Updated Test Plan

Player Testing Plan

Testing	Input	Expected Value
Valid HumanPlayer construction	`HumanPlayer("Alice", 0, 5)`	Player object created successfully
Valid ComputerPlayer construction	`ComputerPlayer("CPU1", 0, 5, randomGenerator)`	Player object created successfully
Empty name	`HumanPlayer("", 0, 5)`	IllegalArgumentException
Null name	`HumanPlayer(null, 0, 5)`	IllegalArgumentException
Negative space index	`HumanPlayer("Bob", -1, 5)`	IllegalArgumentException
Negative max items	`HumanPlayer("Charlie", 0, -2)`	IllegalArgumentException
Get player name	`player.getPlayerName()` (for "Alice")	"Alice"
Get current space index	`player.getCurrentSpaceIndex()`	0
Set current space index	`player.setCurrentSpaceIndex(2)`	No exception
Get items (empty)	`player.getItems()`	Empty List
Add item	`player.addItem(new ItemImpl("Sword", 10, 0))`	true
Add item when inventory full	`player.addItem(new ItemImpl("Shield", 5, 0))`	false
Look around	`player.lookAround(spacesList)`	String describing current and neighboring spaces
Get description	`player.getDescription(spacesList)`	String with player name, current space, and items
Computer player take turn	`computerPlayer.takeTurn(spacesList)`	String describing action taken

GameFacade Testing Plan

Testing	Input	Expected Value
Get world name	`facade.getWorldName()`	Name of the world
Get space info	`facade.getSpaceInfo("Living Room")`	String with space information
Create world map	`facade.createWorldMap()`	BufferedImage object
Add human player	`facade.addHumanPlayer("Alice", "Living Room", 5)`	No exception
Add computer player	`facade.addComputerPlayer("CPU1", "Kitchen", 3)`	No exception
Move player	`facade.movePlayer("Kitchen")`	String confirming move
Player pick up item	`facade.playerPickUpItem("Knife")`	String confirming item pickup
Player look around	`facade.playerLookAround()`	String describing surroundings
Get player info	`facade.getPlayerInfo("Alice")`	String with player information
Get current player name	`facade.getCurrentPlayerName()`	Name of current player
Next turn	`facade.nextTurn()`	No exception

Testing	Input	Expected Value
Set max turns	`facade.setMaxTurns(50)`	No exception
Get current turn	`facade.getCurrentTurn()`	Current turn number
Move target character	`facade.moveTargetCharacter()`	No exception
Check computer player turn	`facade.computerPlayerTurn()`	Boolean indicating if it's a computer's turn
Computer player take turn	`facade.computerPlayerTakeTurn()`	String describing computer player's action
Get player count	`facade.getPlayerCount()`	Number of players in the game

WorldController Testing Plan

Testing	Input	Expected Value
Start game	`controller.startGame(50)`	Game starts, setup phase begins
Setup phase - add human player	Input: "add-human Alice "Living Room" 5"	Player added successfully
Setup phase - add computer player	Input: "add-computer CPU1 Kitchen 3"	Player added successfully
Setup phase - create map	Input: "map"	World map created successfully
Setup phase - start game	Input: "start"	Game setup complete, gameplay begins
Gameplay - move	Input: "move Kitchen"	Player moves to Kitchen
Gameplay - pick up item	Input: "pick Knife"	Player picks up Knife
Gameplay - look around	Input: "look"	Description of surroundings displayed
Gameplay - display space info	Input: "space Kitchen"	Information about Kitchen displayed
Gameplay - display player info	Input: "player-info Alice"	Information about Alice displayed
Gameplay - help	Input: "help"	Help information displayed
Gameplay - quit	Input: "quit"	Game ends

GameCommand Testing Plan

Testing	Input	Expected Value
MoveCommand execute	`new MoveCommand("Kitchen").execute(facade)`	String confirming move to Kitchen
PickUpItemCommand execute	`new PickUpItemCommand("Knife").execute(facade)`	String confirming Knife pickup
LookAroundCommand execute	`new LookAroundCommand().execute(facade)`	String describing surroundings
CreateWorldMapCommand execute	`new CreateWorldMapCommand().execute(facade)`	String confirming map creation
DisplaySpaceInfoCommand execute	`new DisplaySpaceInfoCommand("Kitchen").execute(facade)`	String with Kitchen information
AddHumanPlayerCommand execute	`new AddHumanPlayerCommand("Alice", "Living Room", 5).execute(facade)`	String confirming Alice added
AddComputerPlayerCommand execute	`new AddComputerPlayerCommand("CPU1", "Kitchen", 3).execute(facade)`	String confirming CPU1 added

Testing	Input	Expected Value
DisplayPlayerInfoCommand execute	`new DisplayPlayerInfoCommand("Alice").execute(facade)`	String with Alice's information
HelpCommand execute	`new HelpCommand(false).execute(facade)`	String with gameplay help information

World Testing Plan

Testing	Input	Expected Value
Get world name	`world.getWorldName()`	Name of the world
Get rows	`world.getRows()`	Number of rows in the world
Get columns	`world.getColumns()`	Number of columns in the world
Get total spaces	`world.getTotalSpace()`	Total number of spaces
Get total items	`world.getTotalItems()`	Total number of items
Find neighbors	`world.findNeighbors()`	No exception, neighbors set correctly
Create world map	`world.createWorldMap()`	BufferedImage of the world map
Get target character	`world.getTargetCharacter()`	Copy of the target character
Add player	`world.addPlayer(new HumanPlayer("Alice", 0, 5))`	No exception, player added
Get players	`world.getPlayers()`	List of all players
Get current player	`world.getCurrentPlayer()`	Current player object
Next turn	`world.nextTurn()`	No exception, turn advanced
Get player count	`world.getPlayerCount()`	Number of players in the game
Set max turns	`world.setMaxTurns(50)`	No exception
Get current turn	`world.getCurrentTurn()`	Current turn number
Get max turns	`world.getMaxTurns()`	Maximum number of turns
Get space by index	`world.getSpaceByIndex(0)`	Space object at index 0
Get all spaces	`world.getSpaces()`	List of all spaces in the world