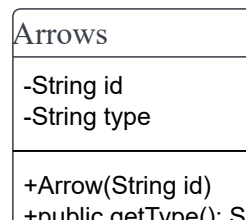
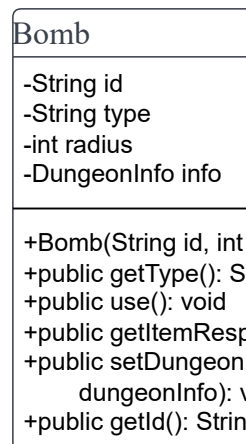
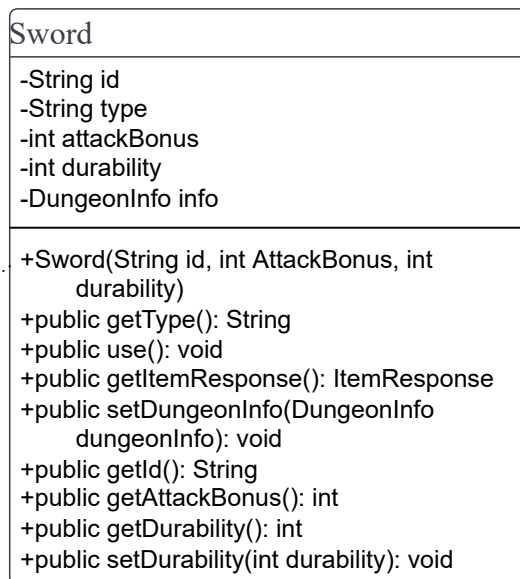
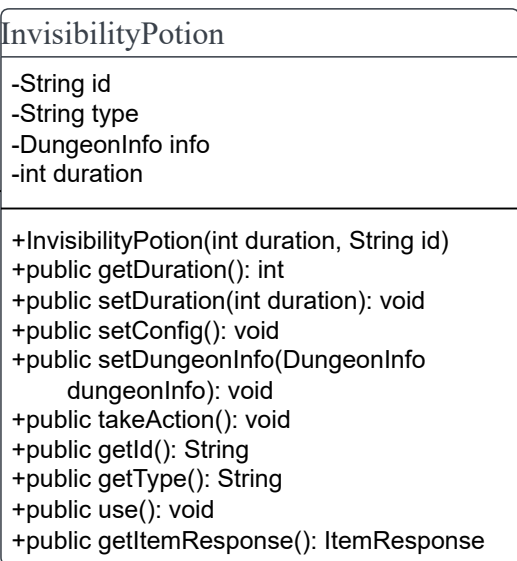
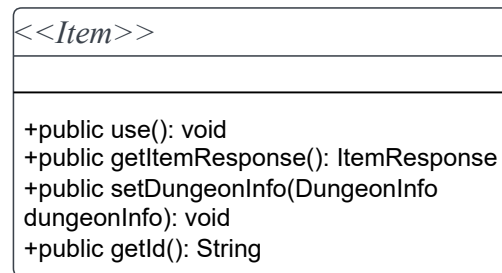
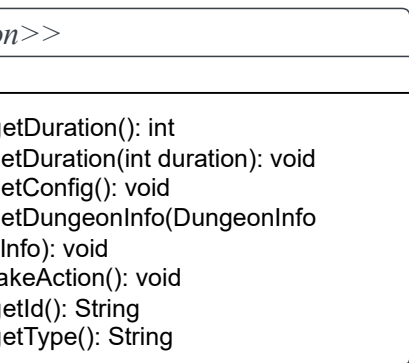
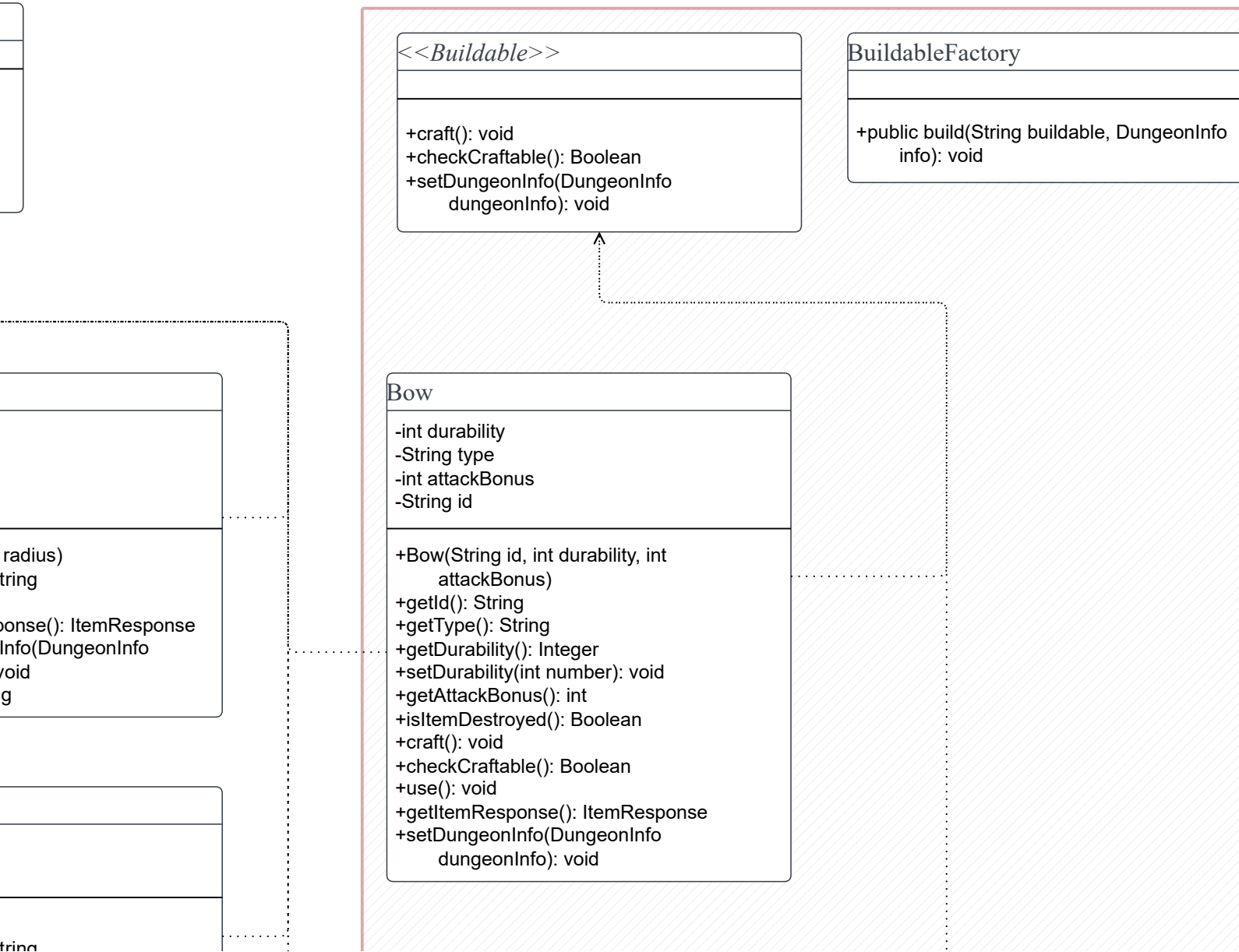


<<Potio

+public g  
+public s  
+public s  
+public s  
+public s  
dungeon  
+public ta  
+public g  
+public g





## DungeonMain

-String dungeon  
-String dungeon  
-int DungeonCo  
-HashMap<Stri

+public getSkin  
+public getLoca  
+public static d  
+public static c  
+public newGar  
String con  
+public getDun  
DungeonF  
+public tick(Stri  
DungeonF  
+public tick(Dire  
DungeonF  
+public build(ST  
DungeonF  
+public interact  
DungeonF  
+public getTrea  
+public remove  
+public isNearS  
boolean  
+public hasWea

## DungeonInfo

-HashMap<String, Entity> entityMap  
-HashMap<String, Integer> configMap  
-List<InvItem> itemList  
-Goal dungeonGoal  
-List<BattleResponse> battleList  
-List<Tick> tickList  
-int entityCounter

+public storeEntitiesInMap(JSONArray arr):  
void  
+public static createEntity(JSONObject json,  
String id, DungeonInfo info): Entity  
+public getListEntityResponse():  
List<EntityResponse>  
+public setConfigs(JSONObject config): void  
+ public getSpecificConfig(String name): int  
+public getEntitiesStringByPosition(Position  
pos): List<String>  
+public getEntitiesByPosition(Position pos):  
List<Entity>  
+public movePPlayer(Direction d): void  
+public getPlayer(): Player  
+public getEntityMap(): HashMap<String,  
Entity>  
+public getConfigMap(): HashMap<String,  
Integer>  
+public getItemList(): List<InvItem>  
+public getListItemResponse():  
List<ItemResponse>  
+public moveAllMovingEntity(): void  
+public getAllMovingEntity(): List<Moving>  
+public getAllMercenary(): List<Mercenary>  
+public getAllZombie(): List<ZombieToast>  
+public initSpawnConfig(): void  
+public isItemInList(String id): boolean  
+public getItemById(String id):  
+public isItemAllowed(String id, List<String>  
allowedList): boolean  
+public getInvItemIdsListByType(String  
type): List<String>  
+public removeInvItemById(String id): void  
+public getNumInvItemType(String type): int  
+public addInvItem(InvItem item): void  
+public Spawn(): void  
+public zombieSpawn(Position p): void  
+public spiderSpawn(): void  
+public storeGoals(JSONObject jsonGoals):  
void  
+ public getGoalsFromJson(JSONObject  
jsonGoals): Goal  
+public getGoalString(): String  
+public addBattleResponse(BattleResponse  
response): void  
+ public getBattleResponses():  
List<BattleResponse>  
+public addTick(Tick tickableEntity): void  
+public getTickableEntityList(): List<TickableEntity>

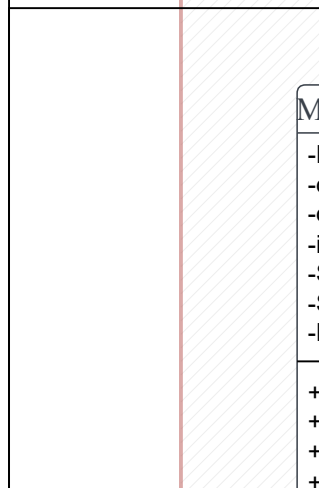
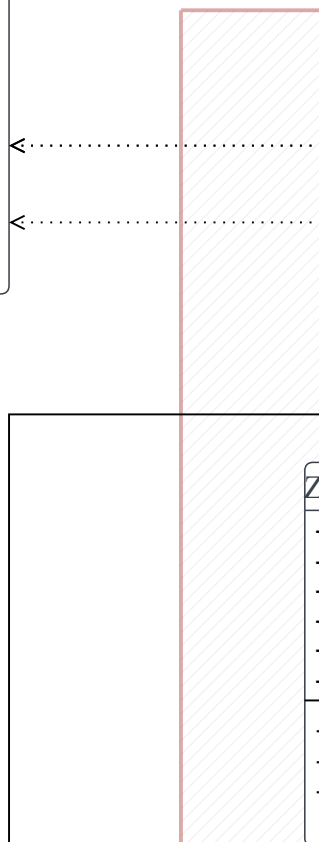
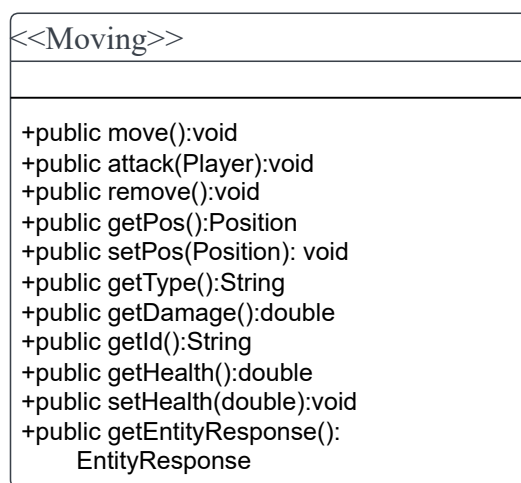
```

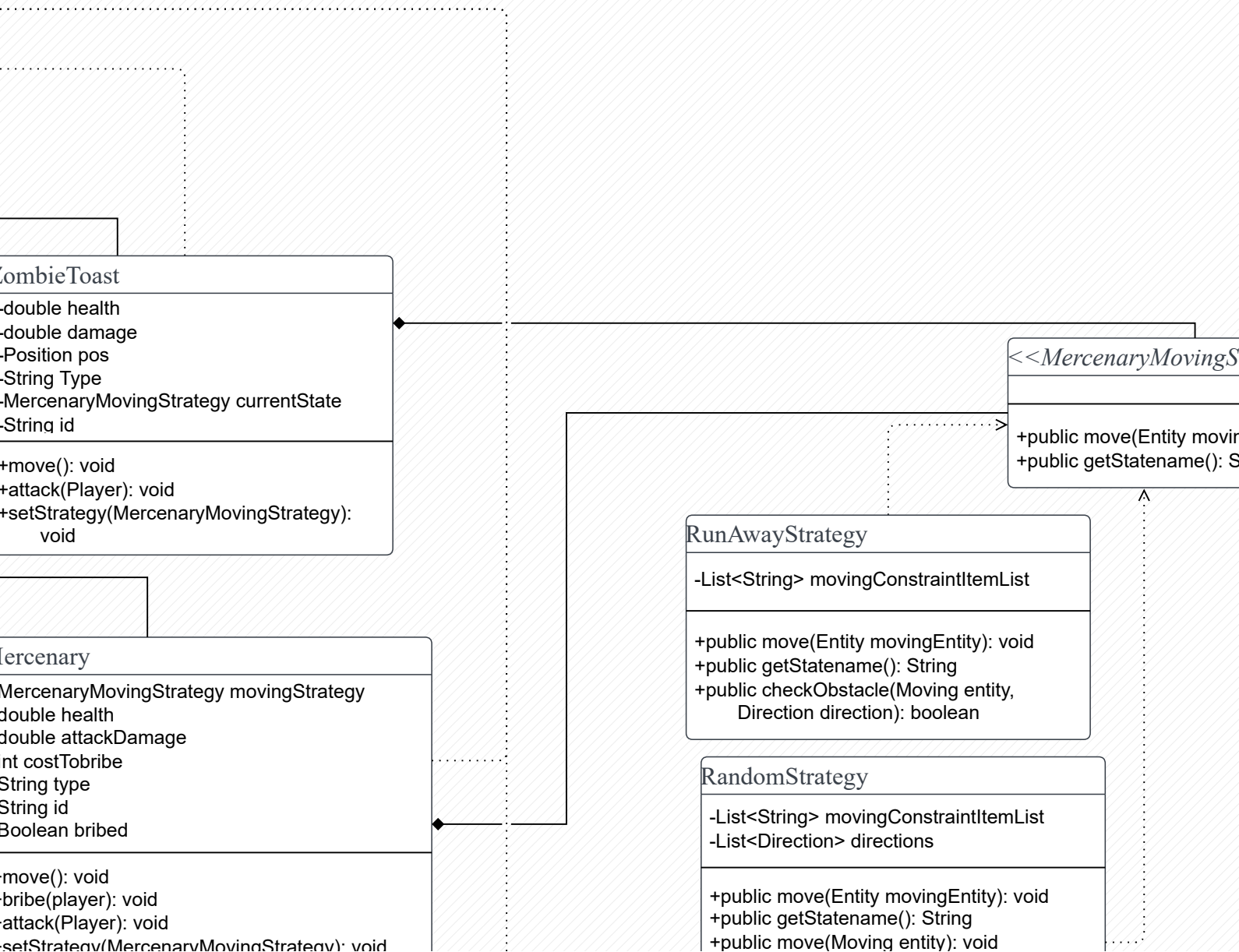
id
hName
ounter
ng, DungeonInfo> infoMap

(): String
alisation(): String
ungeons(): List<String>
onfigs(): List<String>
me(String dungeonName,
figName): DungeonResponse
geonResponseModel():
Response
ng itemUsedId):
Response
ection movementDirection):
Response
tring buildable):
Response
(String entityId):
Response
asureCount(): int
Treasure(int count): void
awner(Position awner):
apon(): boolean

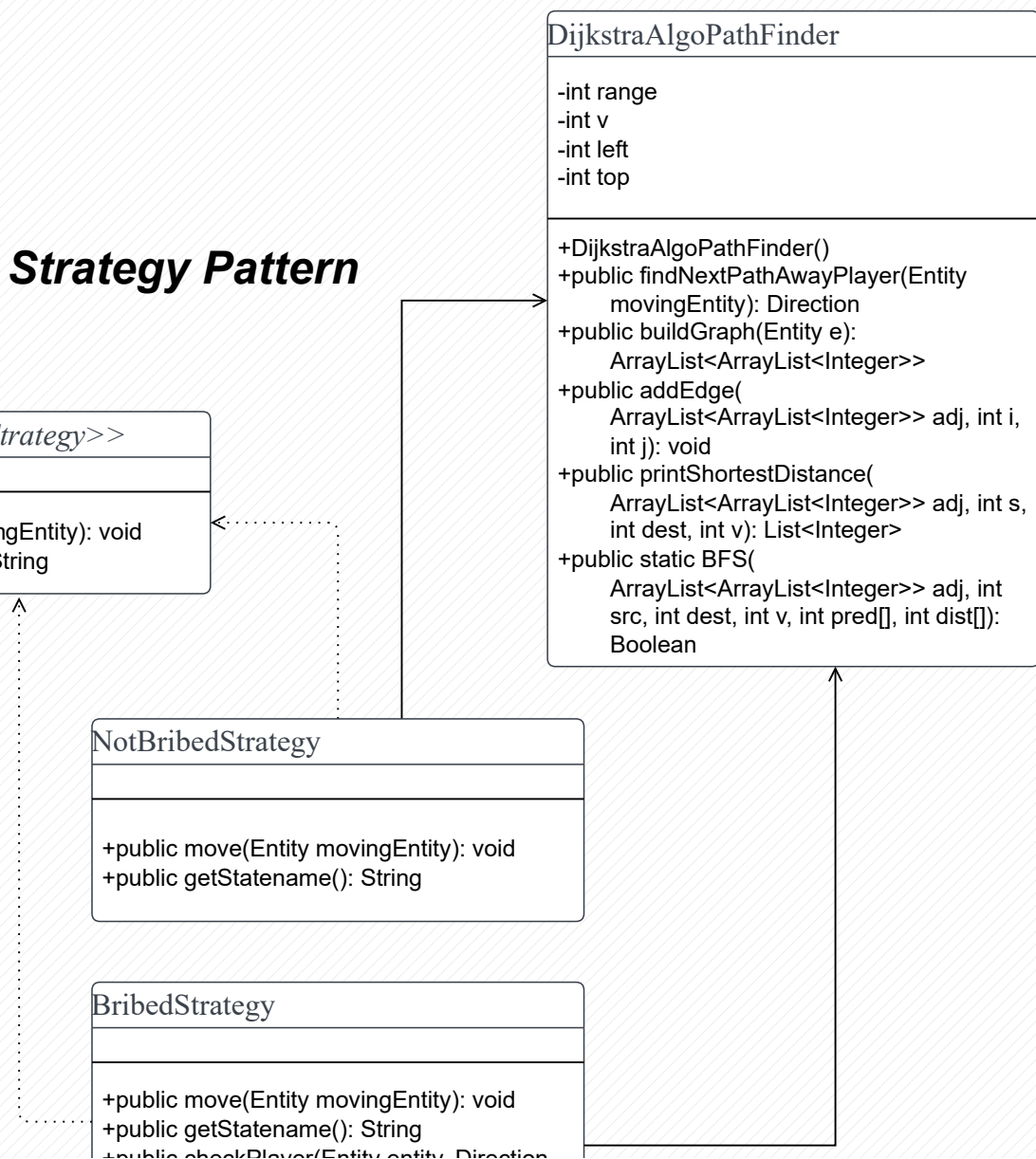
```

```
id  
name  
counter  
ing, DungeonInfo> infoMap  
  
((): String  
alisation(): String  
ungeons(): List<String>  
onfigs(): List<String>  
me(String dungeonName,  
figName): DungeonResponse  
geonResponseModel()):  
response  
ng itemUsedId):  
response  
ection movementDirection):  
Response  
tring buildable):  
Response  
(String entityId):  
response  
sureCount(): int  
Treasure(int count): void  
$awner(Position Spawner):  
  
apon(): boolean
```



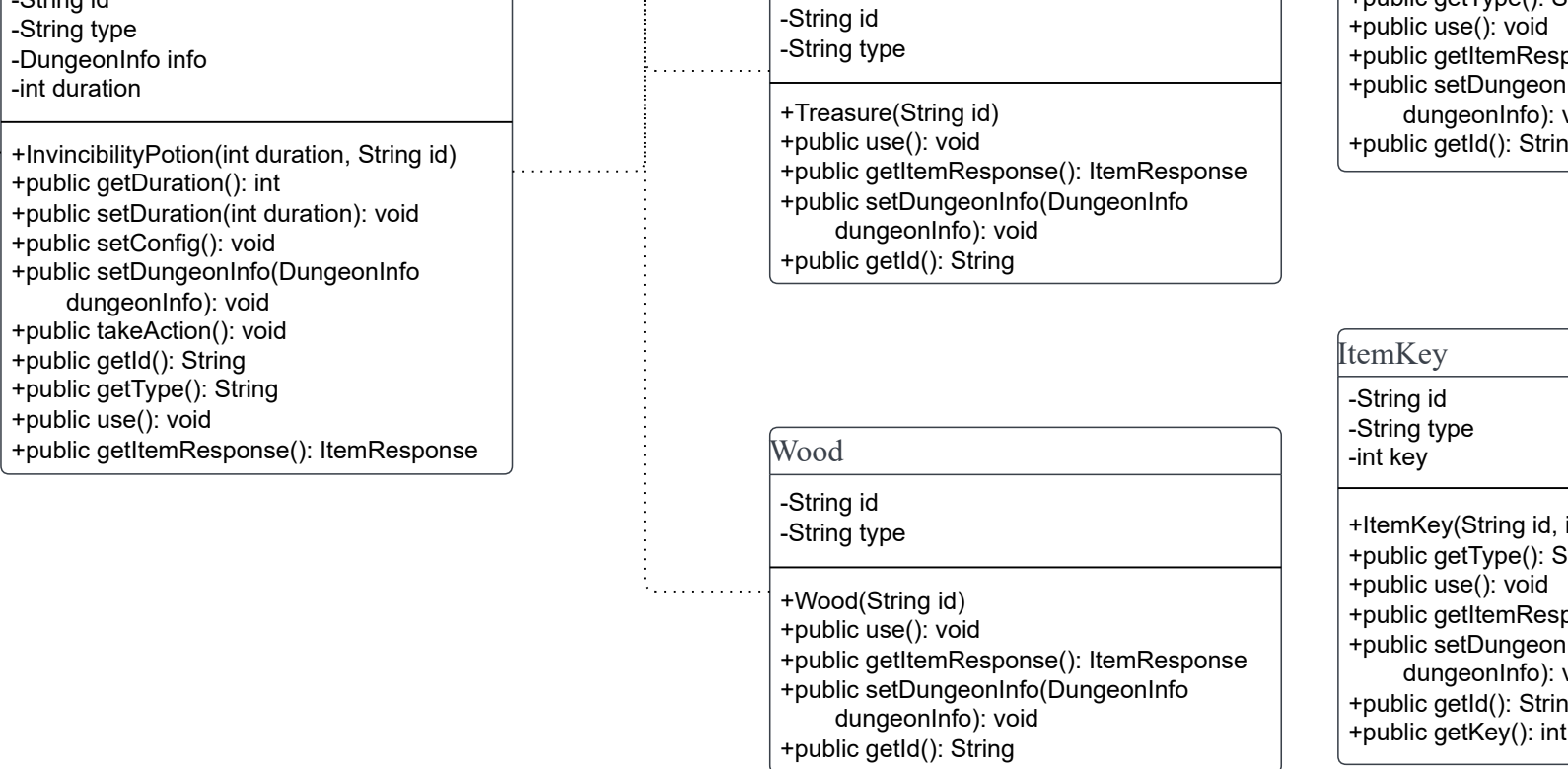


# Strategy Pattern

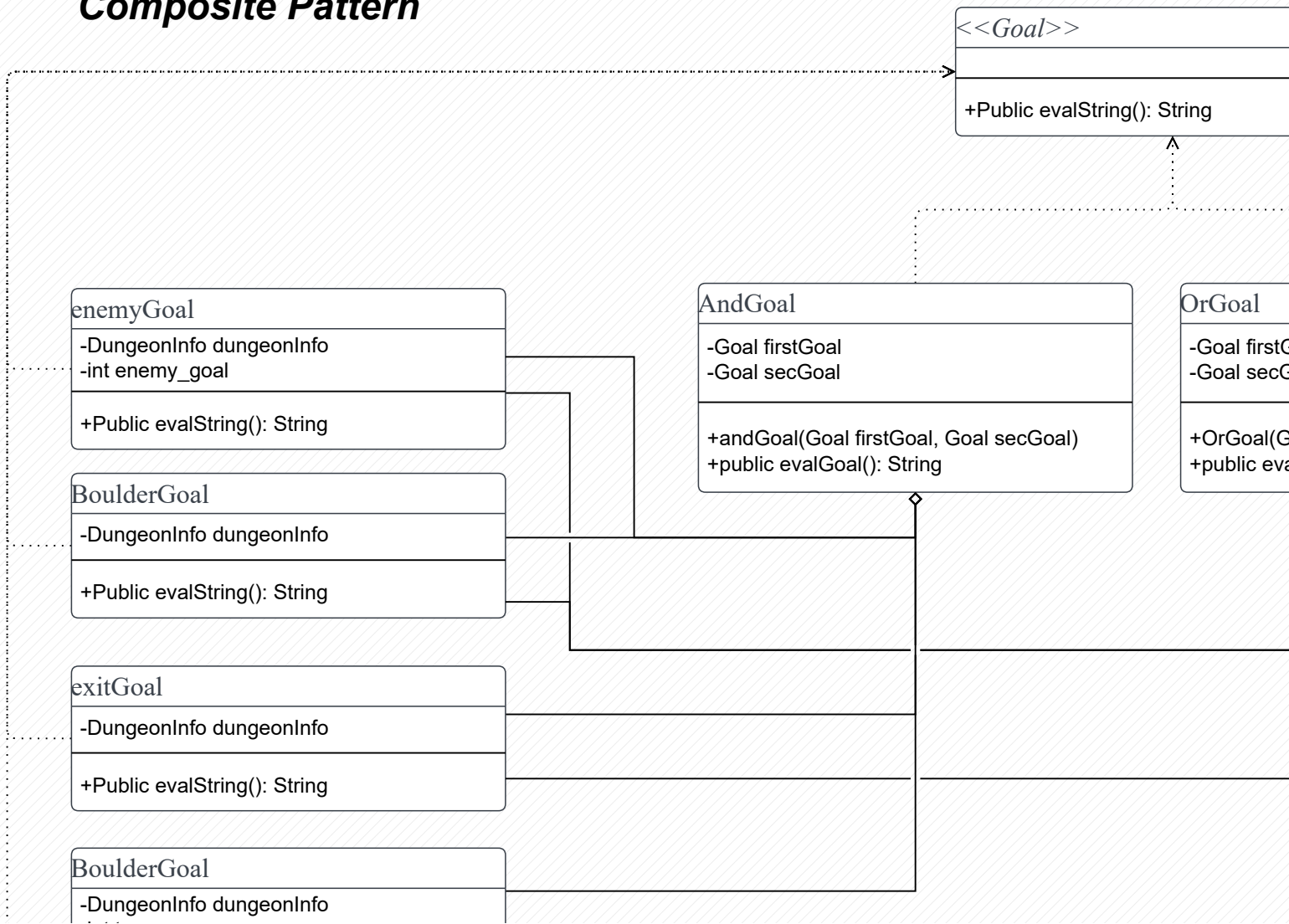








## Composite Pattern





+public getTickList(): List<Tick>  
+public runTicks(): void  
+public getCurrentBuildables(): List<String>

*Entity*

-DungeonInfo du

+getId(): String  
+getType():String  
+getPos():Position  
+getEntityRespon  
+setConfig(): void  
+setPos():void  
+setDungeonInfo  
dungeonInfo  
+getDungeonInfo

Battle

-Player player  
-Moving enemy  
-DungeonInfo info

+public Battle(Player player, Moving enemy,  
DungeonInfo info)  
+public start(): BattleResponse

Player

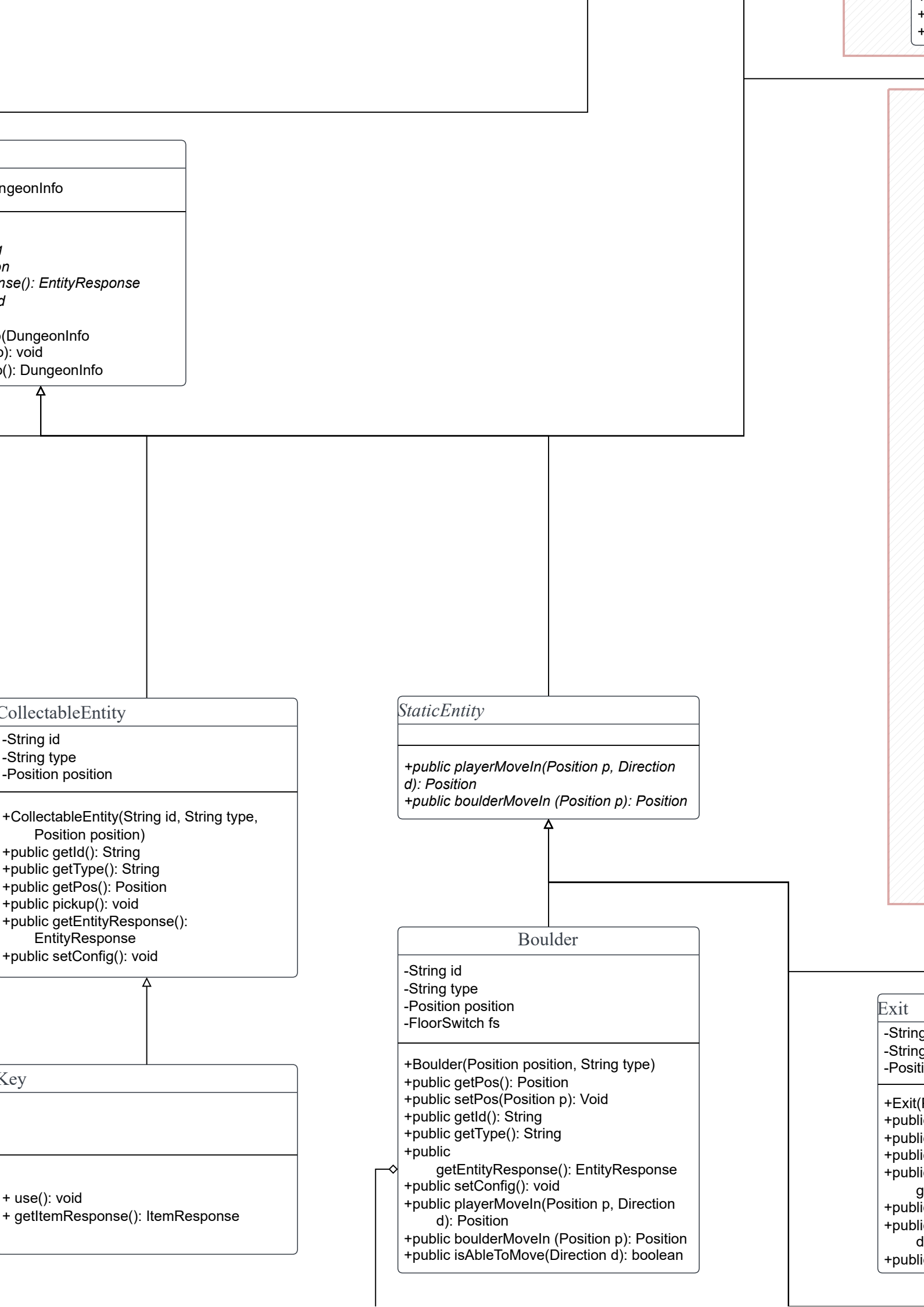
-String id  
-String type  
-double attack  
-double health  
-Position position  
-PlayerState playerState  
-Queue<Potion> potionUsedQueue

+Player(Position position, String id)  
+public setConfig(): void  
+public getAttack(): double  
+public setAttack(double attack): void  
+public getHealth(): double  
+public setHealth(double health): void  
+public reduceHealth(double health): void  
+public getPos(): Position  
+public setPos(int x, int y): void  
+public setPos(Position position): void  
+public move(Direction direction): void  
+public isAlive(): Boolean  
+public getId(): String  
+public getType(): String  
+public getEntityResponse():  
EntityResponse  
+public getPlayerState(): PlayerState  
+public addPotion(Potion potion): void  
+public pullPotion(): Potion  
+public tickPlayerState(): void  
+public isInvincible(): Boolean

InvincibleState

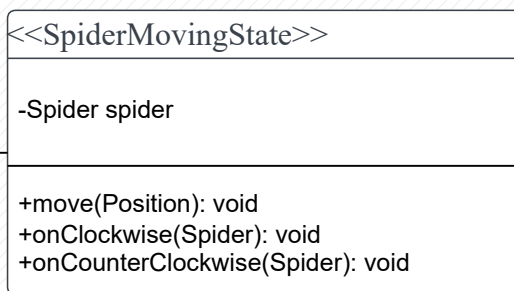
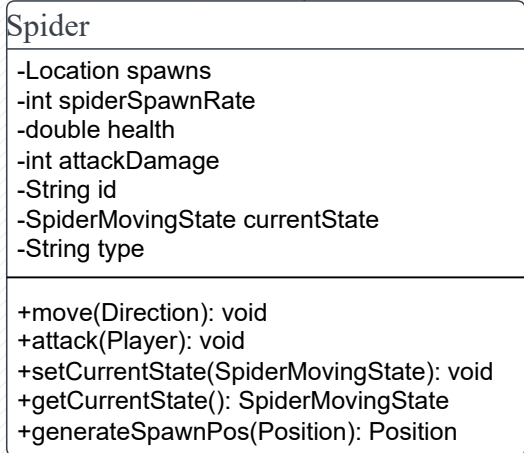
-Player player  
-double attack  
-double health  
-int potionTime

+public InvincibleState(Player player)  
+public getStateName(): String  
+public getAttack(): double  
+public getHealth(): double  
+public getPosition(): Position  
+public getPotionTime(): int

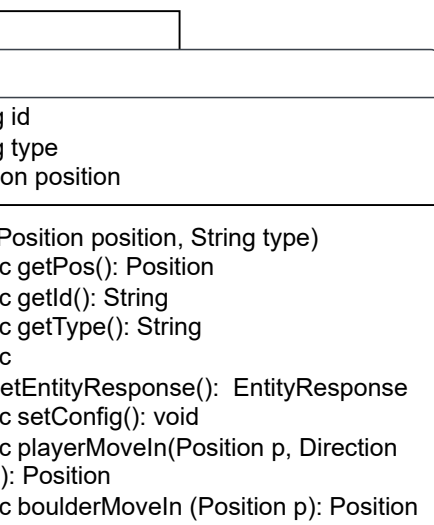
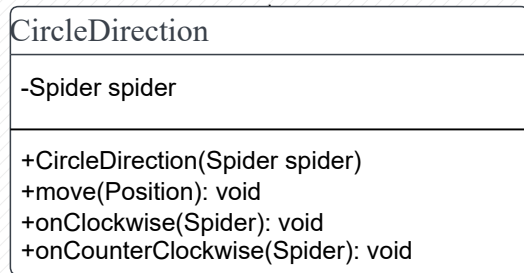
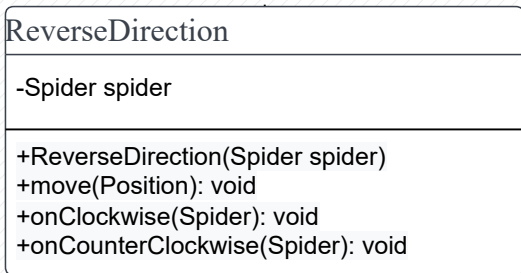


getStrategy(MercenaryMovingStrategy): void  
getCurrentState(): MercenaryMovingStrategy  
getBribed(): boolean

+public checkObstacle(Moving entity,  
Direction direction): boolean



**State Pattern**



+public checkPlayer(Entity entity, Direction  
direction): boolean

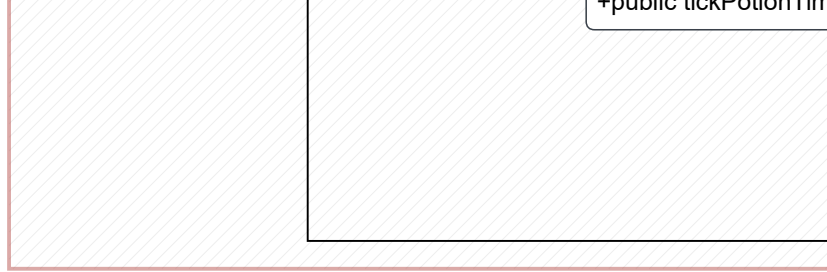
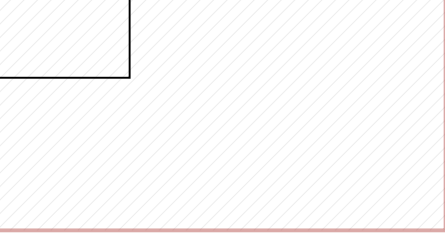
**ern**



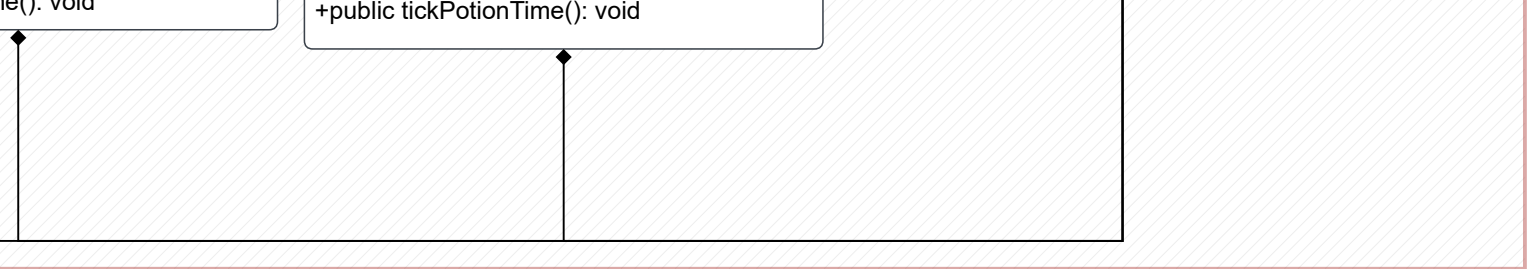
.....-int treasure

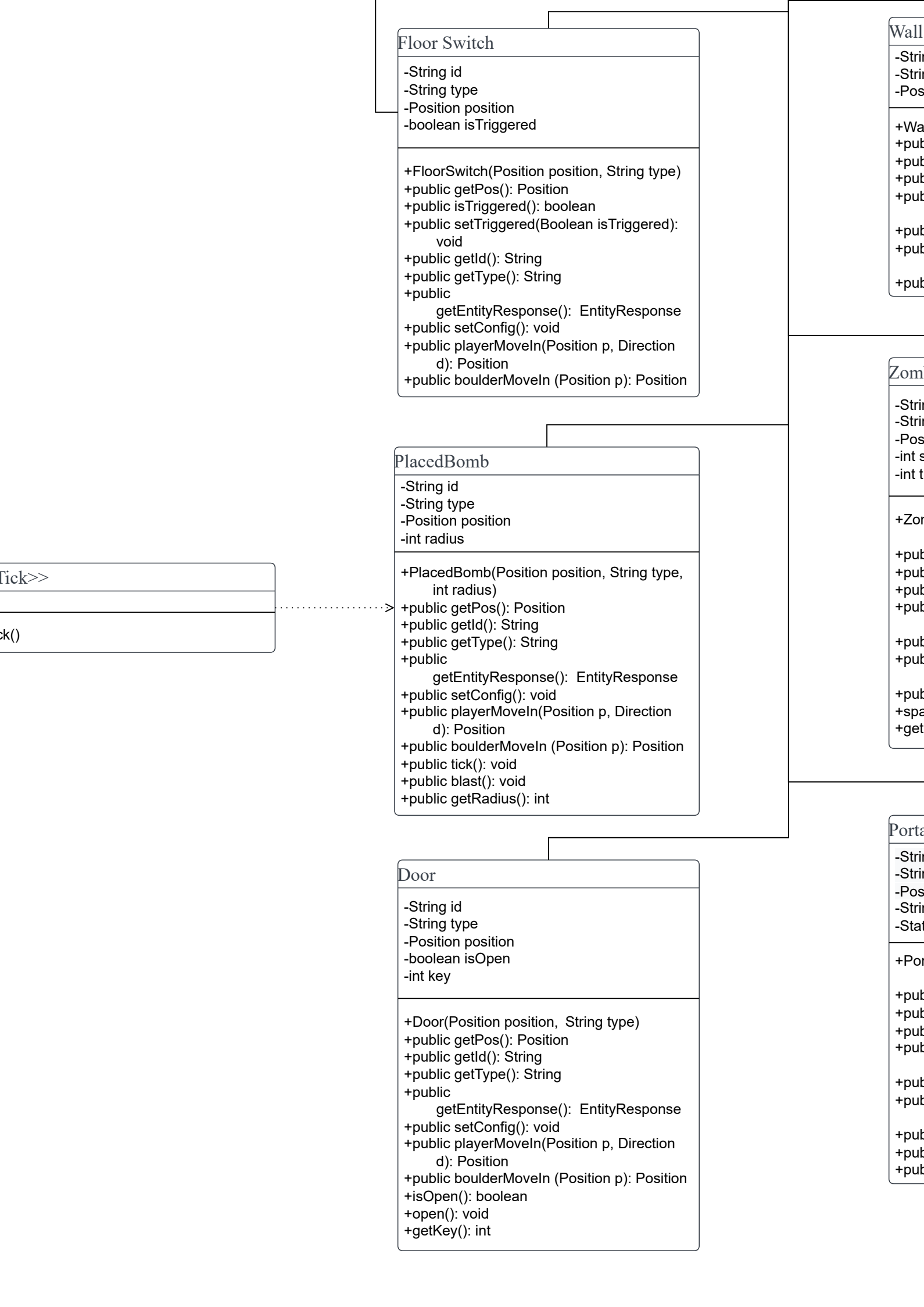
+Public evalString(): String





+public tickPotion m





ng id ng type ition position	
ll(Position position, String type) blic getPos(): Position blic getId(): String blic getType(): String blic getEntityResponse(): EntityResponse blic setConfig(): void blic playerMoveIn(Position p, Direction d): Position blic boulderMoveIn (Position p): Position	

blicToastSpawner	
ng id ng type ition position spawnTime imeToSpawn	
mblicToastSpawner(Position position, String type) blic getPos(): Position blic getId(): String blic getType(): String blic getEntityResponse(): EntityResponse blic setConfig(): void blic playerMoveIn(Position p, Direction d): Position blic boulderMoveIn (Position p): Position awn(): void SpawnPos(): Position	

al	
ng id ng type ition position ng colour ic List<Portal> portalList	
rtal(Position position, String type, String colour) blic getPos(): Position blic getId(): String blic getType(): String blic getEntityResponse(): EntityResponse blic setConfig(): void blic playerMoveIn(Position p, Direction d): Position blic boulderMoveIn (Position p): Position blic getColour(): String blic telerportTo(Position p): Position	

