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

<<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
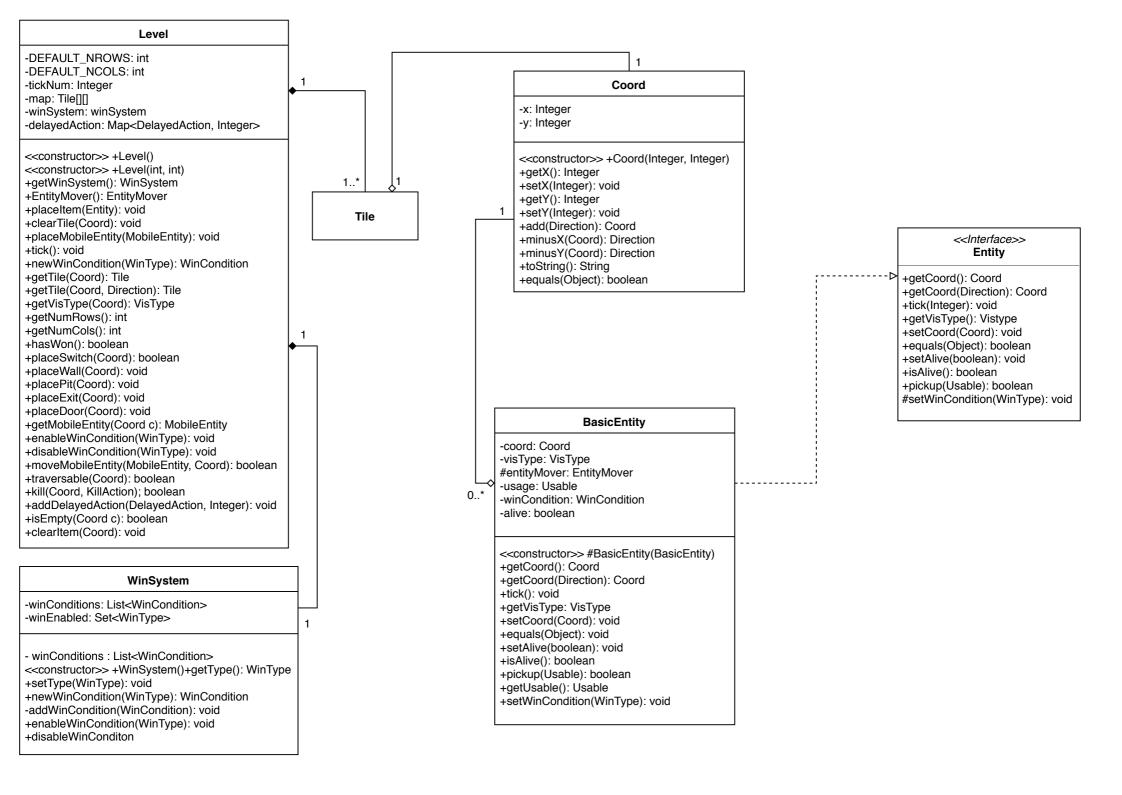#moveMobileEntity(MobileEntity, Direction): boolean
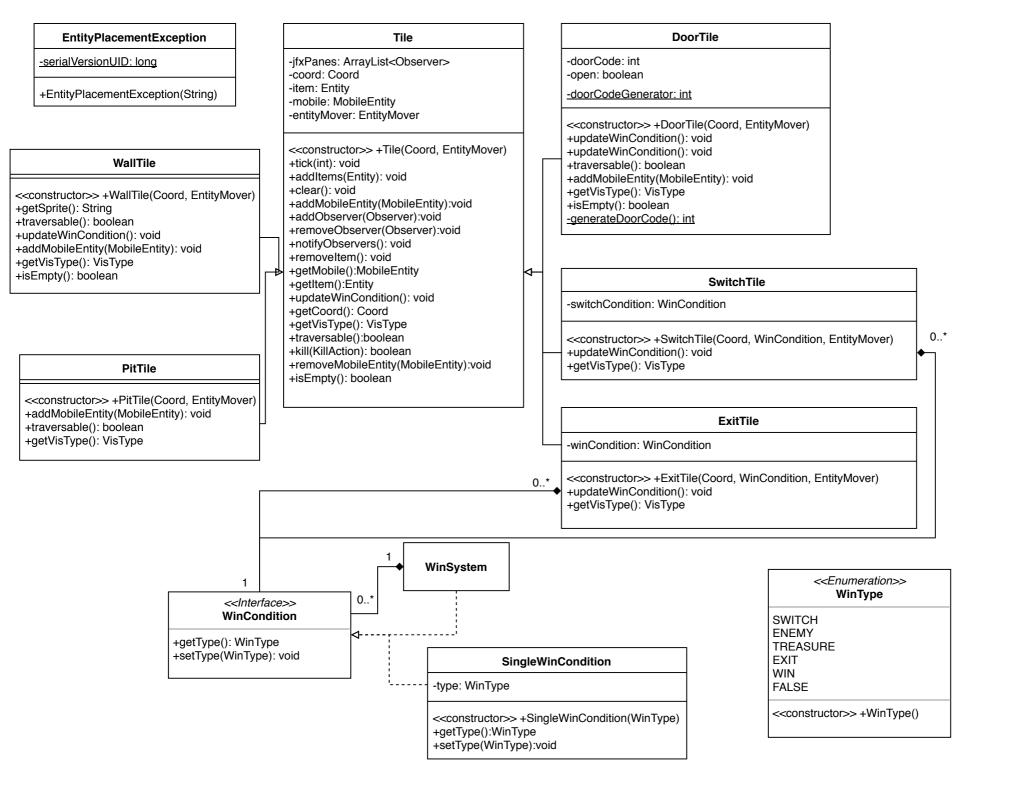+traversable(Coord): boolean
+isEmpty(Coord): boolean
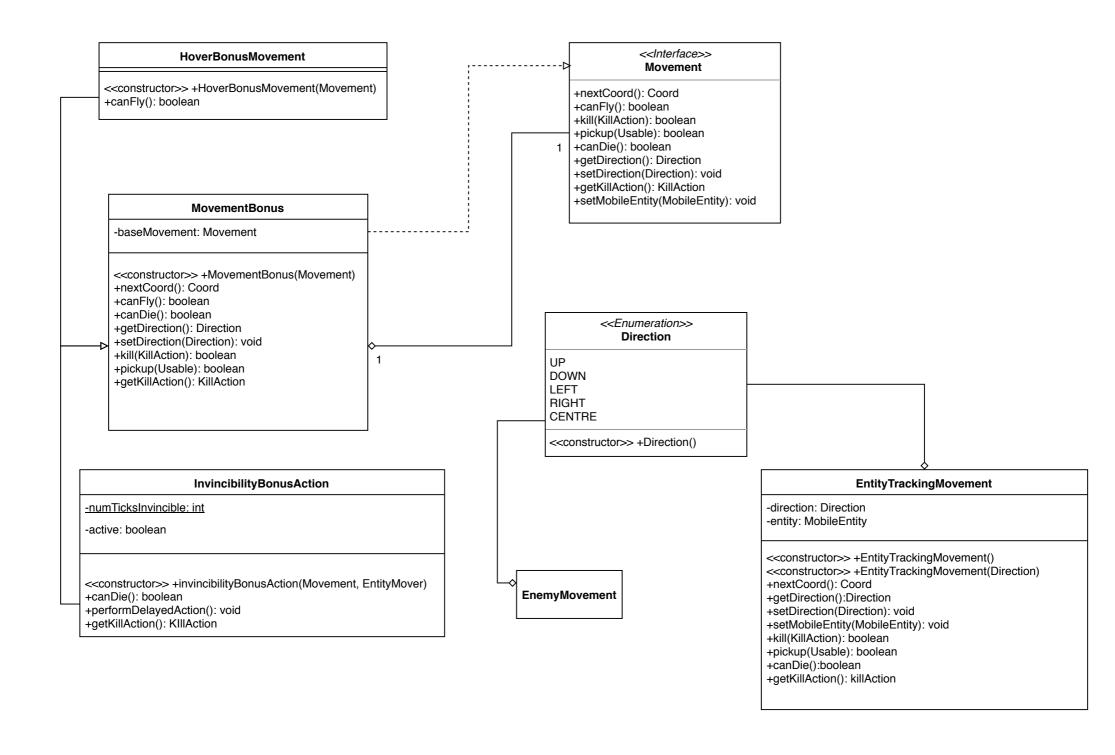+kill(Coord, KillAction): boolean
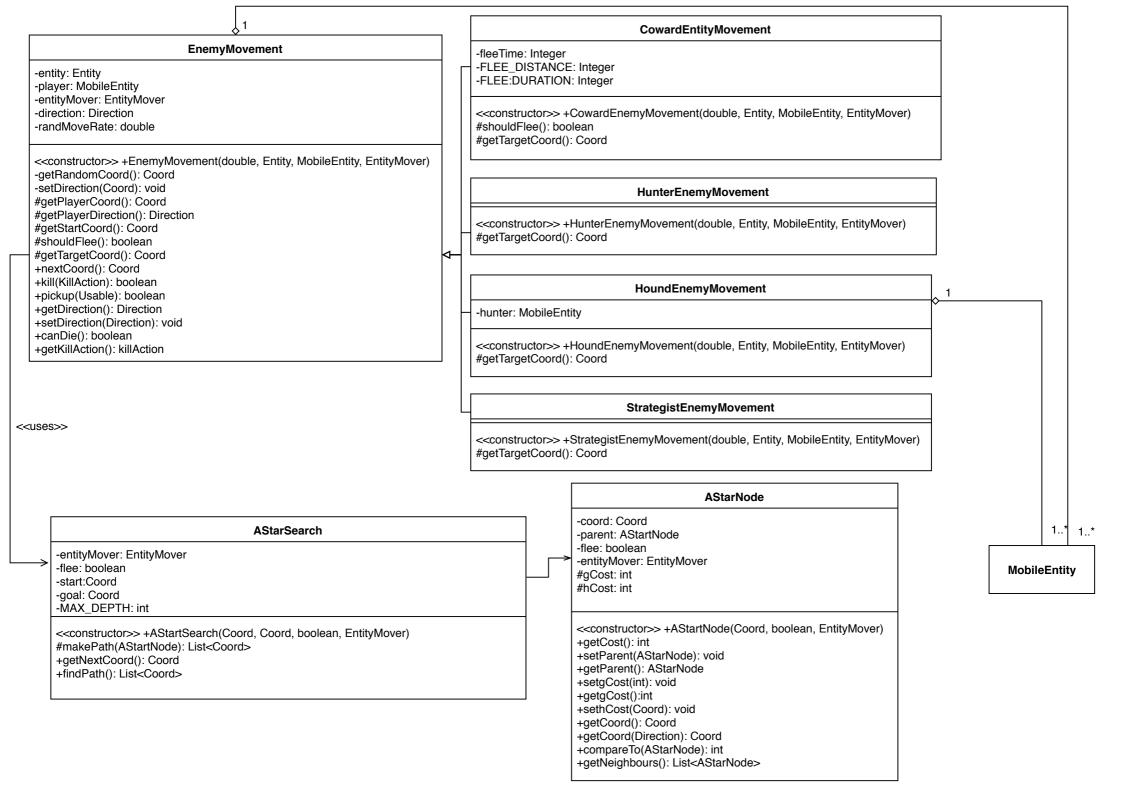+addDelayedAction(DelayedAction, Integer): void
#placeMobileEntity(MobileEntity): void
#placeItem(Entity): void
#clearItem(Coord): void

1

1

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

Tile

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

<<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

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

EntityPlacementException

-serialVersionUID: long

+EntityPlacementException(String)

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

PitTile

<<constructor>> +PitTile(Coord, EntityMover)
+addMobileEntity(MobileEntity): void
+traversable(): boolean
+getVisType(): VisType

SwitchTile

-switchCondition: WinCondition

<<constructor>> +SwitchTile(Coord, WinCondition, EntityMover)
+updateWinCondition(): void
+getVisType(): VisType

ExitTile

-winCondition: WinCondition

<<constructor>> +ExitTile(Coord, WinCondition, EntityMover)
+updateWinCondition(): void
+getVisType(): VisType

WinSystem

WinCondition
<<Interface>>

+getType(): WinType
+setType(WinType): void

SingleWinCondition

-type: WinType

<<constructor>> +SingleWinCondition(WinType)
+getType():WinType
+setType(WinType):void

WinType
<<Enumeration>>

SWITCH
ENEMY
TREASURE
EXIT
WIN
FALSE

<<constructor>> +WinType()

0..*

0..*

1

1

0..*

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

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

<<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

1

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

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

StrategistEnemyMovement

<<constructor>> +StrategistEnemyMovement(double, Entity, MobileEntity, EntityMover)
#getTargetCoord(): Coord

<<uses>>

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>

MobileEntity

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

PlayerMobileEntity

1

MobileEntity

<<Enumeration>>
UseAction

ARROW: UseAction
SWORD: UseAction
BOMB: UseAction
NONE: UseAction
TREASURE: UseAction
INVINCIBILITY: UseAction
HOVER: UseAction
KEY: UseAction

<<constructor>> +UseAction()
+toString(UseAction): String

0..*

0..*

Usable

+use(UseAction): boolean
+applyToPlayer(PlayerMobileEntity): void
+getUseAction(): UseAction
+canBePickedUpWith(Usable): boolean

0..*

«interface»
WinCondition

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