

Gwent

2.0.0

Generated by Doxygen 1.9.1



<b>1 Gwent - A Calculator</b>	<b>1</b>
1.1 Overview	1
1.2 Printing analog cards	1
1.3 Documentation	1
<b>2 Hierarchical Index</b>	<b>3</b>
2.1 Class Hierarchy	3
<b>3 Class Index</b>	<b>5</b>
3.1 Class List	5
<b>4 Class Documentation</b>	<b>9</b>
4.1 Ability Enum Reference	9
4.1.1 Detailed Description	9
4.1.2 Member Data Documentation	9
4.1.2.1 BINDING	9
4.1.2.2 HORN	10
4.1.2.3 MORAL_BOOST	10
4.1.2.4 NONE	10
4.1.2.5 REVENGE	10
4.2 AddCardDialog Class Reference	10
4.2.1 Detailed Description	12
4.2.2 Constructor & Destructor Documentation	12
4.2.2.1 AddCardDialog() [1/2]	12
4.2.2.2 AddCardDialog() [2/2]	12
4.2.3 Member Function Documentation	12
4.2.3.1 onCreate()	12
4.2.4 Member Data Documentation	13
4.2.4.1 disposables	13
4.2.4.2 pickerAdapter	13
4.2.4.3 row	13
4.3 AppDatabase Class Reference	14
4.3.1 Detailed Description	14
4.3.2 Member Function Documentation	14
4.3.2.1 rows()	14
4.3.2.2 units()	15
4.4 BondDamageCalculatorDecorator Class Reference	15
4.4.1 Detailed Description	16
4.4.2 Constructor & Destructor Documentation	16
4.4.2.1 BondDamageCalculatorDecorator()	16
4.4.3 Member Function Documentation	17
4.4.3.1 calculateDamage()	17
4.4.3.2 isBuffed()	18

4.4.4 Member Data Documentation	18
4.4.4.1 idToSquadSize	18
4.5 BurnDialogUseCase Class Reference	18
4.5.1 Detailed Description	19
4.5.2 Member Function Documentation	19
4.5.2.1 burn() [1/2]	19
4.5.2.2 burn() [2/2]	19
4.5.2.3 getBurnUnits()	20
4.6 ResetAlertDialogBuilderAdapter.Callback Interface Reference	20
4.6.1 Detailed Description	20
4.6.2 Member Function Documentation	21
4.6.2.1 reset() [1/2]	21
4.6.2.2 reset() [2/2]	21
4.7 ChangeFactionDialog.Callback Interface Reference	21
4.7.1 Detailed Description	22
4.7.2 Member Function Documentation	22
4.7.2.1 onThemeSelect()	22
4.8 CardListAdapter Class Reference	22
4.8.1 Detailed Description	24
4.8.2 Constructor & Destructor Documentation	24
4.8.2.1 CardListAdapter()	24
4.8.3 Member Function Documentation	24
4.8.3.1 getItemId()	25
4.8.3.2 onBindViewHolder()	25
4.8.3.3 onCreateViewHolder()	25
4.8.4 Member Data Documentation	25
4.8.4.1 onCopy	26
4.8.4.2 onRemove	26
4.9 CardNumberPickerAdapter Class Reference	26
4.9.1 Detailed Description	27
4.9.2 Constructor & Destructor Documentation	27
4.9.2.1 CardNumberPickerAdapter()	27
4.9.3 Member Function Documentation	28
4.9.3.1 addSelectedUnits()	28
4.9.3.2 getDelayedOnValueChangeListener()	28
4.9.4 Member Data Documentation	29
4.9.4.1 abilityPicker	29
4.9.4.2 damagePicker	29
4.9.4.3 EPIC_UNIT_ABILITIES	29
4.9.4.4 epicPicker	29
4.9.4.5 numberPicker	29
4.9.4.6 squadPicker	29

4.10 CardUiState Class Reference	30
4.10.1 Detailed Description	31
4.10.2 Constructor & Destructor Documentation	31
4.10.2.1 CardUiState()	31
4.10.3 Member Function Documentation	32
4.10.3.1 equals()	32
4.10.3.2 getAbilityImageld()	32
4.10.3.3 getDamageBackgroundImageld()	32
4.10.3.4 getDamageString()	33
4.10.3.5 getDamageTextColor()	33
4.10.3.6 getSquadString()	33
4.10.3.7 getUnitId()	34
4.10.3.8 showAbility()	34
4.10.3.9 showSquad()	34
4.10.4 Member Data Documentation	34
4.10.4.1 abilityImageld	34
4.10.4.2 damageBackgroundImageld	35
4.10.4.3 damageString	35
4.10.4.4 damageTextColor	35
4.10.4.5 DIFF_CALLBACK	35
4.10.4.6 squadString	36
4.10.4.7 unitId	36
4.10.4.8 UNUSED	36
4.11 CardUiStateFactory Class Reference	36
4.11.1 Detailed Description	37
4.11.2 Constructor & Destructor Documentation	37
4.11.2.1 CardUiStateFactory()	38
4.11.3 Member Function Documentation	38
4.11.3.1 createCardUiState() [1/2]	38
4.11.3.2 createCardUiState() [2/2]	38
4.11.4 Member Data Documentation	39
4.11.4.1 damageTextColors	39
4.11.4.2 horn	39
4.11.4.3 weather	39
4.12 CardListAdapter.CardViewHolder Class Reference	39
4.12.1 Detailed Description	41
4.12.2 Constructor & Destructor Documentation	41
4.12.2.1 CardViewHolder()	41
4.12.3 Member Function Documentation	41
4.12.3.1 getAbilityView()	41
4.12.3.2 getBindingView()	42
4.12.3.3 getDamageView()	42

4.12.3.4 setItemId()	42
4.12.4 Member Data Documentation	43
4.12.4.1 abilityView	43
4.12.4.2 bindingView	43
4.12.4.3 damageView	43
4.12.4.4 itemId	43
4.13 ChangeFactionDialog Class Reference	44
4.13.1 Detailed Description	45
4.13.2 Constructor & Destructor Documentation	45
4.13.2.1 ChangeFactionDialog()	45
4.13.3 Member Function Documentation	45
4.13.3.1 getOnThemeClickListener()	46
4.13.3.2 onCreate()	46
4.13.4 Member Data Documentation	46
4.13.4.1 callback	46
4.14 CoinFlipDialog Class Reference	46
4.14.1 Detailed Description	48
4.14.2 Constructor & Destructor Documentation	48
4.14.2.1 CoinFlipDialog()	48
4.14.3 Member Function Documentation	48
4.14.3.1 onCreate()	48
4.14.4 Member Data Documentation	48
4.14.4.1 RANDOM	49
4.15 DamageCalculator.Color Enum Reference	49
4.15.1 Detailed Description	49
4.15.2 Member Data Documentation	49
4.15.2.1 BUFFED	49
4.15.2.2 DEBUFFED	49
4.15.2.3 DEFAULT	50
4.16 DamageCalculator Interface Reference	50
4.16.1 Detailed Description	50
4.16.2 Member Function Documentation	51
4.16.2.1 calculateDamage()	51
4.16.2.2 isBuffed()	51
4.17 DamageCalculatorBuildDirector Class Reference	52
4.17.1 Detailed Description	52
4.17.2 Member Function Documentation	53
4.17.2.1 getCalculator()	53
4.17.2.2 setHorns()	53
4.17.2.3 setMoralBoosts()	53
4.17.2.4 setSquads()	54
4.18 DamageCalculatorBuilder Class Reference	54

4.18.1 Detailed Description	55
4.18.2 Member Function Documentation	55
4.18.2.1 getResult()	56
4.18.2.2 setBond()	56
4.18.2.3 setHorn()	56
4.18.2.4 setMoral()	56
4.18.2.5 setWeather()	57
4.18.3 Member Data Documentation	57
4.18.3.1 calculator	57
4.19 DamageCalculatorDecorator Class Reference	57
4.19.1 Detailed Description	58
4.19.2 Constructor & Destructor Documentation	58
4.19.2.1 DamageCalculatorDecorator()	58
4.19.3 Member Data Documentation	58
4.19.3.1 component	59
4.20 DamageCalculatorUseCase Class Reference	59
4.20.1 Detailed Description	59
4.20.2 Member Function Documentation	59
4.20.2.1 getDamageCalculator() [1/2]	60
4.20.2.2 getDamageCalculator() [2/2]	60
4.21 DamageValuePicker Class Reference	60
4.21.1 Detailed Description	62
4.21.2 Constructor & Destructor Documentation	62
4.21.2.1 DamageValuePicker()	62
4.21.3 Member Function Documentation	62
4.21.3.1 getDisplayString()	62
4.21.3.2 getValue()	63
4.21.3.3 setEpicValues()	63
4.21.3.4 setSelectableValues()	63
4.21.3.5 setValue()	64
4.21.4 Member Data Documentation	64
4.21.4.1 EPIC_DAMAGE_VALUES	64
4.21.4.2 epicValues	64
4.21.4.3 NON_EPIC_DAMAGE_VALUES_UPPER_BOUND	64
4.22 ResetDialogUseCase.DialogType Enum Reference	65
4.22.1 Detailed Description	65
4.22.2 Member Data Documentation	65
4.22.2.1 DEFAULT	65
4.22.2.2 MONSTER	65
4.22.2.3 NONE	66
4.23 FactionSwitchListener Class Reference	66
4.23.1 Detailed Description	67

4.23.2 Constructor & Destructor Documentation	67
4.23.2.1 FactionSwitchListener()	68
4.23.3 Member Function Documentation	68
4.23.3.1 getListener()	68
4.23.3.2 onSharedPreferenceChanged()	68
4.23.3.3 setTheme()	69
4.23.4 Member Data Documentation	69
4.23.4.1 ballViews	69
4.23.4.2 cardViews	70
4.23.4.3 factionButton	70
4.23.4.4 THEME_MONSTER	70
4.23.4.5 THEME_NILFGAARD	70
4.23.4.6 THEME_NORTHERN_KINGDOMS	70
4.23.4.7 THEME_PREFERENCE_KEY	71
4.23.4.8 THEME_SCOIATAEL	71
4.23.4.9 unitViews	71
4.24 GameBoardViewModel Class Reference	71
4.24.1 Detailed Description	73
4.24.2 Constructor & Destructor Documentation	73
4.24.2.1 GameBoardViewModel()	73
4.24.3 Member Function Documentation	73
4.24.3.1 getMenuUiState()	73
4.24.3.2 getModel()	73
4.24.3.3 getRepository()	74
4.24.3.4 getRowUiState()	74
4.24.3.5 onBurnButtonPressed()	75
4.24.3.6 onFactionSwitchReset()	75
4.24.3.7 onHornViewPressed()	75
4.24.3.8 onResetButtonPressed()	76
4.24.3.9 onWeatherButtonPressed()	76
4.24.3.10 onWeatherViewPressed()	76
4.24.3.11 reset()	77
4.24.4 Member Data Documentation	77
4.24.4.1 INITIALIZER	77
4.24.4.2 menuUiState	78
4.24.4.3 rowUiStates	78
4.25 GwentApplication Class Reference	78
4.25.1 Detailed Description	79
4.25.2 Member Function Documentation	80
4.25.2.1 getRepository() [1/2]	80
4.25.2.2 getRepository() [2/2]	80
4.25.2.3 onCreate()	80



4.25.3 Member Data Documentation	80
4.25.3.1 database	81
4.25.3.2 repository	81
4.26 HornDamageCalculatorDecorator Class Reference	81
4.26.1 Detailed Description	82
4.26.2 Constructor & Destructor Documentation	83
4.26.2.1 HornDamageCalculatorDecorator()	83
4.26.3 Member Function Documentation	83
4.26.3.1 calculateDamage()	83
4.26.3.2 doubleDamage()	84
4.26.3.3 isBuffered()	84
4.26.4 Member Data Documentation	84
4.26.4.1 unitIds	84
4.27 ImageViewSwitchAnimator Class Reference	85
4.27.1 Detailed Description	85
4.27.2 Member Function Documentation	85
4.27.2.1 animatedSwitch()	85
4.27.2.2 getOverlayView()	86
4.28 IndicatorManager Class Reference	86
4.28.1 Detailed Description	87
4.28.2 Constructor & Destructor Documentation	87
4.28.2.1 IndicatorManager()	87
4.28.3 Member Function Documentation	87
4.28.3.1 updateIndicators()	87
4.28.4 Member Data Documentation	89
4.28.4.1 currentView	89
4.28.4.2 indicators	89
4.29 IntroductionActivity Class Reference	89
4.29.1 Detailed Description	90
4.29.2 Member Function Documentation	90
4.29.2.1 onCreate()	90
4.29.3 Member Data Documentation	91
4.29.3.1 indicatorManager	91
4.30 IntroductionFragment Class Reference	91
4.30.1 Detailed Description	92
4.30.2 Member Function Documentation	92
4.30.2.1 newInstance()	92
4.30.2.2 onCreateView()	93
4.30.3 Member Data Documentation	93
4.30.3.1 ARG_SECTION_NUMBER	93
4.30.3.2 imageIds	93
4.30.3.3 PAGES_COUNT	94

4.30.3.4 stringIds	94
4.31 MainActivity Class Reference	94
4.31.1 Detailed Description	95
4.31.2 Member Function Documentation	95
4.31.2.1 inflateCoinFlipPopup()	95
4.31.2.2 inflateFactionPopup()	96
4.31.2.3 initializeViewModel()	96
4.31.2.4 onCreate()	96
4.31.2.5 onDestroy()	96
4.31.2.6 onResume()	97
4.31.2.7 onWindowFocusChanged()	97
4.31.3 Member Data Documentation	97
4.31.3.1 disposables	97
4.31.3.2 factionSwitchListener	97
4.31.3.3 gameBoardViewModel	97
4.31.3.4 soundManager	97
4.32 MenuUiState Class Reference	98
4.32.1 Detailed Description	98
4.32.2 Constructor & Destructor Documentation	98
4.32.2.1 MenuUiState()	99
4.32.3 Member Function Documentation	99
4.32.3.1 getDamage()	99
4.32.3.2 isBurn()	99
4.32.3.3 isReset()	99
4.32.3.4 isWeather()	100
4.32.4 Member Data Documentation	100
4.32.4.1 burn	100
4.32.4.2 damage	100
4.32.4.3 reset	100
4.32.4.4 weather	100
4.33 MenuUiStateObserver Class Reference	101
4.33.1 Detailed Description	102
4.33.2 Constructor & Destructor Documentation	102
4.33.2.1 MenuUiStateObserver()	102
4.33.3 Member Function Documentation	102
4.33.3.1 accept()	102
4.33.4 Member Data Documentation	103
4.33.4.1 burnButton	103
4.33.4.2 damageView	103
4.33.4.3 resetButton	103
4.33.4.4 weatherButton	103
4.34 MoralDamageCalculatorDecorator Class Reference	103

4.34.1 Detailed Description	105
4.34.2 Constructor & Destructor Documentation	105
4.34.2.1 MoralDamageCalculatorDecorator()	105
4.34.3 Member Function Documentation	105
4.34.3.1 calculateDamage()	105
4.34.3.2 isBuffed()	107
4.34.4 Member Data Documentation	107
4.34.4.1 unitIds	107
4.35 OnValueChangedListener< T extends Comparable Interface Template Reference	108
4.35.1 Detailed Description	108
4.36 OverlayDialog Class Reference	108
4.36.1 Detailed Description	109
4.36.2 Constructor & Destructor Documentation	109
4.36.2.1 OverlayDialog() [1/2]	110
4.36.2.2 OverlayDialog() [2/2]	110
4.36.3 Member Function Documentation	110
4.36.3.1 onCreate()	110
4.36.4 Member Data Documentation	111
4.36.4.1 BACKGROUND	111
4.36.4.2 dismissViewId	111
4.36.4.3 layout	111
4.36.4.4 NO_DISMISS_VIEW	111
4.37 RemoveUnitsUseCase Class Reference	112
4.37.1 Detailed Description	112
4.37.2 Member Function Documentation	112
4.37.2.1 getRevengeDialog()	112
4.37.2.2 remove() [1/2]	113
4.37.2.3 remove() [2/2]	114
4.38 ResetAlertDialogBuilderAdapter Class Reference	114
4.38.1 Detailed Description	115
4.38.2 Constructor & Destructor Documentation	115
4.38.2.1 ResetAlertDialogBuilderAdapter()	115
4.38.3 Member Function Documentation	115
4.38.3.1 create()	115
4.38.3.2 setMonsterDialog()	116
4.38.3.3 setTrigger()	116
4.38.4 Member Data Documentation	116
4.38.4.1 adapteeBuilder	116
4.38.4.2 dialogCallback	117
4.39 ResetDialogUseCase Class Reference	117
4.39.1 Detailed Description	117
4.39.2 Member Function Documentation	118

4.39.2.1	<a href="#">getDialogType()</a>	118
4.39.2.2	<a href="#">reset()</a> [1/2]	118
4.39.2.3	<a href="#">reset()</a> [2/2]	119
4.40	<a href="#">ResetRepositoryUseCase Class Reference</a>	119
4.40.1	<a href="#">Detailed Description</a>	120
4.40.2	<a href="#">Member Function Documentation</a>	120
4.40.2.1	<a href="#">getRandomUnit()</a>	120
4.40.2.2	<a href="#">getRevengeDialog()</a>	120
4.40.2.3	<a href="#">reset()</a> [1/2]	121
4.40.2.4	<a href="#">reset()</a> [2/2]	121
4.41	<a href="#">RevengeAlertDialogBuilderAdapter Class Reference</a>	122
4.41.1	<a href="#">Detailed Description</a>	123
4.41.2	<a href="#">Constructor &amp; Destructor Documentation</a>	123
4.41.2.1	<a href="#">RevengeAlertDialogBuilderAdapter()</a>	123
4.41.3	<a href="#">Member Function Documentation</a>	123
4.41.3.1	<a href="#">create()</a>	123
4.41.3.2	<a href="#">insertAvengers()</a>	124
4.41.3.3	<a href="#">setNegativeCallback()</a>	124
4.41.3.4	<a href="#">setPositiveCallback()</a>	124
4.41.4	<a href="#">Member Data Documentation</a>	125
4.41.4.1	<a href="#">adapteeBuilder</a>	125
4.41.4.2	<a href="#">AVENGER_ABILITY</a>	125
4.41.4.3	<a href="#">AVENGER_DAMAGE</a>	125
4.41.4.4	<a href="#">AVENGER_EPIC</a>	126
4.41.4.5	<a href="#">AVENGER_ROW</a>	126
4.41.4.6	<a href="#">AVENGER_SQUAD</a>	126
4.42	<a href="#">RowDao Interface Reference</a>	126
4.42.1	<a href="#">Detailed Description</a>	127
4.42.2	<a href="#">Member Function Documentation</a>	127
4.42.2.1	<a href="#">clearRows()</a>	127
4.42.2.2	<a href="#">clearWeather()</a>	127
4.42.2.3	<a href="#">insertRow()</a>	127
4.42.2.4	<a href="#">isHorn()</a>	128
4.42.2.5	<a href="#">isHornFlowable()</a>	128
4.42.2.6	<a href="#">isWeather()</a>	128
4.42.2.7	<a href="#">isWeatherFlowable()</a>	129
4.42.2.8	<a href="#">updateHorn()</a>	129
4.42.2.9	<a href="#">updateWeather()</a>	129
4.43	<a href="#">RowEntity Class Reference</a>	130
4.43.1	<a href="#">Detailed Description</a>	131
4.43.2	<a href="#">Constructor &amp; Destructor Documentation</a>	131
4.43.2.1	<a href="#">RowEntity()</a>	131

4.43.3 Member Function Documentation	131
4.43.3.1 getId()	131
4.43.3.2 isHorn()	131
4.43.3.3 isWeather()	131
4.43.3.4 setHorn()	132
4.43.3.5 setWeather()	132
4.43.4 Member Data Documentation	132
4.43.4.1 horn	132
4.43.4.2 id	132
4.43.4.3 weather	132
4.44 RowType Enum Reference	133
4.44.1 Detailed Description	133
4.44.2 Member Data Documentation	133
4.44.2.1 MELEE	133
4.44.2.2 RANGE	133
4.44.2.3 SIEGE	134
4.45 RowUiState Class Reference	134
4.45.1 Detailed Description	134
4.45.2 Constructor & Destructor Documentation	135
4.45.2.1 RowUiState()	135
4.45.3 Member Function Documentation	135
4.45.3.1 getDamage()	135
4.45.3.2 getUnits()	135
4.45.3.3 isHorn()	136
4.45.3.4 isWeather()	136
4.45.4 Member Data Documentation	136
4.45.4.1 damage	136
4.45.4.2 horn	136
4.45.4.3 units	136
4.45.4.4 weather	136
4.46 RowUiStateObserver Class Reference	137
4.46.1 Detailed Description	138
4.46.2 Constructor & Destructor Documentation	138
4.46.2.1 RowUiStateObserver()	138
4.46.3 Member Function Documentation	138
4.46.3.1 accept()	139
4.46.3.2 getObserver()	139
4.46.4 Member Data Documentation	139
4.46.4.1 damageView	140
4.46.4.2 hornView	140
4.46.4.3 unitView	140
4.46.4.4 weatherResource	140

---

4.46.4.5 weatherView . . . . .	140
4.47 RuleActivity Class Reference . . . . .	141
4.47.1 Detailed Description . . . . .	141
4.47.2 Member Function Documentation . . . . .	142
4.47.2.1 onCreate() . . . . .	142
4.47.2.2 onOptionsItemSelected() . . . . .	142
4.47.3 Member Data Documentation . . . . .	142
4.47.3.1 INTENT_EXTRA_TAG . . . . .	142
4.48 RuleSection Enum Reference . . . . .	142
4.48.1 Detailed Description . . . . .	143
4.48.2 Member Data Documentation . . . . .	143
4.48.2.1 CARD_ABILITIES . . . . .	143
4.48.2.2 CARDS . . . . .	143
4.48.2.3 COMMANDER . . . . .	143
4.48.2.4 COURSE . . . . .	144
4.48.2.5 FACTIONS . . . . .	144
4.48.2.6 GENERAL . . . . .	144
4.48.2.7 SPECIAL_CARDS . . . . .	144
4.49 SectionsPagerAdapter Class Reference . . . . .	144
4.49.1 Detailed Description . . . . .	145
4.49.2 Constructor & Destructor Documentation . . . . .	145
4.49.2.1 SectionsPagerAdapter() . . . . .	145
4.49.3 Member Function Documentation . . . . .	145
4.49.3.1 createFragment() . . . . .	145
4.49.3.2 getItemCount() . . . . .	146
4.50 SettingsActivity Class Reference . . . . .	146
4.50.1 Detailed Description . . . . .	147
4.50.2 Member Function Documentation . . . . .	147
4.50.2.1 onCreate() . . . . .	147
4.50.2.2 onOptionsItemSelected() . . . . .	148
4.50.2.3 onPreferenceStartFragment() . . . . .	148
4.50.2.4 onSaveInstanceState() . . . . .	148
4.50.3 Member Data Documentation . . . . .	149
4.50.3.1 CURRENT_FRAGMENT_KEY . . . . .	149
4.51 SettingsHeaderFragment Class Reference . . . . .	149
4.51.1 Detailed Description . . . . .	150
4.51.2 Member Function Documentation . . . . .	150
4.51.2.1 onCreatePreferences() . . . . .	150
4.52 SettingsRuleFragment Class Reference . . . . .	150
4.52.1 Detailed Description . . . . .	151
4.52.2 Member Function Documentation . . . . .	151
4.52.2.1 getSectionClickListener() . . . . .	151

4.52.2.2 onCreatePreferences()	152
4.53 SettingsSoundFragment Class Reference	152
4.53.1 Detailed Description	153
4.53.2 Member Function Documentation	153
4.53.2.1 onCreatePreferences()	153
4.54 ShowUnitsDialog Class Reference	153
4.54.1 Detailed Description	155
4.54.2 Constructor & Destructor Documentation	155
4.54.2.1 ShowUnitsDialog()	155
4.54.3 Member Function Documentation	155
4.54.3.1 getDialog()	155
4.54.3.2 getRow()	156
4.54.3.3 onCreate()	156
4.54.4 Member Data Documentation	157
4.54.4.1 cardListAdapter	157
4.54.4.2 disposables	157
4.54.4.3 row	157
4.54.4.4 unitRecyclerView	157
4.55 Sound Class Reference	158
4.55.1 Detailed Description	158
4.55.2 Constructor & Destructor Documentation	159
4.55.2.1 Sound()	159
4.55.3 Member Function Documentation	159
4.55.3.1 createSound()	159
4.55.3.2 getPreferenceKey()	160
4.55.3.3 getSoundId()	160
4.55.3.4 isActivated()	160
4.55.3.5 setActivated() [1/2]	160
4.55.3.6 setActivated() [2/2]	160
4.55.4 Member Data Documentation	161
4.55.4.1 activated	161
4.55.4.2 preferenceKey	161
4.55.4.3 soundId	161
4.56 SoundManager Class Reference	161
4.56.1 Detailed Description	163
4.56.2 Constructor & Destructor Documentation	163
4.56.2.1 SoundManager()	163
4.56.3 Member Function Documentation	163
4.56.3.1 playBurnSound()	163
4.56.3.2 playCardAddSound()	164
4.56.3.3 playCardRemovedSound()	164
4.56.3.4 playClearWeatherSound()	164

4.56.3.5	<a href="#">playCoinSound()</a>	164
4.56.3.6	<a href="#">playHornSound()</a>	165
4.56.3.7	<a href="#">playResetSound()</a>	165
4.56.3.8	<a href="#">playSound()</a>	165
4.56.3.9	<a href="#">playWeatherSound()</a>	166
4.56.4	<a href="#">Member Data Documentation</a>	166
4.56.4.1	<a href="#">changeListener</a>	166
4.56.4.2	<a href="#">SOUND_BURN</a>	166
4.56.4.3	<a href="#">SOUND_CARDS_EPIC</a>	166
4.56.4.4	<a href="#">SOUND_CARDS_MELEE</a>	167
4.56.4.5	<a href="#">SOUND_CARDS_RANGE</a>	167
4.56.4.6	<a href="#">SOUND_CARDS_SIEGE</a>	167
4.56.4.7	<a href="#">SOUND_COIN</a>	167
4.56.4.8	<a href="#">SOUND_HORN</a>	168
4.56.4.9	<a href="#">SOUND_RESET</a>	168
4.56.4.10	<a href="#">SOUND_WEATHER_FOG</a>	168
4.56.4.11	<a href="#">SOUND_WEATHER_FROST</a>	168
4.56.4.12	<a href="#">SOUND_WEATHER_GOOD</a>	168
4.56.4.13	<a href="#">SOUND_WEATHER_RAIN</a>	169
4.56.4.14	<a href="#">soundPool</a>	169
4.56.4.15	<a href="#">sounds</a>	169
4.57	<a href="#">SquadManager Class Reference</a>	169
4.57.1	<a href="#">Detailed Description</a>	170
4.57.2	<a href="#">Constructor &amp; Destructor Documentation</a>	170
4.57.2.1	<a href="#">SquadManager()</a>	170
4.57.3	<a href="#">Member Function Documentation</a>	171
4.57.3.1	<a href="#">getFirstSquadWithMembers()</a>	171
4.57.3.2	<a href="#">onSquadChanged()</a>	171
4.57.4	<a href="#">Member Data Documentation</a>	171
4.57.4.1	<a href="#">MAX_NR_SQUADS</a>	171
4.57.4.2	<a href="#">states</a>	172
4.58	<a href="#">SquadState Class Reference</a>	172
4.58.1	<a href="#">Detailed Description</a>	173
4.58.2	<a href="#">Constructor &amp; Destructor Documentation</a>	173
4.58.2.1	<a href="#">SquadState()</a>	173
4.58.3	<a href="#">Member Function Documentation</a>	173
4.58.3.1	<a href="#">getMemberBaseDamage()</a>	174
4.58.3.2	<a href="#">getSquadMembers()</a>	174
4.58.3.3	<a href="#">getSquadNumber()</a>	174
4.58.3.4	<a href="#">getState()</a>	174
4.58.3.5	<a href="#">hasMembers()</a>	175
4.58.4	<a href="#">Member Data Documentation</a>	175



4.58.4.1 memberBaseDamage	175
4.58.4.2 squadMembers	175
4.58.4.3 squadNumber	176
4.59 StringValuePicker< T extends Comparable Class Template Reference	176
4.59.1 Detailed Description	176
4.60 ResetDialogUseCase.Trigger Enum Reference	176
4.60.1 Detailed Description	177
4.60.2 Member Data Documentation	177
4.60.2.1 BUTTON_CLICK	177
4.60.2.2 FACTION_SWITCH	177
4.61 UnitDao Interface Reference	177
4.61.1 Detailed Description	178
4.61.2 Member Function Documentation	178
4.61.2.1 countUnits() [1/2]	178
4.61.2.2 countUnits() [2/2]	179
4.61.2.3 deleteUnit()	179
4.61.2.4 deleteUnits()	179
4.61.2.5 getUnit()	179
4.61.2.6 getUnits() [1/2]	180
4.61.2.7 getUnits() [2/2]	180
4.61.2.8 getUnitsFlowable()	180
4.61.2.9 hasNonEpicUnitsFlowable()	181
4.61.2.10 insertUnit() [1/2]	181
4.61.2.11 insertUnit() [2/2]	181
4.62 UnitEntity Class Reference	181
4.62.1 Detailed Description	183
4.62.2 Constructor & Destructor Documentation	183
4.62.2.1 UnitEntity()	183
4.62.3 Member Function Documentation	184
4.62.3.1 calculateDamage()	184
4.62.3.2 collectionToString()	184
4.62.3.3 getAbility()	185
4.62.3.4 getDamage()	185
4.62.3.5 getId()	185
4.62.3.6 getRow()	185
4.62.3.7 getSquad()	186
4.62.3.8 isBuffed()	186
4.62.3.9 isEpic()	186
4.62.3.10 setAbility()	186
4.62.3.11 setDamage()	187
4.62.3.12 setEpic()	187
4.62.3.13 setId()	187

4.62.3.14 setRow()	187
4.62.3.15 setSquad()	188
4.62.3.16 toString()	188
4.62.4 Member Data Documentation	188
4.62.4.1 ability	188
4.62.4.2 damage	188
4.62.4.3 epic	189
4.62.4.4 id	189
4.62.4.5 row	189
4.62.4.6 squad	189
4.63 UnitRepository Class Reference	189
4.63.1 Detailed Description	191
4.63.2 Constructor & Destructor Documentation	191
4.63.2.1 UnitRepository()	191
4.63.3 Member Function Documentation	192
4.63.3.1 clearWeather()	192
4.63.3.2 copy()	192
4.63.3.3 countUnits() [1/2]	192
4.63.3.4 countUnits() [2/2]	193
4.63.3.5 delete() [1/2]	193
4.63.3.6 delete() [2/2]	193
4.63.3.7 getRepository()	194
4.63.3.8 getUnit()	194
4.63.3.9 getUnits() [1/2]	194
4.63.3.10 getUnits() [2/2]	195
4.63.3.11 getUnitsFlowable()	195
4.63.3.12 hasNonEpicUnitsFlowable()	195
4.63.3.13 initializeRows()	196
4.63.3.14 insertUnit() [1/2]	196
4.63.3.15 insertUnit() [2/2]	197
4.63.3.16 isHorn()	197
4.63.3.17 isHornFlowable()	198
4.63.3.18 isWeather()	198
4.63.3.19 isWeatherFlowable()	198
4.63.3.20 reset() [1/2]	199
4.63.3.21 reset() [2/2]	199
4.63.3.22 switchHorn()	199
4.63.3.23 switchWeather()	200
4.63.4 Member Data Documentation	200
4.63.4.1 database	200
4.64 ValuePicker< T extends Comparable Class Template Reference	200
4.64.1 Detailed Description	201

---

4.65 WeatherDamageCalculator Class Reference . . . . .	201
4.65.1 Detailed Description . . . . .	202
4.65.2 Constructor & Destructor Documentation . . . . .	202
4.65.2.1 WeatherDamageCalculator() . . . . .	202
4.65.3 Member Function Documentation . . . . .	203
4.65.3.1 calculateDamage() . . . . .	203
4.65.3.2 isBuffed() . . . . .	203
4.65.4 Member Data Documentation . . . . .	204
4.65.4.1 weather . . . . .	204
<b>Index</b>	<b>205</b>



# 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	<a href="#">gwent_cards_en.zip</a>	<a href="#">gwent_cards_en.tar.gz</a>
German	<a href="#">gwent_cards_de.zip</a>	<a href="#">gwent_cards_de.tar.gz</a>

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	9
BurnDialogUseCase	18
ResetAlertDialogBuilderAdapter.Callback	20
ChangeFactionDialog.Callback	21
CardNumberPickerAdapter	26
CardUiState	30
CardUiStateFactory	36
DamageCalculator.Color	49
DamageCalculator	50
DamageCalculatorDecorator	57
BondDamageCalculatorDecorator	15
HornDamageCalculatorDecorator	81
MoralDamageCalculatorDecorator	103
WeatherDamageCalculator	201
DamageCalculatorBuildDirector	52
DamageCalculatorBuilder	54
DamageCalculatorUseCase	59
ResetDialogUseCase.DialogType	65
ImageViewSwitchAnimator	85
IndicatorManager	86
MenuUiState	98
OnPreferenceStartFragmentCallback	
SettingsActivity	146
OnSharedPreferenceChangeListener	
FactionSwitchListener	66
OnValueChangeListener< T extends Comparable	108
RemoveUnitsUseCase	112
ResetAlertDialogBuilderAdapter	114
ResetDialogUseCase	117
ResetRepositoryUseCase	119
RevengeAlertDialogBuilderAdapter	122
RowDao	126
RowEntity	130
RowType	133
RowUiState	134
RuleSection	142
Sound	158
SoundManager	161
SquadManager	169
SquadState	172

StringValuePicker< T extends Comparable . . . . .	176
ResetDialogUseCase.Trigger . . . . .	176
UnitDao . . . . .	177
UnitEntity . . . . .	181
UnitRepository . . . . .	189
ValuePicker	
DamageValuePicker . . . . .	60
ValuePicker< T extends Comparable . . . . .	200
ViewHolder	
CardListAdapter.CardViewHolder . . . . .	39
AndroidViewModel	
GameBoardViewModel . . . . .	71
AppCompatActivity	
IntroductionActivity . . . . .	89
MainActivity . . . . .	94
RuleActivity . . . . .	141
SettingsActivity . . . . .	146
Application	
GwentApplication . . . . .	78
Consumer	
MenuUiStateObserver . . . . .	101
RowUiStateObserver . . . . .	137
Dialog	
OverlayDialog . . . . .	108
ChangeFactionDialog . . . . .	44
CoinFlipDialog . . . . .	46
AddCardDialog . . . . .	10
ShowUnitsDialog . . . . .	153
Fragment	
IntroductionFragment . . . . .	91
FragmentManager	
SectionsPagerAdapter . . . . .	144
ListAdapter	
CardListAdapter . . . . .	22
PreferenceFragmentCompat	
SettingsHeaderFragment . . . . .	149
SettingsRuleFragment . . . . .	150
SettingsSoundFragment . . . . .	152
RoomDatabase	
AppDatabase . . . . .	14



## Chapter 3

# Class Index

### 3.1 Class List

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

<a href="#">Ability</a>	An enum listing the different card abilities possible . . . . .	9
<a href="#">AddCardDialog</a>	An <a href="#">OverlayDialog</a> used to add new UnitEntity objects to a certain <a href="#">row</a> . . . . .	10
<a href="#">AppDatabase</a>	An abstract class extending RoomDatabase and representing the SQLite database of the application . . . . .	14
<a href="#">BondDamageCalculatorDecorator</a>	A <a href="#">DamageCalculator</a> class responsible for calculating the tight bond buff if necessary, i.e., if there are units with the <a href="#">com.paternaggschga.gwent.data.Ability::BINDING</a> ability . . . . .	15
<a href="#">BurnDialogUseCase</a>	A use case class responsible for dispatching a remove call to <a href="#">RemoveUnitsUseCase</a> . . . . .	18
<a href="#">ResetAlertDialogBuilderAdapter.Callback</a>	An interface defining functions to propagate the user's decision back to the creator . . . . .	20
<a href="#">ChangeFactionDialog.Callback</a>	An interface used for propagating a selected theme back to the creator of a <a href="#">ChangeFactionDialog</a> . . . . .	21
<a href="#">CardListAdapter</a>	A ListAdapter used to provide <a href="#">CardListAdapter.CardViewHolder</a> objects created from <a href="#">CardUiState</a> objects to a RecyclerView . . . . .	22
<a href="#">CardNumberPickerAdapter</a>	A helper class responsible for initializing the NumberPicker views of an <a href="#">AddCardDialog</a> (in <a href="#">#↔CardNumberPickerAdapter()</a> ) and adding UnitEntity objects with the selected attributes when <a href="#">addSelectedUnits()</a> is called . . . . .	26
<a href="#">CardUiState</a>	A data class encapsulating the visible state of a card in the card list shown by the <a href="#">ShowUnitsDialog</a> . . . . .	30
<a href="#">CardUiStateFactory</a>	A factory class responsible for creating <a href="#">CardUiState</a> objects from UnitEntity objects . . . . .	36
<a href="#">CardListAdapter.CardViewHolder</a>	A RecyclerView.ViewHolder class managing a card view that shows a representation of the UnitEntity referenced by <a href="#">itemId</a> . . . . .	39
<a href="#">ChangeFactionDialog</a>	An <a href="#">OverlayDialog</a> class used to change the faction design . . . . .	44
<a href="#">CoinFlipDialog</a>	An <a href="#">OverlayDialog</a> class showing the result of a coin-flip . . . . .	46
<a href="#">DamageCalculator.Color</a>	An enum containing values representing whether a unit is buffed, de-buffed or not affected by status effects . . . . .	49

<a href="#">DamageCalculator</a>	An interface for a class capable of calculating the (de-)buffed damage of a unit for a given <a href="#">UnitEntity::id</a> and <a href="#">UnitEntity::damage</a> . . . . .	50
<a href="#">DamageCalculatorBuildDirector</a>	A build director responsible for the creation of <a href="#">DamageCalculator</a> instances from sets of units using <a href="#">DamageCalculatorBuilder</a> . . . . .	52
<a href="#">DamageCalculatorBuilder</a>	A builder class responsible for creating a decorator structure of <a href="#">DamageCalculator</a> classes with the given parameters . . . . .	54
<a href="#">DamageCalculatorDecorator</a>	An abstract <a href="#">DamageCalculator</a> that is used as a superclass for the decorator classes implementing the damage calculation . . . . .	57
<a href="#">DamageCalculatorUseCase</a>	A use case class responsible for creating <a href="#">DamageCalculator</a> objects for rows using <a href="#">DamageCalculatorBuildDirector</a> . . . . .	59
<a href="#">DamageValuePicker</a>	A <a href="#">ValuePicker</a> used for selection of damage values for <a href="#">UnitEntity::damage</a> . . . . .	60
<a href="#">ResetDialogUseCase.DialogType</a>	An Enum defining which form of Dialog should be shown . . . . .	65
<a href="#">FactionSwitchListener</a>	A class implementing <a href="#">SharedPreferences.OnSharedPreferencesChangeListener</a> for the <a href="#">THEME_PREFERENCE_KEY</a> <a href="#">SharedPreferences</a> . . . . .	66
<a href="#">GameBoardViewModel</a>	An <a href="#">AndroidViewModel</a> class responsible for encapsulating and offering state of views in <a href="#">activity_main.xml</a> , i.e., that show the overall game board . . . . .	71
<a href="#">GwentApplication</a>	An Application encapsulating the Gwent app . . . . .	78
<a href="#">HornDamageCalculatorDecorator</a>	A <a href="#">DamageCalculator</a> class responsible for calculating the horn buff if necessary, i.e., if there are units with the <a href="#">com.peternaggschga.gwent.data.Ability::HORN</a> ability that are not the unit the damage is calculated for or if there is a commander's horn in this row . . . . .	81
<a href="#">ImageViewSwitchAnimator</a>	A class providing functionality for changing the resource shown by an <a href="#">ImageView</a> using a fading animation . . . . .	85
<a href="#">IndicatorManager</a>	A class managing the indicator views shown in the bottom bar of the <a href="#">IntroductionActivity</a> . . . .	86
<a href="#">IntroductionActivity</a>	An <a href="#">AppCompatActivity</a> that gives the user an introduction into the usage of the application . . .	89
<a href="#">IntroductionFragment</a>	A <a href="#">Fragment</a> containing an <a href="#">ImageView</a> and a <a href="#">TextView</a> conveying introductory information . . .	91
<a href="#">MainActivity</a>	An <a href="#">AppCompatActivity</a> that is called on startup and that encapsulates the main view onto the game board . . . . .	94
<a href="#">MenuUiState</a>	A data class encapsulating the visible state of the menu in the main view of the application . .	98
<a href="#">MenuUiStateObserver</a>	An observer class responsible for updating the menu views when notified, i.e., when a new <a href="#">MenuUiState</a> is produced by the subscribed <a href="#">Flowable</a> . . . . .	101
<a href="#">MoralDamageCalculatorDecorator</a>	A <a href="#">DamageCalculator</a> class responsible for calculating the moral buff if necessary, i.e., if there are units with the <a href="#">com.peternaggschga.gwent.data.Ability::MORAL_BOOST</a> ability that are not the unit the damage is calculated for . . . . .	103
<a href="#">OnValueChangeListener&lt; T extends Comparable</a>	An interface defining a callback for changes of the current value . . . . .	108
<a href="#">OverlayDialog</a>	A <a href="#">Dialog</a> class which is used for popups that are shown on top of the calling Activity . . . . .	108
<a href="#">RemoveUnitsUseCase</a>	A use case class responsible for removing units from a <a href="#">UnitRepository</a> . . . . .	112

<a href="#">ResetAlertDialogBuilderAdapter</a>	An adapter class adapting AlertDialog.Builder to provide an interface for creating an AlertDialog asking the user whether he really wants to reset . . . . .	114
<a href="#">ResetDialogUseCase</a>	A use case class responsible for dispatching a reset call to <a href="#">ResetRepositoryUseCase</a> , possibly after a confirmation by the user obtained from a Dialog . . . . .	117
<a href="#">ResetRepositoryUseCase</a>	A use case class responsible for resetting the UnitRepository . . . . .	119
<a href="#">RevengeAlertDialogBuilderAdapter</a>	An adapter class adapting AlertDialog.Builder to provide an interface for creating an AlertDialog asking the user whether they want to invoke the <a href="#">Ability::REVENGE</a> ability . . . . .	122
<a href="#">RowDao</a>	An interface defining data access operations related to <code>rows</code> table . . . . .	126
<a href="#">RowEntity</a>	A class representing the state of an attack row, i.e., weather and commanders horn . . . . .	130
<a href="#">RowType</a>	An enum listing the different types of attack rows . . . . .	133
<a href="#">RowUiState</a>	A data class encapsulating the visible state of a row in the main view of the application . . . . .	134
<a href="#">RowUiStateObserver</a>	An observer class responsible for updating the views of the row defined in <code>#row</code> when notified, i.e., when a new <a href="#">RowUiState</a> is produced the observed Flowable . . . . .	137
<a href="#">RuleActivity</a>	An AppCompatActivity used to present the rules of the game . . . . .	141
<a href="#">RuleSection</a>	An Enum used to discern the different possible sections shown by <a href="#">RuleActivity</a> . . . . .	142
<a href="#">SectionsPagerAdapter</a>	A FragmentStateAdapter that returns an <a href="#">IntroductionFragment</a> corresponding to one of the sections/tabs/pages . . . . .	144
<a href="#">SettingsActivity</a>	An AppCompatActivity implementing PreferenceFragmentCompat.OnPreferenceStartFragmentCallback that is used by the user to manage the SharedPreferences of the application . . . . .	146
<a href="#">SettingsHeaderFragment</a>	A PreferenceFragmentCompat class encapsulating the main preference screen, i.e., the Preferences defined in <code>R.xml#header_preferences</code> . . . . .	149
<a href="#">SettingsRuleFragment</a>	A PreferenceFragmentCompat class encapsulating the rule preference screen, i.e., the rule sections defined in <code>R.xml#rule_preferences</code> . . . . .	150
<a href="#">SettingsSoundFragment</a>	A PreferenceFragmentCompat class encapsulating the sound preference screen, i.e., the Preferences defined in <code>R.xml#sound_preferences</code> . . . . .	152
<a href="#">ShowUnitsDialog</a>	An <a href="#">OverlayDialog</a> used to list the units of a certain row and enabling the user to copy, add, or delete new units . . . . .	153
<a href="#">Sound</a>	A data class encapsulating information on a certain sound effect, e.g., its id in a SoundPool and whether it is activated in settings . . . . .	158
<a href="#">SoundManager</a>	A class responsible for initializing and playing sounds when they are enabled . . . . .	161
<a href="#">SquadManager</a>	A class used by the <a href="#">CardNumberPickerAdapter</a> to determine the first squad with members or set the default damage according to the squads base damage . . . . .	169
<a href="#">SquadState</a>	A data class encapsulating information about the squad defined by <a href="#">squadNumber</a> , i.e., how many <a href="#">squadMembers</a> are there and what is the <a href="#">memberBaseDamage</a> of units in this squad . . . . .	172
<a href="#">StringValuePicker&lt; T extends Comparable</a>	A ValuePicker displaying a resource String for each element in <code>#selectableValues</code> . . . . .	176

<a href="#">ResetDialogUseCase.Trigger</a>	
An Enum listing the possible triggers of a reset	176
<a href="#">UnitDao</a>	
An interface defining data access operations related to <code>units</code> table	177
<a href="#">UnitEntity</a>	
A class representing a card on the game board	181
<a href="#">UnitRepository</a>	
A facade class managing public access to the data layer	189
<a href="#">ValuePicker&lt; T extends Comparable</a>	
A generic abstract wrapper class around <code>NumberPicker</code> used to select values of arbitrary types extending <code>Comparable</code>	200
<a href="#">WeatherDamageCalculator</a>	
A <a href="#">DamageCalculator</a> class responsible for calculating the weather debuff if necessary, i.e., if <a href="#">weather</a> is true	201

## Chapter 4

# Class Documentation

### 4.1 Ability Enum Reference

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



A rectangular box with a black border containing the text "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.*

#### 4.1.1 Detailed Description

An enum listing the different card abilities possible.  
Definition at line 6 of file Ability.java.

#### 4.1.2 Member Data Documentation

##### 4.1.2.1 BINDING

`BINDING`  
Represents a card that has the Tight Bond ability.

Definition at line 25 of file Ability.java.

Referenced by `CardNumberPickerAdapter.CardNumberPickerAdapter()`, `SquadManager.SquadManager()`, `UnitEntity.UnitEntity()`, `DamageCalculatorBuildDirector.getCalculator()`, `UnitRepository.insertUnit()`, `UnitEntity.setSquad()`, and `DamageCalculatorBuildDirector.setSquads()`.

#### 4.1.2.2 HORN

HORN

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

Dandelion.

Definition at line 15 of file Ability.java.

Referenced by `CardNumberPickerAdapter.CardNumberPickerAdapter()`, `DamageCalculatorBuildDirector.getCalculator()`, and `DamageCalculatorBuildDirector.setHorns()`.

#### 4.1.2.3 MORAL\_BOOST

MORAL\_BOOST

Represents a card that has the Morale Boost ability.

Definition at line 31 of file Ability.java.

Referenced by `CardNumberPickerAdapter.CardNumberPickerAdapter()`, `DamageCalculatorBuildDirector.getCalculator()`, and `DamageCalculatorBuildDirector.setMoralBoosts()`.

#### 4.1.2.4 NONE

NONE

Represents a card that has no ability.

Definition at line 10 of file Ability.java.

Referenced by `CardNumberPickerAdapter.CardNumberPickerAdapter()`.

#### 4.1.2.5 REVENGE

REVENGE

Represents a card that has the Summon Avenger ability.

Definition at line 20 of file Ability.java.

Referenced by `CardNumberPickerAdapter.CardNumberPickerAdapter()`, `RemoveUnitsUseCase.remove()`, and `ResetRepositoryUseCase.reset()`.

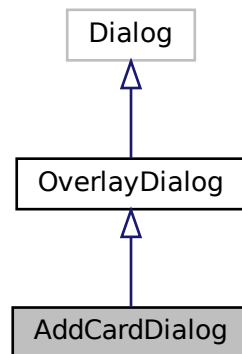
The documentation for this enum was generated from the following file:

- `gwent/data/Ability.java`

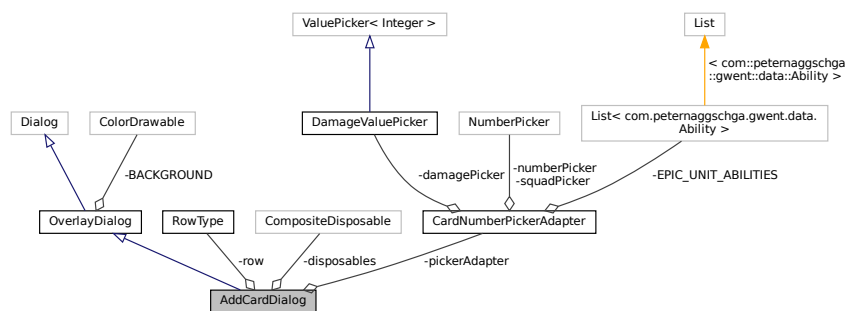
## 4.2 AddCardDialog Class Reference

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)  
*Constructor of an [AddCardDialog](#) called by the given [ShowUnitsDialog](#).*
- [AddCardDialog](#) (@NonNull Context context, @NonNull [RowType](#) row)  
*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.*

## Private Attributes

- final [RowType](#) row  
*RowType defining the row the new [UnitEntity](#) objects are added to.*
- 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

### 4.2.1 Detailed Description

An [OverlayDialog](#) used to add new [UnitEntity](#) objects to a certain [row](#).  
Definition at line 21 of file [AddCardDialog.java](#).

### 4.2.2 Constructor & Destructor Documentation

#### 4.2.2.1 AddCardDialog() [1/2]

```
AddCardDialog (
    @NonNull ShowUnitsDialog caller )
```

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

See also

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

#### Parameters

<i>caller</i>	ShowUnitsDialog that called this Dialog.
---------------	--

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

#### 4.2.2.2 AddCardDialog() [2/2]

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

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 <a href="#">UnitEntity</a> objects are added to.

Definition at line 73 of file [AddCardDialog.java](#).  
References [AddCardDialog.disposables](#), and [AddCardDialog.row](#).

### 4.2.3 Member Function Documentation

#### 4.2.3.1 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 90 of file `AddCardDialog.java`.

References `CardNumberPickerAdapter.addSelectedUnits()`, `AddCardDialog.disposables`, `GwentApplication.getRepository()`, `UnitRepository.getUnits()`, `AddCardDialog.pickerAdapter`, and `AddCardDialog.row`.

## 4.2.4 Member Data Documentation

### 4.2.4.1 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)`  
`#AddCardDialog(Context, RowType)`

Definition at line 38 of file `AddCardDialog.java`.

Referenced by `AddCardDialog.AddCardDialog()`, and `AddCardDialog.onCreate()`.

### 4.2.4.2 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 48 of file `AddCardDialog.java`.

Referenced by `AddCardDialog.onCreate()`.

### 4.2.4.3 row

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

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

Definition at line 26 of file `AddCardDialog.java`.

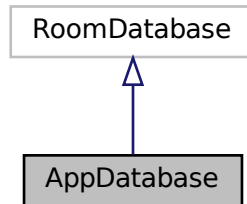
Referenced by `AddCardDialog.AddCardDialog()`, and `AddCardDialog.onCreate()`.

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

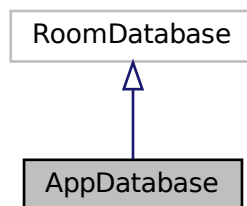
- `gwent/ui/dialogs/addcard/AddCardDialog.java`

## 4.3 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](#).*

#### 4.3.1 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.

#### 4.3.2 Member Function Documentation

##### 4.3.2.1 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()`.

## 4.3.2.2 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.delete()`, `UnitRepository.getUnit()`, `UnitRepository.getUnits()`, `UnitRepository.getUnitsFlowable()`, `UnitRepository.hasNonEpicUnitsFlowable()`, `UnitRepository.insertUnit()`, and `UnitRepository.reset()`.

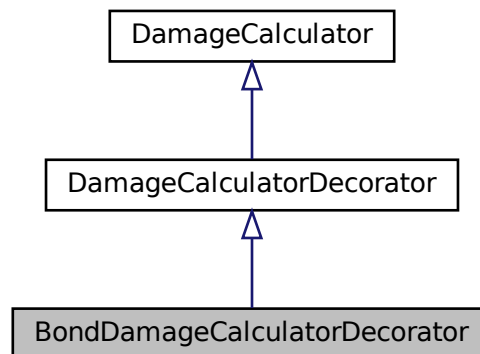
The documentation for this class was generated from the following file:

- `gwent/data/AppDatabase.java`

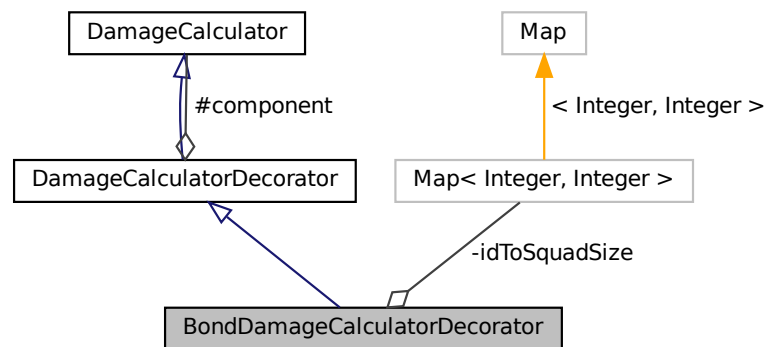
## 4.4 BondDamageCalculatorDecorator Class Reference

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`.*

## Private Attributes

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

## Additional Inherited Members

### 4.4.1 Detailed Description

A `DamageCalculator` class responsible for calculating the tight bond buff if necessary, i.e., if there are units with the `com.paternaggschga.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`.

### 4.4.2 Constructor & Destructor Documentation

#### 4.4.2.1 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>	<a href="#">DamageCalculator</a> that is being decorated by this decorator.
<i>idToSquadSize</i>	Map mapping the ids of all units with the <a href="#">com.peternaggschga.gwent.data.Ability::BINDING</a> 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](#).

### 4.4.3 Member Function Documentation

#### 4.4.3.1 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.peternaggschga.gwent.data.Ability::B](#) ability.

#### Parameters

<i>id</i>	Integer representing the <a href="#">UnitEntity::id</a> 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.idToSquadSize](#).

#### 4.4.3.2 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 <a href="#">UnitEntity::id</a> 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\(\)](#).

### 4.4.4 Member Data Documentation

#### 4.4.4.1 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.calculateDamage()`, and `BondDamageCalculatorDecorator.isBuffed()`.

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

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

## 4.5 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)`  
*Burns the strongest UnitEntity objects in UnitRepository.*

## Static Protected Member Functions

- static Single< Boolean > [burn](#) (@NonNull Context context, @NonNull [UnitRepository](#) repository)  
*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.*

### 4.5.1 Detailed Description

A use case class responsible for dispatching a remove call to [RemoveUnitsUseCase](#).  
Definition at line 27 of file BurnDialogUseCase.java.

### 4.5.2 Member Function Documentation

#### 4.5.2.1 [burn\(\)](#) [1/2]

```
static Single<Boolean> burn (
    @NonNull Context context ) [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.
----------------	---

##### Returns

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

##### See also

[#burn\(Context, UnitRepository\)](#)

[RemoveUnitsUseCase::remove\(Context, UnitRepository, Collection\)](#)

Definition at line 83 of file BurnDialogUseCase.java.

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

Referenced by [GameBoardViewModel.onBurnButtonPressed\(\)](#).

#### 4.5.2.2 [burn\(\)](#) [2/2]

```
static Single<Boolean> burn (
    @NonNull Context context,
    @NonNull UnitRepository repository ) [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.

**Returns**

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

**See also**

RemoveUnitsUseCase::remove(Context, UnitRepository, Collection)

Definition at line 98 of file BurnDialogUseCase.java.

References UnitEntity.collectionToString().

**4.5.2.3 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 36 of file BurnDialogUseCase.java.

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

- gwent/domain/cases/BurnDialogUseCase.java

**4.6 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\(boolean, boolean\)](#).*

**4.6.1 Detailed Description**

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

Definition at line 98 of file ResetAlertDialogBuilderAdapter.java.



## 4.6.2 Member Function Documentation

### 4.6.2.1 `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()`.

### 4.6.2.2 `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`

## 4.7 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](#).*

### 4.7.1 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`.

### 4.7.2 Member Function Documentation

#### 4.7.2.1 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.
--------------	--

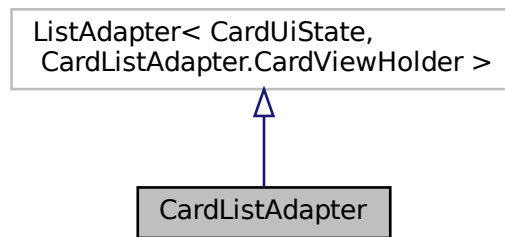
Referenced by `ChangeFactionDialog.getOnThemeClickListener()`.  
 The documentation for this interface was generated from the following file:

- `gwent/ui/dialogs/ChangeFactionDialog.java`

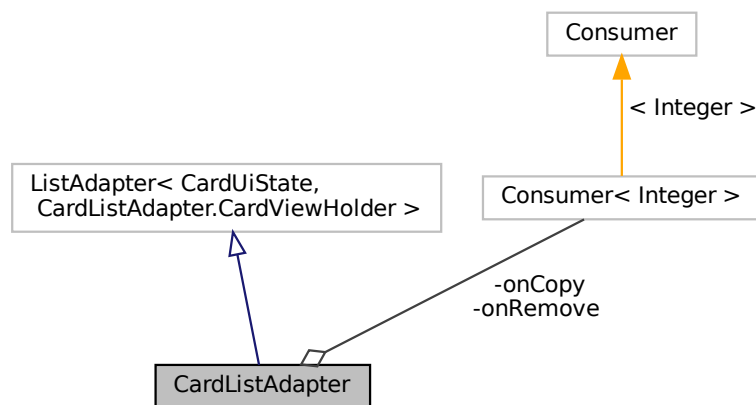
## 4.8 CardListAdapter Class Reference

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`.

### 4.8.1 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`.

### 4.8.2 Constructor & Destructor Documentation

#### 4.8.2.1 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 <a href="#">UnitEntity::id</a> , when the copy-button of the representing card is clicked.
<i>onRemove</i>	Consumer that is called with the <a href="#">UnitEntity::id</a> , 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](#).

### 4.8.3 Member Function Documentation

#### 4.8.3.1 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.

#### 4.8.3.2 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 <a href="#">CardViewHolder</a> 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\(\)](#).

#### 4.8.3.3 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 ViewGroup 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](#).

### 4.8.4 Member Data Documentation

#### 4.8.4.1 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()`.

#### 4.8.4.2 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()`.

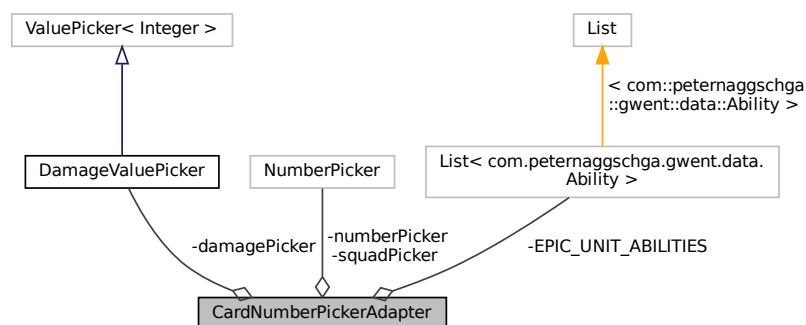
The documentation for this class was generated from the following file:

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

## 4.9 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](#).

- Completable [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 [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.

### 4.9.1 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 31 of file [CardNumberPickerAdapter.java](#).

### 4.9.2 Constructor & Destructor Documentation

#### 4.9.2.1 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>	ViewGroup containing the managed NumberPicker views.
<i>squadManager</i>	<a href="#">SquadManager</a> containing up-to-date <a href="#">SquadState</a> .

Definition at line 81 of file `CardNumberPickerAdapter.java`.

References `CardNumberPickerAdapter.abilityPicker`, `Ability.BINDING`, `CardNumberPickerAdapter.damagePicker`, `CardNumberPickerAdapter.EPIC_UNIT_ABILITIES`, `CardNumberPickerAdapter.epicPicker`, `CardNumberPickerAdapter.getDelayedOnValueChangeListener()`, `Ability.HORN`, `SquadManager.MAX_NR_SQUADS`, `Ability.MORAL_BOOST`, `Ability.NONE`, `CardNumberPickerAdapter.numberPicker`, `Ability.REVENGE`, `DamageValuePicker.setEpicValues()`, and `CardNumberPickerAdapter.squadPicker`.

### 4.9.3 Member Function Documentation

#### 4.9.3.1 addSelectedUnits()

```
Completable 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 Completable tracking operation status.

Definition at line 156 of file `CardNumberPickerAdapter.java`.

References `CardNumberPickerAdapter.abilityPicker`, `CardNumberPickerAdapter.damagePicker`, `CardNumberPickerAdapter.epicPicker`, `GwentApplication.getRepository()`, `DamageValuePicker.getValue()`, `CardNumberPickerAdapter.numberPicker`, and `CardNumberPickerAdapter.squadPicker`.

Referenced by `AddCardDialog.onCreate()`.

#### 4.9.3.2 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>	<code>NumberPicker.OnValueChangeListener</code> that is called when the value does not change.
-------------------------	--

## Returns

A `NumberPicker.OnValueChangeListener` with delayed execution.

Definition at line 142 of file `CardNumberPickerAdapter.java`.

Referenced by `CardNumberPickerAdapter.CardNumberPickerAdapter()`.



## 4.9.4 Member Data Documentation

### 4.9.4.1 abilityPicker

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

ValuePicker used to decide the value of [UnitEntity::ability](#).

See also

[squadPicker](#)

Definition at line 56 of file CardNumberPickerAdapter.java.

Referenced by CardNumberPickerAdapter.CardNumberPickerAdapter(), and CardNumberPickerAdapter.addSelectedUnits().

### 4.9.4.2 damagePicker

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

DamageValuePicker used to decide the value of [UnitEntity::damage](#).

Definition at line 48 of file CardNumberPickerAdapter.java.

Referenced by CardNumberPickerAdapter.CardNumberPickerAdapter(), and CardNumberPickerAdapter.addSelectedUnits().

### 4.9.4.3 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 36 of file CardNumberPickerAdapter.java.

Referenced by CardNumberPickerAdapter.CardNumberPickerAdapter().

### 4.9.4.4 epicPicker

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

ValuePicker used to decide the value of [UnitEntity::epic](#).

Definition at line 42 of file CardNumberPickerAdapter.java.

Referenced by CardNumberPickerAdapter.CardNumberPickerAdapter(), and CardNumberPickerAdapter.addSelectedUnits().

### 4.9.4.5 numberPicker

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

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

Definition at line 70 of file CardNumberPickerAdapter.java.

Referenced by CardNumberPickerAdapter.CardNumberPickerAdapter(), and CardNumberPickerAdapter.addSelectedUnits().

### 4.9.4.6 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 64 of file CardNumberPickerAdapter.java.

Referenced by CardNumberPickerAdapter.CardNumberPickerAdapter(), and CardNumberPickerAdapter.addSelectedUnits().

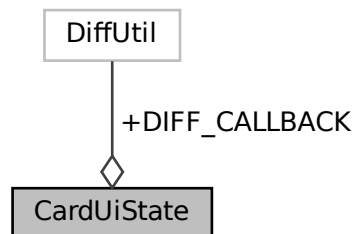
The documentation for this class was generated from the following file:

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

## 4.10 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:



### 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.*

### 4.10.1 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.

### 4.10.2 Constructor & Destructor Documentation

#### 4.10.2.1 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 <a href="#">UnitEntity::id</a> 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 <a href="#">UNUSED</a> .
<i>squad</i>	Integer representing the <a href="#">UnitEntity::squad</a> of the represented UnitEntity.

## Exceptions

<i>IllegalArgumentException</i>	When damage is less than <a href="#">UNUSED</a> 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`.

Referenced by `CardUiState.equals()`.

## 4.10.3 Member Function Documentation

### 4.10.3.1 `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

<i>o</i>	Object that is being compared with this <a href="#">CardUiState</a> .
----------	---

#### Returns

A Boolean defining whether the objects look the same.

Definition at line 212 of file `CardUiState.java`.

References `CardUiState.CardUiState()`, `CardUiState.abilityImageId`, `CardUiState.damageBackgroundImageId`, `CardUiState.damageString`, `CardUiState.damageTextColor`, and `CardUiState.squadString`.

### 4.10.3.2 `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()`.

### 4.10.3.3 `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().

**4.10.3.4 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().

**4.10.3.5 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().

**4.10.3.6 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().

#### 4.10.3.7 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().

#### 4.10.3.8 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().

#### 4.10.3.9 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().

### 4.10.4 Member Data Documentation

#### 4.10.4.1 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().

#### 4.10.4.2 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()`.

#### 4.10.4.3 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()`.

#### 4.10.4.4 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()`.

#### 4.10.4.5 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()`.

#### 4.10.4.6 `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()`.

#### 4.10.4.7 `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()`.

#### 4.10.4.8 `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.show↵Ability()`.

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

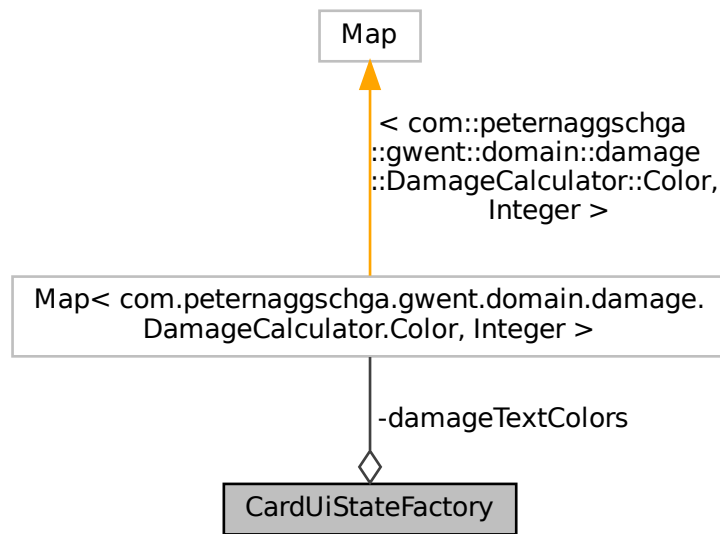
- `gwent/ui/dialogs/cards/CardUiState.java`

## 4.11 `CardUiStateFactory` Class Reference

A factory class responsible for creating [CardUiState](#) objects from `UnitEntity` objects.



Collaboration diagram for 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](#).*

### 4.11.1 Detailed Description

A factory class responsible for creating [CardUiState](#) objects from [UnitEntity](#) objects.

See also

[CardUiState](#)

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

### 4.11.2 Constructor & Destructor Documentation

#### 4.11.2.1 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](#).

### 4.11.3 Member Function Documentation

#### 4.11.3.1 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 <a href="#">CardUiState</a> objects.
--------------	---

##### Returns

A List of [CardUiState](#) objects from the given UnitEntity objects.

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

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

#### 4.11.3.2 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 <a href="#">CardUiState</a> .
<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.

## 4.11.4 Member Data Documentation

### 4.11.4.1 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().

### 4.11.4.2 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().

### 4.11.4.3 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().

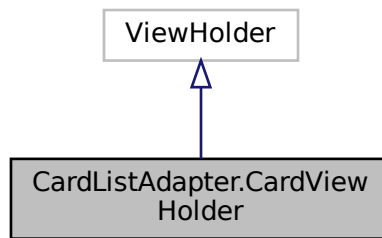
The documentation for this class was generated from the following file:

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

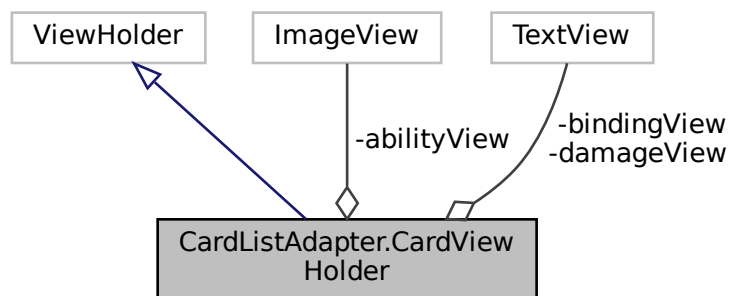
## 4.12 CardListAdapter.CardViewHolder Class Reference

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.*

### 4.12.1 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.

### 4.12.2 Constructor & Destructor Documentation

#### 4.12.2.1 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.damageView](#), [CardListAdapter.CardViewHolder.itemId](#), [CardListAdapter.onCopy](#), and [CardListAdapter.onRemove](#).

### 4.12.3 Member Function Documentation

#### 4.12.3.1 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.

**4.12.3.2 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.

**4.12.3.3 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.

**4.12.3.4 setItemId()**

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

Sets the #unitId of this ViewHolder.

**Parameters**

<i>itemId</i>	Integer referencing the <a href="#">UnitEntity::id</a> of the represented UnitEntity.
---------------	---

**Exceptions**

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

See also

[itemId](#)

Definition at line 232 of file CardListAdapter.java.

References CardListAdapter.CardViewHolder.itemId.

## 4.12.4 Member Data Documentation

### 4.12.4.1 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().

### 4.12.4.2 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().

### 4.12.4.3 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().

### 4.12.4.4 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()`.

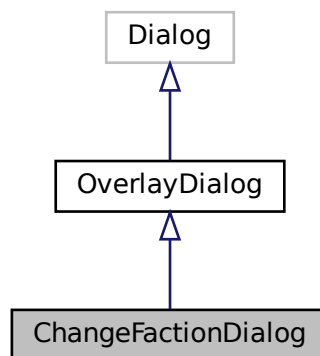
The documentation for this class was generated from the following file:

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

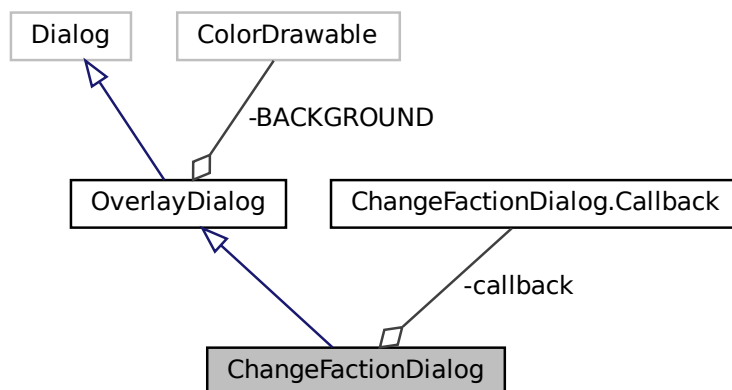
## 4.13 ChangeFactionDialog Class Reference

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.*

## 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

### 4.13.1 Detailed Description

An [OverlayDialog](#) class used to change the faction design.

Definition at line 20 of file ChangeFactionDialog.java.

### 4.13.2 Constructor & Destructor Documentation

#### 4.13.2.1 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>	<a href="#">Callback</a> that is called when a theme is selected.

Definition at line 33 of file ChangeFactionDialog.java.

References [ChangeFactionDialog.callback](#).

### 4.13.3 Member Function Documentation

#### 4.13.3.1 `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()`.

#### 4.13.3.2 `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()`.

### 4.13.4 Member Data Documentation

#### 4.13.4.1 `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()`.

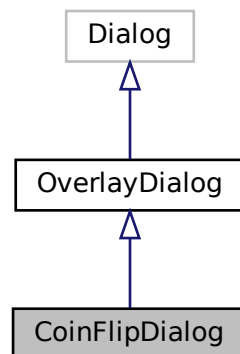
The documentation for this class was generated from the following file:

- `gwent/ui/dialogs/ChangeFactionDialog.java`

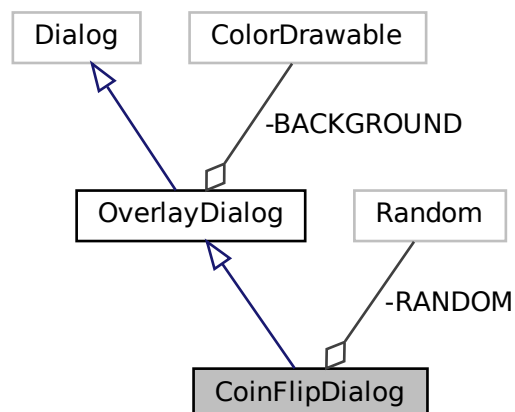
## 4.14 CoinFlipDialog Class Reference

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.*

## Static Private Attributes

- static final Random [RANDOM](#) = new Random()

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

## Additional Inherited Members

### 4.14.1 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.

### 4.14.2 Constructor & Destructor Documentation

#### 4.14.2.1 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.

### 4.14.3 Member Function Documentation

#### 4.14.3.1 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 onSaveInstanceState, or null if this is the first time.
---------------------------	--

Reimplemented from [OverlayDialog](#).

Definition at line 48 of file CoinFlipDialog.java.

References [CoinFlipDialog.RANDOM](#).

### 4.14.4 Member Data Documentation

#### 4.14.4.1 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().

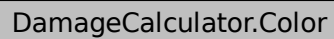
The documentation for this class was generated from the following file:

- gwent/ui/dialogs/CoinFlipDialog.java

## 4.15 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:



```
graph TD; subgraph "DamageCalculator.Color"; end
```

### 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.*

### 4.15.1 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.

### 4.15.2 Member Data Documentation

#### 4.15.2.1 BUFFED

**BUFFED**

The unit is buffed by status effects.

Definition at line 57 of file DamageCalculator.java.

#### 4.15.2.2 DEBUFFED

**DEBUFFED**

The unit is de-buffed by weather effects.

Definition at line 62 of file DamageCalculator.java.

#### 4.15.2.3 DEFAULT

##### DEFAULT

The unit is not affected by status effects.

Definition at line 53 of file DamageCalculator.java.

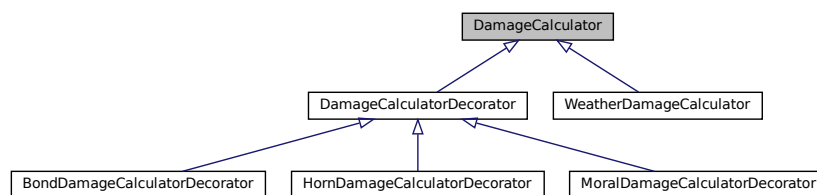
The documentation for this enum was generated from the following file:

- gwent/domain/damage/DamageCalculator.java

## 4.16 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.*

### 4.16.1 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.paternaggschga.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`.

## 4.16.2 Member Function Documentation

### 4.16.2.1 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 <a href="#">UnitEntity::id</a> 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 [WeatherDamageCalculator](#), [MoralDamageCalculatorDecorator](#), [HornDamageCalculatorDecorator](#), and [BondDamageCalculatorDecorator](#).

Referenced by `BondDamageCalculatorDecorator.calculateDamage()`, `HornDamageCalculatorDecorator.calculateDamage()`, and `MoralDamageCalculatorDecorator.calculateDamage()`.

### 4.16.2.2 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 <a href="#">UnitEntity::id</a> 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 [WeatherDamageCalculator](#), [MoralDamageCalculatorDecorator](#), [HornDamageCalculatorDecorator](#), and [BondDamageCalculatorDecorator](#).

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](#)

## 4.17 DamageCalculatorBuildDirector Class Reference

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

Collaboration diagram for DamageCalculatorBuildDirector:



```

classDiagram
    class 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\(\)](#).*

#### 4.17.1 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](#).



## 4.17.2 Member Function Documentation

### 4.17.2.1 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 [Ability.BINDING](#), [DamageCalculatorBuilder.getResult\(\)](#), [Ability.HORN](#), [Ability.MORAL\\_BOOST](#), [DamageCalculatorBuildDirector.setHorns\(\)](#), [DamageCalculatorBuildDirector.setMoralBoosts\(\)](#), [DamageCalculatorBuildDirector.setSquads\(\)](#), and [DamageCalculatorBuilder.setWeather\(\)](#).

Referenced by [DamageCalculatorUseCase.getDamageCalculator\(\)](#).

### 4.17.2.2 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>	<a href="#">DamageCalculatorBuilder</a> where the horn buff is added.

#### See also

[DamageCalculatorBuilder::setHorn\(List\)](#)

Definition at line 106 of file DamageCalculatorBuildDirector.java.

References [UnitEntity.getId\(\)](#), and [Ability.HORN](#).

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

### 4.17.2.3 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>	<a href="#">DamageCalculatorBuilder</a> where the horn buff is added.

#### See also

[DamageCalculatorBuilder::setMoral\(List\)](#)

Definition at line 85 of file [DamageCalculatorBuildDirector.java](#).  
References [UnitEntity.getId\(\)](#), and [Ability.MORAL\\_BOOST](#).  
Referenced by [DamageCalculatorBuildDirector.getCalculator\(\)](#).

#### 4.17.2.4 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>	<a href="#">DamageCalculatorBuilder</a> where the horn buff is added.

#### See also

[DamageCalculatorBuilder::setBond\(Map\)](#)

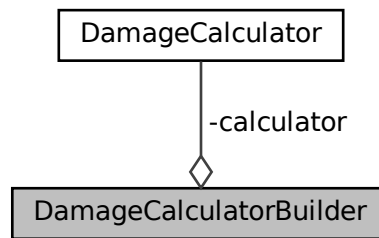
Definition at line 61 of file [DamageCalculatorBuildDirector.java](#).  
References [Ability.BINDING](#).  
Referenced by [DamageCalculatorBuildDirector.getCalculator\(\)](#).  
The documentation for this class was generated from the following file:

- [gwent/domain/damage/DamageCalculatorBuildDirector.java](#)

## 4.18 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](#).*

### 4.18.1 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](#).

### 4.18.2 Member Function Documentation

#### 4.18.2.1 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()`.

#### 4.18.2.2 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 <a href="#">com.paternaggschga.gwent.data.Ability::BINDING</a> ability to the respective squad size.
------------------	--

Definition at line 41 of file `DamageCalculatorBuilder.java`.

References `DamageCalculatorBuilder.calculator`.

#### 4.18.2.3 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 <a href="#">com.paternaggschga.gwent.data.Ability::HORN</a> ability.
----------------	--

Definition at line 62 of file `DamageCalculatorBuilder.java`.

References `DamageCalculatorBuilder.calculator`.

#### 4.18.2.4 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 <a href="#">com.paternaggschga.gwent.data.Ability::MORAL_BOOST</a> ability.
----------------	---

Definition at line 51 of file `DamageCalculatorBuilder.java`.

References DamageCalculatorBuilder.calculator.

#### 4.18.2.5 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().

### 4.18.3 Member Data Documentation

#### 4.18.3.1 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().

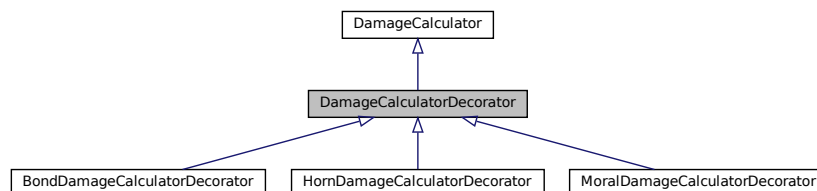
The documentation for this class was generated from the following file:

- gwent/domain/damage/DamageCalculatorBuilder.java

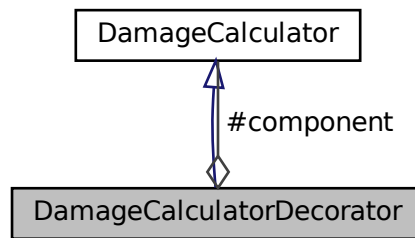
## 4.19 DamageCalculatorDecorator Class Reference

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

### 4.19.1 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`.

### 4.19.2 Constructor & Destructor Documentation

#### 4.19.2.1 DamageCalculatorDecorator()

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

Constructor of a [DamageCalculatorDecorator](#).

#### Parameters

<i>component</i>	<a href="#">DamageCalculator</a> that is being decorated by this decorator.
------------------	---

Definition at line 20 of file `DamageCalculatorDecorator.java`.

References `DamageCalculatorDecorator.component`.

### 4.19.3 Member Data Documentation

#### 4.19.3.1 component

final `DamageCalculator` component [protected]

A `DamageCalculator` which is decorated by this decorator.

Definition at line 13 of file `DamageCalculatorDecorator.java`.

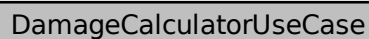
Referenced by `BondDamageCalculatorDecorator.BondDamageCalculatorDecorator()`, `DamageCalculatorDecorator.DamageCalculatorDecorator()`, `HornDamageCalculatorDecorator.HornDamageCalculatorDecorator()`, `MoralDamageCalculatorDecorator.MoralDamageCalculatorDecorator()`, `BondDamageCalculatorDecorator.calculateDamage()`, `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`

## 4.20 DamageCalculatorUseCase Class Reference

A use case class responsible for creating `DamageCalculator` objects for rows using `DamageCalculatorBuildDirector`.  
Collaboration diagram for `DamageCalculatorUseCase`:



```

classDiagram
    class 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.*

#### 4.20.1 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`.

#### 4.20.2 Member Function Documentation

#### 4.20.2.1 `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.  
Referenced by CardUiStateFactory.createCardUiState().

#### 4.20.2.2 `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.  
References [DamageCalculatorBuildDirector.getCalculator\(\)](#).  
The documentation for this class was generated from the following file:

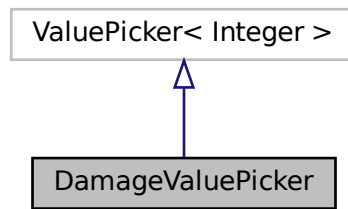
- gwent/domain/cases/DamageCalculatorUseCase.java

## 4.21 DamageValuePicker Class Reference

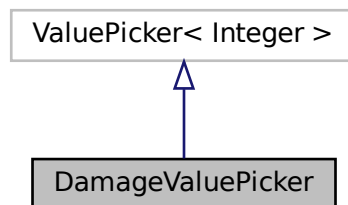
A ValuePicker used for selection of damage values for [UnitEntity::damage](#).



Inheritance diagram for DamageValuePicker:



Collaboration diagram for DamageValuePicker:



### 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.*

### 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.*

### 4.21.1 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 23 of file DamageValuePicker.java.

### 4.21.2 Constructor & Destructor Documentation

#### 4.21.2.1 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 <a href="#">DamageValuePicker</a> .
---------------	---

Definition at line 48 of file DamageValuePicker.java.

References [DamageValuePicker.EPIC\\_DAMAGE\\_VALUES](#), and [DamageValuePicker.setEpicValues\(\)](#).

### 4.21.3 Member Function Documentation

#### 4.21.3.1 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 <a href="#">epicValues</a> is true and #displayIntegers does not contain a mapping for the given value.
------------------------------	--

Definition at line 66 of file DamageValuePicker.java.

References DamageValuePicker.epicValues.

4.21.3.2 `getValue()`

```
Integer getValue ( ) [package]
```

Returns the currently selected value.

## Returns

A value that is selected in #picker.

Definition at line 99 of file DamageValuePicker.java.

References DamageValuePicker.epicValues.

Referenced by CardNumberPickerAdapter.addSelectedUnits().

4.21.3.3 `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 109 of file DamageValuePicker.java.

References DamageValuePicker.EPIC\_DAMAGE\_VALUES, DamageValuePicker.epicValues, and DamageValuePicker.NON\_EPIC\_DAMAGE\_VALUES\_UPPER\_BOUND.

Referenced by CardNumberPickerAdapter.CardNumberPickerAdapter(), and DamageValuePicker.DamageValuePicker().

4.21.3.4 `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 130 of file DamageValuePicker.java.

#### 4.21.3.5 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 <a href="#">epicValues</a> is false and the given value is not in [0,20].
------------------------------	--

##### See also

[ValuePicker::setValue\(Comparable\)](#)

Definition at line 83 of file DamageValuePicker.java.

References [DamageValuePicker.epicValues](#), and [DamageValuePicker.NON\\_EPIC\\_DAMAGE\\_VALUES\\_UPPER\\_BOUND](#).

### 4.21.4 Member Data Documentation

#### 4.21.4.1 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 28 of file DamageValuePicker.java.

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

#### 4.21.4.2 epicValues

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

Boolean defining whether or not this [DamageValuePicker](#) shows epic damage values.

##### See also

[setEpicValues\(boolean\)](#)

Definition at line 39 of file DamageValuePicker.java.

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

#### 4.21.4.3 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 33 of file DamageValuePicker.java.

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

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

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

## 4.22 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.*

### 4.22.1 Detailed Description

An Enum defining which form of Dialog should be shown.

See also

#getDialogType(Context, UnitRepository, Trigger)

Definition at line 146 of file ResetDialogUseCase.java.

### 4.22.2 Member Data Documentation

#### 4.22.2.1 DEFAULT

DEFAULT

A default Dialog asking whether to reset should be invoked.

Definition at line 154 of file ResetDialogUseCase.java.

Referenced by ResetDialogUseCase.getDialogType().

#### 4.22.2.2 MONSTER

MONSTER

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

Definition at line 160 of file ResetDialogUseCase.java.

Referenced by ResetDialogUseCase.reset().

#### 4.22.2.3 NONE

NONE

No Dialog must be invoked.

Definition at line 150 of file ResetDialogUseCase.java.

Referenced by ResetDialogUseCase.getDialogType(), and ResetDialogUseCase.reset().

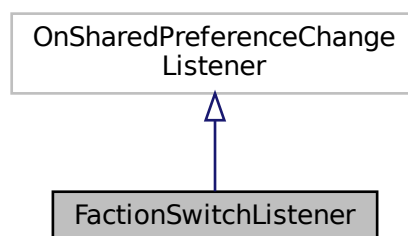
The documentation for this enum was generated from the following file:

- gwent/domain/cases/ResetDialogUseCase.java

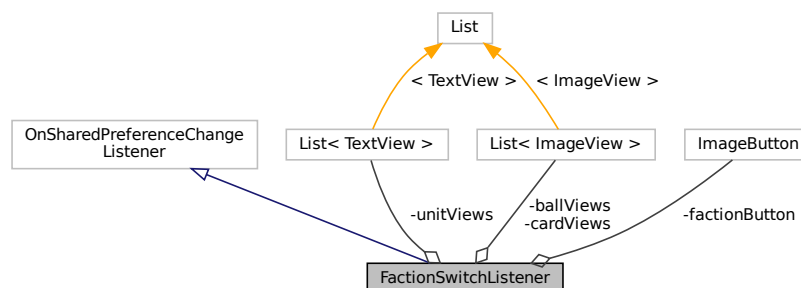
## 4.23 FactionSwitchListener Class Reference

A class implementing `SharedPreferences.OnSharedPreferencesChangeListener` for the `THEME_PREFERENCE_KEY` `SharedPreferences`.

Inheritance diagram for `FactionSwitchListener`:



Collaboration diagram for `FactionSwitchListener`:



### Public Member Functions

- void `onSharedPreferencesChanged` (`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 [onSharedPreferencesChanged\(\)](#) 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.*

#### 4.23.1 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.

#### 4.23.2 Constructor & Destructor Documentation

#### 4.23.2.1 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](#).

Referenced by [FactionSwitchListener.getListener\(\)](#).

### 4.23.3 Member Function Documentation

#### 4.23.3.1 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 <a href="#">FactionSwitchListener</a> .
-------------------	--

##### Returns

A [FactionSwitchListener](#) instance for the given Window.

Definition at line 124 of file FactionSwitchListener.java.

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

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

#### 4.23.3.2 onSharedPreferencesChanged()

```
void onSharedPreferencesChanged (
    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](#).

#### 4.23.3.3 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\\_NORTHERN\\_KINGDOMS](#), [FactionSwitchListener.THEME\\_PREFERENCE\\_KEY](#), and [FactionSwitchListener.THEME\\_SCOIATAEL](#).

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

### 4.23.4 Member Data Documentation

#### 4.23.4.1 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\(\)](#).

#### 4.23.4.2 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.onSharedPreferencesChanged().

#### 4.23.4.3 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.onSharedPreferencesChanged().

#### 4.23.4.4 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.onSharedPreferencesChanged(), and FactionSwitchListener.setTheme().

#### 4.23.4.5 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.onSharedPreferencesChanged(), and FactionSwitchListener.setTheme().

#### 4.23.4.6 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.onSharedPreferencesChanged(), and FactionSwitchListener.setTheme().

#### 4.23.4.7 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().

#### 4.23.4.8 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().

#### 4.23.4.9 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().

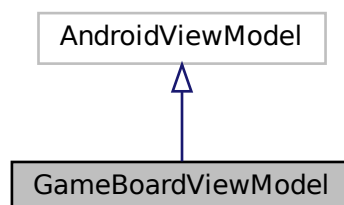
The documentation for this class was generated from the following file:

- gwent/ui/main/FactionSwitchListener.java

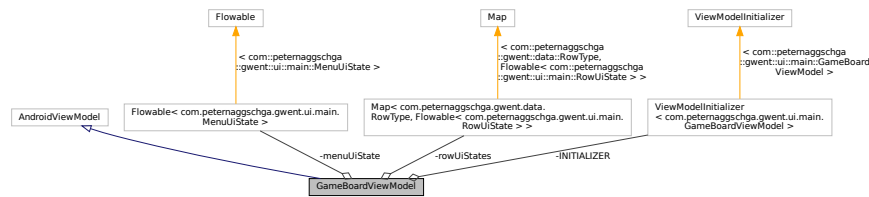
## 4.24 GameBoardViewModel Class Reference

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.*
- Single< Boolean > [onWeatherViewPressed](#) (@NonNull RowType row)  
*Updates the weather debuff of the given row.*
- Single< Boolean > [onHornViewPressed](#) (@NonNull RowType row)  
*Updates the horn buff of the given row.*
- Single< Boolean > [onResetButtonPressed](#) (@NonNull Context context)  
*Triggers a reset and possibly an alert dialog, depending on preferences.*
- Single< Boolean > [onFactionSwitchReset](#) (@NonNull Context context)  
*Triggers a reset and possibly an alert dialog, depending on preferences.*
- Completable [onWeatherButtonPressed](#) ()  
*Clears all weather effects.*
- Single< Boolean > [onBurnButtonPressed](#) (@NonNull Context context)  
*Clears units with the most damage.*

## Static Public Member Functions

- static [GameBoardViewModel getModel](#) (@NonNull ViewModelStoreOwner owner, @NonNull UnitRepository repository)  
*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.*
- Single< Boolean > [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<>().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.*

## Static Private Attributes

- static final ViewModelInitializer< [GameBoardViewModel](#) > [INITIALIZER](#)

*ViewModelInitializer used by androidx.lifecycle.ViewModelProvider.Factory to instantiate the class.*

### 4.24.1 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 37 of file GameBoardViewModel.java.

### 4.24.2 Constructor & Destructor Documentation

#### 4.24.2.1 GameBoardViewModel()

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

Constructor of a [GameBoardViewModel](#) object.

Should only be called in #initializer.

#### Parameters

<i>application</i>	<a href="#">GwentApplication</a> that uses this AndroidViewModel.
--------------------	---

#### See also

[INITIALIZER](#)

Definition at line 59 of file GameBoardViewModel.java.

### 4.24.3 Member Function Documentation

#### 4.24.3.1 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 176 of file GameBoardViewModel.java.

#### 4.24.3.2 getModel()

```
static GameBoardViewModel getModel (
    @NonNull ViewModelStoreOwner owner,
    @NonNull UnitRepository repository ) [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 <a href="#">GameBoardViewModel</a> .
<i>repository</i>	UnitRepository where Flowables are retrieved.

## Returns

A new [GameBoardViewModel](#) instance.

## See also

[ViewModelProvider::ViewModelProvider\(ViewModelStoreOwner, ViewModelProvider.Factory\)](#)

Definition at line 91 of file [GameBoardViewModel.java](#).

References [GameBoardViewModel.INITIALIZER](#), [GameBoardViewModel.reset\(\)](#), and [GameBoardViewModel.rowUiStates](#).

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

**4.24.3.3 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 152 of file [GameBoardViewModel.java](#).

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

**4.24.3.4 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 164 of file [GameBoardViewModel.java](#).

**4.24.3.5 onBurnButtonPressed()**

```
Single<Boolean> 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.

**Parameters**

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

**Returns**

A Single emitting a Boolean defining whether the units were actually removed.

**See also**

BurnDialogUseCase::burn(Context)

Definition at line 267 of file GameBoardViewModel.java.

References BurnDialogUseCase.burn().

**4.24.3.6 onFactionSwitchReset()**

```
Single<Boolean> 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\(\)](#).

**Parameters**

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

**Returns**

A Single emitting a Boolean defining whether a reset was actually conducted.

**See also**

[reset](#)(Context, ResetDialogUseCase.Trigger)

Definition at line 230 of file GameBoardViewModel.java.

References ResetDialogUseCase.Trigger.FACTION\_SWITCH.

Referenced by MainActivity.inflateFactionPopup().

**4.24.3.7 onHornViewPressed()**

```
Single<Boolean> onHornViewPressed (
    @NonNull RowType row )
```

Updates the horn buff of the given row.

Flips between on and off.

**Parameters**

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

**Returns**

A Single emitting a Boolean defining the horn status of the row after the operation.

**See also**

[UnitRepository::switchHorn\(RowType\)](#)

Definition at line 201 of file GameBoardViewModel.java.

Referenced by MainActivity.initializeViewModel().

**4.24.3.8 onResetButtonPressed()**

```
Single<Boolean> 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\(\)](#).

**Parameters**

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

**Returns**

A Single emitting a Boolean defining whether a reset was actually conducted.

**See also**

[reset\(Context, ResetDialogUseCase.Trigger\)](#)

[ResetDialogUseCase.Trigger::BUTTON\\_CLICK](#)

Definition at line 217 of file GameBoardViewModel.java.

References ResetDialogUseCase.Trigger.BUTTON\_CLICK.

**4.24.3.9 onWeatherButtonPressed()**

```
Completable onWeatherButtonPressed ( )
```

Clears all weather effects.

Should only be called by the View.OnClickListener of the weather button.

**Returns**

A Completable tracking operation status.

**See also**

[UnitRepository::clearWeather\(\)](#)

Definition at line 253 of file GameBoardViewModel.java.

References UnitRepository.clearWeather().

**4.24.3.10 onWeatherViewPressed()**

```
Single<Boolean> onWeatherViewPressed (
    @NonNull RowType row )
```

Updates the weather debuff of the given row.

Flips between good and bad weather.



## Parameters

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

## Returns

A Single emitting a Boolean defining the weather status of the row after the operation.

## See also

`UnitRepository::switchWeather(RowType)`

Definition at line 188 of file `GameBoardViewModel.java`.

Referenced by `MainActivity.initializeViewModel()`.

## 4.24.3.11 reset()

```
Single<Boolean> reset (
    @NonNull Context context,
    @NonNull ResetDialogUseCase.Trigger trigger ) [private]
```

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

## Parameters

<i>context</i>	Context object used to acquire SharedPreferences and inflate Dialog views.
<i>trigger</i>	<a href="#">com.paternaggschga.gwent.domain.cases.ResetDialogUseCase.Trigger</a> defining which action triggered the reset.

## Returns

A Single emitting a Boolean defining whether a reset was actually conducted.

## See also

[reset](#)(Context, ResetDialogUseCase.Trigger)

Definition at line 242 of file `GameBoardViewModel.java`.

References `ResetDialogUseCase.reset()`.

Referenced by `GameBoardViewModel.getModel()`.

## 4.24.4 Member Data Documentation

## 4.24.4.1 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 44 of file `GameBoardViewModel.java`.

Referenced by `GameBoardViewModel.getModel()`.

**4.24.4.2 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 79 of file `GameBoardViewModel.java`.

**4.24.4.3 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 70 of file `GameBoardViewModel.java`.

Referenced by `GameBoardViewModel.getModel()`.

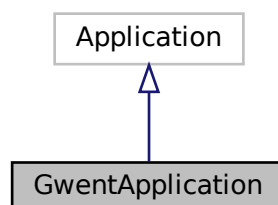
The documentation for this class was generated from the following file:

- `gwent/ui/main/GameBoardViewModel.java`

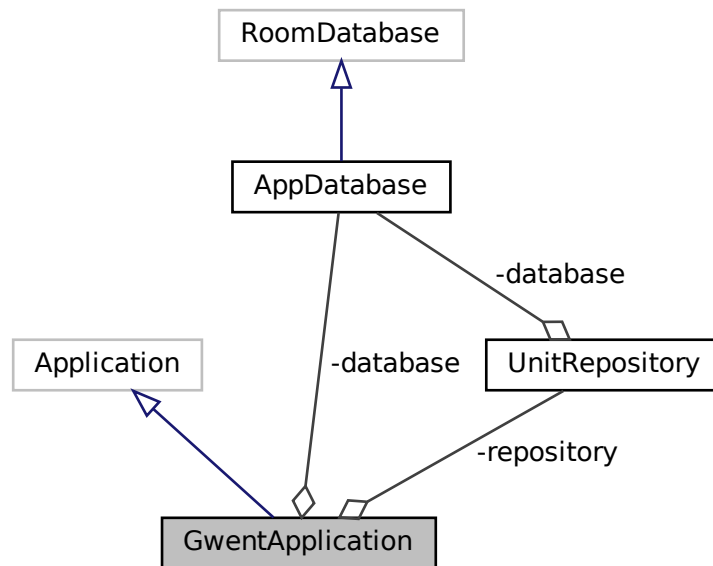
**4.25 GwentApplication Class Reference**

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](#).*

### 4.25.1 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](#).

## 4.25.2 Member Function Documentation

### 4.25.2.1 `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`, `UnitRepository.getRepository()`, and `GwentApplication.repository`.

### 4.25.2.2 `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

<code>context</code>	Context of a <a href="#">GwentApplication</a> .
----------------------	---

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.onCreate()`, `AddCardDialog.onCreate()`, and `ResetDialogUseCase.reset()`.

### 4.25.2.3 `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`.

## 4.25.3 Member Data Documentation

#### 4.25.3.1 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()`.

#### 4.25.3.2 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()`.

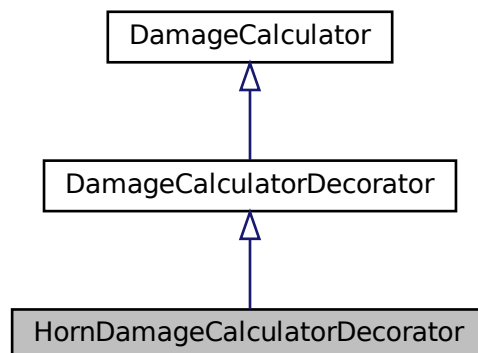
The documentation for this class was generated from the following file:

- `gwent/GwentApplication.java`

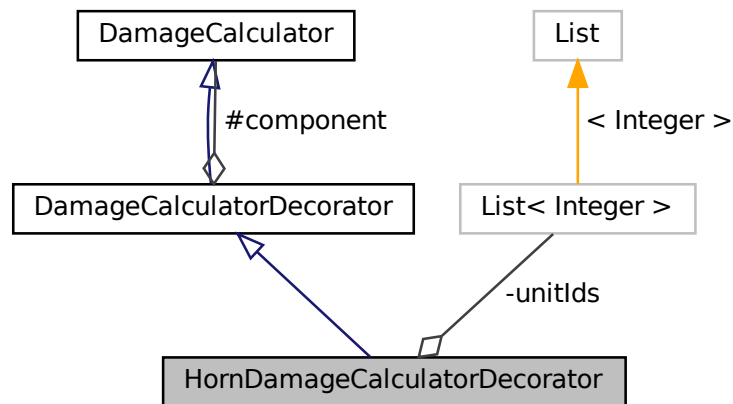
## 4.26 HornDamageCalculatorDecorator Class Reference

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`.*

## 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

### 4.26.1 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`.

## 4.26.2 Constructor & Destructor Documentation

### 4.26.2.1 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>	<a href="#">DamageCalculator</a> that is being decorated by this decorator.
<i>unitIds</i>	List of Integers representing ids of units with the <a href="#">com.paternaggschga.gwent.data.Ability::HORN</a> ability.

#### See also

[DamageCalculatorBuilder](#)

Definition at line 38 of file `HornDamageCalculatorDecorator.java`.

References `DamageCalculatorDecorator.component`, and `HornDamageCalculatorDecorator.unitIds`.

## 4.26.3 Member Function Documentation

### 4.26.3.1 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 <a href="#">UnitEntity::id</a> 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.doubleDamage\(\)](#).

#### 4.26.3.2 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 <a href="#">UnitEntity::id</a> 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\(\)](#).

#### 4.26.3.3 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 <a href="#">UnitEntity::id</a> 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\(\)](#).

### 4.26.4 Member Data Documentation

#### 4.26.4.1 unitIds

```
final List<Integer> unitIds [private]
```

A List of Integers containing the ids of all units with the [com.peternaggschga.gwent.data.Ability::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.doubleDamage()`.

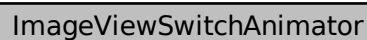
The documentation for this class was generated from the following file:

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

## 4.27 ImageViewSwitchAnimator Class Reference

A class providing functionality for changing the resource shown by an `ImageView` using a fading animation.

Collaboration diagram for `ImageViewSwitchAnimator`:



```

classDiagram
    class 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.*

#### 4.27.1 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`.

#### 4.27.2 Member Function Documentation

##### 4.27.2.1 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>	ImageView, 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.onSharedPreferencesChanged()`.

**4.27.2.2 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>	ImageView 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()`.

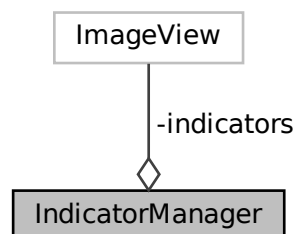
The documentation for this class was generated from the following file:

- `gwent/ui/main/ImageViewSwitchAnimator.java`

**4.28 IndicatorManager Class Reference**

A class managing the indicator views shown in the bottom bar of the [IntroductionActivity](#).

Collaboration diagram for `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 [ImageView](#)s that contains the indicator views.*
- int [currentView](#) = 0  
*Integer referencing the position, i.e.*

### 4.28.1 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](#).

### 4.28.2 Constructor & Destructor Documentation

#### 4.28.2.1 IndicatorManager()

```
IndicatorManager (
    @NonNull Window window ) [package]
```

Constructor of an [IndicatorManager](#) for the given Window. Initializes values in [indicators](#).

#### Parameters

<i>window</i>	Window 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](#).

### 4.28.3 Member Function Documentation

#### 4.28.3.1 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.

## 4.28.4 Member Data Documentation

### 4.28.4.1 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().

### 4.28.4.2 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().

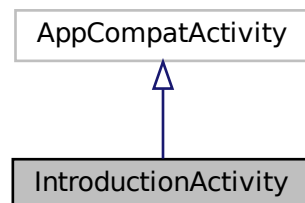
The documentation for this class was generated from the following file:

- gwent/ui/introduction/IndicatorManager.java

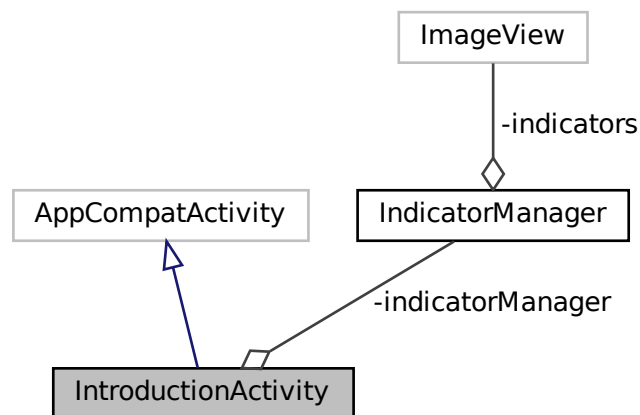
## 4.29 IntroductionActivity Class Reference

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.*

## 4.29.1 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`.

## 4.29.2 Member Function Documentation

### 4.29.2.1 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 Bundle contains the data it most recently supplied in <code>onSaveInstanceState</code> . <b>Note: Otherwise it is null.</b>
---------------------------	--

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 callback for onBack↔Pressed 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`.

## 4.29.3 Member Data Documentation

### 4.29.3.1 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()`.

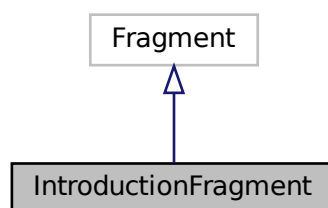
The documentation for this class was generated from the following file:

- `gwent/ui/introduction/IntroductionActivity.java`

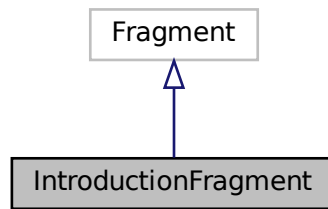
## 4.30 IntroductionFragment Class Reference

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.*

### 4.30.1 Detailed Description

A Fragment containing an `ImageView` and a `TextView` conveying introductory information.  
Definition at line 21 of file `IntroductionFragment.java`.

### 4.30.2 Member Function Documentation

#### 4.30.2.1 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, <a href="#">PAGES_COUNT</a> - 1].
---------------------------------	--

Definition at line 50 of file IntroductionFragment.java.

References [IntroductionFragment.ARG\\_SECTION\\_NUMBER](#), and [IntroductionFragment.PAGES\\_COUNT](#).

Referenced by [SectionsPagerAdapter.createFragment\(\)](#).

## 4.30.2.2 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 LayoutInflater 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 LayoutParams 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](#).

## 4.30.3 Member Data Documentation

## 4.30.3.1 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\(\)](#).

## 4.30.3.2 imageIds

```
final int [] imageIds [static], [private]
```



## 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 [soundManager](#), [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\(\)](#).*
- [SoundManager](#) [soundManager](#)  
*SoundManager used for sound effects on certain events.*
- SharedPreferences.OnSharedPreferenceChangeListener [factionSwitchListener](#)  
*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#THEME\\_PREFERENCE\\_KEY](#), is changed.*
- [GameBoardViewModel](#) [gameBoardViewModel](#)  
*[GameBoardViewModel](#) holding the ui state of this activity.*

### 4.31.1 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.

### 4.31.2 Member Function Documentation

#### 4.31.2.1 inflateCoinFlipPopup()

```
void inflateCoinFlipPopup ( ) [private]
```

Shows a new CoinFlipDialog and plays a coin-flip sound using [SoundManager#playCoinSound\(\)](#).

Definition at line 308 of file MainActivity.java.

References [SoundManager.playCoinSound\(\)](#).

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

#### 4.31.2.2 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 278 of file `MainActivity.java`.

References `GameBoardViewModel.onFactionSwitchReset()`.

Referenced by `MainActivity.onCreate()`.

#### 4.31.2.3 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 197 of file `MainActivity.java`.

References `MainActivity.disposables`, `MainActivity.gameBoardViewModel`, `GameBoardViewModel.onHornViewPressed()`, and `GameBoardViewModel.onWeatherViewPressed()`.

#### 4.31.2.4 onCreate()

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

Sets the theme and layout, initializes [soundManager](#), [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> . <b>Note: Otherwise it is null.</b>
---------------------------	--

Definition at line 88 of file `MainActivity.java`.

References `MainActivity.disposables`, `MainActivity.factionSwitchListener`, `MainActivity.gameBoardViewModel`, `FactionSwitchListener.getListener()`, `GameBoardViewModel.getModel()`, `GwentApplication.getRepository()`, `MainActivity.inflateCoinFlipPopup()`, `MainActivity.inflateFactionPopup()`, `FactionSwitchListener.setTheme()`, and `MainActivity.soundManager`.

#### 4.31.2.5 onDestroy()

```
void onDestroy ( ) [protected]
```

Called when the activity is destroyed.

Disposes and clears all `io.reactivex.rxjava3.disposables.Disposables` in [disposables](#).

Definition at line 186 of file `MainActivity.java`.

References `MainActivity.disposables`.

#### 4.31.2.6 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 130 of file MainActivity.java.

#### 4.31.2.7 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 160 of file MainActivity.java.

### 4.31.3 Member Data Documentation

#### 4.31.3.1 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.initializeViewModel()`, `MainActivity.onCreate()`, and `MainActivity.onDestroy()`.

#### 4.31.3.2 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.paternaggschga.gwent.ui.main.FactionSwitchListener#THEME_PREFERENCE_KEY`, 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 66 of file MainActivity.java.

Referenced by `MainActivity.onCreate()`.

#### 4.31.3.3 gameBoardViewModel

```
GameBoardViewModel gameBoardViewModel [private]
```

`GameBoardViewModel` holding the ui state of this activity.

Definition at line 71 of file MainActivity.java.

Referenced by `MainActivity.initializeViewModel()`, and `MainActivity.onCreate()`.

#### 4.31.3.4 soundManager

```
SoundManager soundManager [private]
```

`SoundManager` used for sound effects on certain events.

Definition at line 57 of file MainActivity.java.

Referenced by MainActivity.onCreate().

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

- gwent/ui/main/MainActivity.java

## 4.32 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.*

#### 4.32.1 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.

#### 4.32.2 Constructor & Destructor Documentation

#### 4.32.2.1 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](#).

### 4.32.3 Member Function Documentation

#### 4.32.3.1 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](#).

#### 4.32.3.2 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](#).

#### 4.32.3.3 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.

#### 4.32.3.4 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.

### 4.32.4 Member Data Documentation

#### 4.32.4.1 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().

#### 4.32.4.2 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().

#### 4.32.4.3 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().

#### 4.32.4.4 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:

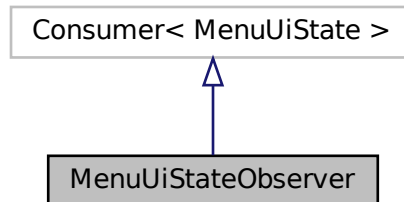
- gwent/ui/main/MenuUiState.java



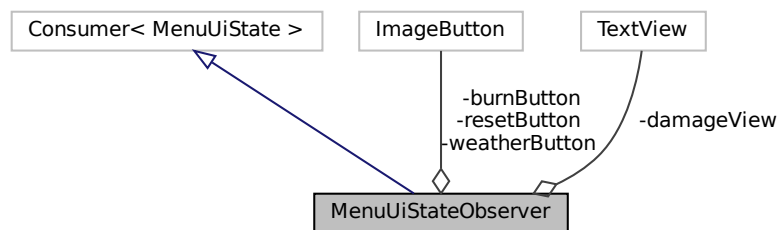
## 4.33 MenuUiStateObserver Class Reference

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.

### 4.33.1 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.

### 4.33.2 Constructor & Destructor Documentation

#### 4.33.2.1 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](#).

### 4.33.3 Member Function Documentation

#### 4.33.3.1 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>	<a href="#">MenuUiState</a> 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](#).

### 4.33.4 Member Data Documentation

#### 4.33.4.1 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().

#### 4.33.4.2 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().

#### 4.33.4.3 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().

#### 4.33.4.4 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().

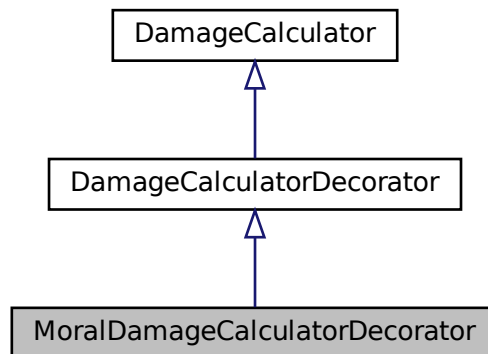
The documentation for this class was generated from the following file:

- gwent/ui/main/MenuUiStateObserver.java

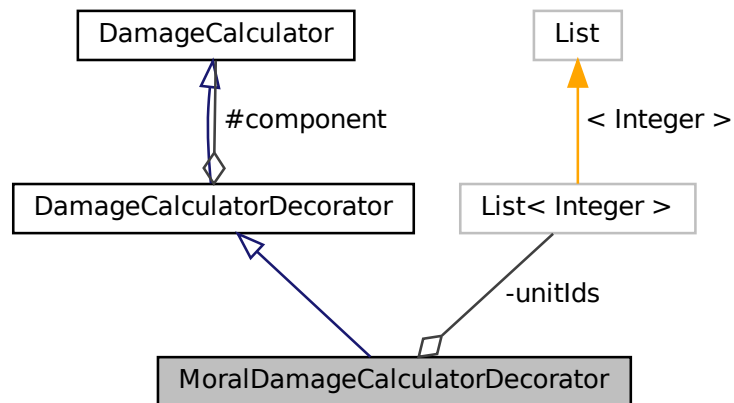
## 4.34 MoralDamageCalculatorDecorator Class Reference

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`.*

## 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

### 4.34.1 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 [WeatherDamageCalculator](#) or a [BondDamageCalculatorDecorator](#) for correct damage calculation. Definition at line 17 of file [MoralDamageCalculatorDecorator.java](#).

### 4.34.2 Constructor & Destructor Documentation

#### 4.34.2.1 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 [WeatherDamageCalculator](#) or [BondDamageCalculatorDecorator](#) for correct damage calculation. The given List must not contain null.

#### Parameters

<i>component</i>	<a href="#">DamageCalculator</a> that is being decorated by this decorator.
<i>unitIds</i>	List of Integers representing ids of units with the <a href="#">com.peternaggschga.gwent.data.Ability::MORAL_BOOST</a> 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](#).

### 4.34.3 Member Function Documentation

#### 4.34.3.1 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.peternaggschga.gwent.data.Ability::MORAL\\_L](#) ability, it is boosted one time less.

## Parameters

<i>id</i>	Integer representing the <a href="#">UnitEntity::id</a> 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.unitIds](#).

## 4.34.3.2 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 <a href="#">UnitEntity::id</a> 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](#).

## 4.34.4 Member Data Documentation

## 4.34.4.1 unitIds

```
final List<Integer> unitIds [private]
```

A List of Integers containing the ids of all units with the [com.paternaggschga.gwent.data.Ability::MORAL\\_BOOST](#) ability.

Definition at line 22 of file MoralDamageCalculatorDecorator.java.

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

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

- gwent/domain/damage/MoralDamageCalculatorDecorator.java

## 4.35 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:



```
graph TD; A[OnValueChangeListener< T extends Comparable];
```

### 4.35.1 Detailed Description

An interface defining a callback for changes of the current value.

Definition at line 211 of file ValuePicker.java.

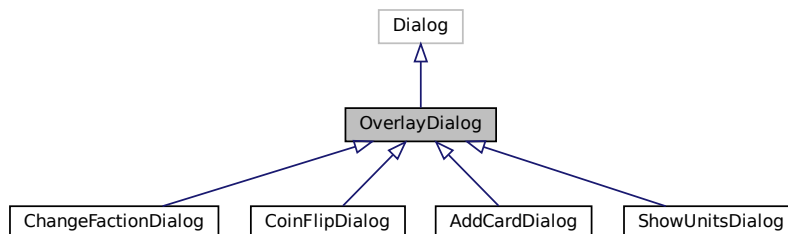
The documentation for this interface was generated from the following file:

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

## 4.36 OverlayDialog Class Reference

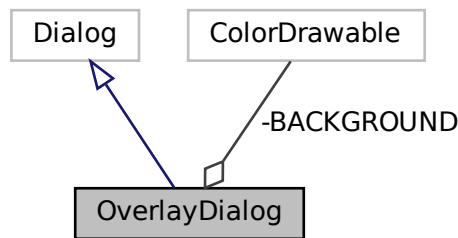
A Dialog class which is used for popups that are shown on top of the calling Activity.

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).*

#### 4.36.1 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.

#### 4.36.2 Constructor & Destructor Documentation

#### 4.36.2.1 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 <a href="#">OverlayDialog</a> .
<i>layout</i>	Integer referencing the layout shown by the created <a href="#">OverlayDialog</a> .
<i>dismissViewId</i>	Integer referencing the cancel view or #NO_CANCEL_VIEW.

Definition at line 54 of file OverlayDialog.java.

References OverlayDialog.dismissViewId, and OverlayDialog.layout.

#### 4.36.2.2 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 <a href="#">OverlayDialog</a> .
<i>layout</i>	Integer referencing the layout shown by the created <a href="#">OverlayDialog</a> .

Definition at line 68 of file OverlayDialog.java.

References OverlayDialog.layout, and OverlayDialog.NO\_DISMISS\_VIEW.

### 4.36.3 Member Function Documentation

#### 4.36.3.1 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 onSaveInstanceState, or null if this is the first time.
---------------------------	--

See also

`#setContentView(int)`  
`#setCancelable(boolean)`

Reimplemented in [CoinFlipDialog](#), [ChangeFactionDialog](#), [ShowUnitsDialog](#), and [AddCardDialog](#).

Definition at line 85 of file `OverlayDialog.java`.

References `OverlayDialog.BACKGROUND`, `OverlayDialog.dismissViewId`, `OverlayDialog.layout`, and `OverlayDialog.NO_DISMISS_VIEW`.

## 4.36.4 Member Data Documentation

### 4.36.4.1 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()`.

### 4.36.4.2 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()`.

### 4.36.4.3 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()`, and `OverlayDialog.onCreate()`.

### 4.36.4.4 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`

## 4.37 RemoveUnitsUseCase Class Reference

A use case class responsible for removing units from a `UnitRepository`.  
 Collaboration diagram for `RemoveUnitsUseCase`:

RemoveUnitsUseCase

### Static Public Member Functions

- static Completable [remove](#) (@NonNull Context context, @NonNull [UnitRepository](#) repository, @NonNull Collection< [UnitEntity](#) > units)  
*Removes the given UnitEntity objects from the given UnitRepository.*
- static Completable [remove](#) (@NonNull Context context, @NonNull [UnitRepository](#) repository, int id)  
*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)  
*Creates a Dialog asking whether the [Ability::REVENGE](#) ability should be activated.*

#### 4.37.1 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 26 of file `RemoveUnitsUseCase.java`.

#### 4.37.2 Member Function Documentation

##### 4.37.2.1 getRevengeDialog()

```
static Dialog getRevengeDialog (
    @NonNull Context context,
    @NonNull UnitRepository repository,
    @NonNull CompletableEmitter emitter,
    @NonNull Collection< UnitEntity > units,
    @IntRange(from=1) int revengeUnits ) [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.
----------------	------------------------------

## Parameters

<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.

## Returns

A Dialog asking whether the [Ability::REVENGE](#) ability should be activated.

## See also

RevengeAlertDialogBuilderAdapter::insertAvengers(UnitRepository, int)

Definition at line 86 of file RemoveUnitsUseCase.java.

References [RevengeAlertDialogBuilderAdapter.create\(\)](#), [RevengeAlertDialogBuilderAdapter.insertAvengers\(\)](#), and [RevengeAlertDialogBuilderAdapter.setPositiveCallback\(\)](#).

Referenced by [RemoveUnitsUseCase.remove\(\)](#).

## 4.37.2.2 remove() [1/2]

```
static Completable remove (
    @NonNull Context context,
    @NonNull UnitRepository repository,
    @NonNull Collection< UnitEntity > units ) [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.

## Returns

A Completable tracking operation status.

## Exceptions

<i>NullPointerException</i>	When units contains a null value.
-----------------------------	-----------------------------------

## See also

[#getRevengeDialog\(Context, UnitRepository, CompletableEmitter, Collection, int\)](#)

[UnitRepository::delete\(Collection\)](#)

Definition at line 41 of file RemoveUnitsUseCase.java.

References [RemoveUnitsUseCase.getRevengeDialog\(\)](#), and [Ability.REVENGE](#).

Referenced by [ShowUnitsDialog.getDialog\(\)](#).

### 4.37.2.3 remove() [2/2]

```
static CompletableFuture remove (
    @NonNull Context context,
    @NonNull UnitRepository repository,
    int id ) [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 CompletableFuture tracking operation status.

#### See also

#remove(Context, UnitRepository, Collection)

Definition at line 66 of file RemoveUnitsUseCase.java.

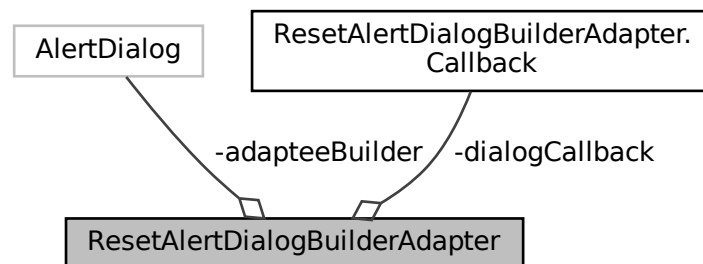
The documentation for this class was generated from the following file:

- gwent/domain/cases/RemoveUnitsUseCase.java

## 4.38 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.*

### 4.38.1 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](#).

### 4.38.2 Constructor & Destructor Documentation

#### 4.38.2.1 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>	<a href="#">Callback</a> used to propagate the user's answer to the creator.

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

References [ResetAlertDialogBuilderAdapter.adapteeBuilder](#), [ResetAlertDialogBuilderAdapter.dialogCallback](#), and [ResetAlertDialogBuilderAdapter.Callback.reset\(\)](#).

### 4.38.3 Member Function Documentation

#### 4.38.3.1 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()`.

#### 4.38.3.2 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()`.

#### 4.38.3.3 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`.

### 4.38.4 Member Data Documentation

#### 4.38.4.1 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.ResetAlertDialogBuilderAdapter()`, `ResetAlertDialogBuilderAdapter.create()`, `ResetAlertDialogBuilderAdapter.setMonsterDialog()`, and `ResetAlertDialogBuilderAdapter.setTrigger()`.



#### 4.38.4.2 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.setMonsterDialog()`.

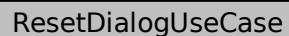
The documentation for this class was generated from the following file:

- gwent/domain/cases/ResetAlertDialogBuilderAdapter.java

## 4.39 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:



```

classDiagram
    class 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)  
*Resets the given UnitRepository.*

### Static Protected Member Functions

- static Single< Boolean > [reset](#) (@NonNull Context context, @NonNull [UnitRepository](#) repository, @NonNull [Trigger](#) trigger)  
*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.*

#### 4.39.1 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 30 of file ResetDialogUseCase.java.

## 4.39.2 Member Function Documentation

### 4.39.2.1 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 <a href="#">DialogType</a> is even necessary.
<i>trigger</i>	<a href="#">Trigger</a> defining what triggered the reset.

Returns

A [DialogType](#) defining the kind of Dialog.

See also

[DialogType](#)

Definition at line 100 of file ResetDialogUseCase.java.

References `ResetDialogUseCase.DialogType.DEFAULT`, and `ResetDialogUseCase.DialogType.NONE`.

Referenced by `ResetDialogUseCase.reset()`.

### 4.39.2.2 reset() [1/2]

```
static Single<Boolean> reset (  
    @NonNull Context context,  
    @NonNull Trigger trigger ) [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>	<a href="#">Trigger</a> defining what triggered this reset.

Returns

A Single emitting a Boolean defining whether the reset really took place.

See also

`#reset(Context, UnitRepository, Trigger)`

`ResetRepositoryUseCase::reset(Context, UnitRepository, boolean)`

Definition at line 45 of file `ResetDialogUseCase.java`.

References `GwentApplication.getRepository()`.

Referenced by `GameBoardViewModel.reset()`.

#### 4.39.2.3 reset() [2/2]

```
static Single<Boolean> reset (
    @NonNull Context context,
    @NonNull UnitRepository repository,
    @NonNull Trigger trigger ) [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>	<code>UnitRepository</code> that is reset.
<i>trigger</i>	<code>Trigger</code> defining what triggered this reset.

Returns

A `Single` emitting a `Boolean` defining whether the reset really took place.

See also

`ResetRepositoryUseCase::reset(Context, UnitRepository, boolean)`

Definition at line 64 of file `ResetDialogUseCase.java`.

References `ResetAlertDialogBuilderAdapter.create()`, `ResetDialogUseCase.getDialogType()`, `ResetDialogUseCase.DialogType.MONSTER`, `ResetDialogUseCase.DialogType.NONE`, and `ResetRepositoryUseCase.reset()`.

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

- `gwent/domain/cases/ResetDialogUseCase.java`

## 4.40 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`)  
*Resets the given UnitRepository.*

## Static Package Functions

- static Maybe< [UnitEntity](#) > [reset](#) (@NonNull Context context, @NonNull [UnitRepository](#) repository, boolean keepUnit)

*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)  
*Creates a Dialog asking whether the [Ability::REVENGE](#) ability should be activated.*

### 4.40.1 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 30 of file ResetRepositoryUseCase.java.

### 4.40.2 Member Function Documentation

#### 4.40.2.1 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 69 of file ResetRepositoryUseCase.java.

Referenced by ResetRepositoryUseCase.reset().

#### 4.40.2.2 getRevengeDialog()

```
static Dialog getRevengeDialog (
    @NonNull Context context,
    @NonNull UnitRepository repository,
    @NonNull CompletableEmitter emitter,
    @Nullable UnitEntity keptUnit,
    @IntRange(from=1) int revengeUnits ) [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.
<i>revengeUnits</i>	Long representing the number of revenge units.

## Returns

A Dialog asking whether the [Ability::REVENGE](#) ability should be activated.

## See also

RevengeAlertDialogBuilderAdapter::insertAvengers(UnitRepository, int)

Definition at line 90 of file ResetRepositoryUseCase.java.

References RevengeAlertDialogBuilderAdapter.create(), RevengeAlertDialogBuilderAdapter.insertAvengers(), and RevengeAlertDialogBuilderAdapter.setPositiveCallback().

Referenced by ResetRepositoryUseCase.reset().

## 4.40.2.3 reset() [1/2]

```
static Completable reset (
    @NonNull Context context,
    @NonNull UnitRepository repository ) [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).

## Parameters

<i>context</i>	Context of the shown Dialog.
<i>repository</i>	UnitRepository that is being reset.

## Returns

A Completable tracking operation status.

## See also

#reset(Context, UnitRepository, boolean)

Definition at line 119 of file ResetRepositoryUseCase.java.

References ResetRepositoryUseCase.reset().

## 4.40.2.4 reset() [2/2]

```
static Maybe<UnitEntity> reset (
    @NonNull Context context,
    @NonNull UnitRepository repository,
    boolean keepUnit ) [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.
<i>repository</i>	UnitRepository that is being reset.
<i>keepUnit</i>	Boolean defining whether a single UnitEntity should be kept.

## Returns

A Maybe emitting the kept UnitEntity or nothing if keepUnit is false.

## See also

`#getRevengeDialog(Context, UnitRepository, CompletableEmitter, UnitEntity, int)`

`UnitRepository::reset(UnitEntity)`

Definition at line 44 of file `ResetRepositoryUseCase.java`.

References `ResetRepositoryUseCase.getRandomUnit()`, `ResetRepositoryUseCase.getRevengeDialog()`, and `Ability.REVENGE`.

Referenced by `ResetRepositoryUseCase.reset()`, and `ResetDialogUseCase.reset()`.

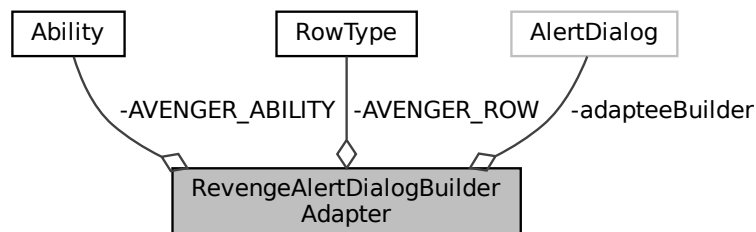
The documentation for this class was generated from the following file:

- `gwent/domain/cases/ResetRepositoryUseCase.java`

## 4.41 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`:



### Static Public Member Functions

- static Completable `insertAvengers` (@NonNull UnitRepository repository, @IntRange(from=0) int number↵ OfAvengers)

*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.*

### 4.41.1 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 22 of file RevengeAlertDialogBuilderAdapter.java.

### 4.41.2 Constructor & Destructor Documentation

#### 4.41.2.1 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 75 of file RevengeAlertDialogBuilderAdapter.java.  
References [RevengeAlertDialogBuilderAdapter.adapteeBuilder](#).

### 4.41.3 Member Function Documentation

#### 4.41.3.1 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 106 of file RevengeAlertDialogBuilderAdapter.java.

References RevengeAlertDialogBuilderAdapter.adapteeBuilder.

Referenced by RemoveUnitsUseCase.getRevengeDialog(), and ResetRepositoryUseCase.getRevengeDialog().

#### 4.41.3.2 insertAvengers()

```
static Completable insertAvengers (
    @NonNull UnitRepository repository,
    @IntRange(from=0) int numberOfAvengers ) [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 95 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().

#### 4.41.3.3 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 131 of file RevengeAlertDialogBuilderAdapter.java.

References RevengeAlertDialogBuilderAdapter.adapteeBuilder.

#### 4.41.3.4 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\(UnitRepository, int\)](#)

Definition at line 119 of file RevengeAlertDialogBuilderAdapter.java.

References [RevengeAlertDialogBuilderAdapter.adapteeBuilder](#).

Referenced by [RemoveUnitsUseCase.getRevengeDialog\(\)](#), and [ResetRepositoryUseCase.getRevengeDialog\(\)](#).

### 4.41.4 Member Data Documentation

#### 4.41.4.1 adapteeBuilder

```
final AlertDialog.Builder adapteeBuilder [private]
```

AlertDialog.Builder that is adapted by this class.

Definition at line 67 of file RevengeAlertDialogBuilderAdapter.java.

Referenced by [RevengeAlertDialogBuilderAdapter.RevengeAlertDialogBuilderAdapter\(\)](#), [RevengeAlertDialogBuilderAdapter.create\(\)](#), [RevengeAlertDialogBuilderAdapter.setNegativeCallback\(\)](#), and [RevengeAlertDialogBuilderAdapter.setPositiveCallback\(\)](#).

#### 4.41.4.2 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 37 of file RevengeAlertDialogBuilderAdapter.java.

Referenced by [RevengeAlertDialogBuilderAdapter.insertAvengers\(\)](#).

#### 4.41.4.3 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 45 of file RevengeAlertDialogBuilderAdapter.java.

Referenced by [RevengeAlertDialogBuilderAdapter.insertAvengers\(\)](#).

#### 4.41.4.4 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 29 of file RevengeAlertDialogBuilderAdapter.java.  
Referenced by RevengeAlertDialogBuilderAdapter.insertAvengers().

#### 4.41.4.5 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 61 of file RevengeAlertDialogBuilderAdapter.java.  
Referenced by RevengeAlertDialogBuilderAdapter.insertAvengers().

#### 4.41.4.6 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 53 of file RevengeAlertDialogBuilderAdapter.java.  
Referenced by RevengeAlertDialogBuilderAdapter.insertAvengers().  
The documentation for this class was generated from the following file:

- gwent/domain/cases/RevengeAlertDialogBuilderAdapter.java

## 4.42 RowDao Interface Reference

An interface defining data access operations related to `rows` table.  
Collaboration diagram for RowDao:



```
graph TD
    RowDao[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`.*

### 4.42.1 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.

### 4.42.2 Member Function Documentation

#### 4.42.2.1 `clearRows()`

Completable `clearRows` ( )  
 Deletes all `RowEntity` objects from `rows`.

##### Returns

A Completable tracking operation status.

Referenced by `UnitRepository.reset()`.

#### 4.42.2.2 `clearWeather()`

Completable `clearWeather` ( )  
 Sets `RowEntity::weather` to `false` for all `RowEntity` elements in `rows`.

##### Returns

A Completable tracking operation status.

Referenced by `UnitRepository.clearWeather()`.

#### 4.42.2.3 `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

<i>row</i>	<a href="#">RowEntity</a> that is being inserted.
------------	---

## Returns

A Completable tracking operation status.

Referenced by `UnitRepository.initializeRows()`.

**4.42.2.4 isHorn()**

```
Single<Boolean> isHorn (  
    @NonNull RowType row )
```

Fetches [RowEntity::horn](#) for the given [RowType](#) in `rows`.

## See also

`#isHornFlowable(RowType)`

## Parameters

<i>row</i>	<a href="#">RowType</a> of the queried <a href="#">RowEntity</a> .
------------	--

## Returns

A Single tracking operation status and returning the value.

Referenced by `UnitRepository.isHorn()`.

**4.42.2.5 isHornFlowable()**

```
Flowable<Boolean> isHornFlowable (  
    @NonNull RowType row )
```

Fetches a Flowable of [RowEntity::horn](#) for the given [RowType](#) in `rows`.

## Parameters

<i>row</i>	<a href="#">RowType</a> of the queried <a href="#">RowEntity</a> .
------------	--

## Returns

A Flowable emitting the values.

## See also

`#isHorn(RowType)`

Referenced by `UnitRepository.isHornFlowable()`.

**4.42.2.6 isWeather()**

```
Single<Boolean> isWeather (  
    @NonNull RowType row )
```

Fetches [RowEntity::weather](#) for the given [RowType](#) in `rows`.

See also

`#isWeatherFlowable(RowType)`

Parameters

<i>row</i>	<a href="#">RowType</a> of the queried <a href="#">RowEntity</a> .
------------	--

Returns

A Single tracking operation status and returning the value.

Referenced by `UnitRepository.isWeather()`.

#### 4.42.2.7 isWeatherFlowable()

```
Flowable<Boolean> isWeatherFlowable (  
    @NonNull RowType row )
```

Fetches a Flowable of [RowEntity::weather](#) for the given [RowType](#) in `rows`.

Parameters

<i>row</i>	<a href="#">RowType</a> of the queried <a href="#">RowEntity</a> .
------------	--

Returns

A Flowable emitting the values.

See also

`#isWeather(RowType)`

Referenced by `UnitRepository.isWeatherFlowable()`.

#### 4.42.2.8 updateHorn()

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

Flips [RowEntity::horn](#) for the given [RowType](#) in `rows`.

Parameters

<i>row</i>	<a href="#">RowType</a> of the updated <a href="#">RowEntity</a> .
------------	--

Returns

A Completable tracking operation status.

Referenced by `UnitRepository.switchHorn()`.

#### 4.42.2.9 updateWeather()

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

Flips [RowEntity::weather](#) for the given [RowType](#) in `rows`.

## Parameters

<i>row</i>	<a href="#">RowType</a> of the updated <a href="#">RowEntity</a> .
------------	--

## Returns

A Completable tracking operation status.

Referenced by `UnitRepository.switchWeather()`.

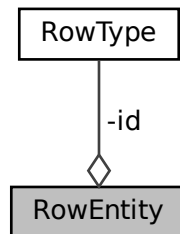
The documentation for this interface was generated from the following file:

- `gwent/data/RowDao.java`

## 4.43 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.*

### 4.43.1 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.

### 4.43.2 Constructor & Destructor Documentation

#### 4.43.2.1 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>	<a href="#">RowType</a> of the represented row.
-----------	---

Definition at line 47 of file RowEntity.java.

### 4.43.3 Member Function Documentation

#### 4.43.3.1 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.

#### 4.43.3.2 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.

#### 4.43.3.3 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.

#### 4.43.3.4 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.

#### 4.43.3.5 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.

### 4.43.4 Member Data Documentation

#### 4.43.4.1 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.

#### 4.43.4.2 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.

#### 4.43.4.3 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.

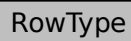
The documentation for this class was generated from the following file:

- gwent/data/RowEntity.java

## 4.44 RowType Enum Reference

An enum listing the different types of attack rows.

Collaboration diagram for RowType:



```
classDiagram
    class 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.*

### 4.44.1 Detailed Description

An enum listing the different types of attack rows.

Definition at line 6 of file RowType.java.

### 4.44.2 Member Data Documentation

#### 4.44.2.1 MELEE

MELEE

Represents the row containing melee combat units.

Definition at line 10 of file RowType.java.

#### 4.44.2.2 RANGE

RANGE

Represents the row containing ranged combat units.

Definition at line 15 of file RowType.java.

#### 4.44.2.3 SIEGE

SIEGE

Represents the row containing siege combat units.

Definition at line 21 of file RowType.java.

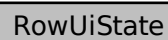
The documentation for this enum was generated from the following file:

- gwent/data/RowType.java

### 4.45 RowUiState Class Reference

A data class encapsulating the visible state of a row in the main view of the application.

Collaboration diagram for RowUiState:



```
graph TD; 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.*

#### 4.45.1 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.

## 4.45.2 Constructor & Destructor Documentation

### 4.45.2.1 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](#).

## 4.45.3 Member Function Documentation

### 4.45.3.1 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](#).

### 4.45.3.2 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](#).

#### 4.45.3.3 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.

#### 4.45.3.4 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.

### 4.45.4 Member Data Documentation

#### 4.45.4.1 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().

#### 4.45.4.2 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().

#### 4.45.4.3 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().

#### 4.45.4.4 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().

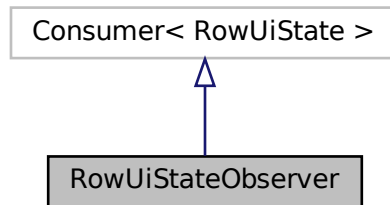
The documentation for this class was generated from the following file:

- gwent/ui/main/RowUiState.java

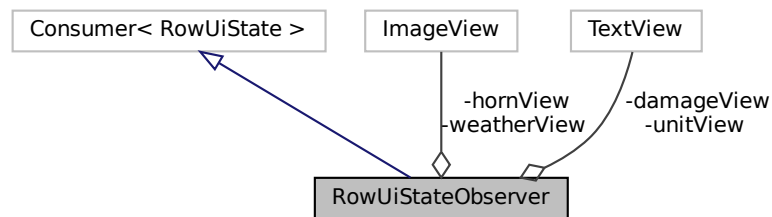
## 4.46 RowUiStateObserver Class Reference

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](#).*

#### 4.46.1 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`.

#### 4.46.2 Constructor & Destructor Documentation

##### 4.46.2.1 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

<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>weatherResource</i>	Integer representing the drawable resource shown when the weather debuff is active.
<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.

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](#).

Referenced by [RowUiStateObserver.getObserver\(\)](#).

#### 4.46.3 Member Function Documentation

#### 4.46.3.1 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>	<a href="#">RowUiState</a> 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](#).

#### 4.46.3.2 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>	<a href="#">RowType</a> defining which row is being observed.
<i>damageView</i>	<a href="#">TextView</a> showing the summed-up damage of all units in the observed row.
<i>weatherView</i>	<a href="#">ImageView</a> showing the current state of the weather debuff of the observed row.
<i>hornView</i>	<a href="#">ImageView</a> showing the current state of the commander's horn buff of the observed row.
<i>unitView</i>	<a href="#">TextView</a> 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.RowUiStateObserver\(\)](#), [RowUiStateObserver.damageView](#), [RowUiStateObserver.hornView](#), [RowUiStateObserver.unitView](#), [RowUiStateObserver.weatherResource](#), and [RowUiStateObserver.weatherView](#).

### 4.46.4 Member Data Documentation

#### 4.46.4.1 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\(\)](#).

#### 4.46.4.2 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\(\)](#).

#### 4.46.4.3 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\(\)](#).

#### 4.46.4.4 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\(\)](#).

#### 4.46.4.5 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:

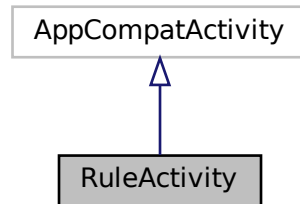
- gwent/ui/main/RowUiStateObserver.java



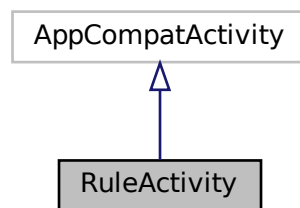
## 4.47 RuleActivity Class Reference

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.*

#### 4.47.1 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.

## 4.47.2 Member Function Documentation

### 4.47.2.1 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 onSaveInstanceState. <b>Note: Otherwise it is null.</b>
---------------------------	--

Definition at line 37 of file RuleActivity.java.

References RuleActivity.INTENT\_EXTRA\_TAG.

### 4.47.2.2 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.

## 4.47.3 Member Data Documentation

### 4.47.3.1 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

## 4.48 RuleSection Enum Reference

An Enum used to discern the different possible sections shown by [RuleActivity](#).

Collaboration diagram for RuleSection:



```
graph TD; 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.*

### 4.48.1 Detailed Description

An Enum used to discern the different possible sections shown by [RuleActivity](#).  
Definition at line 6 of file RuleSection.java.

### 4.48.2 Member Data Documentation

#### 4.48.2.1 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.

#### 4.48.2.2 CARDS

`CARDS`

Represents a section containing information on the different types of cards in Gwent.  
Definition at line 30 of file RuleSection.java.

#### 4.48.2.3 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.

#### 4.48.2.4 COURSE

COURSE

Represents a section containing information on the course of a game of Gwent.  
Definition at line 15 of file RuleSection.java.

#### 4.48.2.5 FACTIONS

FACTIONS

Represents a section containing information on the factions playable in Gwent.  
Definition at line 20 of file RuleSection.java.

#### 4.48.2.6 GENERAL

GENERAL

Represents a section containing general background-information on the game Gwent.  
Definition at line 10 of file RuleSection.java.

#### 4.48.2.7 SPECIAL\_CARDS

SPECIAL\_CARDS

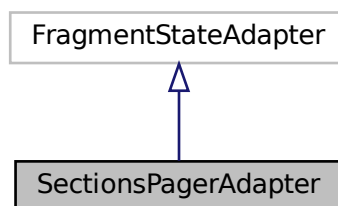
Represents a section containing information on special cards available in Gwent.  
Definition at line 41 of file RuleSection.java.

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

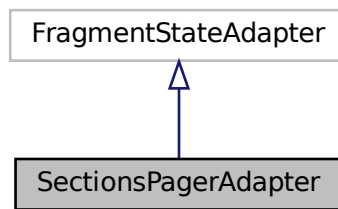
- gwent/ui/settings/RuleSection.java

### 4.49 SectionsPagerAdapter Class Reference

A `FragmentStateAdapter` 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 FragmentActivity fragmentActivity)  
*Constructor of a [SectionsPagerAdapter](#) for the given FragmentActivity.*
- int [getItemCount](#) ()  
*Returns the total number of items in the data set held by the adapter.*
- Fragment [createFragment](#) (int position)  
*Provides a new [IntroductionFragment](#) associated with the specified position.*

### 4.49.1 Detailed Description

A `FragmentStateAdapter` that returns an [IntroductionFragment](#) corresponding to one of the sections/tabs/pages. Definition at line 12 of file `SectionsPagerAdapter.java`.

### 4.49.2 Constructor & Destructor Documentation

#### 4.49.2.1 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`.

### 4.49.3 Member Function Documentation

#### 4.49.3.1 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().

**4.49.3.2 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.

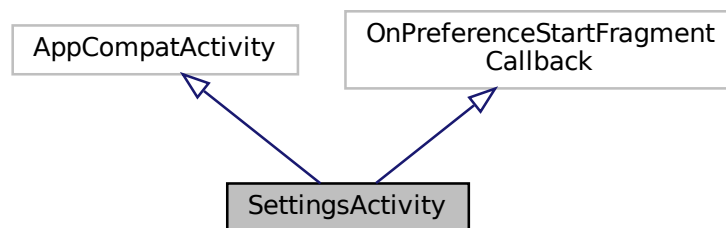
The documentation for this class was generated from the following file:

- gwent/ui/introduction/SectionsPagerAdapter.java

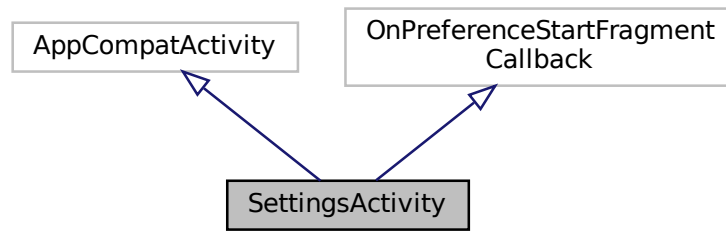
## 4.50 SettingsActivity Class Reference

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](#) (@NonNull 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.*

### 4.50.1 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.

### 4.50.2 Member Function Documentation

#### 4.50.2.1 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 Bundle contains the data it most recently supplied in <a href="#">onSaveInstanceState</a> . <b>Note: Otherwise it is null.</b>
---------------------------	---

Definition at line 38 of file SettingsActivity.java.

References SettingsActivity.CURRENT\_FRAGMENT\_KEY, and FactionSwitchListener.setTheme().

#### 4.50.2.2 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>	MenuItem that was selected.
-------------	-----------------------------

## Returns

Boolean defining whether the call has been handled.

Definition at line 85 of file SettingsActivity.java.

#### 4.50.2.3 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>	PreferenceFragmentCompat requesting navigation.
<i>pref</i>	Preference requesting the Fragment.

## Returns

Boolean defining whether the Fragment creation has been handled.

Definition at line 102 of file SettingsActivity.java.

#### 4.50.2.4 onSaveInstanceState()

```
void onSaveInstanceState (
    @NonNull Bundle outState ) [protected]
```

Saves the currently visible Fragment into the given Bundle.

## Parameters

<i>outState</i>	Bundle in which the currently used Fragment is saved.
-----------------	---



Definition at line 66 of file SettingsActivity.java.

References SettingsActivity.CURRENT\_FRAGMENT\_KEY.

### 4.50.3 Member Data Documentation

#### 4.50.3.1 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().

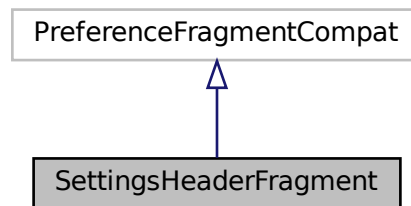
The documentation for this class was generated from the following file:

- gwent/ui/settings/SettingsActivity.java

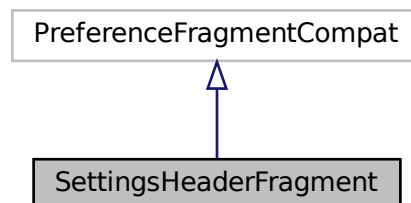
## 4.51 SettingsHeaderFragment Class Reference

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.

### 4.51.1 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.

### 4.51.2 Member Function Documentation

#### 4.51.2.1 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.OnPreferenceClickListener 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 androidx.preference.PreferenceScreen with this key.

Definition at line 31 of file SettingsHeaderFragment.java.

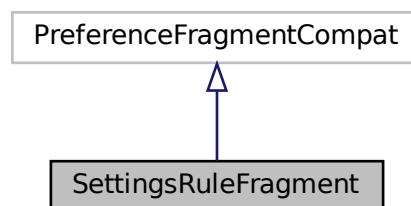
The documentation for this class was generated from the following file:

- gwent/ui/settings/SettingsHeaderFragment.java

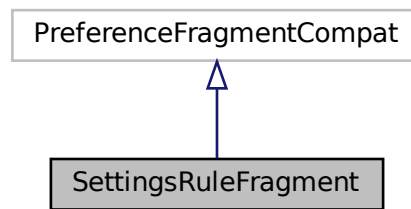
## 4.52 SettingsRuleFragment Class Reference

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.*

### 4.52.1 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.

### 4.52.2 Member Function Documentation

#### 4.52.2.1 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>	<a href="#">RuleSection</a> 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\(\)](#).

#### 4.52.2.2 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 androidx.preference.PreferenceScreen with this key.

##### See also

[#getSectionClickListener\(RuleSection\)](#)

Definition at line 66 of file SettingsRuleFragment.java.

References SettingsRuleFragment.getSectionClickListener().

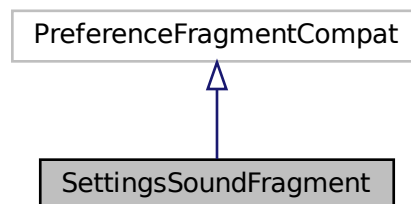
The documentation for this class was generated from the following file:

- gwent/ui/settings/SettingsRuleFragment.java

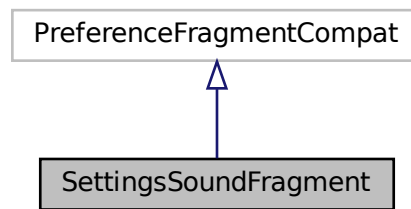
## 4.53 SettingsSoundFragment Class Reference

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.*

### 4.53.1 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.

### 4.53.2 Member Function Documentation

#### 4.53.2.1 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 androidx.preference.PreferenceScreen with this key.

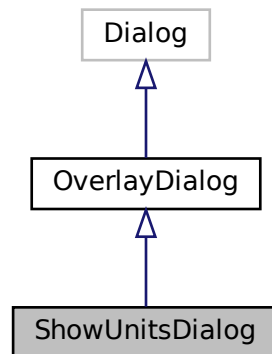
Definition at line 27 of file SettingsSoundFragment.java.  
 The documentation for this class was generated from the following file:

- gwent/ui/settings/SettingsSoundFragment.java

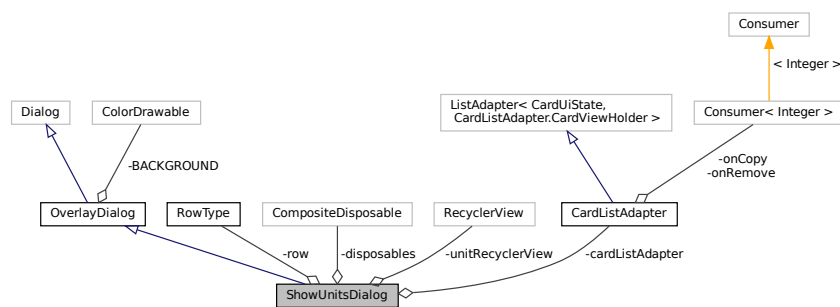
## 4.54 ShowUnitsDialog Class Reference

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)`  
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.

## Private Member Functions

- [ShowUnitsDialog](#) (@NonNull Context context, @NonNull [RowType](#) row, @NonNull [CardListAdapter](#) cardListAdapter)  
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 [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

### 4.54.1 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 29 of file `ShowUnitsDialog.java`.

### 4.54.2 Constructor & Destructor Documentation

#### 4.54.2.1 ShowUnitsDialog()

```
ShowUnitsDialog (
    @NonNull Context context,
    @NonNull RowType row,
    @NonNull CardListAdapter cardListAdapter ) [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>	<a href="#">CardListAdapter</a> providing an always up-to-date list of <a href="#">CardUiState</a> objects for a certain row.

Definition at line 66 of file `ShowUnitsDialog.java`.

References `ShowUnitsDialog.cardListAdapter`, and `ShowUnitsDialog.row`.

Referenced by `ShowUnitsDialog.getDialog()`.

### 4.54.3 Member Function Documentation

#### 4.54.3.1 getDialog()

```
static Single<ShowUnitsDialog> getDialog (
    @NonNull Context context,
    @NonNull RowType row ) [static]
```

Creates a new [ShowUnitsDialog](#) in the given Context and for the given row.

#### Parameters

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

**Parameters**

<i>row</i>	RowType defining the row that is represented by this Dialog.
------------	--

**Returns**

A Single emitting the created [ShowUnitsDialog](#).

Definition at line 80 of file ShowUnitsDialog.java.

References [ShowUnitsDialog.ShowUnitsDialog\(\)](#), [ShowUnitsDialog.disposables](#), [GwentApplication.getRepository\(\)](#), [RemoveUnitsUseCase.remove\(\)](#), and [ShowUnitsDialog.row](#).

**4.54.3.2 getRow()**

[RowType](#) `getRow ( )`

Returns the row this [ShowUnitsDialog](#) is representing.

**Returns**

A RowType defining the represented row.

**See also**

[row](#)

Definition at line 168 of file ShowUnitsDialog.java.

References [ShowUnitsDialog.row](#).

**4.54.3.3 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 <a href="#">onSaveInstanceState</a> , 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 122 of file `ShowUnitsDialog.java`.

References `ShowUnitsDialog.cardListAdapter`, `ShowUnitsDialog.disposables`, and `ShowUnitsDialog.unitRecyclerView`.↵

## 4.54.4 Member Data Documentation

### 4.54.4.1 cardListAdapter

```
final CardListAdapter cardListAdapter [private]
```

[CardListAdapter](#) offering a list of [CardUiState](#) objects to the UI.

Definition at line 42 of file `ShowUnitsDialog.java`.

Referenced by `ShowUnitsDialog.ShowUnitsDialog()`, and `ShowUnitsDialog.onCreate()`.

### 4.54.4.2 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 52 of file `ShowUnitsDialog.java`.

Referenced by `ShowUnitsDialog.getDialog()`, and `ShowUnitsDialog.onCreate()`.

### 4.54.4.3 row

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

`RowType` defining which row is represented by this `Dialog`.

See also

[getRow\(\)](#)

Definition at line 36 of file `ShowUnitsDialog.java`.

Referenced by `ShowUnitsDialog.ShowUnitsDialog()`, `ShowUnitsDialog.getDialog()`, and `ShowUnitsDialog.getRow()`.↵

### 4.54.4.4 unitRecyclerView

```
RecyclerView unitRecyclerView [private]
```

`RecyclerView` presenting the units provided by the [cardListAdapter](#).

Definition at line 57 of file `ShowUnitsDialog.java`.

Referenced by `ShowUnitsDialog.onCreate()`.

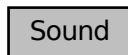
The documentation for this class was generated from the following file:

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

## 4.55 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:



### 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.*

#### 4.55.1 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.

## 4.55.2 Constructor & Destructor Documentation

### 4.55.2.1 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](#).

Referenced by [Sound.createSound\(\)](#).

## 4.55.3 Member Function Documentation

### 4.55.3.1 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 <a href="#">Sound</a> is used.
<i>preferenceRes</i>	Integer referencing the String resource of the preference key for this <a href="#">Sound</a> .
<i>soundPool</i>	SoundPool where the <a href="#">Sound</a> is registered.
<i>resId</i>	Integer referencing the raw resource of the <a href="#">Sound</a> .

#### Returns

A [Sound](#) object referencing the newly registered sound effect.

Definition at line 69 of file Sound.java.

References [Sound.Sound\(\)](#), [Sound.activated](#), [Sound.preferenceKey](#), and [Sound.soundId](#).

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

**4.55.3.2 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.

**4.55.3.3 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.

**4.55.3.4 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.

**4.55.3.5 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().

**4.55.3.6 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 <a href="#">Sound</a> is activated.
------------------	--

See also

`#setActivated(SharedPreferences)`

Definition at line 114 of file Sound.java.

References Sound.activated.

Referenced by Sound.setActivated().

## 4.55.4 Member Data Documentation

### 4.55.4.1 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(), and Sound.setActivated().

### 4.55.4.2 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().

### 4.55.4.3 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().

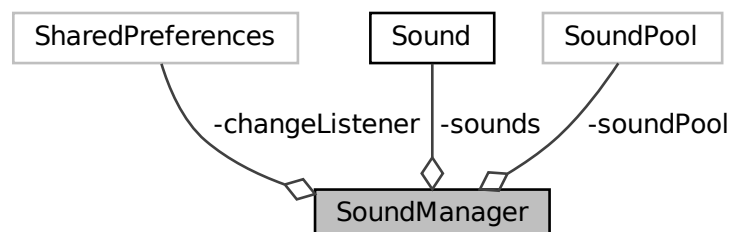
The documentation for this class was generated from the following file:

- gwent/ui/sounds/Sound.java

## 4.56 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.*

### 4.56.1 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](#).

### 4.56.2 Constructor & Destructor Documentation

#### 4.56.2.1 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 [changeListener](#) that updates the [Sound::activated](#) status when sound settings are updated.

Parameters

<i>context</i>	Context, this <a href="#">SoundManager</a> 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](#), [Sound](#)↵  
[Manager.SOUND\\_CARDS\\_EPIC](#), [SoundManager.SOUND\\_CARDS\\_MELEE](#), [SoundManager.SOUND\\_CARDS](#)↵  
[\\_RANGE](#), [SoundManager.SOUND\\_CARDS\\_SIEGE](#), [SoundManager.SOUND\\_COIN](#), [SoundManager.SOUND](#)↵  
[HORN](#), [SoundManager.SOUND\\_RESET](#), [SoundManager.SOUND\\_WEATHER\\_FOG](#), [SoundManager.SOUND](#)↵  
[\\_WEATHER\\_FROST](#), [SoundManager.SOUND\\_WEATHER\\_GOOD](#), [SoundManager.SOUND\\_WEATHER\\_RAIN](#),  
[SoundManager.soundPool](#), and [SoundManager.sounds](#).

### 4.56.3 Member Function Documentation

#### 4.56.3.1 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.

**4.56.3.2 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.

**4.56.3.3 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.

**4.56.3.4 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.

**4.56.3.5 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](#).

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

#### 4.56.3.6 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](#).

#### 4.56.3.7 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](#).

#### 4.56.3.8 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 <a href="#">Sound</a> 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.playCard](#)

RemovedSound(), SoundManager.playClearWeatherSound(), SoundManager.playCoinSound(), SoundManager.playHornSound(), SoundManager.playResetSound(), and SoundManager.playWeatherSound().

#### 4.56.3.9 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](#).

### 4.56.4 Member Data Documentation

#### 4.56.4.1 changeListener

```
final SharedPreferences.OnSharedPreferenceChangeListener changeListener [private]
```

OnSharedPreferenceChangeListener that is registered for the default SharedPreferences and updates the SoundManager::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\(\)](#).

#### 4.56.4.2 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\(\)](#).

#### 4.56.4.3 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().

#### 4.56.4.4 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().

#### 4.56.4.5 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().

#### 4.56.4.6 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().

#### 4.56.4.7 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().

#### 4.56.4.8 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\(\)](#).

#### 4.56.4.9 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\(\)](#).

#### 4.56.4.10 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\(\)](#).

#### 4.56.4.11 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\(\)](#).

#### 4.56.4.12 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().

#### 4.56.4.13 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().

#### 4.56.4.14 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().

#### 4.56.4.15 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().

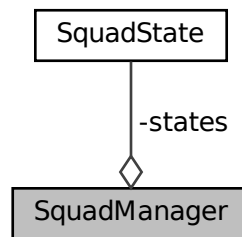
The documentation for this class was generated from the following file:

- gwent/ui/sounds/SoundManager.java

## 4.57 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.

### 4.57.1 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`.

### 4.57.2 Constructor & Destructor Documentation

#### 4.57.2.1 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 Ability.BINDING, SquadState.getState(), SquadManager.MAX\_NR\_SQUADS, and SquadManager.states.

### 4.57.3 Member Function Documentation

#### 4.57.3.1 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.

#### 4.57.3.2 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::memberBaseDamage](#) of said squad.

## Parameters

<i>newVal</i>	Integer representing the newly selected squad.
<i>picker</i>	<a href="#">DamageValuePicker</a> that is updated.

## Exceptions

<i>IllegalArgumentException</i>	When newVal is not between 1 and <a href="#">MAX_NR_SQUADS</a> .
---------------------------------	--

## 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.

### 4.57.4 Member Data Documentation

#### 4.57.4.1 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()`.

#### 4.57.4.2 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.onSquadChanged()`.

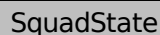
The documentation for this class was generated from the following file:

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

## 4.58 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`:



```

classDiagram
    class 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.*

### 4.58.1 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.

### 4.58.2 Constructor & Destructor Documentation

#### 4.58.2.1 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](#).

Referenced by [SquadState.getState\(\)](#).

### 4.58.3 Member Function Documentation

#### 4.58.3.1 `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()`.

#### 4.58.3.2 `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()`.

#### 4.58.3.3 `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()`.

#### 4.58.3.4 `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 <code>UnitEntity</code> 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 <code>#MAX_NR_SQUADS</code> .
---------------------------------	--

Definition at line 80 of file `SquadState.java`.

References `SquadState.SquadState()`, `UnitEntity.getDamage()`, and `SquadState.squadNumber`.

Referenced by `SquadManager.SquadManager()`.

### 4.58.3.5 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()`.

## 4.58.4 Member Data Documentation

### 4.58.4.1 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()`.

### 4.58.4.2 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()`.

#### 4.58.4.3 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](#)

## 4.59 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:



```

classDiagram
    class StringValuePicker {
        <T extends Comparable>
    }
  
```

### 4.59.1 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](#)

## 4.60 ResetDialogUseCase.Trigger Enum Reference

An Enum listing the possible triggers of a reset.

Collaboration diagram for ResetDialogUseCase.Trigger:



```
graph TD; subgraph "ResetDialogUseCase.Trigger"; end
```

## 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.*

### 4.60.1 Detailed Description

An Enum listing the possible triggers of a reset.  
Definition at line 128 of file ResetDialogUseCase.java.

### 4.60.2 Member Data Documentation

#### 4.60.2.1 BUTTON\_CLICK

`BUTTON_CLICK`  
Represents, that a reset was triggered by a click on the reset button.  
Definition at line 132 of file ResetDialogUseCase.java.  
Referenced by GameBoardViewModel.onResetButtonPressed().

#### 4.60.2.2 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_reset` is true.  
Definition at line 139 of file ResetDialogUseCase.java.  
Referenced by GameBoardViewModel.onFactionSwitchReset(), and ResetAlertDialogBuilderAdapter.setTrigger().  
The documentation for this enum was generated from the following file:

- gwent/domain/cases/ResetDialogUseCase.java

## 4.61 UnitDao Interface Reference

An interface defining data access operations related to `units` table.

Collaboration diagram for UserDao:



## Public Member Functions

- Completable [insertUnit](#) (@NonNull [UnitEntity](#) unit)  
*Inserts the given [UnitEntity](#) into `units`.*
- Completable [insertUnit](#) (boolean epic, @IntRange(from=0) int damage, @NonNull [Ability](#) ability, @IntRange(from=0) @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.*

### 4.61.1 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 UserDao.java.

### 4.61.2 Member Function Documentation

#### 4.61.2.1 [countUnits\(\)](#) [1/2]

Single<Integer> [countUnits](#) ()  
Counts [UnitEntity](#) objects in `units`.

##### Returns

A Single tracking operation status and returning the value.

#### 4.61.2.2 countUnits() [2/2]

```
Single<Integer> countUnits (
    @NonNull RowType row )
```

Counts [UnitEntity](#) objects in `units` in the given row.

##### Parameters

<i>row</i>	<a href="#">RowType</a> defining the <a href="#">UnitEntity::row</a> foreign key.
------------	---

##### Returns

A Single tracking operation status and returning the value.

Referenced by `UnitRepository.countUnits()`.

#### 4.61.2.3 deleteUnit()

```
Completable deleteUnit (
    int id )
```

Deletes the [UnitEntity](#) object with the given `id` from `units`.

##### Parameters

<i>id</i>	Integer representing the <a href="#">UnitEntity::id</a> of the deleted <a href="#">UnitEntity</a> .
-----------	---

##### Returns

A Completable tracking operation status.

Referenced by `UnitRepository.delete()`.

#### 4.61.2.4 deleteUnits()

```
Completable deleteUnits (
    @NonNull Collection< UnitEntity > units )
```

Deletes the given [UnitEntity](#) objects from `units`.

##### Parameters

<i>units</i>	List of <a href="#">UnitEntity</a> that are being inserted.
--------------	---

##### Returns

A Completable tracking operation status.

Referenced by `UnitRepository.delete()`.

#### 4.61.2.5 getUnit()

```
Single<UnitEntity> getUnit (
    int id )
```

Fetches the [UnitEntity](#) object with the given `id` from `units`.

##### Parameters

<i>id</i>	Integer representing the <a href="#">UnitEntity::id</a> of the requested <a href="#">UnitEntity</a> .
-----------	---

**Returns**

A Single tracking operation status and returning the value.

Referenced by `UnitRepository.getUnit()`.

**4.61.2.6 getUnits() [1/2]**

```
Single<List<UnitEntity> > getUnits ( )
```

Fetches all `UnitEntity` objects from `units`.

**Returns**

A Single tracking operation status and returning the value.

**4.61.2.7 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**

<code>row</code>	<code>RowType</code> defining the <code>UnitEntity::row</code> foreign key.
------------------	---

**Returns**

A Single tracking operation status and returning the value.

Referenced by `UnitRepository.getUnits()`.

**4.61.2.8 getUnitsFlowable()**

```
Flowable<List<UnitEntity> > getUnitsFlowable (
    @NonNull RowType row )
```

Fetches a Flowable for all `UnitEntity` objects from `units` in the given row.

**Parameters**

<code>row</code>	<code>RowType</code> defining the <code>UnitEntity::row</code> foreign key.
------------------	---

**Returns**

A Flowable emitting the values.

**See also**

`#getUnits(RowType)`

Referenced by `UnitRepository.getUnitsFlowable()`.



**4.61.2.9 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()`.

**4.61.2.10 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>	<code>UnitEntity</code> that is being inserted.
-------------	---

**Returns**

A Completable tracking operation status.

Referenced by `UnitRepository.insertUnit()`, and `UnitRepository.reset()`.

**4.61.2.11 insertUnit() [2/2]**

```
Completable insertUnit (
    boolean epic,
    @IntRange(from=0) int damage,
    @NonNull Ability ability,
    @IntRange(from=0) @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>	<code>Ability</code> defining the ability of the unit.
<i>squad</i>	Non-Negative value defining the squad of units, if <code>UnitEntity::ability</code> is <code>Ability::BINDING</code> , or <code>null</code> .
<i>row</i>	<code>RowType</code> defining the <code>UnitEntity::row</code> foreign key.

**Returns**

A Completable tracking operation status.

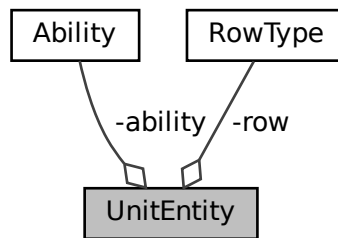
The documentation for this interface was generated from the following file:

- `gwent/data/UnitDao.java`

**4.62 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.*

## 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.

### 4.62.1 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`.

### 4.62.2 Constructor & Destructor Documentation

#### 4.62.2.1 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

<a href="#">epic</a>	Boolean representing whether card is <a href="#">epic</a> .
<a href="#">damage</a>	Non-negative value representing the <a href="#">damage</a> of the card.
<a href="#">ability</a>	<a href="#">Ability</a> representing the <a href="#">ability</a> of the card.
<a href="#">squad</a>	Integer representing the <a href="#">squad</a> of a card that has the <a href="#">Ability::BINDING</a> <a href="#">ability</a> .
<a href="#">row</a>	<a href="#">RowType</a> representing the combat type of the card.

### Exceptions

<i>IllegalArgumentException</i>	When damage is less than zero or if ability is <a href="#">Ability::BINDING</a> and squad is null or less than one or if ability is not <a href="#">Ability::BINDING</a> and squad is not null.
---------------------------------	---

Definition at line 94 of file UnitEntity.java.

References [Ability.BINDING](#).

## 4.62.3 Member Function Documentation

### 4.62.3.1 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>	<a href="#">DamageCalculator</a> visitor used for damage calculation.
-------------------	---

#### Returns

An Integer representing the units (de-)buffed damage.

#### See also

[getDamage\(\)](#)

Definition at line 121 of file UnitEntity.java.

### 4.62.3.2 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 <a href="#">UnitEntity</a> 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 155 of file UnitEntity.java.  
Referenced by BurnDialogUseCase.burn().

#### 4.62.3.3 getAbility()

`Ability` getAbility ( )

Getter for `ability`.

Returns

`Ability` representing the units' ability.

Definition at line 304 of file UnitEntity.java.

#### 4.62.3.4 getDamage()

`int` getDamage ( )

Getter for `damage`.

Returns

Integer representing the card's base-damage.

See also

`#calculateDamage(DamageCalculator)`

Definition at line 280 of file UnitEntity.java.  
Referenced by SquadState.getState().

#### 4.62.3.5 getId()

`int` getId ( )

Getter for `id`.

Returns

Integer representing the units' id.

Definition at line 241 of file UnitEntity.java.  
Referenced by UnitRepository.getUnitsFlowable(), DamageCalculatorBuildDirector.setHorns(), and DamageCalculatorBuildDirector.setMoralBoosts().

#### 4.62.3.6 getRow()

`RowType` getRow ( )

Getter for `row`.

Returns

`RowType` representing the units combat row.

Definition at line 351 of file UnitEntity.java.

#### 4.62.3.7 getSquad()

`Integer getSquad ( )`  
 Getter for [squad](#).

##### Returns

Integer representing the units' squad if [ability](#) is [Ability::BINDING](#) or `null`.

Definition at line 324 of file `UnitEntity.java`.

#### 4.62.3.8 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 136 of file `UnitEntity.java`.

#### 4.62.3.9 isEpic()

`boolean isEpic ( )`  
 Getter for [epic](#).

##### Returns

Boolean representing whether the card is epic.

Definition at line 260 of file `UnitEntity.java`.

#### 4.62.3.10 setAbility()

`void setAbility (`  
     `@NonNull Ability ability ) [package]`

Setter for [ability](#).

Only used by Room extension.

##### Parameters

<i>ability</i>	<a href="#">Ability</a> representing the units' ability.
----------------	--

Definition at line 314 of file `UnitEntity.java`.

**4.62.3.11 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 291 of file UnitEntity.java.

**4.62.3.12 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 270 of file UnitEntity.java.

**4.62.3.13 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 251 of file UnitEntity.java.

**4.62.3.14 setRow()**

```
void setRow (
    @NonNull RowType row ) [package]
```

Setter for [row](#).

Only used by Room extension.

**Parameters**

<i>row</i>	<a href="#">RowType</a> representing the units combat row.
------------	--

Definition at line 361 of file UnitEntity.java.

#### 4.62.3.15 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 <a href="#">ability</a> is <a href="#">Ability::BINDING</a> or null.
--------------	---

##### Exceptions

<i>IllegalArgumentException</i>	When <a href="#">ability</a> is <a href="#">Ability::BINDING</a> and squad is null or less than 1 or if <a href="#">ability</a> is not <a href="#">Ability::BINDING</a> and squad is not null.
---------------------------------	--

Definition at line 335 of file UnitEntity.java.

References [Ability.BINDING](#).

#### 4.62.3.16 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 199 of file UnitEntity.java.

### 4.62.4 Member Data Documentation

#### 4.62.4.1 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 64 of file UnitEntity.java.

#### 4.62.4.2 damage

```
int damage [private]
```

Defines the base-damage of the represented card.

Must be non-negative.



Definition at line 55 of file UnitEntity.java.

#### 4.62.4.3 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.

#### 4.62.4.4 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.

#### 4.62.4.5 row

`RowType row [private]`

Defines the attack row the card lies in.

Must not be `null`.

Definition at line 81 of file UnitEntity.java.

#### 4.62.4.6 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 73 of file UnitEntity.java.

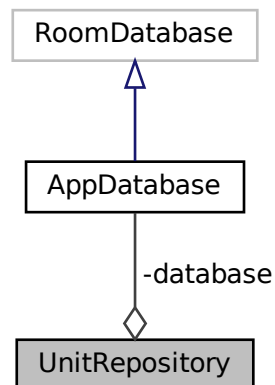
The documentation for this class was generated from the following file:

- gwent/data/UnitEntity.java

## 4.63 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) int damage, @NonNull [Ability](#) ability, @IntRange(from=0) @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) int damage, @NonNull Ability ability, @IntRange(from=0) @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.*

### 4.63.1 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 24 of file [UnitRepository.java](#).

### 4.63.2 Constructor & Destructor Documentation

#### 4.63.2.1 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>	<a href="#">AppDatabase</a> that is injected for the repository.
-----------------	--

**See also**

`#getRepository(AppDatabase)`

Definition at line 40 of file `UnitRepository.java`.

References `UnitRepository.database`.

Referenced by `UnitRepository.getRepository()`.

**4.63.3 Member Function Documentation****4.63.3.1 clearWeather()**

`Completable clearWeather ( )`

Sets `RowEntity::weather` to `false` for all attack rows.

**Returns**

A `Completable` tracking operation status.

Definition at line 204 of file `UnitRepository.java`.

References `RowDao.clearWeather()`, `UnitRepository.database`, and `AppDatabase.rows()`.

Referenced by `GameBoardViewModel.onWeatherButtonPressed()`.

**4.63.3.2 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 273 of file `UnitRepository.java`.

References `UnitRepository.getUnit()`, and `UnitRepository.insertUnit()`.

**4.63.3.3 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 298 of file `UnitRepository.java`.

References `UnitDao.countUnits()`, `UnitRepository.database`, and `AppDatabase.units()`.

#### 4.63.3.4 countUnits() [2/2]

```
Single<Integer> countUnits (
    @NonNull RowType row )
```

Counts the units in the given attack row.

##### Parameters

<i>row</i>	<a href="#">RowEntity::id</a> where the units are counted.
------------	--

##### Returns

A Single tracking operation status and returning the value.

##### See also

[countUnits\(\)](#)

Definition at line 287 of file UnitRepository.java.

References [UnitDao.countUnits\(\)](#), [UnitRepository.database](#), and [AppDatabase.units\(\)](#).

#### 4.63.3.5 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 252 of file UnitRepository.java.

References [UnitRepository.database](#), [UnitDao.deleteUnits\(\)](#), and [AppDatabase.units\(\)](#).

#### 4.63.3.6 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 263 of file `UnitRepository.java`.

References `UnitRepository.database`, `UnitDao.deleteUnit()`, and `AppDatabase.units()`.

**4.63.3.7 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>	<code>AppDatabase</code> managed and initialized by the returned <code>UnitRepository</code> .
-----------------	--

**Returns**

A `Single` emitting the created `UnitRepository` when initialization is finished.

Definition at line 53 of file `UnitRepository.java`.

References `UnitRepository.UnitRepository()`, `UnitRepository.database`, and `UnitRepository.initializeRows()`.

Referenced by `GwentApplication.getRepository()`.

**4.63.3.8 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 309 of file `UnitRepository.java`.

References `UnitRepository.database`, `UnitDao.getUnit()`, and `AppDatabase.units()`.

Referenced by `UnitRepository.copy()`.

**4.63.3.9 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 357 of file UnitRepository.java.

References [UnitRepository.database](#), [UnitDao.getUnits\(\)](#), and [AppDatabase.units\(\)](#).

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

#### 4.63.3.10 [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>	<a href="#">RowEntity::id</a> where the units have been placed.
------------	---

Returns

A Single tracking operation status and returning the value.

See also

[getUnits\(\)](#)

Definition at line 321 of file UnitRepository.java.

References [UnitRepository.database](#), [UnitDao.getUnits\(\)](#), and [AppDatabase.units\(\)](#).

#### 4.63.3.11 [getUnitsFlowable\(\)](#)

```
Flowable<List<UnitEntity> > getUnitsFlowable (
    @NonNull RowType row )
```

Returns a Flowable emitting the units in the given attack row.

Parameters

<i>row</i>	<a href="#">RowEntity::id</a> where the units have been placed.
------------	---

Returns

A Flowable emitting the values.

See also

[#getUnits\(RowType\)](#)

[getUnits\(\)](#)

Definition at line 334 of file UnitRepository.java.

References [UnitRepository.database](#), [UnitEntity.getId\(\)](#), [UnitDao.getUnitsFlowable\(\)](#), and [AppDatabase.units\(\)](#).

#### 4.63.3.12 [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 366 of file UnitRepository.java.

References UnitRepository.database, UnitDao.hasNonEpicUnitsFlowable(), and AppDatabase.units().

#### 4.63.3.13 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 68 of file UnitRepository.java.

References UnitRepository.database, RowDao.insertRow(), and AppDatabase.rows().

Referenced by UnitRepository.getRepository(), and UnitRepository.reset().

#### 4.63.3.14 insertUnit() [1/2]

```
Completable insertUnit (
    boolean epic,
    @IntRange(from=0) int damage,
    @NonNull Ability ability,
    @IntRange(from=0) @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>	<a href="#">Ability</a> representing the #ability of the card.
<i>squad</i>	Integer representing the #squad of a card that has the <a href="#">Ability::BINDING</a> #ability.
<i>row</i>	<a href="#">RowType</a> 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 <a href="#">Ability::BINDING</a> and squad is null or less than zero or if ability is not <a href="#">Ability::BINDING</a> and squad is not null.
---------------------------------	--

Definition at line 118 of file UnitRepository.java.

References Ability.BINDING, UnitRepository.database, UnitDao.insertUnit(), and AppDatabase.units().

Referenced by UnitRepository.copy(), and UnitRepository.insertUnit().



**4.63.3.15 insertUnit()** [2/2]

```
Completable insertUnit (
    boolean epic,
    @IntRange(from=0) int damage,
    @NonNull Ability ability,
    @IntRange(from=0) @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 149 of file UnitRepository.java.

References UnitRepository.insertUnit().

**4.63.3.16 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 225 of file UnitRepository.java.

References UnitRepository.database, RowDao.isHorn(), and AppDatabase.rows().

**4.63.3.17 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>	<a href="#">RowEntity::id</a> where the horn status is queried.
------------	---

**Returns**

A Flowable emitting the values.

**See also**

[#isHorn\(RowType\)](#)

Definition at line 237 of file UnitRepository.java.

References UnitRepository.database, RowDao.isHornFlowable(), and AppDatabase.rows().

**4.63.3.18 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>	<a href="#">RowEntity::id</a> where the weather is queried.
------------	---

**Returns**

A Single tracking operation status and returning the value.

Definition at line 177 of file UnitRepository.java.

References UnitRepository.database, RowDao.isWeather(), and AppDatabase.rows().

**4.63.3.19 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>	<a href="#">RowEntity::id</a> where the weather is queried.
------------	---

## Returns

A Flowable emitting the values.

## See also

[#isWeather\(RowType\)](#)

Definition at line 189 of file UnitRepository.java.

References [UnitRepository.database](#), [RowDao.isWeatherFlowable\(\)](#), and [AppDatabase.rows\(\)](#).

**4.63.3.20 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 85 of file UnitRepository.java.

**4.63.3.21 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>	<a href="#">UnitEntity</a> that should be kept.
-----------------	---

## Returns

A Completable tracking operation status.

## See also

[initializeRows\(\)](#)

Definition at line 98 of file UnitRepository.java.

References [RowDao.clearRows\(\)](#), [UnitRepository.database](#), [UnitRepository.initializeRows\(\)](#), [UnitDao.insertUnit\(\)](#), [AppDatabase.rows\(\)](#), and [AppDatabase.units\(\)](#).

**4.63.3.22 switchHorn()**

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

Flips [RowEntity::horn](#) of the given attack row.

#### Parameters

<i>row</i>	<a href="#">RowEntity::id</a> where the horn status should be updated.
------------	--

#### Returns

A Completable tracking operation status.

Definition at line 214 of file `UnitRepository.java`.

References `UnitRepository.database`, `AppDatabase.rows()`, and `RowDao.updateHorn()`.

#### 4.63.3.23 switchWeather()

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

Flips [RowEntity::weather](#) of the given attack row.

#### Parameters

<i>row</i>	<a href="#">RowEntity::id</a> where the weather should be updated.
------------	--

#### Returns

A Completable tracking operation status.

Definition at line 166 of file `UnitRepository.java`.

References `UnitRepository.database`, `AppDatabase.rows()`, and `RowDao.updateWeather()`.

### 4.63.4 Member Data Documentation

#### 4.63.4.1 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 30 of file `UnitRepository.java`.

Referenced by `UnitRepository.UnitRepository()`, `UnitRepository.clearWeather()`, `UnitRepository.countUnits()`, `UnitRepository.delete()`, `UnitRepository.getRepository()`, `UnitRepository.getUnit()`, `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()`.

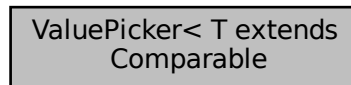
The documentation for this class was generated from the following file:

- `gwent/data/UnitRepository.java`

## 4.64 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`:



#### 4.64.1 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`.

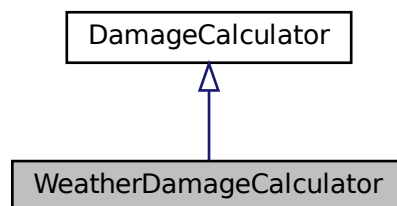
The documentation for this class was generated from the following file:

- `gwent/ui/dialogs/addcard/ValuePicker.java`

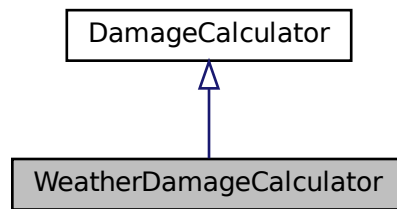
## 4.65 WeatherDamageCalculator Class Reference

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()`.*

### 4.65.1 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`.

### 4.65.2 Constructor & Destructor Documentation

#### 4.65.2.1 WeatherDamageCalculator()

```
WeatherDamageCalculator (
    boolean weather ) [package]
```

Constructor of a `WeatherDamageCalculator`.

Should only be called by `DamageCalculatorBuilder`.

#### Parameters

<code>weather</code>	Boolean defining whether the weather debuff should be applied.
----------------------	--

See also

[DamageCalculatorBuilder](#)

Definition at line 27 of file WeatherDamageCalculator.java.

References WeatherDamageCalculator.weather.

### 4.65.3 Member Function Documentation

#### 4.65.3.1 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 <a href="#">UnitEntity::id</a> 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.

#### 4.65.3.2 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 <a href="#">UnitEntity::id</a> 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.

## 4.65.4 Member Data Documentation

### 4.65.4.1 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



# Index

- Ability, 9
  - BINDING, 9
  - HORN, 10
  - MORAL\_BOOST, 10
  - NONE, 10
  - REVENGE, 10
- ability
  - UnitEntity, 188
- abilityImageId
  - CardUiState, 34
- abilityPicker
  - CardNumberPickerAdapter, 29
- abilityView
  - CardListAdapter.CardViewHolder, 43
- accept
  - MenuUiStateObserver, 102
  - RowUiStateObserver, 138
- activated
  - Sound, 161
- adapteeBuilder
  - ResetAlertDialogBuilderAdapter, 116
  - RevengeAlertDialogBuilderAdapter, 125
- AddCardDialog, 10
  - AddCardDialog, 12
  - disposables, 13
  - onCreate, 12
  - pickerAdapter, 13
  - row, 13
- addSelectedUnits
  - CardNumberPickerAdapter, 28
- animatedSwitch
  - ImageViewSwitchAnimator, 85
- AppDatabase, 14
  - rows, 14
  - units, 15
- ARG\_SECTION\_NUMBER
  - IntroductionFragment, 93
- AVENGER\_ABILITY
  - RevengeAlertDialogBuilderAdapter, 125
- AVENGER\_DAMAGE
  - RevengeAlertDialogBuilderAdapter, 125
- AVENGER\_EPIC
  - RevengeAlertDialogBuilderAdapter, 125
- AVENGER\_ROW
  - RevengeAlertDialogBuilderAdapter, 126
- AVENGER\_SQUAD
  - RevengeAlertDialogBuilderAdapter, 126
- BACKGROUND
  - OverlayDialog, 111
- ballViews
  - FactionSwitchListener, 69
- BINDING
  - Ability, 9
- bindingView
  - CardListAdapter.CardViewHolder, 43
- BondDamageCalculatorDecorator, 15
  - BondDamageCalculatorDecorator, 16
  - calculateDamage, 17
  - idToSquadSize, 18
  - isBuffed, 17
- BUFFED
  - DamageCalculator.Color, 49
- burn
  - BurnDialogUseCase, 19
  - MenuUiState, 100
- burnButton
  - MenuUiStateObserver, 103
- BurnDialogUseCase, 18
  - burn, 19
  - getBurnUnits, 20
- BUTTON\_CLICK
  - ResetDialogUseCase.Trigger, 177
- calculateDamage
  - BondDamageCalculatorDecorator, 17
  - DamageCalculator, 51
  - HornDamageCalculatorDecorator, 83
  - MoralDamageCalculatorDecorator, 105
  - UnitEntity, 184
  - WeatherDamageCalculator, 203
- calculator
  - DamageCalculatorBuilder, 57
- callback
  - ChangeFactionDialog, 46
- CARD\_ABILITIES
  - RuleSection, 143
- CardListAdapter, 22
  - CardListAdapter, 24
  - getItemId, 24
  - onBindViewHolder, 25
  - onCopy, 25
  - onCreateViewHolder, 25
  - onRemove, 26
- cardListAdapter
  - ShowUnitsDialog, 157
- CardListAdapter.CardViewHolder, 39
  - abilityView, 43
  - bindingView, 43
  - CardViewHolder, 41

- damageView, 43
- getAbilityView, 41
- getBindingView, 42
- getDamageView, 42
- itemId, 43
- setItemId, 42
- CardNumberPickerAdapter, 26
  - abilityPicker, 29
  - addSelectedUnits, 28
  - CardNumberPickerAdapter, 27
  - damagePicker, 29
  - EPIC\_UNIT\_ABILITIES, 29
  - epicPicker, 29
  - getDelayedOnValueChangeListener, 28
  - numberPicker, 29
  - squadPicker, 29
- CARDS
  - RuleSection, 143
- CardUiState, 30
  - abilityImageId, 34
  - CardUiState, 31
  - damageBackgroundImageId, 34
  - damageString, 35
  - damageTextColor, 35
  - DIFF\_CALLBACK, 35
  - equals, 32
  - getAbilityImageId, 32
  - getDamageBackgroundImageId, 32
  - getDamageString, 33
  - getDamageTextColor, 33
  - getSquadString, 33
  - getUnitId, 33
  - showAbility, 34
  - showSquad, 34
  - squadString, 35
  - unitId, 36
  - UNUSED, 36
- CardUiStateFactory, 36
  - CardUiStateFactory, 37
  - createCardUiState, 38
  - damageTextColors, 39
  - horn, 39
  - weather, 39
- CardViewHolder
  - CardListAdapter.CardViewHolder, 41
- cardViews
  - FactionSwitchListener, 69
- ChangeFactionDialog, 44
  - callback, 46
  - ChangeFactionDialog, 45
  - getOnThemeClickListener, 45
  - onCreate, 46
- ChangeFactionDialog.Callback, 21
  - onThemeSelect, 22
- changeListener
  - SoundManager, 166
- clearRows
  - RowDao, 127
- clearWeather
  - RowDao, 127
  - UnitRepository, 192
- CoinFlipDialog, 46
  - CoinFlipDialog, 48
  - onCreate, 48
  - RANDOM, 48
- collectionToString
  - UnitEntity, 184
- COMMANDER
  - RuleSection, 143
- component
  - DamageCalculatorDecorator, 58
- copy
  - UnitRepository, 192
- countUnits
  - UnitDao, 178
  - UnitRepository, 192
- COURSE
  - RuleSection, 143
- create
  - ResetAlertDialogBuilderAdapter, 115
  - RevengeAlertDialogBuilderAdapter, 123
- createCardUiState
  - CardUiStateFactory, 38
- createFragment
  - SectionsPagerAdapter, 145
- createSound
  - Sound, 159
- CURRENT\_FRAGMENT\_KEY
  - SettingsActivity, 149
- currentView
  - IndicatorManager, 89
- damage
  - MenuUiState, 100
  - RowUiState, 136
  - UnitEntity, 188
- damageBackgroundImageId
  - CardUiState, 34
- DamageCalculator, 50
  - calculateDamage, 51
  - isBuffed, 51
- DamageCalculator.Color, 49
  - BUFFED, 49
  - DEBUFFED, 49
  - DEFAULT, 49
- DamageCalculatorBuildDirector, 52
  - getCalculator, 53
  - setHorns, 53
  - setMoralBoosts, 53
  - setSquads, 54
- DamageCalculatorBuilder, 54
  - calculator, 57
  - getResult, 55
  - setBond, 56
  - setHorn, 56
  - setMoral, 56
  - setWeather, 57

- DamageCalculatorDecorator, 57
  - component, 58
  - DamageCalculatorDecorator, 58
- DamageCalculatorUseCase, 59
  - getDamageCalculator, 59, 60
- damagePicker
  - CardNumberPickerAdapter, 29
- damageString
  - CardUiState, 35
- damageTextColor
  - CardUiState, 35
- damageTextColors
  - CardUiStateFactory, 39
- DamageValuePicker, 60
  - DamageValuePicker, 62
  - EPIC\_DAMAGE\_VALUES, 64
  - epicValues, 64
  - getDisplayString, 62
  - getValue, 63
  - NON\_EPIC\_DAMAGE\_VALUES\_UPPER\_BOUND, 64
  - setEpicValues, 63
  - setSelectableValues, 63
  - setValue, 64
- damageView
  - CardListAdapter.CardViewHolder, 43
  - MenuUiStateObserver, 103
  - RowUiStateObserver, 139
- database
  - GwentApplication, 80
  - UnitRepository, 200
- DEBUFFED
  - DamageCalculator.Color, 49
- DEFAULT
  - DamageCalculator.Color, 49
  - ResetDialogUseCase.DialogType, 65
- delete
  - UnitRepository, 193
- deleteUnit
  - UnitDao, 179
- deleteUnits
  - UnitDao, 179
- dialogCallback
  - ResetAlertDialogBuilderAdapter, 116
- DIFF\_CALLBACK
  - CardUiState, 35
- dismissViewId
  - OverlayDialog, 111
- disposables
  - AddCardDialog, 13
  - MainActivity, 97
  - ShowUnitsDialog, 157
- doubleDamage
  - HornDamageCalculatorDecorator, 84
- epic
  - UnitEntity, 189
- EPIC\_DAMAGE\_VALUES
  - DamageValuePicker, 64
- EPIC\_UNIT\_ABILITIES
  - CardNumberPickerAdapter, 29
- epicPicker
  - CardNumberPickerAdapter, 29
- epicValues
  - DamageValuePicker, 64
- equals
  - CardUiState, 32
- FACTION\_SWITCH
  - ResetDialogUseCase.Trigger, 177
- factionButton
  - FactionSwitchListener, 70
- FACTIONS
  - RuleSection, 144
- FactionSwitchListener, 66
  - ballViews, 69
  - cardViews, 69
  - factionButton, 70
  - FactionSwitchListener, 67
  - addListener, 68
  - onSharedPreferenceChanged, 68
  - setTheme, 69
  - THEME\_MONSTER, 70
  - THEME\_NILFGAARD, 70
  - THEME\_NORTHERN\_KINGDOMS, 70
  - THEME\_PREFERENCE\_KEY, 70
  - THEME\_SCOIATAEL, 71
  - unitViews, 71
- factionSwitchListener
  - MainActivity, 97
- GameBoardViewModel, 71
  - GameBoardViewModel, 73
  - getMenuUiState, 73
  - getModel, 73
  - getRepository, 74
  - getRowUiState, 74
  - INITIALIZER, 77
  - menuUiState, 78
  - onBurnButtonPressed, 74
  - onFactionSwitchReset, 75
  - onHornViewPressed, 75
  - onResetButtonPressed, 76
  - onWeatherButtonPressed, 76
  - onWeatherViewPressed, 76
  - reset, 77
  - rowUiStates, 78
- gameBoardViewModel
  - MainActivity, 97
- GENERAL
  - RuleSection, 144
- getAbility
  - UnitEntity, 185
- getAbilityImageId
  - CardUiState, 32
- getAbilityView
  - CardListAdapter.CardViewHolder, 41
- getBindingView

- CardListAdapter.CardViewHolder, 42
- getBurnUnits
  - BurnDialogUseCase, 20
- getCalculator
  - DamageCalculatorBuildDirector, 53
- getDamage
  - MenuUiState, 99
  - RowUiState, 135
  - UnitEntity, 185
- getDamageBackgroundImageId
  - CardUiState, 32
- getDamageCalculator
  - DamageCalculatorUseCase, 59, 60
- getDamageString
  - CardUiState, 33
- getDamageTextColor
  - CardUiState, 33
- getDamageView
  - CardListAdapter.CardViewHolder, 42
- getDelayedOnValueChangeListener
  - CardNumberPickerAdapter, 28
- getDialog
  - ShowUnitsDialog, 155
- getDialogType
  - ResetDialogUseCase, 118
- getDisplayString
  - DamageValuePicker, 62
- getFirstSquadWithMembers
  - SquadManager, 171
- getId
  - RowEntity, 131
  - UnitEntity, 185
- getItemCount
  - SectionsPagerAdapter, 146
- getItemId
  - CardListAdapter, 24
- getListener
  - FactionSwitchListener, 68
- getMemberBaseDamage
  - SquadState, 173
- getMenuUiState
  - GameBoardViewModel, 73
- getModel
  - GameBoardViewModel, 73
- getObserver
  - RowUiStateObserver, 139
- getOnThemeClickListener
  - ChangeFactionDialog, 45
- getOverlayView
  - ImageViewSwitchAnimator, 86
- getPreferenceKey
  - Sound, 159
- getRandomUnit
  - ResetRepositoryUseCase, 120
- getRepository
  - GameBoardViewModel, 74
  - GwentApplication, 80
  - UnitRepository, 194
- getResult
  - DamageCalculatorBuilder, 55
- getRevengeDialog
  - RemoveUnitsUseCase, 112
  - ResetRepositoryUseCase, 120
- getRow
  - ShowUnitsDialog, 156
  - UnitEntity, 185
- getRowUiState
  - GameBoardViewModel, 74
- getSectionClickListener
  - SettingsRuleFragment, 151
- getSoundId
  - Sound, 160
- getSquad
  - UnitEntity, 185
- getSquadMembers
  - SquadState, 174
- getSquadNumber
  - SquadState, 174
- getSquadString
  - CardUiState, 33
- getState
  - SquadState, 174
- getUnit
  - UnitDao, 179
  - UnitRepository, 194
- getUnitId
  - CardUiState, 33
- getUnits
  - RowUiState, 135
  - UnitDao, 180
  - UnitRepository, 194, 195
- getUnitsFlowable
  - UnitDao, 180
  - UnitRepository, 195
- getValue
  - DamageValuePicker, 63
- GwentApplication, 78
  - database, 80
  - getRepository, 80
  - onCreate, 80
  - repository, 81
- hasMembers
  - SquadState, 175
- hasNonEpicUnitsFlowable
  - UnitDao, 180
  - UnitRepository, 195
- HORN
  - Ability, 10
- horn
  - CardUiStateFactory, 39
  - RowEntity, 132
  - RowUiState, 136
- HornDamageCalculatorDecorator, 81
  - calculateDamage, 83
  - doubleDamage, 84
  - HornDamageCalculatorDecorator, 83

- isBuffed, [84](#)
  - unitIds, [84](#)
- hornView
  - RowUiStateObserver, [140](#)
- id
  - RowEntity, [132](#)
  - UnitEntity, [189](#)
- idToSquadSize
  - BondDamageCalculatorDecorator, [18](#)
- imageIds
  - IntroductionFragment, [93](#)
- ImageViewSwitchAnimator, [85](#)
  - animatedSwitch, [85](#)
  - getOverlayView, [86](#)
- IndicatorManager, [86](#)
  - currentView, [89](#)
  - IndicatorManager, [87](#)
  - indicators, [89](#)
  - updateIndicators, [87](#)
- indicatorManager
  - IntroductionActivity, [91](#)
- indicators
  - IndicatorManager, [89](#)
- inflateCoinFlipPopup
  - MainActivity, [95](#)
- inflateFactionPopup
  - MainActivity, [95](#)
- INITIALIZER
  - GameBoardViewModel, [77](#)
- initializeRows
  - UnitRepository, [196](#)
- initializeViewModel
  - MainActivity, [96](#)
- insertAvengers
  - RevengeAlertDialogBuilderAdapter, [124](#)
- insertRow
  - RowDao, [127](#)
- insertUnit
  - UnitDao, [181](#)
  - UnitRepository, [196](#)
- INTENT\_EXTRA\_TAG
  - RuleActivity, [142](#)
- IntroductionActivity, [89](#)
  - indicatorManager, [91](#)
  - onCreate, [90](#)
- IntroductionFragment, [91](#)
  - ARG\_SECTION\_NUMBER, [93](#)
  - imageIds, [93](#)
  - newInstance, [92](#)
  - onCreateView, [93](#)
  - PAGES\_COUNT, [94](#)
  - stringIds, [94](#)
- isActivated
  - Sound, [160](#)
- isBuffed
  - BondDamageCalculatorDecorator, [17](#)
  - DamageCalculator, [51](#)
  - HornDamageCalculatorDecorator, [84](#)
  - MoralDamageCalculatorDecorator, [107](#)
  - UnitEntity, [186](#)
  - WeatherDamageCalculator, [203](#)
- isBurn
  - MenuUiState, [99](#)
- isEpic
  - UnitEntity, [186](#)
- isHorn
  - RowDao, [128](#)
  - RowEntity, [131](#)
  - RowUiState, [135](#)
  - UnitRepository, [197](#)
- isHornFlowable
  - RowDao, [128](#)
  - UnitRepository, [198](#)
- isReset
  - MenuUiState, [99](#)
- isWeather
  - MenuUiState, [100](#)
  - RowDao, [128](#)
  - RowEntity, [131](#)
  - RowUiState, [136](#)
  - UnitRepository, [198](#)
- isWeatherFlowable
  - RowDao, [129](#)
  - UnitRepository, [198](#)
- itemId
  - CardListAdapter.CardViewHolder, [43](#)
- layout
  - OverlayDialog, [111](#)
- MainActivity, [94](#)
  - disposables, [97](#)
  - factionSwitchListener, [97](#)
  - gameBoardViewModel, [97](#)
  - inflateCoinFlipPopup, [95](#)
  - inflateFactionPopup, [95](#)
  - initializeViewModel, [96](#)
  - onCreate, [96](#)
  - onDestroy, [96](#)
  - onResume, [96](#)
  - onWindowFocusChanged, [97](#)
  - soundManager, [97](#)
- MAX\_NR\_SQUADS
  - SquadManager, [171](#)
- MELEE
  - RowType, [133](#)
- memberBaseDamage
  - SquadState, [175](#)
- MenuUiState, [98](#)
  - burn, [100](#)
  - damage, [100](#)
  - getDamage, [99](#)
  - isBurn, [99](#)
  - isReset, [99](#)
  - isWeather, [100](#)
  - MenuUiState, [98](#)
  - reset, [100](#)

- weather, [100](#)
- menuUiState
  - GameBoardViewModel, [78](#)
- MenuUiStateObserver, [101](#)
  - accept, [102](#)
  - burnButton, [103](#)
  - damageView, [103](#)
  - MenuUiStateObserver, [102](#)
  - resetButton, [103](#)
  - weatherButton, [103](#)
- MONSTER
  - ResetDialogUseCase.DialogType, [65](#)
- MORAL\_BOOST
  - Ability, [10](#)
- MoralDamageCalculatorDecorator, [103](#)
  - calculateDamage, [105](#)
  - isBuffed, [107](#)
  - MoralDamageCalculatorDecorator, [105](#)
  - unitIds, [107](#)
- newInstance
  - IntroductionFragment, [92](#)
- NO\_DISMISS\_VIEW
  - OverlayDialog, [111](#)
- NON\_EPIC\_DAMAGE\_VALUES\_UPPER\_BOUND
  - DamageValuePicker, [64](#)
- NONE
  - Ability, [10](#)
  - ResetDialogUseCase.DialogType, [65](#)
- numberPicker
  - CardNumberPickerAdapter, [29](#)
- onBindViewHolder
  - CardListAdapter, [25](#)
- onBurnButtonPressed
  - GameBoardViewModel, [74](#)
- onCopy
  - CardListAdapter, [25](#)
- onCreate
  - AddCardDialog, [12](#)
  - ChangeFactionDialog, [46](#)
  - CoinFlipDialog, [48](#)
  - GwentApplication, [80](#)
  - IntroductionActivity, [90](#)
  - MainActivity, [96](#)
  - OverlayDialog, [110](#)
  - RuleActivity, [142](#)
  - SettingsActivity, [147](#)
  - ShowUnitsDialog, [156](#)
- onCreatePreferences
  - SettingsHeaderFragment, [150](#)
  - SettingsRuleFragment, [151](#)
  - SettingsSoundFragment, [153](#)
- onCreateView
  - IntroductionFragment, [93](#)
- onCreateViewHolder
  - CardListAdapter, [25](#)
- onDestroy
  - MainActivity, [96](#)
- onFactionSwitchReset
  - GameBoardViewModel, [75](#)
- onHornViewPressed
  - GameBoardViewModel, [75](#)
- onOptionsItemSelected
  - RuleActivity, [142](#)
  - SettingsActivity, [148](#)
- onPreferenceStartFragment
  - SettingsActivity, [148](#)
- onRemove
  - CardListAdapter, [26](#)
- onResetButtonPressed
  - GameBoardViewModel, [76](#)
- onResume
  - MainActivity, [96](#)
- onSaveInstanceState
  - SettingsActivity, [148](#)
- onSharedPreferenceChanged
  - FactionSwitchListener, [68](#)
- onSquadChanged
  - SquadManager, [171](#)
- onThemeSelect
  - ChangeFactionDialog.Callback, [22](#)
- OnValueChangeListener< T extends Comparable, [108](#)
- onWeatherButtonPressed
  - GameBoardViewModel, [76](#)
- onWeatherViewPressed
  - GameBoardViewModel, [76](#)
- onWindowFocusChanged
  - MainActivity, [97](#)
- OverlayDialog, [108](#)
  - BACKGROUND, [111](#)
  - dismissViewId, [111](#)
  - layout, [111](#)
  - NO\_DISMISS\_VIEW, [111](#)
  - onCreate, [110](#)
  - OverlayDialog, [109, 110](#)
- PAGES\_COUNT
  - IntroductionFragment, [94](#)
- pickerAdapter
  - AddCardDialog, [13](#)
- playBurnSound
  - SoundManager, [163](#)
- playCardAddSound
  - SoundManager, [164](#)
- playCardRemovedSound
  - SoundManager, [164](#)
- playClearWeatherSound
  - SoundManager, [164](#)
- playCoinSound
  - SoundManager, [164](#)
- playHornSound
  - SoundManager, [165](#)
- playResetSound
  - SoundManager, [165](#)
- playSound
  - SoundManager, [165](#)
- playWeatherSound

- SoundManager, 166
- preferenceKey
  - Sound, 161
- RANDOM
  - CoinFlipDialog, 48
- RANGE
  - RowType, 133
- remove
  - RemoveUnitsUseCase, 113
- RemoveUnitsUseCase, 112
  - getRevengeDialog, 112
  - remove, 113
- repository
  - GwentApplication, 81
- reset
  - GameBoardViewModel, 77
  - MenuUiState, 100
  - ResetAlertDialogBuilderAdapter.Callback, 21
  - ResetDialogUseCase, 118, 119
  - ResetRepositoryUseCase, 121
  - UnitRepository, 199
- ResetAlertDialogBuilderAdapter, 114
  - adapteeBuilder, 116
  - create, 115
  - dialogCallback, 116
  - ResetAlertDialogBuilderAdapter, 115
  - setMonsterDialog, 116
  - setTrigger, 116
- ResetAlertDialogBuilderAdapter.Callback, 20
  - reset, 21
- resetButton
  - MenuUiStateObserver, 103
- ResetDialogUseCase, 117
  - getDialogType, 118
  - reset, 118, 119
- ResetDialogUseCase.DialogType, 65
  - DEFAULT, 65
  - MONSTER, 65
  - NONE, 65
- ResetDialogUseCase.Trigger, 176
  - BUTTON\_CLICK, 177
  - FACTION\_SWITCH, 177
- ResetRepositoryUseCase, 119
  - getRandomUnit, 120
  - getRevengeDialog, 120
  - reset, 121
- REVENGE
  - Ability, 10
- RevengeAlertDialogBuilderAdapter, 122
  - adapteeBuilder, 125
  - AVENGER\_ABILITY, 125
  - AVENGER\_DAMAGE, 125
  - AVENGER\_EPIC, 125
  - AVENGER\_ROW, 126
  - AVENGER\_SQUAD, 126
  - create, 123
  - insertAvengers, 124
  - RevengeAlertDialogBuilderAdapter, 123
  - setNegativeCallback, 124
  - setPositiveCallback, 124
- row
  - AddCardDialog, 13
  - ShowUnitsDialog, 157
  - UnitEntity, 189
- RowDao, 126
  - clearRows, 127
  - clearWeather, 127
  - insertRow, 127
  - isHorn, 128
  - isHornFlowable, 128
  - isWeather, 128
  - isWeatherFlowable, 129
  - updateHorn, 129
  - updateWeather, 129
- RowEntity, 130
  - getId, 131
  - horn, 132
  - id, 132
  - isHorn, 131
  - isWeather, 131
  - RowEntity, 131
  - setHorn, 132
  - setWeather, 132
  - weather, 132
- rows
  - AppDatabase, 14
- RowType, 133
  - MELEE, 133
  - RANGE, 133
  - SIEGE, 133
- RowUiState, 134
  - damage, 136
  - getDamage, 135
  - getUnits, 135
  - horn, 136
  - isHorn, 135
  - isWeather, 136
  - RowUiState, 135
  - units, 136
  - weather, 136
- RowUiStateObserver, 137
  - accept, 138
  - damageView, 139
  - getObserver, 139
  - hornView, 140
  - RowUiStateObserver, 138
  - unitView, 140
  - weatherResource, 140
  - weatherView, 140
- rowUiStates
  - GameBoardViewModel, 78
- RuleActivity, 141
  - INTENT\_EXTRA\_TAG, 142
  - onCreate, 142
  - onOptionsItemSelected, 142
- RuleSection, 142



- CARD\_ABILITIES, 143
- CARDS, 143
- COMMANDER, 143
- COURSE, 143
- FACTIONS, 144
- GENERAL, 144
- SPECIAL\_CARDS, 144
- SectionsPagerAdapter, 144
  - createFragment, 145
  - getItemCount, 146
  - SectionsPagerAdapter, 145
- setAbility
  - UnitEntity, 186
- setActivated
  - Sound, 160
- setBond
  - DamageCalculatorBuilder, 56
- setDamage
  - UnitEntity, 186
- setEpic
  - UnitEntity, 187
- setEpicValues
  - DamageValuePicker, 63
- setHorn
  - DamageCalculatorBuilder, 56
  - RowEntity, 132
- setHorns
  - DamageCalculatorBuildDirector, 53
- setId
  - UnitEntity, 187
- setItemId
  - CardListAdapter.CardViewHolder, 42
- setMonsterDialog
  - ResetAlertDialogBuilderAdapter, 116
- setMoral
  - DamageCalculatorBuilder, 56
- setMoralBoosts
  - DamageCalculatorBuildDirector, 53
- setNegativeCallback
  - RevengeAlertDialogBuilderAdapter, 124
- setPositiveCallback
  - RevengeAlertDialogBuilderAdapter, 124
- setRow
  - UnitEntity, 187
- setSelectableValues
  - DamageValuePicker, 63
- setSquad
  - UnitEntity, 188
- setSquads
  - DamageCalculatorBuildDirector, 54
- setTheme
  - FactionSwitchListener, 69
- SettingsActivity, 146
  - CURRENT\_FRAGMENT\_KEY, 149
  - onCreate, 147
  - onOptionsItemSelected, 148
  - onPreferenceStartFragment, 148
  - onSaveInstanceState, 148
- SettingsHeaderFragment, 149
  - onCreatePreferences, 150
- SettingsRuleFragment, 150
  - getSectionClickListener, 151
  - onCreatePreferences, 151
- SettingsSoundFragment, 152
  - onCreatePreferences, 153
- setTrigger
  - ResetAlertDialogBuilderAdapter, 116
- setValue
  - DamageValuePicker, 64
- setWeather
  - DamageCalculatorBuilder, 57
  - RowEntity, 132
- showAbility
  - CardUiState, 34
- showSquad
  - CardUiState, 34
- ShowUnitsDialog, 153
  - cardListAdapter, 157
  - disposables, 157
  - getDialog, 155
  - getRow, 156
  - onCreate, 156
  - row, 157
  - ShowUnitsDialog, 155
  - unitRecyclerView, 157
- SIEGE
  - RowType, 133
- Sound, 158
  - activated, 161
  - createSound, 159
  - getPreferenceKey, 159
  - getSoundId, 160
  - isActive, 160
  - preferenceKey, 161
  - setActivated, 160
  - Sound, 159
  - soundId, 161
- SOUND\_BURN
  - SoundManager, 166
- SOUND\_CARDS\_EPIC
  - SoundManager, 166
- SOUND\_CARDS\_MELEE
  - SoundManager, 167
- SOUND\_CARDS\_RANGE
  - SoundManager, 167
- SOUND\_CARDS\_SIEGE
  - SoundManager, 167
- SOUND\_COIN
  - SoundManager, 167
- SOUND\_HORN
  - SoundManager, 167
- SOUND\_RESET
  - SoundManager, 168
- SOUND\_WEATHER\_FOG
  - SoundManager, 168
- SOUND\_WEATHER\_FROST



- SoundManager, 168
- SOUND\_WEATHER\_GOOD
  - SoundManager, 168
- SOUND\_WEATHER\_RAIN
  - SoundManager, 169
- soundId
  - Sound, 161
- SoundManager, 161
  - changeListener, 166
  - playBurnSound, 163
  - playCardAddSound, 164
  - playCardRemovedSound, 164
  - playClearWeatherSound, 164
  - playCoinSound, 164
  - playHornSound, 165
  - playResetSound, 165
  - playSound, 165
  - playWeatherSound, 166
  - SOUND\_BURN, 166
  - SOUND\_CARDS\_EPIC, 166
  - SOUND\_CARDS\_MELEE, 167
  - SOUND\_CARDS\_RANGE, 167
  - SOUND\_CARDS\_SIEGE, 167
  - SOUND\_COIN, 167
  - SOUND\_HORN, 167
  - SOUND\_RESET, 168
  - SOUND\_WEATHER\_FOG, 168
  - SOUND\_WEATHER\_FROST, 168
  - SOUND\_WEATHER\_GOOD, 168
  - SOUND\_WEATHER\_RAIN, 169
  - SoundManager, 163
  - soundPool, 169
  - sounds, 169
- soundManager
  - MainActivity, 97
- soundPool
  - SoundManager, 169
- sounds
  - SoundManager, 169
- SPECIAL\_CARDS
  - RuleSection, 144
- squad
  - UnitEntity, 189
- SquadManager, 169
  - getFirstSquadWithMembers, 171
  - MAX\_NR\_SQUADS, 171
  - onSquadChanged, 171
  - SquadManager, 170
  - states, 172
- squadMembers
  - SquadState, 175
- squadNumber
  - SquadState, 175
- squadPicker
  - CardNumberPickerAdapter, 29
- SquadState, 172
  - getMemberBaseDamage, 173
  - getSquadMembers, 174
  - getSquadNumber, 174
  - getState, 174
  - hasMembers, 175
  - memberBaseDamage, 175
  - squadMembers, 175
  - squadNumber, 175
  - SquadState, 173
- squadString
  - CardUiState, 35
- states
  - SquadManager, 172
- stringIds
  - IntroductionFragment, 94
- StringValuePicker< T extends Comparable, 176
- switchHorn
  - UnitRepository, 199
- switchWeather
  - UnitRepository, 200
- THEME\_MONSTER
  - FactionSwitchListener, 70
- THEME\_NILFGAARD
  - FactionSwitchListener, 70
- THEME\_NORTHERN\_KINGDOMS
  - FactionSwitchListener, 70
- THEME\_PREFERENCE\_KEY
  - FactionSwitchListener, 70
- THEME\_SCOIATAEL
  - FactionSwitchListener, 71
- toString
  - UnitEntity, 188
- UnitDao, 177
  - countUnits, 178
  - deleteUnit, 179
  - deleteUnits, 179
  - getUnit, 179
  - getUnits, 180
  - getUnitsFlowable, 180
  - hasNonEpicUnitsFlowable, 180
  - insertUnit, 181
- UnitEntity, 181
  - ability, 188
  - calculateDamage, 184
  - collectionToString, 184
  - damage, 188
  - epic, 189
  - getAbility, 185
  - getDamage, 185
  - getId, 185
  - getRow, 185
  - getSquad, 185
  - id, 189
  - isBuffed, 186
  - isEpic, 186
  - row, 189
  - setAbility, 186
  - setDamage, 186
  - setEpic, 187

- setId, 187
- setRow, 187
- setSquad, 188
- squad, 189
- toString, 188
- UnitEntity, 183
- unitId
  - CardUiState, 36
- unitIds
  - HornDamageCalculatorDecorator, 84
  - MoralDamageCalculatorDecorator, 107
- unitRecyclerView
  - ShowUnitsDialog, 157
- UnitRepository, 189
  - clearWeather, 192
  - copy, 192
  - countUnits, 192
  - database, 200
  - delete, 193
  - getRepository, 194
  - getUnit, 194
  - getUnits, 194, 195
  - getUnitsFlowable, 195
  - hasNonEpicUnitsFlowable, 195
  - initializeRows, 196
  - insertUnit, 196
  - isHorn, 197
  - isHornFlowable, 198
  - isWeather, 198
  - isWeatherFlowable, 198
  - reset, 199
  - switchHorn, 199
  - switchWeather, 200
  - UnitRepository, 191
- units
  - AppDatabase, 15
  - RowUiState, 136
- unitView
  - RowUiStateObserver, 140
- unitViews
  - FactionSwitchListener, 71
- UNUSED
  - CardUiState, 36
- updateHorn
  - RowDao, 129
- updateIndicators
  - IndicatorManager, 87
- updateWeather
  - RowDao, 129
- ValuePicker< T extends Comparable, 200
- weather
  - CardUiStateFactory, 39
  - MenuUiState, 100
  - RowEntity, 132
  - RowUiState, 136
  - WeatherDamageCalculator, 204
- weatherButton
  - MenuUiStateObserver, 103
- WeatherDamageCalculator, 201
  - calculateDamage, 203
  - isBuffed, 203
  - weather, 204
- WeatherDamageCalculator, 202
- weatherResource
  - RowUiStateObserver, 140
- weatherView
  - RowUiStateObserver, 140