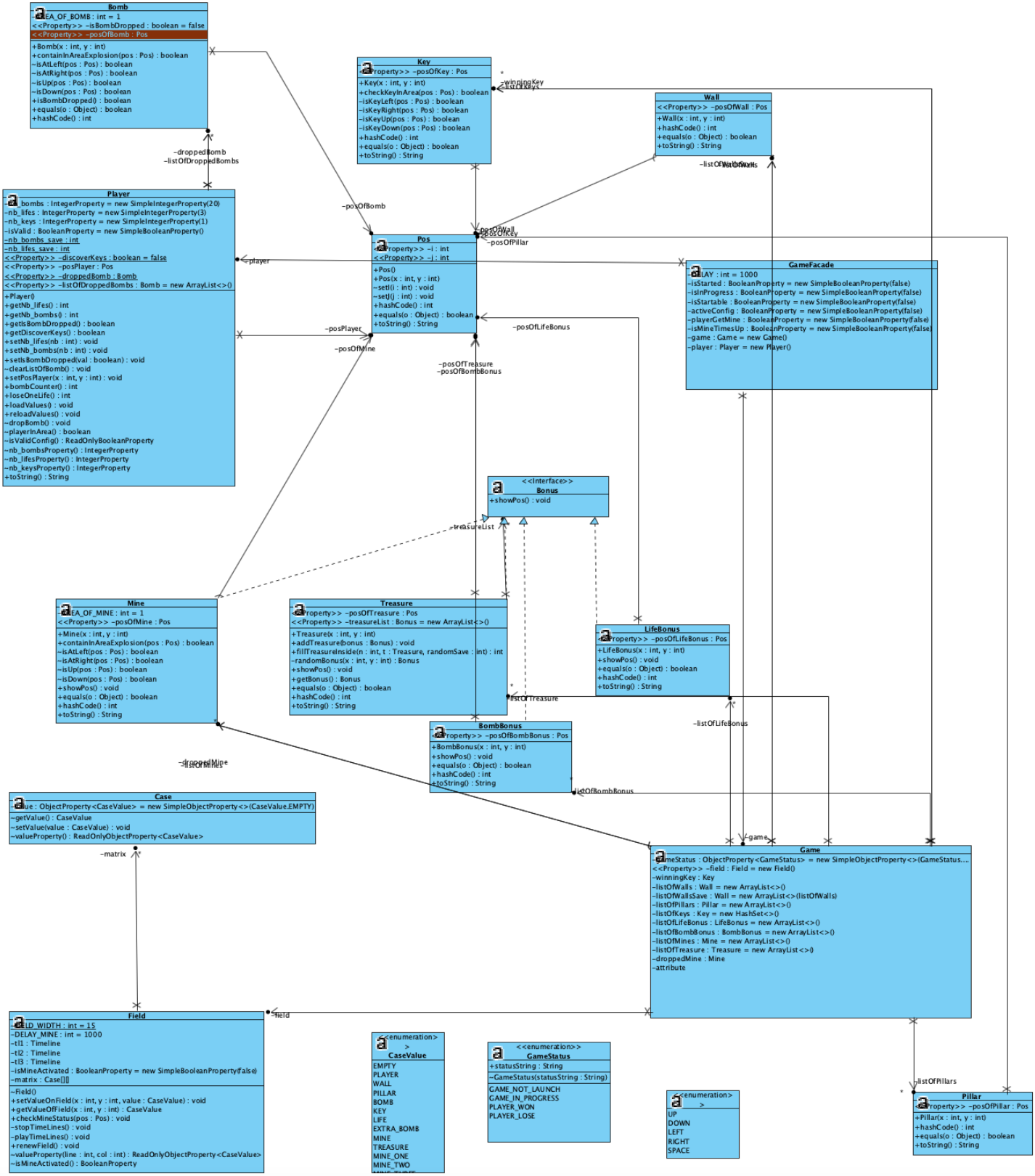
Bomb
AREA_OF_BOMB : int = 1
<<Property>> -isBombDropped : boolean = false
<<Property>> -posOfBomb : Pos
+Bomb(x : int, y : int)
+containInAreaExplosion(pos : Pos) : boolean
~isAtLeft(pos : Pos) : boolean
~isAtRight(pos : Pos) : boolean
~isUp(pos : Pos) : boolean
~isDown(pos : Pos) : boolean
+isBombDropped() : boolean
+equals(o : Object) : boolean
+hashCode() : int
Key
<<Property>> -posOfKey : Pos
+Key(x : int, y : int)
+checkKeyInArea(pos : Pos) : boolean
~isKeyLeft(pos : Pos) : boolean
~isKeyRight(pos : Pos) : boolean
~isKeyUp(pos : Pos) : boolean
~isKeyDown(pos : Pos) : boolean
+hashCode() : int
+equals(o : Object) : boolean
+toString() : String
Wall
<<Property>> -posOfWall : Pos
+Wall(x : int, y : int)
+hashCode() : int
+equals(o : Object) : boolean
+toString() : String
Player
_bombs : IntegerProperty = new SimpleIntegerProperty(20)
-nb_lifes : IntegerProperty = new SimpleIntegerProperty(3)
-nb_keys : IntegerProperty = new SimpleIntegerProperty(1)
-isValid : BooleanProperty = new SimpleBooleanProperty()
-nb_bombs_save : int
-nb_lifes_save : int
<<Property>> -discoverKeys : boolean = false
<<Property>> -posPlayer : Pos
<<Property>> -droppedBomb : Bomb
<<Property>> -listOfDroppedBombs : Bomb = new ArrayList<>()
+Player()
+getNb_lifes() : int
+getNb_bombs() : int
+getIsBombDropped() : boolean
+getDiscoverKeys() : boolean
+setNb_lifes(nb : int) : void
+setNb_bombs(nb : int) : void
+setIsBombDropped(val : boolean) : void
~clearListOfBomb() : void
+setPosPlayer(x : int, y : int) : void
+bombCounter() : int
+loseOneLife() : int
+loadValues() : void
+reloadValues() : void
~dropBomb() : void
~playerInArea() : boolean
~isValidConfig() : ReadOnlyBooleanProperty
~nb_bombsProperty() : IntegerProperty
~nb_lifesProperty() : IntegerProperty
~nb_keysProperty() : IntegerProperty
+toString() : String
Pos
<<Property>> -i : int
<<Property>> -j : int
+Pos()
+Pos(x : int, y : int)
~setI(i : int) : void
~setJ(j : int) : void
+hashCode() : int
+equals(o : Object) : boolean
+toString() : String
GameFacade
DELAY : int = 1000
-isStarted : BooleanProperty = new SimpleBooleanProperty(false)
-isInProgress : BooleanProperty = new SimpleBooleanProperty(false)
-isStartable : BooleanProperty = new SimpleBooleanProperty(false)
-activeConfig : BooleanProperty = new SimpleBooleanProperty(false)
-playerGetMine : BooleanProperty = new SimpleBooleanProperty(false)
-isMineTimesUp : BooleanProperty = new SimpleBooleanProperty(false)
-game : Game = new Game()
-player : Player = new Player()
<<Interface>>
Bonus
+showPos() : void
Mine
AREA_OF_MINE : int = 1
<<Property>> -posOfMine : Pos
+Mine(x : int, y : int)
+containInAreaExplosion(pos : Pos) : boolean
~isAtLeft(pos : Pos) : boolean
~isAtRight(pos : Pos) : boolean
~isUp(pos : Pos) : boolean
~isDown(pos : Pos) : boolean
+showPos() : void
+equals(o : Object) : boolean
+hashCode() : int
+toString() : String
Treasure
<<Property>> -posOfTreasure : Pos
<<Property>> -treasureList : Bonus = new ArrayList<>()
+Treasure(x : int, y : int)
+addTreasure(bonus : Bonus) : void
+fillTreasureInside(n : int, t : Treasure, randomSave : int) : int
-randomBonus(x : int, y : int) : Bonus
+showPos() : void
+getBonus() : Bonus
+equals(o : Object) : boolean
+hashCode() : int
+toString() : String
LifeBonus
<<Property>> -posOfLifeBonus : Pos
+LifeBonus(x : int, y : int)
+showPos() : void
+equals(o : Object) : boolean
+hashCode() : int
+toString() : String
BombBonus
<<Property>> -posOfBombBonus : Pos
+BombBonus(x : int, y : int)
+showPos() : void
+equals(o : Object) : boolean
+hashCode() : int
+toString() : String
Case
value : ObjectProperty<CaseValue> = new SimpleObjectProperty<>(CaseValue.EMPTY)
~getValue() : CaseValue
~setValue(value : CaseValue) : void
~valueProperty() : ReadOnlyObjectProperty<CaseValue>
Game
gameStatus : ObjectProperty<GameStatus> = new SimpleObjectProperty<>(GameStatus....
<<Property>> -field : Field = new Field()
-winningKey : Key
-listOfWalls : Wall = new ArrayList<>()
-listOfWallsSave : Wall = new ArrayList<>(listOfWalls)
-listOfPillars : Pillar = new ArrayList<>()
-listOfKeys : Key = new HashSet<>()
-listOfLifeBonus : LifeBonus = new ArrayList<>()
-listOfBombBonus : BombBonus = new ArrayList<>()
-listOfMines : Mine = new ArrayList<>()
-listOfTreasure : Treasure = new ArrayList<>()
-droppedMine : Mine
-attribute
Field
FIELD_WIDTH : int = 15
-DELAY_MINE : int = 1000
-tl1 : Timeline
-tl2 : Timeline
-tl3 : Timeline
-isMineActivated : BooleanProperty = new SimpleBooleanProperty(false)
-matrix : Case[][]
~Field()
+setValueOnField(x : int, y : int, value : CaseValue) : void
+getValueOfField(x : int, y : int) : CaseValue
+checkMineStatus(pos : Pos) : void
~stopTimeLines() : void
~playTimeLines() : void
+renewField() : void
~valueProperty(line : int, col : int) : ReadOnlyObjectProperty<CaseValue>
~isMineActivated() : BooleanProperty
<enumeration>
CaseValue
EMPTY
PLAYER
WALL
PILLAR
BOMB
KEY
LIFE
EXTRA_BOMB
MINE
TREASURE
MINE_ONE
MINE_TWO
<<enumeration>>
GameStatus
+statusString : String
~GameStatus(statusString : String)
GAME_NOT_LAUNCH
GAME_IN_PROGRESS
PLAYER_WON
PLAYER_LOSE
<enumeration>
UP
DOWN
LEFT
RIGHT
SPACE
Pillar
<<Property>> -posOfPillar : Pos
+Pillar(x : int, y : int)
+hashCode() : int
+equals(o : Object) : boolean
+toString() : String
-winningKey
-listOfKeys
-dropped Bomb
-listOfDroppedBombs
-posOfBomb
-posOfWall
-posOfKey
-posOfPillar
-listOfWalls
-player
-posPlayer
-posOfMine
-posOfLifeBonus
-posOfTreasure
-posOfBombBonus
-treasureList
-listOfTreasure
-listOfLifeBonus
-dropped Mine
-listOfMines
-listOfBombBonus
-game
-matrix
-field
-listOfPillars