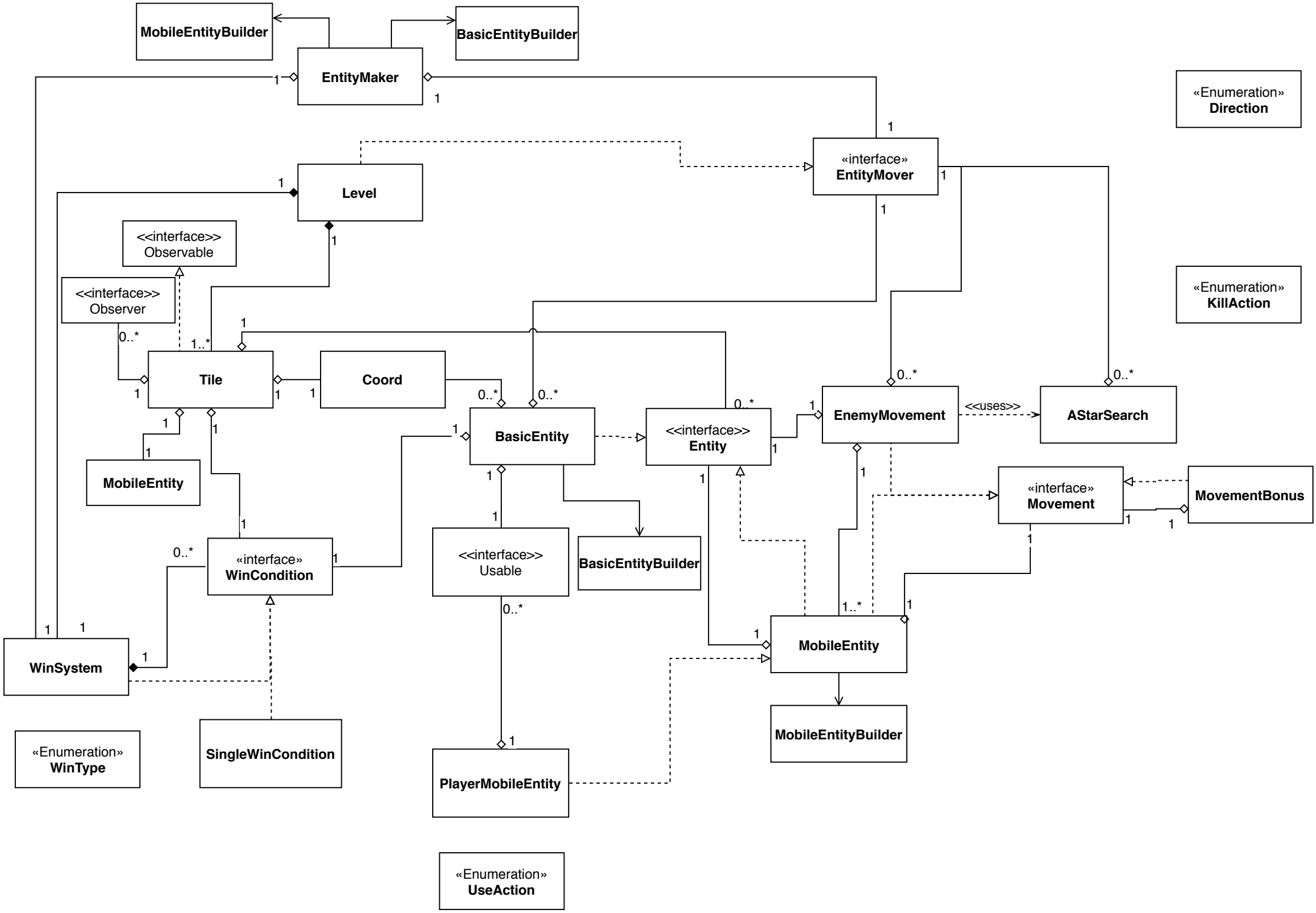
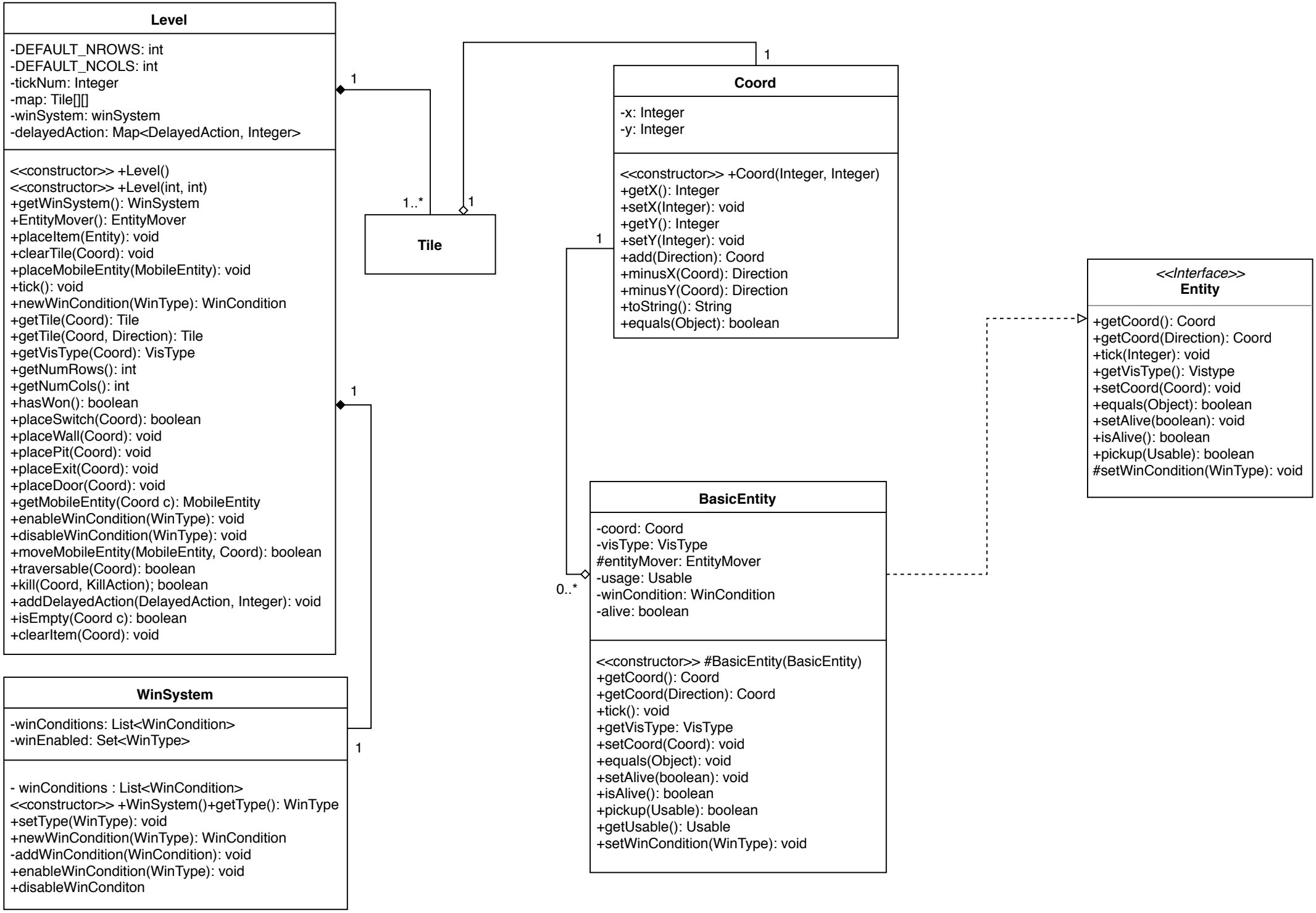
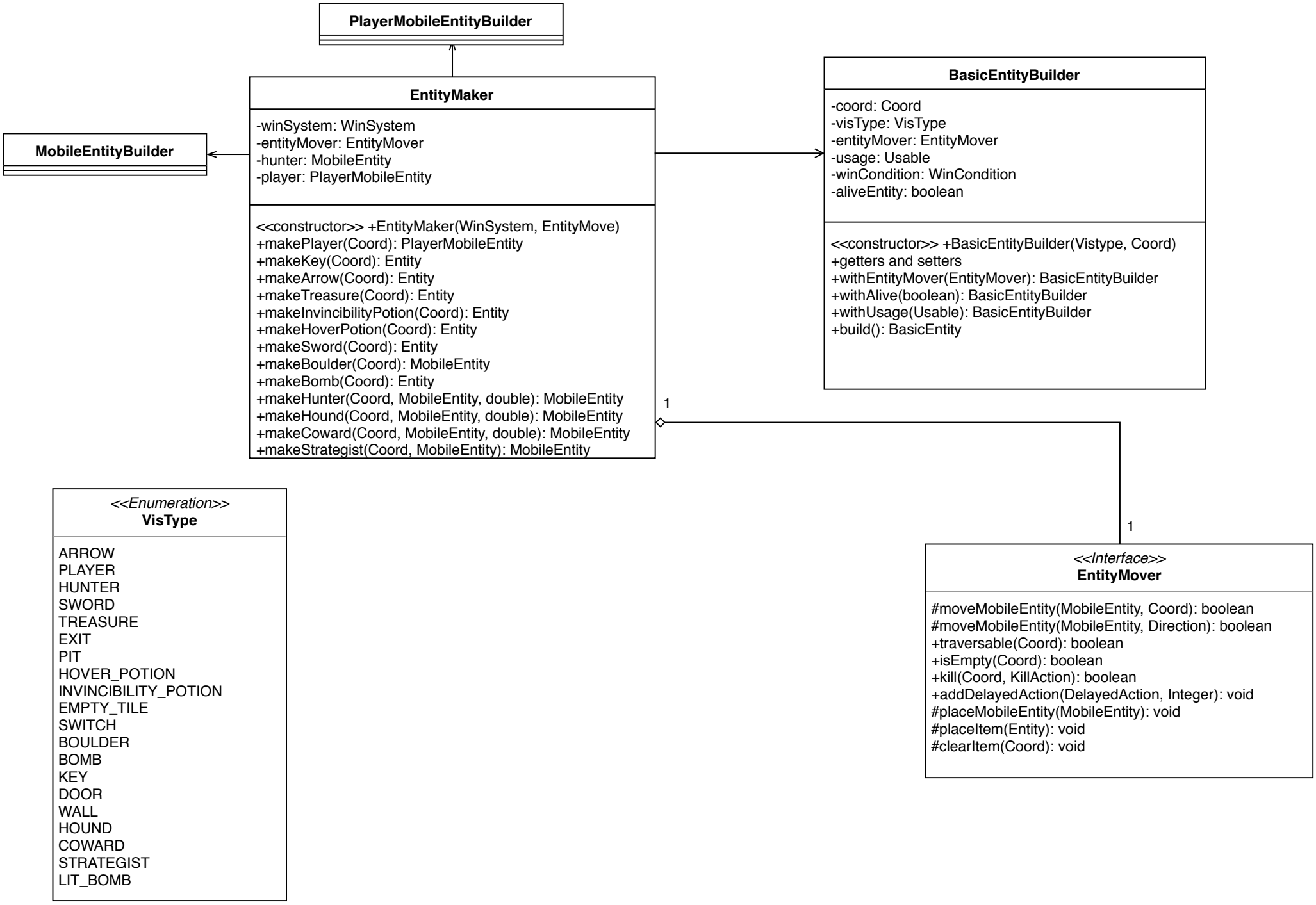
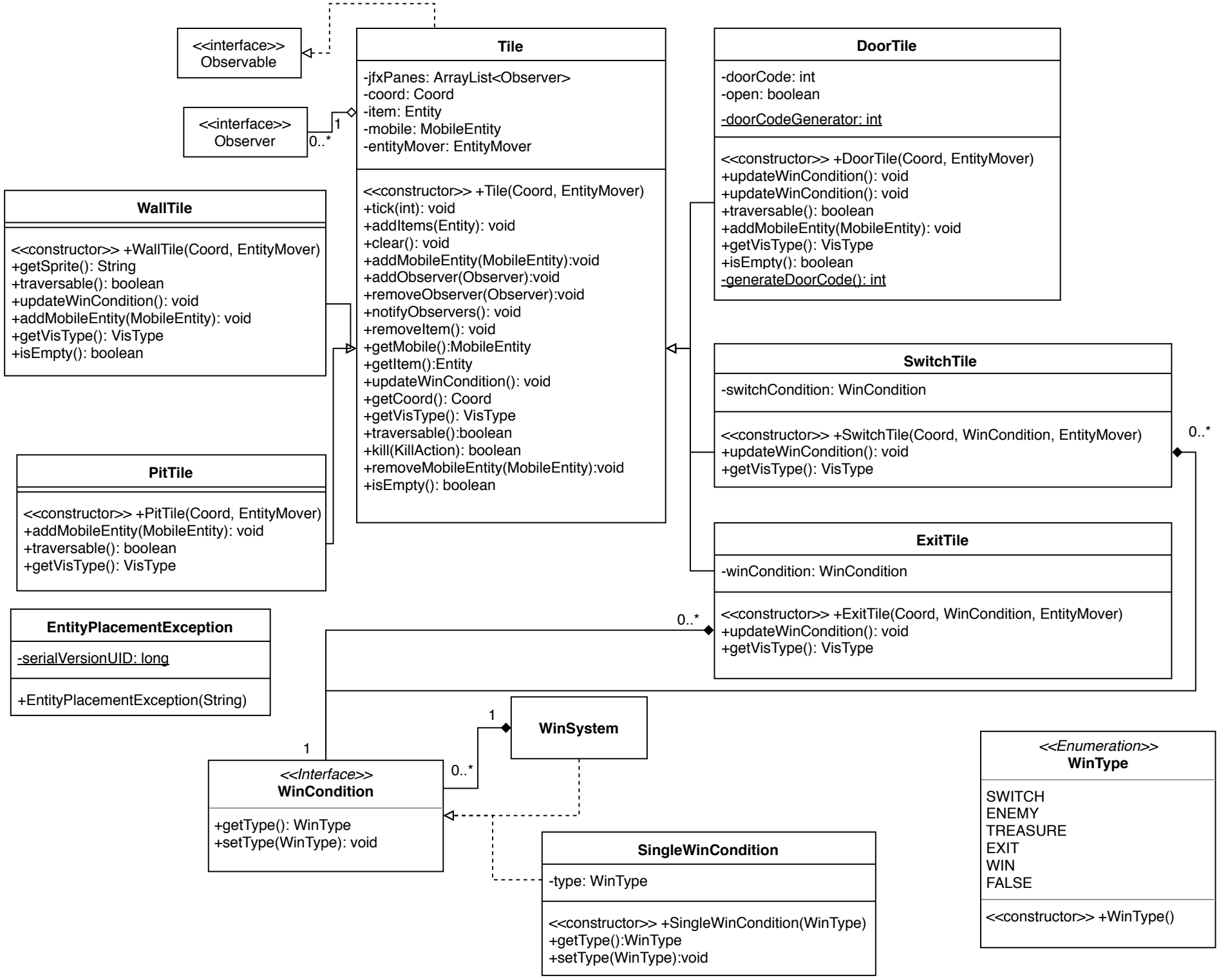
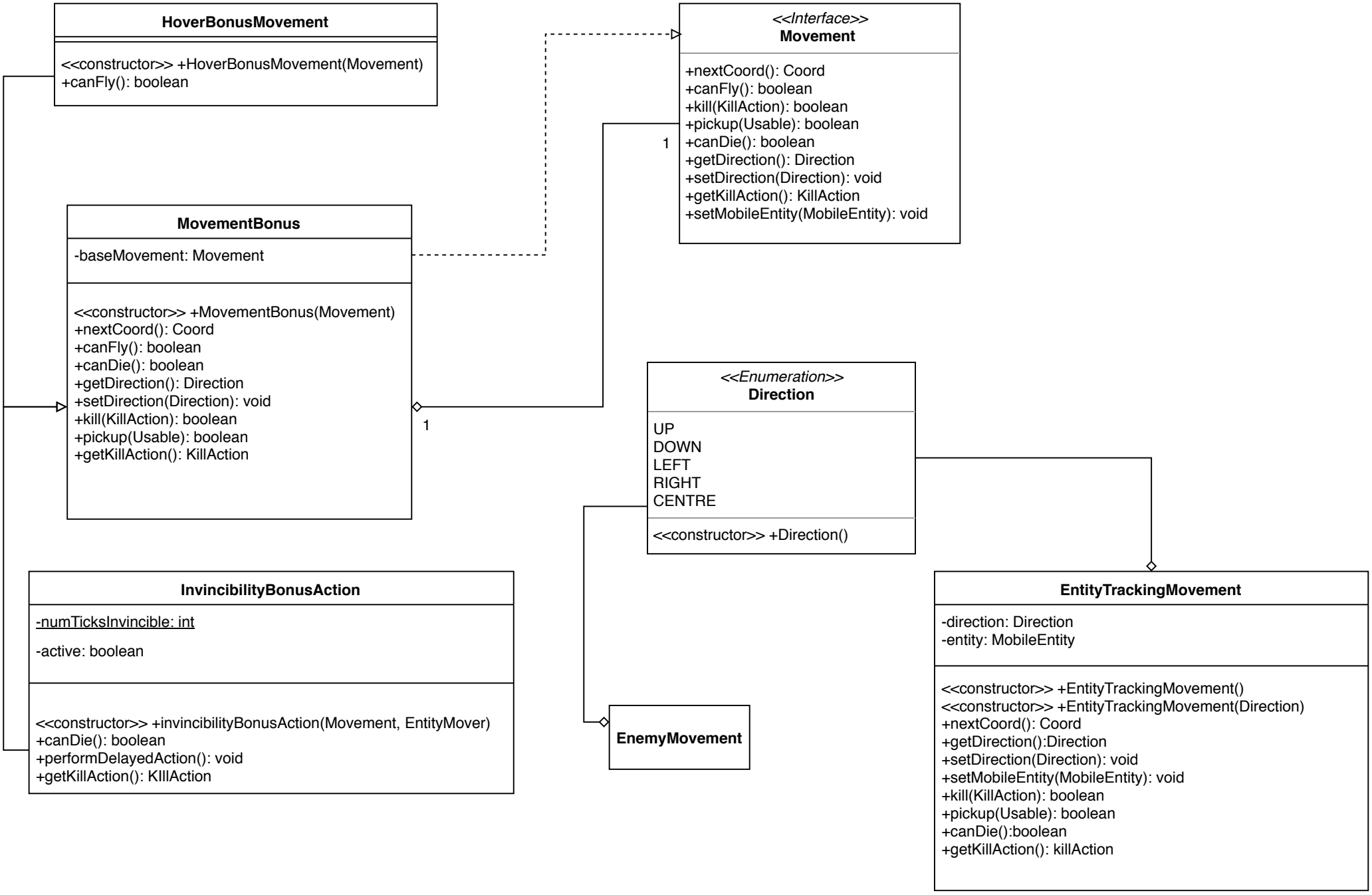
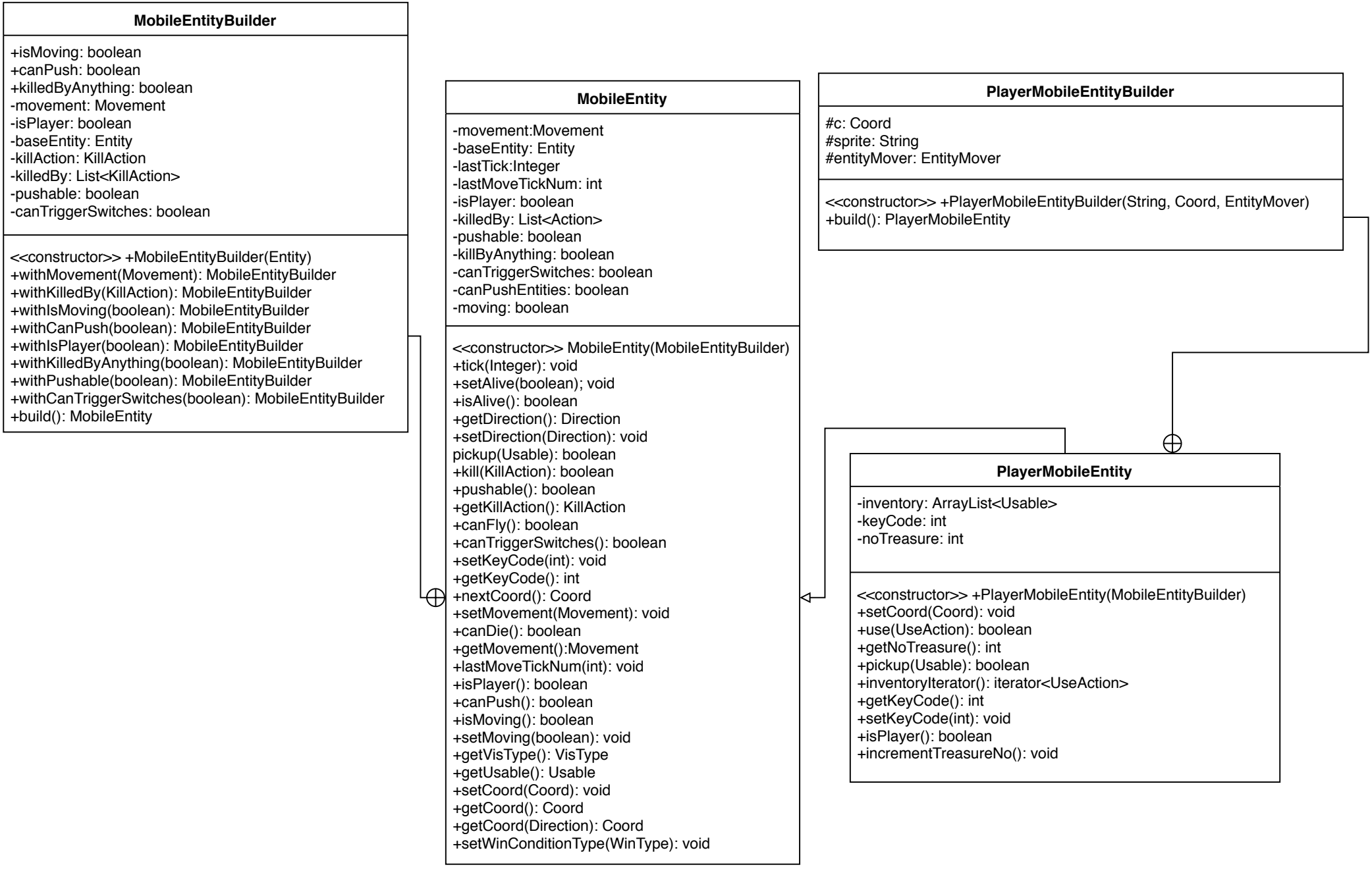

MobileEntityBuilder
BasicEntityBuilder
EntityMaker
«Enumeration»
Direction
«interface»
EntityMover
Level
<<interface>>
Observable
<<interface>>
Observer
«Enumeration»
KillAction
Tile
Coord
BasicEntity
<<interface>>
Entity
EnemyMovement
<<uses>>
AStarSearch
MobileEntity
«interface»
Movement
MovementBonus
«interface»
WinCondition
<<interface>>
Usable
BasicEntityBuilder
WinSystem
MobileEntity
MobileEntityBuilder
«Enumeration»
WinType
SingleWinCondition
PlayerMobileEntity
«Enumeration»
UseAction

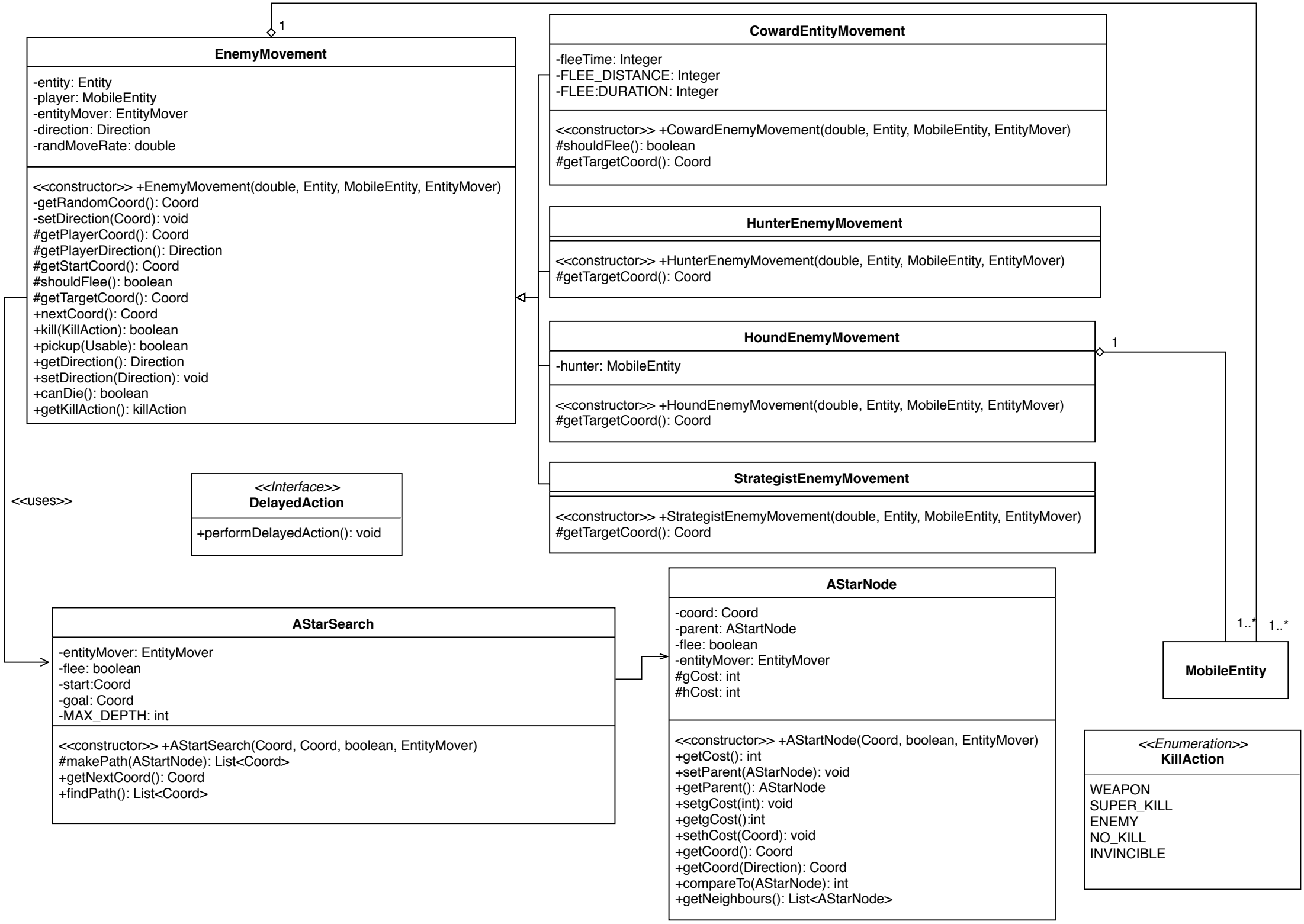

Level
-DEFAULT_NROWS: int
-DEFAULT_NCOLS: int
-tickNum: Integer
-map: Tile[][]
-winSystem: winSystem
-delayedAction: Map<DelayedAction, Integer>
<<constructor>> +Level()
<<constructor>> +Level(int, int)
+getWinSystem(): WinSystem
+EntityMover(): EntityMover
+placeItem(Entity): void
+clearTile(Coord): void
+placeMobileEntity(MobileEntity): void
+tick(): void
+newWinCondition(WinType): WinCondition
+getTile(Coord): Tile
+getTile(Coord, Direction): Tile
+getVisType(Coord): VisType
+getNumRows(): int
+getNumCols(): int
+hasWon(): boolean
+placeSwitch(Coord): boolean
+placeWall(Coord): void
+placePit(Coord): void
+placeExit(Coord): void
+placeDoor(Coord): void
+getMobileEntity(Coord c): MobileEntity
+enableWinCondition(WinType): void
+disableWinCondition(WinType): void
+moveMobileEntity(MobileEntity, Coord): boolean
+traversable(Coord): boolean
+kill(Coord, KillAction); boolean
+addDelayedAction(DelayedAction, Integer): void
+isEmpty(Coord c): boolean
+clearItem(Coord): void
1
1..*
1
Tile
1
Coord
-x: Integer
-y: Integer
<<constructor>> +Coord(Integer, Integer)
+getX(): Integer
+setX(Integer): void
+getY(): Integer
+setY(Integer): void
+add(Direction): Coord
+minusX(Coord): Direction
+minusY(Coord): Direction
+toString(): String
+equals(Object): boolean
1
0..*
<<Interface>>
Entity
+getCoord(): Coord
+getCoord(Direction): Coord
+tick(Integer): void
+getVisType(): Vistype
+setCoord(Coord): void
+equals(Object): boolean
+setAlive(boolean): void
+isAlive(): boolean
+pickup(Usable): boolean
#setWinCondition(WinType): void
BasicEntity
-coord: Coord
-visType: VisType
#entityMover: EntityMover
-usage: Usable
-winCondition: WinCondition
-alive: boolean
<<constructor>> #BasicEntity(BasicEntity)
+getCoord(): Coord
+getCoord(Direction): Coord
+tick(): void
+getVisType: VisType
+setCoord(Coord): void
+equals(Object): void
+setAlive(boolean): void
+isAlive(): boolean
+pickup(Usable): boolean
+getUsable(): Usable
+setWinCondition(WinType): void
1
WinSystem
-winConditions: List<WinCondition>
-winEnabled: Set<WinType>
- winConditions : List<WinCondition>
<<constructor>> +WinSystem()+getType(): WinType
+setType(WinType): void
+newWinCondition(WinType): WinCondition
-addWinCondition(WinCondition): void
+enableWinCondition(WinType): void
+disableWinConditon

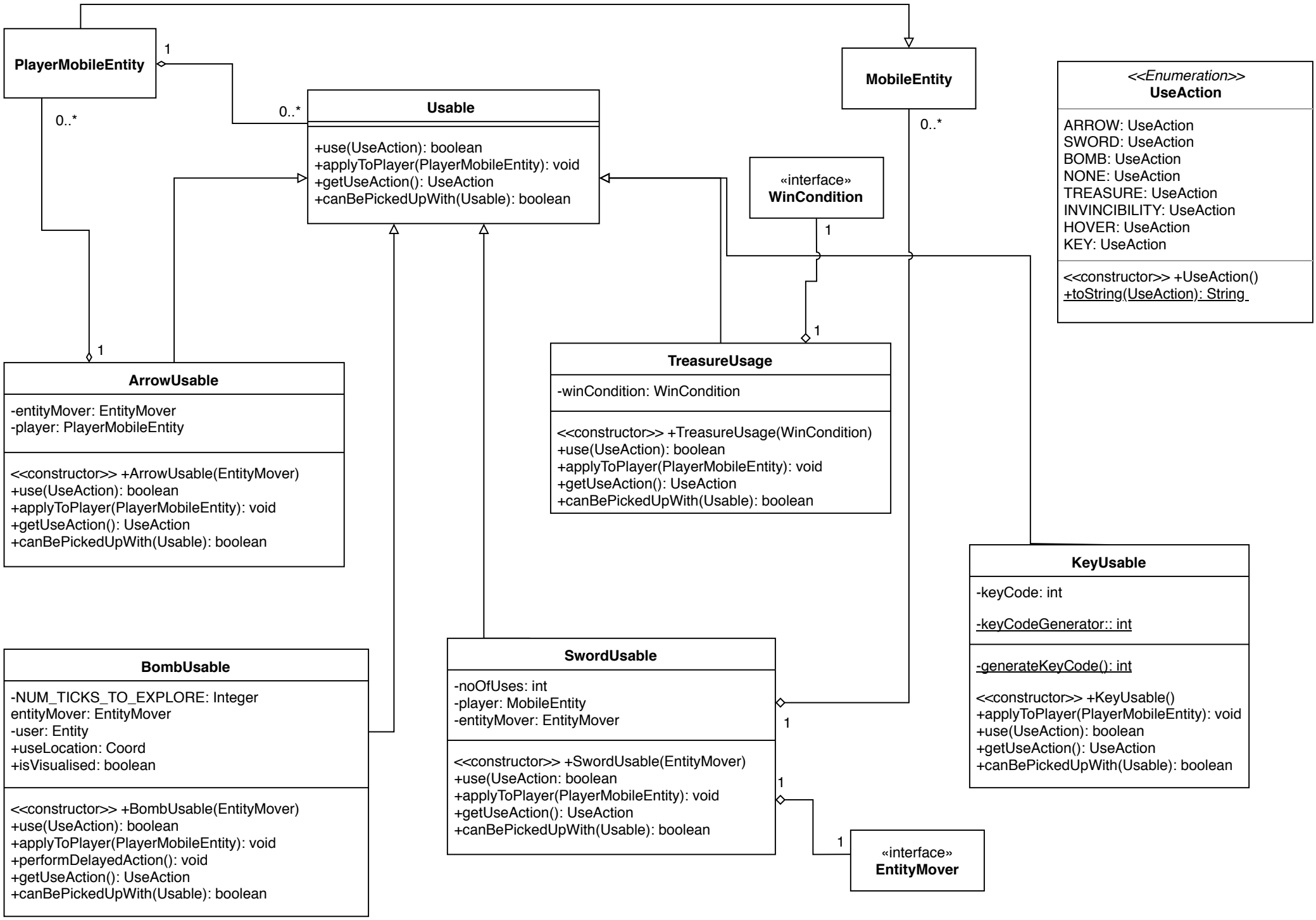

PlayerMobileEntityBuilder
MobileEntityBuilder
EntityMaker
-winSystem: WinSystem
-entityMover: EntityMover
-hunter: MobileEntity
-player: PlayerMobileEntity
<<constructor>> +EntityMaker(WinSystem, EntityMove)
+makePlayer(Coord): PlayerMobileEntity
+makeKey(Coord): Entity
+makeArrow(Coord): Entity
+makeTreasure(Coord): Entity
+makeInvincibilityPotion(Coord): Entity
+makeHoverPotion(Coord): Entity
+makeSword(Coord): Entity
+makeBoulder(Coord): MobileEntity
+makeBomb(Coord): Entity
+makeHunter(Coord, MobileEntity, double): MobileEntity
+makeHound(Coord, MobileEntity, double): MobileEntity
+makeCoward(Coord, MobileEntity, double): MobileEntity
+makeStrategist(Coord, MobileEntity): MobileEntity
BasicEntityBuilder
-coord: Coord
-visType: VisType
-entityMover: EntityMover
-usage: Usable
-winCondition: WinCondition
-aliveEntity: boolean
<<constructor>> +BasicEntityBuilder(Vistype, Coord)
+getters and setters
+withEntityMover(EntityMover): BasicEntityBuilder
+withAlive(boolean): BasicEntityBuilder
+withUsage(Usable): BasicEntityBuilder
+build(): BasicEntity
1
1
<<Enumeration>>
VisType
ARROW
PLAYER
HUNTER
SWORD
TREASURE
EXIT
PIT
HOVER_POTION
INVINCIBILITY_POTION
EMPTY_TILE
SWITCH
BOULDER
BOMB
KEY
DOOR
WALL
HOUND
COWARD
STRATEGIST
LIT_BOMB
<<Interface>>
EntityMover
#moveMobileEntity(MobileEntity, Coord): boolean
#moveMobileEntity(MobileEntity, Direction): boolean
+traversable(Coord): boolean
+isEmpty(Coord): boolean
+kill(Coord, KillAction): boolean
+addDelayedAction(DelayedAction, Integer): void
#placeMobileEntity(MobileEntity): void
#placeItem(Entity): void
#clearItem(Coord): void

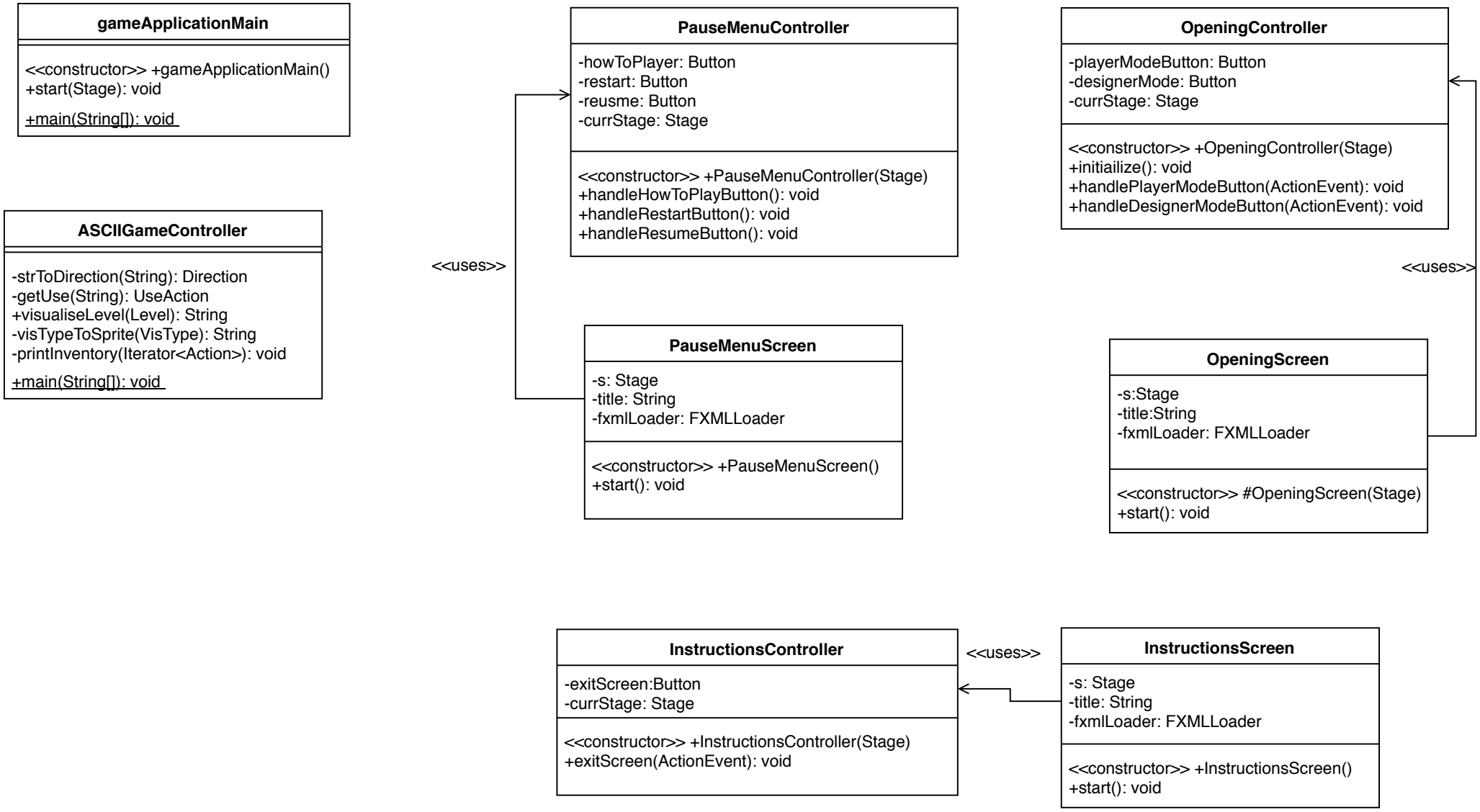

<<interface>>
Observable
<<interface>>
Observer
1
0..*
Tile
-jfxPanes: ArrayList<Observer>
-coord: Coord
-item: Entity
-mobile: MobileEntity
-entityMover: EntityMover
<<constructor>> +Tile(Coord, EntityMover)
+tick(int): void
+addItems(Entity): void
+clear(): void
+addMobileEntity(MobileEntity):void
+addObserver(Observer):void
+removeObserver(Observer):void
+notifyObservers(): void
+removeItem(): void
+getMobile():MobileEntity
+getItem():Entity
+updateWinCondition(): void
+getCoord(): Coord
+getVisType(): VisType
+traversable():boolean
+kill(KillAction): boolean
+removeMobileEntity(MobileEntity):void
+isEmpty(): boolean
DoorTile
-doorCode: int
-open: boolean
-doorCodeGenerator: int
<<constructor>> +DoorTile(Coord, EntityMover)
+updateWinCondition(): void
+updateWinCondition(): void
+traversable(): boolean
+addMobileEntity(MobileEntity): void
+getVisType(): VisType
+isEmpty(): boolean
-generateDoorCode(): int
WallTile
<<constructor>> +WallTile(Coord, EntityMover)
+getSprite(): String
+traversable(): boolean
+updateWinCondition(): void
+addMobileEntity(MobileEntity): void
+getVisType(): VisType
+isEmpty(): boolean
SwitchTile
-switchCondition: WinCondition
<<constructor>> +SwitchTile(Coord, WinCondition, EntityMover)
+updateWinCondition(): void
+getVisType(): VisType
0..*
PitTile
<<constructor>> +PitTile(Coord, EntityMover)
+addMobileEntity(MobileEntity): void
+traversable(): boolean
+getVisType(): VisType
ExitTile
-winCondition: WinCondition
<<constructor>> +ExitTile(Coord, WinCondition, EntityMover)
+updateWinCondition(): void
+getVisType(): VisType
0..*
EntityPlacementException
-serialVersionUID: long
+EntityPlacementException(String)
0..*
1
WinSystem
1
0..*
<<Interface>>
WinCondition
+getType(): WinType
+setType(WinType): void
SingleWinCondition
-type: WinType
<<constructor>> +SingleWinCondition(WinType)
+getType():WinType
+setType(WinType):void
<<Enumeration>>
WinType
SWITCH
ENEMY
TREASURE
EXIT
WIN
FALSE
<<constructor>> +WinType()

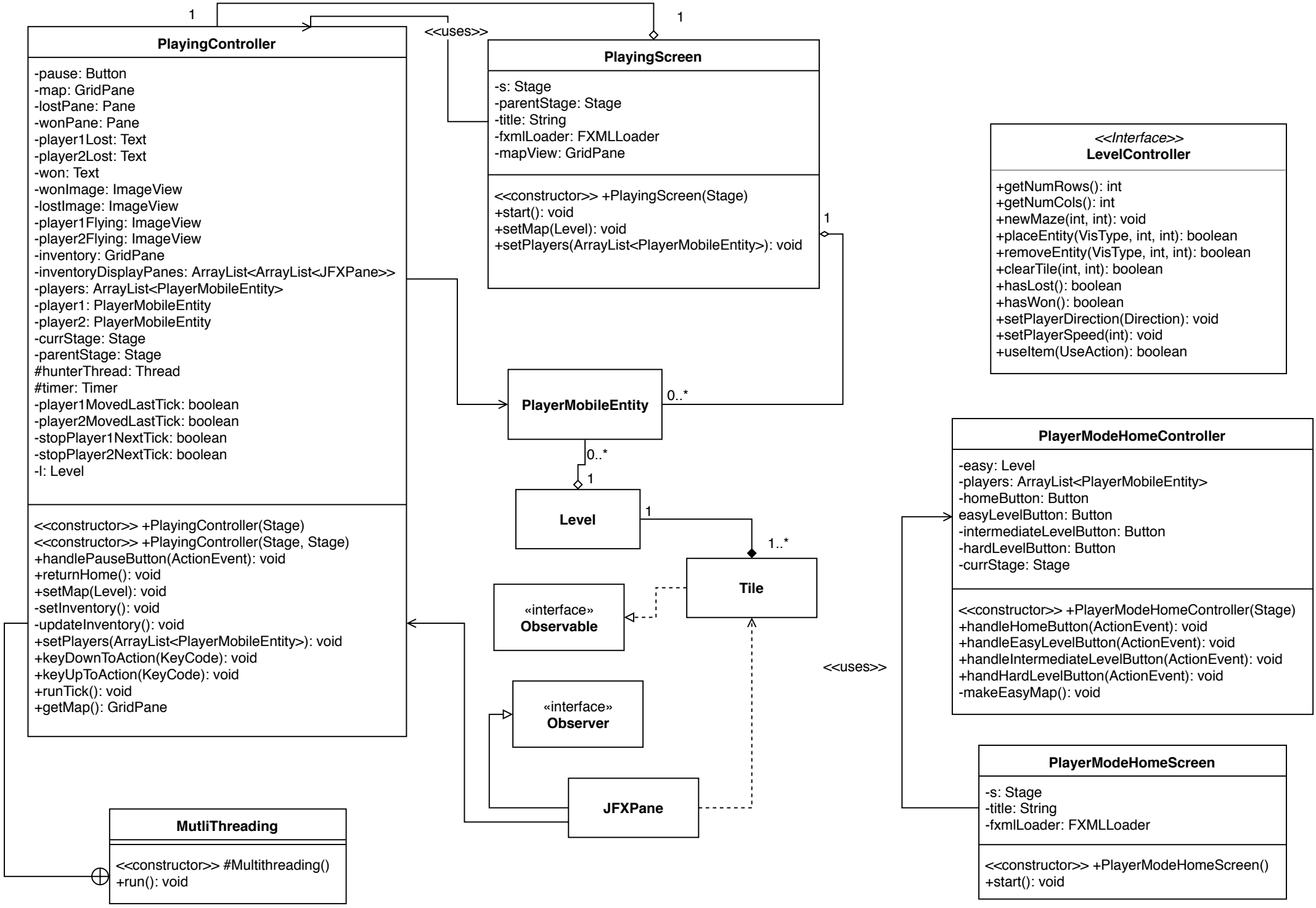

HoverBonusMovement
<<constructor>> +HoverBonusMovement(Movement)
+canFly(): boolean
<<Interface>>
Movement
+nextCoord(): Coord
+canFly(): boolean
+kill(KillAction): boolean
+pickup(Usable): boolean
+canDie(): boolean
+getDirection(): Direction
+setDirection(Direction): void
+getKillAction(): KillAction
+setMobileEntity(MobileEntity): void
1
MovementBonus
-baseMovement: Movement
<<constructor>> +MovementBonus(Movement)
+nextCoord(): Coord
+canFly(): boolean
+canDie(): boolean
+getDirection(): Direction
+setDirection(Direction): void
+kill(KillAction): boolean
+pickup(Usable): boolean
+getKillAction(): KillAction
1
<<Enumeration>>
Direction
UP
DOWN
LEFT
RIGHT
CENTRE
<<constructor>> +Direction()
InvincibilityBonusAction
-numTicksInvincible: int
-active: boolean
<<constructor>> +invincibilityBonusAction(Movement, EntityMover)
+canDie(): boolean
+performDelayedAction(): void
+getKillAction(): KIllAction
EnemyMovement
EntityTrackingMovement
-direction: Direction
-entity: MobileEntity
<<constructor>> +EntityTrackingMovement()
<<constructor>> +EntityTrackingMovement(Direction)
+nextCoord(): Coord
+getDirection():Direction
+setDirection(Direction): void
+setMobileEntity(MobileEntity): void
+kill(KillAction): boolean
+pickup(Usable): boolean
+canDie():boolean
+getKillAction(): killAction

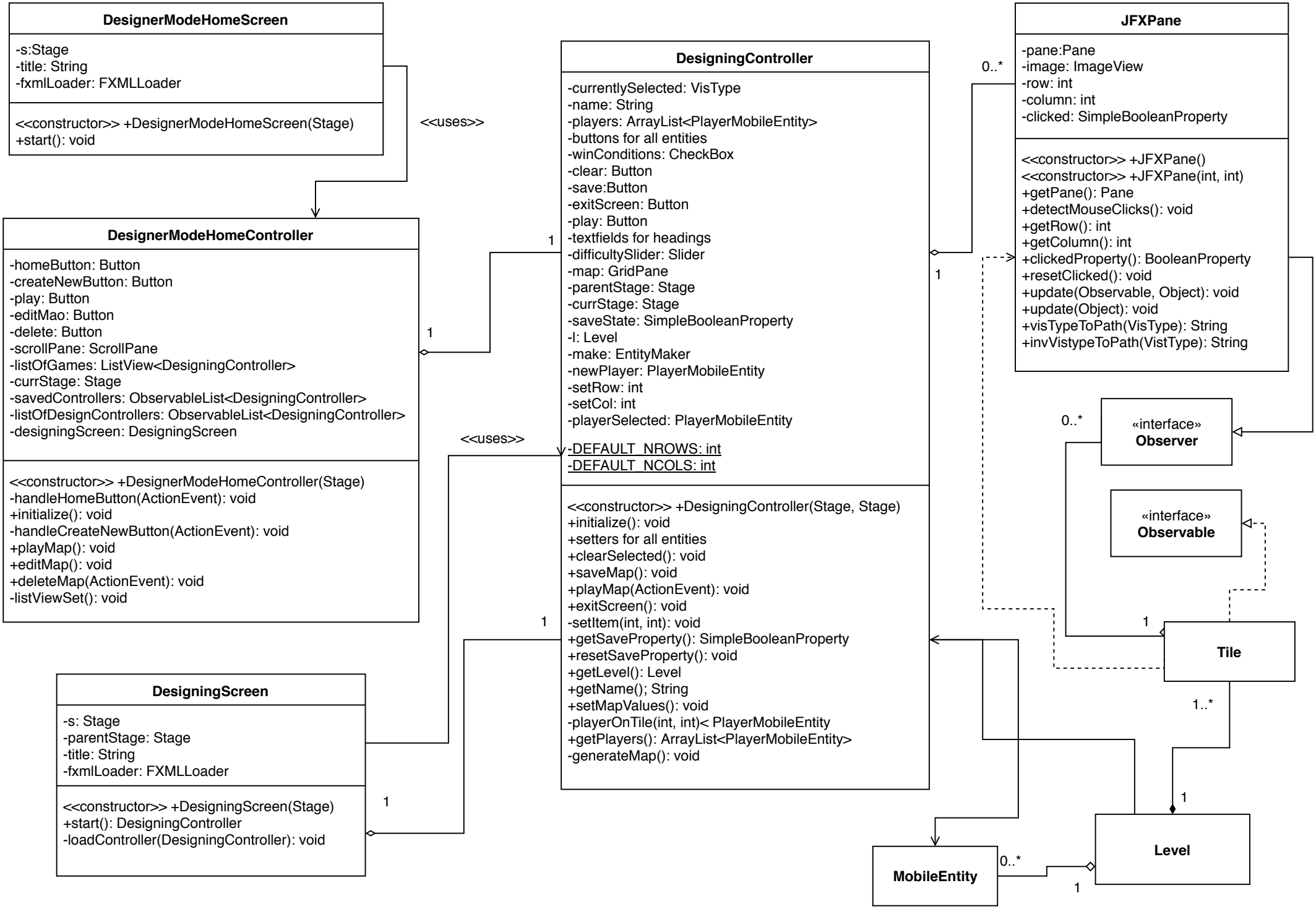

MobileEntityBuilder
+isMoving: boolean
+canPush: boolean
+killedByAnything: boolean
-movement: Movement
-isPlayer: boolean
-baseEntity: Entity
-killAction: KillAction
-killedBy: List<KillAction>
-pushable: boolean
-canTriggerSwitches: boolean
<<constructor>> +MobileEntityBuilder(Entity)
+withMovement(Movement): MobileEntityBuilder
+withKilledBy(KillAction): MobileEntityBuilder
+withIsMoving(boolean): MobileEntityBuilder
+withCanPush(boolean): MobileEntityBuilder
+withIsPlayer(boolean): MobileEntityBuilder
+withKilledByAnything(boolean): MobileEntityBuilder
+withPushable(boolean): MobileEntityBuilder
+withCanTriggerSwitches(boolean): MobileEntityBuilder
+build(): MobileEntity
MobileEntity
-movement:Movement
-baseEntity: Entity
-lastTick:Integer
-lastMoveTickNum: int
-isPlayer: boolean
-killedBy: List<Action>
-pushable: boolean
-killByAnything: boolean
-canTriggerSwitches: boolean
-canPushEntities: boolean
-moving: boolean
<<constructor>> MobileEntity(MobileEntityBuilder)
+tick(Integer): void
+setAlive(boolean); void
+isAlive(): boolean
+getDirection(): Direction
+setDirection(Direction): void
pickup(Usable): boolean
+kill(KillAction): boolean
+pushable(): boolean
+getKillAction(): KillAction
+canFly(): boolean
+canTriggerSwitches(): boolean
+setKeyCode(int): void
+getKeyCode(): int
+nextCoord(): Coord
+setMovement(Movement): void
+canDie(): boolean
+getMovement():Movement
+lastMoveTickNum(int): void
+isPlayer(): boolean
+canPush(): boolean
+isMoving(): boolean
+setMoving(boolean): void
+getVisType(): VisType
+getUsable(): Usable
+setCoord(Coord): void
+getCoord(): Coord
+getCoord(Direction): Coord
+setWinConditionType(WinType): void
PlayerMobileEntityBuilder
#c: Coord
#sprite: String
#entityMover: EntityMover
<<constructor>> +PlayerMobileEntityBuilder(String, Coord, EntityMover)
+build(): PlayerMobileEntity
PlayerMobileEntity
-inventory: ArrayList<Usable>
-keyCode: int
-noTreasure: int
<<constructor>> +PlayerMobileEntity(MobileEntityBuilder)
+setCoord(Coord): void
+use(UseAction): boolean
+getNoTreasure(): int
+pickup(Usable): boolean
+inventoryIterator(): iterator<UseAction>
+getKeyCode(): int
+setKeyCode(int): void
+isPlayer(): boolean
+incrementTreasureNo(): void


EnemyMovement
-entity: Entity
-player: MobileEntity
-entityMover: EntityMover
-direction: Direction
-randMoveRate: double
<<constructor>> +EnemyMovement(double, Entity, MobileEntity, EntityMover)
-getRandomCoord(): Coord
-setDirection(Coord): void
#getPlayerCoord(): Coord
#getPlayerDirection(): Direction
#getStartCoord(): Coord
#shouldFlee(): boolean
#getTargetCoord(): Coord
+nextCoord(): Coord
+kill(KillAction): boolean
+pickup(Usable): boolean
+getDirection(): Direction
+setDirection(Direction): void
+canDie(): boolean
+getKillAction(): killAction
1
CowardEntityMovement
-fleeTime: Integer
-FLEE_DISTANCE: Integer
-FLEE:DURATION: Integer
<<constructor>> +CowardEnemyMovement(double, Entity, MobileEntity, EntityMover)
#shouldFlee(): boolean
#getTargetCoord(): Coord
HunterEnemyMovement
<<constructor>> +HunterEnemyMovement(double, Entity, MobileEntity, EntityMover)
#getTargetCoord(): Coord
HoundEnemyMovement
-hunter: MobileEntity
<<constructor>> +HoundEnemyMovement(double, Entity, MobileEntity, EntityMover)
#getTargetCoord(): Coord
1
StrategistEnemyMovement
<<constructor>> +StrategistEnemyMovement(double, Entity, MobileEntity, EntityMover)
#getTargetCoord(): Coord
<<uses>>
<<Interface>>
DelayedAction
+performDelayedAction(): void
AStarSearch
-entityMover: EntityMover
-flee: boolean
-start:Coord
-goal: Coord
-MAX_DEPTH: int
<<constructor>> +AStartSearch(Coord, Coord, boolean, EntityMover)
#makePath(AStartNode): List<Coord>
+getNextCoord(): Coord
+findPath(): List<Coord>
AStarNode
-coord: Coord
-parent: AStartNode
-flee: boolean
-entityMover: EntityMover
#gCost: int
#hCost: int
<<constructor>> +AStartNode(Coord, boolean, EntityMover)
+getCost(): int
+setParent(AStarNode): void
+getParent(): AStarNode
+setgCost(int): void
+getgCost():int
+sethCost(Coord): void
+getCoord(): Coord
+getCoord(Direction): Coord
+compareTo(AStarNode): int
+getNeighbours(): List<AStarNode>
1..*
1..*
MobileEntity
<<Enumeration>>
KillAction
WEAPON
SUPER_KILL
ENEMY
NO_KILL
INVINCIBLE


PlayerMobileEntity
1
0..*
0..*
Usable
+use(UseAction): boolean
+applyToPlayer(PlayerMobileEntity): void
+getUseAction(): UseAction
+canBePickedUpWith(Usable): boolean
MobileEntity
0..*
<<Enumeration>>
UseAction
ARROW: UseAction
SWORD: UseAction
BOMB: UseAction
NONE: UseAction
TREASURE: UseAction
INVINCIBILITY: UseAction
HOVER: UseAction
KEY: UseAction
<<constructor>> +UseAction()
+toString(UseAction): String
«interface»
WinCondition
1
1
1
ArrowUsable
-entityMover: EntityMover
-player: PlayerMobileEntity
<<constructor>> +ArrowUsable(EntityMover)
+use(UseAction): boolean
+applyToPlayer(PlayerMobileEntity): void
+getUseAction(): UseAction
+canBePickedUpWith(Usable): boolean
TreasureUsage
-winCondition: WinCondition
<<constructor>> +TreasureUsage(WinCondition)
+use(UseAction): boolean
+applyToPlayer(PlayerMobileEntity): void
+getUseAction(): UseAction
+canBePickedUpWith(Usable): boolean
KeyUsable
-keyCode: int
-keyCodeGenerator:: int
-generateKeyCode(): int
<<constructor>> +KeyUsable()
+applyToPlayer(PlayerMobileEntity): void
+use(UseAction): boolean
+getUseAction(): UseAction
+canBePickedUpWith(Usable): boolean
BombUsable
-NUM_TICKS_TO_EXPLORE: Integer
entityMover: EntityMover
-user: Entity
+useLocation: Coord
+isVisualised: boolean
<<constructor>> +BombUsable(EntityMover)
+use(UseAction): boolean
+applyToPlayer(PlayerMobileEntity): void
+performDelayedAction(): void
+getUseAction(): UseAction
+canBePickedUpWith(Usable): boolean
SwordUsable
-noOfUses: int
-player: MobileEntity
-entityMover: EntityMover
<<constructor>> +SwordUsable(EntityMover)
+use(UseAction: boolean
+applyToPlayer(PlayerMobileEntity): void
+getUseAction(): UseAction
+canBePickedUpWith(Usable): boolean
1
1
1
«interface»
EntityMover


gameApplicationMain
<<constructor>> +gameApplicationMain()
+start(Stage): void
+main(String[]): void
ASCIIGameController
-strToDirection(String): Direction
-getUse(String): UseAction
+visualiseLevel(Level): String
-visTypeToSprite(VisType): String
-printInventory(Iterator<Action>): void
+main(String[]): void
PauseMenuController
-howToPlayer: Button
-restart: Button
-reusme: Button
-currStage: Stage
<<constructor>> +PauseMenuController(Stage)
+handleHowToPlayButton(): void
+handleRestartButton(): void
+handleResumeButton(): void
<<uses>>
PauseMenuScreen
-s: Stage
-title: String
-fxmlLoader: FXMLLoader
<<constructor>> +PauseMenuScreen()
+start(): void
OpeningController
-playerModeButton: Button
-designerMode: Button
-currStage: Stage
<<constructor>> +OpeningController(Stage)
+initiailize(): void
+handlePlayerModeButton(ActionEvent): void
+handleDesignerModeButton(ActionEvent): void
<<uses>>
OpeningScreen
-s:Stage
-title:String
-fxmlLoader: FXMLLoader
<<constructor>> #OpeningScreen(Stage)
+start(): void
InstructionsController
-exitScreen:Button
-currStage: Stage
<<constructor>> +InstructionsController(Stage)
+exitScreen(ActionEvent): void
<<uses>>
InstructionsScreen
-s: Stage
-title: String
-fxmlLoader: FXMLLoader
<<constructor>> +InstructionsScreen()
+start(): void


PlayingController
-pause: Button
-map: GridPane
-lostPane: Pane
-wonPane: Pane
-player1Lost: Text
-player2Lost: Text
-won: Text
-wonImage: ImageView
-lostImage: ImageView
-player1Flying: ImageView
-player2Flying: ImageView
-inventory: GridPane
-inventoryDisplayPanes: ArrayList<ArrayList<JFXPane>>
-players: ArrayList<PlayerMobileEntity>
-player1: PlayerMobileEntity
-player2: PlayerMobileEntity
-currStage: Stage
-parentStage: Stage
#hunterThread: Thread
#timer: Timer
-player1MovedLastTick: boolean
-player2MovedLastTick: boolean
-stopPlayer1NextTick: boolean
-stopPlayer2NextTick: boolean
-l: Level
<<constructor>> +PlayingController(Stage)
<<constructor>> +PlayingController(Stage, Stage)
+handlePauseButton(ActionEvent): void
+returnHome(): void
+setMap(Level): void
-setInventory(): void
-updateInventory(): void
+setPlayers(ArrayList<PlayerMobileEntity>): void
+keyDownToAction(KeyCode): void
+keyUpToAction(KeyCode): void
+runTick(): void
+getMap(): GridPane
<<uses>>
1
1
PlayingScreen
-s: Stage
-parentStage: Stage
-title: String
-fxmlLoader: FXMLLoader
-mapView: GridPane
<<constructor>> +PlayingScreen(Stage)
+start(): void
+setMap(Level): void
+setPlayers(ArrayList<PlayerMobileEntity>): void
1
<<Interface>>
LevelController
+getNumRows(): int
+getNumCols(): int
+newMaze(int, int): void
+placeEntity(VisType, int, int): boolean
+removeEntity(VisType, int, int): boolean
+clearTile(int, int): boolean
+hasLost(): boolean
+hasWon(): boolean
+setPlayerDirection(Direction): void
+setPlayerSpeed(int): void
+useItem(UseAction): boolean
PlayerMobileEntity
0..*
0..*
1
Level
1
1..*
Tile
«interface»
Observable
«interface»
Observer
JFXPane
<<uses>>
PlayerModeHomeController
-easy: Level
-players: ArrayList<PlayerMobileEntity>
-homeButton: Button
easyLevelButton: Button
-intermediateLevelButton: Button
-hardLevelButton: Button
-currStage: Stage
<<constructor>> +PlayerModeHomeController(Stage)
+handleHomeButton(ActionEvent): void
+handleEasyLevelButton(ActionEvent): void
+handleIntermediateLevelButton(ActionEvent): void
+handHardLevelButton(ActionEvent): void
-makeEasyMap(): void
PlayerModeHomeScreen
-s: Stage
-title: String
-fxmlLoader: FXMLLoader
<<constructor>> +PlayerModeHomeScreen()
+start(): void
MutliThreading
<<constructor>> #Multithreading()
+run(): void


DesignerModeHomeScreen
-s:Stage
-title: String
-fxmlLoader: FXMLLoader
<<constructor>> +DesignerModeHomeScreen(Stage)
+start(): void
<<uses>>
DesignerModeHomeController
-homeButton: Button
-createNewButton: Button
-play: Button
-editMao: Button
-delete: Button
-scrollPane: ScrollPane
-listOfGames: ListView<DesigningController>
-currStage: Stage
-savedControllers: ObservableList<DesigningController>
-listOfDesignControllers: ObservableList<DesigningController>
-designingScreen: DesigningScreen
<<constructor>> +DesignerModeHomeController(Stage)
-handleHomeButton(ActionEvent): void
+initialize(): void
-handleCreateNewButton(ActionEvent): void
+playMap(): void
+editMap(): void
+deleteMap(ActionEvent): void
-listViewSet(): void
1
1
<<uses>>
1
1
DesigningScreen
-s: Stage
-parentStage: Stage
-title: String
-fxmlLoader: FXMLLoader
<<constructor>> +DesigningScreen(Stage)
+start(): DesigningController
-loadController(DesigningController): void
DesigningController
-currentlySelected: VisType
-name: String
-players: ArrayList<PlayerMobileEntity>
-buttons for all entities
-winConditions: CheckBox
-clear: Button
-save:Button
-exitScreen: Button
-play: Button
-textfields for headings
-difficultySlider: Slider
-map: GridPane
-parentStage: Stage
-currStage: Stage
-saveState: SimpleBooleanProperty
-l: Level
-make: EntityMaker
-newPlayer: PlayerMobileEntity
-setRow: int
-setCol: int
-playerSelected: PlayerMobileEntity
-DEFAULT_NROWS: int
-DEFAULT_NCOLS: int
<<constructor>> +DesigningController(Stage, Stage)
+initialize(): void
+setters for all entities
+clearSelected(): void
+saveMap(): void
+playMap(ActionEvent): void
+exitScreen(): void
-setItem(int, int): void
+getSaveProperty(): SimpleBooleanProperty
+resetSaveProperty(): void
+getLevel(): Level
+getName(); String
+setMapValues(): void
-playerOnTile(int, int)< PlayerMobileEntity
+getPlayers(): ArrayList<PlayerMobileEntity>
-generateMap(): void
0..*
1
JFXPane
-pane:Pane
-image: ImageView
-row: int
-column: int
-clicked: SimpleBooleanProperty
<<constructor>> +JFXPane()
<<constructor>> +JFXPane(int, int)
+getPane(): Pane
+detectMouseClicks(): void
+getRow(): int
+getColumn(): int
+clickedProperty(): BooleanProperty
+resetClicked(): void
+update(Observable, Object): void
+update(Object): void
+visTypeToPath(VisType): String
+invVistypeToPath(VistType): String
0..*
«interface»
Observer
«interface»
Observable
1
Tile
1..*
1
Level
MobileEntity
0..*
1