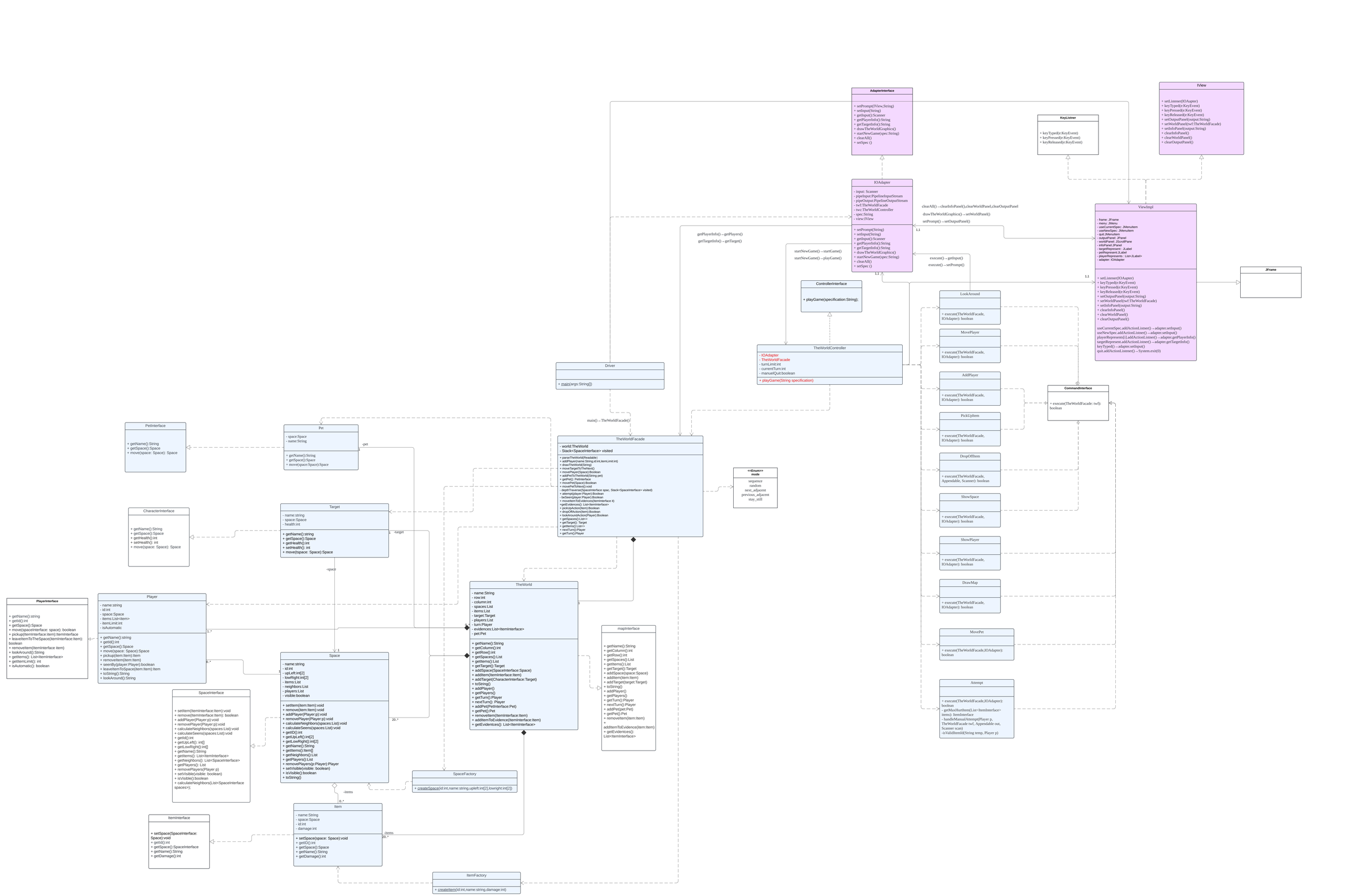
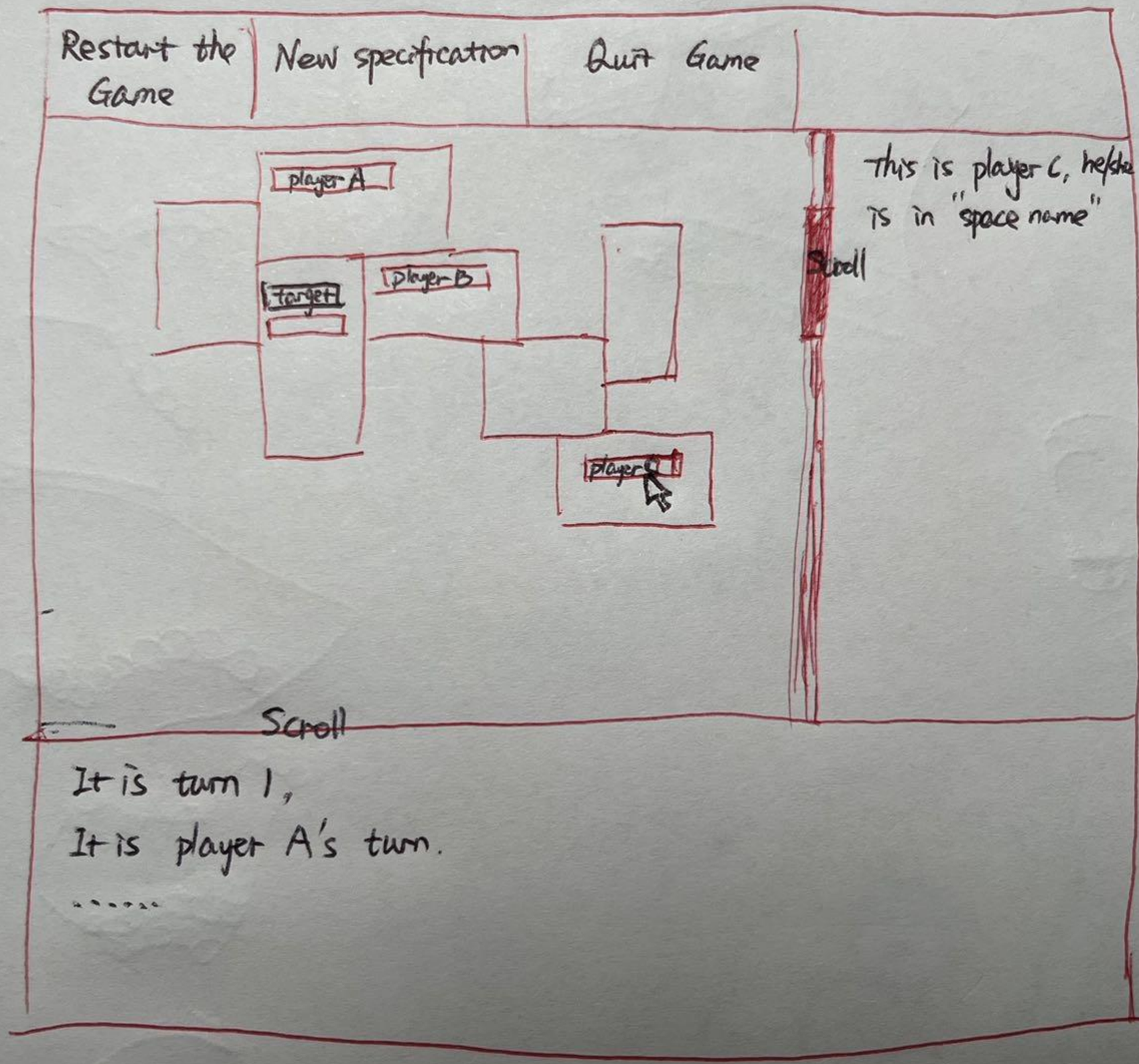

AdapterInterface
+ setPrompt(IView,String)
+ setInput(String)
+ getInput():Scanner
+ getPlayerInfo():String
+ getTargetInfo():String
+ drawTheWorldGraphics()
+ startNewGame(spec:String)
+ clearAll()
+ setSpec ()
IView
+ setListener(IOAapter)
+ keyTyped(e:KeyEvent)
+ keyPressed(e:KeyEvent)
+ keyReleased(e:KeyEvent)
+ setOutputPanel(output:String)
+ setWorldPanel(twf:TheWorldFacade)
+ setInfoPanel(output:String)
+ clearInfoPanel()
+ clearWorldPanel()
+ clearOutputPanel()
KeyListner
+ keyTyped(e:KeyEvent)
+ keyPressed(e:KeyEvent)
+ keyReleased(e:KeyEvent)
IOAdapter
- input: Scanner
- pipeInput:PipelineInputStream
- pipeOutput:PipelineOutputStream
- twf:TheWorldFacade
- twc:TheWorldController
- spec:String
- view:IView
+ setPrompt(String)
+ setInput(String)
+ getInput():Scanner
+ getPlayerInfo():String
+ getTargetInfo():String
+ drawTheWorldGraphics()
+ startNewGame(spec:String)
+ clearAll()
+ setSpec ()
clearAll() → clearInfoPanel(),clearWorldPanel,clearOutputPanel
drawTheWorldGraphics() → setWorldPanel()
setPrompt() → setOutputPanel()
1,1
ViewImpl
- frame: JFrame
- menu: JMenu
- useCurrentSpec: JMenuItem
- useNewSpec: JMenuItem
- quit:JMenuItem
- outputPanel: JPanel
- worldPanel: JScrollPane
- infoPanel:JPanel
- targetRepresent : JLabel
- petRepresent:JLabel
playerRepresents : List<JLabel>
- adapter: IOAdapter
+ setListener(IOAapter)
+ keyTyped(e:KeyEvent)
+ keyPressed(e:KeyEvent)
+ keyReleased(e:KeyEvent)
+ setOutputPanel(output:String)
+ setWorldPanel(twf:TheWorldFacade)
+ setInfoPanel(output:String)
+ clearInfoPanel()
+ clearWorldPanel()
+ clearOutputPanel()
useCurrentSpec.addActionListner() → adapter.setInput()
useNewSpec.addActionListner() → adapter.setInput()
playerRepresents[i].addActionListner() → adapter.getPlayerInfo()
targetRepresent.addActionListner() → adapter.getTargetInfo()
keyTyped() → adapter.setInput()
quit.addActionListner() → System.exit(0)
JFrame
getPlayerInfo() → getPlayers()
getTargetInfo() → getTarget()
startNewGame() → startGame()
startNewGame() → playGame()
execute() → getInput()
execute() → setPrompt()
1,1
1,1
ControllerInterface
+ playGame(specification:String);
TheWorldController
- IOAdapter
- TheWorldFacade
- turnLimit:int
- currentTurn:int
- manuelQuit:boolean
+ playGame(String specification)
LookAround
+ execute(TheWorldFacade, IOAdapter): boolean
MovePlayer
+ execute(TheWorldFacade, IOAdapter): boolean
AddPlayer
+ execute(TheWorldFacade, IOAdapter): boolean
PickUpItem
+ execute(TheWorldFacade, IOAdapter): boolean
DropOffItem
+ execute(TheWorldFacade, Appendable, Scanner): boolean
ShowSpace
+ execute(TheWorldFacade, IOAdapter): boolean
ShowPlayer
+ execute(TheWorldFacade, IOAdapter): boolean
DrawMap
+ execute(TheWorldFacade, IOAdapter): boolean
MovePet
+ execute(TheWorldFacade,IOAdapter): boolean
Attempt
+ execute(TheWorldFacade,IOAdapter): boolean
- getMaxHurtItem(List<ItemInterface> items): ItemInterface
- handleManualAttempt(Player p, TheWorldFacade twf, Appendable out, Scanner scan)
-isValidItemId(String temp, Player p)
CommandInterface
+ execute(TheWorldFacade: twf): boolean
Driver
+ main(args:String[])
main() → TheWorldFacade()
TheWorldFacade
- world:TheWorld
- Stack<SpaceInterface> visited
+ parseTheWorld(Readable)
+ addPlayer(name:String,id:int,itemLimit:int)
+ drawTheWorld(String)
+ moveTargetToTheNext()
+ movePlayer(Space):Boolean
+ addPetToTheWorld(String pet)
+ getPet(): PetInterface
+ movePet(Space):Boolean
+ movePetToNext():void
- depthTraverse(SpaceInterface spac, Stack<SpaceInterface> visited)
+ attempt(player:Player):Boolean
- beSeen(player:Player):Boolean
+ moveItemToEvidences(ItemInterface it)
+getEvidences(): List<ItemInterface>
+ pickUpAction(item):Boolean
+ dropOffAction(item):Boolean
+ lookAround(player:Player):Boolean
+ getSpaces():List<>
+ getTarget(): Target
+ getItems():List<>
+ nextTurn():Player
+ getTurn():Player
<<Enum>>
mode
sequence
random
next_adjacent
previous_adjacent
stay_still
PetInterface
+ getName():String
+ getSpace():Space
+ move(space: Space): Space
Pet
- space:Space
- name:String
+ getName():String
+ getSpace():Space
+ move(space:Space):Space
-pet
1
CharacterInterface
+ getName():String
+ getSpace():Space
+ getHealth():int
+ setHealth(): int
+ move(space: Space): Space
Target
- name:string
- space:Space
- health:int
+ getName():string
+ getSpace():Space
+ getHealth():int
+ setHealth(): int
+ move(space: Space):Space
1
-target
-space
PlayerInterface
+ getName():string
+ getId():int
+ getSpace():Space
+ move(space:SpaceInterface: space): boolean
+ pickup(ItemInterface:item):ItemInterface
+ leaveItemToTheSpace(itemInterface:Item): boolean
+ removeItem(ItemInterface item)
+ lookAround():String
+ getItems(): List<ItemInterface>
+ getItemLimit(): int
+ isAutomatic(): boolean
Player
- name:string
- id:int
- space:Space
- items:List<item>
- itemLimit:int
- isAutomatic
+ getName():string
+ getId():int
+ getSpace():Space
+ move(space: Space):Space
+ pickup(item:Item):Item
+ removeItem(item:Item)
+ seenBy(player:Player):boolean
+ leaveItemToSpace(item:Item):Item
+ toString():String
+ lookAround():String
1..*
0..*
TheWorld
- name:String
- row:int
- column:int
- spaces:List
- items:List
- target:Target
- players:List
- turn:Player
- evidences:List<ItemInterface>
- pet:Pet
+ getName():String
+ getColumn():int
+ getRow():int
+ getSpaces():List
+ getItems():List
+ getTarget():Target
+ addSpace(SpaceInterface:Space)
+ addItem(ItemInterface:Item)
+ addTarget(CharacterInterface:Target)
+ toString()
+ addPlayer()
+ getPlayers()
+ getTurn():Player
+ nextTurn(): Player
+ addPet(PetInterface:Pet)
+ getPet():Pet
+ removeItem(ItemInterface:Item)
+ addItemToEvidence(ItemInterface:Item)
+ getEvidences(): List<ItemInterface>
0..*
1
mapInterface
+ getName():String
+ getColumn():int
+ getRow():int
+ getSpaces():List
+ getItems():List
+ getTarget():Target
+ addSpace(space:Space)
+ addItem(item:Item)
+ addTarget(target:Target)
+ toString()
+ addPlayer()
+ getPlayers()
+ getTurn():Player
+ nextTurn():Player
+ addPet(pet:Pet)
+ getPet():Pet
+ removeItem(item:Item)
+
addItemToEvidence(item:Item)
+ getEvidences():
List<ItemInterface>
Space
- name:string
- id:int
- upLeft:int[2]
- lowRight:int[2]
- items:List
- neighbors:List
- players:List
- visible:boolean
+ setItem(item:Item):void
+ remove(item:Item):void
+ addPlayer(Player:p):void
+ removePlayer(Player:p):void
+ calculateNeighbors(spaces:List):void
+ calculateSeens(spaces:List):void
+ getID():int
+ getUpLeft():int[2]
+ getLowRight():int[2]
+ getName():String
+ getItems(): Item[]
+ getNeighbors():List
+ getPlayers():List
+ removePlayers(p:Player):Player
+ setVisible(visible: boolean)
+ isVisible():boolean
+ toString()
1
20..*
SpaceInterface
+ setItem(ItemInterface:Item):void
+ remove(ItemInterface:Item): boolean
+ addPlayer(Player:p):void
+ removePlayer(Player:p):void
+ calculateNeighbors(spaces:List):void
+ calculateSeens(spaces:List):void
+ getID():int
+ getUpLeft(): int[]
+ getLowRight():int[]
+ getName():String
+ getItems(): List<ItemInterface>
+ getNeighbors(): List<SpaceInterface>
+ getPlayers(): List
+ removePlayers(Player p)
+ setVisible(visible: boolean)
+ isVisible():boolean
+ calculateNeighbors(List<SpaceInterface spaces>);
SpaceFactory
+ createSpace(id:int,name:string,upleft:int[2],lowright:int[2])
-items
0..*
Item
- name:String
- space:Space
- id:int
- damage:int
+ setSpace(space: Space):void
+ getID():int
+ getSpace():Space
+ getName():String
+ getDamage():int
-items
20..*
ItemInterface
+ setSpace(SpaceInterface: Space):void
+ getId():int
+ getSpace():SpaceInterface
+ getName():String
+ getDamage():int
ItemFactory
+ createItem(id:int,name:string,damage:int)


Restart the Game
New specification
Quit Game
player A
target
player B
player C
this is player C, he/she is in "space name"
Scroll
Scroll
It is turn 1,
It is player A's turn.
.....

Test cases design

Updates-blue parts in each sector

New Class:
1. IOAdapter
Other updates:
1. TheWorldController - playGame(String specification)
2. LookAround - execute(TheWorldFacade, IOAdapter):boolean
3. Move - execute(TheWorldFacade, IOAdapter):boolean
4. PickUp - execute(TheWorldFacade, IOAdapter):boolean
5. Dropoff - execute(TheWorldFacade, IOAdapter):boolean
6. AddPlayer - execute(TheWorldFacade, IOAdapter):boolean
7. ShowPlace - execute(TheWorldFacade, IOAdapter):boolean
8. DrawMap - execute(TheWorldFacade, IOAdapter):boolean
9. MovePet - execute(TheWorldFacade, IOAdapter):boolean
10. Attempt - execute(TheWorldFacade, IOAdapter):boolean

This time ,we only need to test IOAdapter class , for other changes, we just simply enclose input and output of execute method into IOAdapter, that means we use IOAdapter to manager all the inputs and outputs between Controller and View.
If we need to test every execute method, we can simply mock an IOAdapter, and test the value of output that IOAdapter enclosed.

1. IOAdapter

This class implements AdapterInterface, it works for pass input and output between Controller and View (sometimes View can get infomation from model the TheWorld facade through IOAdapter)

IOAdapter also take charge in start the game. The playGame() method will not called by main function any more, instead of calling by the startNewGame() function in IOAdapter .

Test constructor	**Input**	**Expected Value**
Normal case	IOAdapter(new TheWorldFacade(), new TheWorldController(), spec:String, view:IView)	getInput() != null
Null in parameter	IOAdapter(new TheWorldFacade(), null, spec:String, view:IView)	Throws IllegalArgumentsException

Test setInput , getInput, setPrompt methods

Test setInput , getInput, setPrompt	**Input**	**Expected Value**

	Before:
	public class MockViewImpl implements IView{
	Private String outputPanel;
	Private String infoPanel;
	Private TheWorldFacade twf;
	Public void setOutputPanel(String s){
	this.outputPanel = s;
	}
	Public void setInfoPanel(String s){
	This.infoPanel = s;	

	}
	Public String getInfoPanel(){
	Return this.infoPanel;
	}
	Public String getoutputPanel(){
	Return this.outputPanel;
	}
	Public void DrawTheWorldGraphics(The WorldFacade twf){
	This.twf = twf;
	}
	Public TheWorldFacade getTWF(){
	Return this.twf;
	}
	}	

setInput()	Adapter=IOAdapter(new TheWorldFacade(), new TheWorldController(), "",view) setInput("123\n")	getInput().nextLine() == "123"
Null in parameter	setInput(null)	Throws IllegalArgumentsException
getInput()	setInput("123\n")	getInput().nextLine() == "123"
setPrompt()	view = new MockViewImpl(); Adapter=IOAdapter(new TheWorldFacade(), new TheWorldController(), "",view) adapter.setPrompt("this is a test")	view.getoutputPanel() == "this is a test"
Null in parameter	adapter.setPrompt(null)	Throws IllegalArgumentsException

Then test 3 functional method drawTheWorldGraphics() that using in draw the map on the view.The getTargetInfo() can provide target infomations to the view. The getPlayerInfo(int) provide player's infomation to the view ,this method takes an integer paramater which representates player Id.

Test drawTheWorldGraphics(), getTargetInfo() , getPlayerInfo(int)	**Input**	**Expected Value**

	Before：
	public class MockViewImpl implements IView{
	Private String outputPanel;
	Private String infoPanel;
	Private TheWorldFacade twf;
	Public void setOutputPanel(String s){
	this.outputPanel = s;
	}
	Public void setInfoPanel(String s){
	This.infoPanel = s;
	}
	Public String getInfoPanel(){
	Return this.infoPanel;
	}
	Public String getoutputPanel(){
	Return this.outputPanel;
	}
	Public void	

	DrawTheWorldGraphics(The
	WorldFacade twf){
	This.twf = twf;
	}
	Public TheWorldFacade
	getTWF(){
	Return this.twf;
	}
	}	

drawTheWorldGraphics()	twf = new TheWorldFacade() twf.createTheWorld("test world" , 10, 10) Adapter=IOAdapter(twf, new TheWorldController(), "",view) drawTheWorldGraphics()	view.getTWF.getWorldName() == "test world"
getPlayerInfo	twf = new TheWorldFacade() twf.createTheWorld("test world" , 10, 10) twf.addPlayerToTheWorld(new Player (new Player(0,"messi",1,true))) adapter=IOAdapter(twf, new TheWorldController(), "",view) adapter.getPlayreInfo(0)	adapter.getPlayreInfo(0).contains("messi")
getPlayerInfo with wrong id	adapter.getPlayerInfo(10)	adapter.getPlayreInfo(0).contains("wrong input")
getTargetInfo	twf = new TheWorldFacade() twf.createTheWorld("test world" , 10, 10) twf.parserWorld(new StringReader("40 40 The Dracula's haunted Castle\n 200 The Earl Decuras\n 2\n 11 9 10 8 Throne Room\n 8 18 24 23 Grand Ballroom\n 20\n 0 3 Bloodthirst Blade\n 1 2 Vampire's Fang Dagger\n "); adapter=IOAdapter(twf, new TheWorldController(), "",view) adapter.getTargetInfo	adapter.getTargetInfo.contains("The Earl Decuras")

Then ,we test the start and restart the game by startNewGame and setSpec , we use setInput method to pass new specification file path. And ClearAll() can help to clear the 3 view panels.

Test startNewGame(), setSpec(String) , clearAll()	**Input**	**Expected Value**

	Before:
	public class MockViewImpl implements IView{
	Private String outputPanel;
	Private String infoPanel;
	Private TheWorldFacade twf;
	Public void clearInfoPanel(){
	this.infoPanel = "this is test";
	}	

	Public void ClearOutputPanel(){
	This.outputPanel = “this is test”;
	}
	Public void clearWorldPanel(){
	this.infoPanel = “this is test”;
	}
	Public String getInfoPanel(){
	Return this.infoPanel;
	}
	Public String getoutputPanel(){
	Return this.outputPanel;
	}
	}	

Start a New Game	twc = new TheWorldControllerMock(adapter, 100); adapter=IOAdapter(twf, new TheWorldController(), ”specification1”,view) adapter.startNewGame()	view.getInfoPanel()== ”this is a test“
Restart a New Game with same specification	twc = new TheWorldControllerMock(adapter, 100); adapter=IOAdapter(twf, new TheWorldController(), ”specification1”,view) adapter.setInput(“q\n specification1”) adapter.startNewGame()	view.getInfoPanel()== ”this is a test for new turn sprification1”
Restart a New Game with different specification	twc = new TheWorldControllerMock(adapter, 100); adapter=IOAdapter(twf, new TheWorldController(), ”specification1”,view) adapter.setInput(“q\n specification2”) adapter.startNewGame()	view.getInfoPanel()== ”this is a test for new turn sprification2”
Restart a New Game with wrong specification file		Throws IllegalArgumentsException
clearAll()	twc = new TheWorldControllerMock(adapter, 100); adapter=IOAdapter(twf, new TheWorldController(), ”specification1”,view) adapter.setInput(“q\n specification2”) adapter.startNewGame() clearAll()	view.getInfoPanel()== ”this is a test for clearInfoPanel” view.getOutputPanel()== ” this is a test for clearOutputPanel”

2. Space

firstly create a new space by call constructor function Space(id:int , name:String, upLeft:int , downRight:int) by pass different arguments to test constructor and toString method.

Test constructor and toString()	**Input**	**Expected Value**
Normal case	Space(1,”bathroom”,new int[]{2,3},new int[]{5,8}).toString()	“id:1 name:bathroom leftcorner:2,3 rightcorner:5,8”

Space index 0	Space(0,"bathroom",new int[]{2,3},new int[]{5,8}).toString()	"id:0 name:bathroom leftcorner:2,3 rightcorner:5,8"
Left corner 0,0	Space(1,"bathroom",new int[]{0,0},new int[]{5,8}).toString()	"id:1 name:bathroom leftcorner:0,0 rightcorner:5,8"
Id < 0	Space(-1,"bathroom",new int[]{0,0},new int[]{5,8}).toString()	Throws IllegalArgumentsException
No name pass in	Space(1,,new int[]{0,0},new int[]{5,8}).toString()	Throws IllegalArgumentsException
Left corner <0	Space(1,"bathroom",new int[]{-1,0},new int[]{5,8}).toString()	Throws IllegalArgumentsException
Left corner is righter than right corner	Space(1,"bathroom",new int[]{-10,0},new int[]{5,8}).toString()	Throws IllegalArgumentsException
Left corner is higher than right corner	Space(1,"bathroom",new int[]{-1,10},new int[]{5,8}).toString()	Throws IllegalArgumentsException

Create a normal Space instance of **Space(1,"bathroom",new int[]{2,3},new int[]{5,8})** and test get,set method

Test setItem removeItem and get method	**Input**	**Expected Value**
getID()	Space(1,"bathroom",new int[]{2,3},new int[]{5,8},true)	1
getName()	above	"bathroom"
getUpLeft()	above	Int[]{2,3}
getDownRight()	above	Int[]{5,8}
Set an Item into space: setItem(Item) getItems()[0].toString()	setItem(New Item(0,"clamper",5))	"id:0 name:clamper,damage:5"
multi items case: getItems()[0].toString() getItems()[1].toString()	setItem(New Item(0,"clamper",5)) setItem(New Item(2,"mop",2))	"id:0 name:clamper,damage:5" "id:2 name:mop,damage:2"
Remove item case: removeItem(Item)	item1=setItem(New Item(0,"clamper",5)) item2=setItem(New Item(2,"mop",2)); removeItem(item1)	**getItems()[0].toString()**: "id:2 name:mop,damage:2"
Remove all cases	item1=setItem(New Item(0,"clamper",5)) item2=setItem(New Item(2,"mop",2)); removeItem(item1) removeItem(item2)	**getItems().size()**: 0
No item in the room , but try to Print items info	Space(1,"bathroom",new int[]{2,3},new int[]{5,8},true) Then directly: Print(getItems[0].toString())	Throws NullPointerException
isInvisible()	Space s=Space(1,"bathroom",new int[]{2,3},new int[]{5,8},true) s.setVisible(false) s.isInvisible()	false
isInvisible()	Space s=Space(1,"bathroom",new int[]{2,3},new int[]{5,8},true) s.setVisible(true) s.isVisible()	true

Create more than 1 spaces ,and make sure some of them are neighbor(share at least one "wall"), some of them are not . test calcNeighbors() which function set neighbors to a space, and getNeighbors();
Create more tha 1 spaces , make sure some of them has continous ID , some are not , if 2 space have continous ID, that means they can be seen by each other, otherwise , they can't. Test calcSeens() which set spaces can be seen by specified space, then getSeens().

Test calcNeighbors(), getNeighbors(),calcSeens() and getSeens()	**Input**	**Expected Value**
calcNeighbors(List spaces) getNeighbors()	space=new Space(1,"bathroom",new int[]{2,3},new int[]{5,8}); list={new Space(2,"kitchen",new int[]{5,3},new int[]{10,8}), New Space(3,"living room",new int[]{15,9},new int[]{22,11})}; space.calcNeighbors(list)	**space.getNeighbors(list).size() :** 1 **space..getNeighbors(list)[0].toString():** "id:2 name:kitchen leftcorner:5,3 rightcorner:10,8"
0 neighbor	space=new Space(1,"bathroom",new int[]{2,3},new int[]{5,8}); list={new Space(2,"kitchen",new int[]{11,3},new int[]{12,8}), New Space(3,"living room",new int[]{15,9},new int[]{22,11})};	**space.getNeighbors(list).size() :** 0
More than 1 neighbor	space=new Space(1,"bathroom",new int[]{2,3},new int[]{5,8}); list={new Space(2,"kitchen",new int[]{5,3},new int[]{10,8}), New Space(3,"living room",new int[]{3,8},new int[]{20,11})};	**space.getNeighbors(list).size() :** 2 **space..getNeighbors(list)[0].toString():** "id:2 name:kitchen leftcorner:5,3 rightcorner:10,8" **space..getNeighbors(list)[1].toString():** "id:3 name:living room leftcorner:3,8 rightcorner:20,11"

Then , test addPlayer(), getPlayers(),and removePlayer();

Add an player to the space	Space.addPlayer(new Player(0,"messi",1,true))	Space.getPlayers.size():1
Add 2 players to the space	Space.addPlayer(new Player(0,"messi",1,true)); Space.addPlayer(new Player(0,"rod",1,true));	Space.getPlayers.size():2
Add 2 players and the getPlayers	Space.addPlayer(new Player(0,"messi",1,true)); Space.addPlayer(new Player(0,"rod",1,true)); getPlayers();	Space.getPlayers.size():2
Add 2 players and remove 1	Player p = new Player(0,"messi",1,true) Player p2 = new Player(0,"rod",1,true) Space.addPlayer(p); Space.addPlayer(p2); getPlayers().size(); Space.removePlayer(p); getPlayers().size() Space.removePlayer(p2); getPlayers().size()	Space.getPlayers.size() from 2 change to 1 to 0

3.Item

Create an item by constructor, then test get , set method

Test constructor, get and set method of Item class	**Input**	**Expected Value**
Test constructor:normal case	Item(1,"mop",2)	**Item(1,"mop",2).toString():**

		“id:1 name:mop damage:2”
Id < 0	Item(-1,”mop”,2)	Throws IllegalArgumentsException
No name passed	Item(-1,,2)	Throws IllegalArgumentsException
Damage <0	Item(1,”mop”,-2)	Throws IllegalArgumentsException
Test get method: getID()	Item(1,”mop”,2)	1
getName()	above	“mop”
getDamage()	above	2
Set in which space the item placed: setSpace(Space space)	setSpace(new Space(1,”bathroom”,new int[]{2,3},new int[]{5,8})	**getSpace().getName():** “bathroom”
getSpace()		
No space been set	Item(1,”mop”,2) Then call getSpace()	Throws NullPointerException

4. Target

Create a target(implements character interface) By constructor, test constructor and get methods

Test constructor, get class	**Input**	**Expected Value**
Test constructor:normal case	Target(”lucky”,200)	**Target(“lucky”,200).toString():** “name:lucky health:200”
No name passed	Target(””,200)	Throws IllegalArgumentsException
Health <1	Target(”lucky”,0)	Throws IllegalArgumentsException
Health <0	Target(”lucky”,-1)	Throws IllegalArgumentsException
getName()	Target(”lucky”,200)	“lucky”
getHealth()	above	200

Then, we can test move() , the character can change the space he/she settled through move() by passing specified space.

Test move, getSpace	**Input**	**Expected Value**
move from space to no.1, by sequence mode, then getSpace	list = {new Space(0,”bathroom”,new int[]{2,3},new int[]{5,8}),new Space(1,”kitchen”,new int[]{5,3},new int[]{10,8}), New Space(2,”living room”,new int[]{3,8},new int[]{20,11})} space =move(list[1])	**target.getSpace().toString():** “id:1 name:kitchen leftcorner:5,3 rightcorner:10,8”

Move twice	For (i=0,i<2,i++){
	Space=move(list[i])
	}
		target.getSpace().toString(): “id:2 name:living room leftcorner:3,8 rightcorner:20,11”

Move thrice, out of the boundry of space list	For (i=0,i<3,i++){
	Space=move(list[i])
	}
		Throws IllegalArgumentsException

5. TheWorld

Create TheWorld by constructor that pass into a simple specification file , the file is like this(3 spaces):
It is a 5 spaces mansion with 7 items, it is saved as Mansion.txt, In Driver class ,we parser the file by line number, and construct TheWorld instance and Spaces/items/target that associated with TheWorld

Test constructor, get/set class	**Input**	**Expected Value**
Test constructor:normal case	world=TheWorld(”lucky’s mansion”,200,100)	**world.toString():** “name:lucky’s mansion rows:200 columns:100”
No name passed	TheWorld(””,200,100)	Throws IllegalArgumentsException
rows<1	TheWorld(”lucky’s mansion”,0,50)	Throws IllegalArgumentsException
rows<0	TheWorld(”lucky’s mansion”,-1,50)	Throws IllegalArgumentsException
columns<1	TheWorld(”lucky’s mansion”,100,0)	Throws IllegalArgumentsException
columns<0	TheWorld(”lucky’s mansion”,100,-1)	Throws IllegalArgumentsException
Test get method: getName()	world=TheWorld(”lucky’s mansion”,200,100)	“lucky’s mansion”
getRows()	above	200
getColumns()	above	100
addSpace(space)	addSpace(new Space(0,”bathroom”,new int[]{2,3},new int[]{5,8}))	**getSpaces()[0].toString():** “id:0 name:bathroom leftcorner:2,3 rightcorner:5,8”
getSpaces	getSpaces()[0]	above
add more than 1 space	addSpace(new Space(0,”bathroom”,new int[]{2,3},new int[]{5,8})); addSpace(new Space(1,”kitchen”,new int[]{8,15},new int[]{9,19}));	**getSpaces()[0].toString():** “id:0 name:bathroom leftcorner:2,3 rightcorner:5,8” **getSpaces()[1].toString():** “id:1 name:kitchen leftcorner:8,15 rightcorner:9,19”
addItem(space)	addItem(new Item(0,”mop”,10))	**getItems()[0].toString():** “id:0 name:mop damage:10”
getItems	above	above
Add more than 1 item	addItem(new Item(0,”mop”,10)) addItem(new Item(1,”helmet”,2))	**getItems()[0].toString():** “id:0 name:mop damage:10” **getItems()[1].toString():** “id:0 name:helmet damage:2”
addTarget(Target)	addTarget(new Target(“Lucky”,200))	**getTarget().toString():** “name:Lucky health:200”
getTarget	above	above

The target can move through the spaces in specified mode , right now there is only one moving mode called “sequence”, it stored in a enum named Mode

Test moveTarget()	**Input**	**Expected Value**
Test moveTarget(list,enum),stop at first step	list = {new Space(0,”bathroom”,new int[]{2,3},new int[]{5,8}),new Space(1,”kitchen”,new int[]{5,3},new int[]{10,8}), New Space(2,”living room”,new int[]{3,8},new int[]{20,11})} space =move(list,Mode.Sequence,1)	**space.toString():** “id:0 name:bathroom leftcorner:2,3 rightcorner:5,8”

Test moveTarget(list,enum),stop at second step	space =move(list,Mode.Sequence,2)	**space.toString():** "id:1 name:kitchen leftcorner:5,3 rightcorner:10,8"
Stop at 5 step, move back to kitchen	space =move(list,Mode.Sequence,5)	**space.toString():** "id:1 name:kitchen leftcorner:5,3 rightcorner:10,8"
Stop at 7 step, move back to kitchen	space =move(list,Mode.Sequence,7)	**space.toString():** "id:0 name:bathroom leftcorner:2,3 rightcorner:5,8"
Stop at 9 step, move back to kitchen	space =move(list,Mode.Sequence,9)	**space.toString():** "id:2 name:living room leftcorner:3,8 rightcorner:20,11"

The world class had been rewrite , add 2 attributes : players and turn ,should test addPlayer(), getPlayers(), getTurn(), nextTurn();

Test addPlayer() and getPlayers()	**Input**	**Expected Value**
Add 1 players to TheWorld	Player p = new Player(0,"messi",1,true) Space.addPlayer(p);	getPlayers().size() == 1; getPlayers().get(0).getName() == "messi"
Add 2 players to TheWorld	Player p2 = new Player(0,"rod",1,true) Space.addPlayer(p); Space.addPlayer(p2); getPlayers().size(); getPlayers().size()	getPlayers().size() == 2;

Test getTurn() and nextTurn(), getTurn() return the current player, and nextTurn() shift the next player in the queue to the current one and return.

Test getTurn() and nextTurn()	**Input**	**Expected Value**
Add 1 players to TheWorld and getTurn()	Player p = new Player(0,"messi",1,true) Space.addPlayer(p);	getTurn().getName() == "messi"
Add 1 players to TheWorld and nextTurn()		nextTurn().getName() == "messi"
Add 2 players to TheWorld and getTurn()	Player p = new Player(0,"messi",1,true) Space.addPlayer(p); Player p = new Player(0,"rodri",1,true) Space.addPlayer(p);	getTurn().getName() == "messi"
Add 2 players to TheWorld and nextTurn()		nextTurn().getName() == "rodri" getTurn().getName() == "rodri"
Add 2 players to TheWorld and keep nextTurn() for 2 times		nextTurn().getName() == "rodri" nextTurn().getName() == "messi" getTurn().getName() == "messi" It turn back to the head again.

6. Player

Player class is designed to provide all the basic function of a player , including getter and setters, and move(), pickup() for pickup an item , dropoff() for dropoff an item ,lookaround() for look around, the class is also implement from CharacterInterface;

Test Constructor	**Input**	**Expected Value**
Id < 0	New Player("messi",-1,10,false)	Throws IllegalArgumentException

itemLimit<=0	New Player(“messi”,1,0,false) New Player(“messi”,1,-1,false)	Throws IllegalArgumentException

Test getter methods

Test getName() getId() getSpace()	**Input**	**Expected Value**
Create a player and test getter	Player p = new Player(0,”messi”,1,true) getName() getId()	getName() == “messi” getId() == “0”
Move a player to a space and getSpace	Player p = new Player(0,”messi”,1,true) Move(New Space(1,”bathroom”,new int[]{2,3},new int[]{5,8})) getSpace(p);	getSpace().getName() == “bathroom”
Move a player from one place to the other	Move(New Space(1,”bathroom”,new int[]{2,3},new int[]{5,8})) Move(New Space(2,”kitchen”,new int[]{10,10},new int[]{20,20}))	getSpace().getName() == “kitchen”

Test other functions : pickup,dropoff, lookaround

Pickup(),dropoff(), lookaround(),toString()	**Input**	**Expected Value**
Pickup an item	Item=New Item(1,”mop”,2) pickup(item)	getItems().get(0).getName() == “mop”

Pickup but out of picking up limit	New Player(“messi”,1,1,false)//limit 1
	Item=New Item(1,”mop”,2)
	pickup(item)
	Item2=New Item(2,”towel”,2)
	pickup(item2)
		getItems().size() == 1
		getItems().get(0).getName() == “mop”

Drop off an item	Item=New Item(1,”mop”,2) pickup(item) dropoff(item)	getItems().size() == 0
Drop off an item but there is no item in the player’s item list	Item=New Item(1,”mop”,2) pickup(item) dropoff(item) dropoff(item)	dropoff(item) == false
Lookaround()	Move(New sp=Space(1,”bathroom”,new int[]{2,3},new int[]{5,8},true)) sp2 =Space(2,”kitchen”,new int[]{10,10},new int[]{20,20},true)) lookaround();	lookaround()=="this is the player messi, he/she is in the space No.1 bathroom\n he/she is watching the space:\n”+ sp2.toString();
toString()	Player p = new Player(0,”messi”,1,true) Item=New Item(1,”mop”,2) pickup(item) p.toString()	toString() =="this is the player messi, he/she is in the space No.1 bathroom\n he/she is carrying the item No.1 mop\n”
toString() but no item carried	Player p = new Player(0,”messi”,1,true)	toString() =="this is the player messi, he/she is in the space No.1 bathroom\n"
seenBy()	Player p = new Player(0,”messi”,1,false) Player p2 = new Player(1,”mac”,1,false) space =	true true

	Space(1,”bathroom”,new int[]{2,3},new int[]{5,8},true) p.move(space) p2.move(space) p.seenBy(p2) p2.seenBy(p)	
seenBy()	Player p = new Player(0,”messi”,1,false) Player p2 = new Player(1,”mac”,1,false) sp=Space(1,”bathroom”,new int[]{2,3},new int[]{5,8},true)) sp2 =Space(2,”kitchen”,new int[]{10,10},new int[]{20,20},true)) p.move(sp) p2.move(sp2) p.seenBy(p2) p2.seenBy(p)	false false

7. Pet

Pet class is designed to provide all the basic function of a pet,it has 2 attributes : name and space , space is where the pet is staying at right now.There are methods and all implement from animal Interface, getName() and getSpace(), and move() method can move pet to any space; Also includes
toString() and equals();

Test Constructor	**Input**	**Expected Value**
Id < 0	New Pet(“yok”)	toString() == “yok”
Name = null	New Pet(null)	Throws IllegalArgumentException

Test getter and move():

Test getName() getSpace(),move()	**Input**	**Expected Value**
Create a pet and test getter	Pet p = new Pet(“yak”) getName()	p.GetName() == “yak”
Move a pet to a space and getSpace	Pet p = new Pet(“yak”) Move(New Space(1,”bathroom”,new int[]{2,3},new int[]{5,8})) getSpace(p);	p.getSpace().getName() == “bathroom”

8. TheWorldFacade

When the World created ,we can test all setter and getter for the facade , since facade is a agent of the model, so most setter and getter is only a wrapper for the method in the base class
This time we add 3 public function
- movePet() : to move pet to specified space
- attempt() : to attack target
- wanderPet(): to move pet by deep-first traversal algorithm.

Despites of above, lookAroundFromSpace has changed by the logic of space visibility.

Test addPlayerToTheWorld() getItems() getSpaces()	**Input**	**Expected Value**

getTarget() getPlayers() getTurnOfTheGame() nextTurn()		
Add Player	Player p = new Player(0,"messi",1,false) addPlayerToTheworld(p)	getPlayers().get(0).getName() == "messi"
Add 2 players and getPlayer	Player p = new Player(0,"messi",1,false) addPlayerToTheworld(p) Player p2 = new Player(1,"rodri",1,false) addPlayerToTheworld(p2)	getPlayers().get(0).getName() == "messi" getPlayers().get(1).getName() == "rodri"
Add 2 players and 1 is a robot	Player p = new Player(0,"messi",1,false) addPlayerToTheworld(p) Player p2 = new Player(1,"rodri",1,true) addPlayerToTheworld(p2)	getPlayers().get(0).getName() == "messi" getPlayers().get(1).getName() == "rodri" getPlayers().get(1).isAutomatic() == true
Get target of the game		getTarget().getName() ==" The Earl Decuras"
getItems()		getItems().size() == 2 getItems().get(0).getName() ==" Bloodthirst Blade" getItems().get(1).getName() ==" Vampire's Fang Dagger"
getSpaces()		getSpaces().size() == 2 getSpaces().get(0).getName() ==" Throne Room" getSpaces().get(1).getName() ==" Grand Ballroom"
getTarget()		getTarget().getName() ==" The Earl Decuras"
getTurnOfTheGame()	Player p = new Player(0,"messi",1,false) addPlayerToTheworld(p) Player p2 = new Player(1,"rodri",1,true) addPlayerToTheworld(p2)	getTurnOfTheGame().getName() =="messi"
nextTurn()	above	getTurnOfTheGame().getName() =="messi" nextTurn.getName() =="rodri" getTurnOfTheGame().getName() =="messi"

Then, start to test major processing method :
moveTargetToTheNext() ,movePlayer(),pickUpAction(),dropOffAction(),lookAroundAction()

Test processing method	**Input**	**Expected Value**
moveTargetToTheNext()		getTarget().getSpace().getName() ==" Throne Room"
moveTargetToTheNext() 2 time	moveTargetToTheNext(); moveTargetToTheNext();	getTarget().getSpace().getName() ==" Grand Ballroom"
moveTargetToTheNext() 3 time,turn back to the first space	moveTargetToTheNext(); moveTargetToTheNext(); moveTargetToTheNext();	getTarget().getSpace().getName() ==" Throne Room"
movePlayer()	Player p = new Player(0,"messi",1,false) movePlayer(p,0) Player p2 = new Player(1,"rodri",1,true) movePlayer(p2,1)	p.getSpace().getName() == " Throne Room" p2.getSpace().getName() ==" Grand Ballroom"
Move to a invalid space number	movePlayer(p,0) movePlayer(p,2)//invalid	movePlayer(p,2) == false p.getSpace().getName() == " Throne Room"

pickUpAction()	pickUpAction(p,0)	p.getItems().get(0).getName() == “Bloodthirst Blade”
pickUpAction() over limit	pickUpAction(p,0) pickUpAction(p,1)//over limit	p.getItems().get(0).getName() == “Bloodthirst Blade” p.getItems().size() == 1
pickUpAction() with invalid number	pickUpAction(p,2)//invalid item	pickUpAction(p,2) == false p.getItems().size() == 0
dropOffAction()	pickUpAction(p,0) dropOffAction(p,p.getItems().get(0))	p.getItems().size() == 0
lookAroundFromSpace()	Player p = new Player(0,”messi”,1,false) movePlayer(p,0) Player p2 = new Player(1,”rodri”,1,true) movePlayer(p2,1) lookAroundFromSpace(p); lookAroundFromSpace(p2)	lookAroundFromSpace(p) == "this is the player messi, he/she is in the space No.0 Throne Room\n now he/she is watching Space No.1 Grand Ballroom; upleft:6,9; downright:12,17;\n includes 1 items\n - Bloodthirst Blade, cause 3 damage\n it has 1 neighbors\n - Space No.0 Throne Room is a neighbor\n" lookAroundFromSpace(p2)== "this is the player rodri, he/she is in the space No.1 Grand Ballroom\n now he/she is watching the space No.0 Throne Room; upleft:2,3; downright:5,8;\n includes 1 items\n -Vampire's Fang Dagger, cause 4 damage\n it has 1 neighbors\n - Space No.1 Grand Ballroom is a neighbor\n"
lookAroundFromSpace() if the pet is in the space()	Player p = new Player(0,”messi”,1,false) movePlayer(p,0) movePet(0) lookAroundFromSpace(p); lookAroundFromSpace(p2)	String s = lookAroundFromSpace(P) s.contains(“pet in the room”);
lookAroundFromSpace() if the pet is in the neighbor space	Player p = new Player(0,”messi”,1,false) Player p1 = new Player(1,”mac”,1,false) movePlayer(p,0) movePlayer(p1,0) movePet(1) lookAroundFromSpace(p)	this.getSpaces(1).isInVisible() == 1 this.getSpaces(0).getNeighbors().size() == 0
movePet	Player p = new Player(0,”messi”,1,false) movePlayer(p,0) movePet(1) lookAroundFromSpace(p)	this.getSpaces(1).isInVisible() == 1
attempt	Player p = new Player(0,”messi”,1,false) movePlayer(p,0) moveTargetToNext() p.attempt()	health = twf.getTarget().getHealth() out.contains(“poking him in the eye”) getTarget().getHealth() == health - 1
attempt and health reduce to 0	Player p = new Player(0,”messi”,1,false) p.addItem(new Item(2,”aaa”,100) movePlayer(p,0) moveTargetToNext() p.attempt()	out.contains(“target attacked by aaa”) getTarget().getHealth() == 0
wanderPet()	movePet(0) wanderPet()	getPet().getSpace().getId() == 1
Other action calls wanderPet()	Player p = new Player(0,”messi”,1,false)	getPet().getSpace().getId() == 1

	p.addItem(new Item(2,"aaa",100) movePlayer(p,0)	

9. MovePet

MovePet is a class that execute "move pet" command from input, it implements commandInterface,
It only includes execute method and ,the method take in TheWorldFacade instance.

Test execute()	Input	Expected Value
	@before twf = new TheWorldFacade() twf.parseTheWorld(new FileReader("A simple file include pet")) Player p = new Player(0,"messi",1,false) addPlayerToTheworld(p)	
Create a MovePet and move to a space	scan = new Scanner("1\n") out = new StringBuilder(); MovePet mp = new MovePet() mp.execute(twf,scan,out)	twf.getPet().getSpace().getName() == "bathroom"
Create a MovePet and try to move to a invalid space	scan = new Scanner("30\n") out = new StringBuilder(); MovePet mp = new MovePet() mp.execute(twf,scan,out)	out.contains("pet couldn't move to the Space, the space is not valid")

10. Attempt

Attempt is a class that execute "attempt" command from input, it implements commandInterface,
It only includes execute method and ,the method take in TheWorldFacade instance.

Test execute()	Input	Expected Value
	@before twf = new TheWorldFacade() twf.parseTheWorld(new FileReader("A simple file include pet")) Player p = new Player(0,"messi",1,false) addPlayerToTheworld(p)	

Create a Attempt and execute() without any items that carried by player	scan = new Scanner("")
	out = new StringBuilder();
	p.move(twf.getSpaces().get(0));
	Target.move(twf.getSpaces().get(0));
	Attempt at= new Attempt()
	at.execute(twf,scan,out)
		health = twf.getTarget().getHealth()
		out.contains("poking him in the eye")
		twf.getTarget().getHealth() == health - 1

Create a Attempt and execute() and there is one item carried by player	Item it = new Item(0,"mop",2)
	p.move(twf.getSpaces().get(0));
	Target.move(twf.getSpaces().get(0));
	p.pickup(it);
	scan = new Scanner("")
	out = new StringBuilder();
	Attempt at= new Attempt()
	at.execute(twf,scan,out)
		health = twf.getTarget().getHealth()
		out.contains("attacked by mop")
		twf.getTarget().getHealth() == health - 2

Create a Attempt and execute() and	Item it = new Item(0,"mop",2) Item it2 = new Item(1,"broomstick",3)	health = twf.getTarget().getHealth() out.contains("attacked by broomstick")

there is more than one item carried by player	p.pickUp(it);
	p.pickUp(it2);
	p.move(twf.getSpaces().get(0));
	Target.move(twf.getSpaces().get(0));
	scan = new Scanner(“”)
	out = new StringBuilder();
	Attempt at= new Attempt()
	at.execute(twf,scan,out)
		twf.getTarget().getHealth() == health - 3

Execute attempt but seen by others	Player p2 = new Player(1,”mac”,1,false)
	addPlayerToTheworld(p2)
	p2.move(twf.getSpaces().get(0));
	p1.move(twf.getSpaces().get(0));
	target.move(twf.getSpaces().get(0));
	scan = new Scanner(“”)
	out = new StringBuilder();
	Attempt at= new Attempt()
	at.execute(twf,scan,out)
		health = twf.getTarget().getHealth()
		out.contains(“attacked fail , seen by player mac.”)
		twf.getTarget().getHealth() == health

Execute attempt but seen by others staying in neighbor space	Player p2 = new Player(1,”mac”,1,false)
	addPlayerToTheworld(p2)
	p2.move(twf.getSpaces().get(0));
	p1.move(twf.getSpaces().get(1));
	target.move(twf.getSpaces().get(0));
	scan = new Scanner(“”)
	out = new StringBuilder();
	Attempt at= new Attempt()
	at.execute(twf,scan,out)
		health = twf.getTarget().getHealth()
		out.contains(“attacked fail , seen by player mac.”)
		twf.getTarget().getHealth() == health

11. TheWorldController

The world controller use playeGame() method to control the model by call TheWorldFacade and pass infomation from the user input/output to the model and vice versa, playGame use 8 different task helper to deal with user command. To test TheWorldController , we should firstly make a simple model by pass an small specification and use a stringReader to receive input from user.

Test constructor	**Input**	**Expected Value**
TheWorldController(Readable in, Appendable out, int turnLimit)()	Readable == null	Throw new IllegalArgumentException
TheWorldController(Readable in, Appendable out, int turnLimit)()	turnLimit < 1	Throw new IllegalArgumentException

Then test playGame().

Test playGame	**Input**	**Expected Value**
	@Before: StringReader specification= new StringReader("40 40 The Dracula's haunted Castle\n 200 The Earl Decuras\n 2\n 11 9 10 8 Throne Room\n 8 18 24 23 Grand Ballroom\n 20\n 0 3 Bloodthirst Blade\n 1 2 Vampire's Fang Dagger\n "); twf = TheWorldFacade();	
1 human-controlled player and movePet to Space 1	StringReader sri = new StringReader("n mac n 0 9 1 q"); StringBuilder out =new	out.contains(“pet has been move to Grand Ballroom”)

	StringBuilder(); TheWorldController(sri,out,3). playGame(twf,specification);	
1 human-controlled player and movePet to Space not available	StringReader sri = new StringReader("n mac n 9 3 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3). playGame(twf,specification);	out.contains(“pet couldn’t move to the Space, the space is not valid”)
1 human-controlled player and attempt sucessifully	StringReader sri = new StringReader("n mac n 0 10 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3). playGame(twf,specification);	out.contains(“poking him in the eye”)
1 human-controlled player and pickup an item and attempt sucessifully	StringReader sri = new StringReader("n mac n 0 2 0 10 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3). playGame(twf,specification);	out.contains(“attacked by Bloodthirst Blade”) twf.world.getEvedences().size() == 1
Attempt fail for no target in space	StringReader sri = new StringReader("n mac n 1 10 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3). playGame(twf,specification);	out.contains(“can not attempt now ,target is not in the space.”) twf.world.getEvedences().size() == 0
2 human-controlled player and 1 attempt with no item and seen by others same space	StringReader sri = new StringReader("n mac y n messi n 1 1 10 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3). playGame(twf,specification);	health = twf.getTarget().getHealth() out.contains(“attacked fail , seen by player messi.”) twf.getTarget().getHealth() == health twf.world.getEvedences().size() == 0
2 human-controlled player and 1 attempt using an item and seen by others neighbor space	StringReader sri = new StringReader("n mac y n messi n 0 1 2 0 2 1 10 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3). playGame(twf,specification);	health = twf.getTarget().getHealth() out.contains(“attacked fail , seen by player messi.”) twf.getTarget().getHealth() == health twf.world.getEvedences().size() == 1
1 robot player and pickup an item and attempt sucessifully	StringReader sri = new StringReader("y mac n"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3). playGame(twf,specification);	health = twf.getTarget().getHealth() out.contains(“attacked by mac.”)
1 robot player attempt but seen by a human player	StringReader sri = new StringReader("n mac y y messi n 1 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3). playGame(twf,specification);	out.contains(“attacked fail , seen by player mac.”) twf.world.getEvedences().size() == 1
Test wanderPet() during other action	StringReader sri = new StringReader("n mac n 1 0 0 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3). playGame(twf,specification);	out.contains(“pet move to Grand Ballroom.”)
Test game over use 1 robot and a large turnLimit	StringReader sri = new StringReader("y mac n"); StringBuilder out =new	out.contains(“game over , target killed by mac”) twf.world.getEvedences().size() > 0

	StringBuilder(); TheWorldController(sri,out,1000). playGame(twf,specification);	
Test game over use 1 robot and terminate by reaching turnLimit	StringReader sri = new StringReader("y mac n"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3). playGame(twf,specification);	out.contains("You have played enough turns, game is over , target is alive")
Add 1 human-controlled player and move to space 0, and quit	StringReader sri = new StringReader("n mac n 0 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3). playGame(twf,specification);	**out.toString().equals**("Add a player controlled by computer? press Y to create a robot,any other key to create a human-controlled player\n please enter his/her name:(only contains alphabeta)\n new player mac has been add\n Press Y to add more player, press any key to continue the game.\n It is turn 0\n Now is mac turn:\n Enter a move for mac to 0. Throne Room 1. Grand Ballroom 2. Blood Fountain\n mac has already move to Throne Room\n It is turn 1\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n User quit game, ByeBye!\n ")
Add 1 robot player and move to space 1, and quit	StringReader sri = new StringReader("y messi n 1 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3). playGame(twf,specification);	**out.toString().equals**("Add a player controlled by computer? press Y to create a robot,any other key to create a human-controlled player\n please enter his/her name:(only contains alphabeta)\n new player messi has been add\n Press Y to add more player, press any key to continue the game.\n It is turn 0\n Now is messi turn:\n Enter a move for messi to 0. Throne Room 1. Grand Ballroom 2. Blood Fountain\n messi has already move to Grand Ballroom\n it is turn 1\n Now is messi turn:\n Choose an action for messi , only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n User quit game, ByeBye!\n")
Add 2 player and 1 is a robot , seperately move to space 0 and space 1 Expected :Stop at turn 2	StringReader sri = new StringReader("n mac y y messi n 1 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3). playGame(twf,specification);	**out.toString().equals**(Add a player controlled by computer? press Y to create a robot,any other key to create a human-controlled player\n please enter his/her name:(only contains alphabeta)\n new player mac has been add\n Press Y to add more player, press any key to continue the game. Add a player controlled by computer? press Y to create a robot,any other key to create a human-controlled player\n please enter his/her name:(only contains alphabeta)\n new player messi has been add\n It is turn 0\n Now is mac turn:\n Enter a move for mac to 0. Throne Room 1. Grand Ballroom 2. Blood Fountain\n mac has already move to Throne Room\n it is turn 1\n Now is messi turn:\n Enter a move for mac to 0. Throne Room 1. Grand Ballroom 2. Blood Fountain\n messi has already move to Grand Ballroom\n it is turn 2\n Now is mac turn:\n Choose an action for messi , only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n User quit game, ByeBye!\n")
Add 1 player and move to room 0, then pick up item and quit Expected : 1 item picked up by player Expected: target move	StringReader sri = new StringReader("n mac n 0 2 0 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3). playGame(twf,specification);	**out.toString().equals**(Add a player controlled by computer? press Y to create a robot,any other key to create a human-controlled player\n please enter his/her name:(only contains alphabeta)\n new player mac has been add\n Press Y to add more player, press any key to continue the game.\n It is turn 0\n

to space 1 Expected: stop at turn 2		Now is mac turn:\n Enter a move for mac to 0. Throne Room 1. Grand Ballroom 2. Blood Fountain\n mac has already move to Throne Room\n The Earl Decuras has already moved to No. 0 Throne Room\n It is turn 1\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n please pick an item showing below.\n 0. Bloodthirst Blade\n the item had been picked up by mac.\n The Earl Decuras has already moved to No. 1 Grand Ballroom\n It is turn 2\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n User quit game, ByeBye!\n")
Add 1 player and move to room 0, then pick up item 0 and try to pick up another(but there is no more in the space)and quit Expected : 1 item picked up by player(1 pickup fail for being out of limit) Expected : prompts for there is no more items	StringReader sri = new StringReader("n mac n 0 2 0 2 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3).playGame(twf,specification);	**out.toString().equals**(Add a player controlled by computer? press Y to create a robot,any other key to create a human-controlled player\n please enter his/her name:(only contains alphabeta)\n Mac\n new player mac has been add\n Press Y to add more player, press any key to continue the game.\n It is turn 0\n Now is mac turn:\n Enter a move for mac to 0. Throne Room 1. Grand Ballroom 2. Blood Fountain\n mac has already move to Throne Room\n The Earl Decuras has already moved to No. 0 Throne Room\n It is turn 1\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n please pick an item showing below.\n 0. Bloodthirst Blade\n the item had been picked up by mac.\n The Earl Decuras has already moved to No. 1 Grand Ballroom\n It is turn 2\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n there is no item list in the room. try something different to do.\n It is turn 2\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n User quit game, ByeBye!\n)
Add 1 player and move to room 0, then pick up item 0 and move to room 1 and pick up item 1 and quit Expected : 1 item picked up by player(1 pickup fail for being out of limit) Expected: target move to space 2 Expected: stop at turn 3	StringReader sri = new StringReader("n mac n 0 2 0 1 1 2 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3).playGame(twf,specification);	**out.toString().equals**(Add a player controlled by computer? press Y to create a robot,any other key to create a human-controlled player\n please enter his/her name:(only contains alphabeta)\n new player mac has been add\n Press Y to add more player, press any key to continue the game.\n It is turn 0\n Now is mac turn:\n Enter a move for mac to 0. Throne Room 1. Grand Ballroom 2. Blood Fountain \n mac has already move to Throne Room\n The Earl Decuras has already moved to No. 0 Throne Room\n It is turn 1\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n please pick an item showing below.\n 0. Bloodthirst Blade\n the item had been picked up by mac.\n The Earl Decuras has already moved to No. 1 Grand Ballroom\n It is turn 2\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another

		space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n Enter a move for mac to 1. Grand Ballroom\n mac has already move to Grand Ballroom\n The Earl Decuras has already moved to No. 2 Blood Fountain\n It is turn 3\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q\n mac has no room to carry more items, try to drop off an item first.\n It is turn 3\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n User quit game, ByeBye!\n)
Add 1 player and move to room 0, then pick up item 0 and dropoff item 0 and quit Expected : 1 item picked up by player Expected: 1 item drop off Expected: target move to space 2 Expected: stop at turn 3	StringReader sri = new StringReader("n mac n 0 2 0 3 0 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3).playGame(twf,specification);	**out.toString().equals**(Add a player controlled by computer? press Y to create a robot,any other key to create a human-controlled player\n please enter his/her name:(only contains alphabeta)\n new player mac has been add\n Press Y to add more player, press any key to continue the game.\n It is turn 0\n Now is mac turn:\n Enter a move for mac to 0. Throne Room 1. Grand Ballroom 2. Blood Fountain\n mac has already move to Throne Room\n The Earl Decuras has already moved to No. 0 Throne Room\n It is turn 1\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n please pick an item showing below.\n 0. Bloodthirst Blade \n the item had been picked up by mac.\n The Earl Decuras has already moved to No. 1 Grand Ballroom\n It is turn 2\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n please leave an item in the space, items are shown below.\n 0. Bloodthirst Blade\n the item had been drop off by mac.\n The Earl Decuras has already moved to No. 2 Blood Fountain\n **It is turn 3\n** Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n User quit game, ByeBye!\n)
Try to drop off item but there is no item carried by the player Expected: stop at turn 1 Drop off do not complete , so it still be turn 1. Target is still in space 0(for the reason that turn 1 was not completed)	StringReader sri = new StringReader("n mac n 0 3 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3).playGame(twf,specification);	**out.toString().equals(** Add a player controlled by computer? press Y to create a robot,any other key to create a human-controlled player\n please enter his/her name:(only contains alphabeta)\n new player mac has been add\n Press Y to add more player, press any key to continue the game.\n It is turn 0\n Now is mac turn:\n Enter a move for mac to 0. Throne Room 1. Grand Ballroom 2. Blood Fountain mac has already move to Throne Room\n **The Earl Decuras has already moved to No. 0 Throne Room\n** It is turn 1\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n there is no item carried by mac. try something different to do.\n **It is turn 1\n** Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n

		User quit game, ByeBye!\n)
Add 1 player and move to room 0, and check space1 description and q Expected : Expected: target move to space 0 Expected: stop at turn 1	StringReader sri = new StringReader("n mac n 0 8 1 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3).playGame(twf,specification);	**out.toString().equals(** Add a player controlled by computer? press Y to create a robot,any other key to create a human-controlled player\n please enter his/her name:(only contains alphabeta)\n new player mac has been add\n Press Y to add more player, press any key to continue the game.\n It is turn 0\n Now is mac turn:\n Enter a move for mac to 0. Throne Room 1. Grand Ballroom 2. Blood Fountain\n mac has already move to Throne Room\n The Earl Decuras has already moved to No. 0 Throne Room\n It is turn 1\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n please pick a space to show description\n 0. Throne Room 1. Grand Ballroom 2. Blood Fountain\n Space No.1 Grand Ballroom; upleft:5,8; downright:18,9;\n includes 1 items\n - Vampire's Fang Dagger, cause 2 damage\n it has 2 neighbors\n - Space No.0 Throne Room is a neighbor\n - Space No.2 Blood Fountain is a neighbor\n It is turn 1\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n User quit game, ByeBye!\n)
Add 1 player and move to room 0, and check player description and q Expected : Expected: target move to space 0 Expected: stop at turn 1	StringReader sri = new StringReader("n mac n 0 7 0 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3).playGame(twf,specification);	**out.toString().equals(** Add a player controlled by computer? press Y to create a robot,any other key to create a human-controlled player\n please enter his/her name:(only contains alphabeta)\n new player mac has been add\n Press Y to add more player, press any key to continue the game.\n It is turn 0\n Now is mac turn:\n Enter a move for mac to 0. Throne Room 1. Grand Ballroom 2. Blood Fountain\n mac has already move to Throne Room\n The Earl Decuras has already moved to No. 0 Throne Room\n It is turn 1\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n please pick a player to show description\n 0. mac \n **this is the player mac, he/she is in the space No.0 Throne Room\n** It is turn 1\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n User quit game, ByeBye!\n)
Add 1 player and move to room 0 and look around and quit Expected: target move to space 1 Expected: stop at turn 2	StringReader sri = new StringReader("n mac n 0 4 q"); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3).playGame(twf,specification);	**out.toString().equals(** Add a player controlled by computer? press Y to create a robot,any other key to create a human-controlled player\n please enter his/her name:(only contains alphabeta)\n new player mac has been add\n Press Y to add more player, press any key to continue the game.\n It is turn 0\n Now is mac turn:\n Enter a move for mac to 0. Throne Room 1. Grand Ballroom 2. Blood Fountain \n mac has already move to Throne Room\n The Earl Decuras has already moved to No. 0 Throne Room\n It is turn 1\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n **this is the player mac, he/she is in the space No.0 Throne Room\n** **now he/she is watching the space:\n** **Space No.1 Grand Ballroom; upleft:5,8; downright:18,9;\n** **includes 1 items\n**

		- Vampire's Fang Dagger, cause 2 damage\n **it has 2 neighbors\n** **- Space No.0 Throne Room is a neighbor\n** **- Space No.2 Blood Fountain is a neighbor\n** **The Earl Decuras has already moved to No. 1 Grand Ballroom\n** It is turn 2\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n User quit game, ByeBye!\n)
move 4 time and game stopped by excess the turn limits Expected: target move to space 0(round back) Expected: player move to space 1(0,1,2,1) Expected: stop at turn 3(run 4 times)	StringReader sri = new StringReader("n mac n 0 1 1 1 2 1 1 "); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3).playGame(twf,specification);	**out.toString().equals**(Add a player controlled by computer? press Y to create a robot,any other key to create a human-controlled player\n please enter his/her name:(only contains alphabeta)\n new player mac has been add\n Press Y to add more player, press any key to continue the game.\n It is turn 0\n Now is mac turn:\n Enter a move for mac to 0. Throne Room 1. Grand Ballroom 2. Blood Fountain\n mac has already move to Throne Room\n The Earl Decuras has already moved to No. 0 Throne Room\n It is turn 1\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n Enter a move for mac to 1. Grand Ballroom\n mac has already move to Grand Ballroom\n The Earl Decuras has already moved to No. 1 Grand Ballroom\n It is turn 2\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n Enter a move for mac to 0. Throne Room 2. Blood Fountain\n mac has already move to Blood Fountain\n The Earl Decuras has already moved to No. 2 Blood Fountain\n It is turn 3\n Now is mac turn:\n Choose an action for mac, only press 1-8: 1.move to another space 2.pickup an item in the space 3.dropoff an item to the space 4.look around 5.add a player 6.draw the map 7.show a player 8.show a space , quit game press q.\n Enter a move for mac to 1. Grand Ballroom\n **mac has already move to Grand Ballroom\n** **The Earl Decuras has already moved to No. 0 Throne Room\n** **You have played enough turns, game is over**, ByeBye!\n)
Add a robot ,and run to the end (stopped by excessing the turn limits) Expected: target move to space 0(round back) Expected: player move to space 1(1) Expected: stop at turn 3(run 4 times)	Use mocking TheWorldFacade, to make sure that robot can run every action orderly(first move to space 1,and pickup an item,and dropoff an item,and lookaround and the move to space 2,and go on...) StringReader sri = new StringReader("y mac n "); StringBuilder out =new StringBuilder(); TheWorldController(sri,out,3).playGame(twf,specification);	**out.toString().equals**(Add a player controlled by computer? press Y to create a robot,any other key to create a human-controlled player\n please enter his/her name:(only contains alphabeta)\n new player mac has been add\n Press Y to add more player, press any key to continue the game.\n It is turn 0\n Now is mac turn:\n **mac has already move to Grand Ballroom\n** The Earl Decuras has already moved to No. 0 Throne Room\n It is turn 1\n Now is mac turn:\n mac decide to pickup an item\n please pick an item showing below.\n 1. Vampire's Fang Dagger \n **the item had been picked up by mac.\n** The Earl Decuras has already moved to No. 1 Grand Ballroom\n It is turn 2\n Now is mac turn:\n mac decide to drop off an item to the space\n please leave an item in the space, items are shown below.\n 1. Vampire's Fang Dagger \n **the item had been left to the space Grand Ballroom by mac.\n** The Earl Decuras has already moved to No. 2 Blood Fountain\n It is turn 3\n Now is mac turn:\n mac decide to look up around\n **this is the player mac, he/she is in the space No.1 Grand**

		Ballroom\n **now he/she is watching the space:\n** **Space No.0 Throne Room; upleft:7,6; downright:13,7;\n** **includes 1 items\n** **- Bloodthirst Blade, cause 3 damage\n** **it has 1 neighbors\n** **- Space No.1 Grand Ballroom is a neighbor\n** **now he/she is watching the space:\n** **Space No.2 Blood Fountain; upleft:2,7; downright:4,10;\n** **includes 1 items\n** **- Shadow Scythe, cause 2 damage\n** **it has 1 neighbors\n** **- Space No.1 Grand Ballroom is a neighbor\n** **The Earl Decuras has already moved to No. 0 Throne Room\n** You have played enough turns, game is over, ByeBye!\n)