

2) TEST PLAN

Class Name	Test Name	Test Condition	Input Examples	Expected Output
Controller	startGame()	Game starts successfully	Initialized game components	Successfully starts the game
Controller	processCommand(Command)	Valid move command execution	Player1 issues a move command to room B	Command executes successfully, Player1 in room B
Controller	processCommand(Command)	Invalid attack command	Player1 attacks a non-adjacent player	Returns "Target not in range"
Controller	displaySpaceInfo(Space)	Display info for populated space	Room A with 2 players and 1 item	Returns detailed room description
Controller	addHumanPlayer(String name)	Adding human-controlled player	Add Player1	Player1 added successfully
Controller	addComputerPlayer()	Adding computer-controlled player	Add AIPlayer	AIPlayer added successfully
Controller	getNextPlayer()	Get next player in a cycle	Players: [Player1, Player2, Player3]	Returns Player2
Controller	isGameOver()	Check game over by turn count	Max turns reached	Returns true

Controller	highlightCurrentPlayer(Player)	Highlight current player in view	Current player: Player1	Player1 is highlighted
MoveCommand	execute()	Valid move command	Player1 moves from room A to room B	Move succeeds, Player1 is in room B
PickUpItemCommand	execute()	Valid item pickup	Player1 picks up "Sword" in room A	Item successfully added to inventory
PickUpItemCommand	execute()	Invalid item pickup	Player1 tries to pick up "Shield" not in room	Returns "Item not found"
AttemptAttackCommand	execute()	Successful attack	Player1 attacks Player2	Attack succeeds, Player2 loses health
AttemptAttackCommand	execute()	Attack fails due to lack of weapon	Player1 attacks with no weapons	Returns "No weapons available"
GameView	renderWorld(World)	Render all spaces and players	World has 5 spaces, 2 players	Successfully renders the world
GameView	renderPlayer(Player)	Highlight active player	Active player: Player1	Player1 is highlighted
GameView	updateGameState(String)	Display game state update	"Player1 moved to room B"	Message displayed successfully
World	moveTarget()	Target moves to adjacent space	Target moves from room A to room B	Target successfully moves
World	getSpaces()	Retrieve all spaces in world	World has 5 spaces	Returns all space objects
World	getDescription()	Retrieve description for populated space	Room A with 2 items, 1 player	Returns room details

Player	move(Space)	Move to valid space	Player1 moves from room A to room B	Move succeeds
Player	pickUpItem(Item)	Add item to inventory	Player picks up "Shield"	Item successfully added to inventory
Player	attemptAttack(Player)	Valid attack	Player1 attacks Player2 in adjacent space	Attack succeeds, Player2 loses health
Player	canSee(Player)	Visibility check	Player1 in room A, Player2 in room B	Returns true
Pet	moveTo(Space)	Move to connected space	Pet moves from room A to room B	Move succeeds
Pet	blocksVisibility()	Visibility blocked by pet	Pet in room A	Returns true
PlayerInventory	addItem(Item)	Inventory full, item added based on priority	Inventory: full, Item: "Sword" (higher priority)	Replaces lower-priority item
PlayerInventory	removeItem(Item)	Remove item not in inventory	Inventory does not contain "Potion"	Remove fails, returns "Item not found"
Target	move(World)	Target moves to a connected space	Target moves from room A to room B	Move succeeds
Target	getDescription()	Retrieve target details	Target in room A	Returns detailed description