gegen Karteneffekte jeglicher Art (positiv wie negativ) immun. Die Einheitenkarten werden während
eines Spielzuges in Angriffsreihen gelegt. Jeder Spieler besitzt davon drei, jeweils eine für
Nahkämpfer, Fernkämpfer und Belagerungsgeräte. In welche dieser Reihen die Einheit gelegt werden kann,
erkennt man am orange hinterlegten Symbol.<br />
00154     Schließlich gibt es noch maximal zehn Sonderkarten, zu denen „Finte“, „Verbrennen“, „Horn des
Kommandanten“ und die Wetterkarten gehören (Effektbeschreibungen sind unter „Spezialkarten“
nachzulesen). Wetterkarten werden in einen abgesonderten Bereich des Spielfelds platziert, da sie die
Reihen beider Spieler betreffen. Das Horn des Kommandanten wird an die Reihe angelegt, für die es
ausgespielt wird.
00155     ]]></string>
00156     <string name="rules_card_abilities_text"><![CDATA[
00157     <p>Einige Einheitenkarten besitzen besondere Fähigkeiten, welche hier genauer erklärt werden.</p>
00158     <p>Spion: <br />Lege diese Karte auf das Schlachtfeld des Gegners - sie wird auf die Gesamtstärke
des Gegners angerechnet - und ziehe zwei Karten.</p>
00159     <p>Enge Bindung: <br />Liegt diese Karte neben einer mit demselben Namen, verdopple die Stärke
beider Karten.</p>
00160     <p>Musterung: <br />Wird diese Karte gespielt, durchsuche deinen Nachziehstapel nach Einheiten mit
demselben Namen und spiele diese sofort aus.</p>
00161     <p>Moralschub: <br />Verstärke alle anderen Einheiten der Reihe um einen Stärkepunkt.</p>
00162     <p>Heilung: <br />Wird diese Karte gespielt, wähle eine Einheitenkarte aus deinem Ablagestapel
(Heldenkarten ausgenommen) und spiele diese aus. Karten mit der Fähigkeit Gewandtheit müssen wieder in
ihre ursprüngliche Reihe zurückgelegt werden.</p>
00163     <p>Horn des Kommandanten: <br />Verdopple die Stärke aller anderen Einheitenkarten dieser Reihe.
Dieser Effekt kumuliert nicht mit dem Effekt der Spezialkarte Horn des Kommandanten! Liegen also beide
Karten in der Angriffsreihe, wird der Schaden aller Einheiten (einschließlich dieser) nur einmal
verdoppelt.</p>
00164     <p>Gewandtheit: <br />Platziere diese Karte in der Nah- oder Fernkampfreihe. Nach dem Ausspielen
kannst du sie nicht mehr verschieben.</p>
00165     <p>Verbrennen: <br />Zerstöre die stärkste(n) Nahkampfeinheit(en) des Gegners, wenn die
Gesamtstärke seiner Nahkampfeinheiten mindestens 10 beträgt.</p>
00166     <p>Rächer herbeirufen: <br />Wird diese Karte vom Spielfeld entfernt (egal auf welche Art), wird
ein Rächer herbeigerufen.</p>
00167     ]]></string>
00168     <string name="rules_special_cards_text"><![CDATA[
00169     <p>Spezialkarten können alternativ zu Einheitenkarten gespielt werden, wobei bestimmte Effekte
ausgelöst werden, die hier erklärt werden.</p>
00170     <p>Klirrende Kälte: <br />Setze die Stärke aller Nahkampfeinheiten beider Spieler auf 1. Wenn

```

```

    diese Karte bereits ausgespielt wurde, lege sie sofort auf den Ablagestapel.</p>
00171 <p>Extrem dichter Nebel: <br />Setze die Stärke aller Fernkampfeinheiten beider Spieler auf 1.
    Wenn diese Karte bereits ausgespielt wurde, lege sie sofort auf den Ablagestapel.</p>
00172 <p>Strömender Regen: <br />Setze die Stärke aller Belagerungseinheiten beider Spieler auf 1. Wenn
    diese Karte bereits ausgespielt wurde, lege sie sofort auf den Ablagestapel.</p>
00173 <p>Gutes Wetter: <br />Entferne alle Effekte von Wetterkarten und lege sie mit dieser Karte auf
    den Ablagestapel.</p>
00174 <p>Finte: <br />Tausche diese Karte gegen eine Einheitenkarte auf dem Schlachtfeld aus, um sie
    wieder deiner Hand hinzuzufügen. </p>
00175 <p>Horn des Kommandanten: <br />Verdopple die Stärke aller Einheitenkarte dieser Reihe (maximal
    einmal pro Reihe). Dieser Effekt kumuliert nicht mit dem Karteneffekt Rittersporns! Liegen also beide
    Karten in der Angriffsreihe, wird der Schaden aller Einheiten (einschließlich Rittersporn) nur einmal
    verdoppelt.</p>
00176 <p>Verbrennen: <br />Entferne die stärkste(n) Karte(n) beider Spieler auf dem Schlachtfeld und
    lege sie mit dieser Karte auf den Ablagestapel.</p>
00177 </string>
00178 <string name="introduction_page1"><![CDATA[
00179 <p>Danke, dass du „Gwint – Eine Rechenhilfe“ installiert hast! Bitte beachte, dass diese App nur
    dazu da ist, dich bei einem Spiel mit analogen Gwintkarten zu unterstützen, indem sie deinen
    Punktestand errechnet. Du kannst in dieser App nicht gegen KI-Gegner oder ohne analoge Gwintkarten
    spielen!</p>
00180 <p>Also falls du es noch nicht getan hast: Drucke dir einen Satz Gwintkarten aus (<a
    href="https://github.com/PeterNaggschga/Gwent#overview">Anleitung hier</a>) und suche dir einen
    Gegner, der dasselbe tut (ihr braucht beide diese App)!</p>
00181 </string>
00182 <string name="introduction_page2"><![CDATA[
00183 <p>Dein Spielfeld besteht aus drei Angriffsreihen: Nahkampfreihe, Fernkampfreihe und
    Belagerungsreihe (von links nach rechts). In diese Reihen werden die Einheitenkarten beim Ausspielen
    gelegt. Jede Angriffsreihe kann außerdem durch eine bestimmte Wetterlage geschwächt oder durch ein
    Horn des Kommandanten verstärkt werden.</p>
00184 </string>
00185 <string name="introduction_page3"><![CDATA[
00186 <p>Die Angriffsreihen sind wie folgt aufgebaut:</p>
00187 <p>1: Die Gesamtstärke aller Einheiten in der Reihe.</p>
00188 <p>2: Das aktuelle Wetter (kann durch antippen geändert werden).</p>
00189 <p>3: Die Verstärkung durch ein Horn des Kommandanten (kann durch antippen geändert werden).</p>
00190 <p>4: Die Anzahl der Einheiten in dieser Reihe (durch Tippen können Einheiten hinzugefügt oder
    entfernt werden).</p>
00191 </string>
00192 <string name="introduction_page4"><![CDATA[
00193 <p>Neben deinem eigentlichen Spielfeld gibt es an der Seite ein Menü mit weiteren nützlichen
    Funktionen:</p>
00194 <p>1: Gesamtstärke aller Einheiten auf dem Spielfeld</p>
00195 <p>2: Fraktion ändern</p>
00196 <p>3: Spielfeld zurücksetzen</p>
00197 <p>4: Wettereffekte entfernen</p>
00198 <p>5: Stärkste Einheit(en) verbrennen</p>
00199 <p>6: Münzwurf</p>
00200 <p>7: Einstellungen</p>
00201 </string>
00202 <string name="introduction_page5"><![CDATA[
00203 <p>Weitere Informationen zum Spiel und seinen Regeln findest du in den Einstellungen. Dort kannst
    du diese Tour auch wiederholen!</p>
00204 <p>Doch jetzt erstmal: viel Spaß mit analogem Gwint! Über Verbesserungsvorschläge, Fehlermeldungen
    und sonstiges Feedback freue ich mich sehr, nutze dazu gerne den <a
    href="https://play.google.com/store/apps/details?id=com.peternaggschga.gwint">Google Play Store</a>
    oder <a href="https://github.com/PeterNaggschga/Gwent/issues/new/choose">GitHub</a>!</p>
00205 </string>
00206 <string name="faction_monster">Monster</string>
00207 <string name="array_design_jaskier">Rittersporn</string>
00208 <string name="array_design_default">Standard</string>
00209 <string name="array_design_geralt">Geralt von Riva</string>
00210 <string name="array_design_yennefer">Yennefer von Vengerberg</string>
00211 <string name="add_picker_epic_epic">Held</string>
00212 <string name="add_picker_ability_default">Keine</string>
00213 <string name="add_picker_ability_moralBoost">Moralschub</string>
00214 <string name="add_picker_ability_revenge">Rächer</string>
00215 <string name="add_picker_ability_binding">Enge Bindung</string>
00216 <string name="popUp_add_card_binding_count">Anzahl Einheiten in %1$d. Gruppe: %2$d</string>
00217 <string name="alertDialog_revenge_negative">Nicht auslösen</string>
00218 </resources>

```

6.103 strings.xml

```

00001 <resources>
00002 <string name="app_name">Gwent</string>
00003 <string name="app_description">An app enabling you to play the Gwent card game from The Witcher 3
    with analog cards.</string>
00004
00005 <string name="faction_monster">Monsters</string>
00006 <string name="faction_nilfgaard">Nilfgaardian Empire</string>
00007 <string name="faction_northern_kingdoms">Northern Kingdoms</string>
00008 <string name="faction_scoiatael" translatable="false">Scoia'tael</string>
00009
00010 <string name="popUp_cards_add">Add card</string>

```



```

00011     <string name="popUp_cards_save">Save</string>
00012     <string name="popUp_add_card_label_epic" translatable="false">Status</string>
00013     <string name="add_picker_epic_normal" translatable="false">Normal</string>
00014     <string name="add_picker_epic_epic">Hero</string>
00015     <string name="popUp_add_card_label_dmz">Strength</string>
00016     <string name="popUp_add_card_label_ability">Ability</string>
00017     <string name="add_picker_ability_default">None</string>
00018     <string name="add_picker_ability_moralBoost">Morale boost</string>
00019     <string name="add_picker_ability_horn" translatable="false">Horn</string>
00020     <string name="add_picker_ability_revenge">Avenger</string>
00021     <string name="add_picker_ability_binding">Tight bond</string>
00022     <string name="popUp_add_card_label_binding">Squad</string>
00023     <string name="popUp_add_card_label_number">Number</string>
00024     <string name="popUp_add_card_binding_count">Number of units in squad %1$d: %2$d</string>
00025
00026     <string name="popUp_coin_normal_win">You begin.</string>
00027     <string name="popUp_coin_normal_lose">Your opponent begins.</string>
00028     <string name="popUp_coin_terry_1">Terry Jeffords folded your coin!</string>
00029     <string name="popUp_coin_terry_2" translatable="false">Oh my god! Did you just fold my quarter in
half?</string>
00030     <string name="popUp_coin_terry_3" translatable="false">Fate\'s a bitch--you make the
choice!</string>
00031     <string name="popUp_coin_stewie_1">Chris Griffin ate your coin!</string>
00032     <string name="popUp_coin_stewie_2" translatable="false">But don\'t worry, it will come out of my
body sooner or later!</string>
00033     <string name="popUp_coin_stewie_3" translatable="false">He\'s right. I ate a dime once.</string>
00034     <string name="popUp_coin_stewie_4" translatable="false">It became a manhole cover for like three
days--but then pow!</string>
00035     <string name="popUp_coin_vin">Vin burned steel and pushed away your coin!</string>
00036
00037     <string name="unit_toString">%1$s %2$s (%3$d) with %4$s ability %5$s</string>
00038     <string name="unit_toString_multiplicity">%1$d x %2$s</string>
00039     <string name="unit_toString_melee">Close combat</string>
00040     <string name="unit_toString_range">Ranged combat</string>
00041     <string name="unit_toString_siege">Siege combat</string>
00042     <string name="unit_toString_epic">hero</string>
00043     <string name="unit_toString_unit">unit</string>
00044     <string name="unit_toString_ability_none">no</string>
00045     <string name="unit_toString_squad">(%1$d)</string>
00046     <string name="unit_collection_toString_accumulation_symbol" translatable="false">%1$s,
%2$s</string>
00047     <string name="unit_collection_toString_accumulation_word">%1$s, and %2$s</string>
00048
00049     <string name="alertDialog_cancel">Cancel</string>
00050
00051     <string name="alertDialog_burn_title">Scorch</string>
00052     <string name="alertDialog_burn_positive">Scorch</string>
00053     <string name="alertDialog_burn_negative">Don\'t scorch</string>
00054     <string name="alertDialog_burn_msg">Do you really want to scorch %1$s?</string>
00055
00056     <string name="alertDialog_reset_title">Reset board</string>
00057     <string name="alertDialog_reset_msg_default">Do you really want to reset the board?</string>
00058     <string name="alertDialog_reset_msg_faction_switch">You activated \'Reset on change of faction\'.
Do you really want to reset the board?</string>
00059     <string name="alertDialog_reset_checkbox">Activate perk of monster faction (you keep one random
unit)</string>
00060     <string name="alertDialog_factionreset_monster_toast_keep">You keep: %1$s</string>
00061     <string name="alertDialog_reset_positive">Reset</string>
00062     <string name="alertDialog_reset_negative">Don\'t reset</string>
00063
00064     <string name="alertDialog_revenge_title">Avenger ability</string>
00065     <string name="alertDialog_revenge_msg">You removed units that can summon an avenger. Do you want
to activate their ability?</string>
00066     <string name="alertDialog_revenge_positive">Summon avenger</string>
00067     <string name="alertDialog_revenge_negative">Don\'t activate</string>
00068
00069     <string name="array_design_default">Default</string>
00070     <string name="array_design_geralt">Geralt of Rivia</string>
00071     <string name="array_design_ciri" translatable="false">Cirilla</string>
00072     <string name="array_design_jaskier">Dandelion</string>
00073     <string name="array_design_yennefer">Yennefer of Vengerberg</string>
00074     <string name="array_design_erodin" translatable="false">Eredin Bréacc Glas</string>
00075     <string name="preference_design">Background</string>
00076     <string name="preference_warnings_title">Warnings</string>
00077     <string name="preference_warnings_desc">Determines if warnings are shown before you can reset the
board.</string>
00078     <string name="preference_faction_reset_title">"Reset on change of faction"</string>
00079     <string name="preference_faction_reset_desc">Determines if the board is reset on change of
faction.</string>
00080
00081     <string name="preference_sounds_header">Sounds</string>
00082     <string name="preference_sounds_category">Events</string>
00083     <string name="preference_sounds_cards_new">New unit</string>
00084     <string name="preference_sounds_cards_delete">Unit removed</string>
00085     <string name="preference_sounds_weather">Change of weather</string>
00086     <string name="preference_sounds_horn">Commander\'s horn</string>
00087     <string name="preference_sounds_burn">Scorch</string>

```

```

00088     <string name="preference_sounds_coin">Coin-flip</string>
00089
00090     <string name="preference_info_category_header" translatable="false">Info</string>
00091     <string name="preference_info_introduction">Introduction</string>
00092
00093     <string name="preference_rules_header">Rules</string>
00094     <string name="preference_rules_general_title">General</string>
00095     <string name="preference_rules_course_title">Gameplay</string>
00096     <string name="preference_rules_factions_title">Factions</string>
00097     <string name="preference_rules_commander_title">Leaders</string>
00098     <string name="preference_rules_cards_title">Deck and playboard</string>
00099     <string name="preference_rules_card_abilities_title">Card abilities</string>
00100     <string name="preference_rules_special_cards_title">Special cards</string>
00101     <string name="preference_rules_general_key" translatable="false">rules_general</string>
00102     <string name="preference_rules_course_key" translatable="false">rules_course</string>
00103     <string name="preference_rules_factions_key" translatable="false">rules_factions</string>
00104     <string name="preference_rules_commander_key" translatable="false">rules_commander</string>
00105     <string name="preference_rules_cards_key" translatable="false">rules_cards</string>
00106     <string name="preference_rules_card_abilities_key"
00107     translatable="false">rules_card_abilities</string>
00108     <string name="preference_rules_special_cards_key"
00109     translatable="false">rules_special_cards</string>
00110     <string name="preference_first_use_key" translatable="false">firstUse</string>
00111
00112     <string name="introduction_button_skip">Skip</string>
00113     <string name="introduction_button_finish">Finish</string>
00114
00115     <string name="content_description_activity_main_background">background</string>
00116     <string name="content_description_activity_main_point_ball">total score</string>
00117     <string name="content_description_activity_main_faction_button">faction logo</string>
00118     <string name="content_description_activity_main_reset_button"
00119     translatable="false">Reset-Icon</string>
00120     <string name="content_description_activity_main_weather_button">Clear-Weather-Icon</string>
00121     <string name="content_description_activity_main_burn_button">Scorch-Icon</string>
00122     <string name="content_description_activity_main_coin_button">Coin-Icon</string>
00123     <string name="content_description_activity_main_settings_button">Settings-Icon</string>
00124     <string name="content_description_card_ability">Ability-Icon</string>
00125     <string name="content_description_card_delete">Delete-Icon</string>
00126     <string name="content_description_card_copy">Copy-Icon</string>
00127     <string name="content_description_coinpopup_coin">coin</string>
00128     <string name="content_description_coinpopup_stewie_1">Chris Griffin who raises his arms in
00129     consternation</string>
00130     <string name="content_description_coinpopup_stewie_2">Stewie Griffin who tells a story</string>
00131     <string name="content_description_coinpopup_stewie_3">Stewie Griffin who yells</string>
00132     <string name="content_description_coinpopup_terry_1">A bent coin lying on a table</string>
00133     <string name="content_description_coinpopup_terry_2">Jake Peralta (Andy Samberg) who says
00134     something in astonishment</string>
00135     <string name="content_description_coinpopup_terry_3">Terry Jeffords (Terry Crews) who says
00136     something angrily and Charles Boyle (Joe Lo Truglio) looking at him in terror</string>
00137     <string name="content_description_coinpopup_vin">A fan art showing Vin from Brandon Sanderson\'s
00138     Mistborn-series.</string>
00139     <string name="content_description_factionpopup_monster">Card with symbol of monster
00140     faction</string>
00141     <string name="content_description_factionpopup_nilfgaard">Card with symbol of nilfgaardian empire
00142     faction</string>
00143     <string name="content_description_factionpopup_northern_kingdoms">Card with symbol of Northern
00144     Kingdoms faction</string>
00145     <string name="content_description_factionpopup_scoia_tael">Card with symbol of Scoia\'tael
00146     faction</string>
00147     <string name="content_description_row_pointball">point display of combat row</string>
00148     <string name="content_description_row_weather">weather display of combat row</string>
00149     <string name="content_description_row_horn">horn display of combat row</string>
00150     <string name="content_description_row_card">card display of combat row</string>
00151     <string name="content_description_introduction_image">screenshots of this app</string>
00152     <string name="content_description_introduction_indicator">dot that indicates progress</string>
00153     <string name="content_description_introduction_next">forward arrow</string>
00154
00155     <string name="rules_general_text"><![CDATA[<p>Gwent is an old card game especially fancied by
00156     dwarfs. It represents the confrontation of two opposing armies. The players are generals and cards
00157     represent their units. Every player uses his own unique deck of cards. Decks can be created for each
00158     of the four factions: Monsters, Nilfgaardian Empire, Northern Kingdoms and Scoia\'tael. Each faction
00159     has unique cards which lead to different play styles. Moreover, there are cards that can be used for
00160     every faction. Your goal is to win two rounds out of three by having more strength than your
00161     opponent.</p>]]></string>
00162     <string name="rules_course_text"><![CDATA[<p>At the beginning of the first round each player draws
00163     ten cards from their deck. After drawing, you can discard two cards and draw a new card for each. This
00164     step is optional, discarded cards are put back into the deck. The beginning player is determined by
00165     flipping a coin.<br>The players place cards alternately. You have to play one card every round! If a
00166     player doesn\'t want to play a card, they have to pass and aren\'t allowed to play another card until
00167     the round has ended. When one player passes, their opponent can still play cards until they pass too.
00168     If one of the opponents uses up all of his cards, he has to pass.<br>A round ends when both players
00169     pass. The player with the highest total score wins the round. When both opponents have the same number
00170     of points, the round is seen as a win for both sides. All cards are moved to the discard pile at the
00171     end of each round. Your hand is reused in the next round--you don\'t draw new cards. The winner of the
00172     last round gets the first turn. Goal of the game is to win two rounds.</p>]]></string>
00173     <string name="rules_factions_text"><![CDATA[<p>There are four factions: Monsters, Nilfgaardian
00174     Empire, Northern Kingdoms and Scoia\'tael. Every faction has got unique cards, leaders and perks.

```

Moreover, there are some additional cards that are neutral and can be used in every faction.</p>Leader of monster faction is Eredin Bréacc Glas, also known as Sparrowhawk who is the King of the Wild Hunt. Their perk is iron skin which makes it possible to keep one unit at the end of each round.
Emhyr var Emreis, also known as "The White Flame Dancing on the Barrows of his Enemies", is emperor of Nilfgaard. Treachery causes any player with a nilfgaardian deck to win if points draw level.
Archenemy of Nilfgaard is Foltest, King of Temeria and leader of the Northern Kingdoms faction. They are able to draw one card after a won round.
In their fight, the Northern Kingdoms try to decimate Nonhumans which are represented by the Scoia'tael faction. Scoia'tael players may decide who gets the first turn in the first round. They are led by Francesca Findabair, also known as Enid an Gleanna (Daisy of the Valleys).]]</string>

00147 <string name="rules_commander_text"><![CDATA[<p>
00148 <p>Each faction has certain leaders with different abilities. They are listed here.</p>
00149 <h5>Monsters: Eredin Bréacc Glas</h5>
00150 <p>The Treacherous:
Passive - Doubles the strength of all spy cards (affects both
players).</p>
00151 <p>Bringer of Death:
Restore a card from your discard pile to your hand.</p>
00152 <p>Commander of the Red Riders:
Double the strength of all your Close combat units (unless a
Commander's horn is also present in that row).</p>
00153 <p>King of the Wild Hunt:
Pick any weather card from your deck and play it instantly.</p>
00154 <p>Destroyer of Worlds:
Discard two cards and draw one card of your choice from your
deck.</p>
00155 <h5>Nilfgaardian Empire: Emhyr var Emreis</h5>
00156 <p>The Relentless:
Draw a card from your opponent's discard pile.</p>
00157 <p>The White Flame:
Cancel your opponent's Leader Ability.</p>
00158 <p>Invader of the North:
Abilities that restore a unit to the battlefield restore a
randomly-chosen unit. Affects both players.</p>
00159 <p>Emperor of Nilfgaard:
Look at three random cards from your opponent's hand.</p>
00160 <p>His Imperial Majesty:
Pick a Torrential Rain card from your deck and play it
instantly.</p>
00161 <h5>Northern Kingdoms: Foltest</h5>
00162 <p>The Steel-Forged:
Destroy your enemy's strongest Siege unit(s) if the combined strength
of all their Siege units is 10 or more.</p>
00163 <p>Lord Commander of the North:
Clear any weather effects (resulting from Biting Frost,
Torrential Rain or Impenetrable Fog cards) in play.</p>
00164 <p>The Siegesmaster:
Doubles the strength of all your Siege units (unless a Commander's
Horn is also present on that row).</p>
00165 <p>King of Temeria:
Pick an Impenetrable Fog card from your deck and play it instantly.</p>
00166 <p>Son of Medell:
Destroy your enemy's strongest Ranged combat unit(s) if the combined
strength of all their Ranged combat units is 10 or more.</p>
00167 <h5>Scoia'tael: Francesca Findabair</h5>
00168 <p>Daisy of the Valley:
Draw an extra card at the beginning of the battle.</p>
00169 <p>Hope of the Aen Seidhe:
Move agile units to whichever valid row maximizes their strength
(don't move units already in optimal row).</p>
00170 <p>Queen of Dol Blathanna:
Destroy your enemy's strongest Close combat unit(s) if the
combined strength of all their Close combat units is 10 or more.</p>
00171 <p>The Beautiful:
Doubles the strength of all your Ranged combat units (unless a
Commander's Horn is also present on that row).</p>
00172 <p>Pureblood Elf:
Pick a Biting Frost card from your deck and play it instantly.</p>
00173]]</string>
00174 <string name="rules_cards_text"><![CDATA[<p>
00175 There are three major types of cards: Leaders, units and Special Cards.
00176 Each deck has one leader card.
00177 It has a certain ability which can be activated once per game.

00178 Moreover, there are at least 22 Unit Cards including Heroes.
00179 Heroes are immune to any boost or weakening.
00180 Units are played in turns.
00181 Each player has got three combat rows, one for Close combat, Ranged combat and Siege units each.
00182 It can be seen on the cards which row must be used.

00183 Also there are Special Cards (maximum 10), "Decoy",
00184 "Scorch", "Commander's Horn" and weather cards.
00185 Weather cards are placed in a separate section of the board
00186 because they influence both players equally.
00187 Commander's Horn is placed at the designated combat row.</p>
00188]]</string>
00189 <string name="rules_card_abilities_text"><![CDATA[
00190 <p>Some unit cards have got special abilities which are explained in the following.</p>
00191 <p>Spy:
Must be placed on your opponent's battlefield (and counts toward your opponent's
total score), and allows you to draw 2 cards extra from your deck.</p>
00192 <p>Tight Bond:
When placed next to a card with the same name, double the strength of both
cards.</p>
00193 <p>Muster:
When played, automatically find any cards with the same name in your deck and
play them instantly.</p>
00194 <p>Morale Boost:
When played, add +1 strength to all other units in the row it is played
in.</p>
00195 <p>Medic:
Allow you to choose one card from your discard pile and play it instantly (no
Heroes or Special Cards). Agile units have to be placed in the combat row they have been in
before.</p>
00196 <p>Commander's Horn:
Double the strength of all other units in this combat row.</p>
00197 <p>Agile:
Can be placed in either the Close combat or the Ranged combat row. They cannot be
moved once placed.</p>
00198 <p>Scorch:
Kill the strongest card(s) in the specified row of the enemy.</p>
00199 <p>Summon Avenger:
Summon a powerful new unit card to take their place when they are removed
from the battlefield.</p>
00200]]</string>
00201 <string name="rules_special_cards_text"><![CDATA[
00202 <p>Special Cards can be played instead of Unit Cards and trigger certain effects which are listed
below.</p>


```

00203     <p>Biting Frost: <br />Sets all Close combat units to strength 1 for both players.</p>
00204     <p>Impenetrable Fog: <br />Sets all Ranged combat units to strength 1 for both players.</p>
00205     <p>Torrential Rain: <br />Sets all Siege units to strength 1 for both players.</p>
00206     <p>Clear Weather: <br />Removes all weather effects.</p>
00207     <p>Decoy: <br />Swap with a card on the battlefield.</p>
00208     <p>Commander's Horn: <br />Doubles the strength of the selected row.</p>
00209     <p>Scorch: <br />Discard after playing. Kills the strongest card(s) on the battlefield.</p>
00210     <]]></string>
00211     <string name="introduction_page1"><![CDATA[
00212     <p>Thanks for installing "Gwent - A Calculator"! Please note that this app is made to support
you in playing Gwent using analog cards by calculating your points. This app does not allow you to
play Gwent against AI or without analog cards!</p>
00213     <p>So if you didn't do so yet: Print out a deck of Gwent cards (<a
href="https://github.com/PeterNaggschga/Gwent#overview">guidance here</a>) and find an opponent who
does the same (both of you need to install this app).</p>
00214     <]]></string>
00215     <string name="introduction_page2"><![CDATA[
00216     <p>Your game board consists of three combat rows: Close combat, Ranged combat and Siege row
(left to right).
00217     These rows contain unit cards.
00218     Each of them can be weakened by certain weather cards or boosted by a Commander's Horn.</p>
00219     <]]></string>
00220     <string name="introduction_page3"><![CDATA[
00221     <p>Combat rows consist of the following parts:</p>
00222     <p>1: Overall strength of all units placed in this row.</p>
00223     <p>2: The current weather (can be changed by tapping).</p>
00224     <p>3: The boost of a Commander's Horn (can be changed by tapping).</p>
00225     <p>4: Number of units in this combat row (you can add or remove units by tapping).</p>
00226     <]]></string>
00227     <string name="introduction_page4"><![CDATA[
00228     <p>Next to the game board there is a menu providing useful functions:</p>
00229     <p>1: Overall strength of all units</p>
00230     <p>2: Change faction</p>
00231     <p>3: Reset board</p>
00232     <p>4: Clear weather</p>
00233     <p>5: Scorch strongest unit(s)</p>
00234     <p>6: Coin-flip</p>
00235     <p>7: Settings</p>
00236     <]]></string>
00237     <string name="introduction_page5"><![CDATA[
00238     <p>More information on Gwent and its rules can be found in the settings. There you can repeat this
introduction too!</p>
00239     <p>But first: Have fun with analog Gwent! Please use the <a
href="https://play.google.com/store/apps/details?id=com.peternaggschga.gwent">Google Play Store</a> or
<a href="https://github.com/PeterNaggschga/Gwent/issues/new/choose">GitHub</a> to share improvement
suggestions, bugs and other feedback with me!</p>
00240     <]]></string>
00241     <string name="settings_title">Settings</string>
00242 </resources>
00243 </resources>

```

6.104 styles.xml

```

00001 <resources>
00002
00003     <style name="BaseTheme" parent="Theme.AppCompat.NoActionBar">
00004         <item name="windowNoTitle">true</item>
00005         <item name="windowActionBar">false</item>
00006         <item name="android:windowFullscreen">true</item>
00007         <item name="android:windowContentOverlay">@null</item>
00008         <item name="android:windowTranslucentNavigation">true</item>
00009     </style>
00010
00011     <style name="MonsterTheme" parent="BaseTheme">
00012         <item name="colorPrimary">@color/color_text_monster</item>
00013         <item name="colorAccent">@color/color_accent_monster</item>
00014         <item name="card_count_text_color">@color/color_text_monster</item>
00015         <item name="android:alertDialogIcon">@drawable/icon_round_monster</item>
00016         <item name="point_ball_mipmap">@drawable/ball_red</item>
00017         <item name="card_view_mipmap">@drawable/card_monster_landscape_free</item>
00018         <item name="card_background_color">@color/color_background_monster</item>
00019         <item name="card_background_gradient">@color/color_gradient_monster</item>
00020         <item name="card_view_button_background">@color/color_card_view_button_background_monster</item>
00021     </style>
00022
00023
00024     <style name="NilfgaardTheme" parent="BaseTheme">
00025         <item name="colorPrimary">@color/color_text_nilfgaard</item>
00026         <item name="colorAccent">@color/color_accent_nilfgaard</item>
00027         <item name="card_count_text_color">@color/color_text_nilfgaard</item>
00028         <item name="android:alertDialogIcon">@drawable/icon_round_nilfgaard</item>
00029         <item name="point_ball_mipmap">@drawable/ball_grey</item>
00030         <item name="card_view_mipmap">@drawable/card_nilfgaard_landscape_free</item>
00031         <item name="card_background_color">@color/color_background_nilfgaard</item>
00032         <item name="card_background_gradient">@color/color_gradient_nilfgaard</item>
00033         <item name="card_view_button_background">

```

```

00034         @color/color_card_view_button_background_nilfgaard
00035     </item>
00036 </style>
00037
00038 <style name="NorthernKingdomsTheme" parent="BaseTheme">
00039     <item name="colorPrimary">@color/color_text_northern_kingdoms</item>
00040     <item name="colorAccent">@color/color_accent_northern_kingdoms</item>
00041     <item name="card_count_text_color">@color/color_text_northern_kingdoms</item>
00042     <item name="android:alertDialogIcon">@drawable/icon_round_northern_kingdoms</item>
00043     <item name="point_ball_mipmap">@drawable/ball_blue</item>
00044     <item name="card_view_mipmap">@drawable/card_northern_kingdoms_landscape_free</item>
00045     <item name="card_background_color">@color/color_background_northern_kingdoms</item>
00046     <item name="card_background_gradient">@color/color_gradient_northern_kingdoms</item>
00047     <item name="card_view_button_background">
00048         @color/color_card_view_button_background_northern_kingdoms
00049     </item>
00050 </style>
00051
00052 <style name="ScoiataelTheme" parent="BaseTheme">
00053     <item name="colorPrimary">@color/color_text_scoiatael</item>
00054     <item name="colorAccent">@color/color_accent_scoiatael</item>
00055     <item name="card_count_text_color">@color/color_text_scoiatael</item>
00056     <item name="android:alertDialogIcon">@drawable/icon_round_scoiatael</item>
00057     <item name="point_ball_mipmap">@drawable/ball_green</item>
00058     <item name="card_view_mipmap">@drawable/card_scoiatael_landscape_free</item>
00059     <item name="card_background_color">@color/color_background_scoiatael</item>
00060     <item name="card_background_gradient">@color/color_gradient_scoiatael</item>
00061     <item name="card_view_button_background">
00062         @color/color_card_view_button_background_scoiatael
00063     </item>
00064 </style>
00065 </resources>

```

6.105 header_preferences.xml

```

00001 <PreferenceScreen xmlns:app="http://schemas.android.com/apk/res-auto">
00002
00003     <ListPreference
00004         app:defaultValue="0"
00005         app:entries="@array/design_names"
00006         app:entryValues="@array/design_values"
00007         app:icon="@drawable/icon_design"
00008         app:key="@string/preference_key_design"
00009         app:title="@string/preference_design"
00010         app:useSimpleSummaryProvider="true" />
00011
00012     <SwitchPreference
00013         app:defaultValue="@bool/warning_preference_default"
00014         app:icon="@drawable/icon_warnings"
00015         app:key="@string/preference_key_warning"
00016         app:summary="@string/preference_warnings_desc"
00017         app:title="@string/preference_warnings_title" />
00018
00019     <SwitchPreference
00020         app:defaultValue="@bool/faction_reset_preference_default"
00021         app:icon="@drawable/icon_factionreset"
00022         app:key="@string/preference_key_faction_reset"
00023         app:summary="@string/preference_faction_reset_desc"
00024         app:title="@string/preference_faction_reset_title" />
00025
00026     <Preference
00027         app:fragment="com.peternaggschga.gwent.ui.settings.SettingsSoundFragment"
00028         app:icon="@drawable/icon_sounds"
00029         app:title="@string/preference_sounds_header" />
00030
00031     <PreferenceCategory
00032         app:allowDividerAbove="true"
00033         app:key="info"
00034         app:title="@string/preference_info_category_header">
00035
00036         <Preference
00037             app:fragment="com.peternaggschga.gwent.ui.settings.SettingsRuleFragment"
00038             app:icon="@drawable/icon_rules"
00039             app:title="@string/preference_rules_header" />
00040
00041         <Preference
00042             app:icon="@drawable/icon_explore"
00043             app:key="@string/preference_key_introduction"
00044             app:title="@string/preference_info_introduction" />
00045
00046     </PreferenceCategory>
00047 </PreferenceScreen>

```

6.106 rule_preferences.xml

```
00001 <?xml version="1.0" encoding="utf-8"?>
00002 <PreferenceScreen xmlns:app="http://schemas.android.com/apk/res-auto">
00003     <Preference
00004         app:key="@string/preference_rules_general_key"
00005         app:title="@string/preference_rules_general_title" />
00006
00007     <Preference
00008         app:key="@string/preference_rules_course_key"
00009         app:title="@string/preference_rules_course_title" />
00010
00011     <Preference
00012         app:key="@string/preference_rules_factions_key"
00013         app:title="@string/preference_rules_factions_title" />
00014
00015     <Preference
00016         app:key="@string/preference_rules_commander_key"
00017         app:title="@string/preference_rules_commander_title" />
00018
00019     <Preference
00020         app:key="@string/preference_rules_cards_key"
00021         app:title="@string/preference_rules_cards_title" />
00022
00023     <Preference
00024         app:key="@string/preference_rules_card_abilities_key"
00025         app:title="@string/preference_rules_card_abilities_title" />
00026
00027     <Preference
00028         app:key="@string/preference_rules_special_cards_key"
00029         app:title="@string/preference_rules_special_cards_title" />
00030 </PreferenceScreen>
```

6.107 sound_preferences.xml

```
00001 <PreferenceScreen xmlns:app="http://schemas.android.com/apk/res-auto">
00002
00003     <SwitchPreference
00004         app:defaultValue="@bool/sound_preference_default"
00005         app:iconSpaceReserved="false"
00006         app:key="@string/preference_key_sounds_all"
00007         app:title="@string/preference_sounds_header" />
00008
00009     <PreferenceCategory
00010         app:allowDividerAbove="@bool/sound_preference_default"
00011         app:dependency="@string/preference_key_sounds_all"
00012         app:iconSpaceReserved="false"
00013         app:title="@string/preference_sounds_category">
00014
00015         <SwitchPreference
00016             app:defaultValue="@bool/sound_preference_default"
00017             app:iconSpaceReserved="false"
00018             app:key="@string/preference_key_sounds_cards"
00019             app:title="@string/preference_sounds_cards_new" />
00020
00021         <SwitchPreference
00022             app:defaultValue="@bool/sound_preference_default"
00023             app:iconSpaceReserved="false"
00024             app:key="@string/preference_key_sounds_reset"
00025             app:title="@string/preference_sounds_cards_delete" />
00026
00027         <SwitchPreference
00028             app:defaultValue="@bool/sound_preference_default"
00029             app:iconSpaceReserved="false"
00030             app:key="@string/preference_key_sounds_weather"
00031             app:title="@string/preference_sounds_weather" />
00032
00033         <SwitchPreference
00034             app:defaultValue="@bool/sound_preference_default"
00035             app:iconSpaceReserved="false"
00036             app:key="@string/preference_key_sounds_horn"
00037             app:title="@string/preference_sounds_horn" />
00038
00039         <SwitchPreference
00040             app:defaultValue="@bool/sound_preference_default"
00041             app:iconSpaceReserved="false"
00042             app:key="@string/preference_key_sounds_burn"
00043             app:title="@string/preference_sounds_burn" />
00044
00045         <SwitchPreference
00046             app:defaultValue="@bool/sound_preference_default"
00047             app:iconSpaceReserved="false"
00048             app:key="@string/preference_key_sounds_coin"
00049             app:title="@string/preference_sounds_coin" />
```

```
00050      </PreferenceCategory>
00051 </PreferenceScreen>
```

Index

/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/background_drawable.xml, 262
252 /home/runner/work/Gwent/Gwent/app/src/main/res/layout/fragment_introduction.xml, 256
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_copy.xml, 257
253 /home/runner/work/Gwent/Gwent/app/src/main/res/layout/popup_add_cards.xml, 264
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_delete.xml, 264
253 /home/runner/work/Gwent/Gwent/app/src/main/res/layout/popup_cards.xml, 264
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_desktop.xml, 265
253 /home/runner/work/Gwent/Gwent/app/src/main/res/layout/popup_coin_notification.xml, 265
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_explore.xml, 267
254 /home/runner/work/Gwent/Gwent/app/src/main/res/layout/popup_coin_statistics.xml, 267
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_factory_reset.xml, 267
254 /home/runner/work/Gwent/Gwent/app/src/main/res/layout/popup_coin_statistics.xml, 267
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_neutral.xml, 269
254 /home/runner/work/Gwent/Gwent/app/src/main/res/layout/popup_coin_statistics.xml, 269
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_rules.xml, 270
254 /home/runner/work/Gwent/Gwent/app/src/main/res/layout/popup_faction_selector.xml, 270
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_solid.xml, 270
254 /home/runner/work/Gwent/Gwent/app/src/main/res/layout/row.xml, 270
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/icon_wings.xml, 270
255 /home/runner/work/Gwent/Gwent/app/src/main/res/layout/res/mipmap-anydpi-v26/icon_launcher.xml, 273
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/indicator_select.xml, 273
255 /home/runner/work/Gwent/Gwent/app/src/main/res/layout/res/mipmap-anydpi-v26/icon_launcher_round.xml, 273
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/indicator_select.xml, 273
255 /home/runner/work/Gwent/Gwent/app/src/main/res/values-land/strings.xml, 275
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/rectangle_black.xml, 275
255 /home/runner/work/Gwent/Gwent/app/src/main/res/values/arrays.xml, 275
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/rectangle_black_transparent.xml, 275
255 /home/runner/work/Gwent/Gwent/app/src/main/res/values/attr.xml, 275
/home/runner/work/Gwent/Gwent/app/src/main/res/drawable/rectangle_cancel_popup.xml, 274
255 /home/runner/work/Gwent/Gwent/app/src/main/res/values/colors.xml, 274
/home/runner/work/Gwent/Gwent/app/src/main/res/layout-land-v26/fragment_introduction.xml, 256
274 /home/runner/work/Gwent/Gwent/app/src/main/res/values/dimens.xml, 274
/home/runner/work/Gwent/Gwent/app/src/main/res/layout-land/fragment_introduction.xml, 256
274 /home/runner/work/Gwent/Gwent/app/src/main/res/values/icon_launcher.xml, 275
/home/runner/work/Gwent/Gwent/app/src/main/res/layout-v26/activity_rules.xml, 258
275 /home/runner/work/Gwent/Gwent/app/src/main/res/values/preferences.xml, 275
/home/runner/work/Gwent/Gwent/app/src/main/res/layout-v26/fragment_introduction.xml, 257
275 /home/runner/work/Gwent/Gwent/app/src/main/res/values/strings.xml, 275
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/activity_introduction.xml, 259
259 /home/runner/work/Gwent/Gwent/app/src/main/res/values/styles.xml, 263
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/activity_main.xml, 263
260 /home/runner/work/Gwent/Gwent/app/src/main/res/xml/header_preferences.xml, 264
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/activity_rules.xml, 258
258 /home/runner/work/Gwent/Gwent/app/src/main/res/xml/rule_preferences.xml, 265
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/activity_settings.xml, 262
262 /home/runner/work/Gwent/Gwent/app/src/main/res/xml/sound_preferences.xml, 265
/home/runner/work/Gwent/Gwent/app/src/main/res/layout/alertdialog_checkbox.xml, 263
263 /home/runner/work/Gwent/Gwent/app/src/main/res/layout/alertdialog_checkbox.xml, 263
Ability, 13
BINDING, 13

- HORN, [13](#)
 - MORAL_BOOST, [14](#)
 - NONE, [14](#)
 - REVENGE, [14](#)
- ability
 - UnitEntity, [183](#)
- abilityImageld
 - CardUiState, [38](#)
- abilityPicker
 - CardNumberPickerAdapter, [32](#)
- abilityView
 - CardListAdapter.CardViewHolder, [45](#)
- accept
 - MenuUiStateObserver, [102](#)
 - RowUiStateObserver, [137](#)
- activated
 - Sound, [158](#)
- adapteeBuilder
 - ResetAlertDialogBuilderAdapter, [114](#)
 - RevengeAlertDialogBuilderAdapter, [123](#)
- AddCardDialog, [14](#)
 - AddCardDialog, [16](#)
 - disposables, [17](#)
 - onCreate, [16](#)
 - pickerAdapter, [17](#)
 - row, [17](#)
 - soundManager, [17](#)
- addSelectedUnits
 - CardNumberPickerAdapter, [32](#)
- animatedSwitch
 - ImageViewSwitchAnimator, [87](#)
- AppDatabase, [18](#)
 - rows, [19](#)
 - units, [19](#)
- ARG_SECTION_NUMBER
 - IntroductionFragment, [93](#)
- AVENGER_ABILITY
 - RevengeAlertDialogBuilderAdapter, [123](#)
- AVENGER_DAMAGE
 - RevengeAlertDialogBuilderAdapter, [123](#)
- AVENGER_EPIC
 - RevengeAlertDialogBuilderAdapter, [123](#)
- AVENGER_ROW
 - RevengeAlertDialogBuilderAdapter, [123](#)
- AVENGER_SQUAD
 - RevengeAlertDialogBuilderAdapter, [123](#)
- BACKGROUND
 - OverlayDialog, [109](#)
- ballViews
 - FactionSwitchListener, [70](#)
- BINDING
 - Ability, [13](#)
- bindingView
 - CardListAdapter.CardViewHolder, [45](#)
- BondDamageCalculatorDecorator, [19](#)
 - BondDamageCalculatorDecorator, [21](#)
 - calculateDamage, [21](#)
 - idToSquadSize, [22](#)
 - isBuffed, [22](#)
- BUFFED
 - DamageCalculator.Color, [52](#)
- burn
 - BurnDialogUseCase, [23](#)
 - MenuUiState, [100](#)
- burnButton
 - MenuUiStateObserver, [102](#)
- BurnDialogUseCase, [22](#)
 - burn, [23](#)
 - getBurnUnits, [24](#)
- BUTTON_CLICK
 - ResetDialogUseCase.Trigger, [173](#)
- calculateDamage
 - BondDamageCalculatorDecorator, [21](#)
 - DamageCalculator, [53](#)
 - HornDamageCalculatorDecorator, [85](#)
 - MoralDamageCalculatorDecorator, [105](#)
 - UnitEntity, [179](#)
 - WeatherDamageCalculator, [197](#)
- calculator
 - DamageCalculatorBuilder, [59](#)
- callback
 - ChangeFactionDialog, [49](#)
- CARD_ABILITIES
 - RuleSection, [141](#)
- CardListAdapter, [26](#)
 - CardListAdapter, [28](#)
 - getItemId, [28](#)
 - onBindViewHolder, [29](#)
 - onCopy, [29](#)
 - onCreateViewHolder, [29](#)
 - onRemove, [29](#)
- cardListAdapter
 - ShowUnitsDialog, [154](#)
- CardListAdapter.CardViewHolder, [42](#)
 - abilityView, [45](#)
 - bindingView, [45](#)
 - CardViewHolder, [44](#)
 - damageView, [45](#)
 - getAbilityView, [44](#)
 - getBindingView, [44](#)
 - getDamageView, [44](#)
 - itemId, [46](#)
 - setItemId, [45](#)
- CardNumberPickerAdapter, [30](#)
 - abilityPicker, [32](#)
 - addSelectedUnits, [32](#)
 - CardNumberPickerAdapter, [31](#)
 - damagePicker, [32](#)
 - EPIC_UNIT_ABILITIES, [33](#)
 - epicPicker, [33](#)
 - getDelayedOnValueChangeListener, [32](#)
 - numberPicker, [33](#)
 - squadManager, [33](#)
 - squadPicker, [33](#)
- CARDS
 - RuleSection, [141](#)

- CardUiState, 33
 - abilityImageId, 38
 - CardUiState, 35
 - damageBackgroundImageId, 38
 - damageString, 38
 - damageTextColor, 38
 - DIFF_CALLBACK, 38
 - equals, 35
 - getAbilityImageId, 36
 - getDamageBackgroundImageId, 36
 - getDamageString, 36
 - getDamageTextColor, 36
 - getSquadString, 37
 - getUnitId, 37
 - showAbility, 37
 - showSquad, 37
 - squadString, 39
 - unitId, 39
 - UNUSED, 39
- CardUiStateFactory, 40
 - CardUiStateFactory, 40
 - createCardUiState, 41
 - damageTextColors, 41
 - horn, 42
 - weather, 42
- CardViewHolder
 - CardListAdapter.CardViewHolder, 44
- cardViews
 - FactionSwitchListener, 71
- ChangeFactionDialog, 46
 - callback, 49
 - ChangeFactionDialog, 48
 - getOnThemeClickListener, 48
 - onCreate, 48
- ChangeFactionDialog.Callback, 25
 - onThemeSelect, 26
- changeListener
 - SoundManager, 163
- clearRows
 - RowDao, 125
- clearWeather
 - RowDao, 125
 - UnitRepository, 187
- CoinFlipDialog, 49
 - CoinFlipDialog, 51
 - onCreate, 51
 - RANDOM, 51
- collectionToString
 - UnitEntity, 179
- COMMANDER
 - RuleSection, 141
- component
 - DamageCalculatorDecorator, 61
- copy
 - UnitRepository, 187
- countUnits
 - UnitDao, 174
 - UnitRepository, 187
- COURSE
 - RuleSection, 141
- create
 - ResetAlertDialogBuilderAdapter, 113
 - RevengeAlertDialogBuilderAdapter, 121
- createCardUiState
 - CardUiStateFactory, 41
- createFragment
 - SectionsPagerAdapter, 143
- createSound
 - Sound, 156
- CURRENT_FRAGMENT_KEY
 - SettingsActivity, 146
- currentView
 - IndicatorManager, 89
- damage
 - MenuUiState, 100
 - RowUiState, 134
 - UnitEntity, 183
- damageBackgroundImageId
 - CardUiState, 38
- DamageCalculator, 52
 - calculateDamage, 53
 - isBuffed, 54
- DamageCalculator.Color, 51
 - BUFFED, 52
 - DEBUFFED, 52
 - DEFAULT, 52
- DamageCalculatorBuildDirector, 54
 - getCalculator, 55
 - setHorns, 56
 - setMoralBoosts, 56
 - setSquads, 56
- DamageCalculatorBuilder, 57
 - calculator, 59
 - getResult, 58
 - setBond, 58
 - setHorn, 58
 - setMoral, 59
 - setWeather, 59
- DamageCalculatorDecorator, 59
 - component, 61
 - DamageCalculatorDecorator, 61
- DamageCalculatorUseCase, 61
 - getDamageCalculator, 62
- damagePicker
 - CardNumberPickerAdapter, 32
- damageString
 - CardUiState, 38
- damageTextColor
 - CardUiState, 38
- damageTextColors
 - CardUiStateFactory, 41
- DamageValuePicker, 63
 - DamageValuePicker, 64
 - epicValues, 66
 - getDisplayString, 64
 - getValue, 64

- setEpicValues, 65
 - setSelectableValues, 65
 - setValue, 65
- damageView
 - CardListAdapter.CardViewHolder, 45
 - MenuUiStateObserver, 103
 - RowUiStateObserver, 138
- database
 - GwentApplication, 82
 - UnitRepository, 194
- DEBUFFED
 - DamageCalculator.Color, 52
- DEFAULT
 - DamageCalculator.Color, 52
 - ResetDialogUseCase.DialogType, 67
- delete
 - UnitRepository, 188
- deleteUnit
 - UnitDao, 174
- deleteUnits
 - UnitDao, 175
- dialogCallback
 - ResetAlertDialogBuilderAdapter, 114
- DIFF_CALLBACK
 - CardUiState, 38
- dismissViewId
 - OverlayDialog, 109
- disposables
 - AddCardDialog, 17
 - MainActivity, 97
 - ShowUnitsDialog, 154
- doubleDamage
 - HornDamageCalculatorDecorator, 85
- epic
 - UnitEntity, 184
- EPIC_DAMAGE_VALUES
 - UnitEntity, 184
- EPIC_UNIT_ABILITIES
 - CardNumberPickerAdapter, 33
- epicPicker
 - CardNumberPickerAdapter, 33
- epicValues
 - DamageValuePicker, 66
- equals
 - CardUiState, 35
 - MenuUiState, 99
 - RowUiState, 133
- FACTION_SWITCH
 - ResetDialogUseCase.Trigger, 173
- factionButton
 - FactionSwitchListener, 71
- FACTIONS
 - RuleSection, 141
- FactionSwitchListener, 67
 - ballViews, 70
 - cardViews, 71
 - factionButton, 71
 - FactionSwitchListener, 69
 - getListener, 69
 - onSharedPreferenceChanged, 70
 - setTheme, 70
 - THEME_MONSTER, 71
 - THEME_NILFGAARD, 71
 - THEME_NORTHERN_KINGDOMS, 71
 - THEME_PREFERENCE_KEY, 71
 - THEME_SCOIATAEL, 72
 - unitViews, 72
- factionSwitchListener
 - MainActivity, 97
- GameBoardViewModel, 72
 - GameBoardViewModel, 74
 - getMenuUiState, 74
 - getModel, 74
 - getRepository, 75
 - getRowUiState, 75
 - getSoundManager, 76
 - INITIALIZER, 79
 - menuUiState, 79
 - onBurnButtonPressed, 76
 - onFactionSwitchReset, 76
 - onHornViewPressed, 77
 - onResetButtonPressed, 77
 - onWeatherButtonPressed, 78
 - onWeatherViewPressed, 78
 - reset, 78
 - rowUiStates, 79
 - soundManager, 79
- gameBoardViewModel
 - MainActivity, 97
- GENERAL
 - RuleSection, 141
- getAbility
 - UnitEntity, 180
- getAbilityImageId
 - CardUiState, 36
- getAbilityView
 - CardListAdapter.CardViewHolder, 44
- getBindingView
 - CardListAdapter.CardViewHolder, 44
- getBurnUnits
 - BurnDialogUseCase, 24
- getCalculator
 - DamageCalculatorBuildDirector, 55
- getDamage
 - MenuUiState, 99
 - RowUiState, 133
 - UnitEntity, 180
- getDamageBackgroundImageId
 - CardUiState, 36
- getDamageCalculator
 - DamageCalculatorUseCase, 62
- getDamageString
 - CardUiState, 36
- getDamageTextColor
 - CardUiState, 36

- getDamageView
 - CardListAdapter.CardViewHolder, [44](#)
- getDelayedOnValueChangeListener
 - CardNumberPickerAdapter, [32](#)
- getDialog
 - ShowUnitsDialog, [153](#)
- getDialogType
 - ResetDialogUseCase, [116](#)
- getDisplayString
 - DamageValuePicker, [64](#)
- getFirstSquadWithMembers
 - SquadManager, [167](#)
- getId
 - RowEntity, [130](#)
 - UnitEntity, [180](#)
- getItemCount
 - SectionsPagerAdapter, [143](#)
- getItemId
 - CardListAdapter, [28](#)
- getListener
 - FactionSwitchListener, [69](#)
- getMemberBaseDamage
 - SquadState, [170](#)
- getMenuUiState
 - GameBoardViewModel, [74](#)
- getModel
 - GameBoardViewModel, [74](#)
- getObserver
 - RowUiStateObserver, [137](#)
- getOnThemeClickListener
 - ChangeFactionDialog, [48](#)
- getOverlayView
 - ImageViewSwitchAnimator, [87](#)
- getPreferenceKey
 - Sound, [157](#)
- getRandomUnit
 - ResetRepositoryUseCase, [118](#)
- getRepository
 - GameBoardViewModel, [75](#)
 - GwentApplication, [81](#)
 - UnitRepository, [188](#)
- getResult
 - DamageCalculatorBuilder, [58](#)
- getRevengeDialog
 - RemoveUnitsUseCase, [110](#)
 - ResetRepositoryUseCase, [118](#)
- getRow
 - ShowUnitsDialog, [153](#)
 - UnitEntity, [181](#)
- getRowUiState
 - GameBoardViewModel, [75](#)
- getSectionClickListener
 - SettingsRuleFragment, [148](#)
- getSoundId
 - Sound, [157](#)
- getSoundManager
 - GameBoardViewModel, [76](#)
- getSquad
 - UnitEntity, [181](#)
- getSquadMembers
 - SquadState, [170](#)
- getSquadNumber
 - SquadState, [170](#)
- getSquadString
 - CardUiState, [37](#)
- getState
 - SquadState, [170](#)
- getUnit
 - UnitDao, [175](#)
 - UnitRepository, [189](#)
- getUnitId
 - CardUiState, [37](#)
- getUnits
 - RowUiState, [134](#)
 - UnitDao, [175](#)
 - UnitRepository, [189](#)
- getUnitsFlowable
 - UnitDao, [176](#)
 - UnitRepository, [190](#)
- getValue
 - DamageValuePicker, [64](#)
- Gwent - A Calculator, [1](#)
- gwent/data/Ability.java, [199](#)
- gwent/data/AppDatabase.java, [200](#)
- gwent/data/RowDao.java, [200](#)
- gwent/data/RowEntity.java, [200](#)
- gwent/data/RowType.java, [201](#)
- gwent/data/UnitDao.java, [201](#)
- gwent/data/UnitEntity.java, [202](#)
- gwent/data/UnitRepository.java, [204](#)
- gwent/domain/cases/BurnDialogUseCase.java, [207](#)
- gwent/domain/cases/DamageCalculatorUseCase.java, [208](#)
- gwent/domain/cases/RemoveUnitsUseCase.java, [209](#)
- gwent/domain/cases/ResetAlertDialogBuilderAdapter.java, [210](#)
- gwent/domain/cases/ResetDialogUseCase.java, [210](#)
- gwent/domain/cases/ResetRepositoryUseCase.java, [212](#)
- gwent/domain/cases/RevengeAlertDialogBuilderAdapter.java, [213](#)
- gwent/domain/damage/BondDamageCalculatorDecorator.java, [214](#)
- gwent/domain/damage/DamageCalculator.java, [214](#)
- gwent/domain/damage/DamageCalculatorBuildDirector.java, [214](#)
- gwent/domain/damage/DamageCalculatorBuilder.java, [215](#)
- gwent/domain/damage/DamageCalculatorDecorator.java, [216](#)
- gwent/domain/damage/HornDamageCalculatorDecorator.java, [216](#)
- gwent/domain/damage/MoralDamageCalculatorDecorator.java, [216](#)
- gwent/domain/damage/WeatherDamageCalculator.java, [217](#)

- gwent/GwentApplication.java, 217
- gwent/ui/dialogs/addcard/AddCardDialog.java, 218
- gwent/ui/dialogs/addcard/CardNumberPickerAdapter.java, 219
- gwent/ui/dialogs/addcard/DamageValuePicker.java, 220
- gwent/ui/dialogs/addcard/SquadManager.java, 221
- gwent/ui/dialogs/addcard/SquadState.java, 222
- gwent/ui/dialogs/addcard/StringValuePicker.java, 223
- gwent/ui/dialogs/addcard/ValuePicker.java, 224
- gwent/ui/dialogs/cards/CardListAdapter.java, 225
- gwent/ui/dialogs/cards/CardUiState.java, 227
- gwent/ui/dialogs/cards/CardUiStateFactory.java, 228
- gwent/ui/dialogs/cards/ShowUnitsDialog.java, 229
- gwent/ui/dialogs/ChangeFactionDialog.java, 231
- gwent/ui/dialogs/CoinFlipDialog.java, 232
- gwent/ui/dialogs/OverlayDialog.java, 232
- gwent/ui/introduction/IndicatorManager.java, 233
- gwent/ui/introduction/IntroductionActivity.java, 233
- gwent/ui/introduction/IntroductionFragment.java, 234
- gwent/ui/introduction/SectionsPagerAdapter.java, 235
- gwent/ui/main/FactionSwitchListener.java, 235
- gwent/ui/main/GameBoardViewModel.java, 237
- gwent/ui/main/ImageViewSwitchAnimator.java, 239
- gwent/ui/main/MainActivity.java, 240
- gwent/ui/main/MenuUiState.java, 243
- gwent/ui/main/MenuUiStateObserver.java, 243
- gwent/ui/main/RowUiState.java, 244
- gwent/ui/main/RowUiStateObserver.java, 245
- gwent/ui/settings/RuleActivity.java, 246
- gwent/ui/settings/RuleSection.java, 247
- gwent/ui/settings/SettingsActivity.java, 247
- gwent/ui/settings/SettingsHeaderFragment.java, 248
- gwent/ui/settings/SettingsRuleFragment.java, 248
- gwent/ui/settings/SettingsSoundFragment.java, 249
- gwent/ui/sounds/Sound.java, 250
- gwent/ui/sounds/SoundManager.java, 250
- GwentApplication, 80
 - database, 82
 - getRepository, 81
 - onCreate, 81
 - repository, 82
- hasMembers
 - SquadState, 171
- hasNonEpicUnitsFlowable
 - UnitDao, 176
 - UnitRepository, 190
- HORN
 - Ability, 13
- horn
 - CardUiStateFactory, 42
 - RowEntity, 131
 - RowUiState, 134
- HornDamageCalculatorDecorator, 82
 - calculateDamage, 85
 - doubleDamage, 85
 - HornDamageCalculatorDecorator, 84
 - isBuffered, 85
 - unitIds, 86
- hornView
 - RowUiStateObserver, 138
- id
 - RowEntity, 131
 - UnitEntity, 184
- idToSquadSize
 - BondDamageCalculatorDecorator, 22
- imageIds
 - IntroductionFragment, 94
- ImageViewSwitchAnimator, 86
 - animatedSwitch, 87
 - getOverlayView, 87
- IndicatorManager, 88
 - currentView, 89
 - IndicatorManager, 88
 - indicators, 89
 - updateIndicators, 89
- indicatorManager
 - IntroductionActivity, 91
- indicators
 - IndicatorManager, 89
- inflateCoinFlipPopup
 - MainActivity, 96
- inflateFactionPopup
 - MainActivity, 96
- INITIALIZER
 - GameBoardViewModel, 79
- initializeRows
 - UnitRepository, 190
- initializeViewModel
 - MainActivity, 96
- insertAvengers
 - RevengeAlertDialogBuilderAdapter, 121
- insertRow
 - RowDao, 125
- insertUnit
 - UnitDao, 176
 - UnitRepository, 190, 191
- INTENT_EXTRA_TAG
 - RuleActivity, 140
- IntroductionActivity, 90
 - indicatorManager, 91
 - onCreate, 91
- IntroductionFragment, 91
 - ARG_SECTION_NUMBER, 93
 - imageIds, 94
 - newInstance, 93
 - onCreateView, 93
 - PAGES_COUNT, 94
 - stringIds, 94
- isActivated
 - Sound, 157
- isBuffered
 - BondDamageCalculatorDecorator, 22
 - DamageCalculator, 54
 - HornDamageCalculatorDecorator, 85
 - MoralDamageCalculatorDecorator, 106
 - UnitEntity, 181

- WeatherDamageCalculator, 197
- isBurn
 - MenuUiState, 99
- isEpic
 - UnitEntity, 181
- isHorn
 - RowDao, 125
 - RowEntity, 130
 - RowUiState, 134
 - UnitRepository, 192
- isHornFlowable
 - RowDao, 126
 - UnitRepository, 192
- isReset
 - MenuUiState, 99
- isWeather
 - MenuUiState, 100
 - RowDao, 126
 - RowEntity, 130
 - RowUiState, 134
 - UnitRepository, 192
- isWeatherFlowable
 - RowDao, 126
 - UnitRepository, 193
- itemId
 - CardListAdapter.CardViewHolder, 46
- layout
 - OverlayDialog, 109
- MainActivity, 94
 - disposables, 97
 - factionSwitchListener, 97
 - gameBoardViewModel, 97
 - inflateCoinFlipPopup, 96
 - inflateFactionPopup, 96
 - initializeViewModel, 96
 - onCreate, 96
 - onDestroy, 97
 - onResume, 97
 - onWindowFocusChanged, 97
- MAX_NR_SQUADS
 - SquadManager, 168
- MELEE
 - RowType, 132
- memberBaseDamage
 - SquadState, 171
- MenuUiState, 98
 - burn, 100
 - damage, 100
 - equals, 99
 - getDamage, 99
 - isBurn, 99
 - isReset, 99
 - isWeather, 100
 - MenuUiState, 99
 - reset, 100
 - weather, 100
- menuUiState
 - GameBoardViewModel, 79
- MenuUiStateObserver, 100
 - accept, 102
 - burnButton, 102
 - damageView, 103
 - MenuUiStateObserver, 102
 - resetButton, 103
 - weatherButton, 103
- MONSTER
 - ResetDialogUseCase.DialogType, 67
- MORAL_BOOST
 - Ability, 14
- MoralDamageCalculatorDecorator, 103
 - calculateDamage, 105
 - isBuffered, 106
 - MoralDamageCalculatorDecorator, 105
 - unitIds, 106
- newInstance
 - IntroductionFragment, 93
- NO_DISMISS_VIEW
 - OverlayDialog, 109
- NON_EPIC_DAMAGE_VALUES_UPPER_BOUND
 - UnitEntity, 184
- NONE
 - Ability, 14
 - ResetDialogUseCase.DialogType, 67
- numberPicker
 - CardNumberPickerAdapter, 33
- onBindViewHolder
 - CardListAdapter, 29
- onBurnButtonPressed
 - GameBoardViewModel, 76
- onCopy
 - CardListAdapter, 29
- onCreate
 - AddCardDialog, 16
 - ChangeFactionDialog, 48
 - CoinFlipDialog, 51
 - GwentApplication, 81
 - IntroductionActivity, 91
 - MainActivity, 96
 - OverlayDialog, 109
 - RuleActivity, 139
 - SettingsActivity, 145
 - ShowUnitsDialog, 153
- onCreatePreferences
 - SettingsHeaderFragment, 147
 - SettingsRuleFragment, 149
 - SettingsSoundFragment, 150
- onCreateView
 - IntroductionFragment, 93
- onCreateViewHolder
 - CardListAdapter, 29
- onDestroy
 - MainActivity, 97
- onFactionSwitchReset
 - GameBoardViewModel, 76

- onHornViewPressed
 - GameBoardViewModel, 77
- onOptionsItemSelected
 - RuleActivity, 140
 - SettingsActivity, 145
- onPreferenceStartFragment
 - SettingsActivity, 145
- onRemove
 - CardListAdapter, 29
- onResetButtonPressed
 - GameBoardViewModel, 77
- onResume
 - MainActivity, 97
- onSaveInstanceState
 - SettingsActivity, 145
- onSharedPreferencesChanged
 - FactionSwitchListener, 70
- onSquadChanged
 - SquadManager, 167
- onThemeSelect
 - ChangeFactionDialog.Callback, 26
- OnValueChangeListener< T extends Comparable, 106
- onWeatherButtonPressed
 - GameBoardViewModel, 78
- onWeatherViewPressed
 - GameBoardViewModel, 78
- onWindowFocusChanged
 - MainActivity, 97
- OverlayDialog, 107
 - BACKGROUND, 109
 - dismissViewId, 109
 - layout, 109
 - NO_DISMISS_VIEW, 109
 - onCreate, 109
 - OverlayDialog, 108
- PAGES_COUNT
 - IntroductionFragment, 94
- pickerAdapter
 - AddCardDialog, 17
- playBurnSound
 - SoundManager, 161
- playCardAddSound
 - SoundManager, 161
- playCardRemovedSound
 - SoundManager, 161
- playClearWeatherSound
 - SoundManager, 161
- playCoinSound
 - SoundManager, 161
- playHornSound
 - SoundManager, 162
- playResetSound
 - SoundManager, 162
- playSound
 - SoundManager, 162
- playWeatherSound
 - SoundManager, 163
- preferenceKey
 - Sound, 158
- RANDOM
 - CoinFlipDialog, 51
- RANGE
 - RowType, 132
- remove
 - RemoveUnitsUseCase, 111, 112
- RemoveUnitsUseCase, 110
 - getRevengeDialog, 110
 - remove, 111, 112
- repository
 - GwentApplication, 82
- reset
 - GameBoardViewModel, 78
 - MenuUiState, 100
 - ResetAlertDialogBuilderAdapter.Callback, 25
 - ResetDialogUseCase, 116
 - ResetRepositoryUseCase, 119
 - UnitRepository, 193
- ResetAlertDialogBuilderAdapter, 112
 - adapteeBuilder, 114
 - create, 113
 - dialogCallback, 114
 - ResetAlertDialogBuilderAdapter, 113
 - setMonsterDialog, 114
 - setTrigger, 114
- ResetAlertDialogBuilderAdapter.Callback, 24
 - reset, 25
- resetButton
 - MenuUiStateObserver, 103
- ResetDialogUseCase, 115
 - getDialogType, 116
 - reset, 116
- ResetDialogUseCase.DialogType, 66
 - DEFAULT, 67
 - MONSTER, 67
 - NONE, 67
- ResetDialogUseCase.Trigger, 172
 - BUTTON_CLICK, 173
 - FACTION_SWITCH, 173
- ResetRepositoryUseCase, 117
 - getRandomUnit, 118
 - getRevengeDialog, 118
 - reset, 119
- REVENGE
 - Ability, 14
- RevengeAlertDialogBuilderAdapter, 120
 - adapteeBuilder, 123
 - AVENGER_ABILITY, 123
 - AVENGER_DAMAGE, 123
 - AVENGER_EPIC, 123
 - AVENGER_ROW, 123
 - AVENGER_SQUAD, 123
 - create, 121
 - insertAvengers, 121
 - RevengeAlertDialogBuilderAdapter, 121
 - setNegativeCallback, 122
 - setPositiveCallback, 122

- row
 - AddCardDialog, 17
 - ShowUnitsDialog, 154
 - UnitEntity, 184
- RowDao, 124
 - clearRows, 125
 - clearWeather, 125
 - insertRow, 125
 - isHorn, 125
 - isHornFlowable, 126
 - isWeather, 126
 - isWeatherFlowable, 126
 - updateHorn, 127
 - updateWeather, 127
- RowEntity, 127
 - getId, 130
 - horn, 131
 - id, 131
 - isHorn, 130
 - isWeather, 130
 - RowEntity, 128
 - setHorn, 130
 - setWeather, 130
 - weather, 131
- rows
 - AppDatabase, 19
- RowType, 131
 - MELEE, 132
 - RANGE, 132
 - SIEGE, 132
- RowUiState, 132
 - damage, 134
 - equals, 133
 - getDamage, 133
 - getUnits, 134
 - horn, 134
 - isHorn, 134
 - isWeather, 134
 - RowUiState, 133
 - units, 134
 - weather, 135
- RowUiStateObserver, 135
 - accept, 137
 - damageView, 138
 - getObserver, 137
 - hornView, 138
 - RowUiStateObserver, 136
 - unitView, 138
 - weatherResource, 138
 - weatherView, 138
- rowUiStates
 - GameBoardViewModel, 79
- RuleActivity, 138
 - INTENT_EXTRA_TAG, 140
 - onCreate, 139
 - onOptionsItemSelected, 140
- RuleSection, 140
 - CARD_ABILITIES, 141
 - CARDS, 141
 - COMMANDER, 141
 - COURSE, 141
 - FACTIONS, 141
 - GENERAL, 141
 - SPECIAL_CARDS, 141
- SectionsPagerAdapter, 142
 - createFragment, 143
 - getItemCount, 143
 - SectionsPagerAdapter, 143
- setAbility
 - UnitEntity, 181
- setActivated
 - Sound, 157, 158
- setBond
 - DamageCalculatorBuilder, 58
- setDamage
 - UnitEntity, 182
- setEpic
 - UnitEntity, 182
- setEpicValues
 - DamageValuePicker, 65
- setHorn
 - DamageCalculatorBuilder, 58
 - RowEntity, 130
- setHorns
 - DamageCalculatorBuildDirector, 56
- setId
 - UnitEntity, 182
- setItemId
 - CardListAdapter.CardViewHolder, 45
- setMonsterDialog
 - ResetAlertDialogBuilderAdapter, 114
- setMoral
 - DamageCalculatorBuilder, 59
- setMoralBoosts
 - DamageCalculatorBuildDirector, 56
- setNegativeCallback
 - RevengeAlertDialogBuilderAdapter, 122
- setPositiveCallback
 - RevengeAlertDialogBuilderAdapter, 122
- setRow
 - UnitEntity, 182
- setSelectableValues
 - DamageValuePicker, 65
- setSquad
 - UnitEntity, 183
- setSquads
 - DamageCalculatorBuildDirector, 56
- setTheme
 - FactionSwitchListener, 70
- SettingsActivity, 143
 - CURRENT_FRAGMENT_KEY, 146
 - onCreate, 145
 - onOptionsItemSelected, 145
 - onPreferenceStartFragment, 145
 - onSaveInstanceState, 145
- SettingsHeaderFragment, 146

- onCreatePreferences, 147
- SettingsRuleFragment, 147
 - getSectionClickListener, 148
 - onCreatePreferences, 149
- SettingsSoundFragment, 149
 - onCreatePreferences, 150
- setTrigger
 - ResetAlertDialogBuilderAdapter, 114
- setValue
 - DamageValuePicker, 65
- setWeather
 - DamageCalculatorBuilder, 59
 - RowEntity, 130
- showAbility
 - CardUiState, 37
- showSquad
 - CardUiState, 37
- ShowUnitsDialog, 151
 - cardListAdapter, 154
 - disposables, 154
 - getDialog, 153
 - getRow, 153
 - onCreate, 153
 - row, 154
 - ShowUnitsDialog, 153
 - soundManager, 155
 - unitRecyclerView, 155
- SIEGE
 - RowType, 132
- Sound, 155
 - activated, 158
 - createSound, 156
 - getPreferenceKey, 157
 - getSoundId, 157
 - isActive, 157
 - preferenceKey, 158
 - setActivated, 157, 158
 - Sound, 156
 - soundId, 158
- SOUND_BURN
 - SoundManager, 163
- SOUND_CARDS_EPIC
 - SoundManager, 163
- SOUND_CARDS_MELEE
 - SoundManager, 163
- SOUND_CARDS_RANGE
 - SoundManager, 164
- SOUND_CARDS_SIEGE
 - SoundManager, 164
- SOUND_COIN
 - SoundManager, 164
- SOUND_HORN
 - SoundManager, 164
- SOUND_RESET
 - SoundManager, 164
- SOUND_WEATHER_FOG
 - SoundManager, 165
- SOUND_WEATHER_FROST
 - SoundManager, 165
- SOUND_WEATHER_GOOD
 - SoundManager, 165
- SOUND_WEATHER_RAIN
 - SoundManager, 165
- soundId
 - Sound, 158
- SoundManager, 158
 - changeListener, 163
 - playBurnSound, 161
 - playCardAddSound, 161
 - playCardRemovedSound, 161
 - playClearWeatherSound, 161
 - playCoinSound, 161
 - playHornSound, 162
 - playResetSound, 162
 - playSound, 162
 - playWeatherSound, 163
 - SOUND_BURN, 163
 - SOUND_CARDS_EPIC, 163
 - SOUND_CARDS_MELEE, 163
 - SOUND_CARDS_RANGE, 164
 - SOUND_CARDS_SIEGE, 164
 - SOUND_COIN, 164
 - SOUND_HORN, 164
 - SOUND_RESET, 164
 - SOUND_WEATHER_FOG, 165
 - SOUND_WEATHER_FROST, 165
 - SOUND_WEATHER_GOOD, 165
 - SOUND_WEATHER_RAIN, 165
 - SoundManager, 160
 - soundPool, 165
 - sounds, 166
- soundManager
 - AddCardDialog, 17
 - GameBoardViewModel, 79
 - ShowUnitsDialog, 155
- soundPool
 - SoundManager, 165
- sounds
 - SoundManager, 166
- SPECIAL_CARDS
 - RuleSection, 141
- squad
 - UnitEntity, 184
- SquadManager, 166
 - getFirstSquadWithMembers, 167
 - MAX_NR_SQUADS, 168
 - onSquadChanged, 167
 - SquadManager, 167
 - states, 168
- squadManager
 - CardNumberPickerAdapter, 33
- squadMembers
 - SquadState, 171
- squadNumber
 - SquadState, 171
- squadPicker

- CardNumberPickerAdapter, 33
- SquadState, 168
 - getMemberBaseDamage, 170
 - getSquadMembers, 170
 - getSquadNumber, 170
 - getState, 170
 - hasMembers, 171
 - memberBaseDamage, 171
 - squadMembers, 171
 - squadNumber, 171
 - SquadState, 169
- squadString
 - CardUiState, 39
- states
 - SquadManager, 168
- stringIds
 - IntroductionFragment, 94
- StringValuePicker< T extends Comparable, 172
- switchHorn
 - UnitRepository, 194
- switchWeather
 - UnitRepository, 194
- THEME_MONSTER
 - FactionSwitchListener, 71
- THEME_NILFGAARD
 - FactionSwitchListener, 71
- THEME_NORTHERN_KINGDOMS
 - FactionSwitchListener, 71
- THEME_PREFERENCE_KEY
 - FactionSwitchListener, 71
- THEME_SCOIATAEL
 - FactionSwitchListener, 72
- toString
 - UnitEntity, 183
- UnitDao, 173
 - countUnits, 174
 - deleteUnit, 174
 - deleteUnits, 175
 - getUnit, 175
 - getUnits, 175
 - getUnitsFlowable, 176
 - hasNonEpicUnitsFlowable, 176
 - insertUnit, 176
- UnitEntity, 177
 - ability, 183
 - calculateDamage, 179
 - collectionToString, 179
 - damage, 183
 - epic, 184
 - EPIC_DAMAGE_VALUES, 184
 - getAbility, 180
 - getDamage, 180
 - getId, 180
 - getRow, 181
 - getSquad, 181
 - id, 184
 - isBuffed, 181
 - isEpic, 181
 - NON_EPIC_DAMAGE_VALUES_UPPER_BOUND, 184
 - row, 184
 - setAbility, 181
 - setDamage, 182
 - setEpic, 182
 - setId, 182
 - setRow, 182
 - setSquad, 183
 - squad, 184
 - toString, 183
 - UnitEntity, 179
- unitId
 - CardUiState, 39
- unitIds
 - HornDamageCalculatorDecorator, 86
 - MoralDamageCalculatorDecorator, 106
- unitRecyclerView
 - ShowUnitsDialog, 155
- UnitRepository, 184
 - clearWeather, 187
 - copy, 187
 - countUnits, 187
 - database, 194
 - delete, 188
 - getRepository, 188
 - getUnit, 189
 - getUnits, 189
 - getUnitsFlowable, 190
 - hasNonEpicUnitsFlowable, 190
 - initializeRows, 190
 - insertUnit, 190, 191
 - isHorn, 192
 - isHornFlowable, 192
 - isWeather, 192
 - isWeatherFlowable, 193
 - reset, 193
 - switchHorn, 194
 - switchWeather, 194
 - UnitRepository, 186
- units
 - AppDatabase, 19
 - RowUiState, 134
- unitView
 - RowUiStateObserver, 138
- unitViews
 - FactionSwitchListener, 72
- UNUSED
 - CardUiState, 39
- updateHorn
 - RowDao, 127
- updateIndicators
 - IndicatorManager, 89
- updateWeather
 - RowDao, 127
- ValuePicker< T extends Comparable, 195

- weather
 - CardUiStateFactory, [42](#)
 - MenuUiState, [100](#)
 - RowEntity, [131](#)
 - RowUiState, [135](#)
 - WeatherDamageCalculator, [198](#)
- weatherButton
 - MenuUiStateObserver, [103](#)
- WeatherDamageCalculator, [195](#)
 - calculateDamage, [197](#)
 - isBuffered, [197](#)
 - weather, [198](#)
 - WeatherDamageCalculator, [197](#)
- weatherResource
 - RowUiStateObserver, [138](#)
- weatherView
 - RowUiStateObserver, [138](#)