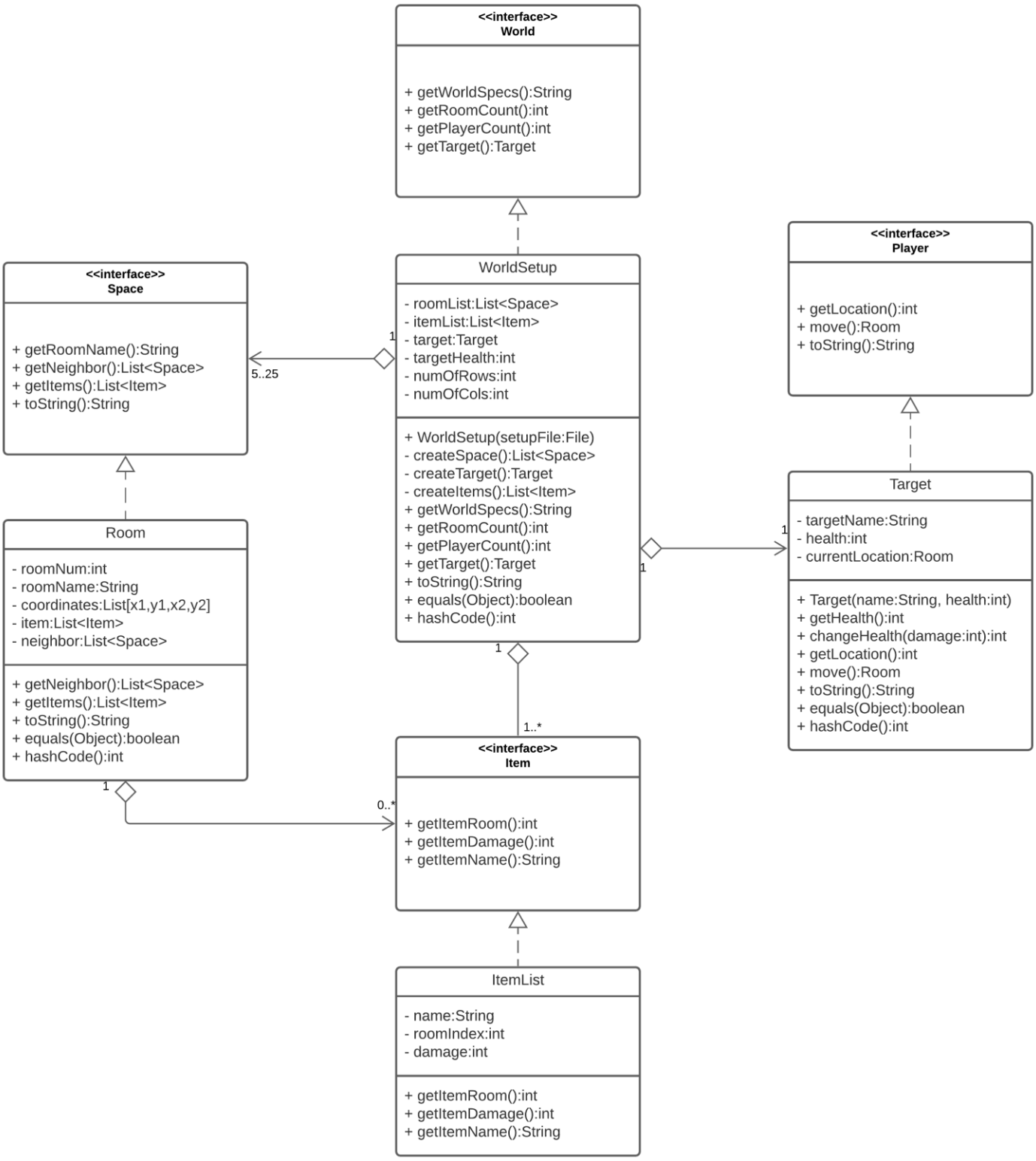
Rakshika Raju || raju.r@northeastern.edu

Kill Doctor Lucky UML



Kill Doctor Lucky Test Plan

Test the "World" Interface

Test WorldSetup() constructor	**Input**	**Expected output**
Input valid file path	mansion.txt	Constructor parses through the file
Input invalid file path	world.txt	Throws FileNotFoundException

Test getWorldSpecs()	**Input**	**Expected output**
Valid file was input	mansion.txt	File content returned as a string

Test getRoomCount()	**Input**	**Expected output**
Valid file was input with valid roomCount (roomCount>0)	mansion.txt – roomCount = 10	10
Valid file was input with invalid roomCount (roomCount<5)	mansion.txt – roomCount = 4	IllegalArgumentException
Valid file was input with invalid roomCount (roomCount>25)	mansion.txt – roomCount = 26	IllegalArgumentException
Valid file was input with negative roomCount (roomCount>25)	mansion.txt – roomCount = -2	IllegalArgumentException

Test getTarget()	**Input**	**Expected output**
Valid file was input with Target as String	mansion.txt – Target = Doctor Lucky	Doctor Lucky
Valid file was input with Target as non-string or no target	mansion.txt – Target = ""	IllegalArgumentException
Valid file was input with Target as non-string or no target	mansion.txt – Target = 123	IllegalArgumentException

Test the "Player" Interface and "Target" Class

Test Target(name:String, health:int) constructor	**Input**	**Expected output**
Valid Target String	mansion.txt – Target = Doctor Lucky	Target object created
Valid Health	mansion.txt – health = 50	Target object created
Invalid Target String	mansion.txt – Target = ""	IllegalArgumentException
Invalid Health	mansion.txt – health = -5 or 0	IllegalArgumentException

Test changeHealth(damage:int)	Input	Expected output
Valid Damage (damage>0)	Damage = 3	Returned value = currentHealth - 3
Valid Damage (damage<=0)	Damage = -1	IllegalArgumentException
Health going below 1	If currentHealth = 3 and Damage >=3	Doctor Lucky is killed

Test move()	Input	Expected output
currentRoom = 1	currentRoom = 1	Target moved to room 2
currentLocation = roomCount	currentLocation = roomCount	Target moved to room 1

Test "Space" Interface

Test getNeighbor()	Input	Expected output
Get neighbors of Room 1	Room1.getNeighbor()	[Room2, Room4, Room5]
Get neighbors of Room 8	Room8.getNeighbor()	[Room7, Room16]
Get neighbors of nonexistent room	Room50.getNeighbor()	NullPointerException

Test getItems()	Input	Expected output
Get items of Room with 1 item	Room11.getItems()	[Item17]
Get items of Room with 0 item	Room3.getItems()	[]
Get items of Room with >1 items	Room2.getItems()	[Item7, Item12]
Get Items of nonexistent room	Room50.getItems()	NullPointerException

Test "Item" Interface

Test getItemRoom()	Input	Expected output
Get Room where particular Item is located	Item2.getItemRoom()	4
Get Room for null Items	Item40.getItemRoom()	NullPointerException

Test getItemDamage()	Input	Expected output
Get damage for Item	Item2.getItemDamage()	2
Get damage for null Items	Item40.getItemRoom()	NullPointerException

Test getItemName()	Input	Expected output
Get name for of Item	Item2.getItemName()	Letter Opener
Get name for null Items	Item40.getItemRoom()	NullPointerException