World

Options

Welcome page

New Game with Current Config

New Game with New Config

Quit

Welcome!

author: Yanting Zheng

Init Players

Random player info

Custom player info

Player number 2

Player 1 AI

Name Phi

Room index 0

Player 1 AI

Name Wil

Room index 1

Confirm

New config means this should also be reset

Init Max Turn

Carried item limit 5

Confirm

Init Max Item Carried by A Player

Carried item limit 5

Confirm

World

Options

Game

12

NN Room

15

MM Room

Current turn is 3

Current player is Phi

order is 0

Phi carries 1 items, is in 2-indexed room Slaughter Beach.

--1 items are in the room:

Index:5, Name:Civil War Cannon, RoomIdx:2, Attack:3.

--Target character is in the room:

Name:Bravo Mabel, RoomIdx:2, Health:8."

--Pet is in the room."

Fortune the Cat"

--1 players are in the room\n"

Please input the index of the item you want to pick, use 'Enter' to confirm.

Game ends

Game Result

Player Phi wins the game!

Confirm

Pick up item/Attempt attack

Move the pet

Select Item

Knife (atk: 4)

Confirm

Select room

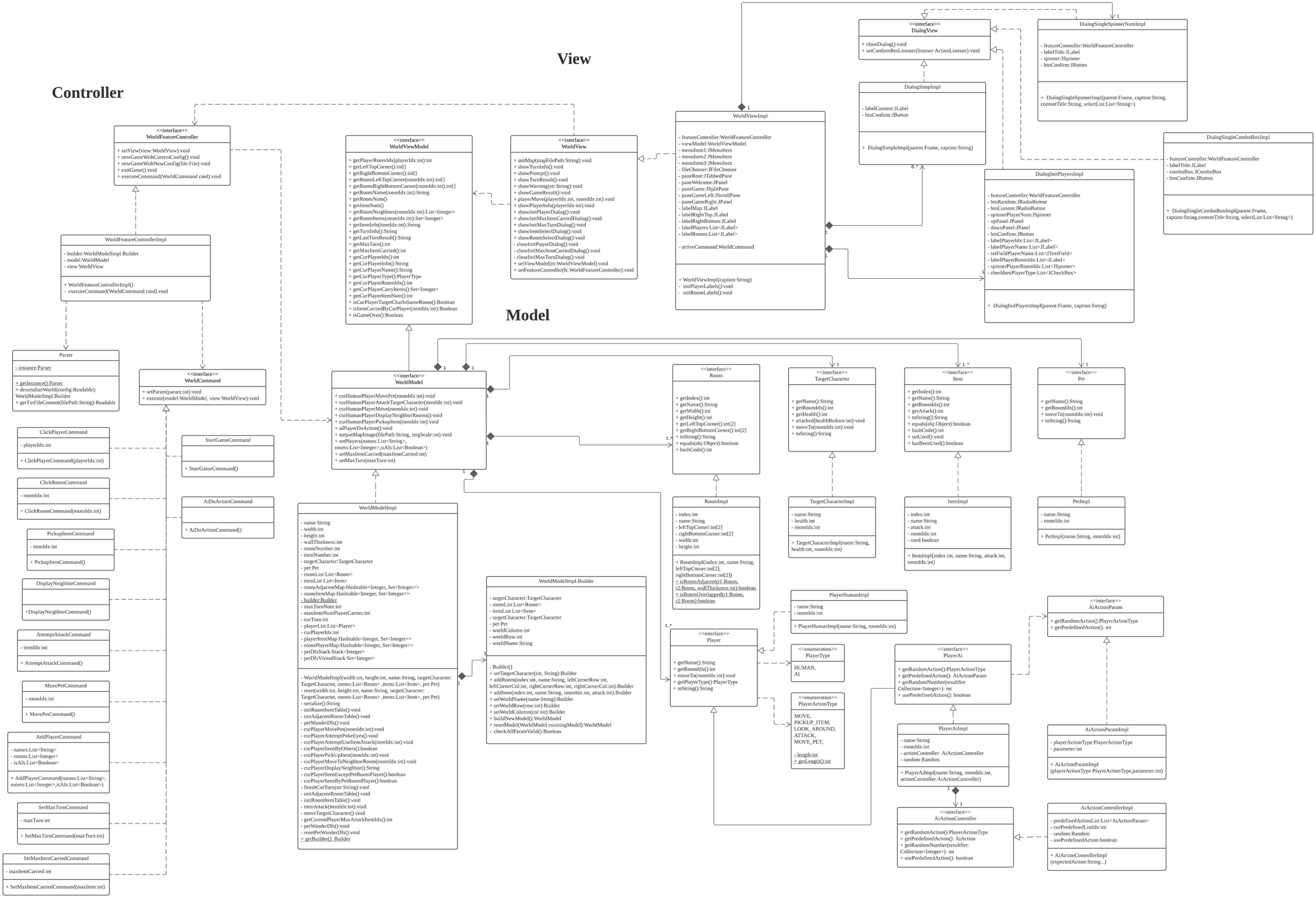
5

Confirm

View

Controller

Model



Changes made to the model

For this milestone, the model interface is split into two parts: one is for mutable functions, one is for immutable functions. The immutable interface is used by the view to get information for display. So compared to previous getter functions, I add more getter functions to meet the needs of the view display.
I also change the constructor of the model. Before I include player information, max turn and max item carried into the constructor of the model. Now the constructor only accepts information from the world specification. As before, this constructor is private, and there is an internal Builder class that helps build and verify the parameters the model needs.
Another change of the model is, I add the reset function to reinitialize the data of the model.This is also private, and only the builder could call it. All the new data will also be checked in the builder before being passed into the reset function.

Testing Plan

****For those related to milestone4, I have highlighted them with green****

for the WorldModel class

Testing Methods	Cases	Input	Expected Result
Constructor of World **This function in model is private, and could be called by the internal Builder class of the model.** **So this is testing the build function of the builder class.**	Wrong section format	Missing fields, extra characters, etc.	IllegalArgumentException
	Negative or zero size of world	"-6 30 World... " "6 -30 World... " "0 30 World... " "10 0 World... "	IllegalArgumentException
	Negative or zero row/col of rooms	"...-6 30 10 50 Room1... " "...6 -30 10 50 Room1... " "...6 30 -10 50 Room1... " "...6 30 10 -50 Room1... " "...0 30 10 50 Room1... " "...6 0 10 50 Room1... " "...6 30 0 50 Room1... " "...6 30 10 0 Room1... "	IllegalArgumentException
	Negative or zero damage of items	"...4 -1 Letter Opener... " "...4 0 Letter Opener... "	IllegalArgumentException
	Negative or zero health of target character	"...0 Doctor Lucky... " "...-20 Doctor Lucky... "	IllegalArgumentException
	Negative index of the room where items stay	"...-4 1 Letter Opener... "	IllegalArgumentException
	Space's number doesn't equate to the length of the list		IllegalArgumentException
	Item's number doesn't equate to the length of the list		IllegalArgumentException
	Room's invalid extent (left-top's row and column should be smaller than that of right-bottom)	"...16 30 10 40 Room1..." "...6 30 10 10 Room1..." "...16 30 16 40 Room1..." "...6 30 10 30 Room1..."	IllegalArgumentException

	Rooms' extents overlap	"...16 30 30 40 Room1 16 30 30 40 Room2..." "...16 30 30 40 Room1 13 35 18 50 Room2..." "...16 30 30 40 Room1 13 20 18 35 Room2..." "...16 30 30 40 Room1 18 20 20 35 Room2..." "...16 30 30 40 Room1 13 32 35 35 Room2..." "...16 30 30 40 Room1 18 32 20 35 Room2..."	IllegalArgumentException
	Room's extent exceeds World's extent	"20 30 World ...6 20 10 40 Room1..." "20 30 World ...0 20 30 25 Room1..."	IllegalArgumentException
	Doesn't add any room		IllegalArgumentException
	Doesn't add any item		IllegalArgumentException
reset **This function in model is private, and could be called by the internal Builder class of the model.** **So this is testing the reset function of the builder class.**	Wrong section format	Missing fields, extra characters, etc.	IllegalArgumentException
	Negative or zero size of world	"-6 30 World... " "6 -30 World... " "0 30 World... " "10 0 World... "	IllegalArgumentException
	Negative or zero row/col of rooms	"...-6 30 10 50 Room1... " "...6 -30 10 50 Room1... " "...6 30 -10 50 Room1... " "...6 30 10 -50 Room1... " "...0 30 10 50 Room1... " "...6 0 10 50 Room1... " "...6 30 0 50 Room1... " "...6 30 10 0 Room1... "	IllegalArgumentException
	Negative or zero damage of items	"...4 -1 Letter Opener... " "...4 0 Letter Opener... "	IllegalArgumentException
	Negative or zero health of target character	"...0 Doctor Lucky... " "...-20 Doctor Lucky... "	IllegalArgumentException
	Negative index of the room where items stay	"...-4 1 Letter Opener... "	IllegalArgumentException
	Space's number doesn't equate to the length of the list		IllegalArgumentException
	Item's number doesn't equate to the length of the list		IllegalArgumentException
	Room's invalid extent (left-top's row and column should be smaller than that of right-bottom)	"...16 30 10 40 Room1..." "...6 30 10 10 Room1..." "...16 30 16 40 Room1..." "...6 30 10 30 Room1..."	IllegalArgumentException
	Rooms' extents overlap	"...16 30 30 40 Room1 16 30 30 40 Room2..." "...16 30 30 40 Room1 13 35 18 50 Room2..." "...16 30 30 40 Room1 13 20 18 35 Room2..." "...16 30 30 40 Room1 18 20 20 35 Room2..." "...16 30 30 40 Room1 13 32 35 35 Room2..." "...16 30 30 40 Room1 18 32 20 35 Room2..."	IllegalArgumentException
	Room's extent exceeds World's extent	"20 30 World ...6 20 10 40 Room1..." "20 30 World ...0 20 30 25 Room1..."	IllegalArgumentException
	Doesn't add any room		IllegalArgumentException
	Doesn't add any item		IllegalArgumentException

	Also need to check whether the player information, max turn, max item carried are reset to default	After calling the reset, execute the functions below: getPlayerInfo(0) getMaxTurn() getMaxItemCarried()	getPlayerInfo should throw IllegalArgumentException, the other two should return 0
setMaxTurn(num:int)	Zero or negative num		IllegalArgumentException
	Valid num	15	getMaxTurn return 15
setMaxItemCarried(num:int)	Zero or negative num		IllegalArgumentException
	Valid num	3	getMaxItemCarried return 15
setPlayers(names:List<String>, rooms:List<Integer>,isAIs:List<Boolean>)	Nullable parameters	name is null, or rooms is null, or isAIs is null	IllegalArgumentException
	Empry name	Any name within the names list is empty	IllegalArgumentException
	Room index is negative or exceeds limit	Any room index within the rooms list is invalid	IllegalArgumentException
	Length of the three doesn't match	Like the length of first list is 10, the second list is 5, the third list is 9	IllegalArgumentException
	Empty list	All three lists are empty	IllegalArgumentException
	Valid	getPlayerInfo(x)	Get the right information of x-indexed player
Suppose we have "6 30 My World Version 1..."			
getName()			"My World Version 1"
getWidth()			6
getHeight()			30
getLeftTopCorner()			{0,0}
getRightBottomCorner()			{5,29}
printItemInfo(itemIdx:int)	Not picked by any player	"...4 10 Letter Opener... " Suppose 'Letter Opener' are in index 0. printItemInfo(0)	"Item index:0. Name:Letter Opener. Attack:10. RoomIdx:4"
	Negative index	printItemInfo(-2)	IllegalArgumentException
	Index out of range	Suppose we have 20 items. printItemInfo(20)	IllegalArgumentException
	Picked up by a player	"...4 10 Letter Opener... " Suppose 'Letter Opener' are in index 0. It is picked by a player, and the player is in 3-indexed room	"Item index:0. Name:Letter Opener. Attack:10. RoomIdx:3"
getItemNameAttack(itemIdx:int)	Negative index	printItemInfo(-2)	IllegalArgumentException
	Index out of range	Suppose we have 20 items. printItemInfo(20)	IllegalArgumentException
	Contains zero item	"...6 0 10 10 Room0 " Suppose 'Room0' is in index 0. printRoomInfo(0)	"Room index:0. Name:Room0. Left-top:{6,0}. Right-bottom:{10,10}. Width:4. Height:10. "
	Contains one item	"...6 0 10 10 Room0 ...0 10 Letter Opener... " Suppose 'Room0' and 'Letter Opener' are in index 0. printRoomInfo(0)	"Room index:0. Name:Room0. Left-top:{6,0}. Right-bottom:{10,10}. Width:4. Height:10. --Items:1 {Index:0. Name:Letter Opener. RoomIdx:0. Attack:10.}"
	Contains more than one item	"...6 0 10 10 Room0 ...0 10 Item0... ...0 4 Item1... ...0 3 Item2... " Suppose 'Room0' is in index 0.Item0, Item1, Item2 are in index0,1,2 printRoomInfo(0)	"Room index:0. Name:Room0. Left-top:{6,0}. Right-bottom:{10,10}. Width:4. Height:10. --Items:3 {Index:0. Name:Item0. RoomIdx:0. Attack:10.}, {Index:1. Name:Item1. RoomIdx:0. Attack:4.}, {Index:2. Name:Item2. RoomIdx:0. Attack:3.}."

printRoomInfo(roomIdx:int)	Contains zero player	As above	"Room index:0. Name:Room0. Left-top:{6,0}. Right-bottom:{10,10}. Width:4. Height:10. --Items:3 {Index:0. Name:Item0. RoomIdx:0. Attack:10.}, {Index:1. Name:Item1. RoomIdx:0. Attack:4.}, {Index:2. Name:Item2. RoomIdx:0. Attack:3.}."
	Contains one player	As above	"Room index:0. Name:Room0. Left-top:{6,0}. Right-bottom:{10,10}. Width:4. Height:10. --Items:3 {Index:0. Name:Item0. RoomIdx:0. Attack:10.}, {Index:1. Name:Item1. RoomIdx:0. Attack:4.}, {Index:2. Name:Item2. RoomIdx:0. Attack:3.}. --Players:1. {Order:4. Name:Nancy. RoomIdx:0}"
	Contains more than one player	As above	"Room index:0. Name:Room0. Left-top:{6,0}. Right-bottom:{10,10}. Width:4. Height:10. --Items:3 {Index:0. Name:Item0. RoomIdx:0. Attack:10.}, {Index:1. Name:Item1. RoomIdx:0. Attack:4.}, {Index:2. Name:Item2. RoomIdx:0. Attack:3.}. --Players:2. {Order:4. Name:Nancy. RoomIdx:0}, {Order:2. Name:Jack. RoomIdx:0}. "
	Doesn't contain target player	As above	"Room index:0. Name:Room0. Left-top:{6,0}. Right-bottom:{10,10}. Width:4. Height:10. --Items:3 {Index:0. Name:Item0. RoomIdx:0. Attack:10.}, {Index:1. Name:Item1. RoomIdx:0. Attack:4.}, {Index:2. Name:Item2. RoomIdx:0. Attack:3.}. --Players:1. {Order:4. Name:Nancy. RoomIdx:0}"
	Contains target player	As above	"Room index:0. Name:Room0. Left-top:{6,0}. Right-bottom:{10,10}. Width:4. Height:10. --Items:3 {Index:0. Name:Item0. RoomIdx:0. Attack:10.}, {Index:1. Name:Item1. RoomIdx:0. Attack:4.}, {Index:2. Name:Item2. RoomIdx:0. Attack:3.}. --TargetPlayer: Name:Doctor Lucky. Health:20. RoomIdx:0 --Players:1. {Order:4. Name:Nancy. RoomIdx:0} --Adjacent rooms:2 {1,5}. "

The item which is originally in this room is picked by a player and the player moves out of this room		room info doesn't contain info of the item that is picked away
Has no neighbors		"Room index:0. Name:Room0. Left-top:{6,0}. Right-bottom:{10,10}. Width:4. Height:10. --Items:3 {Index:0. Name:Item0. RoomIdx:0. Attack:10.}, {Index:1. Name:Item1. RoomIdx:0. Attack:4.}, {Index:2. Name:Item2. RoomIdx:0. Attack:3.}. --TargetPlayer: Name:Doctor Lucky. Health:20. RoomIdx:0 --Players:1. {Order:4. Name:Nancy. RoomIdx:0}"
Has one neighbor		"Room index:0. Name:Room0. Left-top:{6,0}. Right-bottom:{10,10}. Width:4. Height:10. --Items:3 {Index:0. Name:Item0. RoomIdx:0. Attack:10.}, {Index:1. Name:Item1. RoomIdx:0. Attack:4.}, {Index:2. Name:Item2. RoomIdx:0. Attack:3.}. --TargetPlayer: Name:Doctor Lucky. Health:20. RoomIdx:0 --Players:1. {Order:4. Name:Nancy. RoomIdx:0} --Adjacent rooms:1 {1}"
Has more than one neighbor		"Room index:0. Name:Room0. Left-top:{6,0}. Right-bottom:{10,10}. Width:4. Height:10. --Items:3 {Index:0. Name:Item0. RoomIdx:0. Attack:10.}, {Index:1. Name:Item1. RoomIdx:0. Attack:4.}, {Index:2. Name:Item2. RoomIdx:0. Attack:3.}. --TargetPlayer: Name:Doctor Lucky. Health:20. RoomIdx:0 --Players:1. {Order:4. Name:Nancy. RoomIdx:0} --Adjacent rooms:2 {1,5}. "
One of the neighbor rooms is occupied by the pet	Suppose the neighbor room's index is 5	"Room index:0. Name:Room0. Left-top:{6,0}. Right-bottom:{10,10}. Width:4. Height:10. --Items:3 {Index:0. Name:Item0. RoomIdx:0. Attack:10.}, {Index:1. Name:Item1. RoomIdx:0. Attack:4.}, {Index:2. Name:Item2. RoomIdx:0. Attack:3.}. --TargetPlayer: Name:Doctor Lucky. Health:20. RoomIdx:0 --Players:1. {Order:4. Name:Nancy. RoomIdx:0} --Adjacent rooms:1 {1}. "

	The room is occupied by the pet		"Room index:0. Name:Room0. Left-top:{6,0}. Right-bottom:{10,10}. Width:4. Height:10. --TargetPlayer: Name:Doctor Lucky. Health:20. RoomIdx:0 --Pet: Name:Patpat. RoomIdx:0 --Players:1. {Order:4. Name:Nancy. RoomIdx:0} --Adjacent rooms:2 {1,5}. "
	Negative index	printRoomInfo(-2)	IllegalArgumentException
	Index out of range	Suppose we have 20 rooms. printRoomInfo(20)	IllegalArgumentException
getRoomLeftTopCorner(roomIdx: int)	Valid room index	2	{0,4}
	Negative index	-2	IllegalArgumentException
	Index out of range	Suppose we have 20 rooms. 20	IllegalArgumentException
getRoomRightBottomCorner(roomIdx:int)	Valid room index	2	{0,4}
	Negative index	-2	IllegalArgumentException
	Index out of range	Suppose we have 20 rooms. 20	IllegalArgumentException
printTargetCharacterInfo()		"...20 Doctor Lucky... " Suppose the target character has moved to the 11st room and hasn't been attacked. printTargetCharacterInfo()	"Name:Doctor Lucky. Health:20. RoomIdx:10"
getRoomNeighbors(roomIdx:int)	Negative index	getAdjacentRoomIdx(-2)	IllegalArgumentException
	Index out of range	Suppose we have 20 rooms. getAdjacentRoomIdx(20)	IllegalArgumentException
	Input a room with a small number of neighbors	Suppose the 4th room has neighbors with indexes: 1, 5 getAdjacentRoomIdx(3)	{0,4}
	Input a room with many neighbors	Suppose the 4th room has neighbors with indexes: 0, 1, 2, 4, 5, 6 getAdjacentRoomIdx(3)	{0,1,2,4,5,6}
	Input a room with no neighbors		null
outputMapImage(filePath:String)	Path exists	outputMapImage("../res/mapImage.png")	Get a valid image representing the rooms and characters in the world
	Path doesn't exist	outputMapImage("../res/mapImage.png")	Get a valid image representing the rooms and characters in the world
toString()			Get the world specification, which has the layout as the parameter for the constructor
getCurPlayerIdx()		Suppose current player's order should be 2	2
getMaxTurn()		Suppose the max turn should be 30	30
printPlayerInfo(playerIdx:int)	Negative or over the max player index	-4 20(Suppose max room index is 5)	IllegalArgumentException
	Carries zero item	Suppose we have a player "Nancy" currently in the 5-indexed room. Its order is 4.It carries zero item.	"Player 4 Name: Nancy. RoomIdx:5.Carries no items."
	Carries one item	Suppose we have a player "Nancy" currently in the 5-indexed room. Its order is 4. It carries one item: {Item index:0. Name:Letter Opener. Attack:10. RoomIdx:5}.	"Player 4 Name: Nancy. RoomIdx:5.Carries Items: {Item index:0. Name:Letter Opener. Attack:10. RoomIdx:5}"

	Carries more than one items	Suppose we have a player "Nancy" currently in the 5-indexed room. Its order is 4. It carries two item: {Item index:2. Name:Letter Opener. Attack:10. RoomIdx:5}. {Item index:1. Name:Letter Opener2. Attack:3. RoomIdx:5}.	"Player 4 Name: Nancy. RoomIdx:5.Carries Items: {Item index:2. Name:Letter Opener. Attack:10. RoomIdx:5. Item index:1. Name:Letter Opener2. Attack:3. RoomIdx:5}
getPlayerItems(playerIdx:int)	Negative or over the max player index	-4 20(Suppose max room index is 5)	IllegalArgumentException
	Carries zero item	Suppose we have a player "Nancy" currently in the 5-indexed room. Its order is 4.It carries zero item.	null
	Carries one item	Suppose we have a player "Nancy" currently in the 5-indexed room. Its order is 4. It carries one item: {Item index:0. Name:Letter Opener. Attack:10.	{0}
	Carries more than one items	Suppose we have a player "Nancy" currently in the 5-indexed room. Its order is 4. It carries two item: {Item index:2. Name:Letter Opener. Attack:10.	{2,1}
curPlayerMove(roomIdx:int)	Negative or over the max room index	-4 20(Suppose max room index is 19)	IllegalArgumentException
	roomIdx isn't neighbor of the room current player is in	Suppose current player's adjacent room indexes are 2,5. Input is 4	IllegalArgumentException
	roomIdx is neighbor of the room current player is in	Suppose current player's adjacent room indexes are 2,5. Input is 2 printPlayerInfo(currentPlayerIdx)	current player's room index becomes 2
	Reaches max turn limit		IllegalStateException
curHumanPlayerDisplayNeighborRooms()			Should print complete information of neighbor rooms
	Neighbor room has a pet		No details showed for that room. Other rooms' details can be shown normally
	There are other players in the same room		Detail about current room contains information of other players in the same room
	There are no players in the same room		Detail about current room contains information of only current player
	There are other players in the neighbor room, no pet		Detail about neighbor room contains information of other players in that room
	There are no players in the neighbor room		Detail about neighbor room contains no information of players
	Reaches max turn limit		IllegalStateException
curHumanPlayerPickupItem(itemIdx:int)	Negative or over the max item index		IllegalArgumentException
	Item isn't in the same room as the current player		IllegalArgumentException
	Current player's items carried has reached the limit		IllegalStateException
	Picks an item in the same room, and the carried item number hasn't reached the limit	print the player's info before and after it picks the item	player's info now contains info about the newly picked item
	Reaches max turn limit		IllegalStateException
getCurTurnInfo()	Should only show brief information of the room current player is in. Should also contains information of where the target player is.		Current turn is 12 Current player is Phi, order is 0 Phi carries 2 items, is in 15-indexed room Salty Room Target player is in 11-indexed room Mi Room
isGameOver()	Reaches max turn limit or the target character's health reaches zero, then return true, else return false.		
	Pre-defined action list is in use.		The AI player will behave as expected.

aiPlayerDoAction()	Random is in use. It is in the same room with the target player and there are indeed no other player that could see it. In other cases, the AI player will choose any other action.		The AI player will try to make a kill attempt on the target character. The attempt would succeed and the health of the target character will reduce x points, x is the max attack of the items the player carries, or 1 if the player carries no item. The item the player uses is removed from the game.
	Random is in use. It is in the same room with the target player and it thinks that no other player could see it. But actually a pet is in a neighbor room with a player in it.		The AI player will try to make a kill attempt on the target character. The attempt would fail because it is seen by another player.
	Random is in use. Not in the same room with the target character. Or it finds itself visible to other players.		The AI player will choose other actions instead of attack.
curHumanPlayerMovePet(roomIdx:int)	Valid room index		Pet moves to the room with given index.
	room index is the same as pet's current one		IllegalArgumentException
	Negative or over the max room index	-4 20(Suppose max room index is 19)	IllegalArgumentException
	Game ends already		IllegalStateException
curHumanPlayerMovePet(roomIdx:int) **With pet moves in dfs order.**	Valid room index		Pet moves to a neighbor room of the room with given index.
curHumanPlayerAttackTargetCharacter(itemIdx:int)	The player isn't in the same room as the target character		IllegalStateException
	The player is in the same room with the target player. No other players see this player. This player carries one item.		The target character's health reduces x points, x is the attack of the item the player used. The item the player uses is removed from the game.
	The player is in the same room with the target player. No other players see this player. This player carries more than one item with different attacks.		The target character's health reduces x points, x is the attack of the item the player used. The item is the one with max attack among the items the player carries. The item the player uses is removed from the game.
	The player is in the same room with the target player. No other players see this player. This player carries no item.		The target character's health reduces 1 point,
	The player is in the same room with the target player. Another player is in the same room with this player.		The target character's health isn't reduced. This turn still finishes. Users are told the reason why the attack isn't successful. The item the player uses is removed from the game.
	The player is in the same room with the target player. Another player is in the neighbor room with this player.		The target character's health isn't reduced. This turn still finishes. Users are told the reason why the attack isn't successful. The item the player uses is removed from the game.
	given itemIdx isn't carried by the player	negative, exceeds max item index, or index of item that isn't carried by the current player	IllegalArgumentException
	Game ends already		IllegalStateException
Pet moves in a dfs order	Without being moved by any player, the pet moves in a dfs order from 0-indexed room	In every room put a player, in every turn, let the player execute the command of "look around"	The room information should contain information of the pet.
	Pet's dfs record is reset after being moved by a player	Let a player move the pet to a room, see if it will restart dfs order in the new place	However the pet moves before being moved, it will restart dfs order in the new place.

for the WorldFeatureController class			
Testing Methods	**Cases**	**Input**	**Expected Result**
WorldFeatureControllerImpl(WorldModel model)	Nullable parameter	null	IllegalArgumentException
	Valid instance	WorldModel m = new MockModel(stringBuilder, uniqCode);	
setView(v:WorldView)	Nullable parameter	null	IllegalArgumentException
	Valid instance	WorldView v = new MockView(stringBuilder, uniqCode);	Logs within MockView's setFeatureController function are printed Logs within MockView's setViewModel function are printed
newGameWithCurrentConfig()	View isn't set		Logs within MockModel's reset function should be printed
	View is set		Logs within MockModel's reset function are printed Logs within MockView's showInitMaxTurnDialog are printed
newGameWithNewConfig(file:File)	View isn't set	Valid specification file	Logs within MockModel's reset function should be printed
		Some png file MockModel's reset function throws exception	"Error: File isn't text file."
	Nullable parameter	null	Logs within MockView's showWarning are printed
	File is not text format	Some png file MockModel's reset function throws exception	Logs within MockModel's reset function should be printed Logs within MockView's showWarning are printed
	File is text file, but not conform the format of the world specification	MockModel's reset function throws exception	Logs within MockModel's reset function should be printed Logs within MockView's showWarning are printed
	File is valid world specification		Logs within MockModel's reset function are printed Logs within MockView's showInitMaxTurnDialog are printed
exitGame()	View isn't set	System Lambda.catchSystemExit	0
	View is set	System Lambda.catchSystemExit	0
executeCommand(cmd:WorldCommand) **All tests below assumes controller's view has been set**			
	First player is AI MockModel's getCurPlayerType() returns PlayerType.AI	MockModel's AiDoActionCommand() returns PlayerActionType.MOVE, 10	Logs within MockView's initMap are printed Logs within MockView's showTurnInfo are printed Logs within MockModel's getCurPlayerType function are printed Logs within MockModel's aiPlayerDoAction function are printed Logs within MockView's playerMove are printed, which include "10"
		MockModel's AiDoActionCommand() returns PlayerActionType.PICKUP_ITEM, 4	Logs within MockView's initMap are printed Logs within MockView's showTurnInfo are printed Logs within MockModel's getCurPlayerType function are printed Logs within MockModel's aiPlayerDoAction function are printed Logs within MockView's showTurnResult are printed
		MockModel's AiDoActionCommand() returns PlayerActionType.LOOK_AROUND	Logs within MockView's initMap are printed Logs within MockView's showTurnInfo are printed Logs within MockModel's getCurPlayerType function are printed Logs within MockModel's aiPlayerDoAction function are printed Logs within MockView's showTurnResult are printed

StartGameCommand		MockModel's AiDoActionCommand() returns PlayerActionType.ATTACK, -1	Logs within MockView's initMap are printed Logs within MockView's showTurnInfo are printed Logs within MockModel's getCurPlayerType function are printed Logs within MockModel's aiPlayerDoAction function are printed Logs within MockView's showTurnResult are printed
		MockModel's AiDoActionCommand() returns PlayerActionType.ATTACK, 4	Logs within MockView's initMap are printed Logs within MockView's showTurnInfo are printed Logs within MockModel's getCurPlayerType function are printed Logs within MockModel's aiPlayerDoAction function are printed Logs within MockView's showTurnResult are printed
		MockModel's AiDoActionCommand() returns PlayerActionType.MOVE_PET, 5	Logs within MockView's initMap are printed Logs within MockView's showTurnInfo are printed Logs within MockModel's getCurPlayerType function are printed Logs within MockModel's aiPlayerDoAction function are printed Logs within MockView's showTurnResult are printed
	First player is HUMAN MockModel's getCurPlayerType() returns PlayerType.HUMAN		Logs within MockView's initMap are printed Logs within MockView's showTurnInfo are printed Logs within MockModel's getCurPlayerType function are printed Logs within MockView's showPrompt are printed
ClickPlayerCommand	Clicked player isn't current turn's player	MockModel's getCurPlayerIdx() returns 0 Command's parameter is 1	Logs within MockModel's isGameOver function are printed Logs within MockModel's getCurPlayerIdx function are printed Logs within MockView's showWarning are printed
	Clicked player is current turn's player	MockModel's getCurPlayerIdx() returns 0 Command's parameter is 0	Logs within MockModel's isGameOver function are printed Logs within MockModel's getCurPlayerIdx function are printed Logs within MockModel's getCurPlayerInfo function are printed Logs within MockView's showPlayerInfo are printed
	Clicked room isn't neighbor of current turn's room	MockModel's getRoomNeighbors(4) returns {1,3,5,8} Command's parameter is 2	Logs within MockModel's isGameOver function are printed Logs within MockModel's getRoomNeighbors function are printed Logs within MockView's showRoomSelectDialog are printed Logs within MockView's showWarning are printed

ClickRoomCommand	Clicked room is neighbor of current turn's room	MockModel's getRoomNeighbors(4) returns {1,3,5,8} Command's parameter is 3	Logs within MockModel's isGameOver function are printed Logs within MockModel's getRoomNeighbors function are printed Logs within MockView's showRoomSelectDialog are printed Logs within MockModel's curHumanPlayerMove are printed Logs within MockView's playerMove are printed, which include "3" Logs within MockView's showTurnResult are printed
	Game over	MockModel's isGameOver() return true	MockModel's isGameOver function Logs within MockView's showWarning are printed
PickupItemCommand	Item isn't in the current turn's room	MockModel's getRoomItems(4) returns {1,5} Command's parameter is 2	Logs within MockModel's isGameOver function are printed Logs within MockModel's getCurPlayerItemNum function are printed Logs within MockModel's getMaxItemCarried function are printed Logs within MockModel's getRoomItems function are printed Logs within MockView's showItemSelectDialog are printed Logs within MockView's showWarning are printed
	Item is in the current turn's room	MockModel's getRoomItems(4) returns {1,5} Command's parameter is 1	Logs within MockModel's isGameOver function are printed Logs within MockModel's getCurPlayerItemNum function are printed Logs within MockModel's getMaxItemCarried function are printed Logs within MockModel's getRoomItems function are printed Logs within MockView's showItemSelectDialog are printed Logs within MockModel's getRoomItems function are printed Logs within MockView's showTurnResult are printed
	Items carried number reaches limit	MockModel's getCurPlayerItemNum() returns 3 MockModel's getMaxItemCarried() returns 3	Logs within MockModel's isGameOver function are printed Logs within MockModel's getCurPlayerItemNum function are printed Logs within MockModel's getMaxItemCarried function are printed Logs within MockView's showWarning are printed
	Game over	MockModel's isGameOver() return true	MockModel's isGameOver function Logs within MockView's showWarning are printed
DisplayNeighborCommand			Logs within MockModel's isGameOver function are printed Logs within MockModel's curHumanPlayerDisplayNeighborRooms function are printed Logs within MockView's showTurnResult are printed
	Game over	MockModel's isGameOver() return true	MockModel's isGameOver function Logs within MockView's showWarning are printed

AttemptAttackCommand	Item isn't carried by current player	MockModel's getCurPlayerCarryItems(0) returns {1,5} Command's parameter is 2	Logs within MockModel's isGameOver function are printed Logs within MockModel's isCurPlayerTargetCharInSameRoom function are printed Logs within MockModel's getCurPlayerCarryItems function are printed Logs within MockView's showItemSelectDialog are printed Logs within MockModel's getCurPlayerCarryItems function are printed Logs within MockView's showWarning are printed
	Item is carried by current player	MockModel's getCurPlayerCarryItems(0) returns {1,5} Command's parameter is 1	Logs within MockModel's isGameOver function are printed Logs within MockModel's isCurPlayerTargetCharInSameRoom function are printed Logs within MockModel's getCurPlayerCarryItems function are printed Logs within MockView's showItemSelectDialog are printed Logs within MockModel's getCurPlayerCarryItems function are printed Logs within MockView's showTurnResult are printed
	Current player has no item	MockModel's getCurPlayerCarryItems(0) returns null Command's parameter is -1	Logs within MockModel's isGameOver function are printed Logs within MockModel's isCurPlayerTargetCharInSameRoom function are printed Logs within MockModel's getCurPlayerCarryItems function are printed Logs within MockView's showTurnResult are printed
	Not in the same room with the target character	MockModel's isCurPlayerTargetCharInSameRoom() returns false	Logs within MockModel's isGameOver function are printed Logs within MockModel's isCurPlayerTargetCharInSameRoom function are printed Logs within MockView's showWarning function are printed
	Game over	MockModel's isGameOver() return true	MockModel's isGameOver function Logs within MockView's showWarning are printed
MovePetCommand	Room is not valid	MockModel's getRoomNum() returns 20 Command's parameter is 30	Logs within MockModel's isGameOver function are printed Logs within MockView's showRoomSelectDialog are printed are printed Logs within MockModel's getRoomNum function are printed Logs within MockView's showWarning are printed
		MockModel's getRoomNum() returns 20 Command's parameter is -3	Logs within MockModel's isGameOver function are printed Logs within MockView's showRoomSelectDialog are printed are printed Logs within MockModel's getRoomNum function are printed Logs within MockView's showWarning are printed

	Valid room index	Command's parameter is 1	Logs within MockModel's isGameOver function are printed Logs within MockView's showRoomSelectDialog are printed are printed Logs within MockModel's getRoomNum function are printed Logs within MockModel's curHumanPlayerMovePet function are printed Logs within MockView's showTurnResult are printed
	Game over	MockModel's isGameOver() return true	MockModel's isGameOver function Logs within MockView's showWarning are printed
AiDoActionCommand	MockModel's getCurPlayerType() returns PlayerType.AI	MockModel's AiDoActionCommand() returns PlayerActionType.MOVE, 10	Logs within MockModel's aiPlayerDoAction function are printed Logs within MockView's playerMove are printed, which include "10"
		MockModel's AiDoActionCommand() returns PlayerActionType.PICKUP_ITEM, 4	Logs within MockModel's aiPlayerDoAction function are printed Logs within MockView's showTurnResult are printed
		MockModel's AiDoActionCommand() returns PlayerActionType.LOOK_AROUND	Logs within MockModel's aiPlayerDoAction function are printed Logs within MockView's showTurnResult are printed
		MockModel's AiDoActionCommand() returns PlayerActionType.ATTACK, -1	Logs within MockModel's aiPlayerDoAction function are printed Logs within MockView's showTurnResult are printed
		MockModel's AiDoActionCommand() returns PlayerActionType.ATTACK, 4	Logs within MockModel's aiPlayerDoAction function are printed Logs within MockView's showTurnResult are printed
		MockModel's AiDoActionCommand() returns PlayerActionType.MOVE_PET, 5	Logs within MockModel's aiPlayerDoAction function are printed Logs within MockView's showTurnResult are printed
	MockModel's getCurPlayerType() returns PlayerType.HUMAN		IllegalStateException
	Game over	MockModel's isGameOver() return true	MockModel's isGameOver function Logs within MockView's showWarning are printed
AddPlayerCommand	Nullable parameter		Logs within MockView's showWarning are printed
	Any list is empty		Logs within MockView's showWarning are printed
	Three lists are of different length		Logs within MockView's showWarning are printed
	Any name within the name list is empty		Logs within MockView's showWarning are printed
	Any names within the name list are duplicate		Logs within MockView's showWarning are printed
	Any room within the room list is invalid	room index is negative or exceeds the max room index	Logs within MockModel's getRoomNum function are printed Logs within MockView's showWarning are printed
	Valid parameters		Logs within MockModel's getRoomNum function are printed Logs within MockModel's setPlayers function are printed Logs within MockView's closeInitPlayerDialog are printed
	Negative parameter	-2	Logs within MockView's showWarning are printed

SetMaxTurnCommand	Valid parameter	30	Logs within MockModel's setMaxTurn function are printed Logs within MockView's closeInitMaxTurnDialog are printed
SetMaxItemCarriedCommand	Negative parameter	-2	Logs within MockView's showWarning are printed
	Valid parameter	3	Logs within MockModel's setMaxItemCarried function are printed Logs within MockView's closeInitMaxItemCarriedDialog are printed