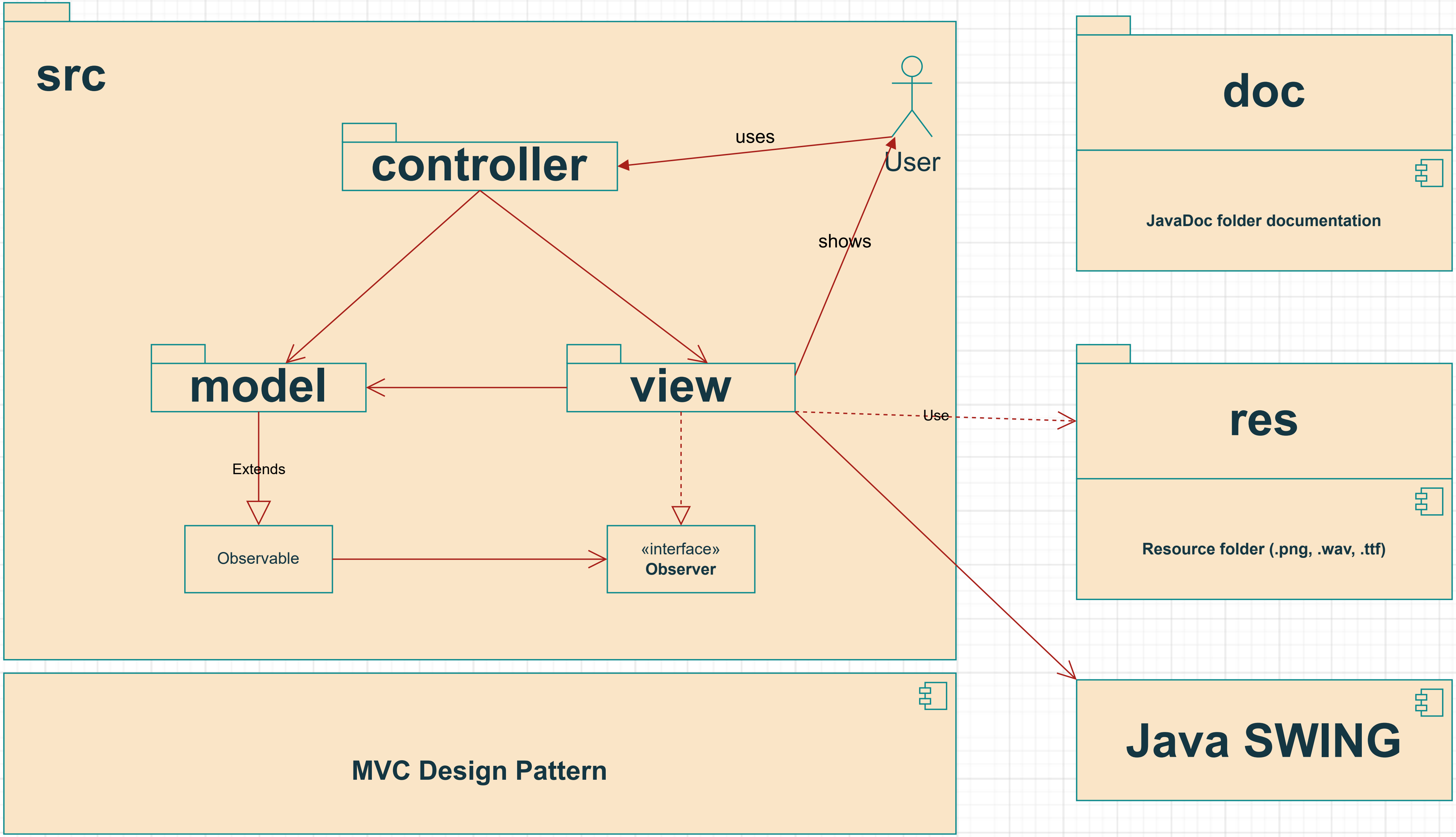
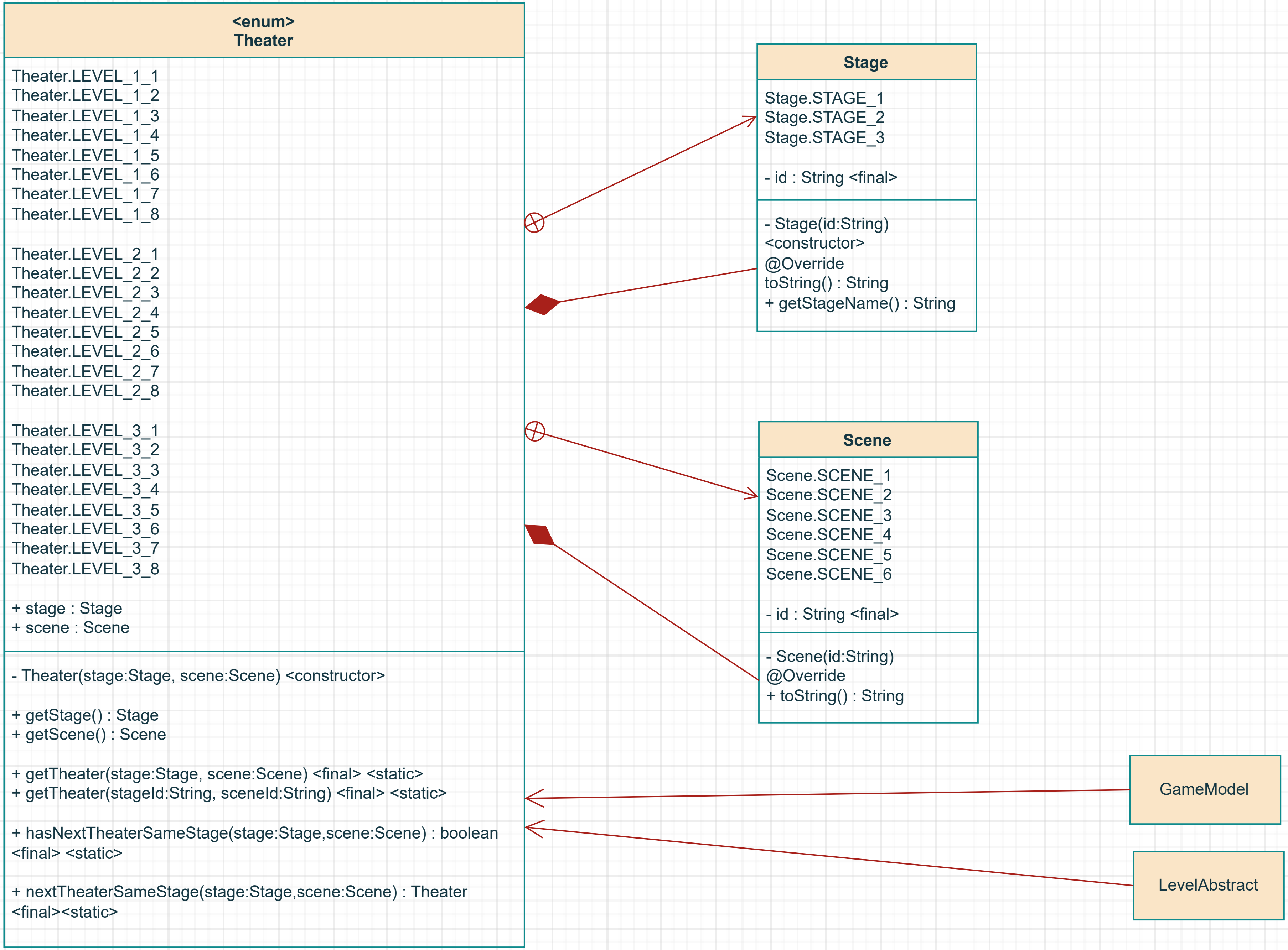
Allegretti Giulio #2029763 - Teledidattica- Metodologie Di Programmazione's Project 2023/2024 - JBomberman - Team 1 Persona - Diagramma UML delle classi



Model UML



<enum>
Theater

Theater.LEVEL_1_1
Theater.LEVEL_1_2
Theater.LEVEL_1_3
Theater.LEVEL_1_4
Theater.LEVEL_1_5
Theater.LEVEL_1_6
Theater.LEVEL_1_7
Theater.LEVEL_1_8

Theater.LEVEL_2_1
Theater.LEVEL_2_2
Theater.LEVEL_2_3
Theater.LEVEL_2_4
Theater.LEVEL_2_5
Theater.LEVEL_2_6
Theater.LEVEL_2_7
Theater.LEVEL_2_8

Theater.LEVEL_3_1
Theater.LEVEL_3_2
Theater.LEVEL_3_3
Theater.LEVEL_3_4
Theater.LEVEL_3_5
Theater.LEVEL_3_6
Theater.LEVEL_3_7
Theater.LEVEL_3_8

+ stage : Stage
+ scene : Scene

- Theater(stage:Stage, scene:Scene) <constructor>

+ getStage() : Stage
+ getScene() : Scene

+ getTheater(stage:Stage, scene:Scene) <final> <static>
+ getTheater(stageId:String, sceneId:String) <final> <static>

+ hasNextTheaterSameStage(stage:Stage,scene:Scene) : boolean <final> <static>

+ nextTheaterSameStage(stage:Stage,scene:Scene) : Theater <final><static>

Stage

Stage.STAGE_1
Stage.STAGE_2
Stage.STAGE_3

- id : String <final>

- Stage(id:String) <constructor>
@Override
toString() : String
+ getStageName() : String

Scene

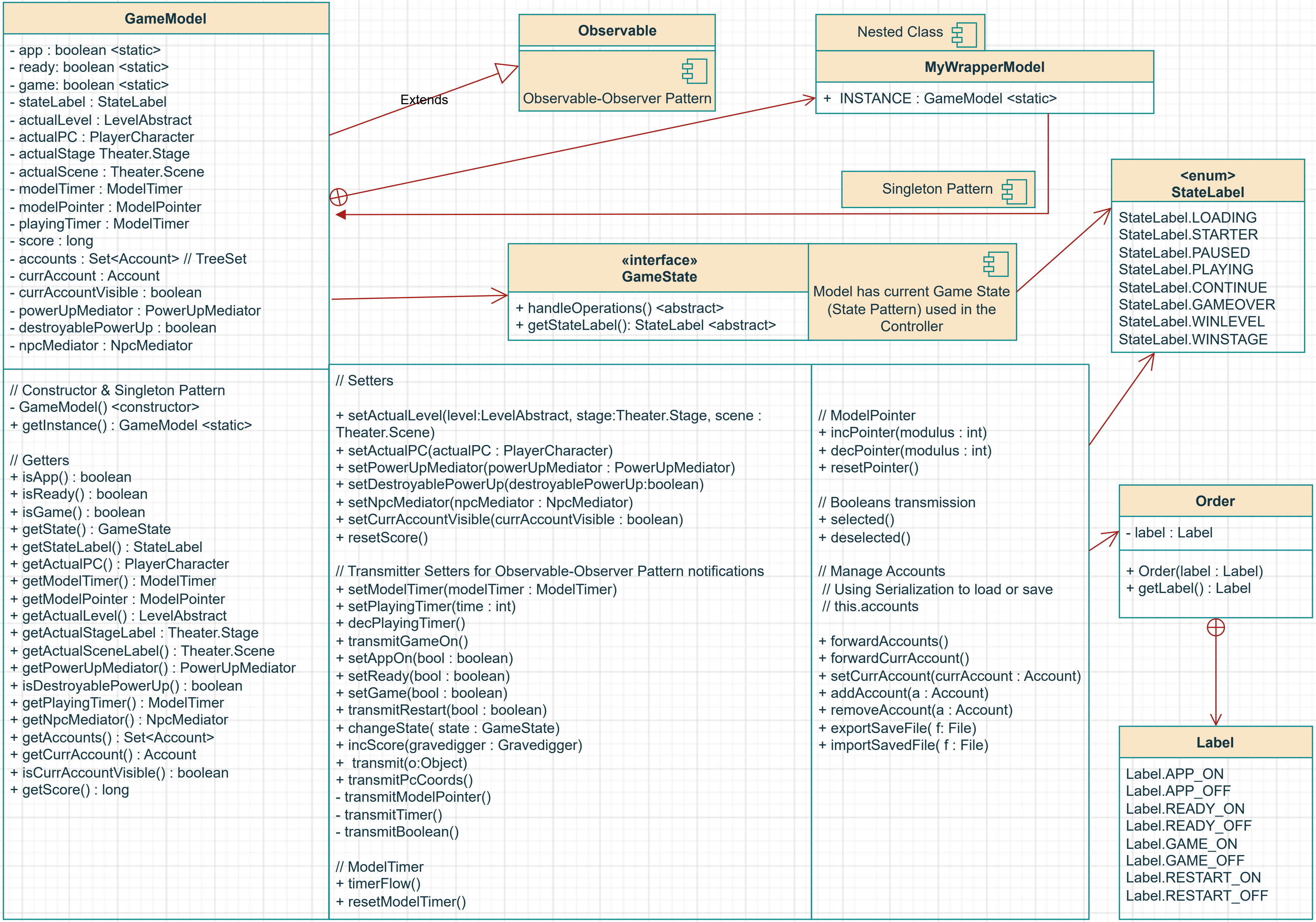
Scene.SCENE_1
Scene.SCENE_2
Scene.SCENE_3
Scene.SCENE_4
Scene.SCENE_5
Scene.SCENE_6

- id : String <final>

- Scene(id:String)
@Override
+ toString() : String

GameModel

LevelAbstract



GameModel

- app : boolean <static>
- ready: boolean <static>
- game: boolean <static>
- stateLabel : StateLabel
- actualLevel : LevelAbstract
- actualPC : PlayerCharacter
- actualStage Theater.Stage
- actualScene : Theater.Scene
- modelTimer : ModelTimer
- modelPointer : ModelPointer
- playingTimer : ModelTimer
- score : long
- accounts : Set<Account> // TreeSet
- currAccount : Account
- currAccountVisible : boolean
- powerUpMediator : PowerUpMediator
- destroyablePowerUp : boolean
- npcMediator : NpcMediator

// Constructor & Singleton Pattern
- GameModel() <constructor>
+ getInstance() : GameModel <static>

// Getters
+ isApp() : boolean
+ isReady() : boolean
+ isGame() : boolean
+ getState() : GameState
+ getStateLabel() : StateLabel
+ getActualPC() : PlayerCharacter
+ getModelTimer() : ModelTimer
+ getModelPointer : ModelPointer
+ getActualLevel() : LevelAbstract
+ getActualStageLabel : Theater.Stage
+ getActualSceneLabel() : Theater.Scene
+ getPowerUpMediator() : PowerUpMediator
+ isDestroyablePowerUp() : boolean
+ getPlayingTimer() : ModelTimer
+ getNpcMediator() : NpcMediator
+ getAccounts() : Set<Account>
+ getCurrAccount() : Account
+ isCurrAccountVisible() : boolean
+ getScore() : long

// Setters

+ setActualLevel(level:LevelAbstract, stage:Theater.Stage, scene : Theater.Scene)
+ setActualPC(actualPC : PlayerCharacter)
+ setPowerUpMediator(powerUpMediator : PowerUpMediator)
+ setDestroyablePowerUp(destroyablePowerUp:boolean)
+ setNpcMediator(npcMediator : NpcMediator)
+ setCurrAccountVisible(currAccountVisible : boolean)
+ resetScore()

// Transmitter Setters for Observable-Observer Pattern notifications
+ setModelTimer(modelTimer : ModelTimer)
+ setPlayingTimer(time : int)
+ decPlayingTimer()
+ transmitGameOn()
+ setAppOn(bool : boolean)
+ setReady(bool : boolean)
+ setGame(bool : boolean)
+ transmitRestart(bool : boolean)
+ changeState(state : GameState)
+ incScore(gravedigger : Gravedigger)
+ transmit(o:Object)
+ transmitPcCoords()
- transmitModelPointer()
- transmitTimer()
- transmitBoolean()

// ModelTimer
+ timerFlow()
+ resetModelTimer()

// ModelPointer
+ incPointer(modulus : int)
+ decPointer(modulus : int)
+ resetPointer()

// Booleans transmission
+ selected()
+ deselected()

// Manage Accounts
 // Using Serialization to load or save
 // this.accounts

+ forwardAccounts()
+ forwardCurrAccount()
+ setCurrAccount(currAccount : Account)
+ addAccount(a : Account)
+ removeAccount(a : Account)
+ exportSaveFile(f: File)
+ importSavedFile(f : File)

Extends

Observable

Observable-Observer Pattern

Nested Class

MyWrapperModel

+ INSTANCE : GameModel <static>

Singleton Pattern

«interface»
GameState

+ handleOperations() <abstract>
+ getStateLabel(): StateLabel <abstract>

Model has current Game State (State Pattern) used in the Controller

<enum>
StateLabel

StateLabel.LOADING
StateLabel.STARTER
StateLabel.PAUSED
StateLabel.PLAYING
StateLabel.CONTINUE
StateLabel.GAMEOVER
StateLabel.WINLEVEL
StateLabel.WINSTAGE

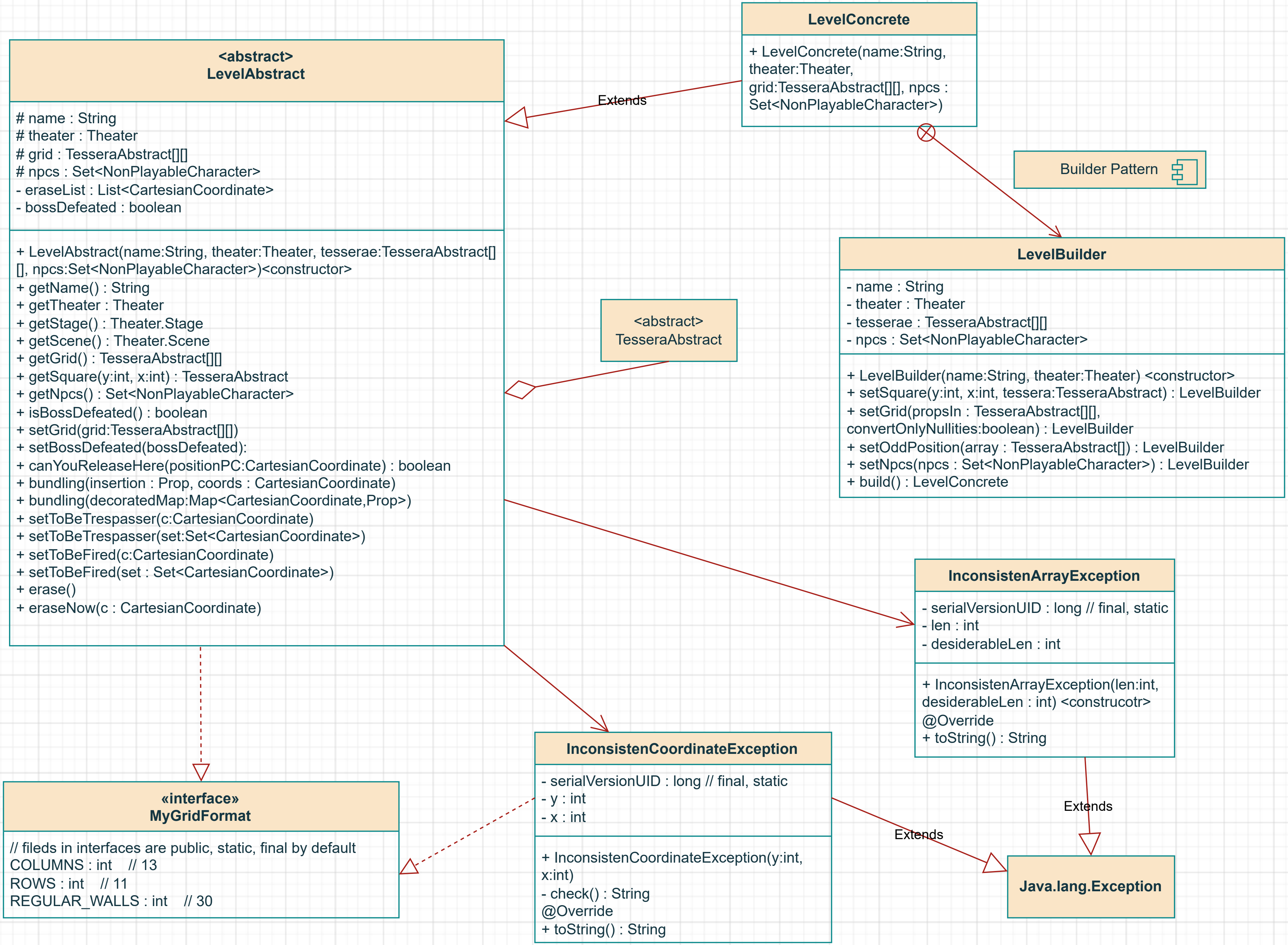
Order

- label : Label

+ Order(label : Label)
+ getLabel() : Label

Label

Label.APP_ON
Label.APP_OFF
Label.READY_ON
Label.READY_OFF
Label.GAME_ON
Label.GAME_OFF
Label.RESTART_ON
Label.RESTART_OFF



LevelConcrete

+ LevelConcrete(name:String, theater:Theater, grid:TesseraAbstract[][], npcs : Set<NonPlayableCharacter>)

<abstract>
LevelAbstract

# name : String
# theater : Theater
# grid : TesseraAbstract[][]
# npcs : Set<NonPlayableCharacter>
- eraseList : List<CartesianCoordinate>
- bossDefeated : boolean

+ LevelAbstract(name:String, theater:Theater, tesserae:TesseraAbstract[][], npcs:Set<NonPlayableCharacter>)<constructor>
+ getName() : String
+ getTheater : Theater
+ getStage() : Theater.Stage
+ getScene() : Theater.Scene
+ getGrid() : TesseraAbstract[][]
+ getSquare(y:int, x:int) : TesseraAbstract
+ getNpcs() : Set<NonPlayableCharacter>
+ isBossDefeated() : boolean
+ setGrid(grid:TesseraAbstract[][])
+ setBossDefeated(bossDefeated):
+ canYouReleaseHere(positionPC:CartesianCoordinate) : boolean
+ bundling(insertion : Prop, coords : CartesianCoordinate)
+ bundling(decoratedMap:Map<CartesianCoordinate,Prop>)
+ setToBeTrespasser(c:CartesianCoordinate)
+ setToBeTrespasser(set:Set<CartesianCoordinate>)
+ setToBeFired(c:CartesianCoordinate)
+ setToBeFired(set : Set<CartesianCoordinate>)
+ erase()
+ eraseNow(c : CartesianCoordinate)

<abstract>
TesseraAbstract

LevelBuilder

- name : String
- theater : Theater
- tesserae : TesseraAbstract[][]
- npcs : Set<NonPlayableCharacter>

+ LevelBuilder(name:String, theater:Theater) <constructor>
+ setSquare(y:int, x:int, tessera:TesseraAbstract) : LevelBuilder
+ setGrid(propsIn : TesseraAbstract[][], convertOnlyNullities:boolean) : LevelBuilder
+ setOddPosition(array : TesseraAbstract[]) : LevelBuilder
+ setNpcs(npcs : Set<NonPlayableCharacter>) : LevelBuilder
+ build() : LevelConcrete

InconsistenArrayException

- serialVersionUID : long // final, static
- len : int
- desiderableLen : int

+ InconsistenArrayException(len:int, desiderableLen : int) <construcotr>
@Override
+ toString() : String

«interface»
MyGridFormat

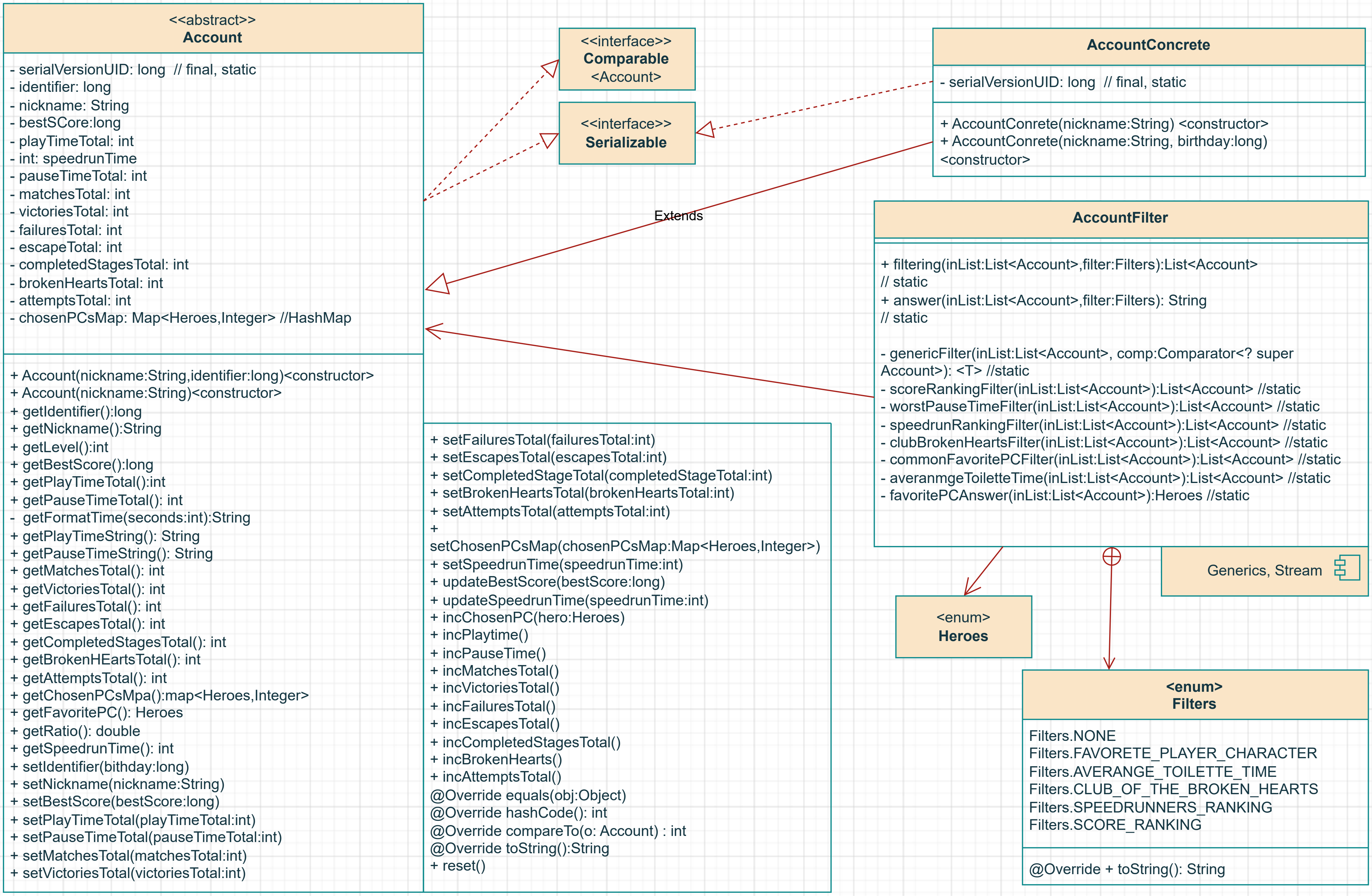
// fileds in interfaces are public, static, final by default
COLUMNS : int // 13
ROWS : int // 11
REGULAR_WALLS : int // 30

InconsistenCoordinateException

- serialVersionUID : long // final, static
- y : int
- x : int

+ InconsistenCoordinateException(y:int, x:int)
- check() : String
@Override
+ toString() : String

Java.lang.Exception



<<abstract>>
Account

- serialVersionUID: long // final, static
- identifier: long
- nickname: String
- bestSCore:long
- playTimeTotal: int
- int: speedrunTime
- pauseTimeTotal: int
- matchesTotal: int
- victoriesTotal: int
- failuresTotal: int
- escapeTotal: int
- completedStagesTotal: int
- brokenHeartsTotal: int
- attemptsTotal: int
- chosenPCsMap: Map<Heroes,Integer> //HashMap

+ Account(nickname:String,identifier:long)<constructor>
+ Account(nickname:String)<constructor>
+ getIdentifier():long
+ getNickname():String
+ getLevel():int
+ getBestScore():long
+ getPlayTimeTotal():int
+ getPauseTimeTotal(): int
- getFormatTime(seconds:int):String
+ getPlayTimeString(): String
+ getPauseTimeString(): String
+ getMatchesTotal(): int
+ getVictoriesTotal(): int
+ getFailuresTotal(): int
+ getEscapesTotal(): int
+ getCompletedStagesTotal(): int
+ getBrokenHEartsTotal(): int
+ getAttemptsTotal(): int
+ getChosenPCsMpa():map<Heroes,Integer>
+ getFavoritePC(): Heroes
+ getRatio(): double
+ getSpeedrunTime(): int
+ setIdentifier(bithday:long)
+ setNickname(nickname:String)
+ setBestScore(bestScore:long)
+ setPlayTimeTotal(playTimeTotal:int)
+ setPauseTimeTotal(pauseTimeTotal:int)
+ setMatchesTotal(matchesTotal:int)
+ setVictoriesTotal(victoriesTotal:int)

+ setFailuresTotal(failuresTotal:int)
+ setEscapesTotal(escapesTotal:int)
+ setCompletedStageTotal(completedStageTotal:int)
+ setBrokenHeartsTotal(brokenHeartsTotal:int)
+ setAttemptsTotal(attemptsTotal:int)
+ setChosenPCsMap(chosenPCsMap:Map<Heroes,Integer>)
+ setSpeedrunTime(speedrunTime:int)
+ updateBestScore(bestScore:long)
+ updateSpeedrunTime(speedrunTime:int)
+ incChosenPC(hero:Heroes)
+ incPlaytime()
+ incPauseTime()
+ incMatchesTotal()
+ incVictoriesTotal()
+ incFailuresTotal()
+ incEscapesTotal()
+ incCompletedStagesTotal()
+ incBrokenHearts()
+ incAttemptsTotal()

@Override equals(obj:Object)
@Override hashCode(): int
@Override compareTo(o: Account) : int
@Override toString():String
+ reset()

<<interface>>
Comparable
<Account>

<<interface>>
Serializable

AccountConcrete

- serialVersionUID: long // final, static

+ AccountConrete(nickname:String) <constructor>
+ AccountConrete(nickname:String, birthday:long) <constructor>

Extends

AccountFilter

+ filtering(inList:List<Account>,filter:Filters):List<Account> // static
+ answer(inList:List<Account>,filter:Filters): String // static

- genericFilter(inList:List<Account>, comp:Comparator<? super Account>): <T> //static
- scoreRankingFilter(inList:List<Account>):List<Account> //static
- worstPauseTimeFilter(inList:List<Account>):List<Account> //static
- speedrunRankingFilter(inList:List<Account>):List<Account> //static
- clubBrokenHeartsFilter(inList:List<Account>):List<Account> //static
- commonFavoritePCFilter(inList:List<Account>):List<Account> //static
- averanmgeToiletteTime(inList:List<Account>):List<Account> //static
- favoritePCAnswer(inList:List<Account>):Heroes //static

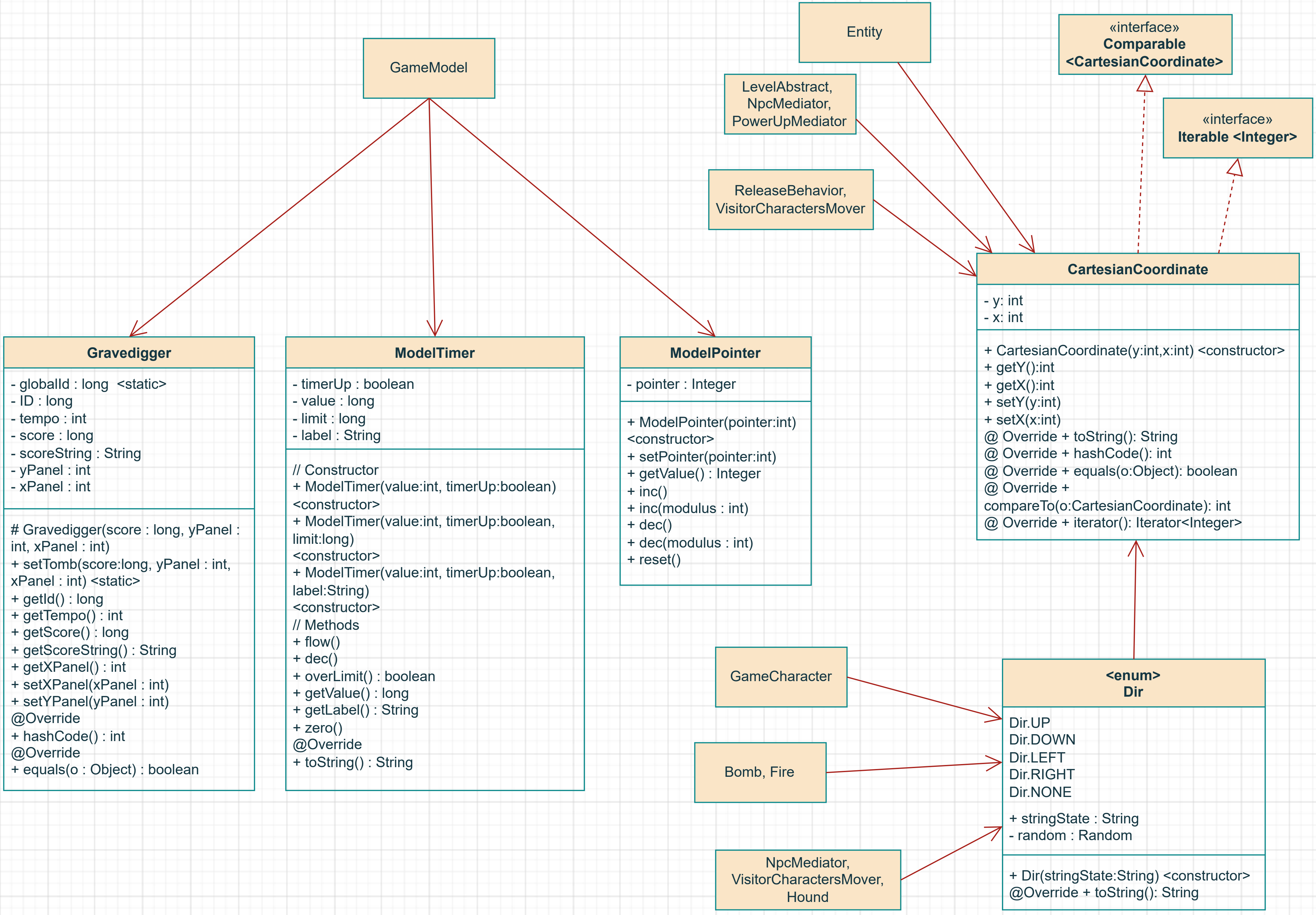
Generics, Stream

<enum>
Heroes

<enum>
Filters

Filters.NONE
Filters.FAVORETE_PLAYER_CHARACTER
Filters.AVERANGE_TOILETTE_TIME
Filters.CLUB_OF_THE_BROKEN_HEARTS
Filters.SPEEDRUNNERS_RANKING
Filters.SCORE_RANKING

@Override + toString(): String



GameModel

Entity

LevelAbstract,
NpcMediator,
PowerUpMediator

ReleaseBehavior,
VisitorCharactersMover

«interface»
Comparable <CartesianCoordinate>

«interface»
Iterable <Integer>

Gravedigger

- globalId : long <static>
- ID : long
- tempo : int
- score : long
- scoreString : String
- yPanel : int
- xPanel : int

# Gravedigger(score : long, yPanel : int, xPanel : int)
+ setTomb(score:long, yPanel : int, xPanel : int) <static>
+ getId() : long
+ getTempo() : int
+ getScore() : long
+ getScoreString() : String
+ getXPanel() : int
+ setXPanel(xPanel : int)
+ setYPanel(yPanel : int)
@Override
+ hashCode() : int
@Override
+ equals(o : Object) : boolean

ModelTimer

- timerUp : boolean
- value : long
- limit : long
- label : String

// Constructor
+ ModelTimer(value:int, timerUp:boolean) <constructor>
+ ModelTimer(value:int, timerUp:boolean, limit:long) <constructor>
+ ModelTimer(value:int, timerUp:boolean, label:String) <constructor>
// Methods
+ flow()
+ dec()
+ overLimit() : boolean
+ getValue() : long
+ getLabel() : String
+ zero()
@Override
+ toString() : String

ModelPointer

- pointer : Integer

+ ModelPointer(pointer:int) <constructor>
+ setPointer(pointer:int)
+ getValue() : Integer
+ inc()
+ inc(modulus : int)
+ dec()
+ dec(modulus : int)
+ reset()

CartesianCoordinate

- y: int
- x: int

+ CartesianCoordinate(y:int,x:int) <constructor>
+ getY():int
+ getX():int
+ setY(y:int)
+ setX(x:int)
@ Override + toString(): String
@ Override + hashCode(): int
@ Override + equals(o:Object): boolean
@ Override + compareTo(o:CartesianCoordinate): int
@ Override + iterator(): Iterator<Integer>

GameCharacter

Bomb, Fire

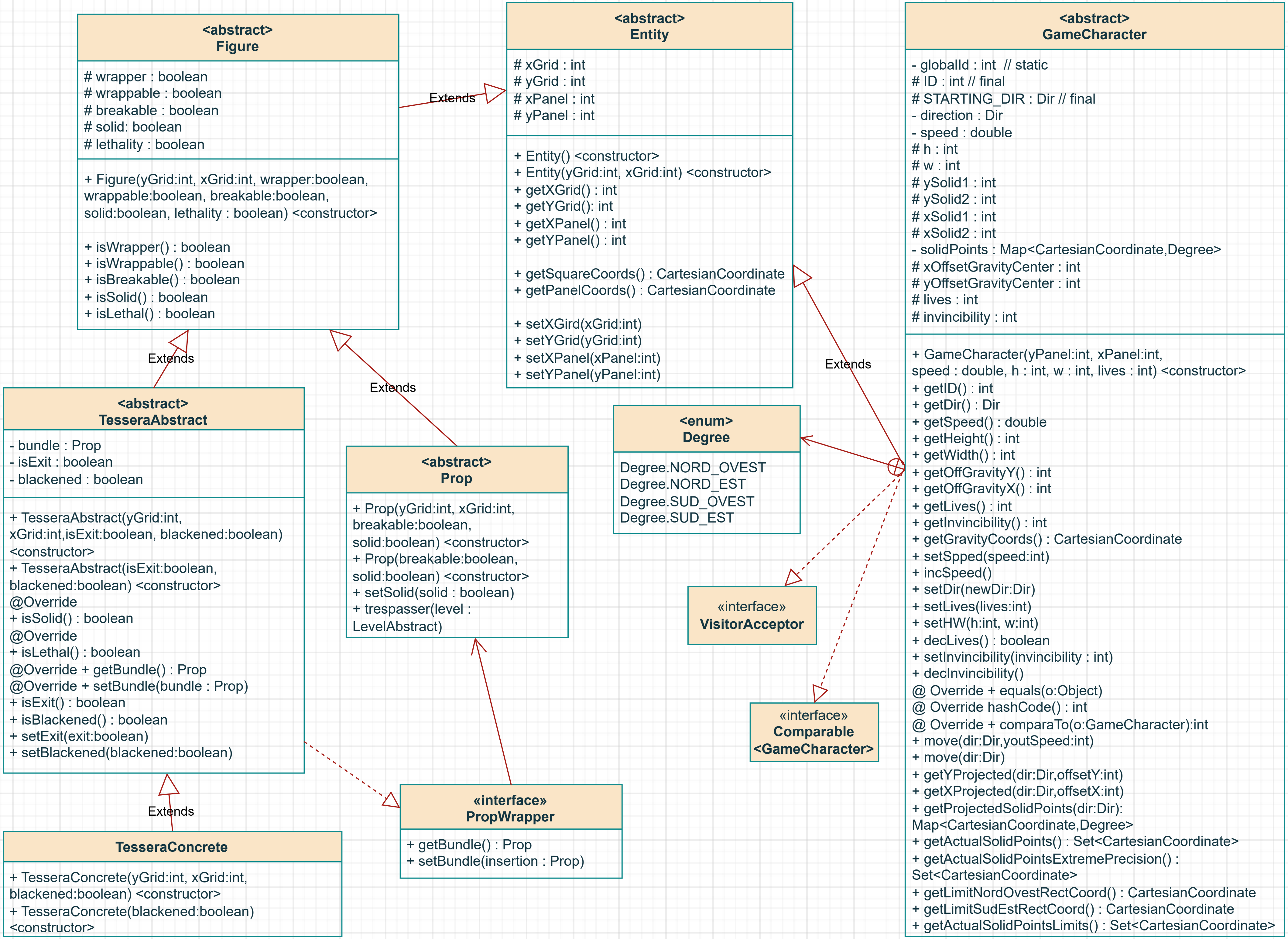
NpcMediator,
VisitorCharactersMover,
Hound

<enum>
Dir

Dir.UP
Dir.DOWN
Dir.LEFT
Dir.RIGHT
Dir.NONE

+ stringState : String
- random : Random

+ Dir(stringState:String) <constructor>
@Override + toString(): String



<abstract>
Figure

# wrapper : boolean
# wrappable : boolean
# breakable : boolean
# solid: boolean
# lethality : boolean

+ Figure(yGrid:int, xGrid:int, wrapper:boolean, wrappable:boolean, breakable:boolean, solid:boolean, lethality : boolean) <constructor>

+ isWrapper() : boolean
+ isWrappable() : boolean
+ isBreakable() : boolean
+ isSolid() : boolean
+ isLethal() : boolean

<abstract>
Entity

# xGrid : int
# yGrid : int
# xPanel : int
# yPanel : int

+ Entity() <constructor>
+ Entity(yGrid:int, xGrid:int) <constructor>
+ getXGrid() : int
+ getYGrid(): int
+ getXPanel() : int
+ getYPanel() : int

+ getSquareCoords() : CartesianCoordinate
+ getPanelCoords() : CartesianCoordinate

+ setXGird(xGrid:int)
+ setYGrid(yGrid:int)
+ setXPanel(xPanel:int)
+ setYPanel(yPanel:int)

<abstract>
GameCharacter

- globalId : int // static
# ID : int // final
# STARTING_DIR : Dir // final
- direction : Dir
- speed : double
# h : int
# w : int
# ySolid1 : int
# ySolid2 : int
# xSolid1 : int
# xSolid2 : int
- solidPoints : Map<CartesianCoordinate,Degree>
# xOffsetGravityCenter : int
# yOffsetGravityCenter : int
# lives : int
# invincibility : int

+ GameCharacter(yPanel:int, xPanel:int, speed : double, h : int, w : int, lives : int) <constructor>
+ getID() : int
+ getDir() : Dir
+ getSpeed() : double
+ getHeight() : int
+ getWidth() : int
+ getOffGravityY() : int
+ getOffGravityX() : int
+ getLives() : int
+ getInvincibility() : int
+ getGravityCoords() : CartesianCoordinate
+ setSpped(speed:int)
+ incSpeed()
+ setDir(newDir:Dir)
+ setLives(lives:int)
+ setHW(h:int, w:int)
+ decLives() : boolean
+ setInvincibility(invincibility : int)
+ decInvincibility()
@ Override + equals(o:Object)
@ Override hashCode() : int
@ Override + comparaTo(o:GameCharacter):int
+ move(dir:Dir,youtSpeed:int)
+ move(dir:Dir)
+ getYProjected(dir:Dir,offsetY:int)
+ getXProjected(dir:Dir,offsetX:int)
+ getProjectedSolidPoints(dir:Dir): Map<CartesianCoordinate,Degree>
+ getActualSolidPoints() : Set<CartesianCoordinate>
+ getActualSolidPointsExtremePrecision() : Set<CartesianCoordinate>
+ getLimitNordOvestRectCoord() : CartesianCoordinate
+ getLimitSudEstRectCoord() : CartesianCoordinate
+ getActualSolidPointsLimits() : Set<CartesianCoordinate>

<abstract>
TesseraAbstract

- bundle : Prop
- isExit : boolean
- blackened : boolean

+ TesseraAbstract(yGrid:int, xGrid:int,isExit:boolean, blackened:boolean) <constructor>
+ TesseraAbstract(isExit:boolean, blackened:boolean) <constructor>
@Override
+ isSolid() : boolean
@Override
+ isLethal() : boolean
@Override + getBundle() : Prop
@Override + setBundle(bundle : Prop)
+ isExit() : boolean
+ isBlackened() : boolean
+ setExit(exit:boolean)
+ setBlackened(blackened:boolean)

<abstract>
Prop

+ Prop(yGrid:int, xGrid:int, breakable:boolean, solid:boolean) <constructor>
+ Prop(breakable:boolean, solid:boolean) <constructor>
+ setSolid(solid : boolean)
+ trespasser(level : LevelAbstract)

<enum>
Degree

Degree.NORD_OVEST
Degree.NORD_EST
Degree.SUD_OVEST
Degree.SUD_EST

«interface»
VisitorAcceptor

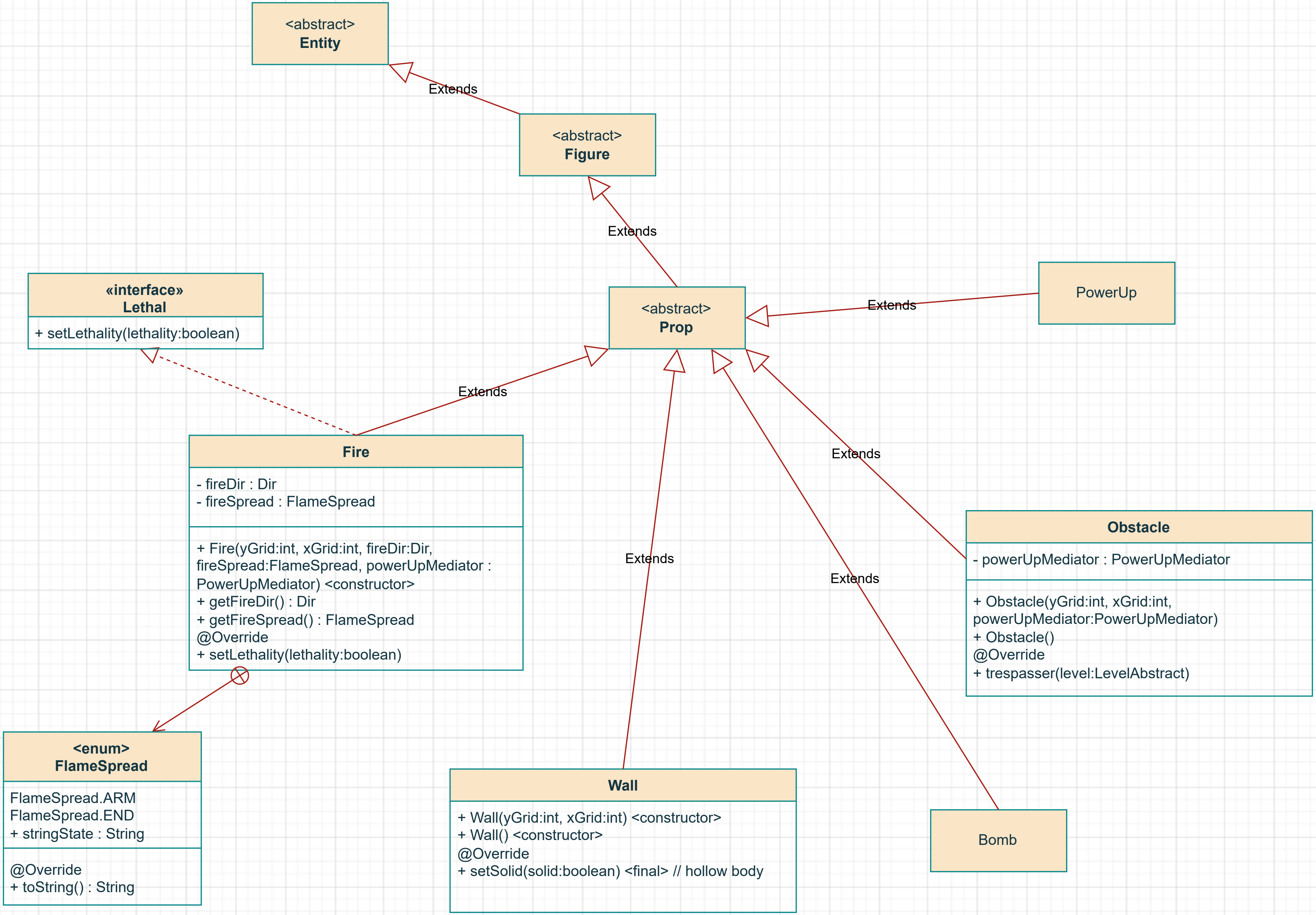
«interface»
Comparable <GameCharacter>

TesseraConcrete

+ TesseraConcrete(yGrid:int, xGrid:int, blackened:boolean) <constructor>
+ TesseraConcrete(blackened:boolean) <constructor>

«interface»
PropWrapper

+ getBundle() : Prop
+ setBundle(insertion : Prop)



<abstract>
Entity

<abstract>
Figure

«interface»
Lethal

+ setLethality(lethality:boolean)

<abstract>
Prop

PowerUp

Fire

- fireDir : Dir
- fireSpread : FlameSpread

+ Fire(yGrid:int, xGrid:int, fireDir:Dir, fireSpread:FlameSpread, powerUpMediator : PowerUpMediator) <constructor>
+ getFireDir() : Dir
+ getFireSpread() : FlameSpread
@Override
+ setLethality(lethality:boolean)

Obstacle

- powerUpMediator : PowerUpMediator

+ Obstacle(yGrid:int, xGrid:int, powerUpMediator:PowerUpMediator)
+ Obstacle()
@Override
+ trespasser(level:LevelAbstract)

<enum>
FlameSpread

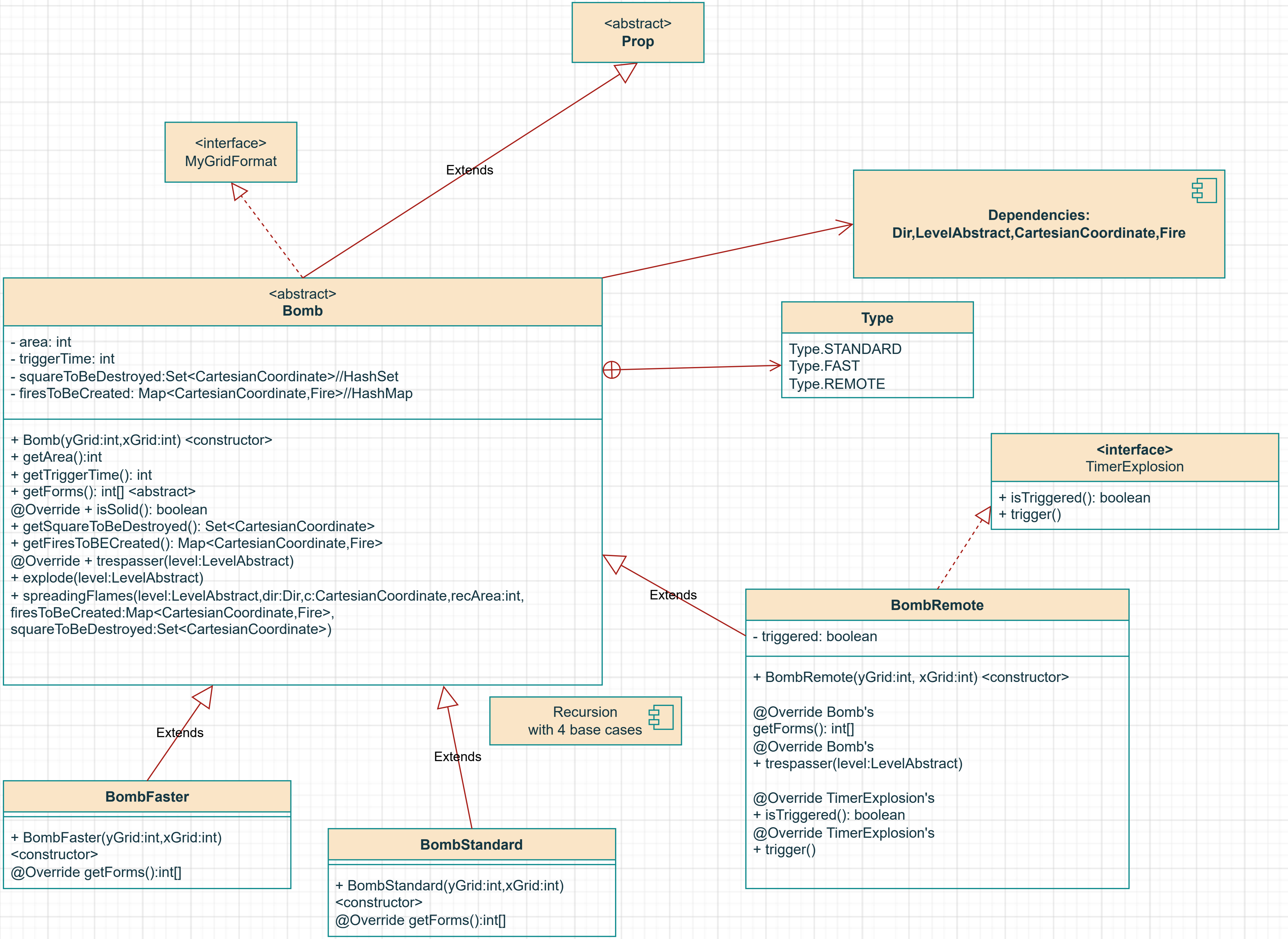
FlameSpread.ARM
FlameSpread.END
+ stringState : String

@Override
+ toString() : String

Wall

+ Wall(yGrid:int, xGrid:int) <constructor>
+ Wall() <constructor>
@Override
+ setSolid(solid:boolean) <final> // hollow body

Bomb



<abstract>
Prop

<interface>
MyGridFormat

Dependencies:
Dir,LevelAbstract,CartesianCoordinate,Fire

<abstract>
Bomb

- area: int
- triggerTime: int
- squareToBeDestroyed:Set<CartesianCoordinate>//HashSet
- firesToBeCreated: Map<CartesianCoordinate,Fire>//HashMap

+ Bomb(yGrid:int,xGrid:int) <constructor>
+ getArea():int
+ getTriggerTime(): int
+ getForms(): int[] <abstract>
@Override + isSolid(): boolean
+ getSquareToBeDestroyed(): Set<CartesianCoordinate>
+ getFiresToBECreated(): Map<CartesianCoordinate,Fire>
@Override + trespasser(level:LevelAbstract)
+ explode(level:LevelAbstract)
+ spreadingFlames(level:LevelAbstract,dir:Dir,c:CartesianCoordinate,recArea:int,
firesToBeCreated:Map<CartesianCoordinate,Fire>,
squareToBeDestroyed:Set<CartesianCoordinate>)

Type

Type.STANDARD
Type.FAST
Type.REMOTE

<interface>
TimerExplosion

+ isTriggered(): boolean
+ trigger()

BombRemote

- triggered: boolean

+ BombRemote(yGrid:int, xGrid:int) <constructor>

@Override Bomb's
getForms(): int[]
@Override Bomb's
+ trespasser(level:LevelAbstract)

@Override TimerExplosion's
+ isTriggered(): boolean
@Override TimerExplosion's
+ trigger()

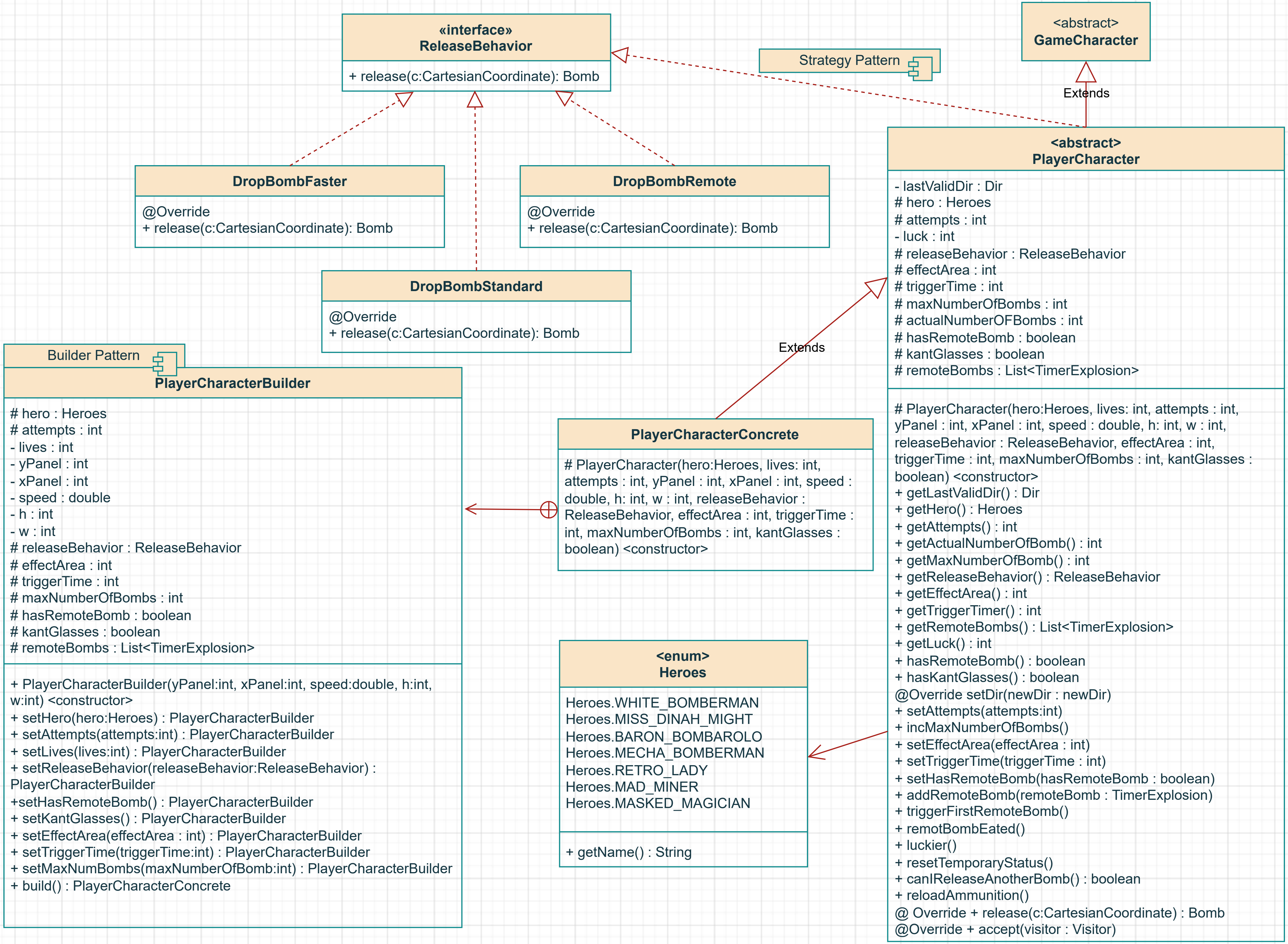
Recursion
with 4 base cases

BombFaster

+ BombFaster(yGrid:int,xGrid:int)
<constructor>
@Override getForms():int[]

BombStandard

+ BombStandard(yGrid:int,xGrid:int)
<constructor>
@Override getForms():int[]



«interface»
ReleaseBehavior

+ release(c:CartesianCoordinate): Bomb

DropBombFaster

@Override
+ release(c:CartesianCoordinate): Bomb

DropBombRemote

@Override
+ release(c:CartesianCoordinate): Bomb

DropBombStandard

@Override
+ release(c:CartesianCoordinate): Bomb

<abstract>
GameCharacter

<abstract>
PlayerCharacter

- lastValidDir : Dir
# hero : Heroes
# attempts : int
- luck : int
# releaseBehavior : ReleaseBehavior
# effectArea : int
# triggerTime : int
# maxNumberOfBombs : int
# actualNumberOFBombs : int
# hasRemoteBomb : boolean
# kantGlasses : boolean
# remoteBombs : List<TimerExplosion>

# PlayerCharacter(hero:Heroes, lives: int, attempts : int, yPanel : int, xPanel : int, speed : double, h: int, w : int, releaseBehavior : ReleaseBehavior, effectArea : int, triggerTime : int, maxNumberOfBombs : int, kantGlasses : boolean) <constructor>
+ getLastValidDir() : Dir
+ getHero() : Heroes
+ getAttempts() : int
+ getActualNumberOfBomb() : int
+ getMaxNumberOfBomb() : int
+ getReleaseBehavior() : ReleaseBehavior
+ getEffectArea() : int
+ getTriggerTimer() : int
+ getRemoteBombs() : List<TimerExplosion>
+ getLuck() : int
+ hasRemoteBomb() : boolean
+ hasKantGlasses() : boolean
@Override setDir(newDir : newDir)
+ setAttempts(attempts:int)
+ incMaxNumberOfBombs()
+ setEffectArea(effectArea : int)
+ setTriggerTime(triggerTime : int)
+ setHasRemoteBomb(hasRemoteBomb : boolean)
+ addRemoteBomb(remoteBomb : TimerExplosion)
+ triggerFirstRemoteBomb()
+ remotBombEated()
+ luckier()
+ resetTemporaryStatus()
+ canIReleaseAnotherBomb() : boolean
+ reloadAmmunition()
@ Override + release(c:CartesianCoordinate) : Bomb
@Override + accept(visitor : Visitor)

PlayerCharacterBuilder

# hero : Heroes
# attempts : int
- lives : int
- yPanel : int
- xPanel : int
- speed : double
- h : int
- w : int
# releaseBehavior : ReleaseBehavior
# effectArea : int
# triggerTime : int
# maxNumberOfBombs : int
# hasRemoteBomb : boolean
# kantGlasses : boolean
# remoteBombs : List<TimerExplosion>

+ PlayerCharacterBuilder(yPanel:int, xPanel:int, speed:double, h:int, w:int) <constructor>
+ setHero(hero:Heroes) : PlayerCharacterBuilder
+ setAttempts(attempts:int) : PlayerCharacterBuilder
+ setLives(lives:int) : PlayerCharacterBuilder
+ setReleaseBehavior(releaseBehavior:ReleaseBehavior) : PlayerCharacterBuilder
+setHasRemoteBomb() : PlayerCharacterBuilder
+ setKantGlasses() : PlayerCharacterBuilder
+ setEffectArea(effectArea : int) : PlayerCharacterBuilder
+ setTriggerTime(triggerTime:int) : PlayerCharacterBuilder
+ setMaxNumBombs(maxNumberOfBomb:int) : PlayerCharacterBuilder
+ build() : PlayerCharacterConcrete

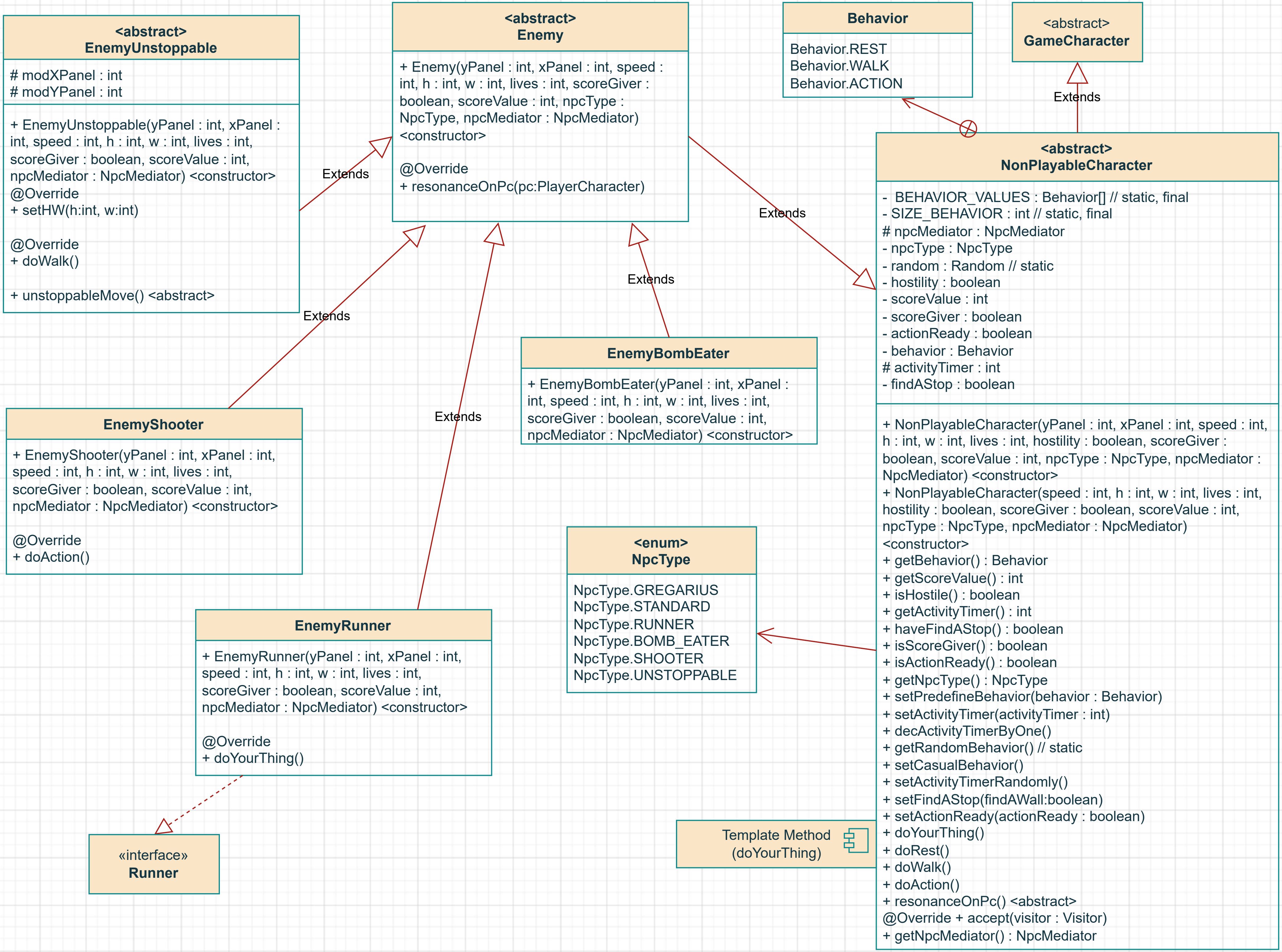
PlayerCharacterConcrete

# PlayerCharacter(hero:Heroes, lives: int, attempts : int, yPanel : int, xPanel : int, speed : double, h: int, w : int, releaseBehavior : ReleaseBehavior, effectArea : int, triggerTime : int, maxNumberOfBombs : int, kantGlasses : boolean) <constructor>

<enum>
Heroes

Heroes.WHITE_BOMBERMAN
Heroes.MISS_DINAH_MIGHT
Heroes.BARON_BOMBAROLO
Heroes.MECHA_BOMBERMAN
Heroes.RETRO_LADY
Heroes.MAD_MINER
Heroes.MASKED_MAGICIAN

+ getName() : String



<abstract>
EnemyUnstoppable

# modXPanel : int
# modYPanel : int

+ EnemyUnstoppable(yPanel : int, xPanel : int, speed : int, h : int, w : int, lives : int, scoreGiver : boolean, scoreValue : int, npcMediator : NpcMediator) <constructor>
@Override
+ setHW(h:int, w:int)

@Override
+ doWalk()

+ unstoppableMove() <abstract>

<abstract>
Enemy

+ Enemy(yPanel : int, xPanel : int, speed : int, h : int, w : int, lives : int, scoreGiver : boolean, scoreValue : int, npcType : NpcType, npcMediator : NpcMediator) <constructor>

@Override
+ resonanceOnPc(pc:PlayerCharacter)

Behavior

Behavior.REST
Behavior.WALK
Behavior.ACTION

<abstract>
GameCharacter

<abstract>
NonPlayableCharacter

- BEHAVIOR_VALUES : Behavior[] // static, final
- SIZE_BEHAVIOR : int // static, final
# npcMediator : NpcMediator
- npcType : NpcType
- random : Random // static
- hostility : boolean
- scoreValue : int
- scoreGiver : boolean
- actionReady : boolean
- behavior : Behavior
# activityTimer : int
- findAStop : boolean

+ NonPlayableCharacter(yPanel : int, xPanel : int, speed : int, h : int, w : int, lives : int, hostility : boolean, scoreGiver : boolean, scoreValue : int, npcType : NpcType, npcMediator : NpcMediator) <constructor>
+ NonPlayableCharacter(speed : int, h : int, w : int, lives : int, hostility : boolean, scoreGiver : boolean, scoreValue : int, npcType : NpcType, npcMediator : NpcMediator) <constructor>
+ getBehavior() : Behavior
+ getScoreValue() : int
+ isHostile() : boolean
+ getActivityTimer() : int
+ haveFindAStop() : boolean
+ isScoreGiver() : boolean
+ isActionReady() : boolean
+ getNpcType() : NpcType
+ setPredefineBehavior(behavior : Behavior)
+ setActivityTimer(activityTimer : int)
+ decActivityTimerByOne()
+ getRandomBehavior() // static
+ setCasualBehavior()
+ setActivityTimerRandomly()
+ setFindAStop(findAWall:boolean)
+ setActionReady(actionReady : boolean)
+ doYourThing()
+ doRest()
+ doWalk()
+ doAction()
+ resonanceOnPc() <abstract>
@Override + accept(visitor : Visitor)
+ getNpcMediator() : NpcMediator

EnemyBombEater

+ EnemyBombEater(yPanel : int, xPanel : int, speed : int, h : int, w : int, lives : int, scoreGiver : boolean, scoreValue : int, npcMediator : NpcMediator) <constructor>

EnemyShooter

+ EnemyShooter(yPanel : int, xPanel : int, speed : int, h : int, w : int, lives : int, scoreGiver : boolean, scoreValue : int, npcMediator : NpcMediator) <constructor>

@Override
+ doAction()

<enum>
NpcType

NpcType.GREGARIUS
NpcType.STANDARD
NpcType.RUNNER
NpcType.BOMB_EATER
NpcType.SHOOTER
NpcType.UNSTOPPABLE

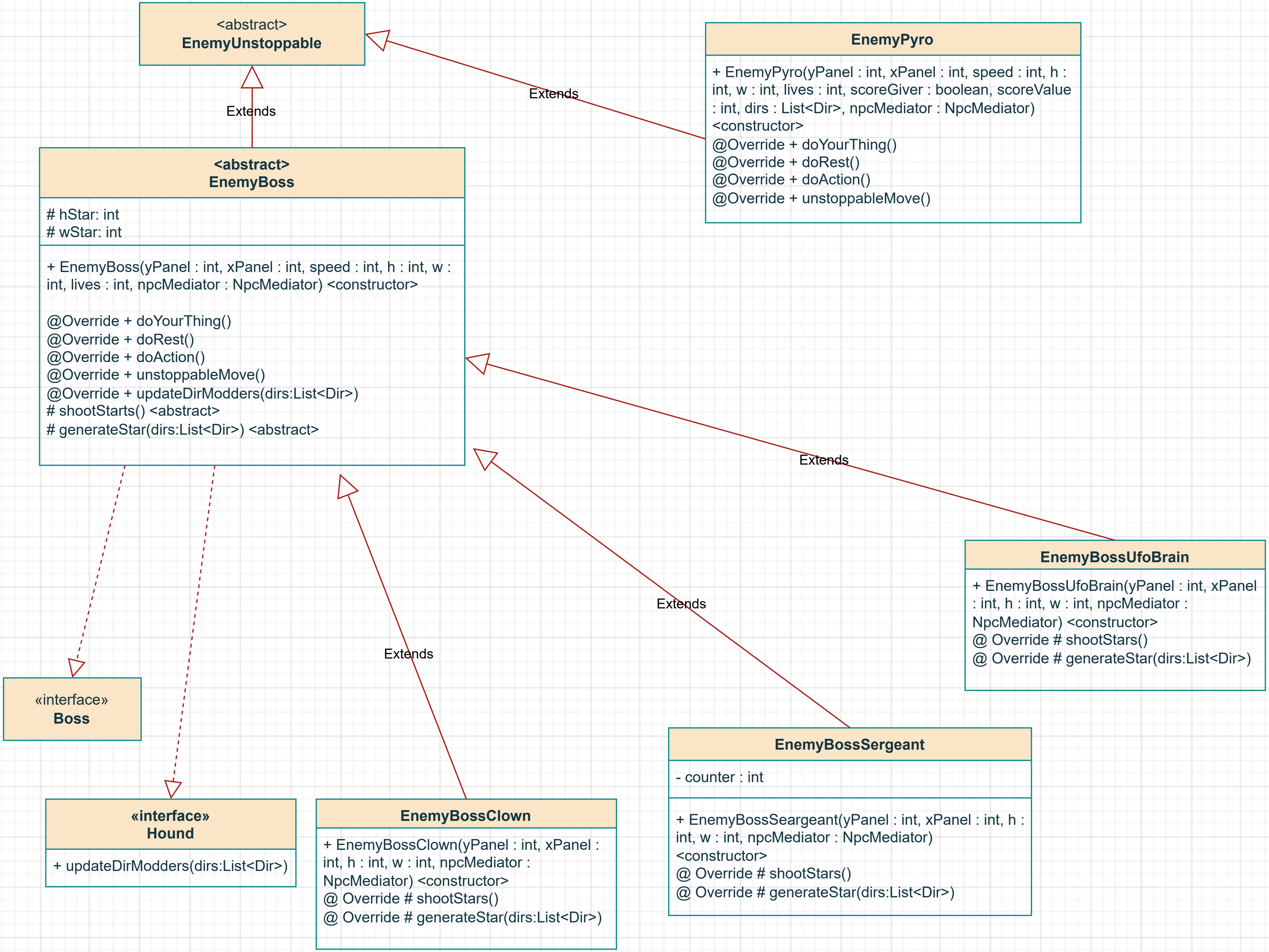
EnemyRunner

+ EnemyRunner(yPanel : int, xPanel : int, speed : int, h : int, w : int, lives : int, scoreGiver : boolean, scoreValue : int, npcMediator : NpcMediator) <constructor>

@Override
+ doYourThing()

«interface»
Runner

Template Method
(doYourThing)



<abstract>
EnemyUnstoppable

EnemyPyro

+ EnemyPyro(yPanel : int, xPanel : int, speed : int, h : int, w : int, lives : int, scoreGiver : boolean, scoreValue : int, dirs : List<Dir>, npcMediator : NpcMediator) <constructor>
@Override + doYourThing()
@Override + doRest()
@Override + doAction()
@Override + unstoppableMove()

<abstract>
EnemyBoss

# hStar: int
# wStar: int

+ EnemyBoss(yPanel : int, xPanel : int, speed : int, h : int, w : int, lives : int, npcMediator : NpcMediator) <constructor>

@Override + doYourThing()
@Override + doRest()
@Override + doAction()
@Override + unstoppableMove()
@Override + updateDirModders(dirs:List<Dir>)
# shootStarts() <abstract>
# generateStar(dirs:List<Dir>) <abstract>

EnemyBossUfoBrain

+ EnemyBossUfoBrain(yPanel : int, xPanel : int, h : int, w : int, npcMediator : NpcMediator) <constructor>
@ Override # shootStars()
@ Override # generateStar(dirs:List<Dir>)

«interface»
Boss

EnemyBossSergeant

- counter : int

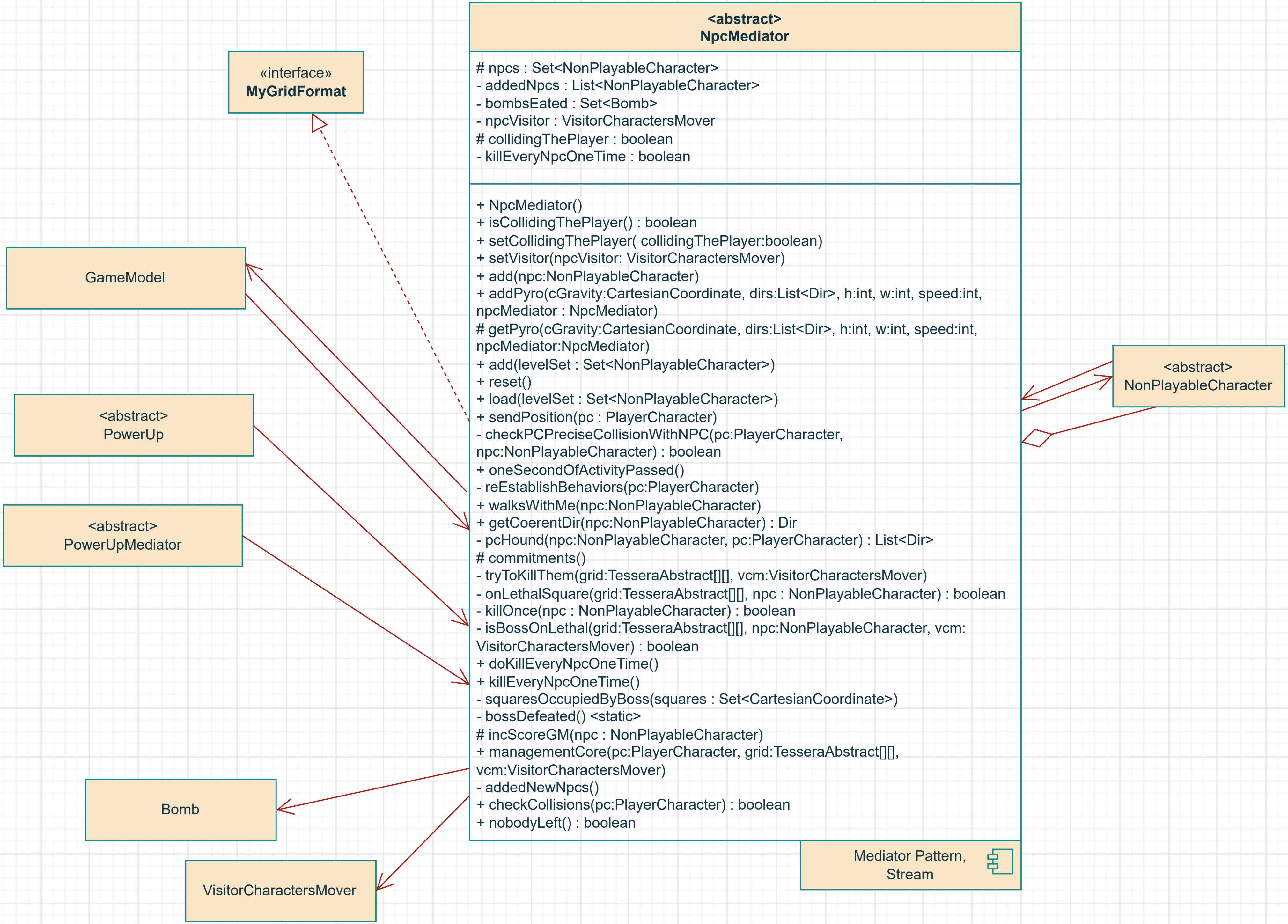
+ EnemyBossSeargeant(yPanel : int, xPanel : int, h : int, w : int, npcMediator : NpcMediator) <constructor>
@ Override # shootStars()
@ Override # generateStar(dirs:List<Dir>)

«interface»
Hound

+ updateDirModders(dirs:List<Dir>)

EnemyBossClown

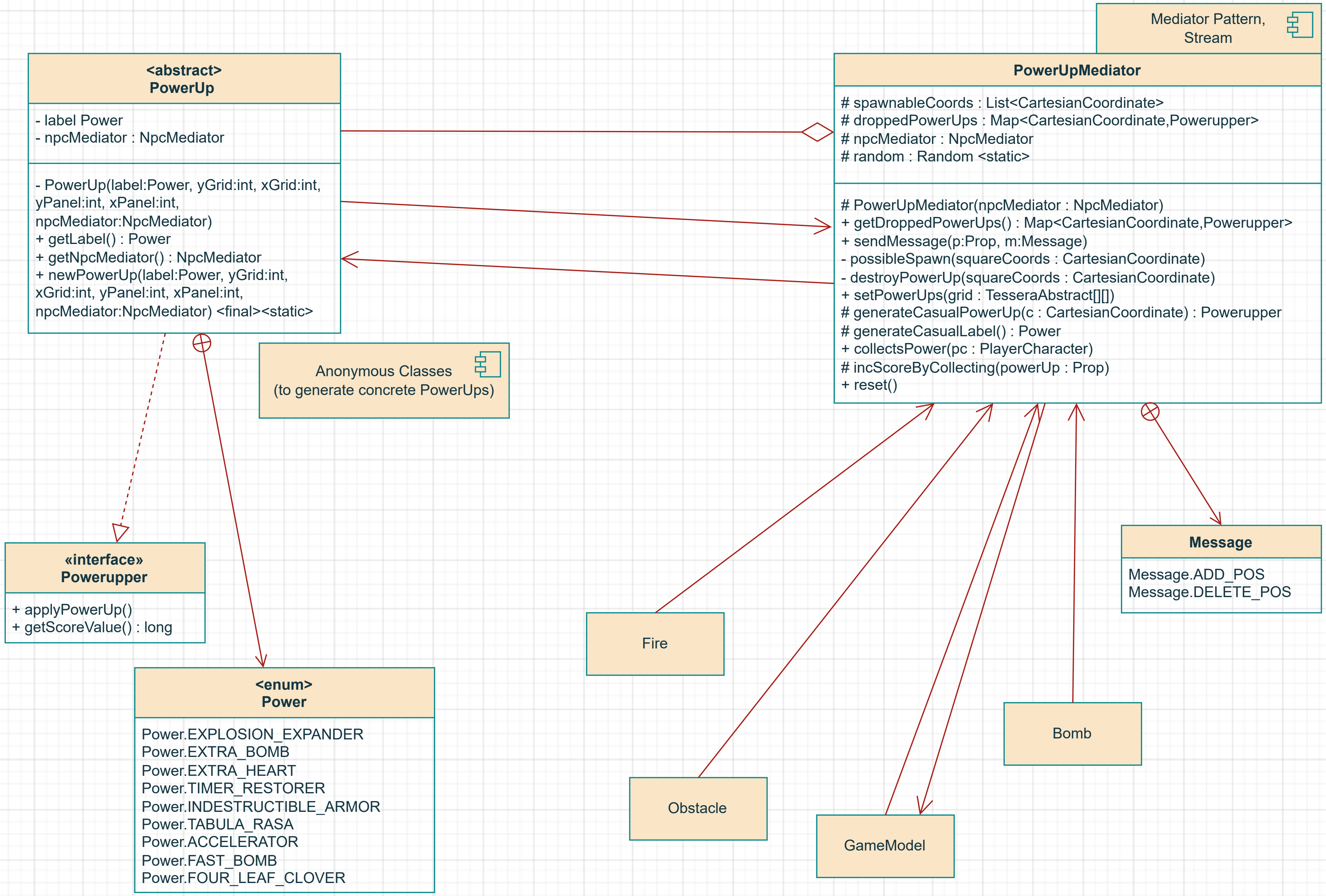
+ EnemyBossClown(yPanel : int, xPanel : int, h : int, w : int, npcMediator : NpcMediator) <constructor>
@ Override # shootStars()
@ Override # generateStar(dirs:List<Dir>)

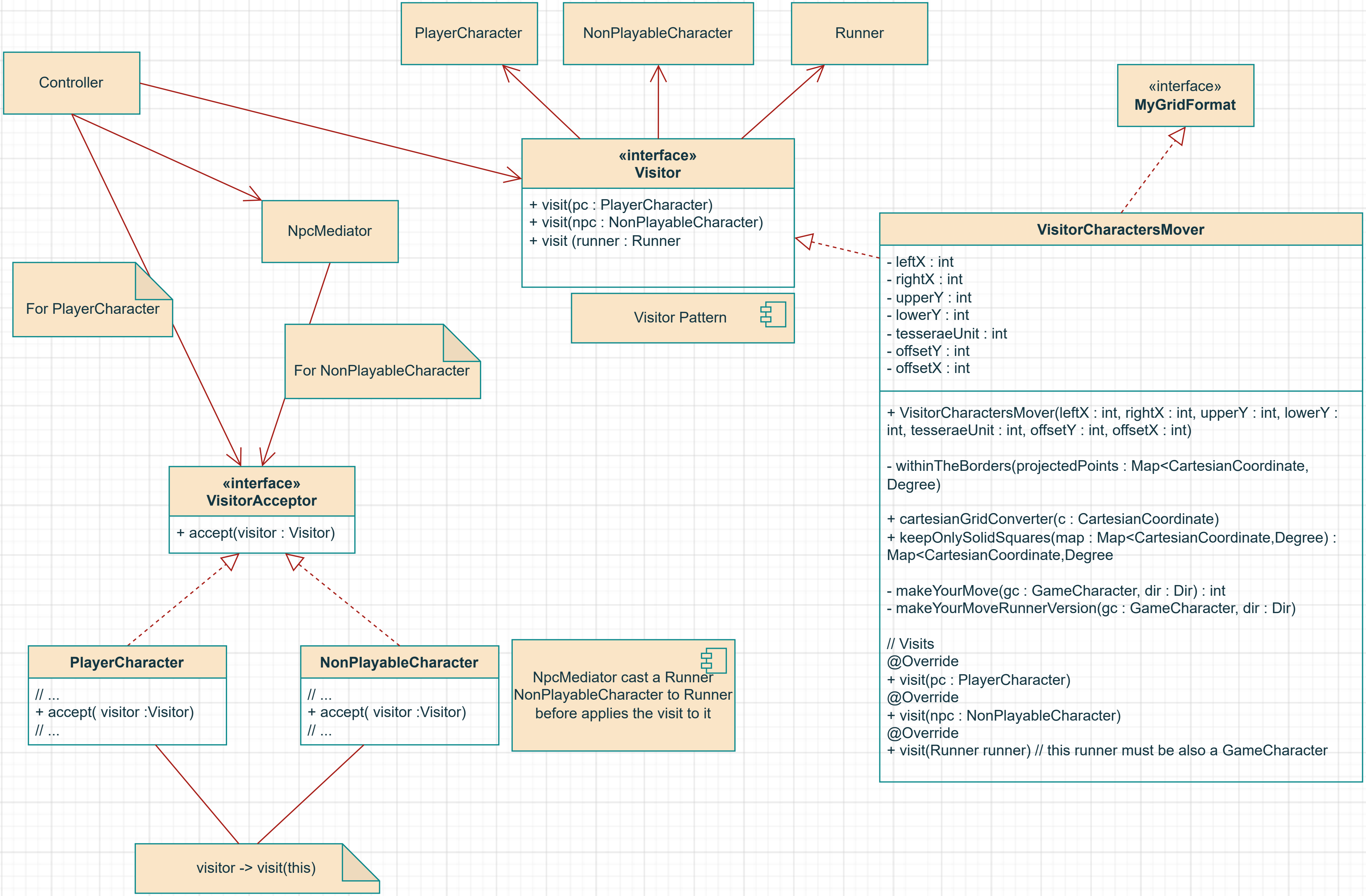


<abstract>
NpcMediator

# npcs : Set<NonPlayableCharacter>
- addedNpcs : List<NonPlayableCharacter>
- bombsEated : Set<Bomb>
- npcVisitor : VisitorCharactersMover
# collidingThePlayer : boolean
- killEveryNpcOneTime : boolean

+ NpcMediator()
+ isCollidingThePlayer() : boolean
+ setCollidingThePlayer(collidingThePlayer:boolean)
+ setVisitor(npcVisitor: VisitorCharactersMover)
+ add(npc:NonPlayableCharacter)
+ addPyro(cGravity:CartesianCoordinate, dirs:List<Dir>, h:int, w:int, speed:int, npcMediator : NpcMediator)
# getPyro(cGravity:CartesianCoordinate, dirs:List<Dir>, h:int, w:int, speed:int, npcMediator:NpcMediator)
+ add(levelSet : Set<NonPlayableCharacter>)
+ reset()
+ load(levelSet : Set<NonPlayableCharacter>)
+ sendPosition(pc : PlayerCharacter)
- checkPCPreciseCollisionWithNPC(pc:PlayerCharacter, npc:NonPlayableCharacter) : boolean
+ oneSecondOfActivityPassed()
- reEstablishBehaviors(pc:PlayerCharacter)
+ walksWithMe(npc:NonPlayableCharacter)
+ getCoerentDir(npc:NonPlayableCharacter) : Dir
- pcHound(npc:NonPlayableCharacter, pc:PlayerCharacter) : List<Dir>
# commitments()
- tryToKillThem(grid:TesseraAbstract[][], vcm:VisitorCharactersMover)
- onLethalSquare(grid:TesseraAbstract[][], npc : NonPlayableCharacter) : boolean
- killOnce(npc : NonPlayableCharacter) : boolean
- isBossOnLethal(grid:TesseraAbstract[][], npc:NonPlayableCharacter, vcm: VisitorCharactersMover) : boolean
+ doKillEveryNpcOneTime()
+ killEveryNpcOneTime()
- squaresOccupiedByBoss(squares : Set<CartesianCoordinate>)
- bossDefeated() <static>
# incScoreGM(npc : NonPlayableCharacter)
+ managementCore(pc:PlayerCharacter, grid:TesseraAbstract[][], vcm:VisitorCharactersMover)
- addedNewNpcs()
+ checkCollisions(pc:PlayerCharacter) : boolean
+ nobodyLeft() : boolean


Mediator Pattern,
Stream
PowerUpMediator
# spawnableCoords : List<CartesianCoordinate>
# droppedPowerUps : Map<CartesianCoordinate,Powerupper>
# npcMediator : NpcMediator
# random : Random <static>
# PowerUpMediator(npcMediator : NpcMediator)
+ getDroppedPowerUps() : Map<CartesianCoordinate,Powerupper>
+ sendMessage(p:Prop, m:Message)
- possibleSpawn(squareCoords : CartesianCoordinate)
- destroyPowerUp(squareCoords : CartesianCoordinate)
+ setPowerUps(grid : TesseraAbstract[][])
# generateCasualPowerUp(c : CartesianCoordinate) : Powerupper
# generateCasualLabel() : Power
+ collectsPower(pc : PlayerCharacter)
# incScoreByCollecting(powerUp : Prop)
+ reset()
<abstract>
PowerUp
- label Power
- npcMediator : NpcMediator
- PowerUp(label:Power, yGrid:int, xGrid:int,
yPanel:int, xPanel:int,
npcMediator:NpcMediator)
+ getLabel() : Power
+ getNpcMediator() : NpcMediator
+ newPowerUp(label:Power, yGrid:int,
xGrid:int, yPanel:int, xPanel:int,
npcMediator:NpcMediator) <final><static>
Anonymous Classes
(to generate concrete PowerUps)
«interface»
Powerupper
+ applyPowerUp()
+ getScoreValue() : long
<enum>
Power
Power.EXPLOSION_EXPANDER
Power.EXTRA_BOMB
Power.EXTRA_HEART
Power.TIMER_RESTORER
Power.INDESTRUCTIBLE_ARMOR
Power.TABULA_RASA
Power.ACCELERATOR
Power.FAST_BOMB
Power.FOUR_LEAF_CLOVER
Fire
Obstacle
GameModel
Bomb
Message
Message.ADD_POS
Message.DELETE_POS



PlayerCharacter

NonPlayableCharacter

Runner

Controller

«interface»
MyGridFormat

«interface»
Visitor

+ visit(pc : PlayerCharacter)
+ visit(npc : NonPlayableCharacter)
+ visit (runner : Runner

Visitor Pattern

NpcMediator

For PlayerCharacter

For NonPlayableCharacter

VisitorCharactersMover

- leftX : int
- rightX : int
- upperY : int
- lowerY : int
- tesseraeUnit : int
- offsetY : int
- offsetX : int

+ VisitorCharactersMover(leftX : int, rightX : int, upperY : int, lowerY : int, tesseraeUnit : int, offsetY : int, offsetX : int)

- withinTheBorders(projectedPoints : Map<CartesianCoordinate, Degree)

+ cartesianGridConverter(c : CartesianCoordinate)
+ keepOnlySolidSquares(map : Map<CartesianCoordinate,Degree) : Map<CartesianCoordinate,Degree

- makeYourMove(gc : GameCharacter, dir : Dir) : int
- makeYourMoveRunnerVersion(gc : GameCharacter, dir : Dir)

// Visits
@Override
+ visit(pc : PlayerCharacter)
@Override
+ visit(npc : NonPlayableCharacter)
@Override
+ visit(Runner runner) // this runner must be also a GameCharacter

«interface»
VisitorAcceptor

+ accept(visitor : Visitor)

PlayerCharacter

// ...
+ accept(visitor :Visitor)
// ...

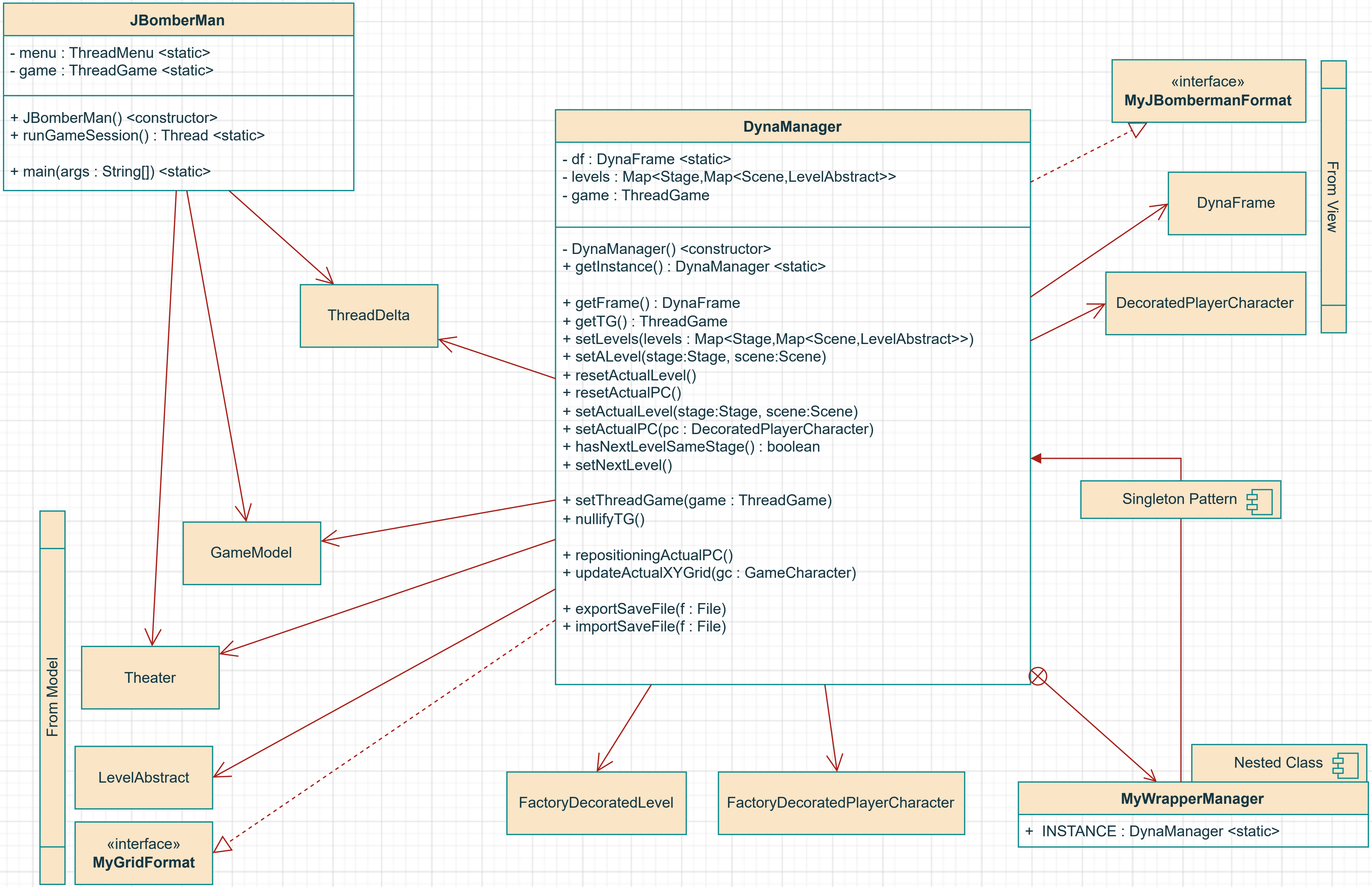
NonPlayableCharacter

// ...
+ accept(visitor :Visitor)
// ...

NpcMediator cast a Runner NonPlayableCharacter to Runner before applies the visit to it

visitor -> visit(this)

Controller UML



JBomberMan

- menu : ThreadMenu <static>
- game : ThreadGame <static>

+ JBomberMan() <constructor>
+ runGameSession() : Thread <static>

+ main(args : String[]) <static>

DynaManager

- df : DynaFrame <static>
- levels : Map<Stage,Map<Scene,LevelAbstract>>
- game : ThreadGame

- DynaManager() <constructor>
+ getInstance() : DynaManager <static>

+ getFrame() : DynaFrame
+ getTG() : ThreadGame
+ setLevels(levels : Map<Stage,Map<Scene,LevelAbstract>>)
+ setALevel(stage:Stage, scene:Scene)
+ resetActualLevel()
+ resetActualPC()
+ setActualLevel(stage:Stage, scene:Scene)
+ setActualPC(pc : DecoratedPlayerCharacter)
+ hasNextLevelSameStage() : boolean
+ setNextLevel()

+ setThreadGame(game : ThreadGame)
+ nullifyTG()

+ repositioningActualPC()
+ updateActualXYGrid(gc : GameCharacter)

+ exportSaveFile(f : File)
+ importSaveFile(f : File)

«interface»
MyJBombermanFormat

DynaFrame

DecoratedPlayerCharacter

From View

ThreadDelta

GameModel

Theater

LevelAbstract

«interface»
MyGridFormat

From Model

FactoryDecoratedLevel

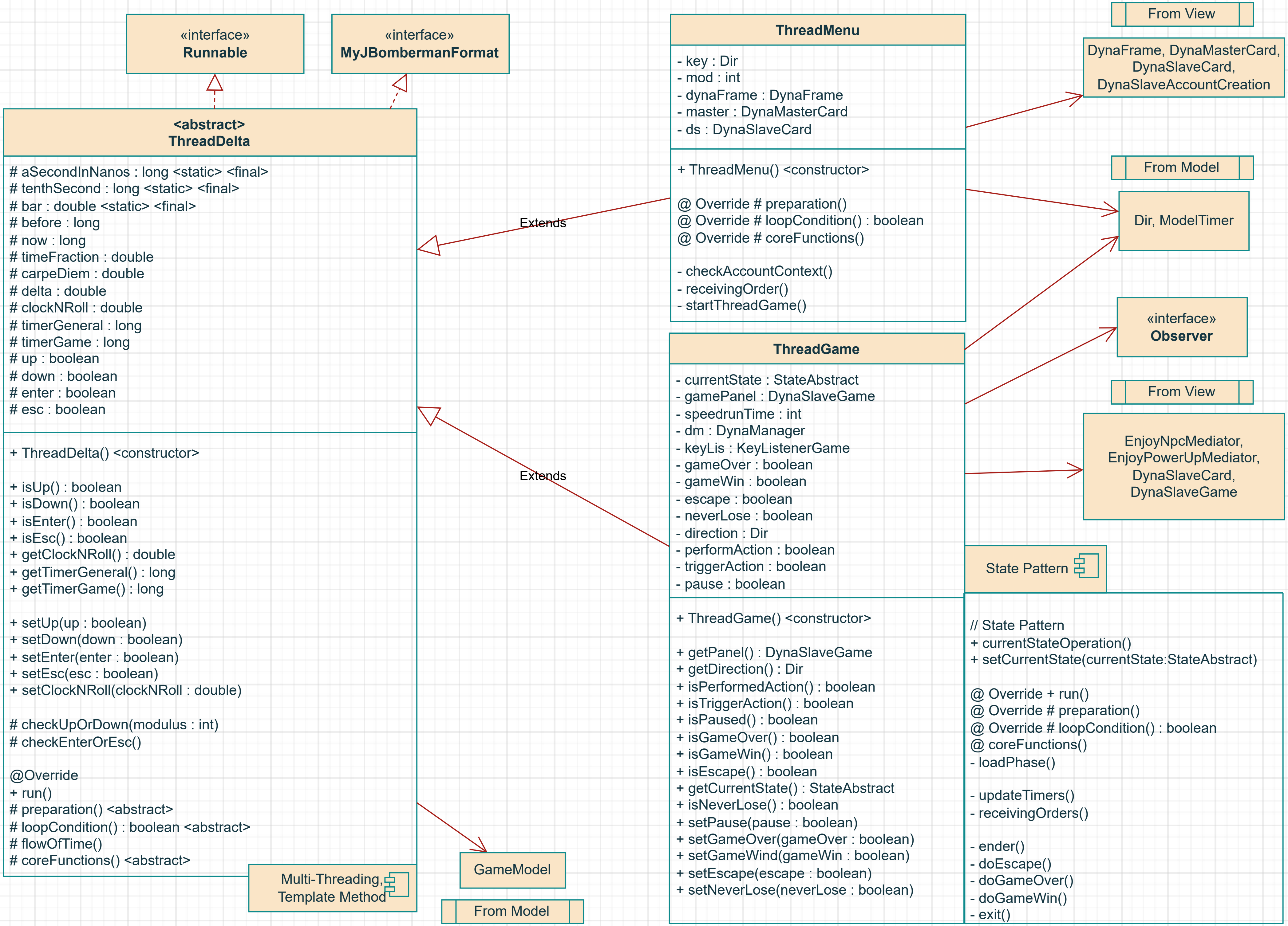
FactoryDecoratedPlayerCharacter

Singleton Pattern

Nested Class

MyWrapperManager

+ INSTANCE : DynaManager <static>



«interface»
Runnable

«interface»
MyJBombermanFormat

<abstract>
ThreadDelta

# aSecondInNanos : long <static> <final>
# tenthSecond : long <static> <final>
# bar : double <static> <final>
# before : long
# now : long
# timeFraction : double
# carpeDiem : double
# delta : double
# clockNRoll : double
# timerGeneral : long
# timerGame : long
# up : boolean
# down : boolean
# enter : boolean
# esc : boolean

+ ThreadDelta() <constructor>

+ isUp() : boolean
+ isDown() : boolean
+ isEnter() : boolean
+ isEsc() : boolean
+ getClockNRoll() : double
+ getTimerGeneral() : long
+ getTimerGame() : long

+ setUp(up : boolean)
+ setDown(down : boolean)
+ setEnter(enter : boolean)
+ setEsc(esc : boolean)
+ setClockNRoll(clockNRoll : double)

# checkUpOrDown(modulus : int)
# checkEnterOrEsc()

@Override
+ run()
# preparation() <abstract>
# loopCondition() : boolean <abstract>
# flowOfTime()
# coreFunctions() <abstract>

Multi-Threading,
Template Method

GameModel

From Model

ThreadMenu

- key : Dir
- mod : int
- dynaFrame : DynaFrame
- master : DynaMasterCard
- ds : DynaSlaveCard

+ ThreadMenu() <constructor>

@ Override # preparation()
@ Override # loopCondition() : boolean
@ Override # coreFunctions()

- checkAccountContext()
- receivingOrder()
- startThreadGame()

ThreadGame

- currentState : StateAbstract
- gamePanel : DynaSlaveGame
- speedrunTime : int
- dm : DynaManager
- keyLis : KeyListenerGame
- gameOver : boolean
- gameWin : boolean
- escape : boolean
- neverLose : boolean
- direction : Dir
- performAction : boolean
- triggerAction : boolean
- pause : boolean

+ ThreadGame() <constructor>

+ getPanel() : DynaSlaveGame
+ getDirection() : Dir
+ isPerformedAction() : boolean
+ isTriggerAction() : boolean
+ isPaused() : boolean
+ isGameOver() : boolean
+ isGameWin() : boolean
+ isEscape() : boolean
+ getCurrentState() : StateAbstract
+ isNeverLose() : boolean
+ setPause(pause : boolean)
+ setGameOver(gameOver : boolean)
+ setGameWind(gameWin : boolean)
+ setEscape(escape : boolean)
+ setNeverLose(neverLose : boolean)

State Pattern

// State Pattern
+ currentStateOperation()
+ setCurrentState(currentState:StateAbstract)

@ Override + run()
@ Override # preparation()
@ Override # loopCondition() : boolean
@ coreFunctions()
- loadPhase()

- updateTimers()
- receivingOrders()

- ender()
- doEscape()
- doGameOver()
- doGameWin()
- exit()

From View

DynaFrame, DynaMasterCard,
DynaSlaveCard,
DynaSlaveAccountCreation

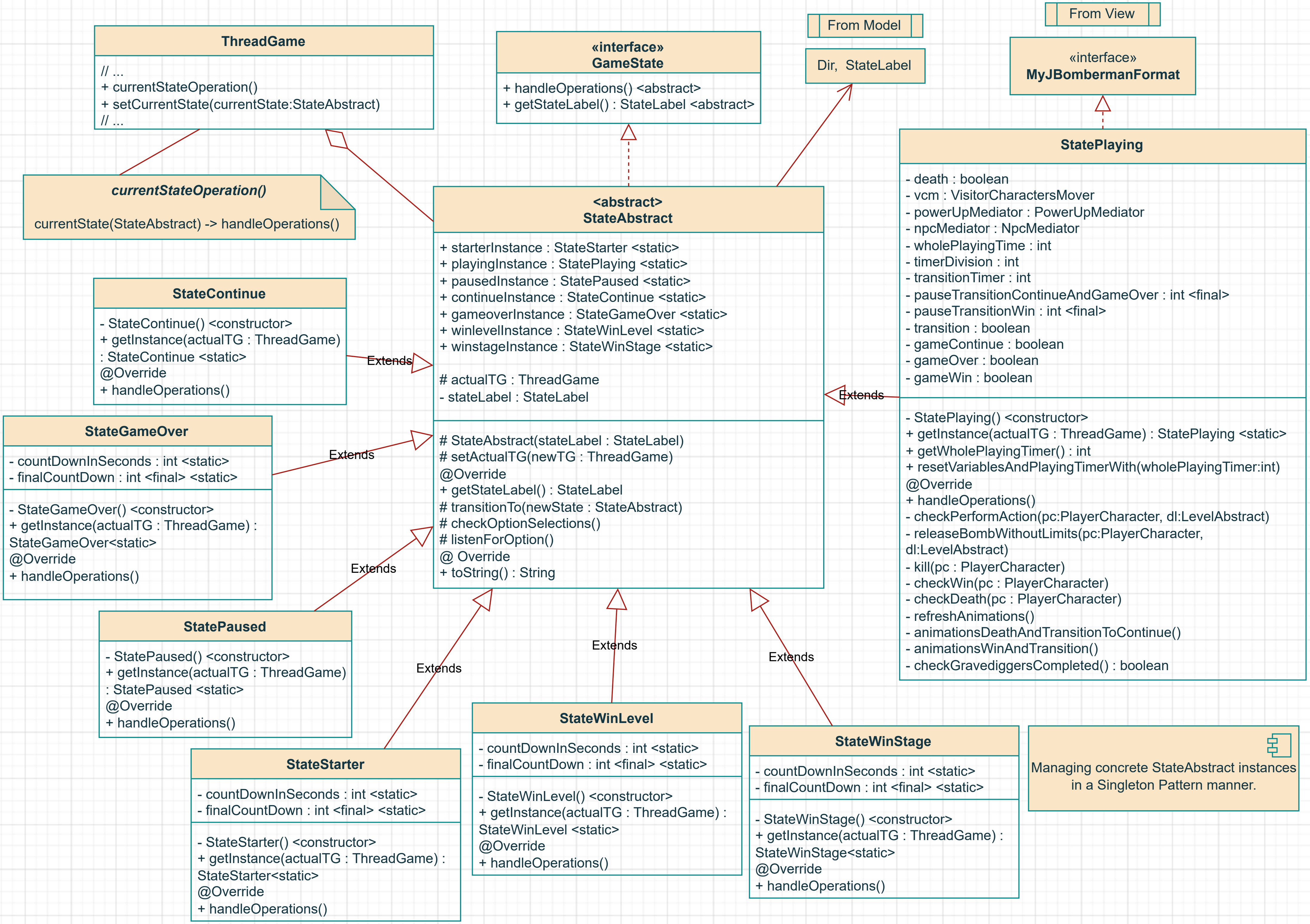
From Model

Dir, ModelTimer

«interface»
Observer

From View

EnjoyNpcMediator,
EnjoyPowerUpMediator,
DynaSlaveCard,
DynaSlaveGame



ThreadGame

// ...
+ currentStateOperation()
+ setCurrentState(currentState:StateAbstract)
// ...

currentStateOperation()

currentState(StateAbstract) -> handleOperations()

«interface»
GameState

+ handleOperations() <abstract>
+ getStateLabel() : StateLabel <abstract>

From Model

Dir, StateLabel

From View

«interface»
MyJBombermanFormat

<abstract>
StateAbstract

+ starterInstance : StateStarter <static>
+ playingInstance : StatePlaying <static>
+ pausedInstance : StatePaused <static>
+ continueInstance : StateContinue <static>
+ gameoverInstance : StateGameOver <static>
+ winlevelInstance : StateWinLevel <static>
+ winstageInstance : StateWinStage <static>

# actualTG : ThreadGame
- stateLabel : StateLabel

# StateAbstract(stateLabel : StateLabel)
# setActualTG(newTG : ThreadGame)
@Override
+ getStateLabel() : StateLabel
# transitionTo(newState : StateAbstract)
# checkOptionSelections()
# listenForOption()
@ Override
+ toString() : String

StatePlaying

- death : boolean
- vcm : VisitorCharactersMover
- powerUpMediator : PowerUpMediator
- npcMediator : NpcMediator
- wholePlayingTime : int
- timerDivision : int
- transitionTimer : int
- pauseTransitionContinueAndGameOver : int <final>
- pauseTransitionWin : int <final>
- transition : boolean
- gameContinue : boolean
- gameOver : boolean
- gameWin : boolean

- StatePlaying() <constructor>
+ getInstance(actualTG : ThreadGame) : StatePlaying <static>
+ getWholePlayingTimer() : int
+ resetVariablesAndPlayingTimerWith(wholePlayingTimer:int)
@Override
+ handleOperations()
- checkPerformAction(pc:PlayerCharacter, dl:LevelAbstract)
- releaseBombWithoutLimits(pc:PlayerCharacter, dl:LevelAbstract)
- kill(pc : PlayerCharacter)
- checkWin(pc : PlayerCharacter)
- checkDeath(pc : PlayerCharacter)
- refreshAnimations()
- animationsDeathAndTransitionToContinue()
- animationsWinAndTransition()
- checkGravediggersCompleted() : boolean

StateContinue

- StateContinue() <constructor>
+ getInstance(actualTG : ThreadGame) : StateContinue <static>
@Override
+ handleOperations()

StateGameOver

- countDownInSeconds : int <static>
- finalCountDown : int <final> <static>

- StateGameOver() <constructor>
+ getInstance(actualTG : ThreadGame) : StateGameOver<static>
@Override
+ handleOperations()

StatePaused

- StatePaused() <constructor>
+ getInstance(actualTG : ThreadGame) : StatePaused <static>
@Override
+ handleOperations()

StateStarter

- countDownInSeconds : int <static>
- finalCountDown : int <final> <static>

- StateStarter() <constructor>
+ getInstance(actualTG : ThreadGame) : StateStarter<static>
@Override
+ handleOperations()

StateWinLevel

- countDownInSeconds : int <static>
- finalCountDown : int <final> <static>

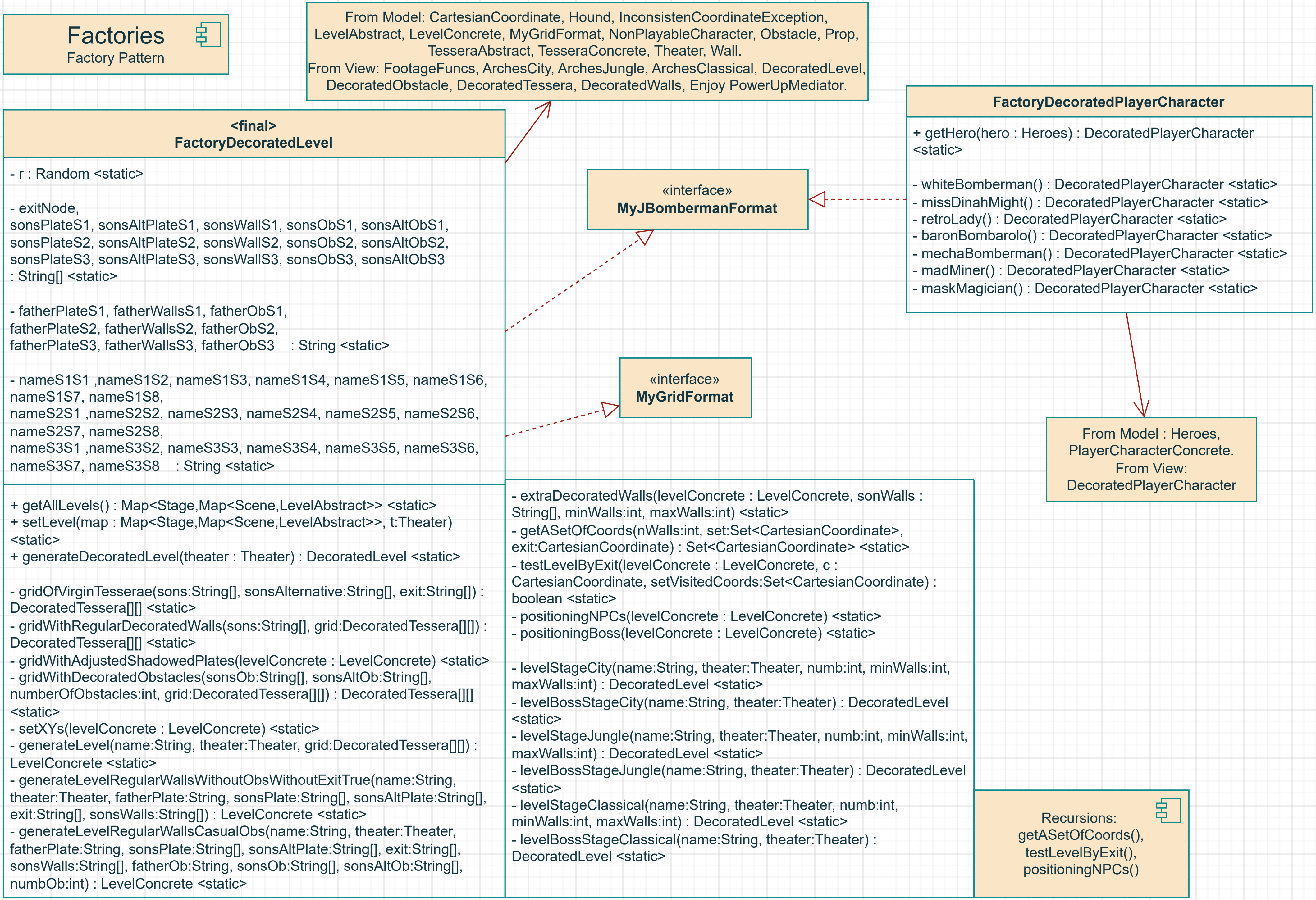
- StateWinLevel() <constructor>
+ getInstance(actualTG : ThreadGame) : StateWinLevel <static>
@Override
+ handleOperations()

StateWinStage

- countDownInSeconds : int <static>
- finalCountDown : int <final> <static>

- StateWinStage() <constructor>
+ getInstance(actualTG : ThreadGame) : StateWinStage<static>
@Override
+ handleOperations()

Managing concrete StateAbstract instances in a Singleton Pattern manner.



Factories

Factory Pattern

From Model: CartesianCoordinate, Hound, InconsistenCoordinateException, LevelAbstract, LevelConcrete, MyGridFormat, NonPlayableCharacter, Obstacle, Prop, TesseraAbstract, TesseraConcrete, Theater, Wall.
From View: FootageFuncs, ArchesCity, ArchesJungle, ArchesClassical, DecoratedLevel, DecoratedObstacle, DecoratedTessera, DecoratedWalls, Enjoy PowerUpMediator.

<final>
FactoryDecoratedLevel

- r : Random <static>

- exitNode,
sonsPlateS1, sonsAltPlateS1, sonsWallS1, sonsObS1, sonsAltObS1,
sonsPlateS2, sonsAltPlateS2, sonsWallS2, sonsObS2, sonsAltObS2,
sonsPlateS3, sonsAltPlateS3, sonsWallS3, sonsObS3, sonsAltObS3
: String[] <static>

- fatherPlateS1, fatherWallsS1, fatherObS1,
fatherPlateS2, fatherWallsS2, fatherObS2,
fatherPlateS3, fatherWallsS3, fatherObS3 : String <static>

- nameS1S1 ,nameS1S2, nameS1S3, nameS1S4, nameS1S5, nameS1S6, nameS1S7, nameS1S8,
nameS2S1 ,nameS2S2, nameS2S3, nameS2S4, nameS2S5, nameS2S6, nameS2S7, nameS2S8,
nameS3S1 ,nameS3S2, nameS3S3, nameS3S4, nameS3S5, nameS3S6, nameS3S7, nameS3S8 : String <static>

+ getAllLevels() : Map<Stage,Map<Scene,LevelAbstract>> <static>
+ setLevel(map : Map<Stage,Map<Scene,LevelAbstract>>, t:Theater) <static>
+ generateDecoratedLevel(theater : Theater) : DecoratedLevel <static>

- gridOfVirginTesserae(sons:String[], sonsAlternative:String[], exit:String[]) : DecoratedTessera[][] <static>
- gridWithRegularDecoratedWalls(sons:String[], grid:DecoratedTessera[][]) : DecoratedTessera[][] <static>
- gridWithAdjustedShadowedPlates(levelConcrete : LevelConcrete) <static>
- gridWithDecoratedObstacles(sonsOb:String[], sonsAltOb:String[], numberOfObstacles:int, grid:DecoratedTessera[][]) : DecoratedTessera[][] <static>
- setXYs(levelConcrete : LevelConcrete) <static>
- generateLevel(name:String, theater:Theater, grid:DecoratedTessera[][]) : LevelConcrete <static>
- generateLevelRegularWallsWithoutObsWithoutExitTrue(name:String, theater:Theater, fatherPlate:String, sonsPlate:String[], sonsAltPlate:String[], exit:String[], sonsWalls:String[]) : LevelConcrete <static>
- generateLevelRegularWallsCasualObs(name:String, theater:Theater, fatherPlate:String, sonsPlate:String[], sonsAltPlate:String[], exit:String[], sonsWalls:String[], fatherOb:String, sonsOb:String[], sonsAltOb:String[], numbOb:int) : LevelConcrete <static>

- extraDecoratedWalls(levelConcrete : LevelConcrete, sonWalls : String[], minWalls:int, maxWalls:int) <static>
- getASetOfCoords(nWalls:int, set:Set<CartesianCoordinate>, exit:CartesianCoordinate) : Set<CartesianCoordinate> <static>
- testLevelByExit(levelConcrete : LevelConcrete, c : CartesianCoordinate, setVisitedCoords:Set<CartesianCoordinate) : boolean <static>
- positioningNPCs(levelConcrete : LevelConcrete) <static>
- positioningBoss(levelConcrete : LevelConcrete) <static>

- levelStageCity(name:String, theater:Theater, numb:int, minWalls:int, maxWalls:int) : DecoratedLevel <static>
- levelBossStageCity(name:String, theater:Theater) : DecoratedLevel <static>
- levelStageJungle(name:String, theater:Theater, numb:int, minWalls:int, maxWalls:int) : DecoratedLevel <static>
- levelBossStageJungle(name:String, theater:Theater) : DecoratedLevel <static>
- levelStageClassical(name:String, theater:Theater, numb:int, minWalls:int, maxWalls:int) : DecoratedLevel <static>
- levelBossStageClassical(name:String, theater:Theater) : DecoratedLevel <static>

«interface»
MyJBombermanFormat

«interface»
MyGridFormat

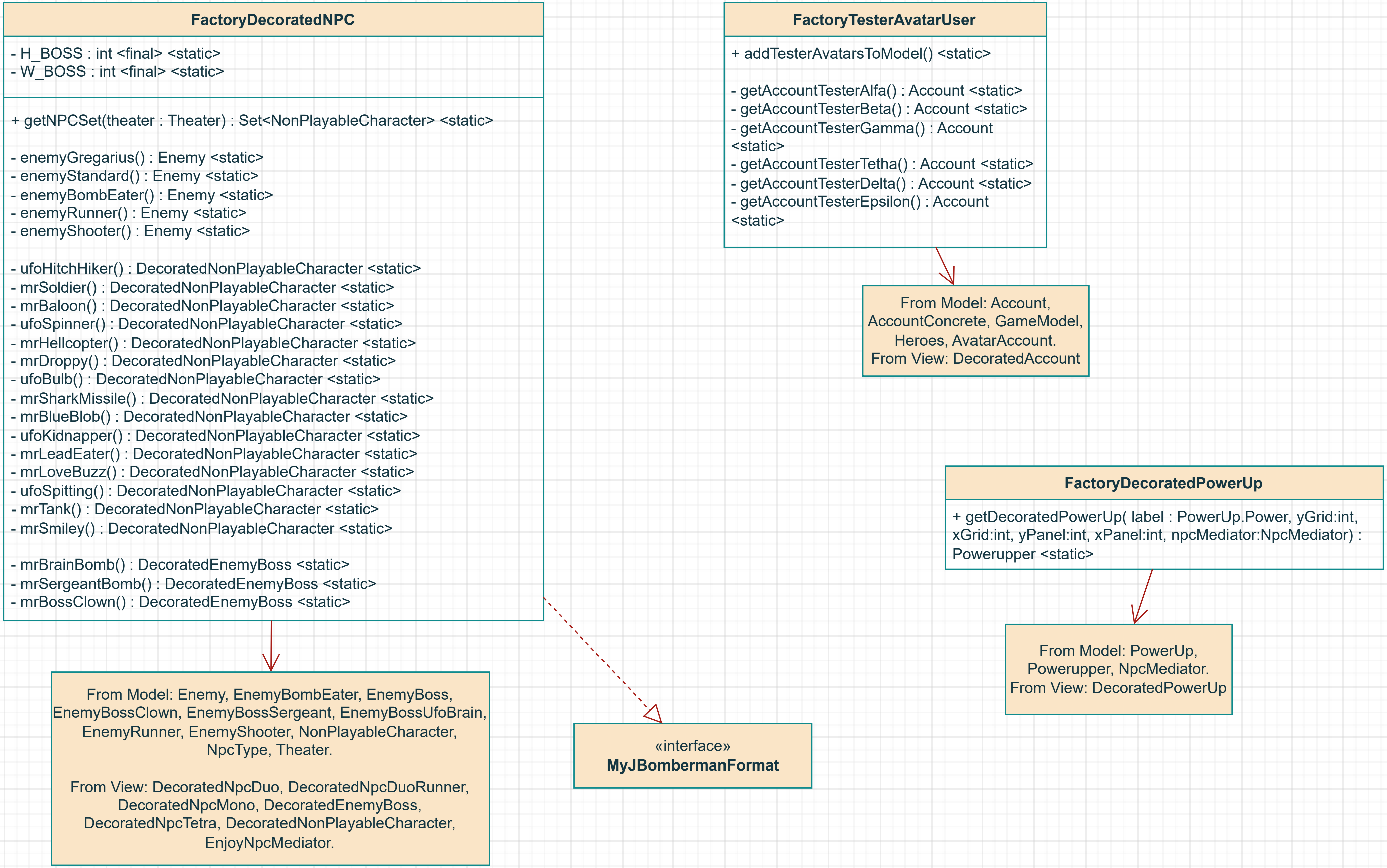
FactoryDecoratedPlayerCharacter

+ getHero(hero : Heroes) : DecoratedPlayerCharacter <static>

- whiteBomberman() : DecoratedPlayerCharacter <static>
- missDinahMight() : DecoratedPlayerCharacter <static>
- retroLady() : DecoratedPlayerCharacter <static>
- baronBombarolo() : DecoratedPlayerCharacter <static>
- mechaBomberman() : DecoratedPlayerCharacter <static>
- madMiner() : DecoratedPlayerCharacter <static>
- maskMagician() : DecoratedPlayerCharacter <static>

From Model : Heroes,
PlayerCharacterConcrete.
From View:
DecoratedPlayerCharacter

Recursions:
getASetOfCoords(),
testLevelByExit(),
positioningNPCs()



FactoryDecoratedNPC

- H_BOSS : int <final> <static>
- W_BOSS : int <final> <static>

+ getNPCSet(theater : Theater) : Set<NonPlayableCharacter> <static>

- enemyGregarius() : Enemy <static>
- enemyStandard() : Enemy <static>
- enemyBombEater() : Enemy <static>
- enemyRunner() : Enemy <static>
- enemyShooter() : Enemy <static>

- ufoHitchHiker() : DecoratedNonPlayableCharacter <static>
- mrSoldier() : DecoratedNonPlayableCharacter <static>
- mrBaloon() : DecoratedNonPlayableCharacter <static>
- ufoSpinner() : DecoratedNonPlayableCharacter <static>
- mrHellcopter() : DecoratedNonPlayableCharacter <static>
- mrDroppy() : DecoratedNonPlayableCharacter <static>
- ufoBulb() : DecoratedNonPlayableCharacter <static>
- mrSharkMissile() : DecoratedNonPlayableCharacter <static>
- mrBlueBlob() : DecoratedNonPlayableCharacter <static>
- ufoKidnapper() : DecoratedNonPlayableCharacter <static>
- mrLeadEater() : DecoratedNonPlayableCharacter <static>
- mrLoveBuzz() : DecoratedNonPlayableCharacter <static>
- ufoSpitting() : DecoratedNonPlayableCharacter <static>
- mrTank() : DecoratedNonPlayableCharacter <static>
- mrSmiley() : DecoratedNonPlayableCharacter <static>

- mrBrainBomb() : DecoratedEnemyBoss <static>
- mrSergeantBomb() : DecoratedEnemyBoss <static>
- mrBossClown() : DecoratedEnemyBoss <static>

From Model: Enemy, EnemyBombEater, EnemyBoss, EnemyBossClown, EnemyBossSergeant, EnemyBossUfoBrain, EnemyRunner, EnemyShooter, NonPlayableCharacter, NpcType, Theater.

From View: DecoratedNpcDuo, DecoratedNpcDuoRunner, DecoratedNpcMono, DecoratedEnemyBoss, DecoratedNpcTetra, DecoratedNonPlayableCharacter, EnjoyNpcMediator.

«interface»
MyJBombermanFormat

FactoryTesterAvatarUser

+ addTesterAvatarsToModel() <static>

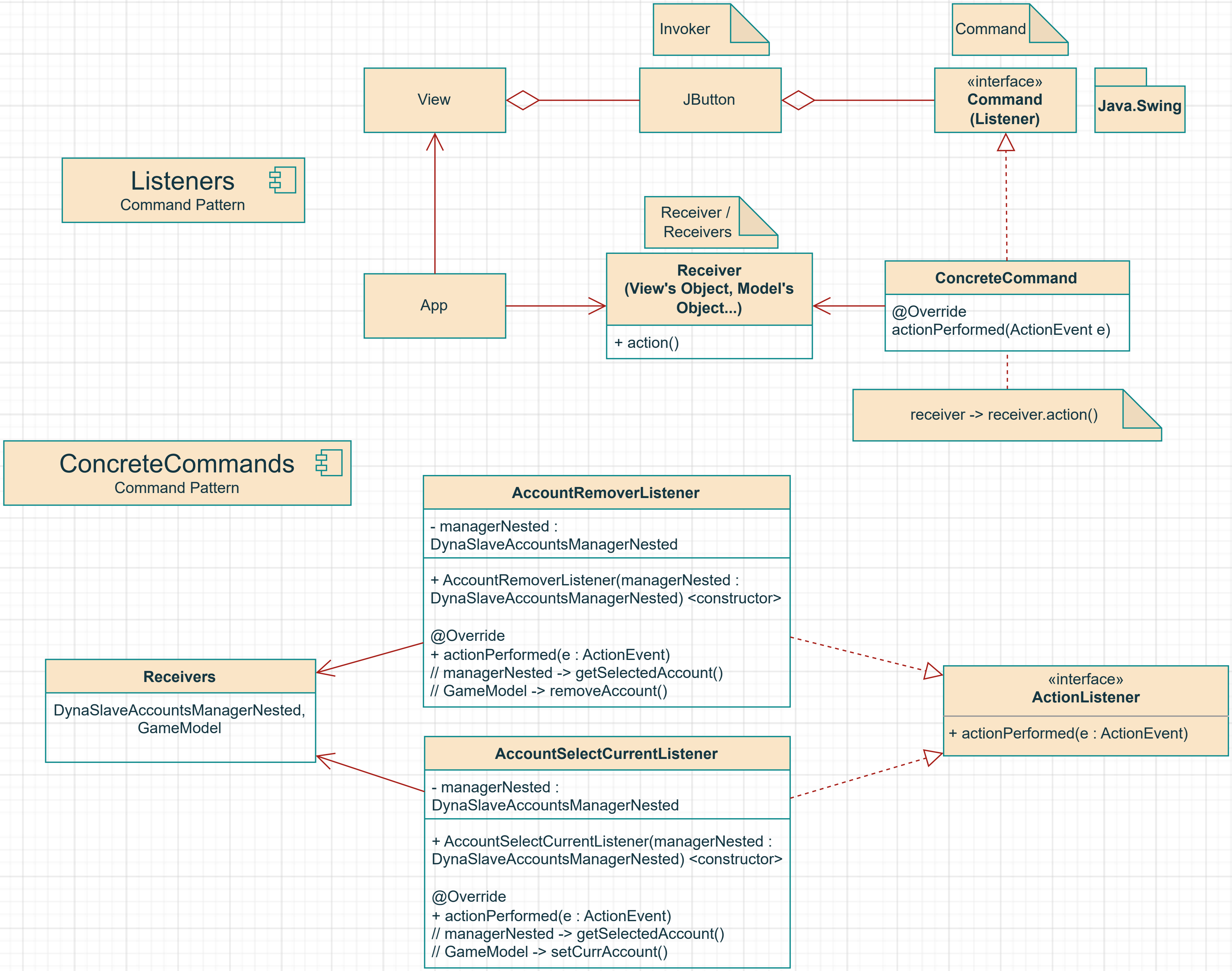
- getAccountTesterAlfa() : Account <static>
- getAccountTesterBeta() : Account <static>
- getAccountTesterGamma() : Account <static>
- getAccountTesterTetha() : Account <static>
- getAccountTesterDelta() : Account <static>
- getAccountTesterEpsilon() : Account <static>

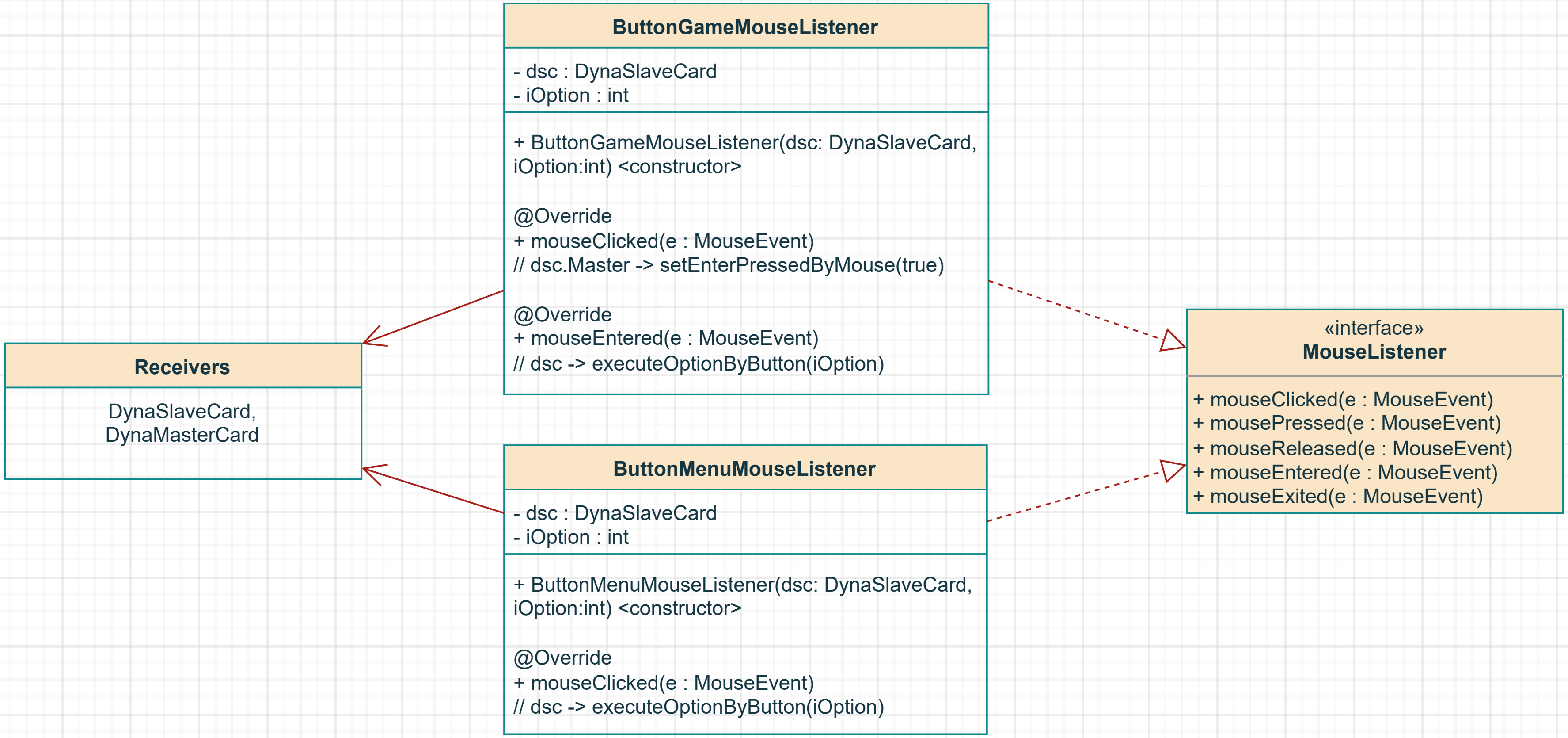
From Model: Account, AccountConcrete, GameModel, Heroes, AvatarAccount.
From View: DecoratedAccount

FactoryDecoratedPowerUp

+ getDecoratedPowerUp(label : PowerUp.Power, yGrid:int, xGrid:int, yPanel:int, xPanel:int, npcMediator:NpcMediator) : Powerupper <static>

From Model: PowerUp, Powerupper, NpcMediator.
From View: DecoratedPowerUp


Invoker
Command
View
JButton
«interface»
Command
(Listener)
Java.Swing
Listeners
Command Pattern
Receiver /
Receivers
App
Receiver
(View's Object, Model's
Object...)
+ action()
ConcreteCommand
@Override
actionPerformed(ActionEvent e)
receiver -> receiver.action()
ConcreteCommands
Command Pattern
AccountRemoverListener
- managerNested :
DynaSlaveAccountsManagerNested
+ AccountRemoverListener(managerNested :
DynaSlaveAccountsManagerNested) <constructor>
@Override
+ actionPerformed(e : ActionEvent)
// managerNested -> getSelectedAccount()
// GameModel -> removeAccount()
Receivers
DynaSlaveAccountsManagerNested,
GameModel
«interface»
ActionListener
+ actionPerformed(e : ActionEvent)
AccountSelectCurrentListener
- managerNested :
DynaSlaveAccountsManagerNested
+ AccountSelectCurrentListener(managerNested :
DynaSlaveAccountsManagerNested) <constructor>
@Override
+ actionPerformed(e : ActionEvent)
// managerNested -> getSelectedAccount()
// GameModel -> setCurrAccount()



ButtonGameMouseListener

- dsc : DynaSlaveCard
- iOption : int

+ ButtonGameMouseListener(dsc: DynaSlaveCard, iOption:int) <constructor>

@Override
+ mouseClicked(e : MouseEvent)
// dsc.Master -> setEnterPressedByMouse(true)

@Override
+ mouseEntered(e : MouseEvent)
// dsc -> executeOptionByButton(iOption)

Receivers

DynaSlaveCard,
DynaMasterCard

ButtonMenuMouseListener

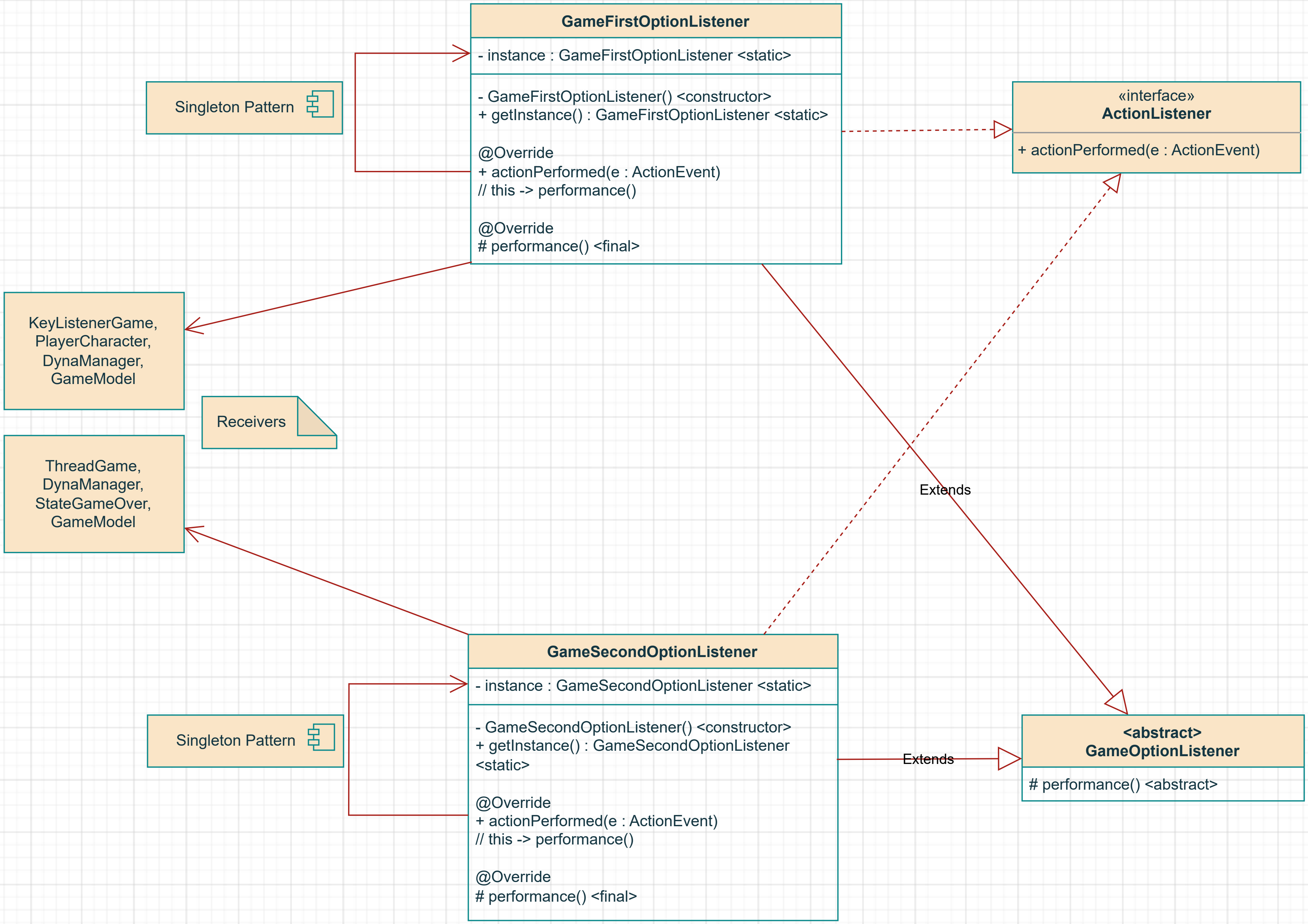
- dsc : DynaSlaveCard
- iOption : int

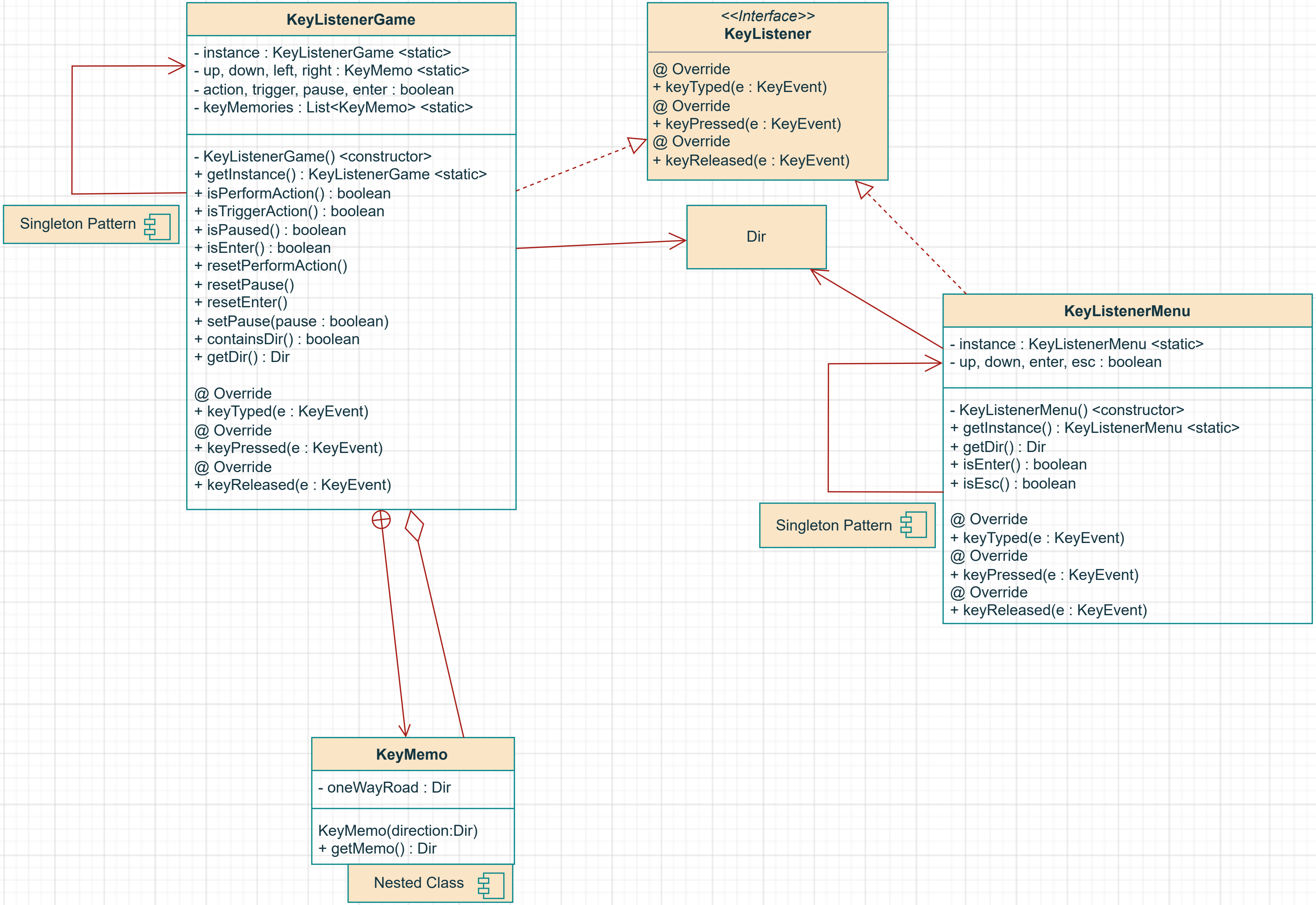
+ ButtonMenuMouseListener(dsc: DynaSlaveCard, iOption:int) <constructor>

@Override
+ mouseClicked(e : MouseEvent)
// dsc -> executeOptionByButton(iOption)

«interface»
MouseListener

+ mouseClicked(e : MouseEvent)
+ mousePressed(e : MouseEvent)
+ mouseReleased(e : MouseEvent)
+ mouseEntered(e : MouseEvent)
+ mouseExited(e : MouseEvent)


GameFirstOptionListener
- instance : GameFirstOptionListener <static>
- GameFirstOptionListener() <constructor>
+ getInstance() : GameFirstOptionListener <static>
@Override
+ actionPerformed(e : ActionEvent)
// this -> performance()
@Override
# performance() <final>
Singleton Pattern
«interface»
ActionListener
+ actionPerformed(e : ActionEvent)
KeyListenerGame,
PlayerCharacter,
DynaManager,
GameModel
Receivers
ThreadGame,
DynaManager,
StateGameOver,
GameModel
Extends
GameSecondOptionListener
- instance : GameSecondOptionListener <static>
- GameSecondOptionListener() <constructor>
+ getInstance() : GameSecondOptionListener
<static>
@Override
+ actionPerformed(e : ActionEvent)
// this -> performance()
@Override
# performance() <final>
Singleton Pattern
Extends
<abstract>
GameOptionListener
# performance() <abstract>



KeyListenerGame

- instance : KeyListenerGame <static>
- up, down, left, right : KeyMemo <static>
- action, trigger, pause, enter : boolean
- keyMemories : List<KeyMemo> <static>

- KeyListenerGame() <constructor>
+ getInstance() : KeyListenerGame <static>
+ isPerformAction() : boolean
+ isTriggerAction() : boolean
+ isPaused() : boolean
+ isEnter() : boolean
+ resetPerformAction()
+ resetPause()
+ resetEnter()
+ setPause(pause : boolean)
+ containsDir() : boolean
+ getDir() : Dir

@ Override
+ keyTyped(e : KeyEvent)
@ Override
+ keyPressed(e : KeyEvent)
@ Override
+ keyReleased(e : KeyEvent)

<<Interface>>
KeyListener

@ Override
+ keyTyped(e : KeyEvent)
@ Override
+ keyPressed(e : KeyEvent)
@ Override
+ keyReleased(e : KeyEvent)

KeyListenerMenu

- instance : KeyListenerMenu <static>
- up, down, enter, esc : boolean

- KeyListenerMenu() <constructor>
+ getInstance() : KeyListenerMenu <static>
+ getDir() : Dir
+ isEnter() : boolean
+ isEsc() : boolean

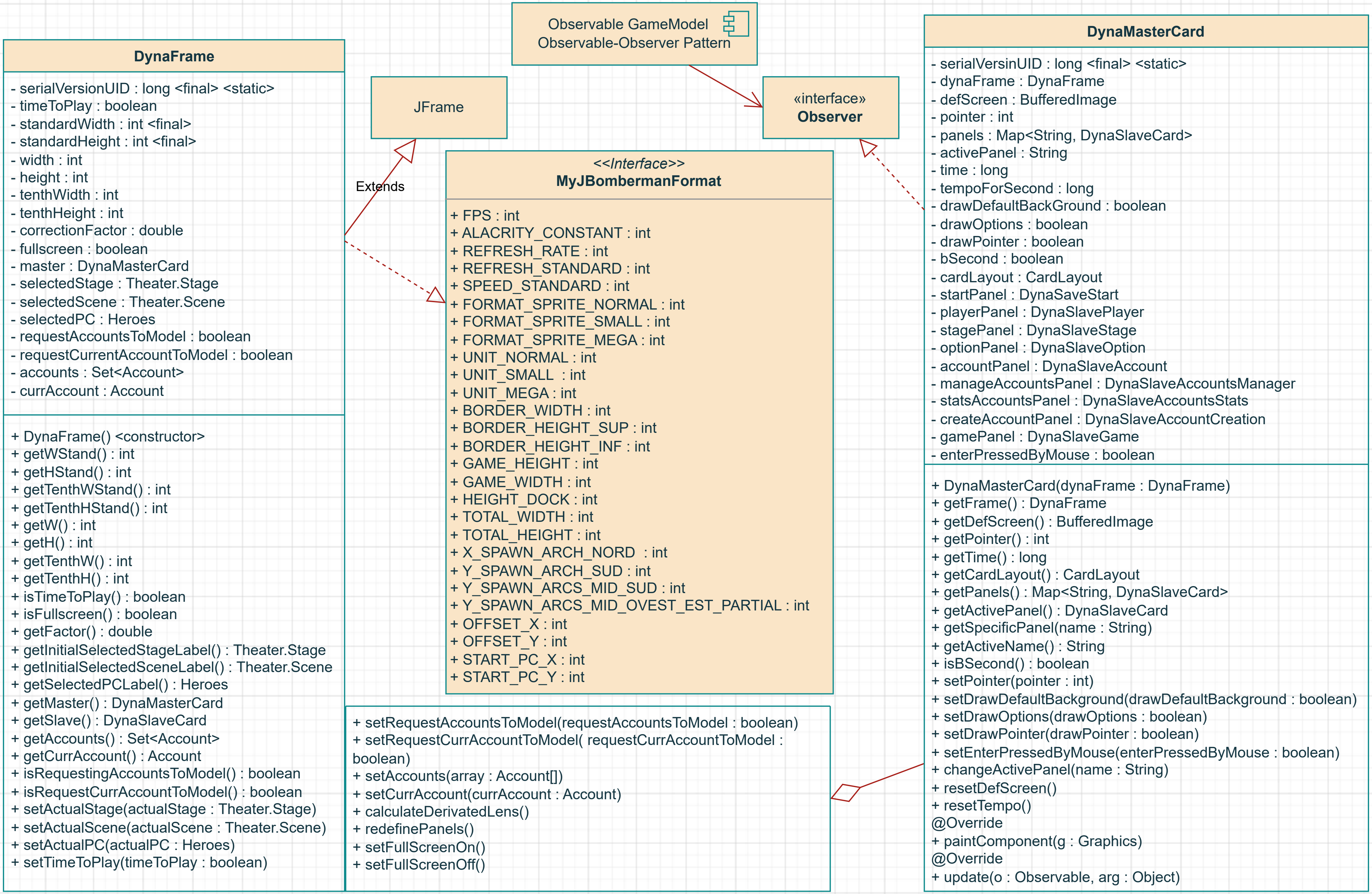
@ Override
+ keyTyped(e : KeyEvent)
@ Override
+ keyPressed(e : KeyEvent)
@ Override
+ keyReleased(e : KeyEvent)

KeyMemo

- oneWayRoad : Dir

KeyMemo(direction:Dir)
+ getMemo() : Dir

View UML: JFrame, JPanel, Layout



DynaFrame

- serialVersionUID : long <final> <static>
- timeToPlay : boolean
- standardWidth : int <final>
- standardHeight : int <final>
- width : int
- height : int
- tenthWidth : int
- tenthHeight : int
- correctionFactor : double
- fullscreen : boolean
- master : DynaMasterCard
- selectedStage : Theater.Stage
- selectedScene : Theater.Scene
- selectedPC : Heroes
- requestAccountsToModel : boolean
- requestCurrentAccountToModel : boolean
- accounts : Set<Account>
- currAccount : Account

+ DynaFrame() <constructor>
+ getWStand() : int
+ getHStand() : int
+ getTenthWStand() : int
+ getTenthHStand() : int
+ getW() : int
+ getH() : int
+ getTenthW() : int
+ getTenthH() : int
+ isTimeToPlay() : boolean
+ isFullscreen() : boolean
+ getFactor() : double
+ getInitialSelectedStageLabel() : Theater.Stage
+ getInitialSelectedSceneLabel() : Theater.Scene
+ getSelectedPCLabel() : Heroes
+ getMaster() : DynaMasterCard
+ getSlave() : DynaSlaveCard
+ getAccounts() : Set<Account>
+ getCurrAccount() : Account
+ isRequestingAccountsToModel() : boolean
+ isRequestCurrAccountToModel() : boolean
+ setActualStage(actualStage : Theater.Stage)
+ setActualScene(actualScene : Theater.Scene)
+ setActualPC(actualPC : Heroes)
+ setTimeToPlay(timeToPlay : boolean)

+ setRequestAccountsToModel(requestAccountsToModel : boolean)
+ setRequestCurrAccountToModel(requestCurrAccountToModel : boolean)
+ setAccounts(array : Account[])
+ setCurrAccount(currAccount : Account)
+ calculateDerivatedLens()
+ redefinePanels()
+ setFullScreenOn()
+ setFullScreenOff()

JFrame

Extends

Observable GameModel
Observable-Observer Pattern

«interface»
Observer

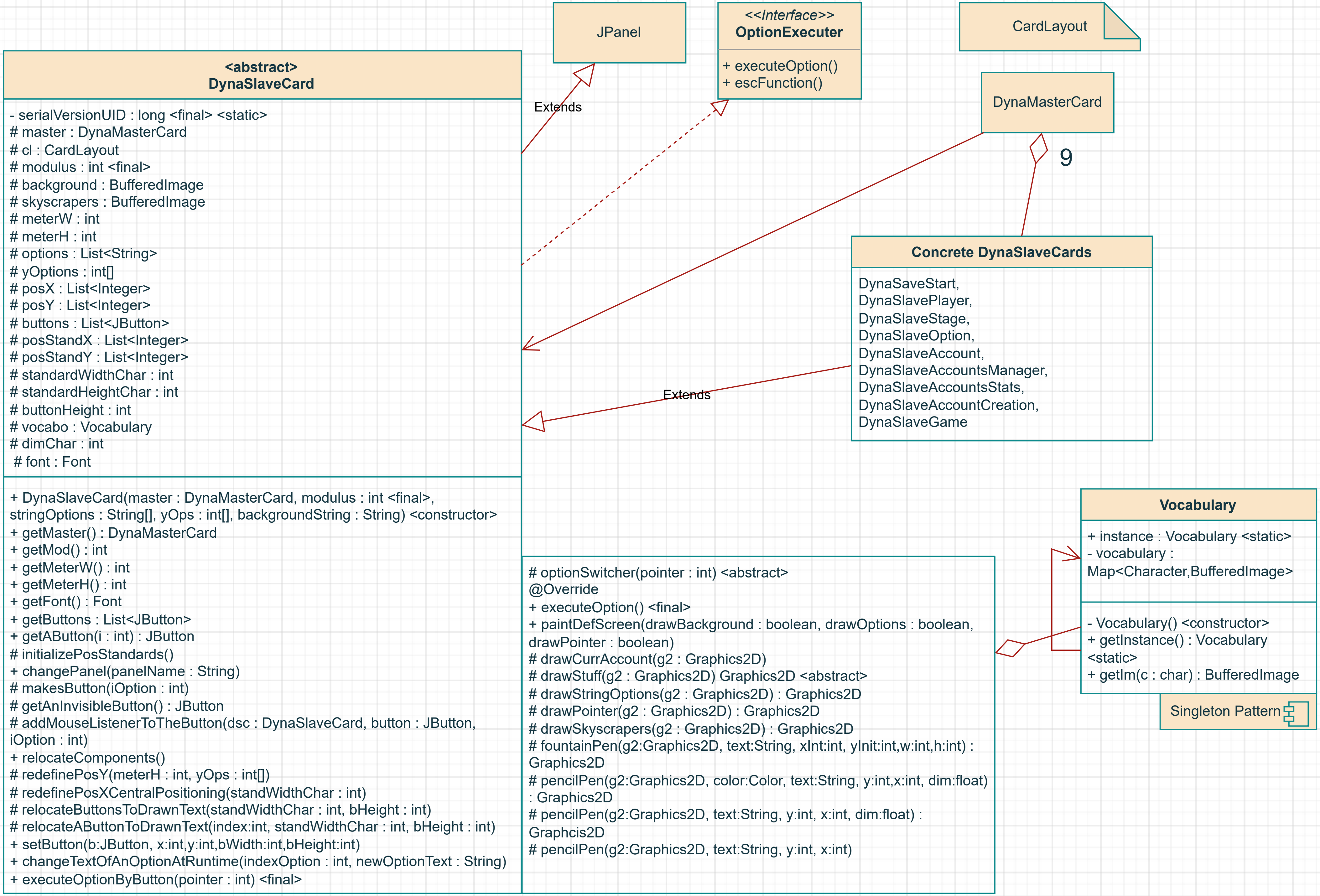
<<Interface>>
MyJBombermanFormat

+ FPS : int
+ ALACRITY_CONSTANT : int
+ REFRESH_RATE : int
+ REFRESH_STANDARD : int
+ SPEED_STANDARD : int
+ FORMAT_SPRITE_NORMAL : int
+ FORMAT_SPRITE_SMALL : int
+ FORMAT_SPRITE_MEGA : int
+ UNIT_NORMAL : int
+ UNIT_SMALL : int
+ UNIT_MEGA : int
+ BORDER_WIDTH : int
+ BORDER_HEIGHT_SUP : int
+ BORDER_HEIGHT_INF : int
+ GAME_HEIGHT : int
+ GAME_WIDTH : int
+ HEIGHT_DOCK : int
+ TOTAL_WIDTH : int
+ TOTAL_HEIGHT : int
+ X_SPAWN_ARCH_NORD : int
+ Y_SPAWN_ARCH_SUD : int
+ Y_SPAWN_ARCS_MID_SUD : int
+ Y_SPAWN_ARCS_MID_OVEST_EST_PARTIAL : int
+ OFFSET_X : int
+ OFFSET_Y : int
+ START_PC_X : int
+ START_PC_Y : int

DynaMasterCard

- serialVersinUID : long <final> <static>
- dynaFrame : DynaFrame
- defScreen : BufferedImage
- pointer : int
- panels : Map<String, DynaSlaveCard>
- activePanel : String
- time : long
- tempoForSecond : long
- drawDefaultBackGround : boolean
- drawOptions : boolean
- drawPointer : boolean
- bSecond : boolean
- cardLayout : CardLayout
- startPanel : DynaSaveStart
- playerPanel : DynaSlavePlayer
- stagePanel : DynaSlaveStage
- optionPanel : DynaSlaveOption
- accountPanel : DynaSlaveAccount
- manageAccountsPanel : DynaSlaveAccountsManager
- statsAccountsPanel : DynaSlaveAccountsStats
- createAccountPanel : DynaSlaveAccountCreation
- gamePanel : DynaSlaveGame
- enterPressedByMouse : boolean

+ DynaMasterCard(dynaFrame : DynaFrame)
+ getFrame() : DynaFrame
+ getDefScreen() : BufferedImage
+ getPointer() : int
+ getTime() : long
+ getCardLayout() : CardLayout
+ getPanels() : Map<String, DynaSlaveCard>
+ getActivePanel() : DynaSlaveCard
+ getSpecificPanel(name : String)
+ getActiveName() : String
+ isBSecond() : boolean
+ setPointer(pointer : int)
+ setDrawDefaultBackground(drawDefaultBackground : boolean)
+ setDrawOptions(drawOptions : boolean)
+ setDrawPointer(drawPointer : boolean)
+ setEnterPressedByMouse(enterPressedByMouse : boolean)
+ changeActivePanel(name : String)
+ resetDefScreen()
+ resetTempo()
@Override
+ paintComponent(g : Graphics)
@Override
+ update(o : Observable, arg : Object)



```
<abstract>
DynaSlaveCard

- serialVersionUID : long <final> <static>
# master : DynaMasterCard
# cl : CardLayout
# modulus : int <final>
# background : BufferedImage
# skyscrapers : BufferedImage
# meterW : int
# meterH : int
# options : List<String>
# yOptions : int[]
# posX : List<Integer>
# posY : List<Integer>
# buttons : List<JButton>
# posStandX : List<Integer>
# posStandY : List<Integer>
# standardWidthChar : int
# standardHeightChar : int
# buttonHeight : int
# vocabo : Vocabulary
# dimChar : int
 # font : Font

+ DynaSlaveCard(master : DynaMasterCard, modulus : int <final>,
stringOptions : String[], yOps : int[], backgroundString : String) <constructor>
+ getMaster() : DynaMasterCard
+ getMod() : int
+ getMeterW() : int
+ getMeterH() : int
+ getFont() : Font
+ getButtons : List<JButton>
+ getAButton(i : int) : JButton
# initializePosStandards()
+ changePanel(panelName : String)
# makesButton(iOption : int)
# getAnInvisibleButton() : JButton
# addMouseListenerToTheButton(dsc : DynaSlaveCard, button : JButton,
iOption : int)
+ relocateComponents()
# redefinePosY(meterH : int, yOps : int[])
# redefinePosXCentralPositioning(standWidthChar : int)
# relocateButtonsToDrawnText(standWidthChar : int, bHeight : int)
# relocateAButtonToDrawnText(index:int, standWidthChar : int, bHeight : int)
+ setButton(b:JButton, x:int,y:int,bWidth:int,bHeight:int)
+ changeTextOfAnOptionAtRuntime(indexOption : int, newOptionText : String)
+ executeOptionByButton(pointer : int) <final>
```

```
# optionSwitcher(pointer : int) <abstract>
@Override
+ executeOption() <final>
+ paintDefScreen(drawBackground : boolean, drawOptions : boolean,
drawPointer : boolean)
# drawCurrAccount(g2 : Graphics2D)
# drawStuff(g2 : Graphics2D) Graphics2D <abstract>
# drawStringOptions(g2 : Graphics2D) : Graphics2D
# drawPointer(g2 : Graphics2D) : Graphics2D
# drawSkyscrapers(g2 : Graphics2D) : Graphics2D
# fountainPen(g2:Graphics2D, text:String, xInt:int, yInit:int,w:int,h:int) :
Graphics2D
# pencilPen(g2:Graphics2D, color:Color, text:String, y:int,x:int, dim:float)
: Graphics2D
# pencilPen(g2:Graphics2D, text:String, y:int, x:int, dim:float) :
Graphcis2D
# pencilPen(g2:Graphics2D, text:String, y:int, x:int)
```

```
JPanel
```

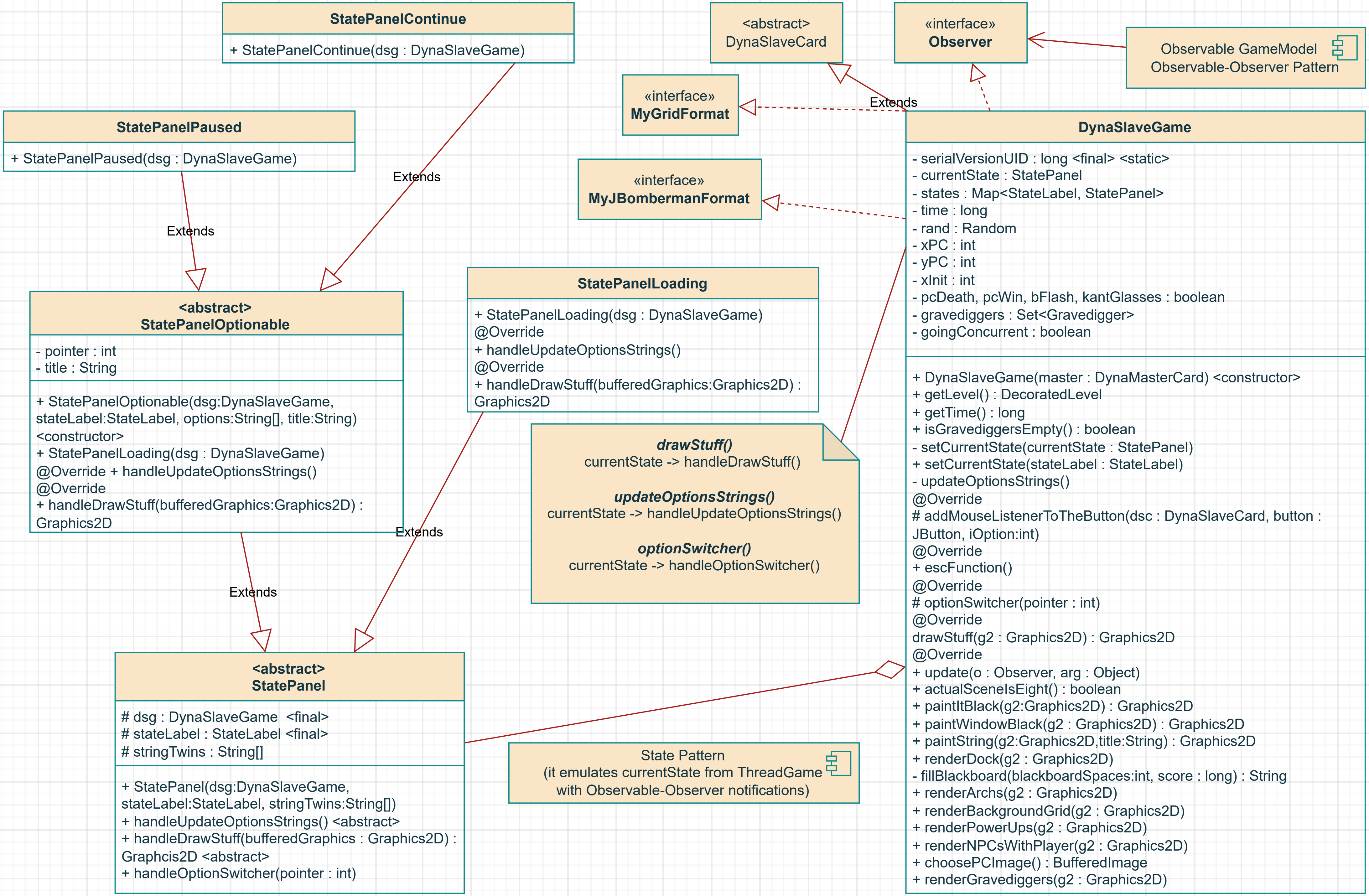
```
<<Interface>>
OptionExecuter

+ executeOption()
+ escFunction()
```

```
CardLayout
```

```
DynaMasterCard
```

```
Concrete DynaSlaveCards

DynaSaveStart,
DynaSlavePlayer,
DynaSlaveStage,
DynaSlaveOption,
DynaSlaveAccount,
DynaSlaveAccountsManager,
DynaSlaveAccountsStats,
DynaSlaveAccountCreation,
DynaSlaveGame
```

```
Vocabulary

+ instance : Vocabulary <static>
- vocabulary :
Map<Character,BufferedImage>

- Vocabulary() <constructor>
+ getInstance() : Vocabulary
<static>
+ getIm(c : char) : BufferedImage
```



StatePanelContinue

+ StatePanelContinue(dsg : DynaSlaveGame)

StatePanelPaused

+ StatePanelPaused(dsg : DynaSlaveGame)

<abstract>
DynaSlaveCard

«interface»
Observer

Observable GameModel
Observable-Observer Pattern

«interface»
MyGridFormat

«interface»
MyJBombermanFormat

**<abstract>
StatePanelOptionable**

- pointer : int
- title : String

+ StatePanelOptionable(dsg:DynaSlaveGame, stateLabel:StateLabel, options:String[], title:String) <constructor>
+ StatePanelLoading(dsg : DynaSlaveGame)
@Override + handleUpdateOptionsStrings()
@Override
+ handleDrawStuff(bufferedGraphics:Graphics2D) : Graphics2D

StatePanelLoading

+ StatePanelLoading(dsg : DynaSlaveGame)
@Override
+ handleUpdateOptionsStrings()
@Override
+ handleDrawStuff(bufferedGraphics:Graphics2D) : Graphics2D

drawStuff()
currentState -> handleDrawStuff()

updateOptionsStrings()
currentState -> handleUpdateOptionsStrings()

optionSwitcher()
currentState -> handleOptionSwitcher()

**<abstract>
StatePanel**

# dsg : DynaSlaveGame <final>
# stateLabel : StateLabel <final>
# stringTwins : String[]

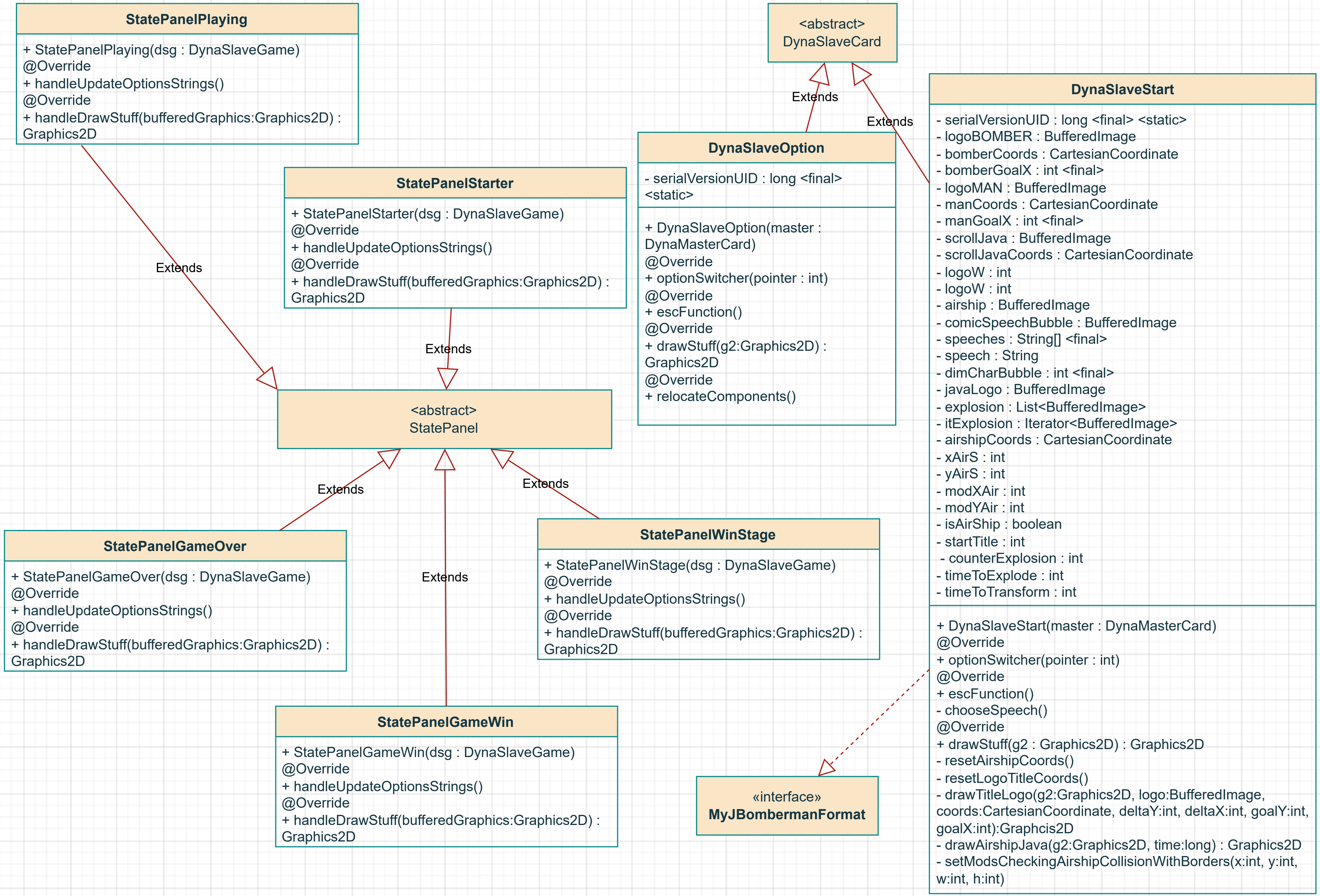
+ StatePanel(dsg:DynaSlaveGame, stateLabel:StateLabel, stringTwins:String[])
+ handleUpdateOptionsStrings() <abstract>
+ handleDrawStuff(bufferedGraphics : Graphics2D) : Graphcis2D <abstract>
+ handleOptionSwitcher(pointer : int)

State Pattern
(it emulates currentState from ThreadGame
with Observable-Observer notifications)

DynaSlaveGame

- serialVersionUID : long <final> <static>
- currentState : StatePanel
- states : Map<StateLabel, StatePanel>
- time : long
- rand : Random
- xPC : int
- yPC : int
- xInit : int
- pcDeath, pcWin, bFlash, kantGlasses : boolean
- gravediggers : Set<Gravedigger>
- goingConcurrent : boolean

+ DynaSlaveGame(master : DynaMasterCard) <constructor>
+ getLevel() : DecoratedLevel
+ getTime() : long
+ isGravediggersEmpty() : boolean
- setCurrentState(currentState : StatePanel)
+ setCurrentState(stateLabel : StateLabel)
- updateOptionsStrings()
@Override
# addMouseListenerToTheButton(dsc : DynaSlaveCard, button : JButton, iOption:int)
@Override
+ escFunction()
@Override
# optionSwitcher(pointer : int)
@Override
drawStuff(g2 : Graphics2D) : Graphics2D
@Override
+ update(o : Observer, arg : Object)
+ actualSceneIsEight() : boolean
+ paintItBlack(g2:Graphics2D) : Graphics2D
+ paintWindowBlack(g2 : Graphics2D) : Graphics2D
+ paintString(g2:Graphics2D,title:String) : Graphics2D
+ renderDock(g2 : Graphics2D)
- fillBlackboard(blackboardSpaces:int, score : long) : String
+ renderArchs(g2 : Graphics2D)
+ renderBackgroundGrid(g2 : Graphics2D)
+ renderPowerUps(g2 : Graphics2D)
+ renderNPCsWithPlayer(g2 : Graphics2D)
+ choosePCImage() : BufferedImage
+ renderGravediggers(g2 : Graphics2D)



StatePanelPlaying

+ StatePanelPlaying(dsg : DynaSlaveGame)
@Override
+ handleUpdateOptionsStrings()
@Override
+ handleDrawStuff(bufferedGraphics:Graphics2D) : Graphics2D

StatePanelStarter

+ StatePanelStarter(dsg : DynaSlaveGame)
@Override
+ handleUpdateOptionsStrings()
@Override
+ handleDrawStuff(bufferedGraphics:Graphics2D) : Graphics2D

<abstract>
StatePanel

StatePanelGameOver

+ StatePanelGameOver(dsg : DynaSlaveGame)
@Override
+ handleUpdateOptionsStrings()
@Override
+ handleDrawStuff(bufferedGraphics:Graphics2D) : Graphics2D

StatePanelWinStage

+ StatePanelWinStage(dsg : DynaSlaveGame)
@Override
+ handleUpdateOptionsStrings()
@Override
+ handleDrawStuff(bufferedGraphics:Graphics2D) : Graphics2D

StatePanelGameWin

+ StatePanelGameWin(dsg : DynaSlaveGame)
@Override
+ handleUpdateOptionsStrings()
@Override
+ handleDrawStuff(bufferedGraphics:Graphics2D) : Graphics2D

<abstract>
DynaSlaveCard

DynaSlaveOption

- serialVersionUID : long <final> <static>

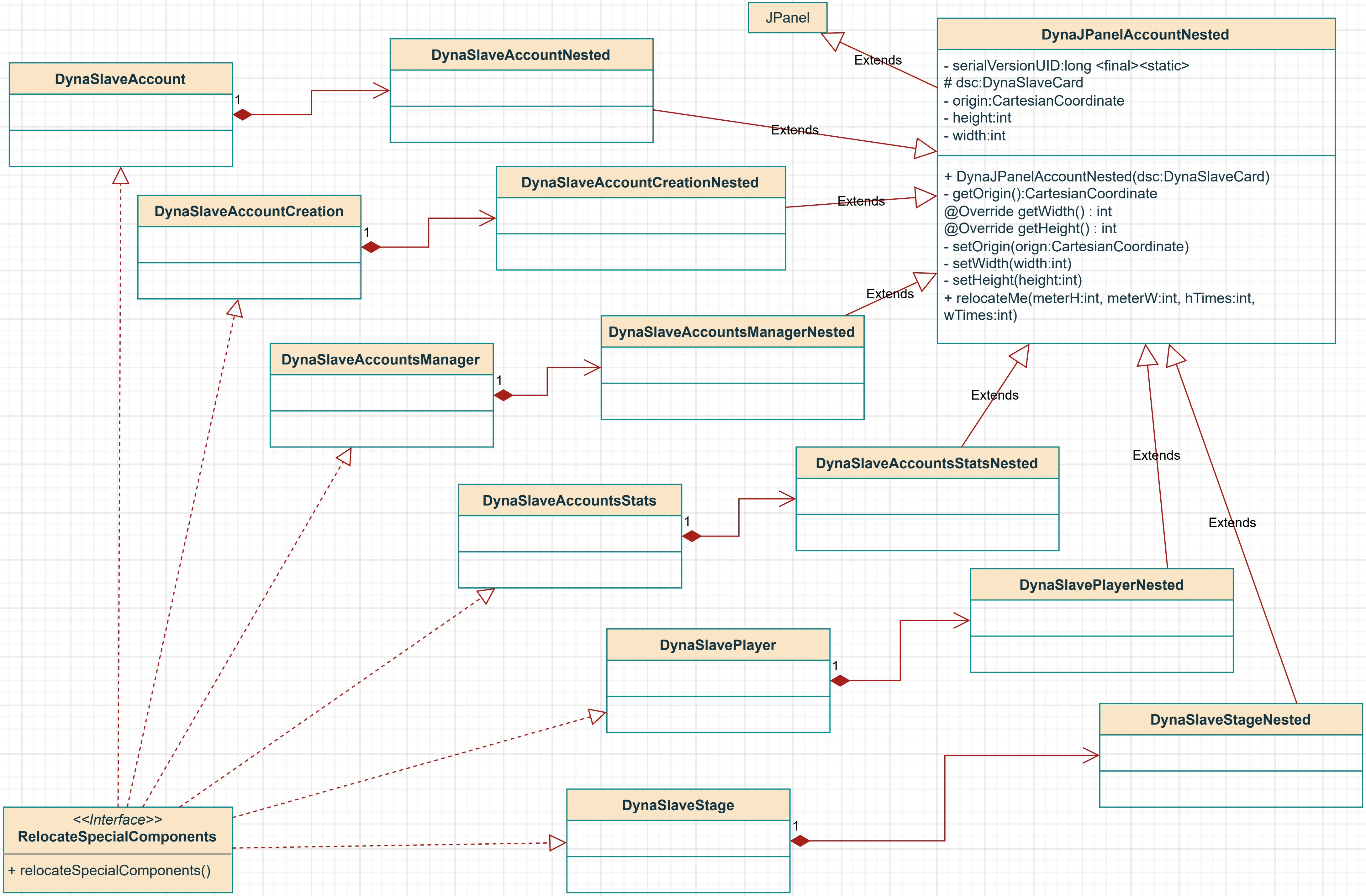
+ DynaSlaveOption(master : DynaMasterCard)
@Override
+ optionSwitcher(pointer : int)
@Override
+ escFunction()
@Override
+ drawStuff(g2:Graphics2D) : Graphics2D
@Override
+ relocateComponents()

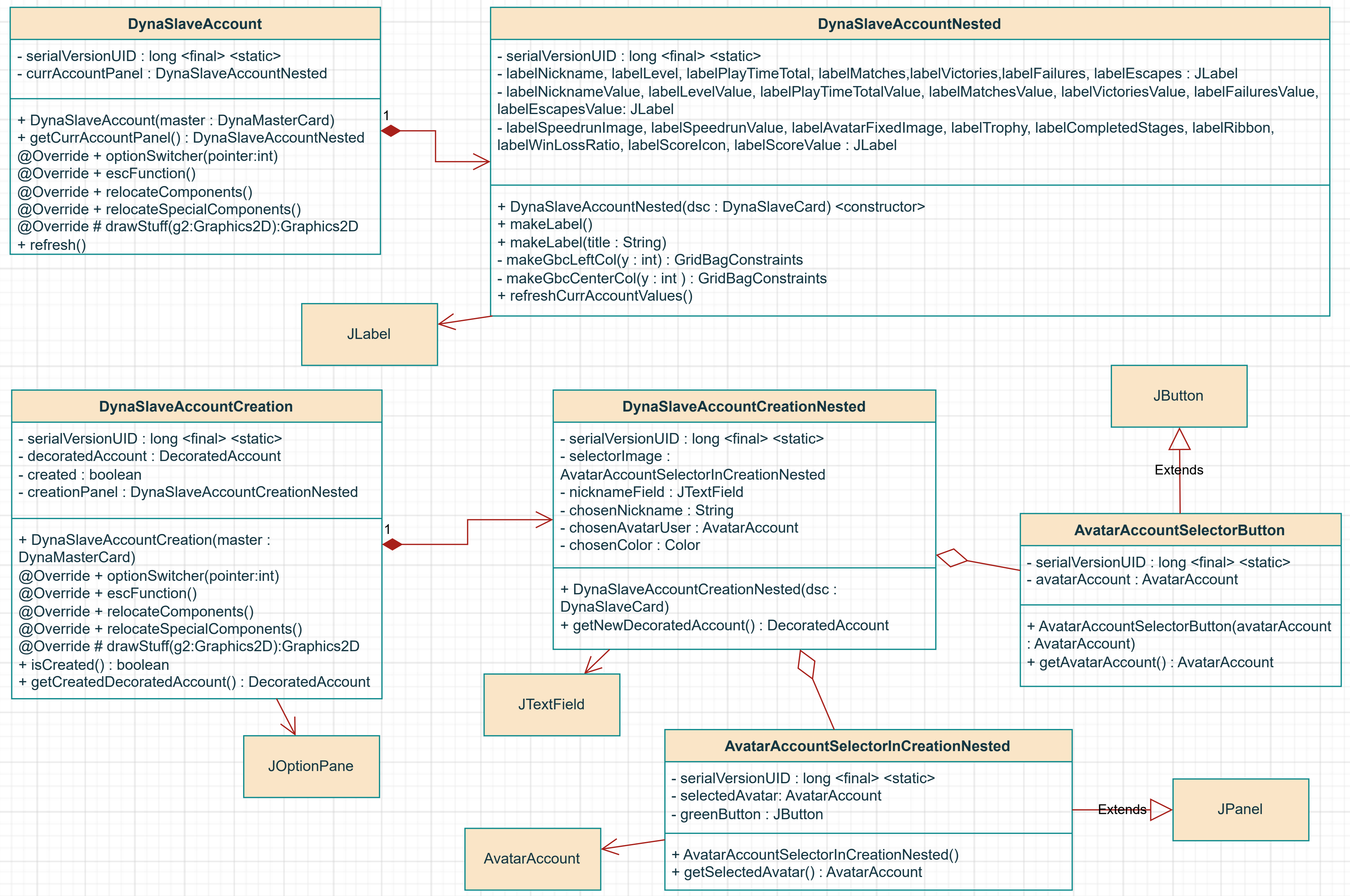
DynaSlaveStart

- serialVersionUID : long <final> <static>
- logoBOMBER : BufferedImage
- bomberCoords : CartesianCoordinate
- bomberGoalX : int <final>
- logoMAN : BufferedImage
- manCoords : CartesianCoordinate
- manGoalX : int <final>
- scrollJava : BufferedImage
- scrollJavaCoords : CartesianCoordinate
- logoW : int
- logoW : int
- airship : BufferedImage
- comicSpeechBubble : BufferedImage
- speeches : String[] <final>
- speech : String
- dimCharBubble : int <final>
- javaLogo : BufferedImage
- explosion : List<BufferedImage>
- itExplosion : Iterator<BufferedImage>
- airshipCoords : CartesianCoordinate
- xAirS : int
- yAirS : int
- modXAir : int
- modYAir : int
- isAirShip : boolean
- startTitle : int
- counterExplosion : int
- timeToExplode : int
- timeToTransform : int

+ DynaSlaveStart(master : DynaMasterCard)
@Override
+ optionSwitcher(pointer : int)
@Override
+ escFunction()
- chooseSpeech()
@Override
+ drawStuff(g2 : Graphics2D) : Graphics2D
- resetAirshipCoords()
- resetLogoTitleCoords()
- drawTitleLogo(g2:Graphics2D, logo:BufferedImage, coords:CartesianCoordinate, deltaY:int, deltaX:int, goalY:int, goalX:int):Graphcis2D
- drawAirshipJava(g2:Graphics2D, time:long) : Graphics2D
- setModsCheckingAirshipCollisionWithBorders(x:int, y:int, w:int, h:int)

«interface»
MyJBombermanFormat


JPanel
DynaJPanelAccountNested
- serialVersionUID:long <final><static>
# dsc:DynaSlaveCard
- origin:CartesianCoordinate
- height:int
- width:int
+ DynaJPanelAccountNested(dsc:DynaSlaveCard)
- getOrigin():CartesianCoordinate
@Override getWidth() : int
@Override getHeight() : int
- setOrigin(orign:CartesianCoordinate)
- setWidth(width:int)
- setHeight(height:int)
+ relocateMe(meterH:int, meterW:int, hTimes:int, wTimes:int)
Extends
DynaSlaveAccountNested
DynaSlaveAccount
1
Extends
DynaSlaveAccountCreationNested
DynaSlaveAccountCreation
1
Extends
Extends
DynaSlaveAccountsManagerNested
DynaSlaveAccountsManager
1
Extends
DynaSlaveAccountsStatsNested
DynaSlaveAccountsStats
1
Extends
Extends
DynaSlavePlayerNested
DynaSlavePlayer
1
DynaSlaveStageNested
<<Interface>>
RelocateSpecialComponents
+ relocateSpecialComponents()
DynaSlaveStage
1



DynaSlaveAccount

- serialVersionUID : long <final> <static>
- currAccountPanel : DynaSlaveAccountNested

+ DynaSlaveAccount(master : DynaMasterCard)
+ getCurrAccountPanel() : DynaSlaveAccountNested
@Override + optionSwitcher(pointer:int)
@Override + escFunction()
@Override + relocateComponents()
@Override + relocateSpecialComponents()
@Override # drawStuff(g2:Graphics2D):Graphics2D
+ refresh()

DynaSlaveAccountNested

- serialVersionUID : long <final> <static>
- labelNickname, labelLevel, labelPlayTimeTotal, labelMatches,labelVictories,labelFailures, labelEscapes : JLabel
- labelNicknameValue, labelLevelValue, labelPlayTimeTotalValue, labelMatchesValue, labelVictoriesValue, labelFailuresValue, labelEscapesValue: JLabel
- labelSpeedrunImage, labelSpeedrunValue, labelAvatarFixedImage, labelTrophy, labelCompletedStages, labelRibbon, labelWinLossRatio, labelScoreIcon, labelScoreValue : JLabel

+ DynaSlaveAccountNested(dsc : DynaSlaveCard) <constructor>
+ makeLabel()
+ makeLabel(title : String)
- makeGbcLeftCol(y : int) : GridBagConstraints
- makeGbcCenterCol(y : int) : GridBagConstraints
+ refreshCurrAccountValues()

JLabel

DynaSlaveAccountCreation

- serialVersionUID : long <final> <static>
- decoratedAccount : DecoratedAccount
- created : boolean
- creationPanel : DynaSlaveAccountCreationNested

+ DynaSlaveAccountCreation(master : DynaMasterCard)
@Override + optionSwitcher(pointer:int)
@Override + escFunction()
@Override + relocateComponents()
@Override + relocateSpecialComponents()
@Override # drawStuff(g2:Graphics2D):Graphics2D
+ isCreated() : boolean
+ getCreatedDecoratedAccount() : DecoratedAccount

DynaSlaveAccountCreationNested

- serialVersionUID : long <final> <static>
- selectorImage : AvatarAccountSelectorInCreationNested
- nicknameField : JTextField
- chosenNickname : String
- chosenAvatarUser : AvatarAccount
- chosenColor : Color

+ DynaSlaveAccountCreationNested(dsc : DynaSlaveCard)
+ getNewDecoratedAccount() : DecoratedAccount

JButton

AvatarAccountSelectorButton

- serialVersionUID : long <final> <static>
- avatarAccount : AvatarAccount

+ AvatarAccountSelectorButton(avatarAccount : AvatarAccount)
+ getAvatarAccount() : AvatarAccount

JTextField

JOptionPane

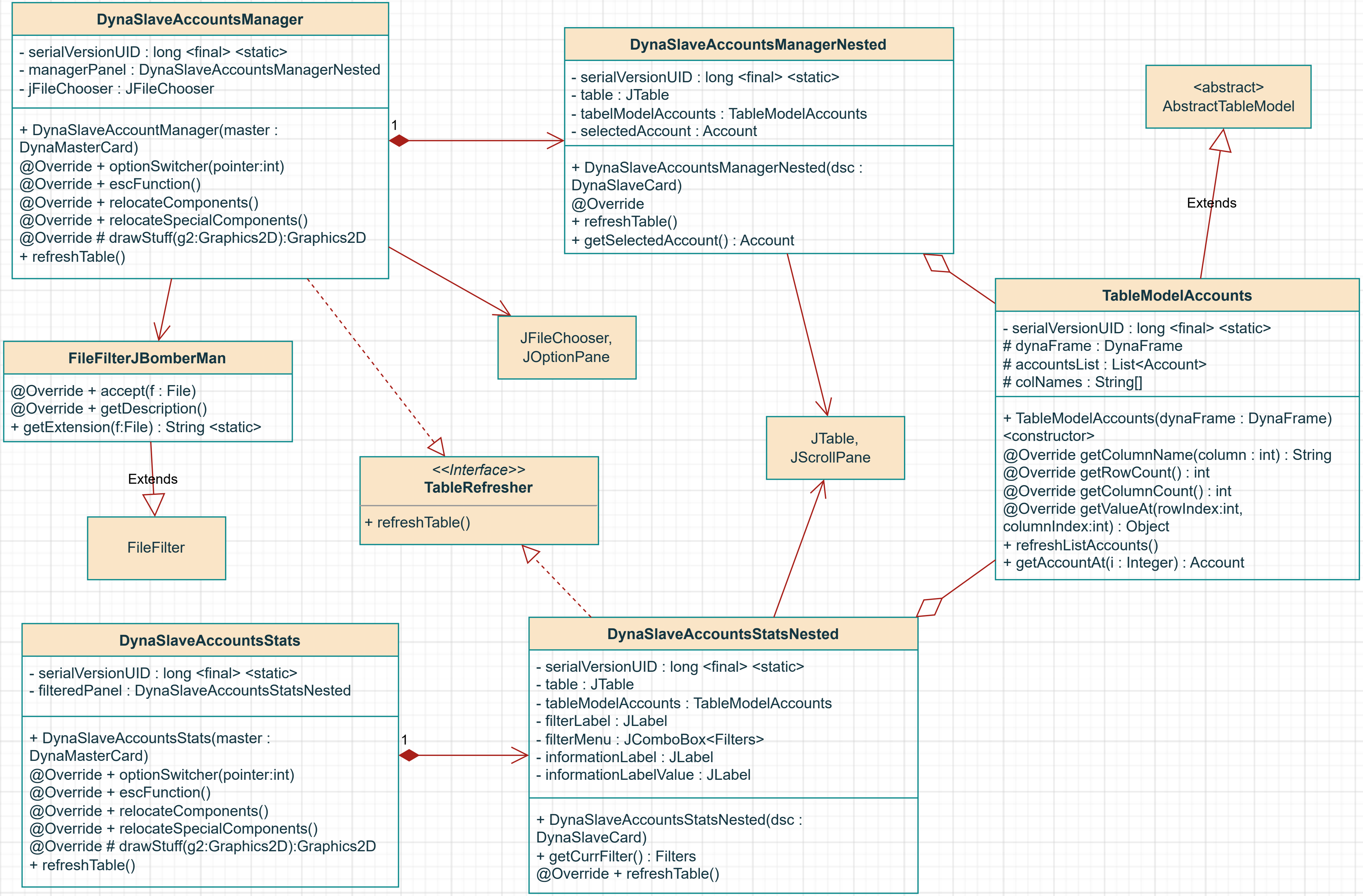
AvatarAccountSelectorInCreationNested

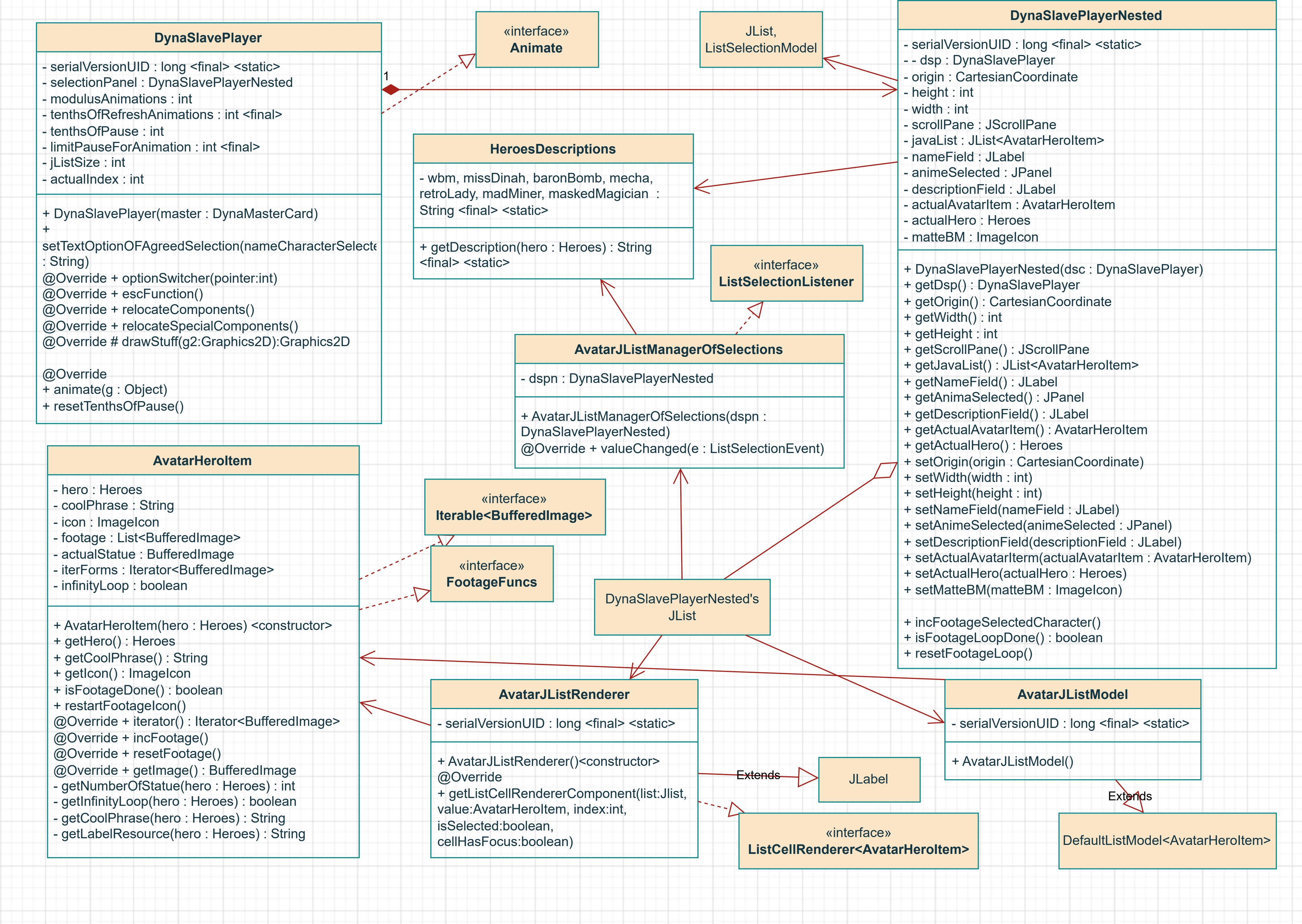
- serialVersionUID : long <final> <static>
- selectedAvatar: AvatarAccount
- greenButton : JButton

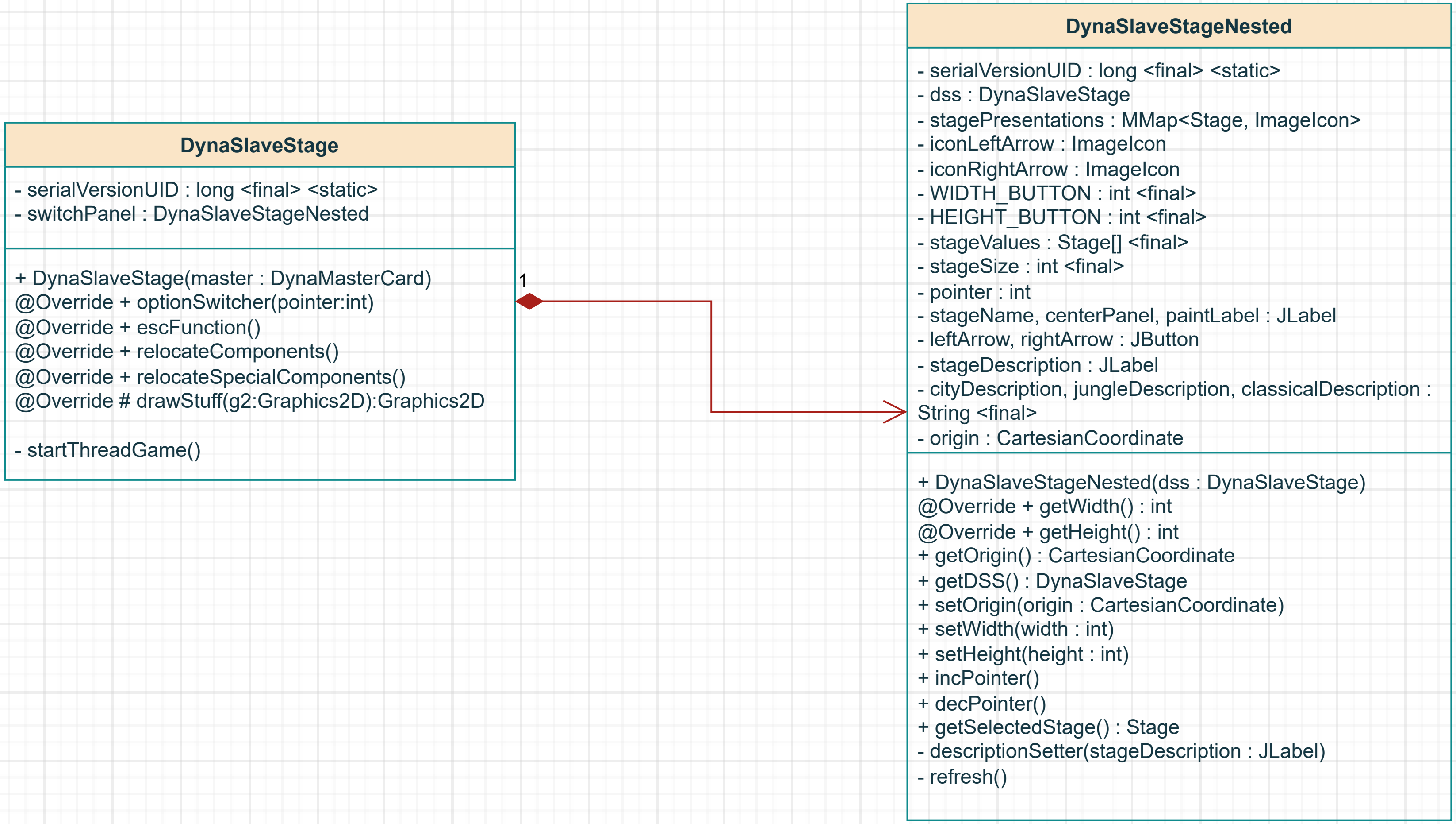
+ AvatarAccountSelectorInCreationNested()
+ getSelectedAvatar() : AvatarAccount

AvatarAccount

JPanel


DynaSlaveAccountsManager
- serialVersionUID : long <final> <static>
- managerPanel : DynaSlaveAccountsManagerNested
- jFileChooser : JFileChooser
+ DynaSlaveAccountManager(master : DynaMasterCard)
@Override + optionSwitcher(pointer:int)
@Override + escFunction()
@Override + relocateComponents()
@Override + relocateSpecialComponents()
@Override # drawStuff(g2:Graphics2D):Graphics2D
+ refreshTable()
1
DynaSlaveAccountsManagerNested
- serialVersionUID : long <final> <static>
- table : JTable
- tabelModelAccounts : TableModelAccounts
- selectedAccount : Account
+ DynaSlaveAccountsManagerNested(dsc : DynaSlaveCard)
@Override
+ refreshTable()
+ getSelectedAccount() : Account
<abstract>
AbstractTableModel
Extends
TableModelAccounts
- serialVersionUID : long <final> <static>
# dynaFrame : DynaFrame
# accountsList : List<Account>
# colNames : String[]
+ TableModelAccounts(dynaFrame : DynaFrame) <constructor>
@Override getColumnName(column : int) : String
@Override getRowCount() : int
@Override getColumnCount() : int
@Override getValueAt(rowIndex:int, columnIndex:int) : Object
+ refreshListAccounts()
+ getAccountAt(i : Integer) : Account
FileFilterJBomberMan
@Override + accept(f : File)
@Override + getDescription()
+ getExtension(f:File) : String <static>
Extends
FileFilter
JFileChooser,
JOptionPane
<<Interface>>
TableRefresher
+ refreshTable()
JTable,
JScrollPane
DynaSlaveAccountsStats
- serialVersionUID : long <final> <static>
- filteredPanel : DynaSlaveAccountsStatsNested
+ DynaSlaveAccountsStats(master : DynaMasterCard)
@Override + optionSwitcher(pointer:int)
@Override + escFunction()
@Override + relocateComponents()
@Override + relocateSpecialComponents()
@Override # drawStuff(g2:Graphics2D):Graphics2D
+ refreshTable()
1
DynaSlaveAccountsStatsNested
- serialVersionUID : long <final> <static>
- table : JTable
- tableModelAccounts : TableModelAccounts
- filterLabel : JLabel
- filterMenu : JComboBox<Filters>
- informationLabel : JLabel
- informationLabelValue : JLabel
+ DynaSlaveAccountsStatsNested(dsc : DynaSlaveCard)
+ getCurrFilter() : Filters
@Override + refreshTable()


DynaSlavePlayer
- serialVersionUID : long <final> <static>
- selectionPanel : DynaSlavePlayerNested
- modulusAnimations : int
- tenthsOfRefreshAnimations : int <final>
- tenthsOfPause : int
- limitPauseForAnimation : int <final>
- jListSize : int
- actualIndex : int
+ DynaSlavePlayer(master : DynaMasterCard)
+ setTextOptionOFAgreedSelection(nameCharacterSelecte : String)
@Override + optionSwitcher(pointer:int)
@Override + escFunction()
@Override + relocateComponents()
@Override + relocateSpecialComponents()
@Override # drawStuff(g2:Graphics2D):Graphics2D
@Override
+ animate(g : Object)
+ resetTenthsOfPause()
1
«interface»
Animate
JList,
ListSelectionModel
DynaSlavePlayerNested
- serialVersionUID : long <final> <static>
- - dsp : DynaSlavePlayer
- origin : CartesianCoordinate
- height : int
- width : int
- scrollPane : JScrollPane
- javaList : JList<AvatarHeroItem>
- nameField : JLabel
- animeSelected : JPanel
- descriptionField : JLabel
- actualAvatarItem : AvatarHeroItem
- actualHero : Heroes
- matteBM : ImageIcon
+ DynaSlavePlayerNested(dsc : DynaSlavePlayer)
+ getDsp() : DynaSlavePlayer
+ getOrigin() : CartesianCoordinate
+ getWidth() : int
+ getHeight : int
+ getScrollPane() : JScrollPane
+ getJavaList() : JList<AvatarHeroItem>
+ getNameField() : JLabel
+ getAnimaSelected() : JPanel
+ getDescriptionField() : JLabel
+ getActualAvatarItem() : AvatarHeroItem
+ getActualHero() : Heroes
+ setOrigin(origin : CartesianCoordinate)
+ setWidth(width : int)
+ setHeight(height : int)
+ setNameField(nameField : JLabel)
+ setAnimeSelected(animeSelected : JPanel)
+ setDescriptionField(descriptionField : JLabel)
+ setActualAvatarIterm(actualAvatarItem : AvatarHeroItem)
+ setActualHero(actualHero : Heroes)
+ setMatteBM(matteBM : ImageIcon)
+ incFootageSelectedCharacter()
+ isFootageLoopDone() : boolean
+ resetFootageLoop()
HeroesDescriptions
- wbm, missDinah, baronBomb, mecha, retroLady, madMiner, maskedMagician : String <final> <static>
+ getDescription(hero : Heroes) : String <final> <static>
«interface»
ListSelectionListener
AvatarJListManagerOfSelections
- dspn : DynaSlavePlayerNested
+ AvatarJListManagerOfSelections(dspn : DynaSlavePlayerNested)
@Override + valueChanged(e : ListSelectionEvent)
AvatarHeroItem
- hero : Heroes
- coolPhrase : String
- icon : ImageIcon
- footage : List<BufferedImage>
- actualStatue : BufferedImage
- iterForms : Iterator<BufferedImage>
- infinityLoop : boolean
+ AvatarHeroItem(hero : Heroes) <constructor>
+ getHero() : Heroes
+ getCoolPhrase() : String
+ getIcon() : ImageIcon
+ isFootageDone() : boolean
+ restartFootageIcon()
@Override + iterator() : Iterator<BufferedImage>
@Override + incFootage()
@Override + resetFootage()
@Override + getImage() : BufferedImage
- getNumberOfStatue(hero : Heroes) : int
- getInfinityLoop(hero : Heroes) : boolean
- getCoolPhrase(hero : Heroes) : String
- getLabelResource(hero : Heroes) : String
«interface»
Iterable<BufferedImage>
«interface»
FootageFuncs
DynaSlavePlayerNested's
JList
AvatarJListRenderer
- serialVersionUID : long <final> <static>
+ AvatarJListRenderer()<constructor>
@Override
+ getListCellRendererComponent(list:Jlist, value:AvatarHeroItem, index:int, isSelected:boolean, cellHasFocus:boolean)
Extends
JLabel
«interface»
ListCellRenderer<AvatarHeroItem>
AvatarJListModel
- serialVersionUID : long <final> <static>
+ AvatarJListModel()
Extends
DefaultListModel<AvatarHeroItem>



DynaSlaveStage

- serialVersionUID : long <final> <static>
- switchPanel : DynaSlaveStageNested

+ DynaSlaveStage(master : DynaMasterCard)
@Override + optionSwitcher(pointer:int)
@Override + escFunction()
@Override + relocateComponents()
@Override + relocateSpecialComponents()
@Override # drawStuff(g2:Graphics2D):Graphics2D

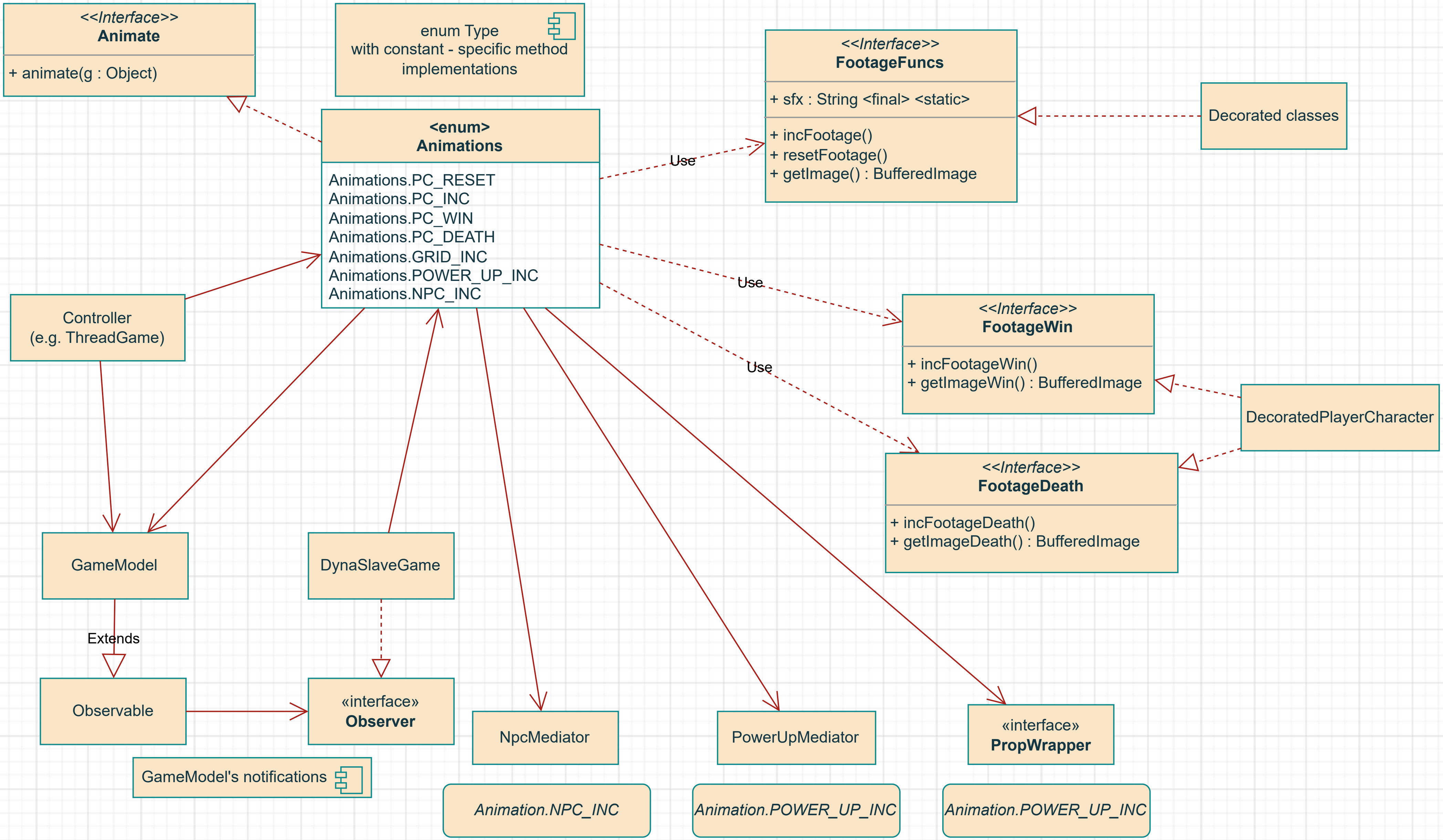
- startThreadGame()

DynaSlaveStageNested

- serialVersionUID : long <final> <static>
- dss : DynaSlaveStage
- stagePresentations : MMap<Stage, ImageIcon>
- iconLeftArrow : ImageIcon
- iconRightArrow : ImageIcon
- WIDTH_BUTTON : int <final>
- HEIGHT_BUTTON : int <final>
- stageValues : Stage[] <final>
- stageSize : int <final>
- pointer : int
- stageName, centerPanel, paintLabel : JLabel
- leftArrow, rightArrow : JButton
- stageDescription : JLabel
- cityDescription, jungleDescription, classicalDescription : String <final>
- origin : CartesianCoordinate

+ DynaSlaveStageNested(dss : DynaSlaveStage)
@Override + getWidth() : int
@Override + getHeight() : int
+ getOrigin() : CartesianCoordinate
+ getDSS() : DynaSlaveStage
+ setOrigin(origin : CartesianCoordinate)
+ setWidth(width : int)
+ setHeight(height : int)
+ incPointer()
+ decPointer()
+ getSelectedStage() : Stage
- descriptionSetter(stageDescription : JLabel)
- refresh()

View UML: Animations with Iterator Pattern and Decorated Classes with Decorator Pattern



<<Interface>>
Animate

+ animate(g : Object)

enum Type
with constant - specific method
implementations

<enum>
Animations

Animations.PC_RESET
Animations.PC_INC
Animations.PC_WIN
Animations.PC_DEATH
Animations.GRID_INC
Animations.POWER_UP_INC
Animations.NPC_INC

<<Interface>>
FootageFuncs

+ sfx : String <final> <static>

+ incFootage()
+ resetFootage()
+ getImage() : BufferedImage

Decorated classes

Use

Use

Use

<<Interface>>
FootageWin

+ incFootageWin()
+ getImageWin() : BufferedImage

DecoratedPlayerCharacter

<<Interface>>
FootageDeath

+ incFootageDeath()
+ getImageDeath() : BufferedImage

Controller
(e.g. ThreadGame)

GameModel

DynaSlaveGame

Extends

Observable

«interface»
Observer

GameModel's notifications

NpcMediator

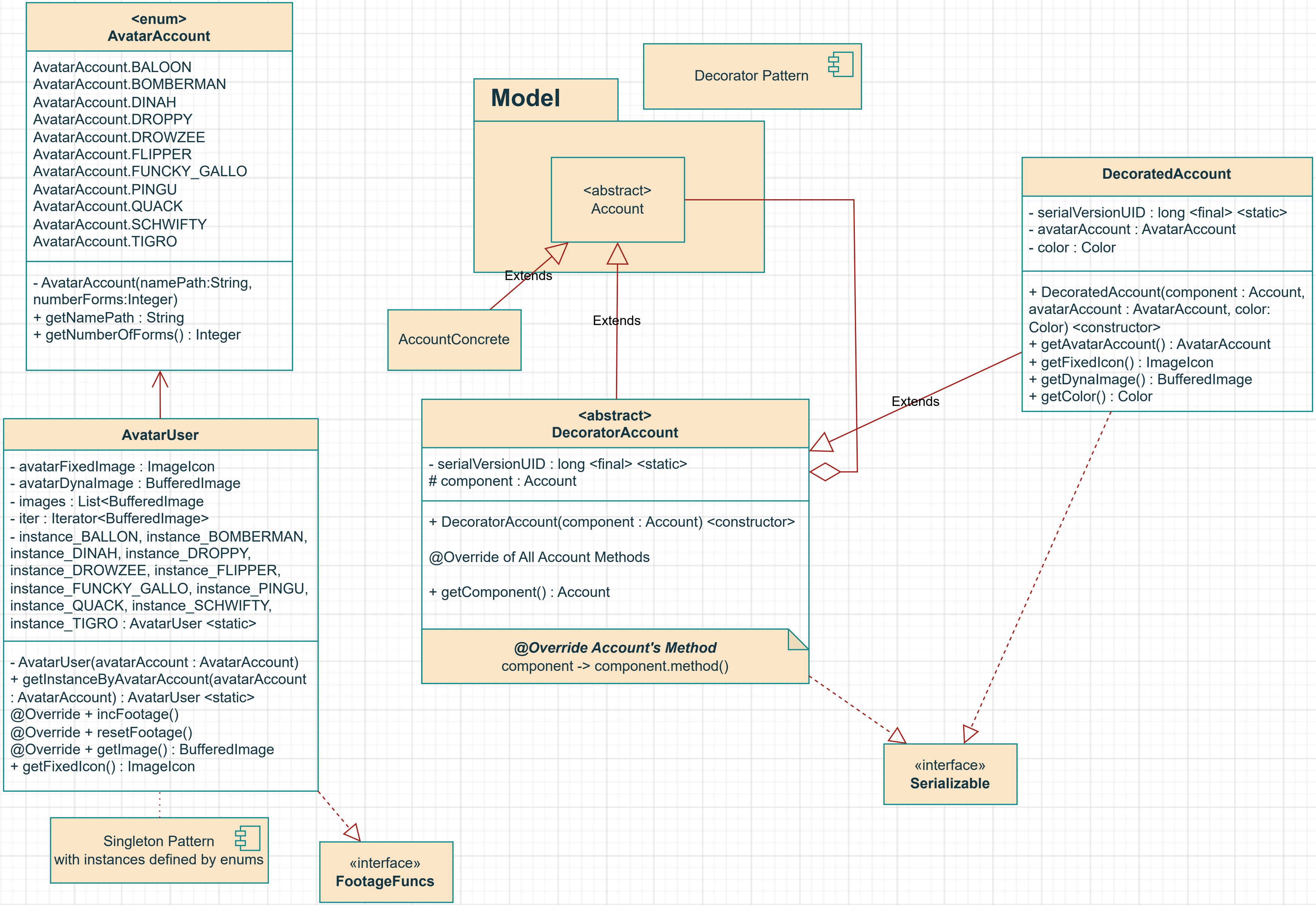
PowerUpMediator

«interface»
PropWrapper

Animation.NPC_INC

Animation.POWER_UP_INC

Animation.POWER_UP_INC



<enum>
AvatarAccount

AvatarAccount.BALOON
AvatarAccount.BOMBERMAN
AvatarAccount.DINAH
AvatarAccount.DROPPY
AvatarAccount.DROWZEE
AvatarAccount.FLIPPER
AvatarAccount.FUNCKY_GALLO
AvatarAccount.PINGU
AvatarAccount.QUACK
AvatarAccount.SCHWIFTY
AvatarAccount.TIGRO

- AvatarAccount(namePath:String, numberForms:Integer)
+ getNamePath : String
+ getNumberOfForms() : Integer

Model

<abstract>
Account

AccountConcrete

Decorator Pattern

DecoratedAccount

- serialVersionUID : long <final> <static>
- avatarAccount : AvatarAccount
- color : Color

+ DecoratedAccount(component : Account, avatarAccount : AvatarAccount, color: Color) <constructor>
+ getAvatarAccount() : AvatarAccount
+ getFixedIcon() : ImageIcon
+ getDynaImage() : BufferedImage
+ getColor() : Color

AvatarUser

- avatarFixedImage : ImageIcon
- avatarDynaImage : BufferedImage
- images : List<BufferedImage
- iter : Iterator<BufferedImage>
- instance_BALLON, instance_BOMBERMAN, instance_DINAH, instance_DROPPY, instance_DROWZEE, instance_FLIPPER, instance_FUNCKY_GALLO, instance_PINGU, instance_QUACK, instance_SCHWIFTY, instance_TIGRO : AvatarUser <static>

- AvatarUser(avatarAccount : AvatarAccount)
+ getInstanceByAvatarAccount(avatarAccount : AvatarAccount) : AvatarUser <static>
@Override + incFootage()
@Override + resetFootage()
@Override + getImage() : BufferedImage
+ getFixedIcon() : ImageIcon

<abstract>
DecoratorAccount

- serialVersionUID : long <final> <static>
# component : Account

+ DecoratorAccount(component : Account) <constructor>

@Override of All Account Methods

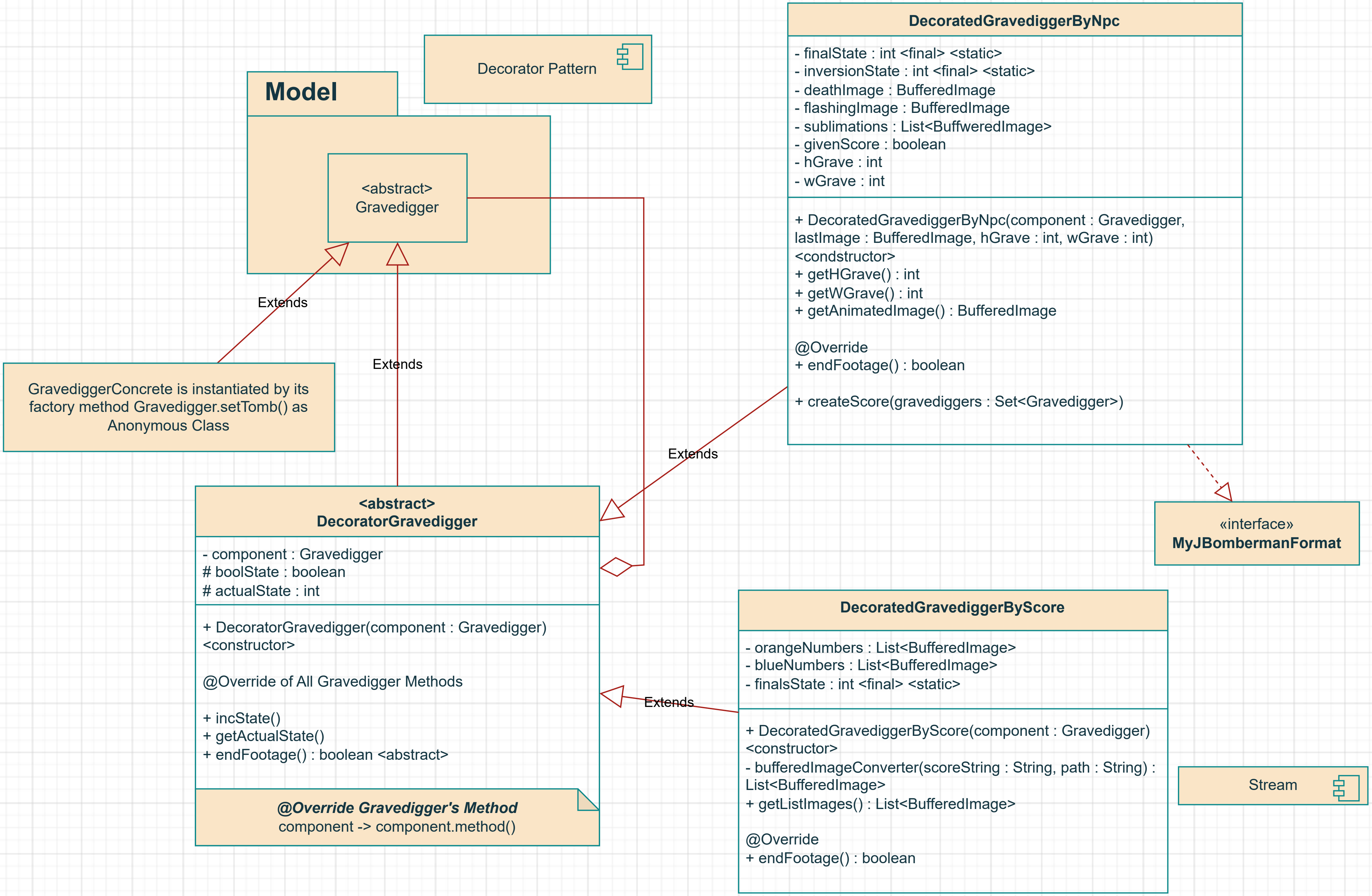
+ getComponent() : Account

@Override Account's Method
component -> component.method()

«interface»
Serializable

Singleton Pattern
with instances defined by enums

«interface»
FootageFuncs



Decorator Pattern

Model

<abstract>
Gravedigger

GravediggerConcrete is instantiated by its factory method Gravedigger.setTomb() as Anonymous Class

<abstract>
DecoratorGravedigger

- component : Gravedigger
# boolState : boolean
# actualState : int

+ DecoratorGravedigger(component : Gravedigger) <constructor>

@Override of All Gravedigger Methods

+ incState()
+ getActualState()
+ endFootage() : boolean <abstract>

@Override Gravedigger's Method
component -> component.method()

DecoratedGravediggerByNpc

- finalState : int <final> <static>
- inversionState : int <final> <static>
- deathImage : BufferedImage
- flashingImage : BufferedImage
- sublimations : List<BuffweredImage>
- givenScore : boolean
- hGrave : int
- wGrave : int

+ DecoratedGravediggerByNpc(component : Gravedigger, lastImage : BufferedImage, hGrave : int, wGrave : int) <condstructor>
+ getHGrave() : int
+ getWGrave() : int
+ getAnimatedImage() : BufferedImage

@Override
+ endFootage() : boolean

+ createScore(gravediggers : Set<Gravedigger>)

«interface»
MyJBombermanFormat

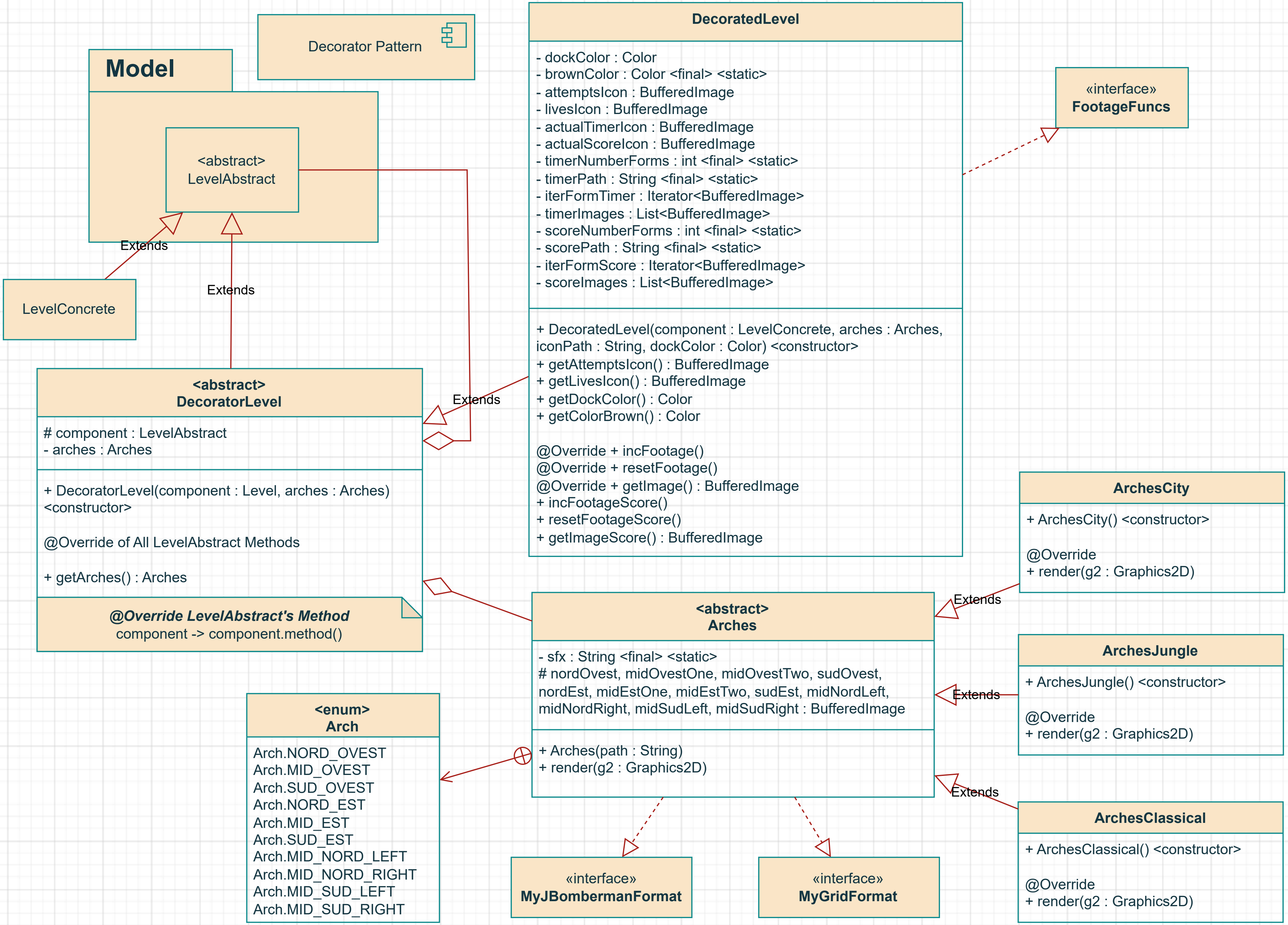
DecoratedGravediggerByScore

- orangeNumbers : List<BufferedImage>
- blueNumbers : List<BufferedImage>
- finalsState : int <final> <static>

+ DecoratedGravediggerByScore(component : Gravedigger) <constructor>
- bufferedImageConverter(scoreString : String, path : String) : List<BufferedImage>
+ getListImages() : List<BufferedImage>

@Override
+ endFootage() : boolean

Stream



DecoratedLevel

- dockColor : Color
- brownColor : Color <final> <static>
- attemptsIcon : BufferedImage
- livesIcon : BufferedImage
- actualTimerIcon : BufferedImage
- actualScoreIcon : BufferedImage
- timerNumberForms : int <final> <static>
- timerPath : String <final> <static>
- iterFormTimer : Iterator<BufferedImage>
- timerImages : List<BufferedImage>
- scoreNumberForms : int <final> <static>
- scorePath : String <final> <static>
- iterFormScore : Iterator<BufferedImage>
- scoreImages : List<BufferedImage>

+ DecoratedLevel(component : LevelConcrete, arches : Arches, iconPath : String, dockColor : Color) <constructor>
+ getAttemptsIcon() : BufferedImage
+ getLivesIcon() : BufferedImage
+ getDockColor() : Color
+ getColorBrown() : Color

@Override + incFootage()
@Override + resetFootage()
@Override + getImage() : BufferedImage
+ incFootageScore()
+ resetFootageScore()
+ getImageScore() : BufferedImage

<abstract>
DecoratorLevel

# component : LevelAbstract
- arches : Arches

+ DecoratorLevel(component : Level, arches : Arches) <constructor>

@Override of All LevelAbstract Methods

+ getArches() : Arches

@Override LevelAbstract's Method
component -> component.method()

ArchesCity

+ ArchesCity() <constructor>

@Override
+ render(g2 : Graphics2D)

<abstract>
Arches

- sfx : String <final> <static>
# nordOvest, midOvestOne, midOvestTwo, sudOvest, nordEst, midEstOne, midEstTwo, sudEst, midNordLeft, midNordRight, midSudLeft, midSudRight : BufferedImage

+ Arches(path : String)
+ render(g2 : Graphics2D)

ArchesJungle

+ ArchesJungle() <constructor>

@Override
+ render(g2 : Graphics2D)

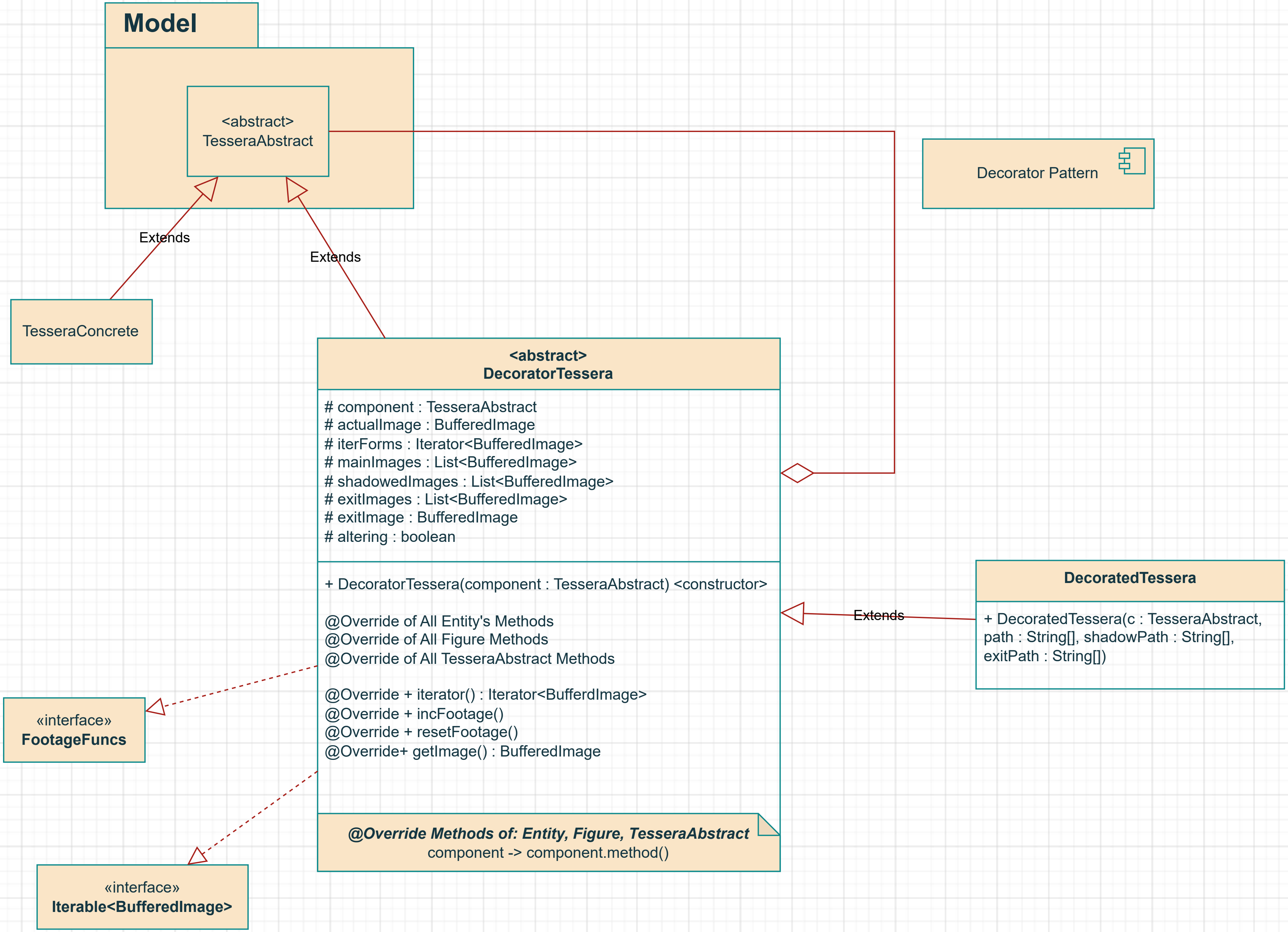
<enum>
Arch

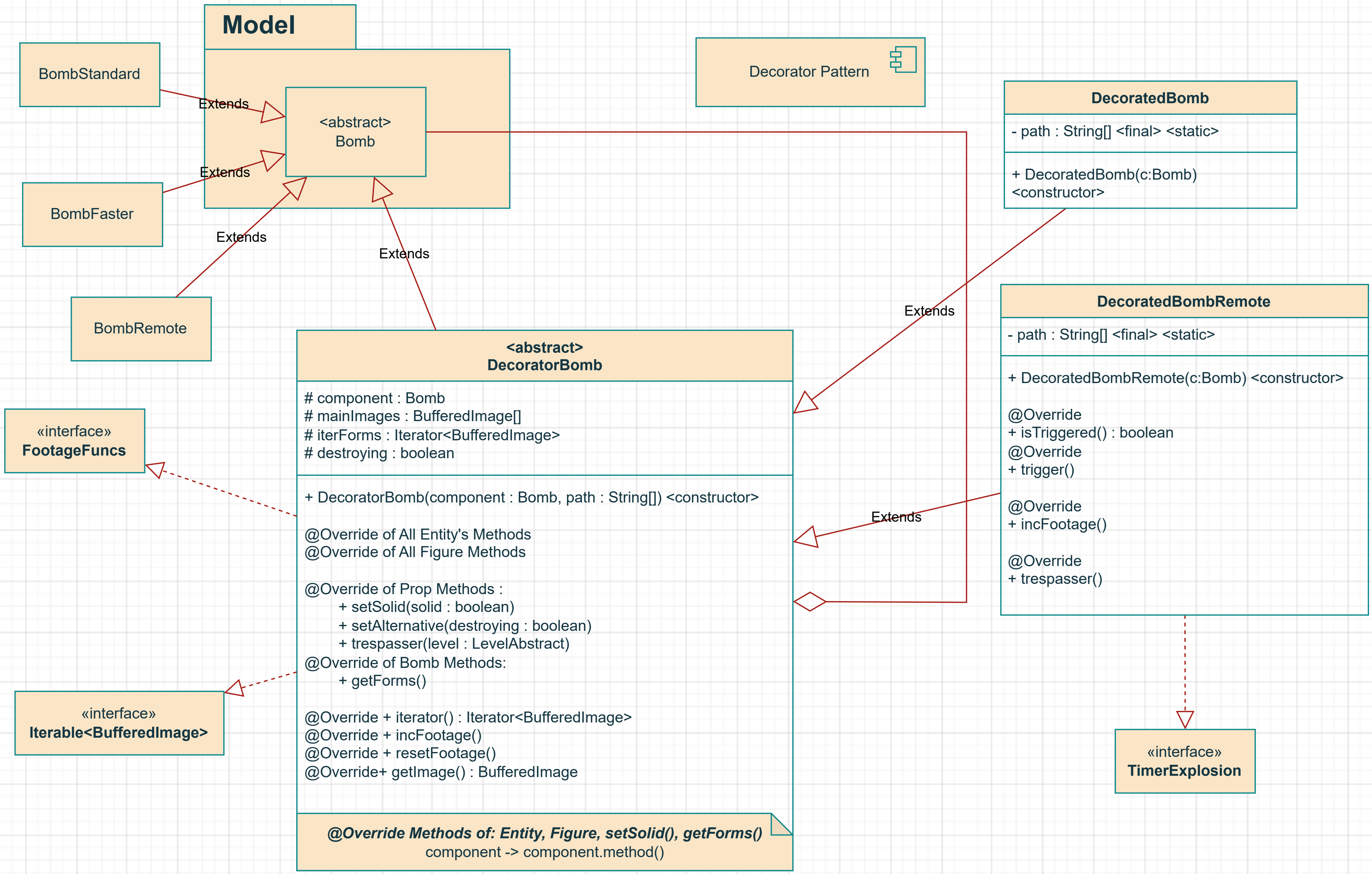
Arch.NORD_OVEST
Arch.MID_OVEST
Arch.SUD_OVEST
Arch.NORD_EST
Arch.MID_EST
Arch.SUD_EST
Arch.MID_NORD_LEFT
Arch.MID_NORD_RIGHT
Arch.MID_SUD_LEFT
Arch.MID_SUD_RIGHT

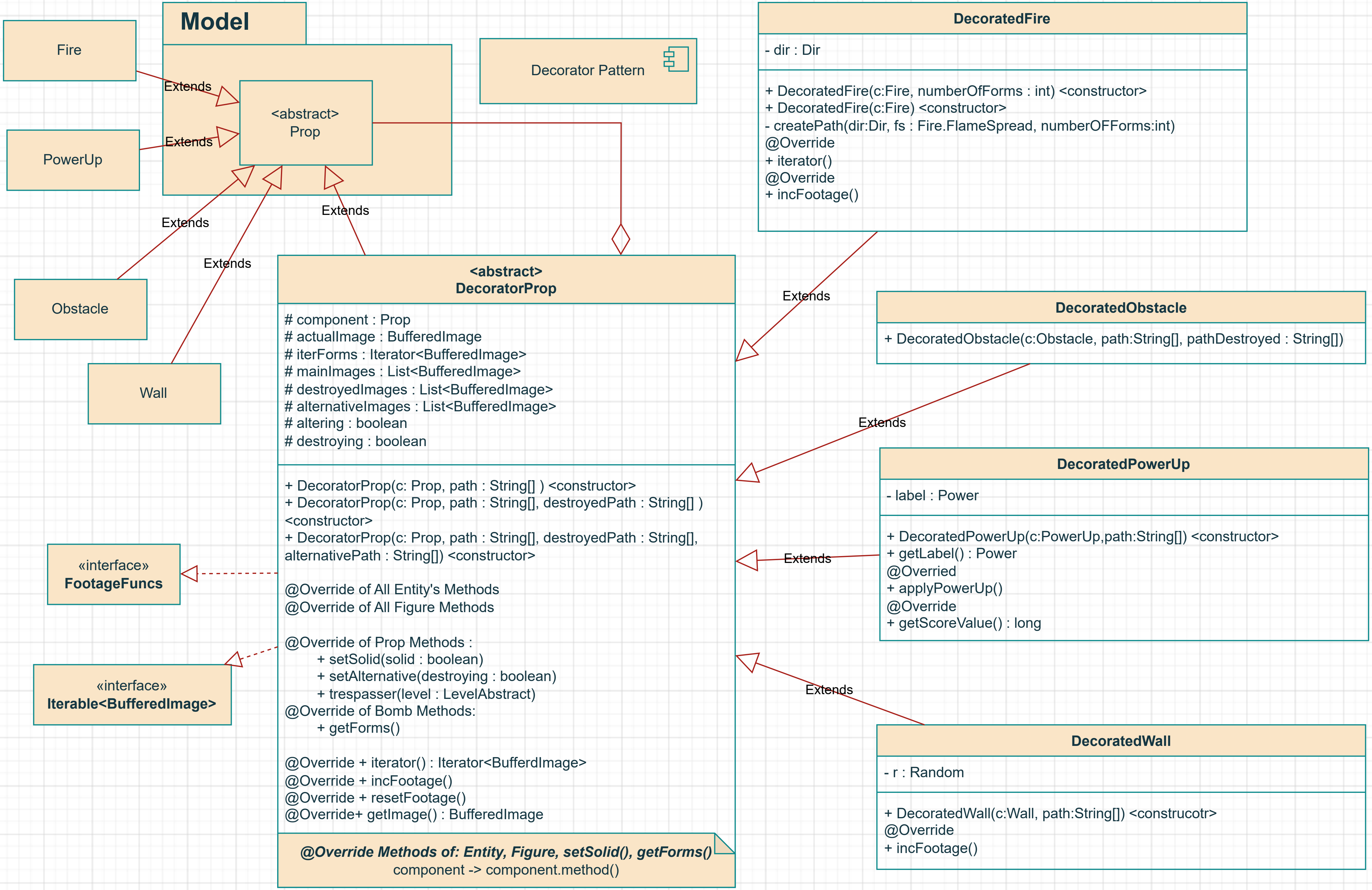
ArchesClassical

+ ArchesClassical() <constructor>

@Override
+ render(g2 : Graphics2D)


Model
<abstract>
TesseraAbstract
Decorator Pattern
Extends
Extends
TesseraConcrete
<abstract>
DecoratorTessera
# component : TesseraAbstract
# actualImage : BufferedImage
# iterForms : Iterator<BufferedImage>
# mainImages : List<BufferedImage>
# shadowedImages : List<BufferedImage>
# exitImages : List<BufferedImage>
# exitImage : BufferedImage
# altering : boolean
+ DecoratorTessera(component : TesseraAbstract) <constructor>
@Override of All Entity's Methods
@Override of All Figure Methods
@Override of All TesseraAbstract Methods
@Override + iterator() : Iterator<BufferdImage>
@Override + incFootage()
@Override + resetFootage()
@Override+ getImage() : BufferedImage
@Override Methods of: Entity, Figure, TesseraAbstract
component -> component.method()
DecoratedTessera
Extends
+ DecoratedTessera(c : TesseraAbstract,
path : String[], shadowPath : String[],
exitPath : String[])
«interface»
FootageFuncs
«interface»
Iterable<BufferedImage>


Model
BombStandard
BombFaster
BombRemote
<abstract>
Bomb
Extends
Extends
Extends
Extends
Decorator Pattern
DecoratedBomb
- path : String[] <final> <static>
+ DecoratedBomb(c:Bomb)
<constructor>
Extends
DecoratedBombRemote
- path : String[] <final> <static>
+ DecoratedBombRemote(c:Bomb) <constructor>
@Override
+ isTriggered() : boolean
@Override
+ trigger()
@Override
+ incFootage()
@Override
+ trespasser()
Extends
«interface»
TimerExplosion
«interface»
FootageFuncs
<abstract>
DecoratorBomb
# component : Bomb
# mainImages : BufferedImage[]
# iterForms : Iterator<BufferedImage>
# destroying : boolean
+ DecoratorBomb(component : Bomb, path : String[]) <constructor>
@Override of All Entity's Methods
@Override of All Figure Methods
@Override of Prop Methods :
+ setSolid(solid : boolean)
+ setAlternative(destroying : boolean)
+ trespasser(level : LevelAbstract)
@Override of Bomb Methods:
+ getForms()
@Override + iterator() : Iterator<BufferedImage>
@Override + incFootage()
@Override + resetFootage()
@Override+ getImage() : BufferedImage
@Override Methods of: Entity, Figure, setSolid(), getForms()
component -> component.method()
«interface»
Iterable<BufferedImage>



DecoratedFire

- dir : Dir

+ DecoratedFire(c:Fire, numberOfForms : int) <constructor>
+ DecoratedFire(c:Fire) <constructor>
- createPath(dir:Dir, fs : Fire.FlameSpread, numberOFForms:int)
@Override
+ iterator()
@Override
+ incFootage()

<abstract>
DecoratorProp

# component : Prop
# actualImage : BufferedImage
# iterForms : Iterator<BufferedImage>
# mainImages : List<BufferedImage>
# destroyedImages : List<BufferedImage>
# alternativeImages : List<BufferedImage>
# altering : boolean
# destroying : boolean

+ DecoratorProp(c: Prop, path : String[]) <constructor>
+ DecoratorProp(c: Prop, path : String[], destroyedPath : String[]) <constructor>
+ DecoratorProp(c: Prop, path : String[], destroyedPath : String[], alternativePath : String[]) <constructor>

@Override of All Entity's Methods
@Override of All Figure Methods

@Override of Prop Methods :
    + setSolid(solid : boolean)
    + setAlternative(destroying : boolean)
    + trespasser(level : LevelAbstract)
@Override of Bomb Methods:
    + getForms()

@Override + iterator() : Iterator<BufferdImage>
@Override + incFootage()
@Override + resetFootage()
@Override+ getImage() : BufferedImage

@Override Methods of: Entity, Figure, setSolid(), getForms()
component -> component.method()

DecoratedObstacle

+ DecoratedObstacle(c:Obstacle, path:String[], pathDestroyed : String[])

DecoratedPowerUp

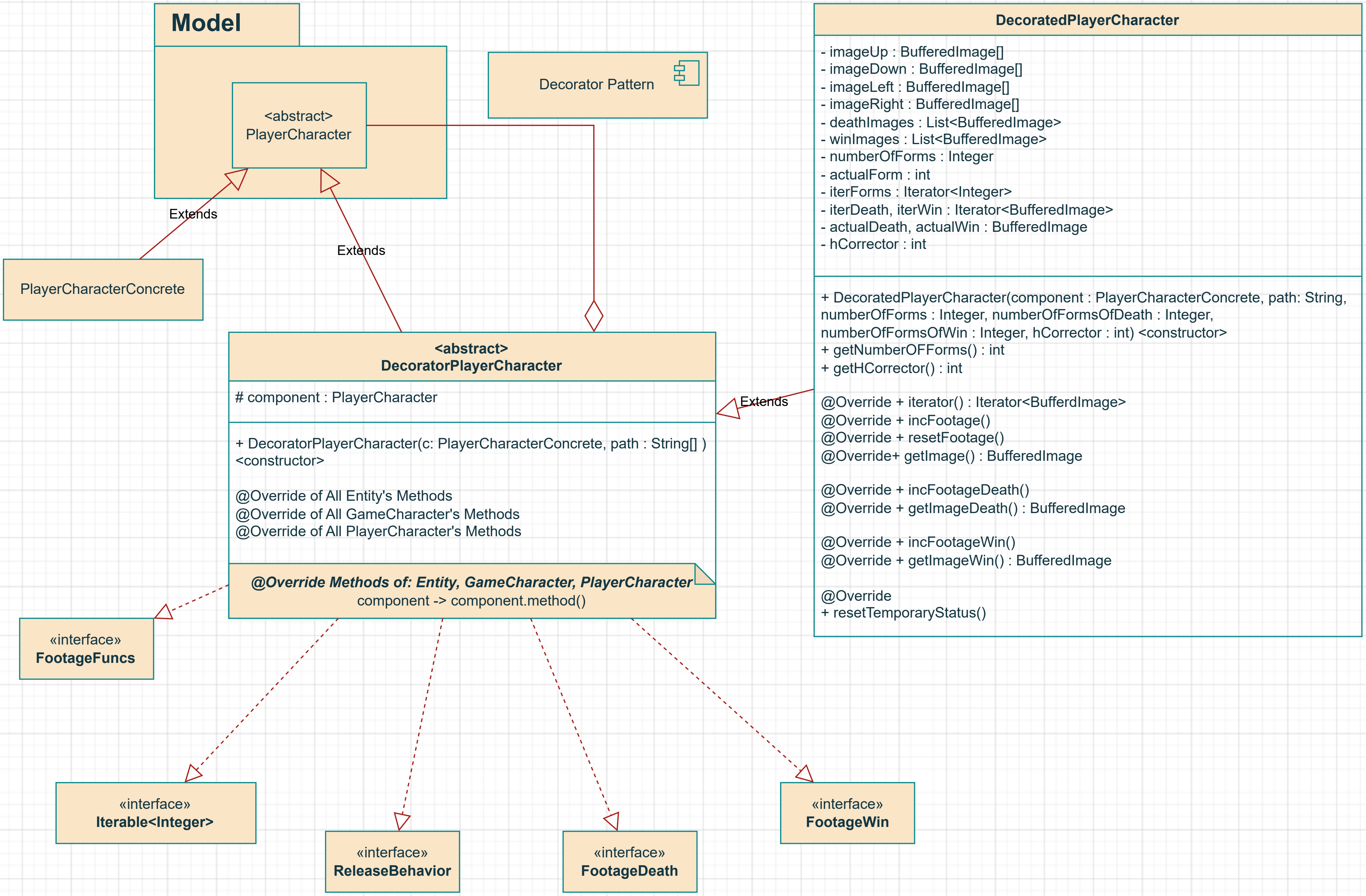
- label : Power

+ DecoratedPowerUp(c:PowerUp,path:String[]) <constructor>
+ getLabel() : Power
@Overried
+ applyPowerUp()
@Override
+ getScoreValue() : long

DecoratedWall

- r : Random

+ DecoratedWall(c:Wall, path:String[]) <construcotr>
@Override
+ incFootage()



<abstract>
PlayerCharacter

PlayerCharacterConcrete

DecoratedPlayerCharacter

- imageUp : BufferedImage[]
- imageDown : BufferedImage[]
- imageLeft : BufferedImage[]
- imageRight : BufferedImage[]
- deathImages : List<BufferedImage>
- winImages : List<BufferedImage>
- numberOfForms : Integer
- actualForm : int
- iterForms : Iterator<Integer>
- iterDeath, iterWin : Iterator<BufferedImage>
- actualDeath, actualWin : BufferedImage
- hCorrector : int

+ DecoratedPlayerCharacter(component : PlayerCharacterConcrete, path: String, numberOfForms : Integer, numberOfFormsOfDeath : Integer, numberOfFormsOfWin : Integer, hCorrector : int) <constructor>
+ getNumberOFForms() : int
+ getHCorrector() : int

@Override + iterator() : Iterator<BufferdImage>
@Override + incFootage()
@Override + resetFootage()
@Override+ getImage() : BufferedImage

@Override + incFootageDeath()
@Override + getImageDeath() : BufferedImage

@Override + incFootageWin()
@Override + getImageWin() : BufferedImage

@Override
+ resetTemporaryStatus()

<abstract>
DecoratorPlayerCharacter

# component : PlayerCharacter

+ DecoratorPlayerCharacter(c: PlayerCharacterConcrete, path : String[]) <constructor>

@Override of All Entity's Methods
@Override of All GameCharacter's Methods
@Override of All PlayerCharacter's Methods

@Override Methods of: Entity, GameCharacter, PlayerCharacter
component -> component.method()

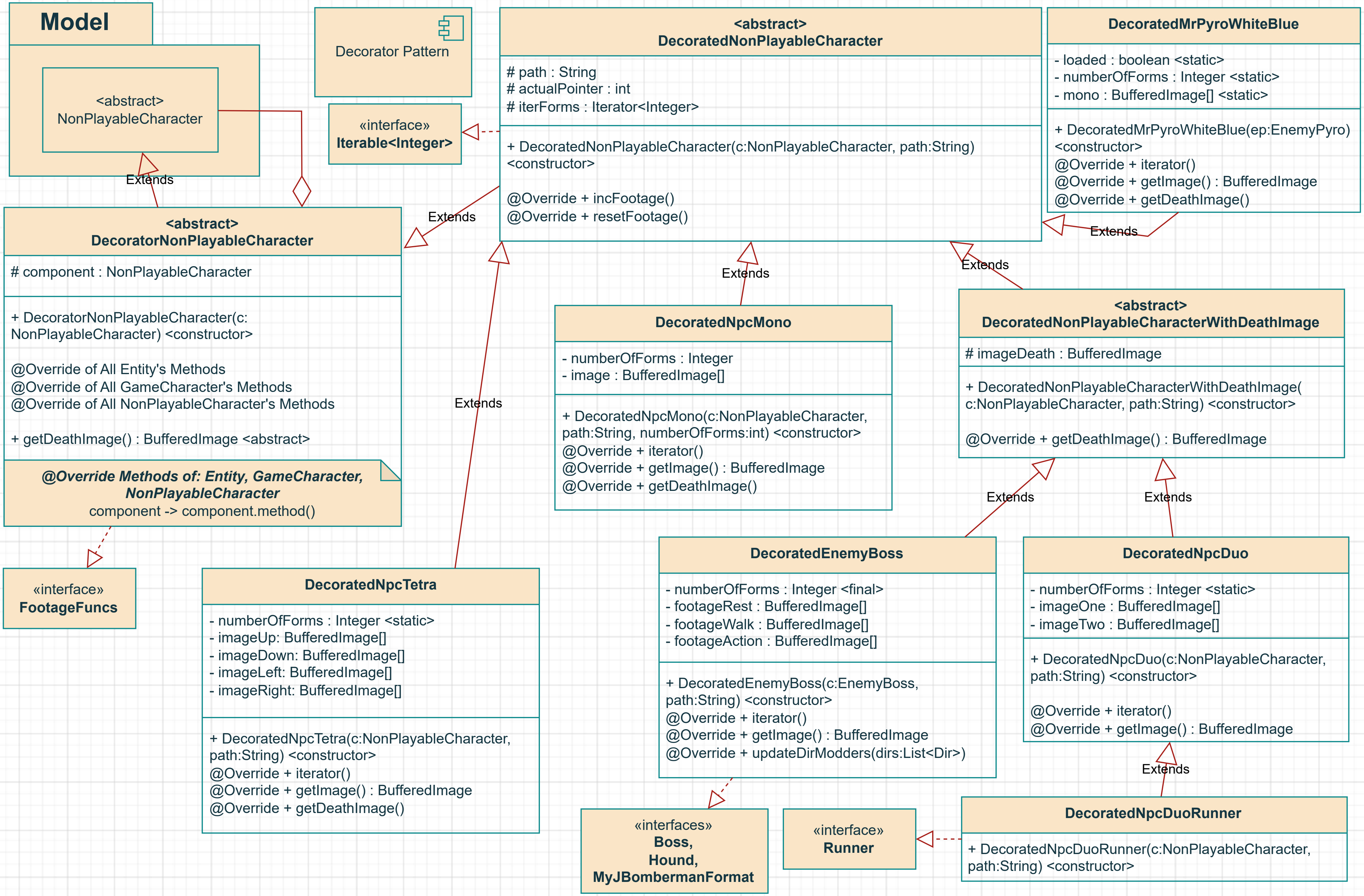
«interface»
FootageFuncs

«interface»
Iterable<Integer>

«interface»
ReleaseBehavior

«interface»
FootageDeath

«interface»
FootageWin



Model

<abstract>
NonPlayableCharacter

Decorator Pattern

«interface»
Iterable<Integer>

<abstract>
DecoratedNonPlayableCharacter

# path : String
# actualPointer : int
# iterForms : Iterator<Integer>

+ DecoratedNonPlayableCharacter(c:NonPlayableCharacter, path:String) <constructor>

@Override + incFootage()
@Override + resetFootage()

DecoratedMrPyroWhiteBlue

- loaded : boolean <static>
- numberOfForms : Integer <static>
- mono : BufferedImage[] <static>

+ DecoratedMrPyroWhiteBlue(ep:EnemyPyro) <constructor>
@Override + iterator()
@Override + getImage() : BufferedImage
@Override + getDeathImage()

<abstract>
DecoratorNonPlayableCharacter

# component : NonPlayableCharacter

+ DecoratorNonPlayableCharacter(c: NonPlayableCharacter) <constructor>

@Override of All Entity's Methods
@Override of All GameCharacter's Methods
@Override of All NonPlayableCharacter's Methods

+ getDeathImage() : BufferedImage <abstract>

@Override Methods of: Entity, GameCharacter, NonPlayableCharacter
component -> component.method()

«interface»
FootageFuncs

DecoratedNpcMono

- numberOfForms : Integer
- image : BufferedImage[]

+ DecoratedNpcMono(c:NonPlayableCharacter, path:String, numberOfForms:int) <constructor>
@Override + iterator()
@Override + getImage() : BufferedImage
@Override + getDeathImage()

<abstract>
DecoratedNonPlayableCharacterWithDeathImage

# imageDeath : BufferedImage

+ DecoratedNonPlayableCharacterWithDeathImage(c:NonPlayableCharacter, path:String) <constructor>

@Override + getDeathImage() : BufferedImage

DecoratedNpcTetra

- numberOfForms : Integer <static>
- imageUp: BufferedImage[]
- imageDown: BufferedImage[]
- imageLeft: BufferedImage[]
- imageRight: BufferedImage[]

+ DecoratedNpcTetra(c:NonPlayableCharacter, path:String) <constructor>
@Override + iterator()
@Override + getImage() : BufferedImage
@Override + getDeathImage()

DecoratedEnemyBoss

- numberOfForms : Integer <final>
- footageRest : BufferedImage[]
- footageWalk : BufferedImage[]
- footageAction : BufferedImage[]

+ DecoratedEnemyBoss(c:EnemyBoss, path:String) <constructor>
@Override + iterator()
@Override + getImage() : BufferedImage
@Override + updateDirModders(dirs:List<Dir>)

DecoratedNpcDuo

- numberOfForms : Integer <static>
- imageOne : BufferedImage[]
- imageTwo : BufferedImage[]

+ DecoratedNpcDuo(c:NonPlayableCharacter, path:String) <constructor>

@Override + iterator()
@Override + getImage() : BufferedImage

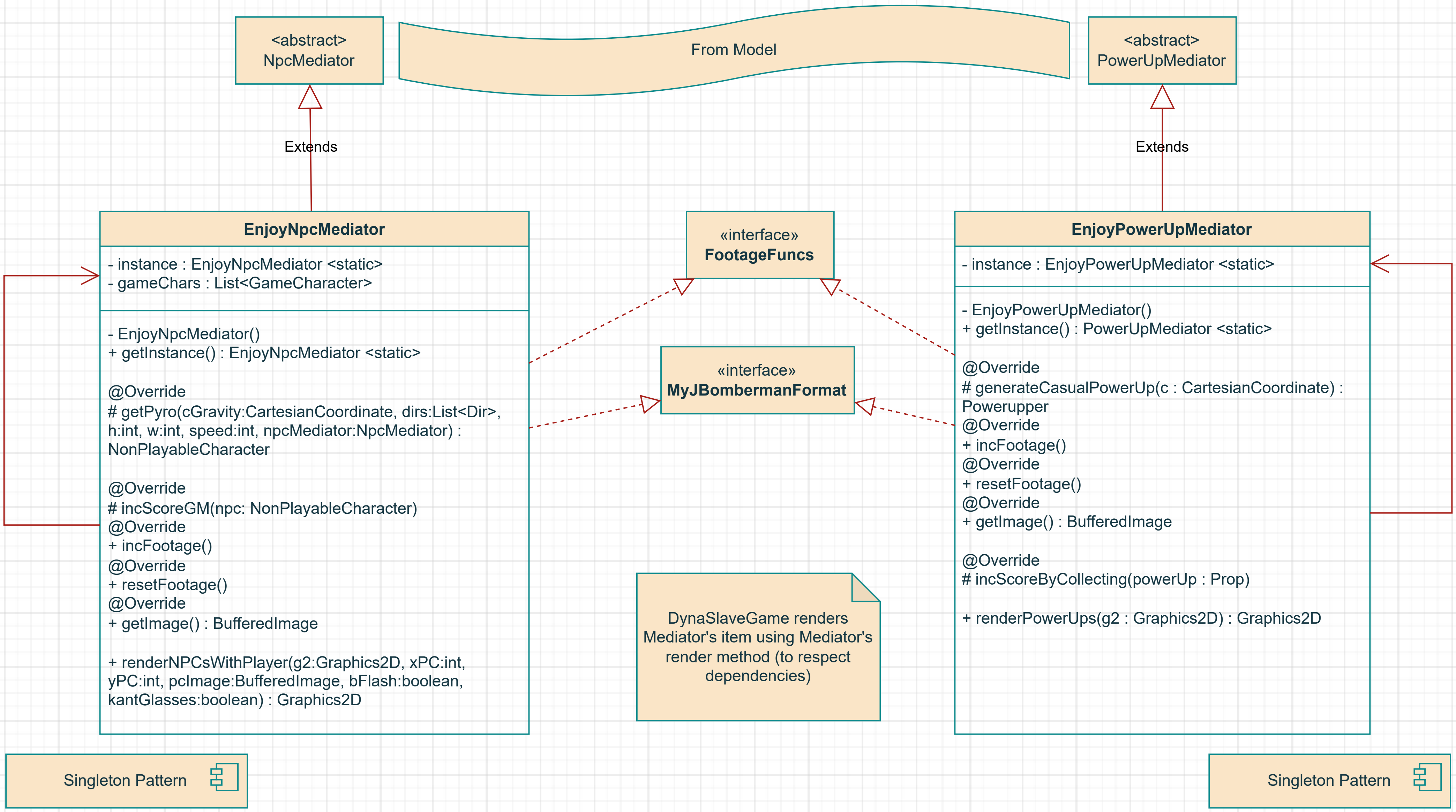
«interfaces»
Boss,
Hound,
MyJBombermanFormat

«interface»
Runner

DecoratedNpcDuoRunner

+ DecoratedNpcDuoRunner(c:NonPlayableCharacter, path:String) <constructor>

View UML: Concrete Mediators by View



<abstract>
NpcMediator

From Model

<abstract>
PowerUpMediator

Extends

Extends

EnjoyNpcMediator

- instance : EnjoyNpcMediator <static>
- gameChars : List<GameCharacter>

- EnjoyNpcMediator()
+ getInstance() : EnjoyNpcMediator <static>

@Override
# getPyro(cGravity:CartesianCoordinate, dirs:List<Dir>, h:int, w:int, speed:int, npcMediator:NpcMediator) : NonPlayableCharacter

@Override
# incScoreGM(npc: NonPlayableCharacter)
@Override
+ incFootage()
@Override
+ resetFootage()
@Override
+ getImage() : BufferedImage

+ renderNPCsWithPlayer(g2:Graphics2D, xPC:int, yPC:int, pcImage:BufferedImage, bFlash:boolean, kantGlasses:boolean) : Graphics2D

«interface»
FootageFuncs

«interface»
MyJBombermanFormat

DynaSlaveGame renders Mediator's item using Mediator's render method (to respect dependencies)

EnjoyPowerUpMediator

- instance : EnjoyPowerUpMediator <static>

- EnjoyPowerUpMediator()
+ getInstance() : PowerUpMediator <static>

@Override
# generateCasualPowerUp(c : CartesianCoordinate) : Powerupper
@Override
+ incFootage()
@Override
+ resetFootage()
@Override
+ getImage() : BufferedImage

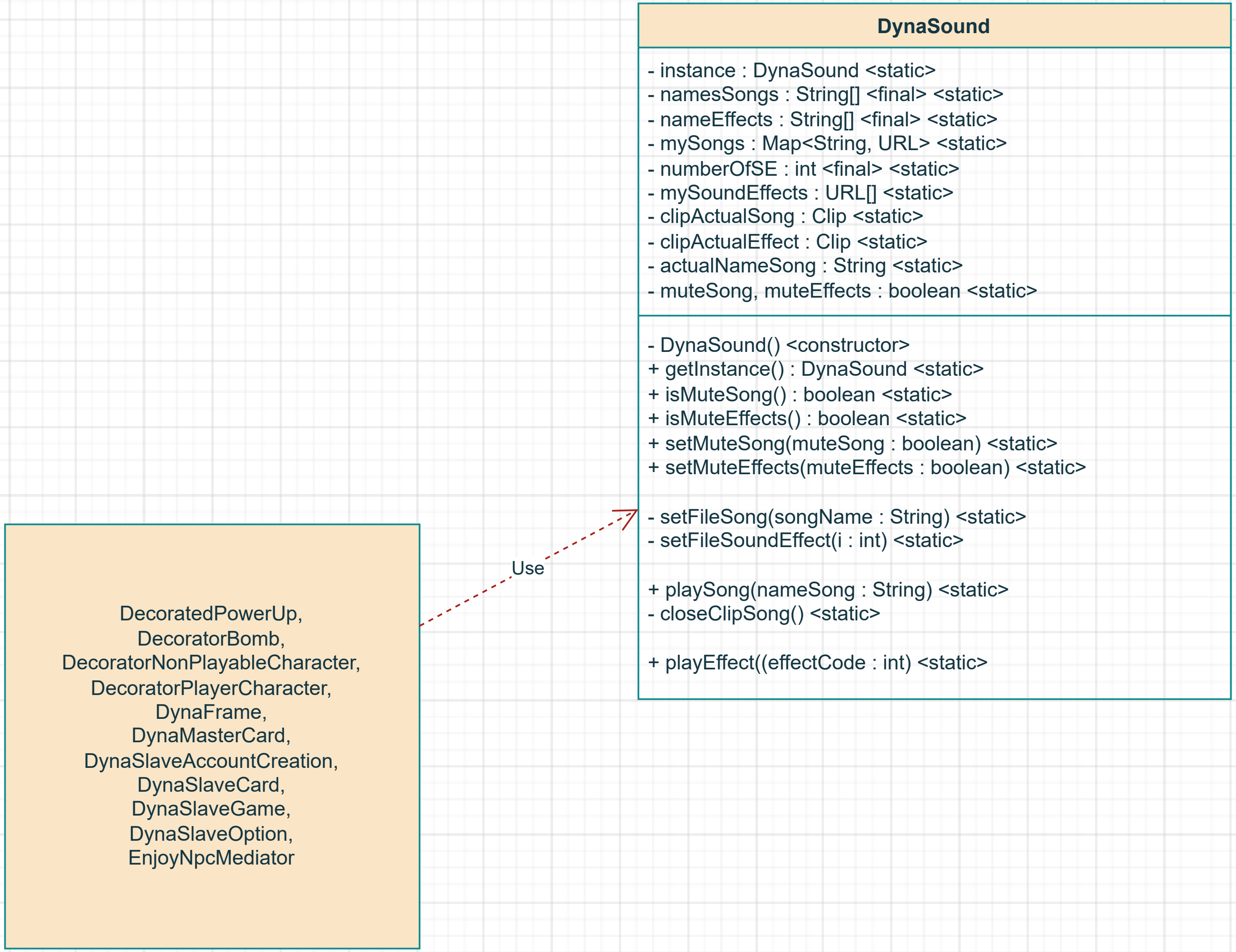
@Override
# incScoreByCollecting(powerUp : Prop)

+ renderPowerUps(g2 : Graphics2D) : Graphics2D

Singleton Pattern

Singleton Pattern

View UML: Music & Sound Effects



DynaSound

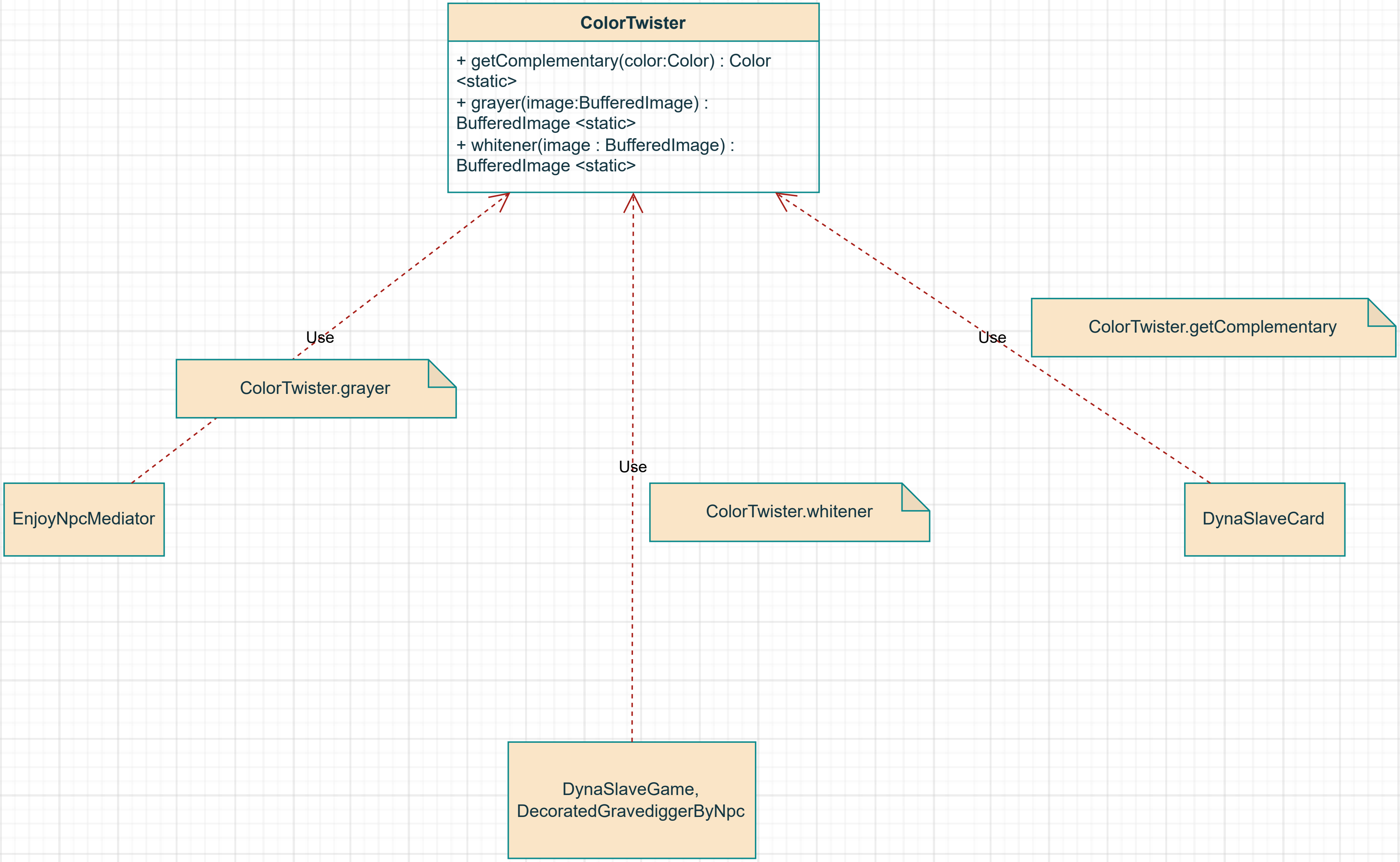
```
- instance : DynaSound <static>
- namesSongs : String[] <final> <static>
- nameEffects : String[] <final> <static>
- mySongs : Map<String, URL> <static>
- numberOfSE : int <final> <static>
- mySoundEffects : URL[] <static>
- clipActualSong : Clip <static>
- clipActualEffect : Clip <static>
- actualNameSong : String <static>
- muteSong, muteEffects : boolean <static>
```

```
- DynaSound() <constructor>
+ getInstance() : DynaSound <static>
+ isMuteSong() : boolean <static>
+ isMuteEffects() : boolean <static>
+ setMuteSong(muteSong : boolean) <static>
+ setMuteEffects(muteEffects : boolean) <static>

- setFileSong(songName : String) <static>
- setFileSoundEffect(i : int) <static>

+ playSong(nameSong : String) <static>
- closeClipSong() <static>

+ playEffect((effectCode : int) <static>
```

Use

DecoratedPowerUp,
DecoratorBomb,
DecoratorNonPlayableCharacter,
DecoratorPlayerCharacter,
DynaFrame,
DynaMasterCard,
DynaSlaveAccountCreation,
DynaSlaveCard,
DynaSlaveGame,
DynaSlaveOption,
EnjoyNpcMediator

View UML: Image Color Filter



ColorTwister

+ getComplementary(color:Color) : Color <static>
+ grayer(image:BufferedImage) : BufferedImage <static>
+ whitener(image : BufferedImage) : BufferedImage <static>

Use

ColorTwister.grayer

EnjoyNpcMediator

Use

ColorTwister.whitener

DynaSlaveGame, DecoratedGravediggerByNpc

Use

ColorTwister.getComplementary

DynaSlaveCard