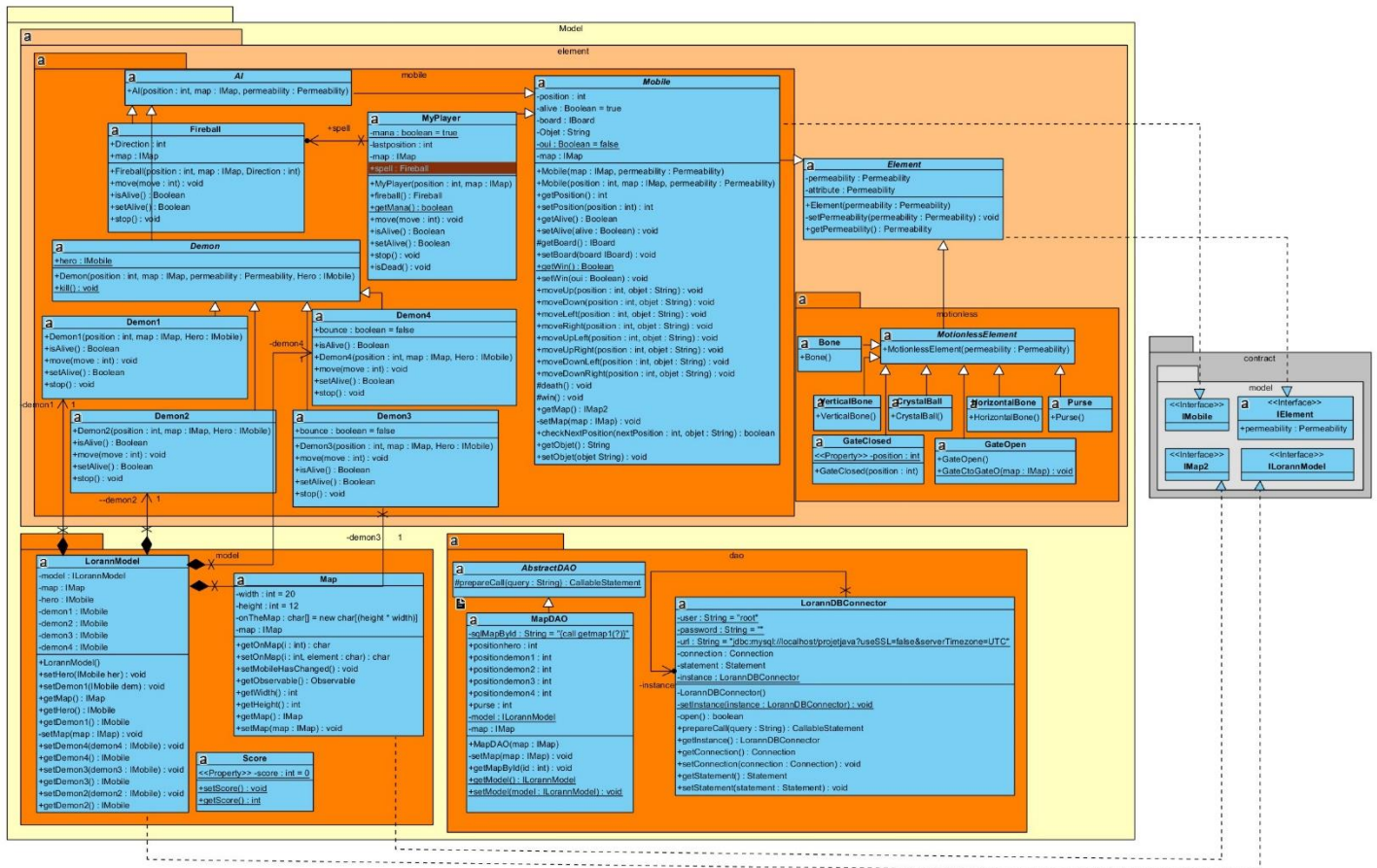# Project deliverable Class Diagram
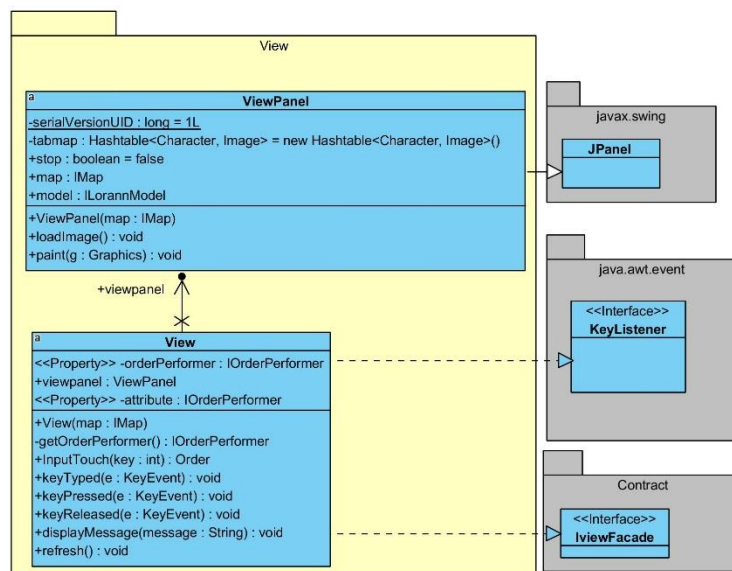
HANZO Claire, CASTELLANI Jean-Baptiste, DIDIER Léo, MOLA Axel| 05/06/2018

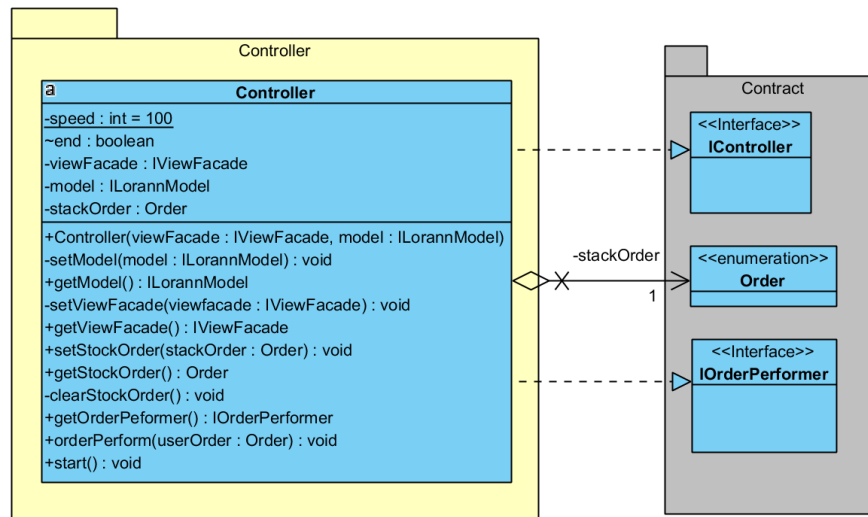Here are 5 class diagrams representing the code of our game, and respecting the DP MVC.
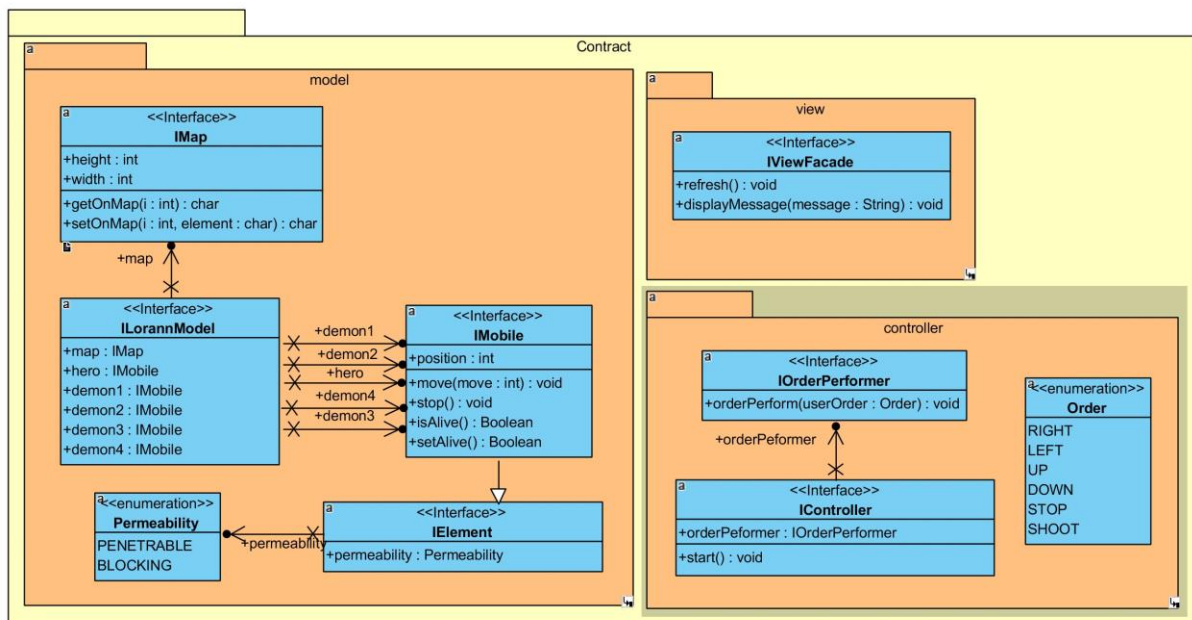
Class Diagram Model

Model

element

mobile

AI
+AI(position : int, map : IMap, permeability : Permeability)

Mobile
-position : int
-alive : Boolean = true
-board : IBoard
-Objet : String
-oui : Boolean = false
-map : IMap
+Mobile(map : IMap, permeability : Permeability)
+Mobile(position : int, map : IMap, permeability : Permeability)
+getPosition() : int
+setPosition(position : int) : int
+getAlive() : Boolean
+setAlive(alive : Boolean) : void
#getBoard() : IBoard
+setBoard(board : IBoard) : void
+getWin() : Boolean
+setWin(oui : Boolean) : void
+moveUp(position : int, objet : String) : void
+moveDown(position : int, objet : String) : void
+moveLeft(position : int, objet : String) : void
+moveRight(position : int, objet : String) : void
+moveUpLeft(position : int, objet : String) : void
+moveUpRight(position : int, objet : String) : void
+moveDownLeft(position : int, objet : String) : void
+moveDownRight(position : int, objet : String) : void
#death() : void
#win() : void
+getMap() : IMap2
+setMap(map : IMap) : void
+checkNextPosition(nextPosition : int, objet : String) : boolean
+getObjet() : String
+setObjet(objet : String) : void

Fireball
+Direction : int
+map : IMap
+spell : Fireball
+Fireball(position : int, map : IMap, Direction : int)
+move(move : int) : void
+isAlive() : Boolean
+setAlive() : Boolean
+stop() : void

MyPlayer
-mana : boolean = true
-lastposition : int
-map : IMap
+spell : Fireball
+MyPlayer(position : int, map : IMap)
+fireball() : Fireball
+getMana() : boolean
+move(move : int) : void
+isAlive() : Boolean
+setAlive() : Boolean
+stop() : void
+isDead() : Boolean

Element
+permeability : Permeability
+attribute : Permeability
+Element(permeability : Permeability)
+setPermeability(permeability : Permeability) : void
+getPermeability() : Permeability

Demon
+hero : IMobile
+Demon(position : int, map : IMap, permeability : Permeability, Hero : IMobile)
+kill() : void

motionless

Bone
+Bone()

MotionlessElement
+MotionlessElement(permeability : Permeability)

Demon1
+Demon1(position : int, map : IMap, Hero : IMobile)
+isAlive() : Boolean
+move(move : int) : void
+setAlive() : Boolean
+stop() : void

Demon4
+bounce : boolean = false
+isAlive() : Boolean
+Demon4(position : int, map : IMap, Hero : IMobile)
+move(move : int) : void
+setAlive() : Boolean
+stop() : void

VerticalBone
+VerticalBone()

CrystalBall
+CrystalBall()

HorizontalBone
+HorizontalBone()

Purse
+Purse()

Demon2
+Demon2(position : int, map : IMap, Hero : IMobile)
+isAlive() : Boolean
+move(move : int) : void
+setAlive() : Boolean
+stop() : void

Demon3
+bounce : boolean = false
+Demon3(position : int, map : IMap, Hero : IMobile)
+move(move : int) : void
+isAlive() : Boolean
+setAlive() : Boolean
+stop() : void

GateClosed
<<Property>> -position : int
+GateClosed(position : int)

GateOpen
+GateOpen()
+GateCtoGateO(map : IMap) : void

contract

model

<<Interface>> **IMobile**

<<Interface>> **IElement**
+permeability : Permeability

<<Interface>> **IMap2**

<<Interface>> **ILorannModel**

LorannModel
-model : ILorannModel
-map : IMap
-hero : IMobile
-demon1 : IMobile
-demon2 : IMobile
-demon3 : IMobile
-demon4 : IMobile
+LorannModel()
+setHero(Mobile her) : void
+setDemon1(Mobile dem) : void
+getMap() : IMap
+getHero() : IMobile
+getDemon1() : IMobile
+setMap(map : IMap) : void
+setDemon4(demon4 : IMobile) : void
+getDemon4() :
+setDemon3(demon3 : IMobile) : void
+getDemon3() : IMobile
+setDemon2(demon2 : IMobile) : void
+getDemon2() : IMobile

Map
-width : int = 20
-height : int = 12
-onTheMap : char[] = new char[(height * width)]
-map : IMap
+getOnMap(i : int) : char
+setOnMap(i : int, element : char) : char
+setMobileHasChanged() : void
+getObservable() : Observable
+getWidth() : int
+getHeight() : int
+getMap() : IMap
+setMap(map : IMap) : void

Score
<<Property>> -score : int = 0
+setScore() : void
+getScore() : int

dao

AbstractDAO
#prepareCall(query : String) : CallableStatement

MapDAO
-sqlMapById : String = "(call getmap1(?))"
+positionhero : int
+positiondemon1 : int
+positiondemon2 : int
+positiondemon3 : int
+positiondemon4 : int
+purse : int
-model : ILorannModel
-map : IMap
+MapDAO(map : IMap)
+setMap(map : IMap) : void
+getMapById(id : int) : void
+getModel() : ILorannModel
+setModel(model : ILorannModel) : void

LorannDBConnector
-user : String = "root"
-password : String = ""
-url : String = "jdbc:mysql://localhost/projetjava?useSSL=false&serverTimezone=UTC"
-connection : Connection
-statement : Statement
-instance : LorannDBConnector
-LorannDBConnector()
-setInstance(instance : LorannDBConnector) : void
-open() : boolean
+prepareCall(query : String) : CallableStatement
+getInstance() : LorannDBConnector
+getConnection() : Connection
+setConnection(connection : Connection) : void
+getStatement() : Statement
+setStatement(statement : Statement) : void

Class Diagram View

View

ViewPanel
-serialVersionUID : long = 1L
-tabmap : Hashtable<Character, Image> = new Hashtable<Character, Image>()
+stop : boolean = false
+map : IMap
+model : ILorannModel
+ViewPanel(map : IMap)
+loadImage() : void
+paint(g : Graphics) : void

javax.swing
JPanel

View
<<Property>> -orderPerformer : IOrderPerformer
+viewpanel : ViewPanel
<<Property>> -attribute : IOrderPerformer
+View(map : IMap)
-getOrderPerformer() : IOrderPerformer
+InputTouch(key : int) : Order
+keyTyped(e : KeyEvent) : void
+keyPressed(e : KeyEvent) : void
+keyReleased(e : KeyEvent) : void
+displayMessage(message : String) : void
+refresh() : void

java.awt.event
<<Interface>> **KeyListener**

Contract
<<Interface>> **IviewFacade**

Class Diagram Controller

Class Diagram Contract

Class Diagram Main