

Controller Interaction Testing

Add Player

Testing Action	Input	Expected Value
Add Human Player	<code>controller.handleAddPlayer(true)</code>	Name prompt, space picker, and item capacity prompt shown; player added to facade and view list

Add Computer Player

Testing Action	Input	Expected Value
Add Computer Player	<code>controller.handleAddPlayer(false)</code>	Computer player added to facade and view list

Game Start Validation

Testing Action	Input	Expected Value
No Players Added	<code>controller.handleGameStart()</code>	Error shown: "Add at least one player"
1 Player Added	<code>controller.handleGameStart()</code>	Game screen shown, turn display updated

Space Click During Player Turn

Testing Action	Input	Expected Value
Player Space Click	<code>controller.handleSpaceClick()</code>	Space processed, world refreshed, status updated

Space Click During Computer Turn

Testing Action	Input	Expected Value
Computer Turn Space Click	<code>controller.handleSpaceClick()</code>	No space processing if it's computer's turn

Pet Movement System

Testing Action	Input	Expected Value
Enter Pet Move Mode	<code>controller.handleMovePet()</code>	Pet movement mode activated
Move Pet	<code>controller.handlePetMove()</code>	Pet moved, world refreshed

End of Game Testing

Testing Action	Input	Expected Value
Game End with Winner	<code>controller.handleGameEnd()</code>	Win dialog shown
Game End with Escape	<code>controller.handleGameEnd()</code>	Escape message shown with "Target escaped!"

Menu Operations

Testing Action	Input	Expected Value
New Game with New World	<code>controller.handleNewGame()</code>	File chooser shown
New Game with Current World	<code>controller.handleNewGameCurrentWorld()</code>	Setup screen shown, file chooser not shown

Player Limit Enforcement

Testing Action	Input	Expected Value
Add More Players Than Limit	<code>controller.handleAddPlayer(true)</code>	Error message about player limit shown

Invalid Actions Testing

Testing Action	Input	Expected Value
Invalid Space Click	<code>controller.handleClick()</code>	Invalid click should not process
Invalid Pet Movement	<code>controller.handlePetMove()</code>	Invalid movement should not be processed

Player Information Display

Testing Action	Input	Expected Value
Player Info on Click	<code>controller.handleClick()</code>	Player info shown upon clicking on player

Game State Transitions

Testing Action	Input	Expected Value
Transition to Gameplay	<code>controller.handleGameStart()</code>	Game screen shown, turn display initialized

Player Movement Testing

Testing Action	Input	Expected Value
Player Movement Click	<code>controller.handleClick()</code>	Space checked, player moves, world refreshed

ViewModel Testing Plan

Testing Construction	Input	Expected Value
Valid WorldImpl construction	<code>WorldImpl("Test World", 3, 3, spaces, target, 9, 5, pet)</code>	WorldImpl object created successfully
Valid addPlayer()	<code>world.addPlayer(new HumanPlayer("Alice", 0, 0))</code>	Player added successfully

Testing Construction	Input	Expected Value
Invalid addPlayer() with null player	world.addPlayer(null)	IllegalArgumentException
Empty world name	WorldImpl("", 3, 3, spaces, target, 9, 5, pet)	IllegalArgumentException
Invalid number of rows	WorldImpl("Test World", -1, 3, spaces, target, 9, 5, pet)	IllegalArgumentException
Invalid number of columns	WorldImpl("Test World", 3, -1, spaces, target, 9, 5, pet)	IllegalArgumentException

Testing getWorldName()

Testing Construction	Input	Expected Value
Get world name	viewModel.getWorldName()	"Test World"

Testing getRows() and getColumns()

Testing Construction	Input	Expected Value
Get number of rows	viewModel.getRows()	3
Get number of columns	viewModel.getColumns()	3

Testing getPlayerCopies()

Testing Construction	Input	Expected Value
Get player copies	viewModel.getPlayerCopies()	List of 2 Player objects
Ensure copied player is not same	viewModel.getPlayerCopies().get(0) != ((WorldImpl) viewModel).getPlayers().get(0)	True

Testing getSpaceCopies()

Testing Construction	Input	Expected Value
Get space copies	viewModel.getSpaceCopies()	List of 3 Space objects
Ensure copied space is not same	viewModel.getSpaceCopies().get(0) != ((WorldImpl) viewModel).getSpaces().get(0)	True

Testing getTargetCopy()

Testing Construction	Input	Expected Value
Get target copy	viewModel.getTargetCopy()	Non-null TargetCharacter object
Ensure copied target is not same	viewModel.getTargetCopy() != ((WorldImpl) viewModel).getTargetCharacter()	True

Testing getPetCopy()

Testing Construction	Input	Expected Value
Get pet copy	viewModel.getPetCopy()	Non-null Pet object
Ensure copied pet is not same	viewModel.getPetCopy() != ((WorldImpl) viewModel).getPet()	True

Testing createWorldMap()

Testing Construction	Input	Expected Value
Generate world map	viewModel.createWorldMap()	Non-null BufferedImage with positive dimensions

Testing getCurrentPlayerCopy()

Testing Construction	Input	Expected Value
Get current player copy	viewModel.getCurrentPlayerCopy()	Non-null Player object
Ensure copied player is not same	viewModel.getCurrentPlayerCopy() != ((WorldImpl) viewModel).getCurrentPlayerCopy()	True

Testing getCurrentTurn()

Testing Construction	Input	Expected Value
Get current turn	viewModel.getCurrentTurn()	1 (default turn)