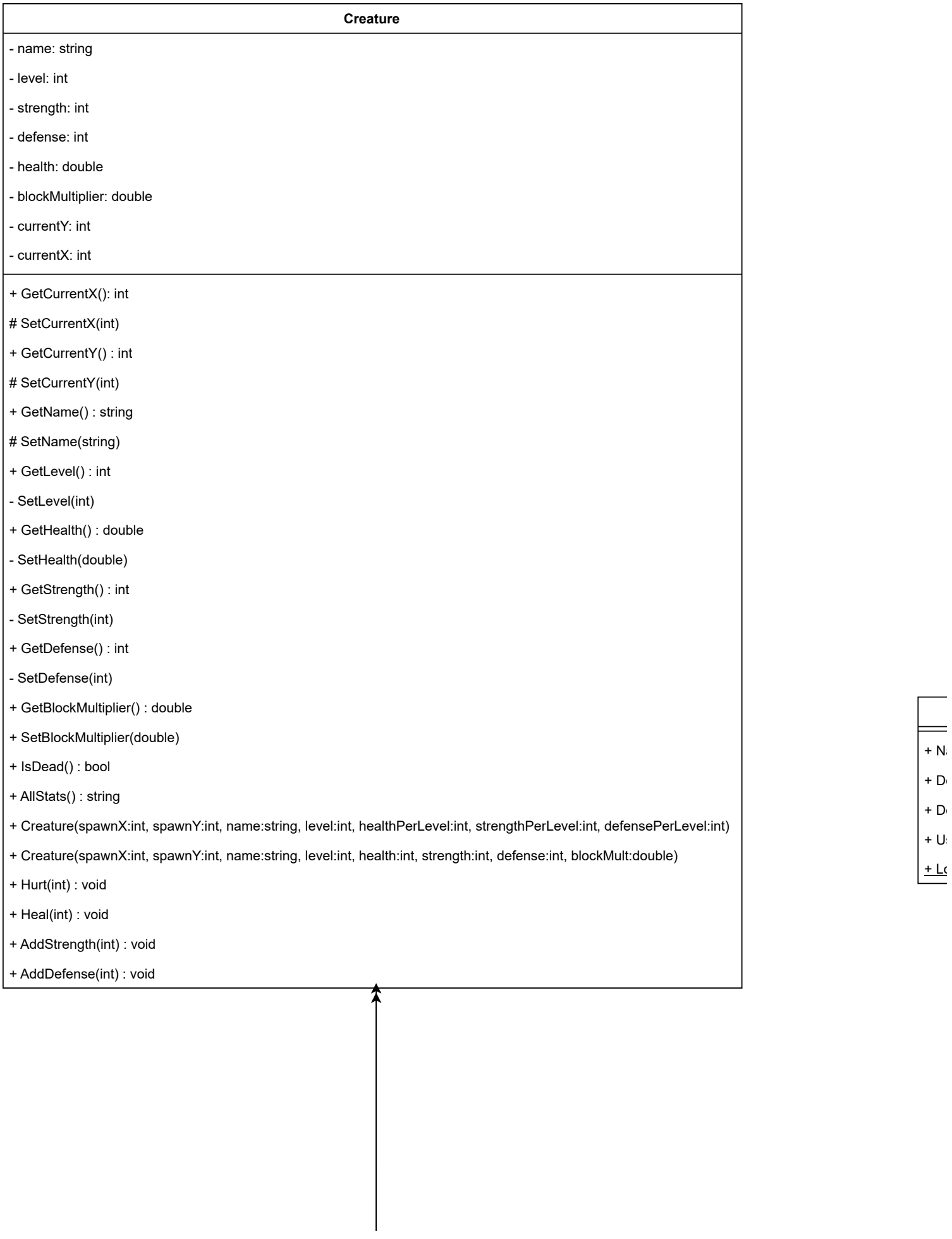
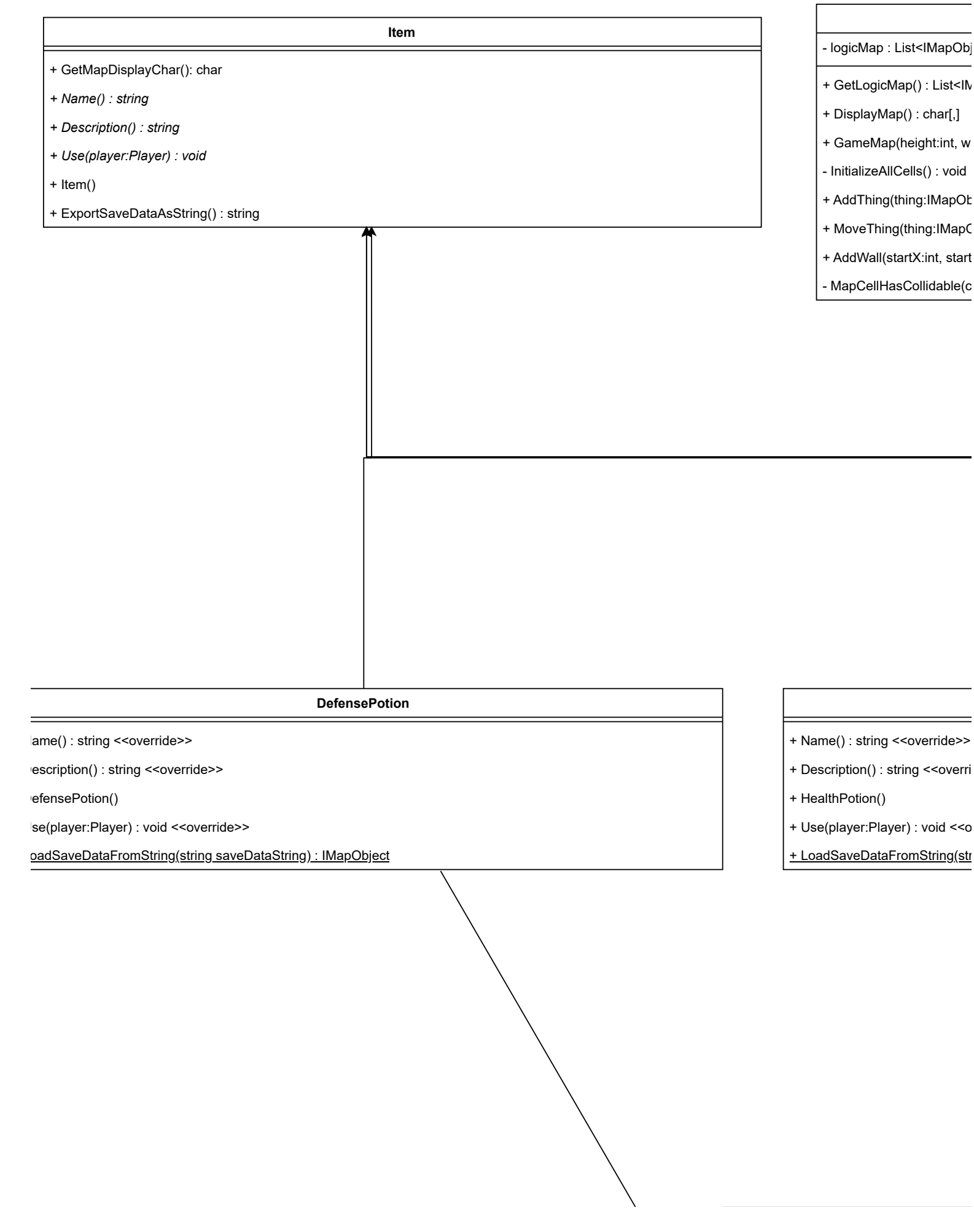
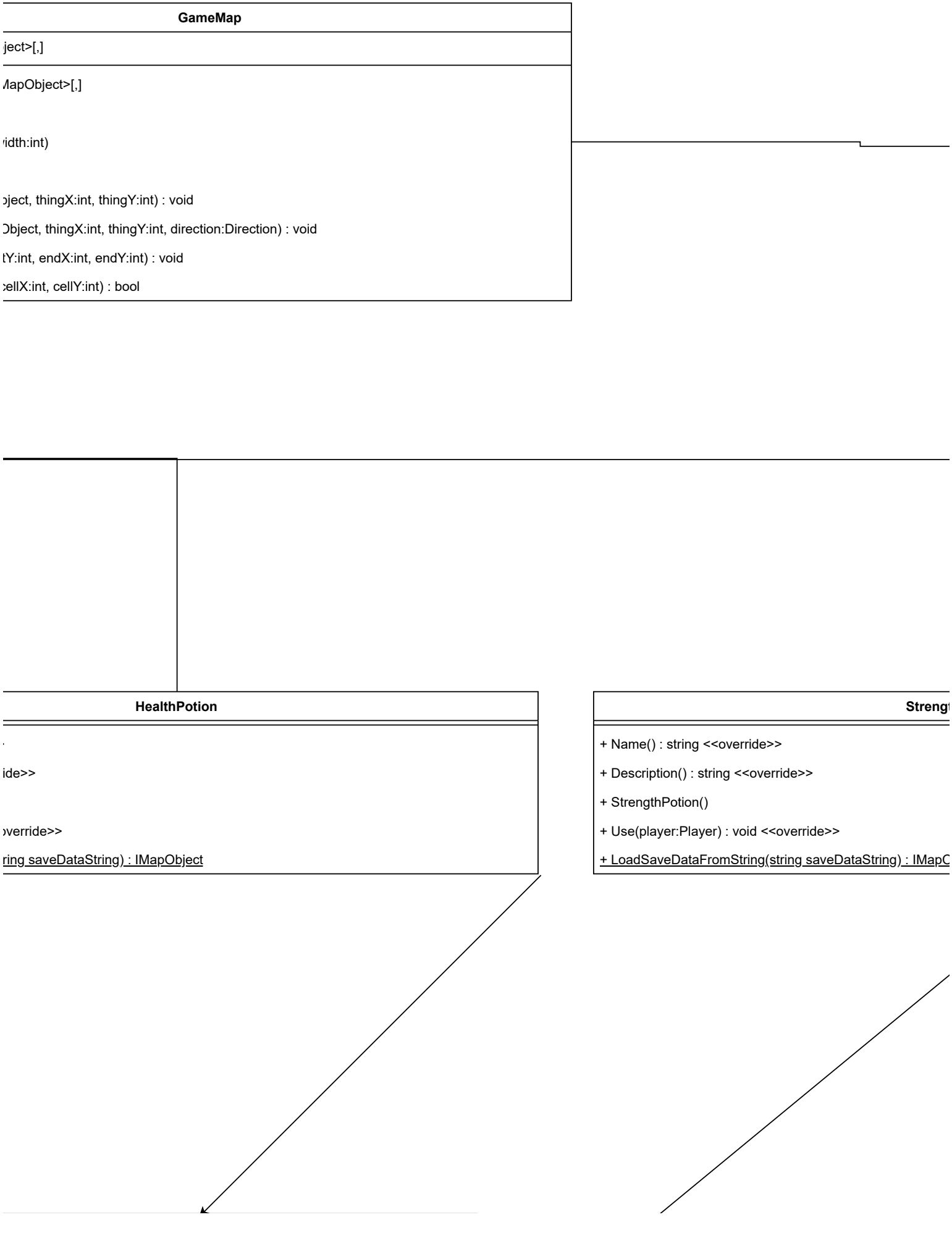
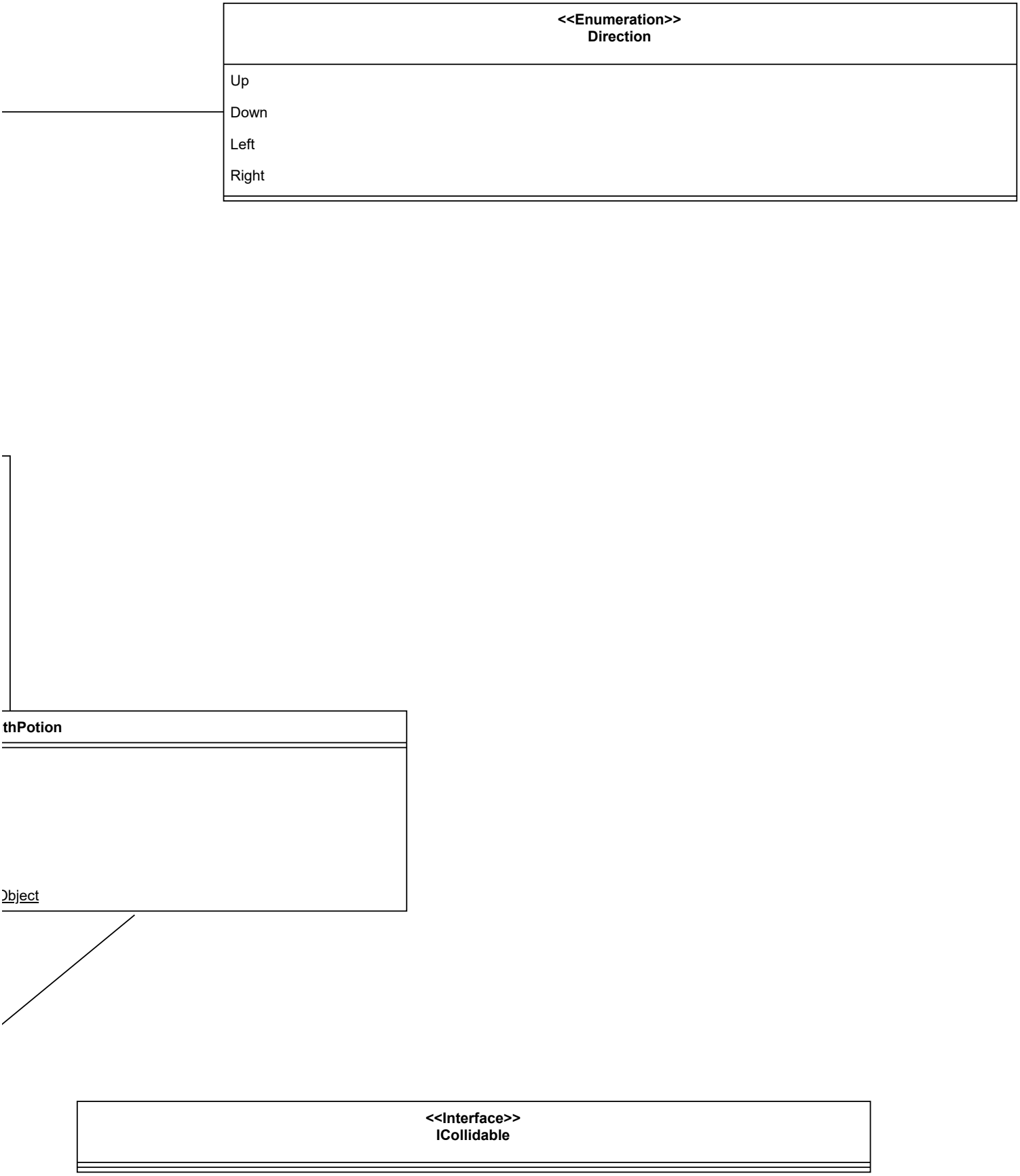
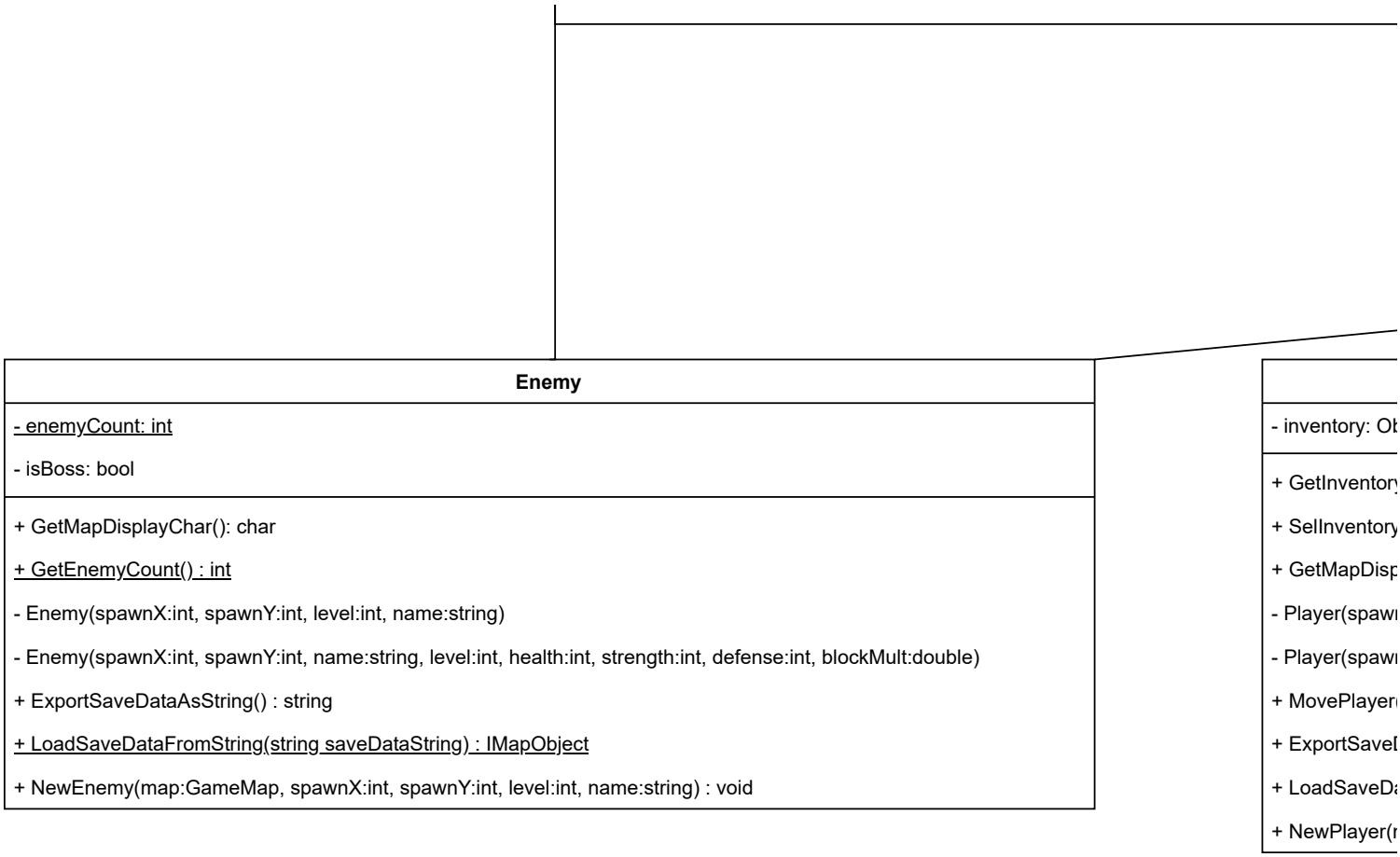
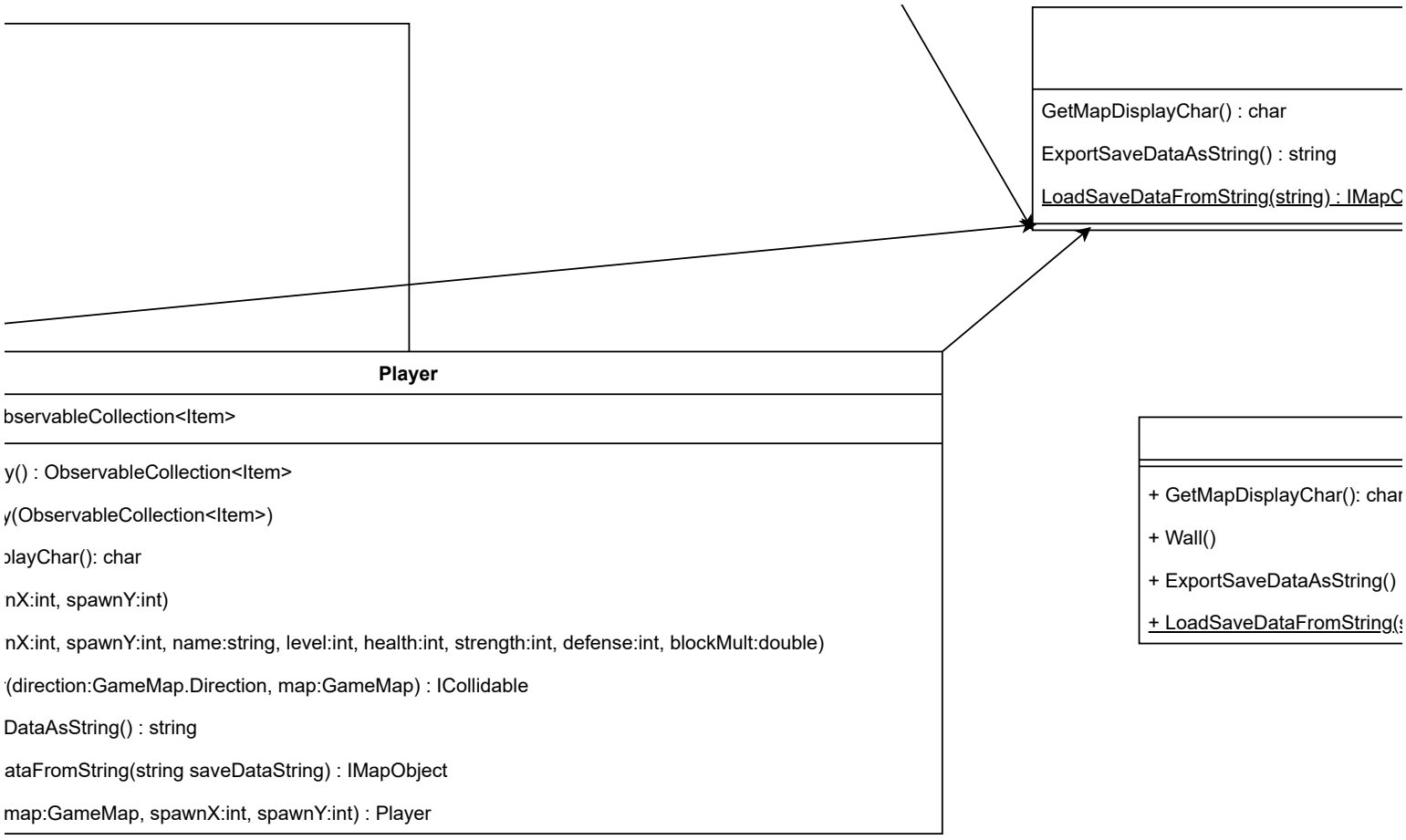

Creature
- name: string
- level: int
- strength: int
- defense: int
- health: double
- blockMultiplier: double
- currentY: int
- currentX: int
+ GetCurrentX(): int
# SetCurrentX(int)
+ GetCurrentY() : int
# SetCurrentY(int)
+ GetName() : string
# SetName(string)
+ GetLevel() : int
- SetLevel(int)
+ GetHealth() : double
- SetHealth(double)
+ GetStrength() : int
- SetStrength(int)
+ GetDefense() : int
- SetDefense(int)
+ GetBlockMultiplier() : double
+ SetBlockMultiplier(double)
+ IsDead() : bool
+ AllStats() : string
+ Creature(spawnX:int, spawnY:int, name:string, level:int, healthPerLevel:int, strengthPerLevel:int, defensePerLevel:int)
+ Creature(spawnX:int, spawnY:int, name:string, level:int, health:int, strength:int, defense:int, blockMult:double)
+ Hurt(int) : void
+ Heal(int) : void
+ AddStrength(int) : void
+ AddDefense(int) : void

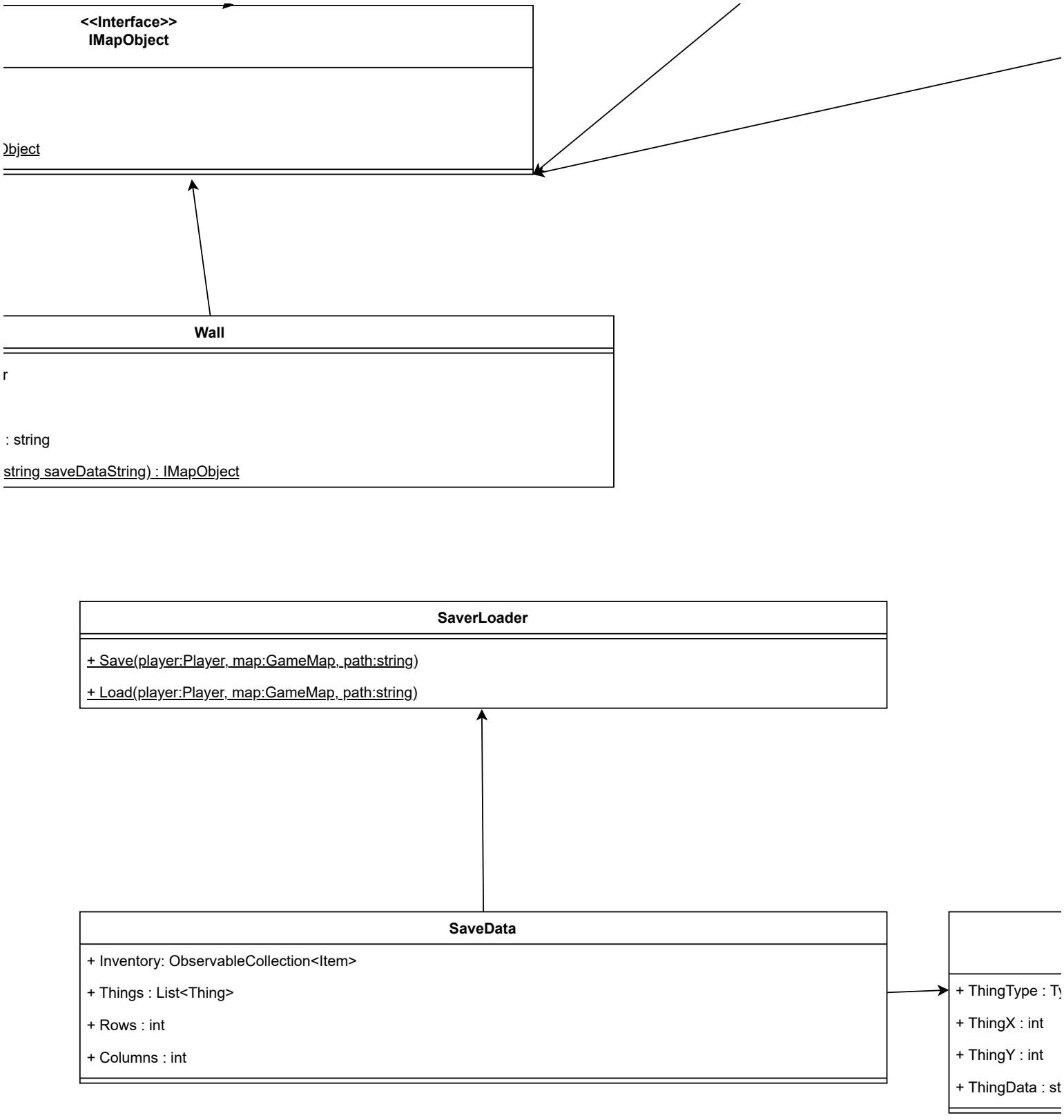

Item
+ GetMapDisplayChar(): char
+ Name() : string
+ Description() : string
+ Use(player:Player) : void
+ Item()
+ ExportSaveDataAsString() : string
- logicMap : List<IMapObj
+ GetLogicMap() : List<IM
+ DisplayMap() : char[,]
+ GameMap(height:int, w
- InitializeAllCells() : void
+ AddThing(thing:IMapOb
+ MoveThing(thing:IMapO
+ AddWall(startX:int, start
- MapCellHasCollidable(c
DefensePotion
ame() : string <<override>>
escription() : string <<override>>
efensePotion()
se(player:Player) : void <<override>>
oadSaveDataFromString(string saveDataString) : IMapObject
+ Name() : string <<override>>
+ Description() : string <<overri
+ HealthPotion()
+ Use(player:Player) : void <<o
+ LoadSaveDataFromString(str

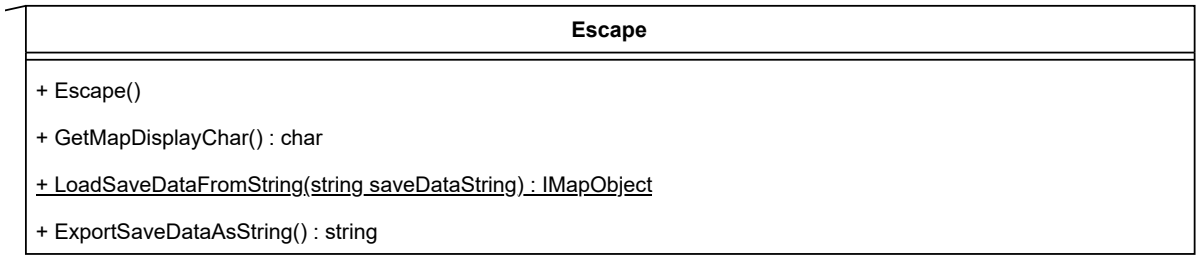

GameMap
ject>[,]
MapObject>[,]
idth:int)
oject, thingX:int, thingY:int) : void
Object, thingX:int, thingY:int, direction:Direction) : void
tY:int, endX:int, endY:int) : void
cellX:int, cellY:int) : bool
HealthPotion
ide>>
verride>>
ring saveDataString) : IMapObject
Strengt
+ Name() : string <<override>>
+ Description() : string <<override>>
+ StrengthPotion()
+ Use(player:Player) : void <<override>>
+ LoadSaveDataFromString(string saveDataString) : IMapO

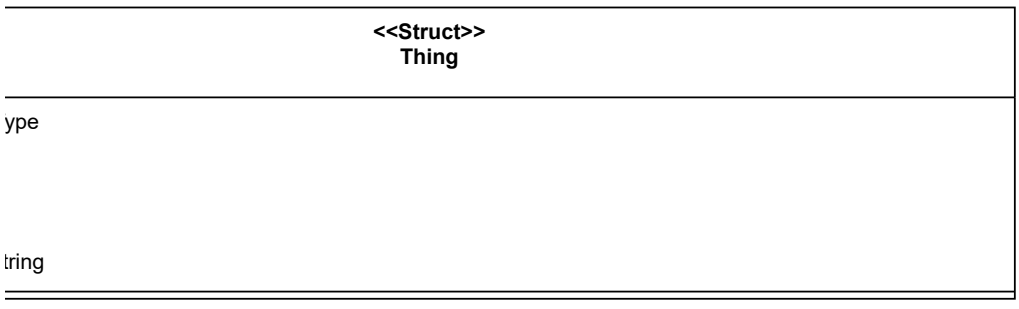

<<Enumeration>>
Direction
Up
Down
Left
Right
thPotion
Object
<<Interface>>
ICollidable


Enemy
- enemyCount: int
- isBoss: bool
+ GetMapDisplayChar(): char
+ GetEnemyCount() : int
- Enemy(spawnX:int, spawnY:int, level:int, name:string)
- Enemy(spawnX:int, spawnY:int, name:string, level:int, health:int, strength:int, defense:int, blockMult:double)
+ ExportSaveDataAsString() : string
+ LoadSaveDataFromString(string saveDataString) : IMapObject
+ NewEnemy(map:GameMap, spawnX:int, spawnY:int, level:int, name:string) : void
- inventory: Ob
+ GetInventory
+ SelInventory
+ GetMapDisp
- Player(spawn
- Player(spawn
+ MovePlayer(
+ ExportSaveD
+ LoadSaveDa
+ NewPlayer(n


GetMapDisplayChar() : char
ExportSaveDataAsString() : string
LoadSaveDataFromString(string) : IMapC
Player
bservableCollection<Item>
y() : ObservableCollection<Item>
y(ObservableCollection<Item>)
olayChar(): char
nX:int, spawnY:int)
nX:int, spawnY:int, name:string, level:int, health:int, strength:int, defense:int, blockMult:double)
(direction:GameMap.Direction, map:GameMap) : ICollidable
DataAsString() : string
ataFromString(string saveDataString) : IMapObject
map:GameMap, spawnX:int, spawnY:int) : Player
+ GetMapDisplayChar(): char
+ Wall()
+ ExportSaveDataAsString()
+ LoadSaveDataFromString(


<<Interface>>
IMapObject
)bject
Wall
r
: string
string saveDataString) : IMapObject
SaverLoader
+ Save(player:Player, map:GameMap, path:string)
+ Load(player:Player, map:GameMap, path:string)
SaveData
+ Inventory: ObservableCollection<Item>
+ Things : List<Thing>
+ Rows : int
+ Columns : int
+ ThingType : Ty
+ ThingX : int
+ ThingY : int
+ ThingData : st


Escape
+ Escape()
+ GetMapDisplayChar() : char
+ LoadSaveDataFromString(string saveDataString) : IMapObject
+ ExportSaveDataAsString() : string


<<Struct>>
Thing
ype
tring