





Player Interface	Testing Methods/constructors	Input	Expected output	Exception Message(if any)
	TestValidPlayerName()	player("Player1","HUMAN",1,3)	"Player1"	
	TestInvalidPlayerName()	player(" ", "HUMAN",1,3)	IllegalArgumentException	"Player name can not be blank"
	TestValidPlayerType()	player("Player1","HUMAN",1,3)	"HUMAN"	
	TestInvalidPlayerType()	player("Player1","xyz",1,3)	IllegalArgumentException	"Player can be either HUMAN or COMPUTER"
	TestValidPlayerOrder()	player("Player1","HUMAN",1,3)		1
	TestInvalidPlayerOrder()	player("Player1","HUMAN",-1,3)	IllegalArgumentException	"player order can not be negative"
	TestValidPlayerPosition()	player("Player1","HUMAN",1,3)		3
	TestInvalidPlayerPosition()	player("Player1","HUMAN",1,-3)	IllegalArgumentException	"player position can not be negative"
World Interface(Milestone 2 Test Cases)				
	TestValidAddPlayer()	addPlayer("Player1",3)	Player List containing:-{ "Player1",3}	
	TestInvalidAddPlayer()	addPlayer("Player2",40)	IllegalArgumentException	"Room num is invalid and does not exist"
	TestdisplayValidPlayer()	displayPlayer("Player1") for player("Player1","HUMAN",1,3)	A Player list containing:- playerName:Player1 playerType:"HUMAN" playerOrder:1 playerPosition:3 items:list of items as empty	
	TestdisplayInvalidPlayer()	displayPlayer("Ram")	IllegalArgumentException	"The player does not exist"
	TestValidlookAround()	lookAround(3) for Player("Player1","HUMAN",1,3) Space("Drawing room",3,"Player1",{ "Knife","stick"})	A space list containing:- roomName:- "Drawing room" roomNum:-3 playerName:-Player1 playerType:-"HUMAN" items:- { "knife","stick"}	
	TestInvalidlookAround()	lookAround(40)	IllegalArgumentException	"The room does not exist"
	TestpickItem()	pickItem("Knife") for Player("Player1","HUMAN",1,3) Space("Drawing room",3,"Player1",{ "Knife","stick"})	A Player list containing:- playerName:Player1 playerType:"HUMAN" playerOrder:1 playerPosition:3 items:list of items as { "Knife"}	
	TestInvalidPickItem()	pickItem("Scissors") for for Player("Player1","HUMAN",1,3) Space("Drawing room",3,"Player1",{ "Knife","stick"})	IllegalArgumentException	"Item not found"
	TestValidMovePlayer()	movePlayer(4)	currentPosition = 4	
	TestInvalidMovePlayer()	movePlayer(5)	IllegalArgumentException	"The room is not a neighbor"
		movePlayer(40)	IllegalArgumentException	"The room does not exist"
	TestTurn()	addPlayer("Player2",10)	"Player2"	
Space interface Milestone 2 test cases				
	TestGetPlayers()	Space("Drawing room",3,"Player1",{ "Knife","stick"})	"Player1"	
	TestInvalidGetPlayers()	Space("Drawing room",3,"",{ "Knife","stick"})	IllegalArgumentException	"Player Name can not be blank"
	TestRemoveItem()	RemoveItem("Knife") for Space("Drawing room",3,"",{ "Knife","stick"})	Item List:-{ "stick"}	
	TestInvalidRemoveItem()	RemoveItem("Ball") for Space("Drawing room",3,"",{ "Knife","stick"})	IllegalArgumentException	"item not found"
Traget interface Milestone 2 test cases				
	TestgetTargetName()	Target("Lucky",50,0,1)	"Lucky"	
	TestgetTargetHealth()	Target("Lucky",50,0,1)		50
	TestgetCurrentPos()	Target("Lucky",50,0,1)		0
	TestInvalidTargetName()	Target(" ",50,0,1)	IllegalArgumentException	"Target can not be blank"
	TestInvalidTargetHealth()	Target("Lucky",-9,0,1)	IllegalArgumentException	"Targethealth can not be negative"



Item Interface(MileStone1)	Testing Methods/constructors	Input	Expected output
	testGetItemRoom()	itemHelper("Crepe Pan", 8, 3, 21)	8
	testGetItemName()	itemHelper("Crepe Pan", 8, 3, 21)	"Crepe Pan"
	testGetItemDamage()	itemHelper("Crepe Pan", 8, 3, 21)	3
	testInvalidRoomNum()	itemHelper("Crepe Pan", -4, 3, 21); itemHelper("Crepe Pan", 22, 3,	IllegalArgumentException
	testInvalidItemDamage()	itemHelper(" ", 4, 3, 21); itemHelper("123456", 4, 3, 21);	IllegalArgumentException
	testToString()	itemHelper("Crepe Pan", 8, 3, 21)	"Item (ItemName = Crepe Pan, ItemDamage = 3, roomNum = 8)";
Space Interface(Milestone1)	testGetRoomName()	spaceHelper(16, 21, 21, 28, "Billiard Room", 1, itemList, 4)	"Billiard Room"
	testGetRoomDimension()	spaceHelper(16, 21, 21, 28, "Billiard Room", 1, itemList, 4)	{16,21,21,18}
	testGetRoomNum()	spaceHelper(16, 21, 21, 28, "Billiard Room", 1, itemList, 4)	1
	testGetnoOfItems()	spaceHelper(16, 21, 21, 28, "Billiard Room", 0, itemList, 4)	1
	testItems()	spaceHelper(22, 19, 23, 26, "Armory", 0, itemList, 4).getItems();	{"Crepe Pan" "Revolver"}
	testInvalidRoomName()	spaceHelper(16, 21, 21, 28, " ", 1, itemList, 4);	IllegalArgumentException
	testInvalidRoomNum()	spaceHelper(16, 21, 21, 28, "Billiard Room ", -4, itemList, 4); spaceHelper(16, 21, 21, 28,	IllegalArgumentException
	testInvalidRoomDimension()	spaceHelper(-7, 21, 21, 28, "Billiard Room ", 4, itemList, 4); spaceHelper(29, 21, 21, 28,	IllegalArgumentException
	testToString()	spaceHelper(16, 21, 21, 28, "Billiard Room", 1, itemList, 4)	"Rooms (roomNum = 1, roomName = Billiard Room, " + "roomDimension = 16, 21, 21, 28)"
World Interfcae(Milestone1)	testGetWorldRows()	fileName1	36
	testGetWorldColumns()	fileName1	30
	testGetTotalRooms()	fileName1	"Doctor Lucky's Mansion"
	testTargetHealth()	fileName1	50
	testMoveTargetChar	currentPosition =0	Next Position = 1
	testGetTotalRooms()	fileName1	21
	testGetRoomDetails()	roomIndex =1;	expectedRoomName = "Billiard Room", expecteditemName = "Billiard Cue";
	testGetNeighborDetails()	roomIndex = 1	{0,3,18}
	testInvalidWorldName()	fileName2	IllegalArgumentException
	testInvalidTotalRooms()	fileName2	IllegalArgumentException