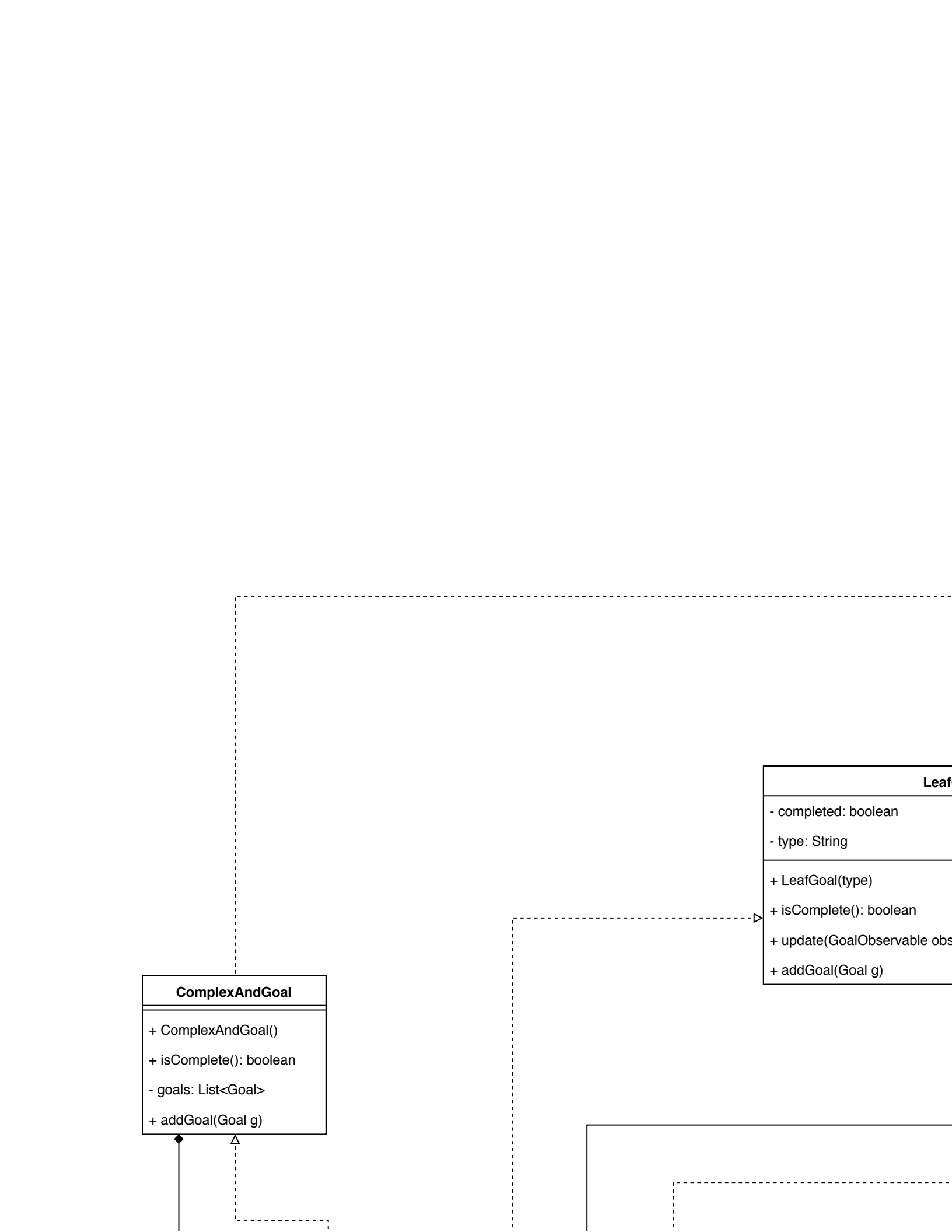
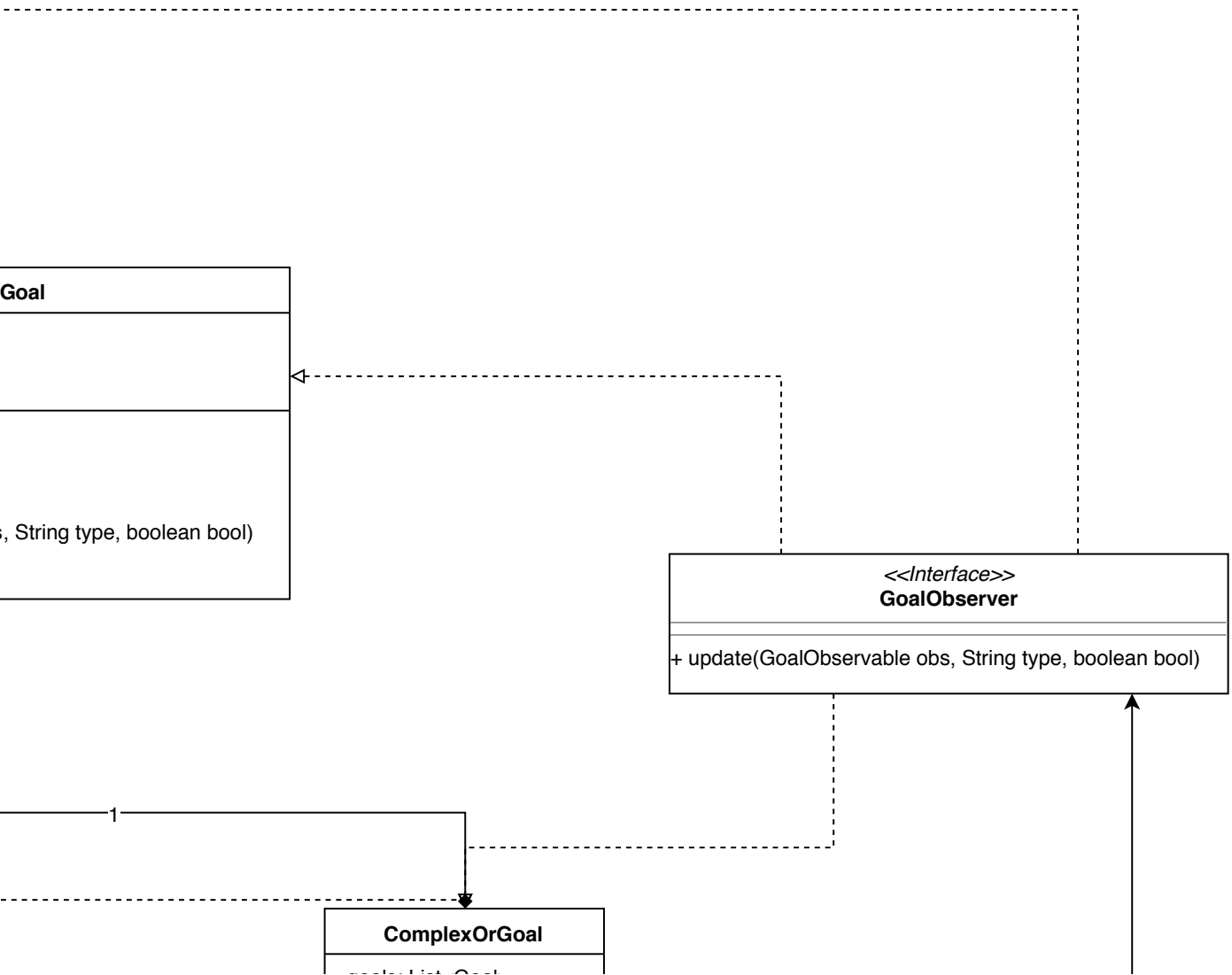
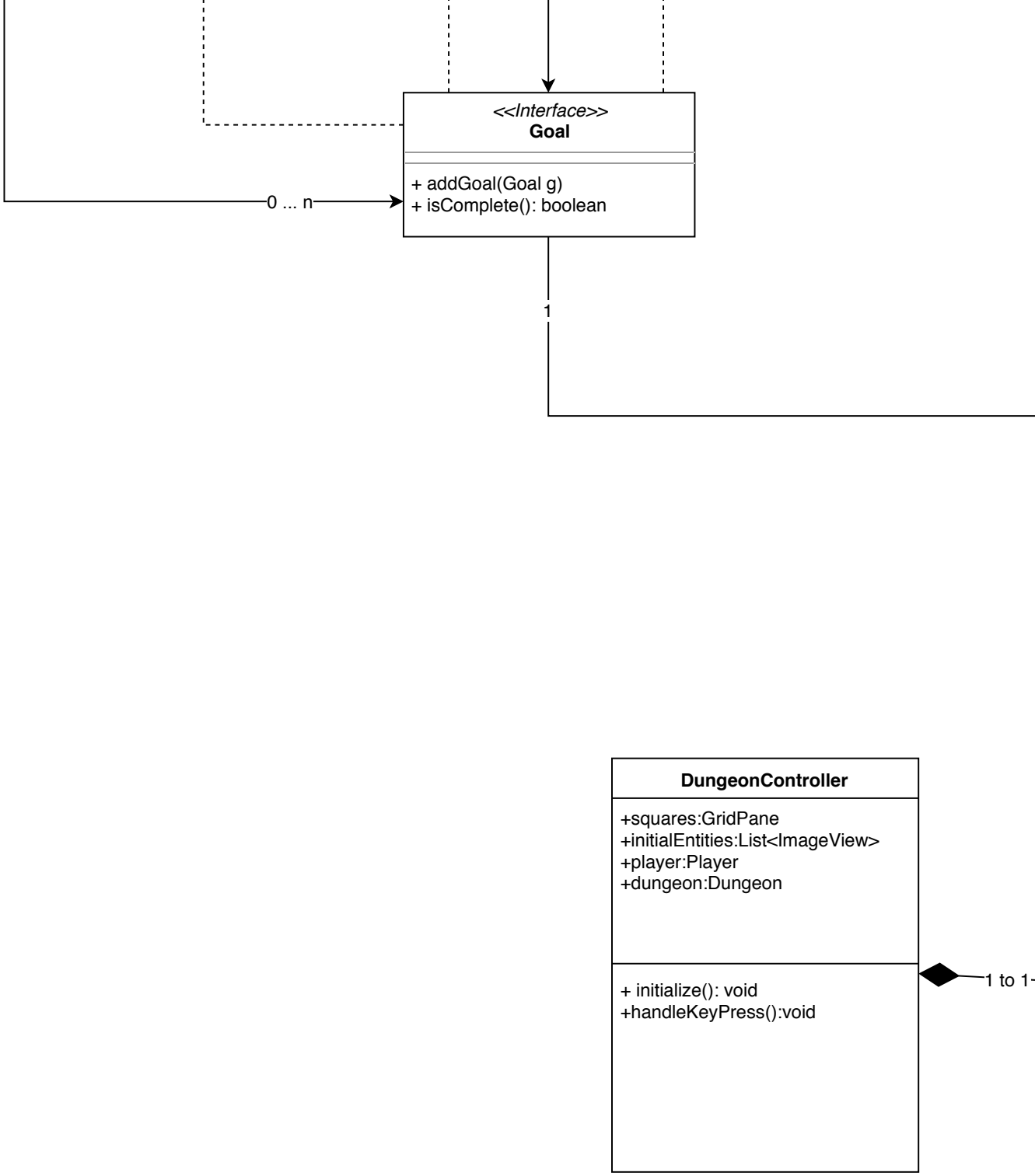

Leaf
- completed: boolean
- type: String
+ LeafGoal(type)
+ isComplete(): boolean
+ update(GoalObservable obs
+ addGoal(Goal g)
ComplexAndGoal
+ ComplexAndGoal()
+ isComplete(): boolean
- goals: List<Goal>
+ addGoal(Goal g)

1
0 ... n

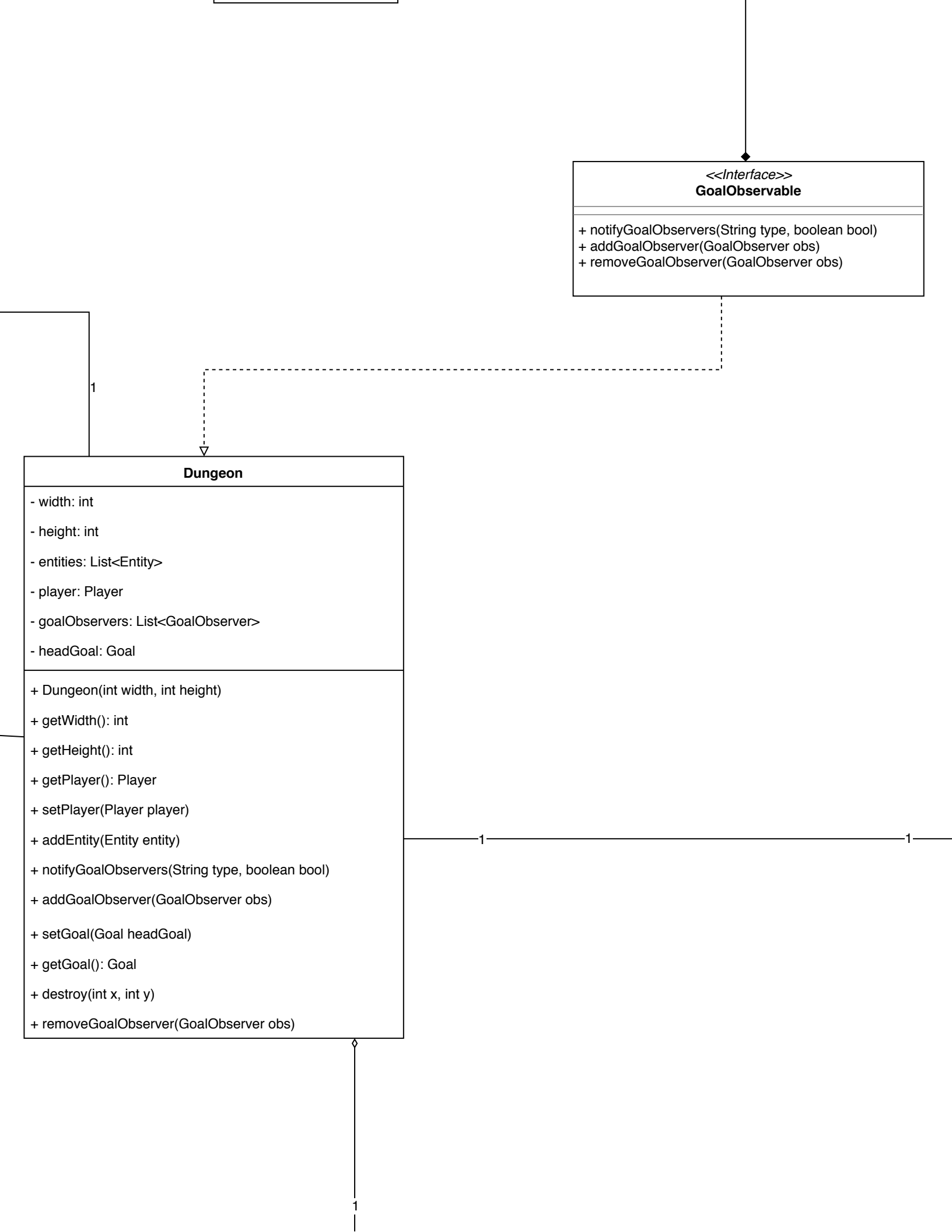

Goal
s, String type, boolean bool)
<<Interface>>
GoalObserver
+ update(GoalObservable obs, String type, boolean bool)
1
ComplexOrGoal

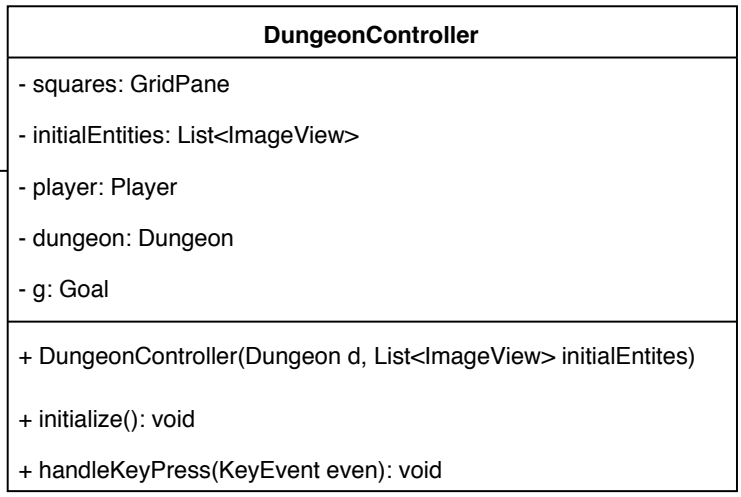
- goals: List<Goal>
+ ComplexOrGoal()
+ isComplete(): boolean
+ addGoal(Goal g)

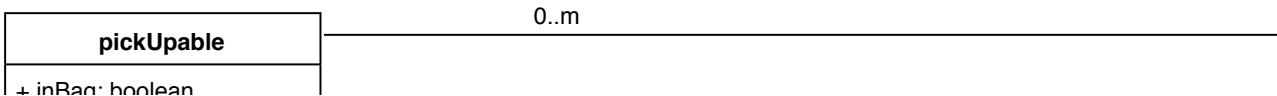

<<Interface>>
Goal
+ addGoal(Goal g)
+ isComplete(): boolean
0 ... n
1

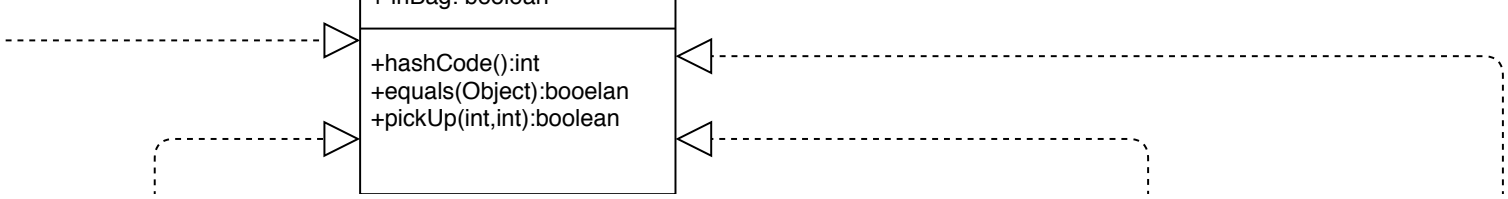
DungeonController
+squares:GridPane
+initialEntities:List<ImageView>
+player:Player
+dungeon:Dungeon
+ initialize(): void
+handleKeyPress():void
1 to 1

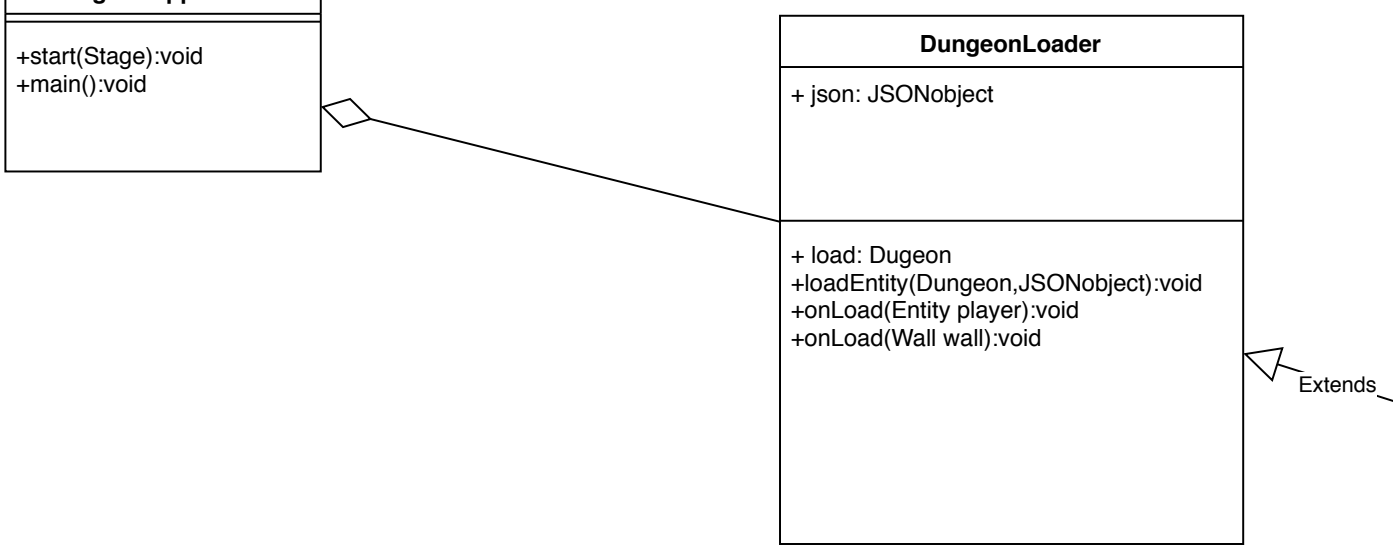
DungeonApplication

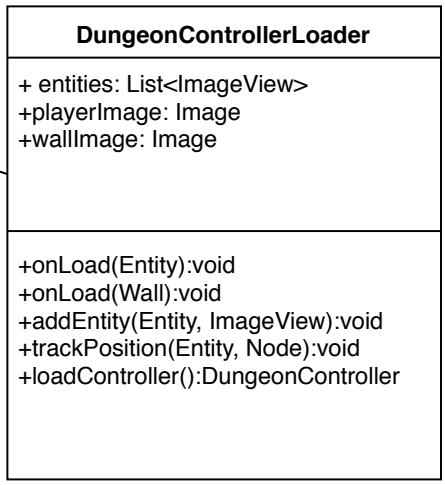

<<Interface>>
GoalObservable
+ notifyGoalObservers(String type, boolean bool)
+ addGoalObserver(GoalObserver obs)
+ removeGoalObserver(GoalObserver obs)
1
Dungeon
- width: int
- height: int
- entities: List<Entity>
- player: Player
- goalObservers: List<GoalObserver>
- headGoal: Goal
+ Dungeon(int width, int height)
+ getWidth(): int
+ getHeight(): int
+ getPlayer(): Player
+ setPlayer(Player player)
+ addEntity(Entity entity)
+ notifyGoalObservers(String type, boolean bool)
+ addGoalObserver(GoalObserver obs)
+ setGoal(Goal headGoal)
+ getGoal(): Goal
+ destroy(int x, int y)
+ removeGoalObserver(GoalObserver obs)
1
1
1

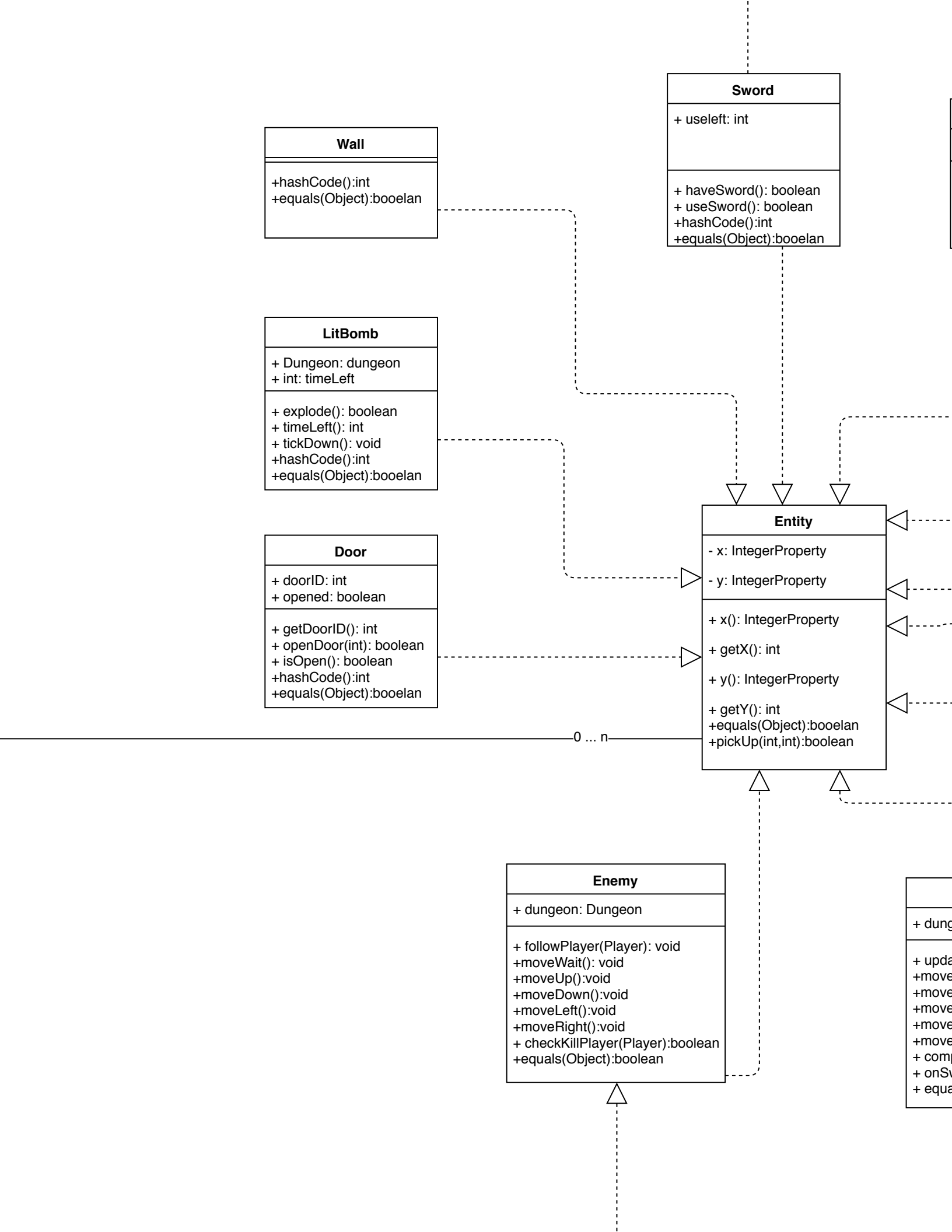

DungeonController
- squares: GridPane
- initialEntities: List<ImageView>
- player: Player
- dungeon: Dungeon
- g: Goal
+ DungeonController(Dungeon d, List<ImageView> initialEntites)
+ initialize(): void
+ handleKeyPress(KeyEvent even): void

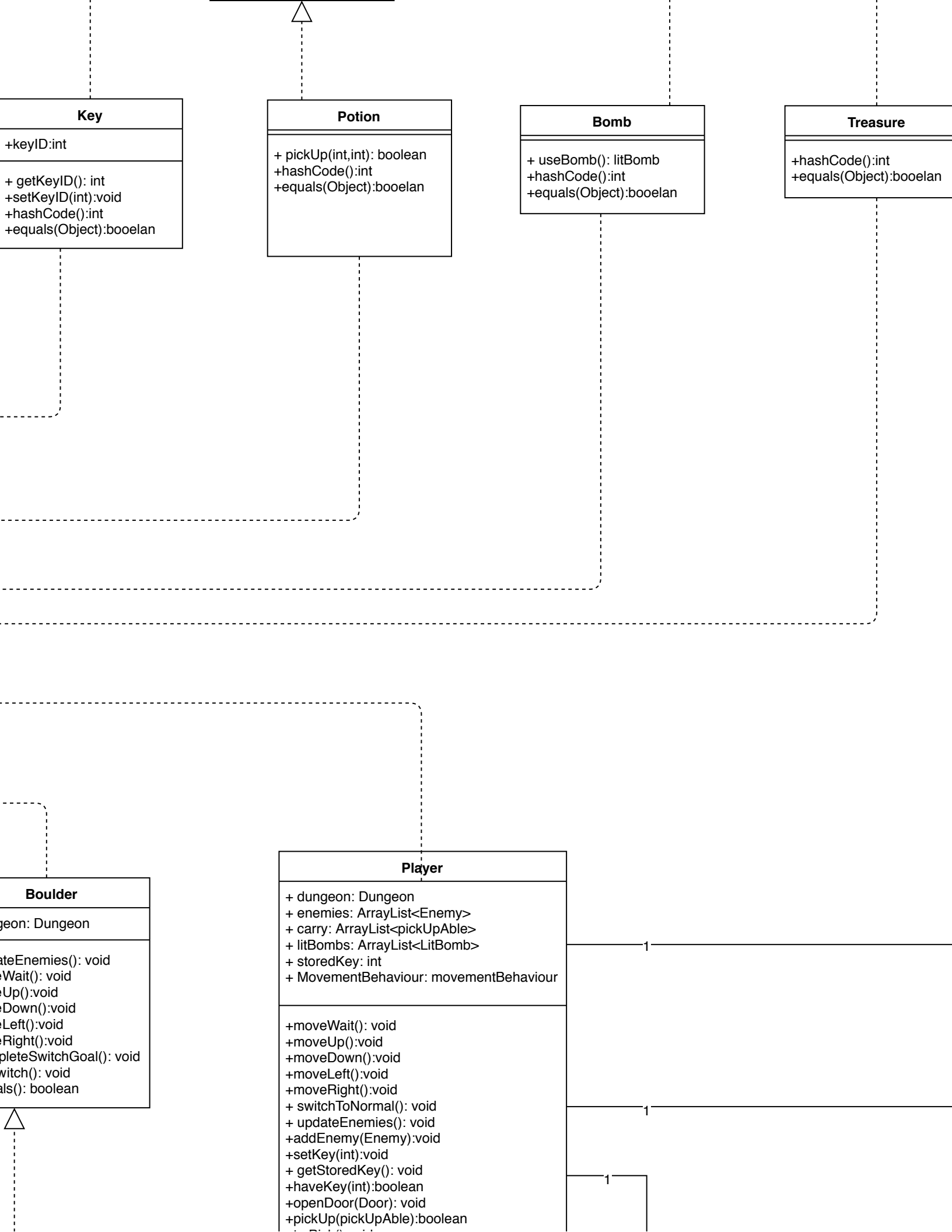

0..m
pickUpable
+ inBag: boolean

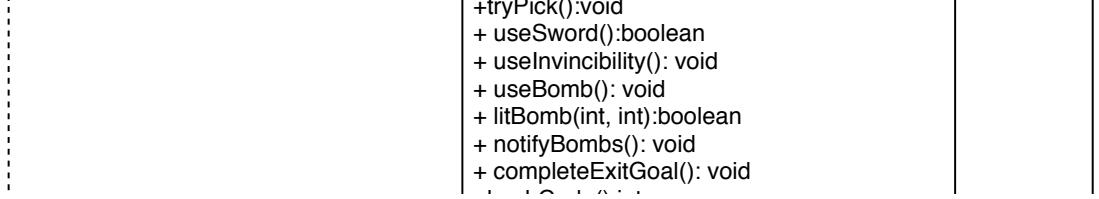

+hashCode():int
+equals(Object):booelan
+pickUp(int,int):boolean

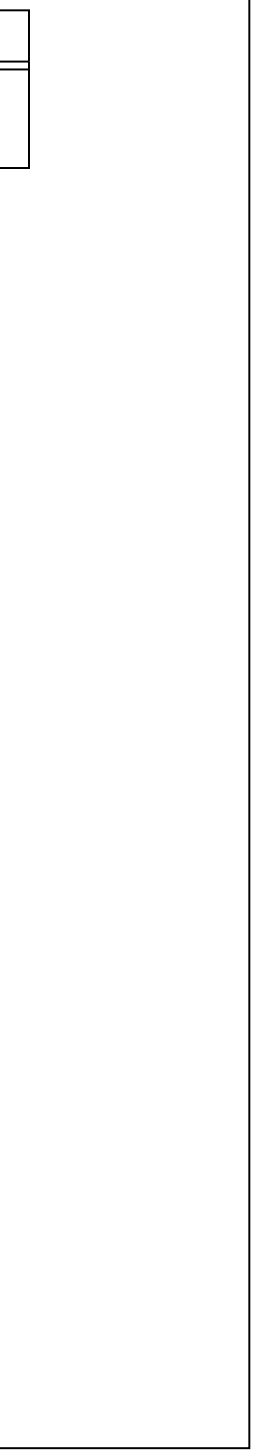

+start(Stage):void
+main():void
DungeonLoader
+ json: JSONobject
+ load: Dugeon
+loadEntity(Dungeon,JSONobject):void
+onLoad(Entity player):void
+onLoad(Wall wall):void
Extends

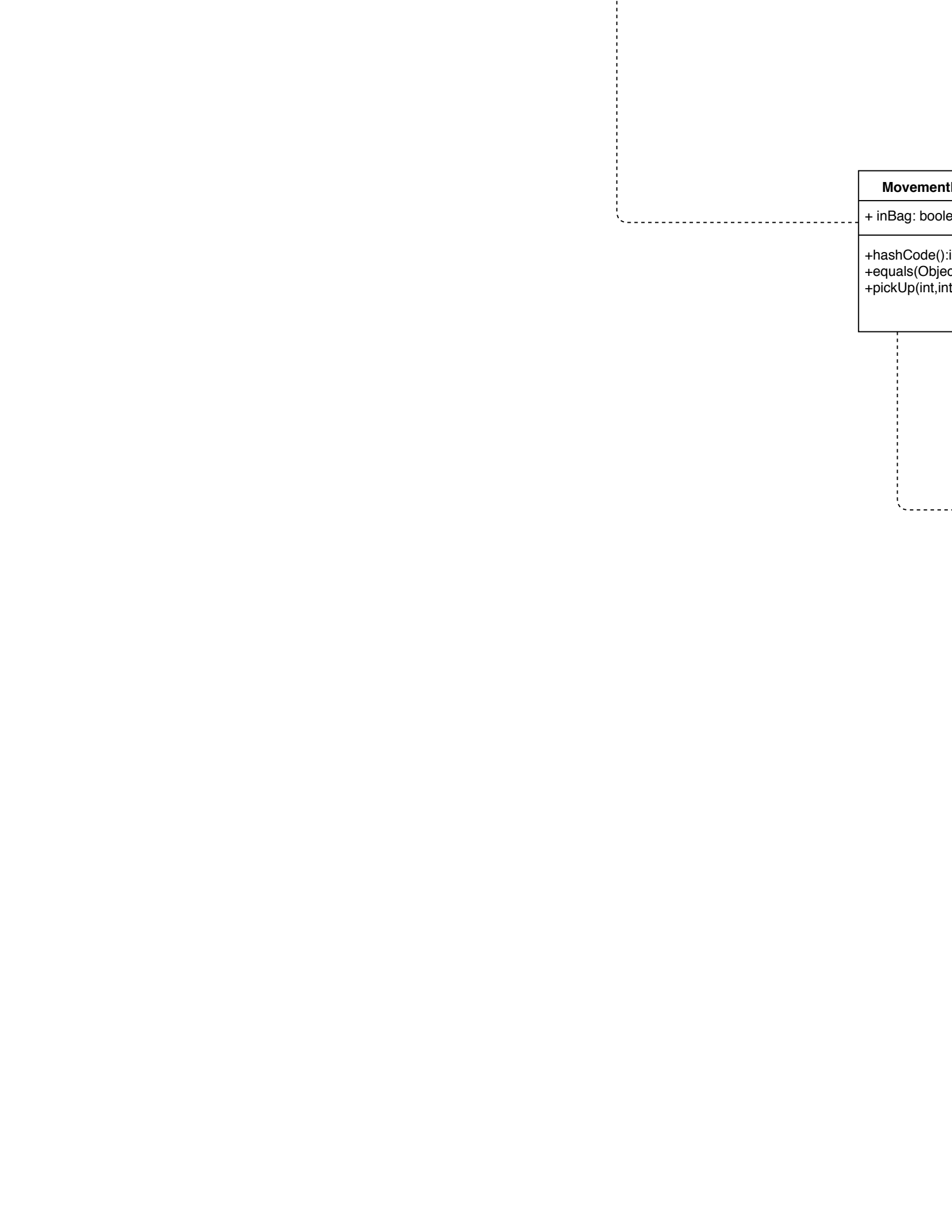

DungeonControllerLoader
+ entities: List<ImageView>
+playerImage: Image
+wallImage: Image
+onLoad(Entity):void
+onLoad(Wall):void
+addEntity(Entity, ImageView):void
+trackPosition(Entity, Node):void
+loadController():DungeonController

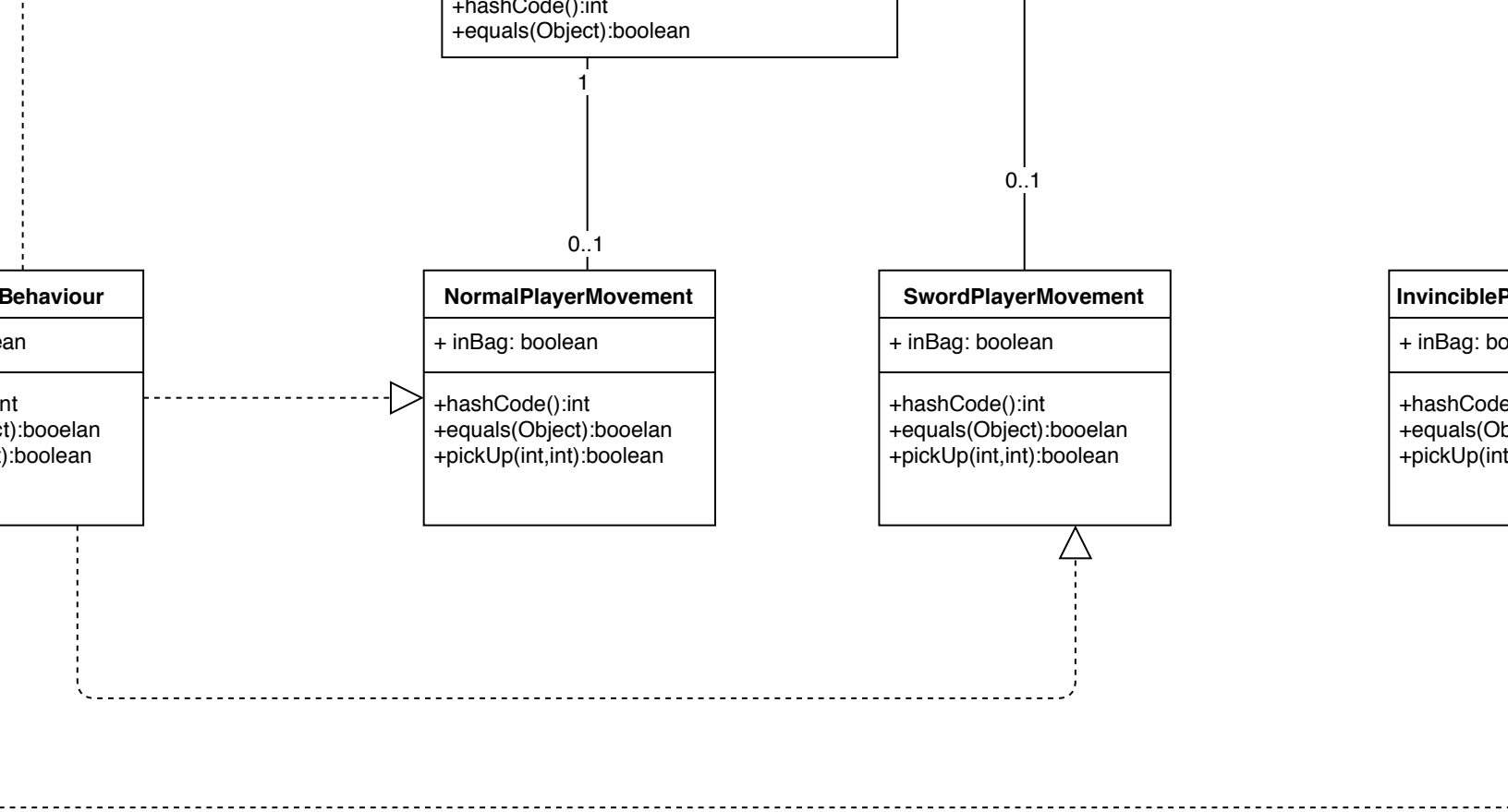

Sword
+ useleft: int
+ haveSword(): boolean
+ useSword(): boolean
+hashCode():int
+equals(Object):booelan
Wall
+hashCode():int
+equals(Object):booelan
LitBomb
+ Dungeon: dungeon
+ int: timeLeft
+ explode(): boolean
+ timeLeft(): int
+ tickDown(): void
+hashCode():int
+equals(Object):booelan
Door
+ doorID: int
+ opened: boolean
+ getDoorID(): int
+ openDoor(int): boolean
+ isOpen(): boolean
+hashCode():int
+equals(Object):booelan
Entity
- x: IntegerProperty
- y: IntegerProperty
+ x(): IntegerProperty
+ getX(): int
+ y(): IntegerProperty
+ getY(): int
+equals(Object):booelan
+pickUp(int,int):boolean
0 ... n
Enemy
+ dungeon: Dungeon
+ followPlayer(Player): void
+moveWait(): void
+moveUp():void
+moveDown():void
+moveLeft():void
+moveRight():void
+ checkKillPlayer(Player):boolean
+equals(Object):boolean

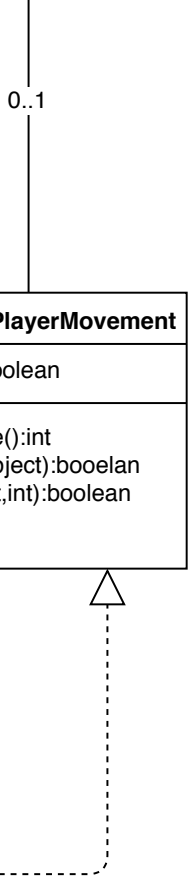

Key
+keyID:int
+ getKeyID(): int
+setKeyID(int):void
+hashCode():int
+equals(Object):booelan
Potion
+ pickUp(int,int): boolean
+hashCode():int
+equals(Object):booelan
Bomb
+ useBomb(): litBomb
+hashCode():int
+equals(Object):booelan
Treasure
+hashCode():int
+equals(Object):booelan
Boulder
geon: Dungeon
ateEnemies(): void
eWait(): void
eUp():void
eDown():void
eLeft():void
eRight():void
pleteSwitchGoal(): void
witch(): void
als(): boolean
Player
+ dungeon: Dungeon
+ enemies: ArrayList<Enemy>
+ carry: ArrayList<pickUpAble>
+ litBombs: ArrayList<LitBomb>
+ storedKey: int
+ MovementBehaviour: movementBehaviour
+moveWait(): void
+moveUp():void
+moveDown():void
+moveLeft():void
+moveRight():void
+ switchToNormal(): void
+ updateEnemies(): void
+addEnemy(Enemy):void
+setKey(int):void
+ getStoredKey(): void
+haveKey(int):boolean
+openDoor(Door): void
+pickUp(pickUpAble):boolean
1
1
1


+tryPick():void
+ useSword():boolean
+ useInvincibility(): void
+ useBomb(): void
+ litBomb(int, int):boolean
+ notifyBombs(): void
+ completeExitGoal(): void




Movement
+ inBag: boole
+hashCode():i
+equals(Objec
+pickUp(int,int


+hashCode():int
+equals(Object):boolean
1
0..1
0..1
Behaviour
ean
nt
ct):booelan
):boolean
NormalPlayerMovement
+ inBag: boolean
+hashCode():int
+equals(Object):booelan
+pickUp(int,int):boolean
SwordPlayerMovement
+ inBag: boolean
+hashCode():int
+equals(Object):booelan
+pickUp(int,int):boolean
InvincibleP
+ inBag: bo
+hashCode
+equals(Ob
+pickUp(int


0..1
PlayerMovement
oolean
e():int
ject):booelan
,int):boolean