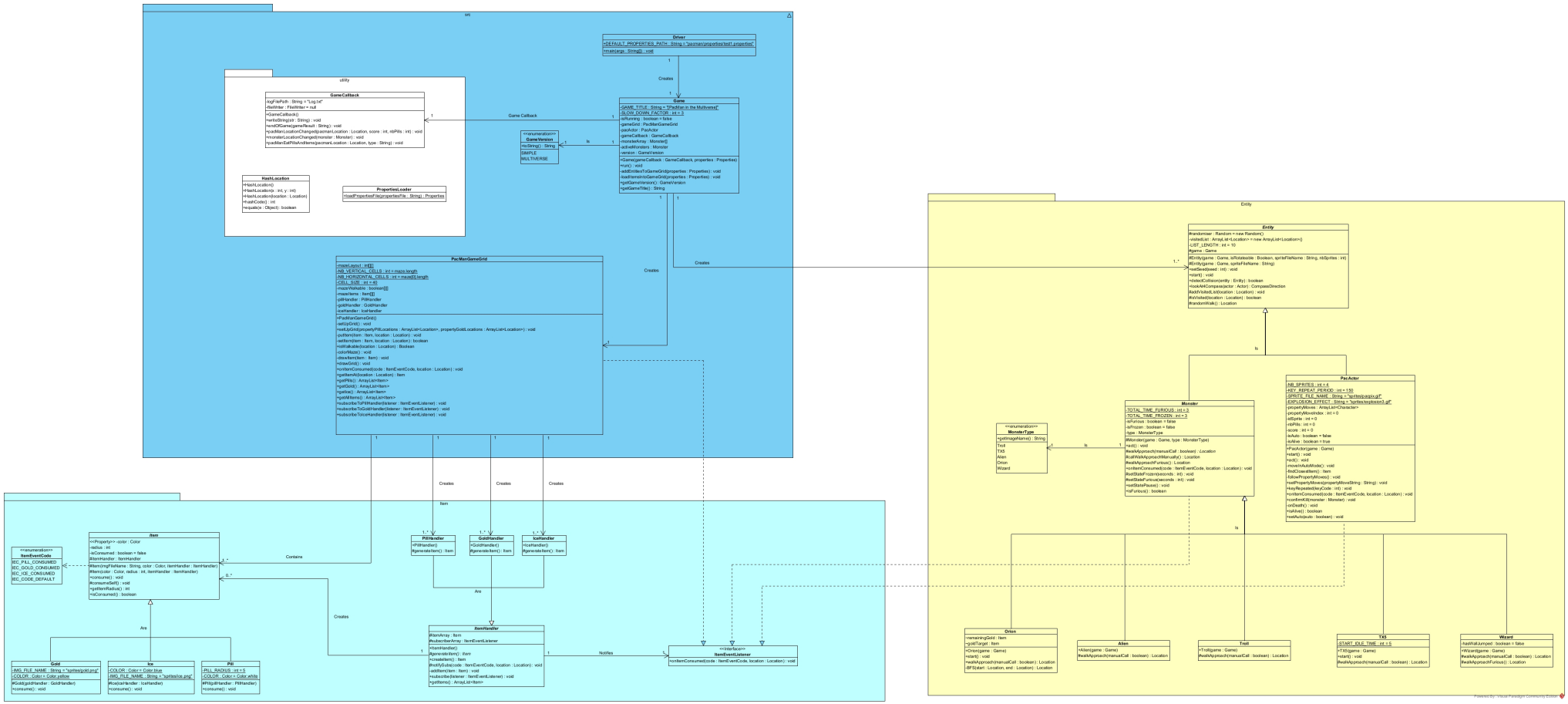
RPG
Driver
+DEFAULT_PROPERTIES_PATH : String = "pacman/properties/test1.properties"
+main(args : String[]) : void
Creates
utility
GameCallback
-logFilePath : String = "Log.txt"
-fileWriter : FileWriter = null
+GameCallback()
+writeString(str : String) : void
+endOfGame(gameResult : String) : void
+pacManLocationChanged(pacmanLocation : Location, score : int, nbPills : int) : void
+monsterLocationChanged(monster : Monster) : void
+pacManEatPillsAndItems(pacmanLocation : Location, type : String) : void
Game Callback
Game
-GAME_TITLE : String = "PacMan in the Multiverse!"
-SLOW_DOWN_FACTOR : int = 3
-isRunning : boolean = false
-gameGrid : PacManGameGrid
-pacActor : PacActor
-gameCallback : GameCallback
-monsterArray : Monster[]
-activeMonsters : Monster
-version : GameVersion
+Game(gameCallback : GameCallback, properties : Properties)
+run() : void
-addEntitiesToGameGrid(properties : Properties) : void
-loadItemsIntoGameGrid(properties : Properties) : void
+getGameVersion() : GameVersion
+getGameTitle() : String
Is
<<enumeration>>
GameVersion
+toString() : String
SIMPLE
MULTIVERSE
HashLocation
+HashLocation()
+HashLocation(x : int, y : int)
+HashLocation(location : Location)
+hashCode() : int
+equals(x : Object) : boolean
PropertiesLoader
+loadPropertiesFile(propertiesFile : String) : Properties
Creates
PacManGameGrid
-mazeLayout : int[][]
-NB_VERTICAL_CELLS : int = maze.length
-NB_HORIZONTAL_CELLS : int = maze[0].length
-CELL_SIZE : int = 40
-mazeWalkable : boolean[][]
-mazeItems : Item[][]
-pillHandler : PillHandler
-goldHandler : GoldHandler
-iceHandler : IceHandler
+PacManGameGrid()
-setUpGrid() : void
-setUpGrid(propertyPillLocations : ArrayList<Location>, propertyGoldLocations : ArrayList<Location>) : void
-putItem(item : Item, location : Location) : void
-setItem(item : Item, location : Location) : boolean
+isWalkable(location : Location) : Boolean
-colorMaze() : void
-drawItem(item : Item) : void
+drawGrid() : void
+onItemConsumed(code : ItemEventCode, location : Location) : void
+getItemAt(location : Location) : Item
+getPills() : ArrayList<Item>
+getGold() : ArrayList<Item>
+getIce() : ArrayList<Item>
+getAllItems() : ArrayList<Item>
+subscribeToPillHandler(listener : ItemEventListener) : void
+subscribeToGoldHandler(listener : ItemEventListener) : void
+subscribeToIceHandler(listener : ItemEventListener) : void
Creates
Creates
Creates
Item
<<enumeration>>
ItemEventCode
IEC_PILL_CONSUMED
IEC_GOLD_CONSUMED
IEC_ICE_CONSUMED
IEC_CODE_DEFAULT
Item
<<Property>> -color : Color
-radius : int
-isConsumed : boolean = false
#itemHandler : ItemHandler
#item(imgFileName : String, color : Color, itemHandler : ItemHandler)
#item(color : Color, radius : int, itemHandler : ItemHandler)
+consume() : void
#consumeSelf() : void
+getItemRadius() : int
+isConsumed() : boolean
Contains
1..*
0..*
1..*
PillHandler
+PillHandler()
#generateItem() : Item
GoldHandler
+GoldHandler()
#generateItem() : Item
IceHandler
+IceHandler()
#generateItem() : Item
Are
Are
Creates
Gold
-IMG_FILE_NAME : String = "sprites/gold.png"
-COLOR : Color = Color.yellow
#Gold(goldHandler : GoldHandler)
+consume() : void
Ice
-COLOR : Color = Color.blue
-IMG_FILE_NAME : String = "sprites/ice.png"
#Ice(iceHandler : IceHandler)
+consume() : void
Pill
-PILL_RADIUS : int = 5
-COLOR : Color = Color.white
#Pill(pillHandler : PillHandler)
+consume() : void
ItemHandler
#Item Array : Item
#subscriberArray : ItemEventListener
+ItemHandler()
#generateItem() : Item
+createItem() : Item
#notify(code : ItemEventCode, location : Location) : void
-addItem(item : Item) : void
+subscribe(listener : ItemEventListener) : void
+getItems() : ArrayList<Item>
1
Notifies
1..*
<<Interface>>
ItemEventListener
+onItemConsumed(code : ItemEventCode, location : Location) : void
Entity
Entity
#randomiser : Random = new Random()
-visitedList : ArrayList<Location> = new ArrayList<Location>()
-LIST_LENGTH : int = 10
#game : Game
#Entity(game : Game, isRotatable : Boolean, spriteFileName : String, nbSprites : int)
#Entity(game : Game, spriteFileName : String)
+setSpeed(speed : int) : void
+start() : void
+detectCollision(entity : Entity) : boolean
+lookAtCompass(actor : Actor) : CompassDirection
#addVisitedList(location : Location) :void
#isVisited(location : Location) : boolean
#randomWalk() : Location
1..*
Is
PacActor
-NB_SPRITES : int = 4
-KEY_REPEAT_PERIOD : int = 150
-SPRITE_FILE_NAME : String = "sprites/pacpix.gif"
-EXPLOSION_EFFECT : String = "sprites/explosion3.gif"
-propertyMoves : ArrayList<Character>
-propertyMoveIndex : int = 0
-initSprite : int = 0
-initPills : int = 0
-score : int = 0
-isAuto : boolean = false
-isAlive : boolean = true
+PacActor(game : Game)
+start() : void
+act() : void
-moveInAutoMode() : void
-findClosestItem() : Item
-FollowPropertyMoves() : void
+setPropertyMoves(propertyMoveString : String) : void
+keyRepeated(keyCode : int) : void
+onItemConsumed(code : ItemEventCode, location : Location) : void
+confirmKill(monster : Monster) : void
+onDeath() : void
+isAlive() : boolean
+setAutopilot(b : boolean) : void
Monster
-TOTAL_TIME_FURIOUS : int = 3
-TOTAL_TIME_FROZEN : int = 3
-isFurious : boolean = false
-isFrozen : boolean = false
-type : MonsterType
#Monster(game : Game, type : MonsterType)
#act() : void
#walkApproach(manualCall : boolean) : Location
#calWalkApproachManually() : Location
#walkApproachFurious() : Location
+onItemConsumed(code : ItemEventCode, location : Location) : void
#setStateFrozen(seconds : int) : void
#setStateFurious(seconds : int) : void
+setStatePassive() : void
+isFurious() : boolean
<<enumeration>>
MonsterType
+getImageName() : String
Troll
TXS
Alien
Orion
Wizard
1..*
Is
1..*
Is
Orion
+remainingGrid : Item
+getTarget : Item
+Orion(game : Game)
+start() : void
+walkApproach(manualCall : boolean) : Location
-BFS(start : Location, end : Location) : Location
Alien
+Alien(game : Game)
#walkApproach(manualCall : boolean) : Location
Troll
+Troll(game : Game)
#walkApproach(manualCall : boolean) : Location
TXS
-START_IDLE_TIME : int = 5
+TXS(game : Game)
+start() : void
#walkApproach(manualCall : boolean) : Location
Wizard
-hasWallJumped : boolean = false
#Wizard(game : Game)
#walkApproach(manualCall : boolean) : Location
#walkApproachFurious() : Location
Powered by Visual Paradigm Community Edition