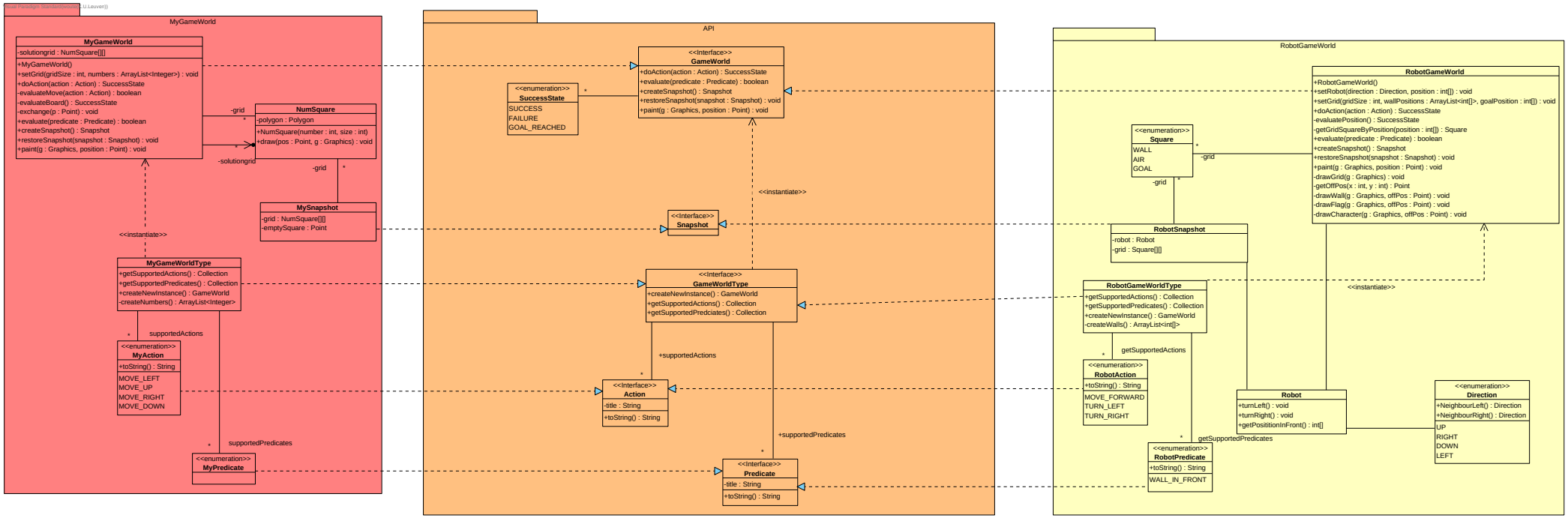
Visual Paradigm Standard(xvoutte(K.U.Leuven))
MyGameWorld
MyGameWorld
-solutiongrid : NumSquare[][]
+MyGameWorld()
+setGrid(gridSize : int, numbers : ArrayList<Integer>) : void
+doAction(action : Action) : SuccessState
-evaluateMove(action : Action) : boolean
-evaluateBoard() : SuccessState
-exchange(p : Point) : void
+evaluate(predicate : Predicate) : boolean
+createSnapshot() : Snapshot
+restoreSnapshot(snapshot : Snapshot) : void
+paint(g : Graphics, position : Point) : void
-grid
-solutiongrid
NumSquare
-polygon : Polygon
+NumSquare(number : int, size : int)
+draw(pos : Point, g : Graphics) : void
-grid
MySnapshot
-grid : NumSquare[][]
-emptySquare : Point
<<instantiate>>
MyGameWorldType
+getSupportedActions() : Collection
+getSupportedPredicates() : Collection
+createNewInstance() : GameWorld
-createNumbers() : ArrayList<Integer>
supportedActions
<<enumeration>>
MyAction
+toString() : String
MOVE_LEFT
MOVE_UP
MOVE_RIGHT
MOVE_DOWN
supportedPredicates
<<enumeration>>
MyPredicate
API
<<Interface>>
GameWorld
+doAction(action : Action) : SuccessState
+evaluate(predicate : Predicate) : boolean
+createSnapshot() : Snapshot
+restoreSnapshot(snapshot : Snapshot) : void
+paint(g : Graphics, position : Point) : void
<<enumeration>>
SuccessState
SUCCESS
FAILURE
GOAL_REACHED
<<instantiate>>
<<Interface>>
Snapshot
<<Interface>>
GameWorldType
+createNewInstance() : GameWorld
+getSupportedActions() : Collection
+getSupportedPredciates() : Collection
+supportedActions
<<Interface>>
Action
-title : String
+toString() : String
+supportedPredicates
<<Interface>>
Predicate
-title : String
+toString() : String
RobotGameWorld
RobotGameWorld
+RobotGameWorld()
+setRobot(direction : Direction, position : int[]) : void
+setGrid(gridSize : int, wallPositions : ArrayList<int[]>, goalPosition : int[]) : void
+doAction(action : Action) : SuccessState
-evaluatePosition() : SuccessState
-getGridSquareByPosition(position : int[]) : Square
+evaluate(predicate : Predicate) : boolean
+createSnapshot() : Snapshot
+restoreSnapshot(snapshot : Snapshot) : void
+paint(g : Graphics, position : Point) : void
-drawGrid(g : Graphics) : void
-getOffPos(x : int, y : int) : Point
-drawWall(g : Graphics, offPos : Point) : void
-drawFlag(g : Graphics, offPos : Point) : void
-drawCharacter(g : Graphics, offPos : Point) : void
<<enumeration>>
Square
WALL
AIR
GOAL
-grid
-grid
RobotSnapshot
-robot : Robot
-grid : Square[][]
<<instantiate>>
RobotGameWorldType
+getSupportedActions() : Collection
+getSupportedPredicates() : Collection
+createNewInstance() : GameWorld
-createWalls() : ArrayList<int[]>
getSupportedActions
<<enumeration>>
RobotAction
+toString() : String
MOVE_FORWARD
TURN_LEFT
TURN_RIGHT
Robot
+turnLeft() : void
+turnRight() : void
+getPosititionInFront() : int[]
<<enumeration>>
Direction
+NeighbourLeft() : Direction
+NeighbourRight() : Direction
UP
RIGHT
DOWN
LEFT
getSupportedPredicates
<<enumeration>>
RobotPredicate
+toString() : String
WALL_IN_FRONT