Name/s:	**Joshua Benedict B. Co | Renzel Vince E. Eleydo**	***ROG3 MCO2 GROUP 17***
Section:	**S28B**	
Date:	**June 30, 2025**	

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
BattleCharSelectionView(int playerID)	1	Normal: Create for Player 1	new BattleCharSelectionView(1)	Window titled "Player 1 Battle Character Selection" opens with all UI elements initialized.	Window titled "Player 1 Battle Character Selection" opens with all UI elements initialized.	pass
	2	Boundary: Create for invalid playerID (0)	new BattleCharSelectionView(0)	Window titled "Player Battle Character Selection" opens with default UI.	Window titled "Player Battle Character Selection" opens with default UI.	pass
	3	Edge: Create for high playerID (999)	new BattleCharSelectionView (999)	Window titled "Player Battle Character Selection" opens with default UI.	Window titled "Player Battle Character Selection" opens with default UI.	pass
setActionListener (ActionListener)	1	Normal: Attach a valid ActionListener	Listener that prints action	Buttons and dropdown fire listener on click/selection.	Buttons and dropdown fire listener on click/selection.	pass
	2	Edge: Set ActionListener multiple times	Attach two listeners in sequence	Only latest listener responds, no crash/duplication.	Only latest listener responds, no crash/duplication.	pass
	3	Invalid: Pass null as listener	setActionListener(null)	No action occurs, but app does not crash.	No action occurs, but app does not crash.	pass
updateCharacterList(String)	1	Normal: Update with character list	"Alice\nBob\nCecil"	Text area displays list formatted as input.	Text area displays list formatted as input.	pass
	2	Boundary: Empty list string	""	Text area displays blank.	Text area displays blank.	pass
	3	Invalid: Pass null	null	Text area is cleared or remains unchanged, no crash.	Text area is cleared or remains unchanged, no crash.	pass
confirmCharacterSelection (String)	1	Normal: User confirms selection	"Alice" (user clicks YES)	Method returns true.	Method returns true.	pass

	2	Boundary: User cancels selection	"Alice" (user clicks NO)	Method returns false.	Method returns false.	pass
confirmCharacterList(String)	3	Invalid: Pass null as name	null	Dialog appears with "null" in prompt, method returns user's choice.	Dialog appears with "null" in prompt, method returns user's choice.	pass
setCharacterOptions(String[])	1	Normal: Set valid character list	["Alice", "Bob"]	Dropdown lists "Alice" and "Bob".	Dropdown lists "Alice" and "Bob".	pass
	2	Boundary: Set empty array	[]	Dropdown is empty.	Dropdown is empty.	pass
	3	Invalid: Pass null	null	Dropdown is cleared, no crash.	Dropdown is cleared, no crash.	pass
resetDropdowns()	1	Normal: Reset after selection	Select first item, then resetDropdowns()	Dropdown selection becomes empty (no item selected).	Dropdown selection becomes empty (no item selected).	pass
	2	Boundary: Reset with empty dropdown	Dropdown already empty, call resetDropdowns()	No error, remains empty.	No error, remains empty.	pass
	3	Edge: Reset after setting new options	Set options, select one, reset	Selection is cleared as expected.	Selection is cleared as expected.	pass
getSelectedCharacter()	1	Normal: After selection	Select "Alice" in dropdown, call method	Returns "Alice".	Returns "Alice".	pass
	2	Boundary: Nothing selected	No selection, call method	Returns null.	Returns null.	pass
	3	Edge: After dropdown reset	Select then reset, call method	Returns null.	Returns null.	pass

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
BattleView(int mode)	1	Normal: Create PvP view	new BattleView(BattleView.BATTLE_PVP)	Window with PvP UI, correct title and panels.	Window with PvP UI, correct title and panels.	PASS
BattleView(int mode)	2	Normal: Create PvB view	new BattleView(BattleView.BATTLE_PVB)	Window with PvB UI, correct title and panels.	Window with PvB UI, correct title and panels.	PASS
BattleView(int mode)	3	Edge: Invalid mode	new BattleView(-1)	Defaults to PvB title/UI; no crash.	Defaults to PvB title/UI; no crash.	PASS
BattleView(Character, Character)	1	Normal: Construct with two characters	Two valid Character objects	View initializes in PvP mode, chars available.	View initializes in PvP mode, chars available.	PASS
BattleView(Character, Character)	2	Edge: Pass null for c1	new BattleView(null, validChar)	UI builds, but no info for P1; does not crash.	UI builds, but no info for P1; does not crash.	PASS

	BattleView(Character, Character)	3	Edge: Both null	new BattleView(null, null)	UI builds, both panels empty; no crash.	UI builds, both panels empty; no crash.	PASS
	BattleView(int, Character, Character)	1	Normal: Construct PvB with chars	new BattleView(BATTLE_PVB, c1, c2)	PvB UI, correct character info.	PvB UI, correct character info.	PASS
	BattleView(int, Character, Character)	2	Edge: High mode value	new BattleView(999, c1, c2)	Treated as PvB; no crash.	Treated as PvB; no crash.	PASS
	BattleView(int, Character, Character)	3	Invalid: Pass all null	new BattleView(-1, null, null)	UI builds, panels empty; no crash.	UI builds, panels empty; no crash.	PASS
	setActionListener (ActionListener)	1	Normal: Attach listener	Valid ActionListener	All buttons/dropdowns trigger listener.	All buttons/dropdowns trigger listener.	PASS
	setActionListener (ActionListener)	2	Boundary: Set twice	Attach 2 listeners in sequence	Only latest attached responds.	Only latest attached responds.	PASS
	setActionListener (ActionListener)	3	Invalid: Null listener	setActionListener(null)	No action, app remains stable.	No action, app remains stable.	PASS
	updateAbilityDropdown(int, List<String>)	1	Normal: Populate P1 options	(1, ["Attack", "Heal"])	Dropdown 1 lists both options.	Dropdown 1 lists both options.	PASS
	updateAbilityDropdown(int, List<String>)	2	Boundary: Empty list	(2, [])	Dropdown 2 is empty.	Dropdown 2 is empty.	PASS
	updateAbilityDropdown(int, List<String>)	3	Invalid: Null list	(1, null)	Dropdown cleared, no crash.	Dropdown cleared, no crash.	PASS
	setPlayerNameAndCharName (int, String)	1	Normal: Set for P2	(2, "Bob | Warrior")	P2 name area shows text.	P2 name area shows text.	PASS
	setPlayerNameAndCharName (int, String)	2	Boundary: Empty text	(1, "")	P1 name area cleared.	P1 name area cleared.	PASS
	setPlayerNameAndCharName (int, String)	3	Invalid: Null text	(2, null)	Area cleared or unchanged, no crash.	Area cleared or unchanged, no crash.	PASS
	setPlayerStatus(int, String)	1	Normal: Set P1 status	(1, "HP: 50/100")	Status area updates.	Status area updates.	PASS
	setPlayerStatus(int, String)	2	Boundary: Empty status	(2, "")	P2 status area cleared.	P2 status area cleared.	PASS
	setPlayerStatus(int, String)	3	Invalid: Null status	(1, null)	Area cleared/unchanged, no crash.	Area cleared/unchanged, no crash.	PASS
	setPlayerAbilitiesItems(int, String)	1	Normal: Set P1 items	(1, "Potion\nSword")	P1 abilities/items area updated.	P1 abilities/items area updated.	PASS
	setPlayerAbilitiesItems(int, String)	2	Boundary: Empty string	(2, "")	P2 abilities/items area cleared.	P2 abilities/items area cleared.	PASS
	setPlayerAbilitiesItems(int, String)	3	Invalid: Null	(1, null)	Area cleared/unchanged, no crash.	Area cleared/unchanged, no crash.	PASS

	appendBattleLog(String)	1	Normal: Append entry	"P1 attacks P2"	Log area appends line.	Log area appends line.	PASS
	appendBattleLog(String)	2	Boundary: Append empty	""	Log area unchanged or new line.	Log area unchanged or new line.	PASS
	appendBattleLog(String)	3	Invalid: Null	null	No change; no crash.	No change; no crash.	PASS
	resetBattleLog()	1	Normal: Log not empty	Call after appending	Log cleared, lastLogIndex = 0	Log cleared, lastLogIndex = 0	PASS
	resetBattleLog()	2	Boundary: Already empty	Call again	No effect, remains empty.	No effect, remains empty.	PASS
	resetBattleLog()	3	Multiple calls	Call several times	Remains cleared.	Remains cleared.	PASS
	setBattleOutcome(String)	1	Normal: Show result	"P1 wins!"	Outcome area shows centered result.	Outcome area shows centered result.	PASS
	setBattleOutcome(String)	2	Boundary: Empty string	""	Outcome area blank.	Outcome area blank.	PASS
	setBattleOutcome(String)	3	Invalid: Null	null	Area cleared/unchanged; no crash.	Area cleared/unchanged; no crash.	PASS
	clearBattleOutcome()	1	Normal: Clear after result	After setBattleOutcome, call clear	Outcome area blank.	Outcome area blank.	PASS
	clearBattleOutcome()	2	Boundary: Clear when already blank	Call when blank	No change.	No change.	PASS
	clearBattleOutcome()	3	Multiple clears	Call several times	Remains blank.	Remains blank.	PASS
	setRoundNumber(int)	1	Normal: Set round 3	3	Label says "Round 3"	Label says "Round 3"	PASS
	setRoundNumber(int)	2	Boundary: Round 0	0	Label says "Round 0"	Label says "Round 0"	PASS
	setRoundNumber(int)	3	Edge: Negative round	-5	Label says "Round -5"	Label says "Round -5"	PASS
	getSelectedAbility(int)	1	Normal: After select	Select "Attack" for P1, call with 1	Returns "Attack"	Returns "Attack"	PASS
	getSelectedAbility(int)	2	Boundary: Nothing selected	No selection, call	Returns null.	Returns null.	PASS
	getSelectedAbility(int)	3	Edge: Player 2 with empty dropdown	Empty options, call with 2	Returns null.	Returns null.	PASS
	setPlayer2ControlsEnabled (boolean)	1	Normal: Disable controls	FALSE	P2 ability dropdown & button disabled.	P2 ability dropdown & button disabled.	PASS
	setPlayer2ControlsEnabled (boolean)	2	Normal: Enable controls	TRUE	Both enabled.	Both enabled.	PASS

setPlayer2ControlsEnabled (boolean)	3	Toggle repeatedly	Alternate true/false	Controls reflect state.	Controls reflect state.	PASS
setBattleControlsEnabled (boolean)	1	Normal: Disable all	FALSE	Both P1/P2 controls disabled.	Both P1/P2 controls disabled.	PASS
setBattleControlsEnabled (boolean)	2	Enable all	TRUE	Both P1/P2 controls enabled.	Both P1/P2 controls enabled.	PASS
setBattleControlsEnabled (boolean)	3	Toggle repeatedly	Alternate true/false	Controls reflect state.	Controls reflect state.	PASS
setEndButtonsEnabled(boolean)	1	Enable end buttons	TRUE	Rematch/Return enabled.	Rematch/Return enabled.	PASS
setEndButtonsEnabled(boolean)	2	Disable end buttons	FALSE	Rematch/Return disabled.	Rematch/Return disabled.	PASS
setEndButtonsEnabled(boolean)	3	Edge: Toggle repeatedly	Alternate true/false	Buttons match state.	Buttons match state.	PASS
displayBattleStart(Character, Character)	1	Normal: Start with 2 chars	Two valid characters	Log and outcome cleared, ready for new battle.	Log and outcome cleared, ready for new battle.	PASS
displayBattleStart(Character, Character)	2	Edge: Null characters	Both null	Log/outcome cleared; no crash.	Log/outcome cleared; no crash.	PASS
displayBattleStart(Character, Character)	3	Multiple calls	Call repeatedly	Log/outcome cleared each time.	Log/outcome cleared each time.	PASS
displayTurnResults(CombatLog)	1	Normal: Add entries	Valid CombatLog w/entries	New entries appended to log.	New entries appended to log.	PASS
displayTurnResults(CombatLog)	2	Edge: No new entries	CombatLog unchanged	No new text added.	No new text added.	PASS
displayTurnResults(CombatLog)	3	Invalid: Null CombatLog	null	No update, no crash.	No update, no crash.	PASS
displayBattleEnd(Character)	1	Normal: Winner exists	Valid Character	Outcome area displays "<name> wins!", controls disabled, end buttons enabled.	Outcome area displays "<name> wins!", controls disabled, end buttons enabled.	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	PENDING
BattleModesView()	1	Normal: Create window	new BattleModesView()	Window titled "Battle Mode Selection" opens with all UI elements visible and enabled.	Window titled "Battle Mode Selection" opens with all UI elements visible and enabled.	PASS

BattleModesView()	2	Boundary: Create window twice in succession	"new BattleModesView(); new BattleModesView();"	Both windows appear independently, no conflicts or crash.	Both windows appear independently, no conflicts or crash.	PASS
BattleModesView()	3	Edge: User closes window (clicks X, confirms	User triggers close event, selects YES	Window closes successfully, application continues (unless main window).	Window closes successfully, application continues (unless main window).	PASS
setActionListener (ActionListener)	1	Normal: Attach working ActionListener	Listener that logs source button	Each button calls listener with correct action command.	Each button calls listener with correct action command.	PASS
setActionListener (ActionListener)	2	Boundary: Attach ActionListener multiple	Call setActionListener twice with different listeners	Only the latest listener responds to events.	Only the latest listener responds to events.	PASS
setActionListener (ActionListener)	3	Invalid: Pass null as listener	setActionListener(null)	No crash; buttons have no listener attached, do nothing.	No crash; buttons have no listener attached, do nothing.	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
CharacterAutoCreationView (String, int)	1	Normal: Open for Player 1	("Alice", 1)	Window titled "Player Alice Auto Character Creation" opens with initialized UI.	Window titled "Player Alice Auto Character Creation" opens with initialized UI.	PASS
CharacterAutoCreationView (String, int)	2	Boundary: Empty player name	("", 1)	Window titled "Player Auto Character Creation" (double space), UI ok.	Window titled "Player Auto Character Creation" (double space), UI ok.	PASS
CharacterAutoCreationView (String, int)	3	Edge: Invalid playerID	("Bob", -1)	Window opens, logo fallback, UI ok.	Window opens, logo fallback, UI ok.	PASS
addActionListener (ActionListener)	1	Normal: Attach working listener	Valid ActionListener	All three buttons fire listener with correct action command.	All three buttons fire listener with correct action command.	PASS
addActionListener (ActionListener)	2	Edge: Attach twice	Attach two listeners in sequence	Only the latest responds, no duplicate events.	Only the latest responds, no duplicate events.	PASS
addActionListener (ActionListener)	3	Invalid: Null listener	null	No action, app stable, no crash.	No action, app stable, no crash.	PASS
confirmCharacterCreation (String)	1	Normal: User confirms	"Aegis" (user clicks YES)	Method returns true.	Method returns true.	PASS
confirmCharacterCreation (String)	2	Boundary: User cancels	"Aegis" (user clicks NO)	Method returns false.	Method returns false.	PASS

	confirmCharacterCreation (String)	3	Invalid: Null name	null	Dialog prompt says "null", returns user's choice.	Dialog prompt says "null", returns user's choice.	PASS
	showGeneratedDetails(String)	1	Normal: Show details	"RACE: Elf\nCLASS: Mage"	Details area displays text.	Details area displays text.	PASS
	showGeneratedDetails(String)	2	Boundary: Empty string	""	Details area cleared.	Details area cleared.	PASS
	showGeneratedDetails(String)	3	Invalid: Null	null	Area cleared or unchanged, no crash.	Area cleared or unchanged, no crash.	PASS
	showInfoMessage(String)	1	Normal: Info dialog	"Character created!"	Info dialog appears, correct message.	Info dialog appears, correct message.	PASS
	showInfoMessage(String)	2	Boundary: Empty string	""	Dialog appears, no message.	Dialog appears, no message.	PASS
	showInfoMessage(String)	3	Invalid: Null	null	Dialog appears, title "Info", blank body, no crash.	Dialog appears, title "Info", blank body, no crash.	PASS
	showErrorMessage(String)	1	Normal: Error dialog	"Name required"	Error dialog appears with correct message.	Error dialog appears with correct message.	PASS
	showErrorMessage(String)	2	Boundary: Empty string	""	Error dialog, blank message.	Error dialog, blank message.	PASS
	showErrorMessage(String)	3	Invalid: Null	null	Dialog appears, title "Error", blank body, no crash.	Dialog appears, title "Error", blank body, no crash.	PASS
	resetFields()	1	Normal: Reset after input	Name/area non-empty, call method	Both fields cleared.	Both fields cleared.	PASS
	resetFields()	2	Already empty	Call method	Both fields remain empty.	Both fields remain empty.	PASS
	resetFields()	3	Multiple calls	Call method several times	Fields remain cleared.	Fields remain cleared.	PASS
	getCharacterName()	1	Normal: Name entered	Set name to "Jill", call method	Returns "Jill".	Returns "Jill".	PASS
	getCharacterName()	2	Boundary: Empty name	Name field empty	Returns "".	Returns "".	PASS
	getCharacterName()	3	Whitespace name	Name field is spaces	Returns "".	Returns "".	PASS
	setCharacterName(String)	1	Normal: Set name	"Fay"	Name field shows "Fay".	Name field shows "Fay".	PASS
	setCharacterName(String)	2	Boundary: Empty string	""	Name field cleared.	Name field cleared.	PASS
	setCharacterName(String)	3	Invalid: Null	null	Field cleared or unchanged, no crash.	Field cleared or unchanged, no crash.	PASS

getGeneratedCharacterDetails()	1	Normal: Details present	Details area filled	Returns details string (trimmed).	Returns details string (trimmed).	PASS
getGeneratedCharacterDetails()	2	Boundary: Empty area	Area blank	Returns "".	Returns "".	PASS
getGeneratedCharacterDetails()	3	Edge: Whitespace area	Area is spaces	Returns "".	Returns "".	PASS
setController (CharacterAutoCreationController)	1	Normal: Set controller	Valid controller object	Controller is set, no visible effect.	Controller is set, no visible effect.	PASS
setController (CharacterAutoCreationController)	2	Edge: Set controller to null	null	Field set to null, app stable.	Field set to null, app stable.	PASS
setController (CharacterAutoCreationController)	3	Multiple sets	Set twice	Controller field updates, no error.	Controller field updates, no error.	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
CharacterCreationManagementView(String, int)	1	Normal: Open for Player 1	("Alice", 1)	Window titled "Fatal Fantasy: TacticsAlice Character Creation Modes" opens, all UI present.	Window titled "Fatal Fantasy: TacticsAlice Character Creation Modes" opens, all UI present.	PASS
CharacterCreationManagementView(String, int)	2	Boundary: Empty player name	("", 2)	Window titled "Fatal Fantasy: TacticsCharacter Creation Modes", logo uses playerID=2, UI OK.	Window titled "Fatal Fantasy: TacticsCharacter Creation Modes", logo uses playerID=2, UI OK.	PASS
CharacterCreationManagementView(String, int)	3	Edge: Invalid playerID	("Bob", -1)	Window opens, logo fallback or missing, UI remains stable.	Window opens, logo fallback or missing, UI remains stable.	PASS
showInfoMessage(String)	1	Normal: Show info dialog	"Character created successfully!"	Info dialog pops up, correct message shown.	Info dialog pops up, correct message shown.	PASS
showInfoMessage(String)	2	Boundary: Empty string	""	Info dialog pops up, blank message.	Info dialog pops up, blank message.	PASS
showInfoMessage(String)	3	Invalid: Null	null	Info dialog with title "Info", blank message, no crash.	Info dialog with title "Info", blank message, no crash.	PASS

showErrorMessage(String)	1	Normal: Show error dialog	"Name is required"	Error dialog appears with message.	Error dialog appears with message.	PASS
showErrorMessage(String)	2	Boundary: Empty string	""	Error dialog, blank message.	Error dialog, blank message.	PASS
showErrorMessage(String)	3	Invalid: Null	null	Error dialog, title "Error", blank message, no crash.	Error dialog, title "Error", blank message, no crash.	PASS
setActionListener (ActionListener)	1	Normal: Attach working listener	Valid ActionListener	All three buttons call listener when clicked.	All three buttons call listener when clicked.	PASS
setActionListener (ActionListener)	2	Edge: Attach twice	Set two listeners in sequence	Only the last one responds.	Only the last one responds.	PASS
setActionListener (ActionListener)	3	Invalid: Null listener	null	Buttons do nothing, app remains stable.	Buttons do nothing, app remains stable.	PASS
getPlayerName()	1	Normal: Player name set	"Alice" in constructor, call method	Returns "Alice".	Returns "Alice".	PASS
getPlayerName()	2	Boundary: Empty player name	"" in constructor, call method	Returns "".	Returns "".	PASS
getPlayerName()	3	Edge: Unusual characters	"@@!!" in constructor, call method	Returns "@@!!".	Returns "@@!!".	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
CharacterDeleteView(int playerID)	1	Normal: Player 1	1	Window titled "Player 1 Character Deletion" opens, all UI present.	Window titled "Player 1 Character Deletion" opens, all UI present.	PASS
CharacterDeleteView(int playerID)	2	Edge: Player 2	2	Window titled "Player 2 Character Deletion" opens, all UI present.	Window titled "Player 2 Character Deletion" opens, all UI present.	PASS
CharacterDeleteView(int playerID)	3	Invalid: Negative ID	-1	Window titled "Player -1 Character Deletion", UI still opens, no crash.	Window titled "Player -1 Character Deletion", UI still opens, no crash.	PASS
setActionListener (ActionListener)	1	Normal: Attach listener	Valid ActionListener	All buttons and dropdown call listener on user action.	All buttons and dropdown call listener on user action.	PASS
setActionListener (ActionListener)	2	Boundary: Attach twice	Attach two listeners in sequence	Only last listener responds.	Only last listener responds.	PASS
setActionListener (ActionListener)	3	Invalid: Null listener	null	Buttons do nothing, app remains stable.	Buttons do nothing, app remains stable.	PASS

	showInfoMessage(String)	1	Normal: Show info	"Character deleted!"	Info dialog with correct message appears.	Info dialog with correct message appears.	PASS
	showInfoMessage(String)	2	Boundary: Empty message	""	Info dialog, blank message, no crash.	Info dialog, blank message, no crash.	PASS
	showInfoMessage(String)	3	Invalid: Null	null	Info dialog, blank message, no crash.	Info dialog, blank message, no crash.	PASS
	showErrorMessage(String)	1	Normal: Show error	"No character selected"	Error dialog appears with message.	Error dialog appears with message.	PASS
	showErrorMessage(String)	2	Boundary: Empty message	""	Error dialog, blank message, no crash.	Error dialog, blank message, no crash.	PASS
	showErrorMessage(String)	3	Invalid: Null	null	Error dialog, blank message, no crash.	Error dialog, blank message, no crash.	PASS
	confirmCharacterDeletion(String)	1	Normal: User confirms	"Aegis" (user clicks YES)	Method returns true.	Method returns true.	PASS
	confirmCharacterDeletion(String)	2	Boundary: User cancels	"Aegis" (user clicks NO)	Method returns false.	Method returns false.	PASS
	confirmCharacterDeletion(String)	3	Invalid: Null name	null	Dialog prompt says "null", returns user's choice.	Dialog prompt says "null", returns user's choice.	PASS
	updateCharacterList(String)	1	Normal: Update with list	"Alice\nBob\nCecil"	List area displays characters, formatted as input.	List area displays characters, formatted as input.	PASS
	updateCharacterList(String)	2	Boundary: Empty string	""	List area cleared, no crash.	List area cleared, no crash.	PASS
	updateCharacterList(String)	3	Invalid: Null	null	List area cleared/unchanged, no crash.	List area cleared/unchanged, no crash.	PASS
	setCharacterOptions(String[])	1	Normal: Set options	["Alice","Bob"]	Dropdown lists both, Delete enabled.	Dropdown lists both, Delete enabled.	PASS
	setCharacterOptions(String[])	2	Boundary: Empty array	[]	Dropdown empty, Delete disabled.	Dropdown empty, Delete disabled.	PASS
	setCharacterOptions(String[])	3	Invalid: Null array	null	Dropdown cleared, Delete disabled, no crash.	Dropdown cleared, Delete disabled, no crash.	PASS
	setDeleteEnabled(boolean)	1	Normal: Enable button	TRUE	Delete button enabled.	Delete button enabled.	PASS
	setDeleteEnabled(boolean)	2	Disable button	FALSE	Delete button disabled.	Delete button disabled.	PASS
	setDeleteEnabled(boolean)	3	Toggle repeatedly	Alternate true/false	Button state matches enabled/disabled.	Button state matches enabled/disabled.	PASS

getSelectedCharacter()	1	Normal: One selected	"Alice" selected in dropdown	Returns "Alice".	Returns "Alice".	PASS
getSelectedCharacter()	2	Boundary: None selected	Nothing selected	Returns null.	Returns null.	PASS
getSelectedCharacter()	3	After setCharacterOptions([])	Set options to empty, call method	Returns null.	Returns null.	PASS
getDeleteButton()	1	Normal: Get button	Call after construction	Returns btnDelete instance (not null).	Returns btnDelete instance (not null).	PASS
getDeleteButton()	2	After disabling	Disable button, get button	Returns btnDelete, .isEnabled() is false.	Returns btnDelete, . isEnabled() is false.	PASS
getDeleteButton()	3	Multiple calls	Call multiple times	Same instance each time.	Same instance each time.	PASS
getReturnButton()	1	Normal: Get button	Call after construction	Returns btnReturn instance (not null).	Returns btnReturn instance (not null).	PASS
getReturnButton()	2	After disabling	Disable button, get button	Returns btnReturn, .isEnabled() matches.	Returns btnReturn, . isEnabled() matches.	PASS
getReturnButton()	3	Multiple calls	Call multiple times	Same instance each time.	Same instance each time.	PASS
getCharacterDropdown()	1	Normal: Get dropdown	Call after construction	Returns JComboBox instance.	Returns JComboBox instance.	PASS
getCharacterDropdown()	2	After setCharacterOptions	Set options, get dropdown	Dropdown contains correct items.	Dropdown contains correct items.	PASS
getCharacterDropdown()	3	Multiple calls	Call multiple times	Same instance each time.	Same instance each time.	PASS
setCharacterInfoLabel(String)	1	Normal: Set info label	"Player 1 (Editing)"	Info label displays new text.	Info label displays new text.	PASS
setCharacterInfoLabel(String)	2	Boundary: Empty string	""	Info label cleared.	Info label cleared.	PASS
setCharacterInfoLabel(String)	3	Invalid: Null	null	Info label cleared/unchanged, no crash.	Info label cleared/unchanged, no crash.	PASS
getPlayerID()	1	Normal: Player 1	Construct with 1, call	Returns 1.	Returns 1.	PASS
getPlayerID()	2	Player 2	Construct with 2, call	Returns 2.	Returns 2.	PASS
getPlayerID()	3	Edge: Negative ID	Construct with -1, call	Returns -1.	Returns -1.	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
CharacterEditView(int playerID)	1	Normal: Player 1	1	Window "Player 1 Character Edit" opens, all UI present.	Window "Player 1 Character Edit" opens, all UI present.	PASS
CharacterEditView(int playerID)	2	Edge: Player 2	2	Window "Player 2 Character Edit" opens.	Window "Player 2 Character Edit" opens.	PASS
CharacterEditView(int playerID)	3	Invalid: Negative ID	-1	Window "Player -1 Character Edit", UI opens, no crash.	Window "Player -1 Character Edit", UI opens, no crash.	PASS
setActionListener (ActionListener)	1	Normal: Attach listener	Valid ActionListener	All buttons and dropdowns fire listener on action.	All buttons and dropdowns fire listener on action.	PASS
setActionListener (ActionListener)	2	Attach twice	Attach two listeners sequentially	Only last one responds.	Only last one responds.	PASS
setActionListener (ActionListener)	3	Invalid: Null listener	null	No listeners, app stable.	No listeners, app stable.	PASS
setCharacterOptions(String[])	1	Normal: Add characters	["Alice","Bob"]	Dropdown shows both, selection resets.	Dropdown shows both, selection resets.	PASS
setCharacterOptions(String[])	2	Empty array	[]	Dropdown cleared.	Dropdown cleared.	PASS
setCharacterOptions(String[])	3	Invalid: Null	null	Dropdown cleared, no crash.	Dropdown cleared, no crash.	PASS
setAbilityOptions(int, String[])	1	Normal: Set slot 1	(1, ["Fire","Heal"])	Ability 1 dropdown lists both.	Ability 1 dropdown lists both.	PASS
setAbilityOptions(int, String[])	2	Empty array	(2, [])	Ability 2 dropdown empty.	Ability 2 dropdown empty.	PASS
setAbilityOptions(int, String[])	3	Invalid: Slot 0	(0, ["Fire"])	Throws IllegalArgumentException.	Throws IllegalArgumentException.	PASS
setMagicItemOptions(String[])	1	Normal: Set items	["Ring","Cape"]	Magic item dropdown lists both.	Magic item dropdown lists both.	PASS
setMagicItemOptions(String[])	2	Empty array	[]	Dropdown empty.	Dropdown empty.	PASS
setMagicItemOptions(String[])	3	Invalid: Null	null	Dropdown cleared, no crash.	Dropdown cleared, no crash.	PASS
setSelectedAbility(int, String)	1	Normal: Select slot 2	(2, "Heal")	Ability 2 dropdown selects "Heal".	Ability 2 dropdown selects "Heal".	PASS

setSelectedAbility(int, String)	2	Invalid: Slot 0	(0, "Fire")	Throws IllegalArgumentException.	Throws IllegalArgumentException.	PASS
setSelectedAbility(int, String)	3	Invalid: Null name	(1, null)	Dropdown selection cleared.	Dropdown selection cleared.	PASS
setSelectedMagicItem(String)	1	Normal: Select item	"Ring"	Magic item dropdown selects "Ring".	Magic item dropdown selects "Ring".	PASS
setSelectedMagicItem(String)	2	Non-existent item	"Crown"	Selection not found, remains unchanged.	Selection not found, remains unchanged.	PASS
setSelectedMagicItem(String)	3	Null	null	Selection cleared or unchanged, no crash.	Selection cleared or unchanged, no crash.	PASS
getCharacterDropdown()	1	Normal: Call	After construction	Returns JComboBox instance.	Returns JComboBox instance.	PASS
getCharacterDropdown()	2	After setCharacterOptions	Set options, get dropdown	Contains correct items.	Contains correct items.	PASS
getCharacterDropdown()	3	Multiple calls	Call multiple times	Same instance returned.	Same instance returned.	PASS
getMagicItemDropdown()	1	Normal: Call	After construction	Returns JComboBox instance.	Returns JComboBox instance.	PASS
getMagicItemDropdown()	2	After setMagicItemOptions	Set items, get dropdown	Contains correct items.	Contains correct items.	PASS
getMagicItemDropdown()	3	Multiple calls	Call multiple times	Same instance returned.	Same instance returned.	PASS
resetFields()	1	Normal: After user selection	Fields are not -1, call resetFields	All dropdowns set to -1 (no selection).	All dropdowns set to -1 (no selection).	PASS
resetFields()	2	Already reset	Fields at -1, call resetFields	No change.	No change.	PASS
resetFields()	3	Multiple calls	Call several times	Fields remain reset.	Fields remain reset.	PASS
confirmCharacterEdit(String)	1	Normal: User confirms	"Alice" (user clicks YES)	Returns true.	Returns true.	PASS
confirmCharacterEdit(String)	2	User cancels	"Alice" (user clicks NO)	Returns false.	Returns false.	PASS
confirmCharacterEdit(String)	3	Null name	null	Dialog shows "null", returns user"s choice.	Dialog shows "null", returns user"s choice.	PASS
showInfoMessage(String)	1	Normal: Show info	"Edit successful!"	Info dialog appears.	Info dialog appears.	PASS
showInfoMessage(String)	2	Empty message	""	Dialog with blank message.	Dialog with blank message.	PASS
showInfoMessage(String)	3	Null	null	Dialog with blank message, no crash.	Dialog with blank message, no crash.	PASS

showErrorMessage(String)	1	Normal: Show error	"Invalid ability"	Error dialog appears.	Error dialog appears.	PASS
showErrorMessage(String)	2	Empty message	""	Error dialog with blank message.	Error dialog with blank message.	PASS
showErrorMessage(String)	3	Null	null	Error dialog with blank message, no crash.	Error dialog with blank message, no crash.	PASS
getSelectedCharacter()	1	Normal: One selected	"Alice" selected	Returns "Alice".	Returns "Alice".	PASS
getSelectedCharacter()	2	None selected	No selection	Returns null.	Returns null.	PASS
getSelectedCharacter()	3	After resetFields	Call after reset	Returns null.	Returns null.	PASS
getSelectedAbilities()	1	Normal: Three selected	3 dropdowns set	Returns array of 3 ability names.	Returns array of 3 ability names.	PASS
getSelectedAbilities()	2	None selected	All dropdowns -1	Returns array of 3 nulls.	Returns array of 3 nulls.	PASS
getSelectedAbilities()	3	Ability 4 visible	Ability 4 shown & set, call method	Returns array of 4 ability names.	Returns array of 4 ability names.	PASS
getSelectedAbility(int)	1	Normal: Slot 1 selected	1 after selecting "Fire"	Returns "Fire".	Returns "Fire".	PASS
getSelectedAbility(int)	2	Invalid: Slot 0	0	Throws IllegalArgumentException.	Throws IllegalArgumentException.	PASS
getSelectedAbility(int)	3	Unselected slot	Valid slot, nothing selected	Returns null.	Returns null.	PASS
getSelectedMagicItem()	1	Normal: Item selected	"Ring" selected	Returns MagicItem object represer	Returns MagicItem object "	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
CharacterListViewingView(int playerID)	1	Normal: Player 1	1	Window titled "Player 1 Characters Viewing" opens, UI present.	Window titled "Player 1 Characters Viewing" opens, UI present.	PASS
CharacterListViewingView(int playerID)	2	Edge: Player 2	2	Window titled "Player 2 Characters Viewing" opens, UI present.	Window titled "Player 2 Characters Viewing" opens, UI present.	PASS
CharacterListViewingView(int playerID)	3	Invalid: Negative ID	-1	Window titled "Player -1 Characters Viewing", UI present, no crash.	Window titled "Player -1 Characters Viewing", UI present, no crash.	PASS

setActionListener (ActionListener)	1	Normal: Attach listener	Valid ActionListener	Both buttons call listener on user action.	Both buttons call listener on user action.	PASS
setActionListener (ActionListener)	2	Boundary: Attach twice	Attach two listeners sequentially	Only last responds, or both fire.	Only last responds, or both fire.	PASS
setActionListener (ActionListener)	3	Invalid: Null listener	null	Buttons do nothing, app remains stable.	Buttons do nothing, app remains stable.	PASS
updateCharacterList(String)	1	Normal: Display character list	"Alice\nBob\nCecil"	List area displays characters, formatted as input.	List area displays characters, formatted as input.	PASS
updateCharacterList(String)	2	Boundary: Empty string	""	List area cleared, no crash.	List area cleared, no crash.	PASS
updateCharacterList(String)	3	Invalid: Null	null	List area cleared or unchanged, no crash.	List area cleared or unchanged, no crash.	PASS
showInfoMessage(String)	1	Normal: Show info	"View details loaded"	Info dialog with correct message appears.	Info dialog with correct message appears.	PASS
showInfoMessage(String)	2	Boundary: Empty message	""	Info dialog, blank message, no crash.	Info dialog, blank message, no crash.	PASS
showInfoMessage(String)	3	Invalid: Null	null	Info dialog, blank message, no crash.	Info dialog, blank message, no crash.	PASS
showErrorMessage(String)	1	Normal: Show error	"No character found"	Error dialog appears with message.	Error dialog appears with message.	PASS
showErrorMessage(String)	2	Boundary: Empty message	""	Error dialog, blank message, no crash.	Error dialog, blank message, no crash.	PASS
showErrorMessage(String)	3	Invalid: Null	null	Error dialog, blank message, no crash.	Error dialog, blank message, no crash.	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
CharacterManagementMenuView()	1	Normal: Open menu	N/A	Window titled "Character Management" opens, all buttons/labels present.	Window titled "Character Management" opens, all buttons/labels present.	PASS
CharacterManagementMenuView()	2	Boundary: Window resizable	N/A	Window not resizable.	Window not resizable.	PASS
CharacterManagementMenuView()	3	Edge: Multiple instances	Create 2 windows	Both windows operate independently, no crash.	Both windows operate independently, no crash.	PASS
setPlayer1Name(String)	1	Normal: Set valid name	"Alice"	Button 1 label: "Manage Alice", enabled.	Button 1 label: "Manage Alice", enabled.	PASS

setPlayer1Name(String)	2	Boundary: Empty string	""	Button 1 label: "Manage Player 1", disabled.	Button 1 label: "Manage Player 1", disabled.	PASS
setPlayer1Name(String)	3	Invalid: Null	null	Button 1 label: "Manage Player 1", disabled.	Button 1 label: "Manage Player 1", disabled.	PASS
setPlayer2Name(String)	1	Normal: Set valid name	"Bob"	Button 2 label: "Manage Bob", enabled.	Button 2 label: "Manage Bob", enabled.	PASS
setPlayer2Name(String)	2	Boundary: Empty string	""	Button 2 label: "Manage Player 2", disabled.	Button 2 label: "Manage Player 2", disabled.	PASS
setPlayer2Name(String)	3	Invalid: Null	null	Button 2 label: "Manage Player 2", disabled.	Button 2 label: "Manage Player 2", disabled.	PASS
setActionListener (ActionListener)	1	Normal: Attach listener	Valid ActionListener	All three buttons call listener on click, correct action commands.	All three buttons call listener on click, correct action commands.	PASS
setActionListener (ActionListener)	2	Boundary: Attach twice	Attach two listeners in sequence	Only the last one responds, or multiple fire.	Only the last one responds, or multiple fire.	PASS
setActionListener (ActionListener)	3	Invalid: Null listener	null	Buttons do nothing, app remains stable.	Buttons do nothing, app remains stable.	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
CharacterManualCreationView (String)	1	Normal: Valid player name	"Alice"	Window titled for "Alice" opens with empty fields, UI present.	Window titled for "Alice" opens with empty fields, UI present.	PASS
CharacterManualCreationView (String)	2	Boundary: Empty name	""	Window opens, title includes empty name.	Window opens, title includes empty name.	PASS
CharacterManualCreationView (String)	3	Invalid: Null name	null	Window opens, title has "null", UI present, no crash.	Window opens, title has "null", UI present, no crash.	PASS
CharacterManualCreationView (String,int)	1	Normal: Player 2	"Bob", 2	Window title: "Manual Character Creation - Bob", Player 2 logo shown.	Window title: "Manual Character Creation - Bob", Player 2 logo shown.	PASS
CharacterManualCreationView (String,int)	2	Boundary: ID zero	"Cecil", 0	Title includes Player 0, UI present.	Title includes Player 0, UI present.	PASS
CharacterManualCreationView (String,int)	3	Invalid: Negative ID	"Daisy", -5	Title includes Player -5, UI present, no crash.	Title includes Player -5, UI present, no crash.	PASS

	addCreateCharacterListener (ActionListener)	1	Normal: Add listener	Valid ActionListener	"Create button fires listener on click."	"Create button fires listener on click."	PASS
	addCreateCharacterListener (ActionListener)	2	Boundary: Add twice	Add two listeners	"Both fire, or only last; no crash."	"Both fire, or only last; no crash."	PASS
	addCreateCharacterListener (ActionListener)	3	Invalid: Null	null	No action, no crash.	No action, no crash.	PASS
	addReturnListener (ActionListener)	1	Normal: Add listener	Valid ActionListener	"Return button fires listener."	"Return button fires listener."	PASS
	addReturnListener (ActionListener)	2	Boundary: Add twice	Add two listeners	"Both fire, or only last; no crash."	"Both fire, or only last; no crash."	PASS
	addReturnListener (ActionListener)	3	Invalid: Null	null	No action, no crash.	No action, no crash.	PASS
	addClassDropdownListener (ActionListener)	1	Normal: Add listener	Valid ActionListener	"Listener fires on class change."	"Listener fires on class change."	PASS
	addClassDropdownListener (ActionListener)	2	Boundary: Add twice	Add two listeners	"Both fire, or only last; no crash."	"Both fire, or only last; no crash."	PASS
	addClassDropdownListener (ActionListener)	3	Invalid: Null	null	No action, no crash.	No action, no crash.	PASS
	addRaceDropdownListener (ActionListener)	1	Normal: Add listener	Valid ActionListener	"Listener fires on race change."	"Listener fires on race change."	PASS
	addRaceDropdownListener (ActionListener)	2	Boundary: Add twice	Add two listeners	"Both fire, or only last; no crash."	"Both fire, or only last; no crash."	PASS
	addRaceDropdownListener (ActionListener)	3	Invalid: Null	null	No action, no crash.	No action, no crash.	PASS
	setRaceOptions(String[])	1	Normal: Set 3 races	"[""Elf"",""Human"",""Dwarf""]"	Dropdown has 3 items.	Dropdown has 3 items.	PASS
	setRaceOptions(String[])	2	Boundary: Empty	[]	Dropdown cleared, no crash.	Dropdown cleared, no crash.	PASS
	setRaceOptions(String[])	3	Invalid: Null	null	Dropdown cleared, no crash.	Dropdown cleared, no crash.	PASS
	setClassOptions(String[])	1	Normal: Set 3 classes	"[""Mage"",""Fighter"",""Cleric""]"	Dropdown has 3 items.	Dropdown has 3 items.	PASS
	setClassOptions(String[])	2	Boundary: Empty	[]	Dropdown cleared.	Dropdown cleared.	PASS
	setClassOptions(String[])	3	Invalid: Null	null	Dropdown cleared, no crash.	Dropdown cleared, no crash.	PASS
	setAbilityOptions(int,String[])	1	Normal: Slot 2, 3 abilities	2,[""Slash"",""Heal"",""Block""]	Ability 2 dropdown has 3 items.	Ability 2 dropdown has 3 items.	PASS

	setAbilityOptions(int,String[])	2	Boundary: Slot 4	4,[""Ultimate""]	Ability 4 dropdown has 1 item.	Ability 4 dropdown has 1 item.	PASS
	setAbilityOptions(int,String[])	3	Invalid: Invalid slot	5,[""Fire""]	Exception thrown (IllegalArgumentException).	Exception thrown (IllegalArgumentException).	PASS
	resetFields()	1	Normal: With data entered	User fills fields, call reset	All fields reset to blank/unselected.	All fields reset to blank/unselected.	PASS
	resetFields()	2	Boundary: Already blank	Fields already empty, call reset	No change, no crash.	No change, no crash.	PASS
	resetFields()	3	Invalid: Dropdown with no items	Remove all options, call reset	No crash, fields still reset.	No crash, fields still reset.	PASS
	getCharacterName()	1	Normal: Name entered	"Ezra"	Returns "Ezra".	Returns "Ezra".	PASS
	getCharacterName()	2	Boundary: Empty field	(blank field)	Returns ""	Returns ""	PASS
	getCharacterName()	3	Special chars	" Lancelot "	Returns "Lancelot" (trimmed).	Returns "Lancelot" (trimmed).	PASS
	getSelectedRace()	1	Normal: Race selected	Select "Elf"	Returns "Elf".	Returns "Elf".	PASS
	getSelectedRace()	2	Boundary: Nothing selected	No selection	Returns null.	Returns null.	PASS
	getSelectedRace()	3	Invalid: Index out of bounds	Remove all, select	Returns null.	Returns null.	PASS
	getSelectedClass()	1	Normal: Class selected	Select "Mage"	Returns "Mage".	Returns "Mage".	PASS
	getSelectedClass()	2	Boundary: Nothing selected	No selection	Returns null.	Returns null.	PASS
	getSelectedClass()	3	Invalid: Index out of bounds	Remove all, select	Returns null.	Returns null.	PASS
	getSelectedAbilities()	1	Normal: Three selected	Select 3 abilities	Array of 3 selected strings.	Array of 3 selected strings.	PASS
	getSelectedAbilities()	2	Boundary: Less than 3 selected	Select 2, leave 1 blank	Array with null for unselected.	Array with null for unselected.	PASS
	getSelectedAbilities()	3	Invalid: Ability count mismatch	Internally set to 4, only 3 shown	Returns as per abilityCount.	Returns as per abilityCount.	PASS
	showInfoMessage(String)	1	Normal: Show info	"Character created!"	Info dialog with correct message.	Info dialog with correct message.	PASS
	showInfoMessage(String)	2	Boundary: Empty	""	Blank dialog, no crash.	Blank dialog, no crash.	PASS
	showInfoMessage(String)	3	Invalid: Null	null	Blank dialog, no crash.	Blank dialog, no crash.	PASS

showErrorMessage(String)	1	Normal: Show error	"Failed to create!"	Error dialog with correct	Error dialog with correct	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor	1	Normal: Player 1	1	Window titled "Player 1 Character Specific Viewing" opens with correct logo and fields	Window titled "Player 1 Character Specific Viewing" opens with correct logo and fields	PASS
Constructor	2	Boundary: Player 2	2	Window titled "Player 2 Character Specific Viewing", Player 2 logo shown	Window titled "Player 2 Character Specific Viewing", Player 2 logo shown	PASS
Constructor	3	Invalid: Negative ID	-3	Window titled "Player -3 Character Specific Viewing", logo for -3, no crash	Window titled "Player -3 Character Specific Viewing", logo for -3, no crash	PASS
setActionListener (ActionListener)	1	Normal: Add listener	Valid ActionListener	Listener fires on dropdown or return button	Listener fires on dropdown or return button	PASS
setActionListener (ActionListener)	2	Boundary: Add twice	Add two listeners	Both fire, or only last; no crash	Both fire, or only last; no crash	PASS
setActionListener (ActionListener)	3	Invalid: Null	null	No action, no crash	No action, no crash	PASS
updateCharacterDetails(String)	1	Normal: Show character	"Elf - LV5"	Text area shows "Elf - LV5"	Text area shows "Elf - LV5"	PASS
updateCharacterDetails(String)	2	Boundary: Empty string	""	Text area cleared	Text area cleared	PASS
updateCharacterDetails(String)	3	Invalid: Null	null	Text area blank, no crash	Text area blank, no crash	PASS
setCharacterOptions(String[])	1	Normal: 3 names	"[""Ezra"",""Merlin"",""Gwen""]"	Dropdown has all 3 options	Dropdown has all 3 options	PASS
setCharacterOptions(String[])	2	Boundary: Empty array	[]	Dropdown empty, no crash	Dropdown empty, no crash	PASS
setCharacterOptions(String[])	3	Invalid: Null	null	Dropdown cleared, no crash	Dropdown cleared, no crash	PASS
setCharacterSelectionEnabled (boolean)	1	Normal: Enable	TRUE	Dropdown is enabled	Dropdown is enabled	PASS

setCharacterSelectionEnabled (boolean)	2	Normal: Disable	FALSE	Dropdown is disabled	Dropdown is disabled	PASS
setCharacter selection enabled (boolean)	3	Multiple calls	true then false then true	State toggles accordingly, no crash	State toggles accordingly, no crash	PASS
getSelectedCharacter()	1	Normal: One selected	Select "Gwen"	Returns "Gwen"	Returns "Gwen"	PASS
getSelectedCharacter()	2	Boundary: None selected	No selection	Returns null	Returns null	PASS
getSelectedCharacter()	3	Invalid: Dropdown empty	Remove all, select	Returns null	Returns null	PASS
showInfoMessage(String)	1	Normal: Show info	"Character loaded!"	Info dialog with correct message	Info dialog with correct message	PASS
showInfoMessage(String)	2	Boundary: Empty	""	Blank info dialog, no crash	Blank info dialog, no crash	PASS
showInfoMessage(String)	3	Invalid: Null	null	Blank info dialog, no crash	Blank info dialog, no crash	PASS
showErrorMessage(String)	1	Normal: Show error	"Load failed!"	Error dialog with correct message	Error dialog with correct message	PASS
showErrorMessage(String)	2	Boundary: Empty	""	Blank error dialog, no crash	Blank error dialog, no crash	PASS
showErrorMessage(String)	3	Invalid: Null	null	Blank error dialog, no crash	Blank error dialog, no crash	PASS
resetView()	1	Normal: Reset after use	After selection + details, call reset	Dropdown unselected, text area cleared	Dropdown unselected, text area cleared	PASS
resetView()	2	Boundary: Already empty	No selection/text, call reset	No crash, remains blank	No crash, remains blank	PASS
resetView()	3	Invalid: No dropdown options	Remove all, call reset	No crash, dropdown at -1, text cleared	No crash, dropdown at -1, text cleared	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor HallOfFameCharactersView()	1	Open window normally	–	Window opens titled "Fatal Fantasy: Tactics	Window opens titled "Fatal Fantasy: Tactics	PASS
Constructor HallOfOfFameCharactersView()	2	Open and close window (confirm exit)	–	Window closes after user confirms	Window closes after user confirms	PASS
Constructor HallOfOfFameCharactersView()	3	Open and close window (cancel exit)	–	Window stays open after user cancels	Window stays open after user cancels	PASS

setActionListener (ActionListener)	1	Add valid ActionListener	listener that logs clicks	"Return" button triggers listener on click	"Return" button triggers listener on click	PASS
setActionListener (ActionListener)	2	Add ActionListener twice	same listener twice	Listener runs once or twice, but no error/crash	Listener runs once or twice, but no error/crash	PASS
setActionListener (ActionListener)	3	Add null listener	null	No crash, button does not throw error	No crash, button does not throw error	PASS
updateTopCharactersList(String)	1	Display regular list	"Auron\nYuna\nWakka"	Text area shows each name on a new line	Text area shows each name on a new line	PASS
updateTopCharactersList(String)	2	Display empty list	""	Text area is empty	Text area is empty	PASS
updateTopCharactersList(String)	3	Display null	null	Text area is blank, no crash	Text area is blank, no crash	PASS
showInfoMessage(String)	1	Show info message	"Saved!"	Popup appears with info icon and "Saved!"	Popup appears with info icon and "Saved!"	PASS
showInfoMessage(String)	2	Show empty info message	""	Popup appears, blank message	Popup appears, blank message	PASS
showInfoMessage(String)	3	Show null info message	null	Popup appears, blank message	Popup appears, blank message	PASS
showErrorMessage(String)	1	Show error message	"Failed to load!"	Popup appears with error icon and "Failed to load!"	Popup appears with error icon and "Failed to load!"	PASS
showErrorMessage(String)	2	Show empty error message	""	Popup appears, blank message	Popup appears, blank message	PASS
showErrorMessage(String)	3	Show null error message	null	Popup appears, blank message	Popup appears, blank message	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor HallOfFameManagementView()	1	Opens window with correct title	None	Window titled "Fatal Fantasy: Tactics	Expected Output	PASS
Constructor HallOfFameManagementView()	2	Window cannot be closed without confirmation	User clicks window close (X)	Confirmation dialog appears; closes only on "Yes"	Window titled "Fatal Fantasy: Tactics	PASS
Constructor HallOfFameManagementView()	3	Window stays open if "No" is clicked	User clicks X and selects "No"	Window stays open	Confirmation dialog appears; closes only on "Yes"	PASS
initUI()	1	All buttons visible	None	All 3 buttons are visible: Top Players, Top Characters, Return to Menu	Window stays open	PASS

initUI()	2	All action commands set	None	Each button has correct action command string	All 3 buttons are visible: Top Players, Top Characters, Return to Menu	PASS
setActionListener (ActionListener)	1	Listener attached to all buttons	Valid ActionListener	Clicking each button triggers the listener	Each button has correct action command string	PASS
setActionListener (ActionListener)	2	Listener is null	null	No exceptions thrown, buttons do not throw error	Clicking each button triggers the listener	PASS
setController (HallOfFameController)	1	Controller is set and wired	Valid HallOfFameController	Clicking each button triggers controller action	No exceptions thrown, buttons do not throw error	PASS
showInfoMessage(String)	1	Show info message	"Saved!"	Info dialog with "Saved!" text and info icon	Clicking each button triggers controller action	PASS
showInfo message(String)	2	Show info message (empty string)	""	Info dialog with blank message	Info dialog with "Saved!" text and info icon	PASS
show info message (String)	3	Show info info message (null)	null	Info dialog with blank message	Info dialog with blank message	PASS
showErrorMessage(String)	1	Show error message	"Error loading!"	Error dialog with "Error loading!" and error icon	Info dialog with blank message	PASS
showErrorMessage(String)	2	Show error message (empty string)	""	Error dialog with blank message	Error dialog with "Error loading!" and error icon	PASS
show error message (String)	3	Show error error message (null)	null	Error dialog with blank message	Error dialog with blank message	PASS
					Error dialog with blank message	

Method	#	#	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor HallOfFamePlayersView()		1	Open window normally	–	Window opens titled "Fatal Fantasy: Tactics	Window opens titled "Fatal Fantasy: Tactics	PASS
Constructor HallOfFamePlayersView()		2	Open and close window (confirm exit)	–	Window closes after user confirms	Window closes after user confirms	PASS
Constructor HallOfFamePlayersView()		3	Open and close window (cancel exit)	–	Window stays open after user cancels	Window stays open after user cancels	PASS
setActionListener (ActionListener)		1	Add valid ActionListener	listener that logs clicks	"Return" button triggers listener on click	"Return" button triggers listener on click	PASS
setActionListener (ActionListener)		2	Add ActionListener twice	same listener twice	Listener runs once or twice, but no error/crash	Listener runs once or twice, but no error/crash	PASS

setActionListener (ActionListener)	3	Add null listener	null	No crash, button does not throw error	No crash, button does not throw error	PASS
updateTopPlayersList(String)	1	Display regular list	"Tidus\nAuron\nWakka"	Text area shows each name on a new line	Text area shows each name on a new line	PASS
updateTopPlayersList(String)	2	Display empty list	""	Text area is empty	Text area is empty	PASS
updateTopPlayersList(String)	3	Display null	null	Text area is blank, no crash	Text area is blank, no crash	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Notes
OutlinedLabel (Constructor)	1	Create label with normal text	"Victory!"	Text is set to "Victory!"	Text is set to "Victory!"	PASS
OutlinedLabel (Constructor)	2	Create label with null text	null	Text is set to empty, no exception	Text is set to empty, no exception	PASS
OutlinedLabel (Constructor)	3	Create label with special/unicode text	"★CCPROG3"	Text is set to "★CCPROG3" with unicode	Text is set to "★CCPROG3" with unicode	PASS
getPreferredSize	1	Get preferred size with standard text	"Label with 'Test'"	Dimension larger than string bounds (outline)	Dimension larger than string bounds (outline)	PASS
getPreferredSize	2	Get preferred size when Graphics is null	"Label with 'Edge'"	Returns super.getPreferredSize()	Returns super. getPreferredSize()	PASS
getPreferredSize	3	Get preferred size with long text	"Label with 'This is a long label text'"	Large width, height matches font + outline	Large width, height matches font + outline	PASS
paintComponent	1	Render with default font and color	"Set text 'Hello'"	White fill, black outline, centered	White fill, black outline, centered	PASS
paintComponent	2	Render with changed font and color	"Set font Arial 20pt, text 'Danger', set foreground RED"	Red fill, black outline, Arial font	Red fill, black outline, Arial font	PASS
paintComponent	3	Render with empty string	"Set text ''"	Nothing drawn, no error	Nothing drawn, no error	PASS
paintComponent	4	Render with unicode/multilingual text	"Set text 'テスト★123'"	Correct outline and fill for unicode	Correct outline and fill for unicode	PASS
paintComponent	5	Resize label after text set	"Resize label, set text 'WideTest'"	Text remains centered, outline scales	Text remains centered, outline scales	PASS
paintComponent	6	Render with label size smaller than text	"Set small label size, text 'OverflowTest'"	Text is clipped, outline at edges, no crash	Text is clipped, outline at edges, no crash	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor	1	Instantiate with default constructor	N/A	Window title is set, fields are initialized	Window title is set, fields are initialized	PASS
Constructor	2	Instantiate and check field placeholders	N/A	Fields have correct placeholder text	Fields have correct placeholder text	PASS
Constructor	3	Instantiate and check window size	N/A	Window size is 800x700	Window size is 800x700	PASS
initUI	1	Logo loads and displays	N/A	Logo appears in center panel	Logo appears in center panel	PASS
initUI	2	Background image loads	N/A	Background painted without error	Background painted without error	PASS
initUI	3	Button labels are set	N/A	Register/Return buttons have correct labels	Register/Return buttons have correct labels	PASS
setActionListener	1	Register action listener for Register	Simulate button click	Listener fires on btnRegister	Listener fires on btnRegister	PASS
setActionListener	2	Register action listener for Return	Simulate button click	Listener fires on btnReturn	Listener fires on btnReturn	PASS
setActionListener	3	SetActionListener with null	Pass null as listener	No crash or NPE	No crash or NPE	PASS
resetFields	1	Reset after entering names	player1="A", player2="B"	Both fields are cleared	Both fields are cleared	PASS
resetFields	2	Reset with empty fields	Fields already blank	Fields remain blank	Fields remain blank	PASS
resetFields	3	Reset with long names	player1="LongName1...", player2="LongName2..."	Fields are cleared	Fields are cleared	PASS
getPlayer1Name	1	Get player 1 name after input	"Set field: Alice"	Returns "Alice"	Returns "Alice"	PASS
getPlayer1Name	2	Get player 1 with leading/trailing spaces	" Bob "	Returns "Bob" (trimmed)	Returns "Bob" (trimmed)	PASS
getPlayer1Name	3	Get player 1 with empty field	""	Returns ""	Returns ""	PASS
getPlayer2Name	1	Get player 2 name after input	"Set field: Eve"	Returns "Eve"	Returns "Eve"	PASS
getPlayer2Name	2	Get player 2 with leading/trailing spaces	" Eve2 "	Returns "Eve2" (trimmed)	Returns "Eve2" (trimmed)	PASS
getPlayer2Name	3	Get player 2 with empty field	""	Returns ""	Returns ""	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor	1	Create default RoundedDisplayBox	N/A	Panel is not opaque, layout is BorderLayout	Panel is not opaque, layout is BorderLayout	PASS
Constructor	2	Set size and check preferred layout	N/A	BorderLayout is set as layout manager	BorderLayout is set as layout manager	PASS
Constructor	3	Set opaque to true manually	N/A	Opaque remains false after constructor	Opaque remains false after constructor	PASS
paintComponent	1	Paint component on default size	N/A	Draws a rounded rectangle (30px radius) with semi-transparent black color	Draws a rounded rectangle (30px radius) with semi-transparent black color	PASS
paintComponent	2	Paint component after resize	Resize to (400,200)	Rounded rect fills new size	Rounded rect fills new size	PASS
paintComponent	3	Custom background test	Set background color (should not paint)	Only rounded rectangle is visible	Only rounded rectangle is visible	PASS
paintComponent	4	Check anti-aliasing	N/A	Graphics2D anti-aliasing is enabled	Graphics2D anti-aliasing is enabled	PASS
paintComponent	5	Super call test	N/A	super.paintComponent(g) is called last	super.paintComponent (g) is called last	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor	1	Instantiate MainMenuView	No input	Window titled 'Fatal Fantasy: Tactics | Main Menu' appears	Window titled 'Fatal Fantasy: Tactics | Main Menu' appears	PASS
Constructor	2	Window should be centered	No input	Window location is centered on screen	Window location is centered on screen	PASS
Constructor	3	Window is not resizable	No input	Resizable property is false	Resizable property is false	PASS
Constructor	4	Close window via X button	No input	Window is disposed	Window is disposed	PASS

initUI	1	Logo loads and is centered	No input	Logo displayed and centered at top	Logo displayed and centered at top	PASS
initUI	2	All buttons created	No input	All 6 buttons (Register Players, Manage Characters, Start Battle, Trading Hall, Hall Of Fame, Exit) present	All 6 buttons (Register Players, Manage Characters, Start Battle, Trading Hall, Hall Of Fame, Exit) present	PASS
initUI	3	Background image painted correctly	No input	Background image fills the frame	Background image fills the frame	PASS
setActionListener	1	Register a listener and click Register Players	Click Register Players	Listener is triggered with ACTION_REGISTER_PLAYERS	Listener is triggered with ACTION_REGISTER_PLAYERS	PASS
setActionListener	2	Register listener and click Exit	Click Exit	Listener is triggered with ACTION_EXIT	Listener is triggered with ACTION_EXIT	PASS
setActionListener	3	Register listener, click all buttons	Click each button	Listener called for each action command	Listener called for each action command	PASS
setActionListener	4	Multiple listeners on buttons	Add new listener after first	New listener is triggered	New listener is triggered	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor	1	Instantiate label with simple text	"Victory!"	Label displays 'Victory!' in white text with black outline	Label displays 'Victory!' in white text with black outline	PASS
Constructor	2	Instantiate label with empty text	""	No text shown, outline area is minimal	No text shown, outline area is minimal	PASS
Constructor	3	Instantiate label with long text	"Congratulations, Hero!"	Label displays full string, outline remains correct	Label displays full string, outline remains correct	PASS
getPreferredSize	1	Preferred size for short text	"Win"	Width and height are larger than text bounds (outline included)	Width and height are larger than text bounds (outline included)	PASS
getPreferredSize	2	Preferred size for long text	"VeryLongLabelText"	Width increases accordingly, outline is not clipped	Width increases accordingly, outline is not clipped	PASS
getPreferredSize	3	Null graphics fallback	Call in headless/test context	Super's preferred size is returned	Super's preferred size is returned	PASS
paintComponent	1	Paints with black outline	"Battle Start!"	Text has visible black outline and white fill	Text has visible black outline and white fill	PASS

paintComponent	2	Font change	"Set font to Serif, 36pt"	Outline and fill adjust to new font size	Outline and fill adjust to new font size	PASS
paintComponent	3	Component resize	Resize label to 500x100	Text remains centered and fully outlined	Text remains centered and fully outlined	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor	1	Create view for Player 1	playerID = 1	Title displays "Player 1 Management"	Title displays "Player 1 Management"	PASS
Constructor	2	Create view for Player 2	playerID = 2	Title displays "Player 2 Management"	Title displays "Player 2 Management"	PASS
Constructor	3	Check all buttons exist	N/A	All six buttons are visible and enabled	All six buttons are visible and enabled	PASS
initUI	1	Logo loads for Player 1	playerID = 1	Player 1 logo shown and scaled	Player 1 logo shown and scaled	PASS
initUI	2	Buttons have correct size	N/A	Each button is 250x50 px	Each button is 250x50 px	PASS
initUI	3	Background image renders	N/A	Background image fills frame with scaling	Background image fills frame with scaling	PASS
setActionListener	1	Attach listener to all buttons	Mock ActionListener	Listener invoked on any button click	Listener invoked on any button click	PASS
setActionListener	2	Action command set for "Edit Character"	Click Edit Character	ActionEvent command="Edit Character"	ActionEvent command=" Edit Character"	PASS
setActionListener	3	Action command set for "Inventory"	Click Inventory	ActionEvent command=" Inventory"	ActionEvent command=" Inventory"	PASS
getPlayerID	1	Get player ID after construction	playerID = 2	Returns 2	Returns 2	PASS
getPlayerID	2	Get player ID for edge case	playerID = 0	Returns 0	Returns 0	PASS
getPlayerID	3	Get player ID for negative value	playerID = -1	Returns -1	Returns -1	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor	1	Window initializes with correct title	N/A	Window title is "Fatal Fantasy: Tactics | Player Deletion"	Window title is "Fatal Fantasy: Tactics | Player Deletion"	PASS

	Constructor	2	Window is not resizable	N/A	Window cannot be resized by user	Window cannot be resized by user	PASS
	Constructor	3	Window closes after confirmation	Close window and confirm YES	Window closes	Window closes	PASS
	initUI	1	Player list text area is not editable	N/A	playerListArea is not editable	playerListArea is not editable	PASS
	initUI	2	All buttons present	N/A	Delete and Return buttons are present	Delete and Return buttons are present	PASS
	initUI	3	Dropdown appears	N/A	Player dropdown is visible	Player dropdown is visible	PASS
	setActionListener	1	Listener set for Delete	Mock ActionListener	Listener triggered on Delete click	Listener triggered on Delete click	PASS
	setActionListener	2	Listener set for Return	Mock ActionListener	Listener triggered on Return click	Listener triggered on Return click	PASS
	setActionListener	3	Listener set for dropdown	Mock ActionListener	Listener triggered on dropdown change	Listener triggered on dropdown change	PASS
	updatePlayerList	1	Set player list	updatePlayerList("P1\nP2")	Text area shows "P1\nP2"	Text area shows "P1\nP2"	PASS
	updatePlayerList	2	Clear player list	updatePlayerList("")	Text area is empty	Text area is empty	PASS
	updatePlayerList	3	Update with long list	updatePlayerList ("P1\nP2\nP3\nP4")	Text area shows all names	Text area shows all names	PASS
	confirmPlayerDeletion	1	User confirms deletion	playerName="Alice"	Returns true when user clicks YES	Returns true when user clicks YES	PASS
	confirmPlayerDeletion	2	User cancels deletion	playerName="Bob"	Returns false when user clicks NO	Returns false when user clicks NO	PASS
	confirmPlayerDeletion	3	User closes dialog	playerName="Carol"	Returns false if dialog closed	Returns false if dialog closed	PASS
	setPlayerOptions	1	Add two players	setPlayerOptions(["P1","P2"])	Dropdown has P1 and P2	Dropdown has P1 and P2	PASS
	setPlayerOptions	2	Clear options	setPlayerOptions([])	Dropdown is empty	Dropdown is empty	PASS
	setPlayerOptions	3	Update options	setPlayerOptions(["X","Y","Z"])	Dropdown lists X, Y, Z	Dropdown lists X, Y, Z	PASS
	resetDropdowns	1	Reset selection after choice	setPlayerOptions(["A"]), select "A", resetDropdowns()	Dropdown is unselected	Dropdown is unselected	PASS
	resetDropdowns	2	Reset empty dropdown	setPlayerOptions([]), resetDropdowns()	Dropdown remains unselected	Dropdown remains unselected	PASS

resetDropdowns	3	Reset after multiple options	setPlayerOptions(["A","B"]), select "B", resetDropdowns()	Dropdown is unselected	Dropdown is unselected	PASS
getSelectedPlayer	1	Get selected player after choice	Select "P1"	Returns "P1"	Returns "P1"	PASS
getSelectedPlayer	2	No selection made	No selection	Returns null	Returns null	PASS
getSelectedPlayer	3	Change selection twice	Select "P2", then "P3"	Returns "P3"	Returns "P3"	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
setPlayer1Options	1	Dropdown updates with two options	["Alice","Bob"]	Dropdown1 contains Alice, Bob	Dropdown1 contains Alice, Bob	PASS
setPlayer1Options	2	Dropdown clears previous options before adding new	["Sam"]	Dropdown1 contains only Sam	Dropdown1 contains only Sam	PASS
setPlayer1Options	3	Handles empty options gracefully	[]	Dropdown1 contains no options	Dropdown1 contains no options	PASS
setPlayer2Options	1	Dropdown updates with three options	["Leo","Mia","Ken"]	Dropdown2 contains Leo, Mia, Ken	Dropdown2 contains Leo, Mia, Ken	PASS
setPlayer2Options	2	Dropdown clears previous before adding new	["Zara"]	Dropdown2 contains only Zara	Dropdown2 contains only Zara	PASS
setPlayer2Options	3	Handles empty list	[]	Dropdown2 contains no options	Dropdown2 contains no options	PASS
resetDropdowns	1	Dropdowns set to no selection	Select "X" in dropdown1 & "Y" in dropdown2	No item selected in dropdown1 & dropdown2	No item selected in dropdown1 & dropdown2	PASS
resetDropdowns	2	Works with already unselected	Dropdowns unselected	Dropdowns remain unselected	Dropdowns remain unselected	PASS
resetDropdowns	3	Works after setPlayer1/2Options	Add options then reset	No item selected in both dropdowns	No item selected in both dropdowns	PASS
getSelectedPlayer1	1	Returns selected player	Select "Ryu" in dropdown1	"Ryu"	"Ryu"	PASS
getSelectedPlayer1	2	Returns null if nothing selected	No selection in dropdown1	null	null	PASS
getSelectedPlayer1	3	Returns correct after change	Select "A", then "B" in dropdown1	"B"	"B"	PASS
getSelectedPlayer2	1	Returns selected player	Select "Sage" in dropdown2	"Sage"	"Sage"	PASS
getSelectedPlayer2	2	Returns null if nothing selected	No selection in dropdown2	null	null	PASS

getSelectedPlayer2	3	Returns correct after change	Select "Y", then "Z" in dropdown2	"Z"	"Z"	PASS
setActionListener	1	Fires action when register clicked	Click register	Listener receives event	Listener receives event	PASS
setActionListener	2	Fires action on return	Click return	Listener receives event	Listener receives event	PASS
setActionListener	3	Fires action on dropdown1 selection	Change dropdown1 selection	Listener receives event	Listener receives event	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
setMerchantCharacters	1	Merchant dropdown is populated with character names	["Auron","Lulu"]	Dropdown contains "Auron", "Lulu" and selected index is 0	Dropdown contains "Auron", "Lulu" and selected index is 0	PASS
setMerchantCharacters	2	Merchant dropdown handles empty list	[]	Dropdown is empty, selected index is -1	Dropdown is empty, selected index is -1	PASS
setMerchantCharacters	3	Merchant dropdown replaces previous contents	First ["A"], then ["B","C"]	Dropdown contains "B", "C"	Dropdown contains "B", "C"	PASS
setClientCharacters	1	Client dropdown is populated with character names	["Cloud","Tifa"]	Dropdown contains "Cloud", "Tifa" and selected index is 0	Dropdown contains "Cloud", "Tifa" and selected index is 0	PASS
setClientCharacters	2	Client dropdown handles empty list	[]	Dropdown is empty, selected index is -1	Dropdown is empty, selected index is -1	PASS
setClientCharacters	3	Client dropdown replaces previous contents	First ["Cid"], then ["Vincent","Yuffie"]	Dropdown contains "Vincent", "Yuffie"	Dropdown contains "Vincent", "Yuffie"	PASS
updateMerchantItems	1	Merchant item list is updated	2 items given	Merchant JList shows 2 items	Merchant JList shows 2 items	PASS
updateMerchantItems	2	Merchant item list clears previous items	First 2, then 1 item	Merchant JList only shows new 1 item	Merchant JList only shows new 1 item	PASS
updateMerchantItems	3	Handles empty item list	[]	Merchant JList shows no items	Merchant JList shows no items	PASS
updateClientItems	1	Client item list is updated	3 items given	Client JList shows 3 items	Client JList shows 3 items	PASS
updateClientItems	2	Client item list clears previous items	First 3, then 2 items	Client JList only shows new 2 items	Client JList only shows new 2 items	PASS

	updateClientItems	3	Handles empty item list	[]	Client JList shows no items	Client JList shows no items	PASS
	appendTradeLog	1	Appends text to log	"Test log"	Trade log text area contains "Test log"	Trade log text area contains "Test log"	PASS
	appendTradeLog	2	Appends multiple lines	"A","B"	Log area contains "A" and "B" on separate lines	Log area contains "A" and "B" on separate lines	PASS
	appendTradeLog	3	Handles empty string	""	Trade log unchanged or adds blank line	Trade log unchanged or adds blank line	PASS
	getSelectedMerchantCharacter	1	Returns correct character after dropdown selection	Select index 1 in merchantDropdown	Returns merchantChars.get(1)	Returns merchantChars.get(1)	PASS
	getSelectedMerchantCharacter	2	Returns null when nothing selected	Set merchantDropdown to -1	Returns null	Returns null	PASS
	getSelectedMerchantCharacter	3	Returns first if list populated	Set merchantChars=["Alpha"]	Returns "Alpha"	Returns "Alpha"	PASS
	getSelectedClientCharacter	1	Returns correct character after dropdown selection	Select index 0 in clientDropdown	Returns clientChars.get(0)	Returns clientChars.get(0)	PASS
	getSelectedClientCharacter	2	Returns null if list is empty	clientChars is empty	Returns null	Returns null	PASS
	getSelectedClientCharacter	3	Returns correct when switched	Switch clientDropdown from index 0 to 1	Returns updated clientChar	Returns updated clientChar	PASS
	getSelectedMerchantItems	1	Returns list of selected merchant items	Select 2 items in merchant JList	List contains those 2 items	List contains those 2 items	PASS
	getSelectedMerchantItems	2	Returns empty when none selected	Select none	Empty list	Empty list	PASS
	getSelectedMerchantItems	3	Works after clearing selection	Select item, clear selection	Empty list	Empty list	PASS
	getSelectedClientItems	1	Returns list of selected client items	Select 1 item in client JList	List contains that item	List contains that item	PASS
	getSelectedClientItems	2	Returns empty when none selected	Select none	Empty list	Empty list	PASS
	getSelectedClientItems	3	Works after multiple select/unselect	Select 2, unselect all	Empty list	Empty list	PASS
	showError	1	Displays error dialog with correct message	"Trade not allowed"	Dialog appears with "Trade not allowed"	Dialog appears with "Trade not allowed"	PASS
	showError	2	Can display after previous dialog	"Error2"	Dialog appears with "Error2"	Dialog appears with "Error2"	PASS
	showError	3	Handles empty string	""	Dialog appears with blank or generic error	Dialog appears with blank or generic error	PASS

showInfo	1	Displays info dialog with correct message	"Trade complete"	Dialog appears with "Trade complete"	Dialog appears with "Trade complete"	PASS
showInfo	2	Multiple info dialogs	"A","B"	Each dialog appears in order	Each dialog appears in order	PASS
showInfo	3	Handles empty string	""	Dialog appears with blank or generic info	Dialog appears with blank or generic info	PASS
refresh	1	Updates both JLists after changing selected characters	Select character with new inventory	Both JLists show new items	Both JLists show new items	PASS
refresh	2	Disables trade if characters not selected	Set both dropdowns to -1	Trade button disabled	Trade button disabled	PASS
refresh	3	Resets selections on refresh	Select items, call refresh	Selections cleared	Selections cleared	PASS
resetFields	1	Clears all selections in lists	Select in both lists, call resetFields	Both JLists have no selection	Both JLists have no selection	PASS
resetFields	2	Does not affect dropdowns	Change dropdown, call resetFields	Dropdown selection unchanged	Dropdown selection unchanged	PASS
resetFields	3	Works with no selection active	Nothing selected, call resetFields	No error, nothing changes	No error, nothing changes	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
setMerchantOptions	1	Populates merchant dropdown with player names	["Cloud","Tifa"]	Dropdown1 contains "Cloud", "Tifa" and selected index is 0	Dropdown1 contains "Cloud", "Tifa" and selected index is 0	PASS
setMerchantOptions	2	Handles empty merchant options	[]	Dropdown1 is empty, selected index is -1	Dropdown1 is empty, selected index is -1	PASS
setMerchantOptions	3	Replaces old merchant options	First ["A"], then ["B","C"]	Dropdown1 contains "B","C"	Dropdown1 contains "B"," C"	PASS
setClientOptions	1	Populates client dropdown with player names	["Barret","Aeris"]	Dropdown2 contains "Barret", "Aeris" and selected index is 0	Dropdown2 contains "Barret", "Aeris" and selected index is 0	PASS
setClientOptions	2	Handles empty client options	[]	Dropdown2 is empty, selected index is -1	Dropdown2 is empty, selected index is -1	PASS
setClientOptions	3	Replaces old client options	First ["X"], then ["Y","Z"]	Dropdown2 contains "Y","Z"	Dropdown2 contains "Y"," Z"	PASS

resetDropdowns	1	Resets both dropdown selections after selection	Select both, then call resetDropdowns	Both dropdowns have selectedIndex -1	Both dropdowns have selectedIndex -1	PASS
resetDropdowns	2	No error if already no selection	Nothing selected, call resetDropdowns	Both dropdowns stay -1	Both dropdowns stay -1	PASS
resetDropdowns	3	Works with partially selected	Select only merchant, call resetDropdowns	Both dropdowns set to -1	Both dropdowns set to -1	PASS
getSelectedMerchant	1	Returns selected merchant name	"Cloud" selected in dropdown1	Returns "Cloud"	Returns "Cloud"	PASS
getSelectedMerchant	2	Returns null when nothing selected	No selection in dropdown1	Returns null	Returns null	PASS
getSelectedMerchant	3	Returns updated value after change	Change selection from "A" to "B"	Returns "B"	Returns "B"	PASS
getSelectedClient	1	Returns selected client name	"Barret" selected in dropdown2	Returns "Barret"	Returns "Barret"	PASS
getSelectedClient	2	Returns null when nothing selected	No selection in dropdown2	Returns null	Returns null	PASS
getSelectedClient	3	Returns updated value after change	Change selection from "X" to "Z"	Returns "Z"	Returns "Z"	PASS
setStartTradingEnabled	1	Enables Start Trading button	TRUE	Start Trading button is enabled	Start Trading button is enabled	PASS
setStartTradingEnabled	2	Disables Start Trading button	FALSE	Start Trading button is disabled	Start Trading button is disabled	PASS
setStartTradingEnabled	3	Can toggle button enabled/disabled	true, then false, then true	Button state toggles accordingly	Button state toggles accordingly	PASS
setActionListener	1	Fires listener for Start Trading	Click Start Trading	Listener receives event with START_TRADING	Listener receives event with START_TRADING	PASS
setActionListener	2	Fires listener for Return to Menu	Click Return to Menu	Listener receives event with RETURN_TO_MENU	Listener receives event with RETURN_TO_MENU	PASS
setActionListener	3	Fires listener for dropdown changes	Change dropdown1 or dropdown2	Listener receives action event	Listener receives action event	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
AIController (constructor)	1	Creates AIController with valid AIMoveStrategy	MockStrategy instance	No exception, object created	No exception, object created	PASS
AIController (constructor)	2	Throws GameException if strategy is null	null	Throws GameException	Throws GameException	PASS

AIController (constructor)	3	Strategy field is set correctly	Pass in MockStrategy	"strategy" equals passed instance	"strategy" equals passed instance	PASS
requestMove	1	Returns move from strategy with valid characters	MockStrategy, bot, opponent	MockStrategy.decideMove() result	MockStrategy. decideMove() result	PASS
requestMove	2	Throws GameException if botCharacter is null	null bot, valid opponent	Throws GameException	Throws GameException	PASS
requestMove	3	Throws GameException if opponentCharacter is null	valid bot, null opponent	Throws GameException	Throws GameException	PASS
requestMove	4	Throws GameException if strategy throws exception	Strategy that throws in decideMove()	GameException thrown	GameException thrown	PASS
requestMove	5	Returns correct Move for given characters	MockStrategy returns MockMove	Returns MockMove instance	Returns MockMove instance	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
constructor	1	Controller sets ActionListener on view	Mock view, player, onSelect, onReturn	View has controller as ActionListener	View has controller as ActionListener	PASS
constructor	2	refresh sets character options and details with characters	Player with 2 chars	View shows correct options/details	View shows correct options/details	PASS
constructor	3	refresh sets details to 'No characters' when list empty	Player with 0 chars	Details string is 'No characters available.'	Details string is 'No characters available.'	PASS
actionPerformed	1	Handles SELECT and a character is selected	ActionEvent(SELECT)	onSelect.accept() called with correct char	onSelect.accept() called with correct char	PASS
actionPerformed	2	Handles SELECT and no character selected	ActionEvent(SELECT)	Shows error dialog 'No character selected.'	Shows error dialog 'No character selected.'	PASS
actionPerformed	3	Handles SELECT but character not found	ActionEvent(SELECT)	Shows error dialog 'Character not found.'	Shows error dialog 'Character not found.'	PASS
actionPerformed	4	Handles RETURN command	ActionEvent(RETURN)	View is disposed, onReturn.run() called	View is disposed, onReturn.run() called	PASS
handleSelect	1	Correct character is passed to onSelect	Valid selected name	View disposed, onSelect.accept (char) called	View disposed, onSelect. accept(char) called	PASS
handleSelect	2	No character selected	No selection	Error dialog shown	Error dialog shown	PASS
handleSelect	3	Selected character not found	Nonexistent name	Error dialog shown	Error dialog shown	PASS
handleSelect	4	User cancels in confirmCharacterSelection	User returns false	Nothing happens	Nothing happens	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
getBattleCopy	1	Returns battle copy for originalC1	set originalC1 and battleC1	Returns battleC1	Returns battleC1	PASS
getBattleCopy	2	Returns battle copy for originalC2	set originalC2 and battleC2	Returns battleC2	Returns battleC2	PASS
getBattleCopy	3	Returns null for unknown character	Character not part of battle	Returns null	Returns null	PASS
startBattle	1	Starts battle with both characters alive	Alive c1, Alive c2	BattleView shows battle start, selections cleared	BattleView shows battle start, selections cleared	PASS
startBattle	2	Fails with dead character	Dead c1 or c2	Throws GameException	Throws GameException	PASS
startBattle	3	Correct battle copies created	Valid c1/c2	New objects for battleC1/battleC2	New objects for battleC1/battleC2	PASS
startBattleVsBot	1	Starts AI battle with valid inputs	Human, Bot, AIController	Sets AI fields, disables player2 controls	Sets AI fields, disables player2 controls	PASS
startBattleVsBot	2	Fails with null human	Null human param	Throws GameException	Throws GameException	PASS
startBattleVsBot	3	Fails with null AI controller	Null AI param	Throws GameException	Throws GameException	PASS
submitMove	1	Accepts moves for both characters	Valid character & Move	Both in selections, executes turn	Both in selections, executes turn	PASS
submitMove	2	Throws if character not in battle	Wrong character	Throws GameException	Throws GameException	PASS
submitMove	3	Throws if move/user is null	Null inputs	Throws GameException	Throws GameException	PASS
handlePlayerChoice	1	Handles valid ability choice	Supplied ability name	Calls submitMove with AbilityMove	Calls submitMove with AbilityMove	PASS
handlePlayerChoice	2	Handles valid single-use item	Equipped SingleUseItem	Calls submitMove with ItemMove	Calls submitMove with ItemMove	PASS
handlePlayerChoice	3	Handles invalid choice	Unknown choice	Adds error entry to log	Adds error entry to log	PASS
handlePlayerChoice	4	Handles passive item	Equipped PassiveItem	Adds info entry to log	Adds info entry to log	PASS
useSingleUseItem	1	Uses valid item	Character with item	Removes item, applies effect	Removes item, applies effect	PASS
useSingleUseItem	2	Throws if item not in inventory	Missing item	Throws GameException	Throws GameException	PASS

useSingleUseItem	3	Throws if character not in battle	Wrong character	Throws GameException	Throws GameException	PASS
defend	1	Submits Defend move	Valid character	Defend submitted	Defend submitted	PASS
defend	2	Throws if character not in battle	Wrong character	Throws GameException	Throws GameException	PASS
defend	3	Throws if user null	Null user	Throws GameException	Throws GameException	PASS
recharge	1	Submits Recharge move	Valid character	Recharge submitted	Recharge submitted	PASS
recharge	2	Throws if character not in battle	Wrong character	Throws GameException	Throws GameException	PASS
recharge	3	Throws if user null	Null user	Throws GameException	Throws GameException	PASS
executeTurn	1	Ends battle if one character dies	Character dies mid-turn	BattleView shows battle end	BattleView shows battle end	PASS
executeTurn	2	Processes turn order by priority	Abilities with different priority	Higher priority goes first	Higher priority goes first	PASS
executeTurn	3	Awards XP, winner/loser levels	Winner and loser	Correct XP and level up dialog	Correct XP and level up dialog	PASS
syncInventory	1	Removes used SingleUseItems	Battle copy has less items	Persistent inventory is updated	Persistent inventory is updated	PASS
syncInventory	2	No change for Passive items	Only passive items	Inventory unchanged	Inventory unchanged	PASS
syncInventory	3	Handles empty inventory	No items	No errors	No errors	PASS
updatePlayerPanels	1	Sets player panels correctly	Valid characters	Panels updated	Panels updated	PASS
updatePlayerPanels	2	Throws if no active battle	Battle is null	Throws GameException	Throws GameException	PASS
abilityNames	1	Returns all abilities	Character with 3 abilities	List contains all ability names	List contains all ability names	PASS
abilityNames	2	Appends equipped item	Character with equipped item	Item name in list	Item name in list	PASS
abilityNames	3	Handles no abilities	Character with 0 abilities	Empty or only item in list	Empty or only item in list	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
actionPerformed	1	Handles PvP action with 2 players	PvP button clicked & players.size()==2	Calls handlePvP() and starts PvP setup	Calls handlePvP() and starts PvP setup	PASS

actionPerformed	2	Handles PvB action with 1+ player	PvB button clicked & players.size()==1	Calls handlePvB() and starts PvB setup	Calls handlePvB() and starts PvB setup	PASS
actionPerformed	3	Handles Return action	Return button clicked	Disposes view and calls navigateBackToMainMenu()	Disposes view and calls navigateBackToMainMenu()	PASS
actionPerformed	4	Shows error if less than 2 players for PvP	PvP button clicked & players.size()==1	JOptionPane error shown	JOptionPane error shown	PASS
actionPerformed	5	Shows error if no players for PvB	PvB button clicked & players.size()==0	JOptionPane error shown	JOptionPane error shown	PASS
handlePvP	1	Starts PvP setup with exactly 2 players	players.size()==2	View disposed, BattleSetupController.startPvP() called	View disposed, BattleSetupController.startPvP() called	PASS
handlePvP	2	Shows error for only 1 player	players.size()==1	JOptionPane error shown	JOptionPane error shown	PASS
handlePvP	3	Shows error for 0 players	players.size()==0	JOptionPane error shown	JOptionPane error shown	PASS
handlePvB	1	Starts PvB setup with 1 player	players.size()==1	View disposed, BattleSetupController.startPvB() called	View disposed, BattleSetupController.startPvB() called	PASS
handlePvB	2	Starts PvB setup with multiple players	players.size()==3	View disposed, BattleSetupController.startPvB() called	View disposed, BattleSetupController.startPvB() called	PASS
handle PvB	3	Shows error for 0 players	players.size()==0	JOptionPane error shown	JOptionPane error shown	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
startPvP	1	Starts PvP selection	Players list with 2 players	Calls selectPlayer1ForPvP()	Calls selectPlayer1ForPvP()	PASS
startPvP	2	Handles player 1 character selection	User selects char for player 1	Proceeds to selectPlayer2ForPvP()	Proceeds to selectPlayer2ForPvP()	PASS
startPvP	3	Handles player 2 character selection	User selects char for player 2	Proceeds to launchPvP()	Proceeds to launchPvP()	PASS
startPvB	1	Starts PvB selection	Players list with at least 1 player	Calls selectPlayerForPvB()	Calls selectPlayerForPvB()	PASS
startPvB	2	Handles player 1 character selection	User selects char for player 1	Proceeds to launchPvB()	Proceeds to launchPvB()	PASS

startPvB	3	Handles return to mode select	User cancels char selection	Calls sceneManager. showBattleModes()	Calls sceneManager. showBattleModes()	PASS
launchPvP	1	Launches PvP battle	After both chars are selected	Calls sceneManager. showPlayerVsPlayerBattle() with both players and their chars	Calls sceneManager. showPlayerVsPlayerBattle() with both players and their chars	PASS
launchPvB	1	Launches PvB with valid bot	After char selection and RandomCharacterGenerator works	Calls sceneManager. showPlayerVsBotBattle() with bot and AIController	Calls sceneManager. showPlayerVsBotBattle() with bot and AIController	PASS
launchPvB	2	Handles GameException in bot generation	RandomCharacterGenerator throws exception	Shows error dialog with DialogUtils.showErrorDialog()	Shows error dialog with DialogUtils. showErrorDialog()	PASS
launchPvB	3	Creates new SmartBot for every battle	Multiple calls to launchPvB	Each battle creates unique AIController	Each battle creates unique AIController	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
handleAutoCreateCharacter	1	Create character with valid input	Name filled, race/class/abilities set	Character created, view closed, game saved	Character created, view closed, game saved	PASS
handleAutoCreateCharacter	2	Attempt create without name	Blank name	Show error: "Please enter a character name."	Show error: "Please enter a character name."	PASS
handleAutoCreateCharacter	3	Attempt create before randomize	Race/class/abilities null	Show error: "Please randomize a character first."	Show error: "Please randomize a character first."	PASS
handleAutoCreateCharacter	4	GameException thrown during character creation	Throw GameException in Character constructor	Show error: "Failed to create character: ..."	Show error: "Failed to create character: ..."	PASS
handleRandomize	1	Randomizes new character	Any state	Generated name/race/class/abilities previewed in view	Generated name/race/class/abilitie s previewed in view	PASS
handleRandomize	2	Race is GNOME	Randomly GNOME	Adds extra ability (4th) to abilities	Adds extra ability (4th) to abilities	PASS
handleRandomize	3	No available races	RaceService returns empty list	Show error: "No available races for character generation."	Show error: "No available races for character generation."	PASS
handleReturn	1	Return navigation	Player exists	View closes, navigates to player management	View closes, navigates to player management	PASS

handleReturn	2	Return navigation with unknown player	Player not in controller list	Throws IllegalArgumentException	Throws IllegalArgumentException	PASS
generateRandomName	1	Returns valid name	Called repeatedly	Name is always from NAME_POOL	Name is always from NAME_POOL	PASS
getRandomRace	1	Returns race from available races	Normal pool	Returns race from RaceService. getAvailableRaces()	Returns race from RaceService. getAvailableRaces()	PASS
getRandomRace	2	No races available	Empty RaceService	Throws IllegalStateException	Throws IllegalStateException	PASS
getRandomClass	1	Returns any class type	Called repeatedly	Returns value from ClassType. values()	Returns value from ClassType.values()	PASS
getPlayerByName	1	Finds player (case-insensitive)	Player name in controller	Returns Player instance	Returns Player instance	PASS
getPlayerByName	2	Player not found	Unknown name	Throws IllegalArgumentException	Throws IllegalArgumentException	PASS
formatCharacter	1	Formats correct details	Character with all fields	Returns multi-line string with correct info	Returns multi-line string with correct info	PASS
formatCharacter	2	Handles multiple abilities	Character with 4 abilities	All abilities listed in output	All abilities listed in output	PASS

Method	#	Test Description	Test Input/Scenario	Expected Output	Actual Output	Status
openManualCreationView	1	Opens manual creation view	Call method	View becomes visible	Expected Output	PASS
openManualCreationView	2	Manual view race options set	View opened	View has correct races in dropdown	View becomes visible	PASS
openManualCreationView	3	Manual view class options set	View opened	View has correct classes in dropdown	View has correct races in dropdown	PASS
openCharacterListView	1	Opens character list view	Call method	View is displayed with characters	View has correct classes in dropdown	PASS
openCharacterListView	2	No characters in player	Player has none	Shows "No characters available."	View is displayed with characters	PASS
openCharacterListView	3	Characters present	Player has characters	List displays all character names	Shows "No characters available."	PASS

	openCharacterEditView	1	Edit view opens	Call method	Edit view opens	List displays all character names	PASS
	openCharacterEditView	2	Edit view return	Return action	Edit view closes	Edit view opens	PASS
	openCharacterDeleteView	1	Delete view opens	Call method	Delete view opens	Edit view closes	PASS
	openCharacterDeleteView	2	Delete character confirmed	Character exists & confirmed	Character deleted, info message shown	Delete view opens	PASS
	openCharacterDeleteView	3	Delete character not found	Character doesn't exist	Error message shown	Character deleted, info message shown	PASS
	handleCreateCharacterRequest	1	Valid create request	Valid data	Character added, info message shown	Error message shown	PASS
	handleCreateCharacterRequest	2	Missing name	Blank name	Error message (requireNonBlank)	Character added, info message shown	PASS
	handleCreateCharacterRequest	3	Null race or classType	Null params	Error message (requireNonNull)	Error message (requireNonBlank)	PASS
	getAvailableRaces	1	Returns all races	RaceService has races	Returns correct race list	Error message (requireNonNull)	PASS
	getAvailableRaces	2	No races available	RaceService empty	Returns empty list	Returns correct race list	PASS
	getAvailableRaces	3	RaceService returns custom races	Custom races set	Returns those races	Returns empty list	PASS
	getAvailableAbilities	1	Valid class type	Valid class	Returns correct ability list	Returns those races	PASS
	getAvailableAbilities	2	Null class type	Null input	Throws NullPointerException	Returns correct ability list	PASS
	getAvailableAbilities	3	Class with no abilities	ClassType with none	Returns empty list	Throws NullPointerException	PASS
	getInventoryForCharacter	1	Valid character	Character with items	Returns item list	Returns empty list	PASS
	getInventoryForCharacter	2	Character has no items	Character with empty inventory	Returns empty list	Returns item list	PASS
	getInventoryForCharacter	3	Null character	Null input	Throws NullPointerException	Returns empty list	PASS
	refreshInventoryDisplay	1	Updates view with inventory	Valid char/view	View is updated with items	Throws NullPointerException	PASS
	refreshInventoryDisplay	2	Null character or view	Null input(s)	Throws NullPointerException	View is updated with items	PASS
	refreshInventoryDisplay	3	No items in inventory	Character with no items	View updates with empty	Throws NullPointerException	PASS

bindCharacterManualCreationView	1	Dropdowns populated	Bind method called	Dropdowns show races/classes	View updates with empty	PASS
bindCharacterManualCreationView	2	Ability slots update	Class changed	Ability slots updated	Dropdowns show races/classes	PASS
bindCharacterManualCreationView	3	Create button works	Click with valid input	Character created	Ability slots updated	PASS
bindCharacterListViewingView	1	ActionListener set	Bind called	Actions respond to events	Character created	PASS
bindCharacterListViewingView	2	View/Return actions	VIEW_CHAR/RETURN events	Character list refresh or close	Actions respond to events	PASS
bindCharacterListViewingView	3	No characters	Empty player	Shows "No characters available."	Character list refresh or close	PASS
bindCharacterSpecViewingView	1	Dropdown populated	Bind called	Dropdown has character names	Shows "No characters available."	PASS
bindCharacterSpecViewingView	2	Return action closes	RETURN event	View closes	Dropdown has character names	PASS
bindCharacterSpecViewingView	3	Select character	Select event	Details shown in view	View closes	PASS
					Details shown in view	

Method	#	Test Description	Test Input/Scenario	Expected Output	Actual Output	Status
Constructor	1	Null view	view is null	Throws GameException with message about view	Throws GameException with message about view	PASS
Constructor	2	Null RaceService	valid view, RaceService is null	Throws GameException with message about raceService	Throws GameException with message about raceService	PASS
Constructor	3	No races available	RaceService returns empty list	Throws GameException about races loaded	Throws GameException about races loaded	PASS
Constructor	4	Null saveCallback	All valid except saveCallback null	Throws GameException	Throws GameException	PASS
Constructor	5	All valid	All dependencies valid	Controller created	Controller created	PASS
onRandomize (private, via action)	1	Valid randomization	Randomize button pressed	Preview shown with random char	Preview shown with random char	PASS
onRandomize	2	Class with <3 abilities	ClassService returns 2	Shows error about abilities	Shows error about abilities	PASS

onRandomize	3	Random fails (service issue)	Service throws	Error message displayed	Error message displayed	PASS
onCreate (private, via action)	1	Blank name	User input empty	Shows error about name	Shows error about name	PASS
onCreate	2	User cancels confirmation	Name entered but confirm = false	No character created, fields not reset	No character created, fields not reset	PASS
onCreate	3	Valid creation	User enters valid name and confirms	Character created, info shown, fields reset	Character created, info shown, fields reset	PASS
onCreate	4	Service throws on creation	GameException thrown	Error message displayed	Error message displayed	PASS
generateRandomCharacter (private)	1	Normal	All services valid	Returns Character with random values	Returns Character with random values	PASS
generateRandomCharacterWithName (private)	1	All valid	Services valid	Character with given name created	Character with given name created	PASS
generateRandomCharacterWithName	2	Not enough abilities	ClassService returns <3	Throws GameException	Throws GameException	PASS
generateRandomCharacterWithName	3	Abilities not unique	ClassService returns dupes	Throws GameException	Throws GameException	PASS
format (private)	1	Valid formatting	Character with known data	Correct string output	Correct string output	PASS
wireView (private)	1	Event binding	Events trigger methods	Randomize/Create/Return invoke handlers	Randomize/Create/Return invoke handlers	PASS

Method	# #	Test Description	Test Input/Scenario	Expected Output	Actual Output	Status
Constructor	1	Valid construction	Valid view/playerName/controller	No error, all services and fields initialized	No error, all services and fields initialized	PASS
Constructor	2	Null view	view = null	Throws GameException for null view	Throws GameException for null view	PASS
Constructor	3	Null playerName	playerName = null	Throws GameException for null playerName	Throws GameException for null playerName	PASS
Constructor	4	Null gameManagerController	gameManagerController = null	Throws GameException for null controller	Throws GameException for null controller	PASS
handleCreateCharacter	1	All valid inputs	All fields provided and valid	Character created, attached to player, view closes	Character created, attached to player, view closes	PASS
handleCreateCharacter	2	Blank name	Name is ""	Error message shown, character not created	Error message shown, character not created	PASS

handleCreateCharacter	3	Null race	No race selected	Error message shown	Error message shown	PASS
handleCreateCharacter	4	Null class	No class selected	Error message shown	Error message shown	PASS
handleCreateCharacter	5	Too few abilities	Only 2 selected	Error message	Error message	PASS
handleCreateCharacter	6	Too many abilities	4 selected (not GNOME)	Error message	Error message	PASS
handleCreateCharacter	7	Duplicate abilities	Same ability selected multiple times	Error message	Error message	PASS
handleCreateCharacter	8	Invalid ability for class	Ability not in allowed list	Error message	Error message	PASS
handleCreateCharacter	9	GNOME with 4 distinct abilities	Valid GNOME race, All 4 unique	Character created	Character created	PASS
handleCreateCharacter	10	GNOME with duplicate in 4 slots	Duplicate	Error message	Error message	PASS
handleCreateCharacter	11	Ability4 for non-GNOME	4th slot filled but not GNOME	Error message	Error message	PASS
handleClassSelection	1	Class selected	Class dropdown changed	Ability dropdowns populated accordingly	Ability dropdowns populated accordingly	PASS
handleRaceSelection	1	GNOME selected	Race dropdown set to GNOME, class selected	Ability4 visible	Ability4 visible	PASS
handleRaceSelection	2	Non-GNOME race selected	Race dropdown not GNOME, class selected	Ability4 hidden	Ability4 hidden	PASS
refreshAbilityOptions	1	Class selected	Class dropdown set	Ability options populated	Ability options populated	PASS
refreshAbilityOptions	2	Class not selected	No class chosen	All dropdowns empty	All dropdowns empty	PASS
getPlayerByName	1	Existing player name	Correct Player returned	Returns Player object named Alice	Returns Player object name	PASS
getPlayerByName	2	Name not in list	Throws IllegalArgumentException	Throws IllegalArgumentException	Throws IllegalArgumentExc	PASS

Method	#	Test Description	Test Input/Scenario	Expected Output	Actual Output	Status
Constructor	1	Valid construction	View, non-empty player list, valid sceneManager	No errors, labels set to player names	No errors, labels set to player names	PASS

Constructor	2	Empty player list	View, empty list, valid sceneManager	Info dialog "No players registered", both labels null	Info dialog "No players registered", both labels null	PASS
bind	1	MANAGE_PLAYER1 pressed	Button fires event for MANAGE_PLAYER1	openPlayerView(0) called, sceneManager. showPlayerCharacterManagement(p1) called	openPlayerView(0) called, sceneManager. showPlayerCharacterManagement(p1) called	PASS
bind	2	MANAGE_PLAYER2 pressed	Button fires event for MANAGE_PLAYER2	openPlayerView(1) called, sceneManager. showPlayerCharacterManagement(p2) called	openPlayerView(1) called, sceneManager. showPlayerCharacterManagement(p2) called	PASS
bind	3	RETURN_TO_MENU pressed	Button fires event for RETURN_TO_MENU	view disposed, sceneManager. showMainMenu() called	view disposed, sceneManager. showMainMenu() called	PASS
openPlayerView	1	Index in range	idx=0 or 1 (players present)	sceneManager. showPlayerCharacterManagement called with correct Player	sceneManager. showPlayerCharacterManagement called with correct Player	PASS
openPlayerView	2	Index out of range	idx=-1 or idx>=players.size()	No action (no exception, no UI change)	No action (no exception, no UI change)	PASS
refresh	1	Called after player add/remove	refresh called	updateLabels called, labels updated	updateLabels called, labels updated	PASS
updateLabels	1	No players	players.isEmpty()	Info dialog shown, both player labels set null	Info dialog shown, both player labels set null	PASS
updateLabels	2	One player	players.size()==1	player1 label set, player2 label null	player1 label set, player2 label null	PASS
updateLabels	3	Two players	players.size()==2	player1 and player2 labels set	player1 and player2 labels set	PASS

Method	#	Test Description	Test Input/Scenario	Expected Output	Actual Output	Status
Constructor	1	Load game with existing data	Valid sceneManager, hallOfFameController, mainMenuView	Players loaded from SaveLoadService	Players loaded from SaveLoadService	PASS
Constructor	2	Null arguments provided	Any argument null	Throws GameException	Throws GameException	PASS

	actionPerformed	1	Register players action	Command=ACTION_REGISTER_PLAYERS	sceneManager. showPlayerRegistration called, mainMenuView disposed	sceneManager. showPlayerRegistration called, mainMenuView disposed	PASS
	actionPerformed	2	Manage characters	Command=ACTION_MANAGE_CHARACTERS, players empty	Error dialog "Please register players first"	Error dialog "Please register players first"	PASS
	actionPerformed	3	Manage characters	Command=ACTION_MANAGE_CHARACTERS, players not empty	sceneManager. showCharacterManagementMenu called, mainMenuView disposed	sceneManager. showCharacterManagementMenu called, mainMenuView disposed	PASS
	actionPerformed	4	Trading hall	Command=ACTION_TRADING_HALL, players empty	Error dialog	Error dialog	PASS
	actionPerformed	5	Trading hall	Command=ACTION_TRADING_HALL, players present	sceneManager.showTradingHall called, mainMenuView disposed	sceneManager. showTradingHall called, mainMenuView disposed	PASS
	actionPerformed	6	Start battle	Command=ACTION_START_BATTLE, players empty	Error dialog	Error dialog	PASS
	actionPerformed	7	Start battle	Command=ACTION_START_BATTLE, players present	sceneManager. showBattleModes called, mainMenuView disposed	sceneManager. showBattleModes called, mainMenuView disposed	PASS
	actionPerformed	8	Unknown command	Command="UNKNOWN"	Warning dialog shown	Warning dialog shown	PASS
	actionPerformed		Exit	Command=ACTION_EXIT	Application shuts down (calls quitApplication)	Application shuts down (calls quitApplication)	PASS
	handleRegisterPlayers	1	Same player names	player1Name==player2Name	Returns false, shows error dialog	Returns false, shows error dialog	PASS
	handleRegisterPlayers	2	Existing player names conflict	Name exists in SaveLoadService	Returns false, shows error dialog	Returns false, shows error dialog	PASS
	handleRegisterPlayers	3	New unique names	Unique names	Players added, gameData updated, returns true	Players added, gameData updated, returns true	PASS
	handleRegisterSavedPlayers	1	Null or blank names	Any name null or blank	Error dialog, returns false	Error dialog, returns false	PASS
	handleRegisterSavedPlayers	2	Same name for both	Same string	Error dialog, returns false	Error dialog, returns false	PASS
	handleRegisterSavedPlayers	3	Players not found	Names not in save data	Error dialog, returns false	Error dialog, returns false	PASS
	handleRegisterSavedPlayers	4	Players found	Names exist in save data	Players loaded into session, returns true	Players loaded into session, returns true	PASS

deletePlayerByName	1	Delete existing	Name matches a player	Player removed, no error	Player removed, no error	PASS
deletePlayerByName	2	Delete non-existent	Name not found	Throws GameException	Throws GameException	PASS
handlePlayerWin	1	Winner is Bot	"Bot" player name	Hall of Fame not updated	Hall of Fame not updated	PASS
handlePlayerWin	2	Regular player wins	Valid winner/character	Win increments, Hall of Fame updates, possible item reward, game saved	Win increments, Hall of Fame updates, possible item reward, game saved	PASS
getPlayers	1	Get after load or register	Players list requested	Unmodifiable list of current players returned	Unmodifiable list of current players returned	PASS
getGameData	1	Get snapshot	No special scenario	Returns new GameData containing players and hall of fame	Returns new GameData containing players and hall of fame	PASS
handleSaveGameRequest	1	Save invoked	Any state	GameData saved via SaveLoadService	GameData saved via SaveLoadService	PASS
handleLoad game request	1	Load invoked	GameData present	Players and Hall of Fame updated	Players and Hall of Fame updated	PASS
formatAwardMessage	1	Single-use item	MagicItem is SingleUse	Effect description matches effect type	Effect description matches effect type	PASS
formatAwardMessage	2	Passive item	MagicItem is Passive	Description field shown	Description field shown	PASS
generateUniqueReward	1	All rewards owned	10+ attempts	Returns any item, does not crash	Returns any item, does not crash	PASS
quitApplication	1	Quit called	Any state	All frames disposed, Main. shutdown called	All frames disposed, Main.shutdown called	PASS

Method	#	Test Description	Test Input/Scenario	Expected Output	Actual Output	Status
Constructor	1	Normal load	Valid view/sceneManager, player & character entries loaded from SaveLoadService	Valid view and sceneManager initia	Valid view and sceneMana	PASS
Constructor	2	HallOfFame file missing or error	Throws GameException	Entries initialized empty, error dialog shown	Entries initialized empty, error dialog shown	PASS
addWinForPlayer	1	Add win to existing player	Player name matches entry	Wins incremented	Wins incremented	PASS
addWinForPlayer	2	Add win to new player	Player not in list	New HallOfFameEntry created	New HallOfFameEntry created	PASS

	addWinForPlayer	3	Null player	Null argument	Throws GameException	Throws GameException	PASS
	addWinForCharacter	1	Add win to existing character	Character name matches entry	Wins incremented, XP updated	Wins incremented, XP updated	PASS
	addWinForCharacter	2	Add win to new character	Character not in list	New HallOfFameEntry created with correct XP	New HallOfFameEntry created with correct XP	PASS
	addWinForCharacter	3	Null character	Null argument	Throws GameException	Throws GameException	PASS
	getTopPlayersByWins	1	Get top N players	N=3	Returns up to 3, ranked by wins (ties by XP)	Returns up to 3, ranked by wins (ties by XP)	PASS
	getTopPlayersByWins	2	Zero players	N=0	Returns empty list	Returns empty list	PASS
	getTopPlayersByWins	3	Negative N	N<0	Throws GameException	Throws GameException	PASS
	getTopCharactersByWins	1	Get top N characters	N=5	Returns up to 5, ranked by wins (ties by XP)	Returns up to 5, ranked by wins (ties by XP)	PASS
	getTopCharactersByWins	2	Zero characters	N=0	Returns empty list	Returns empty list	PASS
	getTopCharactersByWins	3	Negative N	N<0	Throws GameException	Throws GameException	PASS
	getHallOfFame	1	Defensive copy	Modify return list	Original entries unchanged	Original entries unchanged	PASS
	getHallOfFameCharacters	1	Defensive copy	Modify return list	Original entries unchanged	Original entries unchanged	PASS
	setHallOfFame	1	Set new list	Valid entries	List updated, persisted	List updated, persisted	PASS
	setHallOfFame	2	Null list	Null argument	Throws GameException	Throws GameException	PASS
	persistHallOfFame	1	Normal save	Any state	SaveLoadService. saveHallOfFame called	SaveLoadService. saveHallOfFame called	PASS
	bindHallOfFameCharactersView	1	Empty entries	No character entries	View shows "No records yet!"	View shows "No records yet!"	PASS
	bindHallOfFameCharactersView	2	Multiple entries	Entries exist	View lists top 10, formatted	View lists top 10, formatted	PASS
	bindHallOfFamePlayersView	1	Empty entries	No player entries	View shows "No records yet!"	View shows "No records yet!"	PASS
	bindHallOfFamePlayersView	2	Multiple entries	Entries exist	View lists top 10, formatted	View lists top 10, formatted	PASS
	actionPerformed	1	Show top players	SHOW_TOP_PLAYERS	showTopPlayers() called	showTopPlayers() called	PASS

actionPerformed	2	Show top characters	SHOW_TOP_CHARACTERS	showTopCharacters() called	showTopCharacters() called	PASS
actionPerformed	3	Return to main menu	RETURN	sceneManager.showMainMenu() called	sceneManager. showMainMenu() called	PASS
actionPerformed	4	Unknown action	Other string	Error shown via view. showErrorMessage	Error shown via view. showErrorMessage	PASS

Method	#	Test Description	Test Input/Scenario	Expected Output	Actual Output	Status
Constructor	1	Normal construction	Valid Character, GameManagerController,Fields initialized	Fields initialized	Fields initialized	PASS
Constructor	2	Null character	Null argument	Throws GameException	Throws GameException	PASS
Constructor	3	Null GameManagerController	Null argument	Throws GameException	Throws GameException	PASS
Constructor	4	With view	Valid view provided	View is bound and inventory is shown	View is bound and inventory is shown	PASS
setView	1	Valid view	View passed	View set and shows inventory	View set and shows inventory	PASS
setView	2	Null view	Null argument	Throws GameException	Throws GameException	PASS
getInventoryForCharacter	1	Non-empty inventory	Character has items	Returns unmodifiable list of all items	Returns unmodifiable list of all items	PASS
getInventoryForCharacter	2	Empty inventory	Character has no items	Returns empty list	Returns empty list	PASS
refreshInventoryDisplay	1	Normal call	Any state	View updated with latest inventory and equipped item	View updated with latest inventory and equipped item	PASS
handleEquipItem	1	Equip valid item	Valid MagicItem	Item is equipped, inventory and view updated, persisted	Item is equipped, inventory and view updated, persisted	PASS
handleEquipItem	2	Equip null item	Null argument	Error message shown	Error message shown	PASS
handleEquipItem	3	Equip when another is already equipped	Valid MagicItem replaces old	Old unequipped, new equipped	Old unequipped, new equipped	PASS
handleUnequipItem	1	Item equipped	Item equipped	Item unequipped, view and persistence updated	Item unequipped, view and persistence updated	PASS

handleUnequipItem	2	No item equipped	None equipped	No error, no crash	No error, no crash	PASS
persist	1	Normal	Called	gameManagerController. handleSaveGameRequest() called	gameManagerController. handleSaveGameRequest() called	PASS
actionPerformed	1	Equip button pressed	EQUIP	Selected item equipped and view updated	Selected item equipped and view updated	PASS
actionPerformed	2	Unequip button pressed	UNEQUIP	Item unequipped and view updated	Item unequipped and view updated	PASS
actionPerformed	3	Return button pressed	RETURN	View disposed	View disposed	PASS
actionPerformed	4	View item button pressed	VIEW_ITEM with item selected	Details dialog shown for selected item	Details dialog shown for selected item	PASS
actionPerformed	5	View item button pressed	VIEW_ITEM with no item selected	Info message "Select an item to view." shown	Info message "Select an item to view." shown	PASS
actionPerformed	6	Back button pressed	BACK	Inventory list is shown	Inventory list is shown	PASS

Method	#	Test Description	Test Input/Scenario	Expected Output	Actual Output	Status
Constructor	1	Valid construction	Valid View, Player, GameManagerController,Fields initialized	Fields initialized	Fields initialized	PASS
Constructor	2	Null view	Null View arg	Throws NullPointerException or fails gracefully	Throws NullPointerException or fails gracefully	PASS
Constructor	3	Null player	Null Player arg	Throws NullPointerException or fails gracefully	Throws NullPointerException or fails gracefully	PASS
Constructor	4	Null GameManagerController	Null GameManagerController arg	Throws NullPointerException or fails gracefully	Throws NullPointerException or fails gracefully	PASS
bind	1	All menu buttons wired	N/A	All menu actions trigger correct methods	All menu actions trigger correct methods	PASS
openCharacterList	1	Player with characters	Characters listed	View shows list	View shows list	PASS
openCharacterList	2	Player with no characters	No characters message	View shows "No characters available."	View shows "No characters available."	PASS

	openCharacterList	3	Return pressed	User returns	View disposed	View disposed	PASS
	openCharacterList	4	View character pressed	Shows spec view	Spec view opens	Spec view opens	PASS
	refreshCharacterList	1	Non-empty	Player with chars	Correct details in list	Correct details in list	PASS
	refreshCharacterList	2	Empty	Player with no chars	"No characters available."	"No characters available."	PASS
	openCharacterSpecView	1	No characters	No selection enabled	Selection disabled	Selection disabled	PASS
	openCharacterSpecView	2	Characters exist	Selection enabled	Selection enabled	Selection enabled	PASS
	openCharacterSpecView	3	Character selected	Shows correct details	Details correct	Details correct	PASS
	openEditCharacter	1	No characters	Dropdown empty	Dropdown empty	Dropdown empty	PASS
	openEditCharacter	2	Valid edit	Select & edit abilities	Updates abilities, saves and closes	Updates abilities, saves and closes	PASS
	openEditCharacter	3	Invalid ability selection	Duplicates or invalid class	Error message shown	Error message shown	PASS
	openEditCharacter	4	Invalid magic item	Equip item not in inventory	No equip, no crash	No equip, no crash	PASS
	handleEditConfirmation	1	Valid confirmation	Valid abilities/items	Character updated	Character updated	PASS
	handleEditConfirmation	2	No selection	No character selected	Error shown	Error shown	PASS
	handleEditConfirmation	3	User cancels edit	Cancel in confirm dialog	No update, view stays	No update, view stays	PASS
	handleEditConfirmation	4	Duplicate abilities	Same ability twice	Error message shown	Error message shown	PASS
	handleEditConfirmation	5	Incorrect ability count	Less/more than allowed	Error message shown	Error message shown	PASS
	handleEditConfirmation	6	Invalid for class	Ability not part of class	Error message shown	Error message shown	PASS
	openDeleteCharacter	1	Delete existing	User confirms deletion	Character removed, saved	Character removed, saved	PASS
	openDeleteCharacter	2	No character selected	User attempts to delete	Error message	Error message	PASS
	openDeleteCharacter	3	Cancel deletion	User cancels dialog	No removal	No removal	PASS
	openDeleteCharacter	4	Return pressed	Return pressed	View disposed	View disposed	PASS

openDeleteCharacter	5	Delete not found	Name does not exist	Error shown	Error shown	PASS
openInventory	1	No characters	Player has no characters	Error dialog	Error dialog	PASS
	2	Valid open	Character selected	Inventory view opens	Inventory view opens	PASS
openInventory	3	Invalid character	Name does not exist	No crash, no view	No crash, no view	PASS
	4	User cancels selection	Cancel dialog	No view opens	No view opens	PASS

Method	# #	Test Description	Test Input/Scenario	Expected Output	Actual Output	Status
Constructor	1	Normal construction	All non-null arguments	Fields initialized, actionListener set, dropdown/list populated	Fields initialized, actionListener set, dropdown/list populated	PASS
Constructor	2	Null view	Null View arg	NPE or fail gracefully	NPE or fail gracefully	PASS
Constructor	3	Null gameManager	Null GameManagerController	NPE or fail gracefully	NPE or fail gracefully	PASS
Constructor	4	Null sceneManager	Null SceneManager	NPE or fail gracefully	NPE or fail gracefully	PASS
refresh	1	Players exist	Non-empty player list	Dropdown/list show all names	Dropdown/list show all names	PASS
refresh	2	No players	Empty player list	Dropdown empty, list empty or no crash	Dropdown empty, list empty or no crash	PASS
actionPerformed (RETURN)	1	Return pressed	RETURN command	View disposed, sceneManager. showPlayerRegistration() called	View disposed, sceneManager. showPlayerRegistration() called	PASS
actionPerformed (DELETE)	1	Delete with selection/confirmation	Valid player selected, user confirms	Player deleted, saved, message shown	Player deleted, saved, message shown	PASS
actionPerformed (DELETE)	2	Delete with no selection	No player selected	Error dialog shown, no delete	Error dialog shown, no delete	PASS
actionPerformed (DELETE)	3	Delete but user cancels confirmation	Player selected, user cancels	No delete	No delete	PASS
actionPerformed (DELETE)	4	Delete not found	Selected player not in manager	Error dialog with GameException	Error dialog with GameException	PASS

actionPerformed (DELETE)	5	Delete when SaveLoadService fails	Simulate GameException on save	Error dialog shown	Error dialog shown	PASS
refresh	3	After delete	Player removed from options/list	List updates to not show deleted player	List updates to not show deleted player	PASS

Method	#	Test Description	Test Input/Scenario	Expected Output	Actual Output	Status
Constructor	1	Startup initializes main menu	App launched	rror dialog shown; battle view dispc	rror dialog shown; battle vie	PASS
showMainMenu	1	Normal invocation	Main menu not yet shown	MainMenuView visible, size 800x700	MainMenuView visible, size 800x700	PASS
showMainMenu	2	Already initialized	MainMenuView already present	MainMenuView reused (no reinit)	MainMenuView reused (no reinit)	PASS
showPlayerRegistration	1	First show	PlayerRegistrationView not yet shown	New view added, visible	New view added, visible	PASS
showPlayerRegistration	2	Already exists	PlayerRegistrationView already cached	Reused, not re-created	Reused, not re-created	PASS
showNewPlayersRegistration	1	Valid names entered	p1/p2 non-empty	players registered, main menu shown, success dialog	players registered, main menu shown, success dialog	PASS
showNewPlayersRegistration	2	One/both names empty	p1 or p2 empty	error dialog shown, no registration	error dialog shown, no registration	PASS
showNewPlayersRegistration	3	Duplicate/invalid names	handleRegisterPlayers returns false	fields reset, no registration	fields reset, no registration	PASS
showSavedPlayersRegistration	1	Valid selection	both n1/n2 non-null	players loaded, main menu shown	players loaded, main menu shown	PASS
showSavedPlayersRegistration	2	Null/empty selection	one or both null	error dialog, no navigation	error dialog, no navigation	PASS
showSavedPlayersRegistration	3	GameData load error	GameException thrown	error dialog, cards not changed	error dialog, cards not changed	PASS
showPlayerDelete	1	First call	PlayerDeleteView null	creates new view, controller set, visible	creates new view, controller set, visible	PASS
showPlayer delete	2	View already exists	uses refresh	list updated, view visible	list updated, view visible	PASS
showHallOfFameManagement	1	First call	View/controller null	both created, view visible	both created, view visible	PASS
showHallOfFameManagement	2	View/controller exists	no reinit	view visible	view visible	PASS

showTradingHall	1	First call	View/controller null	new ones created, visible	new ones created, visible	PASS
showTradingHall	2	Already exists	controller.refresh called	view visible	view visible	PASS
showTradeView	1	Normal trade	Valid players	tradeView opened, TradingHallView disposed	tradeView opened, TradingHallView disposed	PASS
showTradeView	2	Controller error	GameException on TradeController	error dialog, tradeView not shown	error dialog, tradeView not shown	PASS
showCharacterManagementMenu	1	First call	View null	controller set, view visible	controller set, view visible	PASS
showCharacterManagementMenu	2	Controller exists	controller.refresh called	view visible	view visible	PASS
showPlayerCharacterManagement	1	Show for player	valid player	view created, controller set, visible	view created, controller set, visible	PASS
showCharacterManagement	1	Show for player	valid player	view created, controller set, visible	view created, controller set, visible	PASS
showBattleModes	1	Normal	valid players	BattleModesView created, controller set, visible	BattleModesView created, controller set, visible	PASS
showPlayerVsBotBattle	1	Valid	all args valid	BattleView created, controller starts bot battle, controls set	BattleView created, controller starts bot battle, controls set	PASS
showPlayerVsBotBattle	2	GameException from BattleController	error dialog, view disposed	Error dialog shown; battle view dis	Error dialog shown; battle v	PASS
showPlayerVsPlayerBattle	1	Normal	valid p1,c1,p2,c2	BattleView created, controller starts battle, controls set	BattleView created, controller starts battle, controls set	PASS
showPlayerVsPlayerBattle	2	GameException from BattleController	error dialog, view disposed	Error dialog shown; battle view disp	Error dialog shown; battle v	PASS
main/start	1	Entry point	Calls showMainMenu()	main menu visible	main menu visible	PASS

Method	@Test	Test Description	Test Input	Expected Output	Actual Output	Status
getEligiblePlayers	1	"Only human players are included"	players=[Alice, Bot, Bob]	[Alice, Bob]	[Alice, Bob]	PASS

getEligiblePlayers	2	"No eligible players if all are bots"	players=[Bot, bot]	[],	[],	PASS
getEligiblePlayers	3	"Empty players list returns empty"	players=[],	[],	[],	PASS
getCharactersForPlayer	1	"Returns player's characters"	player=Alice(has X, Y)	[X, Y]	[X, Y]	PASS
getCharactersForPlayer	2	"Returns empty if player has none"	player=Alice()	[],	[],	PASS
getCharactersForPlayer	3	"Throws on null input"	player=null	GameException	GameException	PASS
getInventory	1	"Returns all items for character"	char=X(items=[A,B])	[A,B]	[A,B]	PASS
getInventory	2	"Returns empty for for char with no items"	char=Y(items=[])	[],	[],	PASS
getInventory	3	"Throws on null input"	char=null	GameException	GameException	PASS
executeTrade (single)	1	"Swaps one item each way"	X:has A, Y:has B, A↔B	A->Y, B->X		PASS
executeTrade (single)	2	"Cannot trade with self"	X, A, X, B	GameException	GameException	PASS
executeTrade (single)	3	"Throws if item not owned"	X:has A, Y:has B, X gives C	GameException	GameException	PASS
executeTrade (multi)	1	"Swaps multiple items"	X:has [A,B], Y:has [C], [A,B]↔[C]	X:has C, Y:has A,B		PASS
executeTrade (multi)	2	"Throws if both give none"	X,Y, [], [],	GameException	GameException	PASS
executeTrade (multi)	3	"Throws if bot involved"	X (player:Bot), Y, [A], [B]	GameException	GameException	PASS
isBot	1	"Detects exact 'Bot'"	p=Bot	TRUE	TRUE	PASS
isBot	2	"Detects 'bot' lowercase"	p=bot	TRUE	TRUE	PASS
isBot	3	"Returns false for non-bot"	p=Alice	FALSE	FALSE	PASS
findPlayerForCharacter	1	"Returns player that owns character"	players=[P:has C], C	P	P	PASS
findPlayerForCharacter	2	"Throws if char not found"	players=[P:has X], Y	GameException	GameException	PASS
findPlayerForCharacter	3	"Works if char in second player"	players=[P1:has X, P2:has Y], Y	P2		PASS

Method	@Test	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor	1	"Initializes controller with valid inputs"	view!=null, players!=null, sceneManager!=null	No Exception	No Exception	PASS
Constructor	2	"Throws on null view"	view=null	GameException	GameException	PASS
Constructor	3	"Throws on null players"	players=null	GameException	GameException	PASS
refresh	1	"Updates dropdowns, no bots"	players=[Alice,Bob]	merchantOptions=[Alice,Bob]	merchantOptions=[Alice, Bob]	PASS
refresh	2	"Bots are not in options"	players=[Alice,Bot,Bob]	merchantOptions=[Alice,Bob]	merchantOptions=[Alice, Bob]	PASS
refresh	3	"Handles empty list"	players=[],	merchantOptions=[],	merchantOptions=[],	PASS
validateSelections	1	"Enable when two different players selected"	merchant=Alice, client=Bob	setStartTradingEnabled(true)	setStartTradingEnabled (true)	PASS
validateSelections	2	"Disable when both selections same"	merchant=Alice, client=Alice	setStartTradingEnabled(false)	setStartTradingEnabled (false)	PASS
validateSelections	3	"Disable when any selection is null"	merchant=null, client=Bob	setStartTradingEnabled(false)	setStartTradingEnabled (false)	PASS
handleStartTrading	1	"Valid selections, starts trading"	mName=Alice, cName=Bob, Alice.hasChar, Bob.hasChar	calls showTradeView	calls showTradeView	PASS
handleStartTrading	2	"Error when names same"	mName=Alice, cName=Alice	show error dialog	show error dialog	PASS
handleStartTrading	3	"Error when player has no chars"	mName=Alice(no chars), cName=Bob(has char)	show error dialog	show error dialog	PASS
findPlayerByName	1	"Finds player by name"	name="Alice", players=[Alice,Bob]	Alice	Alice	PASS
findPlayerByName	2	"Finds player by name (case-insensitive)"	name="alice", players=[Alice,Bob]	Alice	Alice	PASS
findPlayerByName	3	"Throws if player not found"	name="Charlie", players=[Alice, Bob]	GameException	GameException	PASS
						PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
addEntry	1	Add valid entry	entry = "Player1 attacks"	Entry added to log	Entry added to log	PASS

addEntry	2	Add blank entry	entry = " "	GameException thrown	GameException thrown	PASS
addEntry	3	Add null entry	entry = null	GameException thrown	GameException thrown	PASS
getLogEntries	1	Return entries after adding two	addEntry("A"), addEntry("B")	["A", "B"]	["A", "B"]	PASS
getLogEntries	2	Return empty after clearLog	addEntry("X"), clearLog(), getLogEntries()	[],	[],	PASS
getLogEntries	3	Immutability test	getLogEntries(), try to add ("should fail")	UnsupportedOperationException thrown	UnsupportedOperationException thrown	PASS
clearLog	1	Clears entries after several adds	addEntry("1"), addEntry("2"), clearLog(), getLogEntries()	[],	[],	PASS
clearLog	2	Clears empty log	clearLog() on new CombatLog	No exception	No exception	PASS
clearLog	3	Add after clear	addEntry("X"), clearLog(), addEntry("Y")	["Y"]	["Y"]	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
execute	1	Valid DAMAGE ability with enough EP	user=char(50hp,20ep), target=char(40hp,15ep), ability=DAMAGE(10),log	Target loses 10hp, log records action	Target loses 10hp, log records action	PASS
execute	2	Not enough EP for ability	user=char(50hp,1ep),target=char (40hp,15ep),ability=DAMAGE(10, epCost=5),log	GameException: not enough EP	GameException: not enough EP	PASS
execute	3	HEAL ability applied	user=char(20hp/50),target=char (40hp),ability=HEAL(15),log	User gains 15hp, log records healing	User gains 15hp, log records healing	PASS
getName	1	Returns correct name	ability.name="Fireball",,	"Fireball"	"Fireball"	PASS
getName	2	Returns name for custom ability	ability.name="Summon Wolf",,	"Summon Wolf"	"Summon Wolf"	PASS
getName	3	Returns name for utility ability	ability.name="Magic Shield",,	"Magic Shield"	"Magic Shield"	PASS
getDescription	1	Returns correct description	ability.desc="Deals fire damage",,	"Deals fire damage"	"Deals fire damage"	PASS
getDescription	2	Returns empty description	ability.desc="",,	""	""	PASS
getDescription	3	Returns long description	ability.desc="A very powerful spell...",,	"A very powerful spell..."	"A very powerful spell..."	PASS

getEpCost	1	Returns correct EP cost	ability.epCost=5,,	5	5	PASS
getEpCost	2	Returns zero EP cost	ability.epCost=0,,	0	0	PASS
getEpCost	3	Returns negative EP cost (edge)	ability.epCost=-1,,	-1	-1	PASS
getAbility	1	Returns correct ability object	ability=Heal(10),,	ability object returned	ability object returned	PASS
getAbility	2	Returns correct ability object for status	ability=Poison,,	ability object returned	ability object returned	PASS
getAbility	3	Returns correct ability object for custom	ability=Magic Boost,,	ability object returned	ability object returned	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
Battle (constructor)	1	Create battle with two distinct characters	combatant1=Alice, combatant2=Bob	Instance created, log entry added	Instance created, log entry added	PASS
Battle (constructor)	2	Null first combatant	combatant1=null, combatant2=Bob	GameException thrown	GameException thrown	PASS
Battle (constructor)	3	Same object for both combatants	combatant1=Alice, combatant2=Alice	GameException thrown	GameException thrown	PASS
getCombatants	1	Returns both combatants in order	Alice, Bob	[Alice, Bob]	[Alice, Bob]	PASS
getCombatants	2	Immutability test	try to modify returned list	UnsupportedOperationException thrown	UnsupportedOperationException thrown	PASS
getCombatants	3	No extra combatants	Check size == 2	2	2	PASS
getCharacter1	1	Returns first combatant	Alice, Bob	Alice	Alice	PASS
getCharacter1	2	Check type	combatant1=Alice,instanceof Character	TRUE	TRUE	PASS
getCharacter1	3	First is not second	combatant1=Alice, combatant2=Bob	character1 != character2	TRUE	PASS
getCharacter2	1	Returns second combatant	Alice, Bob	Bob	Bob	PASS
getCharacter2	2	Check type	combatant2=Bob,instanceof Character	TRUE	TRUE	PASS
getCharacter2	3	Second is not first	combatant1=Alice, combatant2=Bob	character2 != character1	TRUE	PASS

getRoundNumber	1	Initial round is 1		1	1	PASS
getRoundNumber	2	After one round increment	call nextRound()	2	2	PASS
getRoundNumber	3	After two increments	call nextRound() twice	3	3	PASS
isFinished	1	Initial state is not finished		FALSE	FALSE	PASS
isFinished	2	Set finished true	call setFinished(true)	TRUE	TRUE	PASS
isFinished	3	Set finished then false	call setFinished(false)	FALSE	FALSE	PASS
getCombatLog	1	Returns CombatLog instance	constructor called	instance of CombatLog	instance of CombatLog	PASS
get Combat Log	2	Log contains battle start entry	constructor	log has 1 entry	log has 1 entry	PASS
getCombatLog	3	Log reference is always same	call getCombatLog() twice	same reference	same reference	PASS
nextRound	1	Increments round number	call nextRound()	roundNumber +1	roundNumber +1	PASS
nextRound	2	Throws if finished	setFinished(true), call nextRound()	GameException thrown	GameException thrown	PASS
nextRound	3	Multiple rounds	setFinished(false), call nextRound() three times	roundNumber=4	roundNumber=4	PASS
setFinished	1	Set finished true	call setFinished(true)	isFinished returns true	isFinished returns true	PASS
setFinished	2	Set finished false	call setFinished(false)	isFinished returns false	isFinished returns false	PASS
setFinished	3	Toggle finish	setFinished(true), setFinished (false)	isFinished false	isFinished false	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
getName	1	Returns name as 'Defend'	call getName()	"Defend"	"Defend"	PASS
getName	2	Check not null	call getName()	not null	not null	PASS
getName	3	Check equals ignore case	call getName()	"defend".equalsIgnoreCase (result)	TRUE	PASS
getDescription	1	Returns correct description	call getDescription()	"Brace yourself to reduce incoming damage."	"Brace yourself to reduce incoming damage."	PASS

getDescription	2	Check not empty	call getDescription()	not empty	not empty	PASS
getDescription	3	Check contains 'defensive'	call getDescription()	contains 'defensive'	TRUE	PASS
getEpCost	1	Returns constant value	Constants.DEFEND_EP_COST (e.g. 3)	3	3	PASS
getEpCost	2	EP cost non-negative	getEpCost() >= 0	TRUE	TRUE	PASS
getEpCost	3	Consistent across calls	call twice, equal results	TRUE	TRUE	PASS
execute	1	Successful defend (enough EP)	user has EP ≥ cost	log has "defensive stance"	log has "defensive stance"	PASS
execute	2	Not enough EP	user EP < cost	GameException thrown	GameException thrown	PASS
execute	3	Null user argument	user=null	GameException thrown	GameException thrown	PASS
execute	4	Null log argument	log=null	GameException thrown	GameException thrown	PASS
execute	5	Status effect applied	user has status effect DEFENSE_UP	user has status effect DEFENSE_UP	user has status effect DEFENSE_UP	PASS
execute	6	EP deducted	user initial EP = 10, EP after = 7	user EP decreased by cost	user EP decreased by cost	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
getName	1	Returns correct name	SingleUseItem with name "Potion"	"Use Potion"	"Use Potion"	PASS
getName	2	Returns name for another item	SingleUseItem with name "Bomb"	"Use Bomb"	"Use Bomb"	PASS
getName	3	Special characters in item name	SingleUseItem with name "Elixir+1"	"Use Elixir+1"	"Use Elixir+1"	PASS
getDescription	1	Returns correct description	SingleUseItem with description "Restores 50 HP"	"Restores 50 HP"	"Restores 50 HP"	PASS
getDescription	2	Description with special characters	SingleUseItem with description "Deals 100% Fire Damage"	"Deals 100% Fire Damage"	"Deals 100% Fire Damage"	PASS
getDescription	3	Multiple calls	Call getDescription() twice with same item	same result	same result	PASS
getEpCost	1	Always returns 0	Any SingleUseItem	0	0	PASS

getEpCost	2	Non-negative check	Any SingleUseItem	>=0	0	PASS
getEpCost	3	Consistency check	Call getEpCost() twice	same result	same result	PASS
execute	1	Item present in inventory	User has "Potion" in inventory	"Log entry added, effect applied, item removed"	"Log entry added, effect applied, item removed"	PASS
execute	2	Item missing in inventory	User does not have item	GameException thrown	GameException thrown	PASS
execute	3	Null user input	user=null	GameException thrown	GameException thrown	PASS
execute	4	Null log input	log=null	GameException thrown	GameException thrown	PASS
execute	5	Effect applies correctly	Item applyEffect updates HP	User HP increased	User HP increased	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
calculateXpGained	1	Normal case	winner(lvl 2), loser(lvl 3)	55	55	PASS
calculateXpGained	2	Loser at level 1	winner(lvl 5), loser(lvl 1)	35	35	PASS
calculateXpGained	3	Winner and loser are the same	same Character	GameException thrown	GameException thrown	PASS
calculateXpGained	4	Null winner	null winner, valid loser	GameException thrown	GameException thrown	PASS
calculateXpGained	5	Null loser	valid winner, null loser	GameException thrown	GameException thrown	PASS
processLevelUp	1	No level up	XP = 50, Level = 1	FALSE	FALSE	PASS
processLevelUp	2	Level up to 2	XP = 100, Level = 1	TRUE	TRUE	PASS
processLevelUp	3	Multi-level up (skipping levels)	XP = 460, Level = 2	TRUE	TRUE	PASS
processLevelUp	4	No change at max	XP = 9999, Level = 5	FALSE	FALSE	PASS
processLevelUp	5	Null character	null	GameException thrown	GameException thrown	PASS
processLevelUp	6	Level up sets HP/EP	newMaxHp = oldMaxHp + 10 * (levels gained)	newMaxHp as expected	newMaxHp as expected	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
getName	1	AbilityMove returns ability name	AbilityMove(DamageAbility)	"Fireball"	"Fireball"	PASS
getName	2	Defend returns correct name	Defend()	"Defend"	"Defend"	PASS
getName	3	ItemMove returns use text	ItemMove(Potion)	"Use Potion"	"Use Potion"	PASS
getDescription	1	AbilityMove returns ability description	AbilityMove(HealAbility)	"Heals target for 20 HP"	"Heals target for 20 HP"	PASS
getDescription	2	Defend returns correct description	Defend()	"Brace yourself to reduce incoming damage."	"Brace yourself to reduce incoming damage."	PASS
getDescription	3	ItemMove returns item description	ItemMove(Bomb)	"Deals 30 damage to enemy."	"Deals 30 damage to enemy."	PASS
getEpCost	1	AbilityMove returns correct EP cost	AbilityMove(ability with cost 3)	3	3	PASS
getEpCost	2	Defend returns constant EP cost	Defend()	2	2	PASS
getEpCost	3	ItemMove returns 0	ItemMove(Potion)	0	0	PASS
execute	1	AbilityMove applies damage	execute(user, target, log)	Target HP reduced	Target HP reduced	PASS
execute	2	Defend applies status effect	execute(user, user, log)	user has DEFENSE_UP	user has DEFENSE_UP	PASS
execute	3	ItemMove uses item and logs	execute(user with item, target, log)	item effect applied, log updated	item effect applied, log updated	PASS
execute	4	AbilityMove not enough EP	execute(user with low EP, target, log)	GameException thrown	GameException thrown	PASS
execute	5	ItemMove item not in inventory	execute(user missing item, target, log)	GameException thrown	GameException thrown	PASS
execute	6	Null combat log	execute(user, target, null)	GameException thrown	GameException thrown	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
getName	1	Defend returns correct name	Defend()	"Defend"	"Defend"	PASS

getName	2	Custom MoveAction returns custom name	CustomMoveAction("Guard")	"Guard"	"Guard"	PASS
getName	3	Null implementation returns null	NullMoveAction()	null	null	PASS
getDescription	1	Defend returns correct description	Defend()	"Brace yourself to reduce incoming damage."	"Brace yourself to reduce incoming damage."	PASS
getDescription	2	Custom MoveAction returns custom description	CustomMoveAction("Guard")	"Custom action description"	"Custom action description"	PASS
getDescription	3	Null implementation returns null	NullMoveAction()	null	null	PASS
getEpCost	1	Defend returns correct EP cost	Defend()	2	2	PASS
getEpCost	2	Custom MoveAction returns custom cost	CustomMoveAction(cost=4)	4	4	PASS
getEpCost	3	Null implementation returns 0	NullMoveAction()	0	0	PASS
execute	1	Defend applies status effect	execute(user, user, log)	user has DEFENSE_UP	user has DEFENSE_UP	PASS
execute	2	Defend throws exception for not enough EP	execute(user with EP=0, user, log)	GameException thrown	GameException thrown	PASS
execute	3	Defend logs action	execute(user, user, log)	log includes "takes a defensive stance."	log includes "takes a defensive stance."	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
getName	1	Defend returns correct name	Defend()	"Defend"	"Defend"	PASS
getName	2	Custom MoveAction returns custom name	CustomMoveAction("Guard")	"Guard"	"Guard"	PASS
getName	3	Null implementation returns null	NullMoveAction()	null	null	PASS
getDescription	1	Defend returns correct description	Defend()	"Brace yourself to reduce incoming damage."	"Brace yourself to reduce incoming damage."	PASS
getDescription	2	Custom MoveAction returns custom description	CustomMoveAction("Guard")	"Custom action description"	"Custom action description"	PASS
getDescription	3	Null implementation returns null	NullMoveAction()	null	null	PASS
getEpCost	1	Defend returns correct EP cost	Defend()	2	2	PASS

getEpCost	2	Custom MoveAction returns custom cost	CustomMoveAction(cost=4)	4	4	PASS
getEpCost	3	Null implementation returns 0	NullMoveAction()	0	0	PASS
execute	1	Defend applies status effect	execute(user, user, log)	user has DEFENSE_UP	user has DEFENSE_UP	PASS
execute	2	Defend throws exception for not enough EP	execute(user with EP=0, user, log)	GameException thrown	GameException thrown	PASS
execute	3	Defend logs action	execute(user, user, log)	log includes "takes a defensive stance."	log includes "takes a defensive stance."	PASS
getName	1	Returns correct name	Recharge()	"Recharge"	"Recharge"	PASS
getName	2	Method does not throw exception	Recharge()	No Exception	No Exception	PASS
getName	3	Check if not blank	Recharge()	Non-blank	Non-blank	PASS
getDescription	1	Returns correct description	Recharge()	"Focus to regain a little EP."	"Focus to regain a little EP."	PASS
getDescription	2	Method does not throw exception	Recharge()	No Exception	No Exception	PASS
getDescription	3	Check if not blank	Recharge()	Non-blank	Non-blank	PASS
getEpCost	1	Returns zero as cost	Recharge()	0	0	PASS
getEpCost	2	Does not throw exception	Recharge()	No Exception	No Exception	PASS
getEpCost	3	Check cost is not negative	Recharge()	Cost >= 0	Cost >= 0	PASS
execute	1	Recharges user by correct EP	User with 1 EP, execute()	User EP increased by Constants. RECHARGE_EP_GAIN	User EP increased by Constants. RECHARGE_EP_GAIN	PASS
execute	2	Adds correct log entry	User, execute()	Log includes "recharges and gains"	Log includes "recharges and gains"	PASS
execute	3	Throws exception on null user	execute(null, target, log)	GameException thrown	GameException thrown	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	PASS
Ability (constructor)	1	Valid ability creation	All valid fields	Ability instance created	Ability instance created	PASS

	Ability (constructor)	2	Blank name	name="", GameException	GameException	GameException	PASS
	Ability (constructor)	3	Null effect type	effectType=null, GameException	GameException	GameException	PASS
	getName	1	Returns correct name	Valid ability	"Fireball"	"Fireball"	PASS
	getName	2	Returns name after creation	Ability("Ice",...)	"Ice"	"Ice"	PASS
	getName	3	Non-blank test	Ability("Heal",...)	Non-blank	Non-blank	PASS
	getDescription	1	Returns correct description	Ability("Fireball", "Deals fire damage", ...)	"Deals fire damage"	"Deals fire damage"	PASS
	getDescription	2	Non-blank description	Ability("Stun", "Stuns foe", ...),	Non-blank	Non-blank	PASS
	getDescription	3	Returns description after creation	Ability("Poison", "Poisons", ...),	"Poisons"	"Poisons"	PASS
	getEpCost	1	Returns correct epCost	Ability(..., 3, ...),	3	3	PASS
	getEpCost	2	Zero cost	Ability(..., 0, ...),	0	0	PASS
	getEpCost	3	Max allowed cost	Ability(..., Constants. MAX_EP_COST, ...),	Constants.MAX_EP_COST	Constants. MAX_EP_COST	PASS
	getAbilityEffectType	1	Returns effect type	Ability(...,AbilityEffectType. DAMAGE,...),	AbilityEffectType.DAMAGE	AbilityEffectType. DAMAGE	PASS
	getAbilityEffectType	2	APPLY_STATUS type	Ability(...,AbilityEffectType. APPLY_STATUS,...),	AbilityEffectType. APPLY_STATUS	AbilityEffectType. APPLY_STATUS	PASS
	getAbilityEffectType	3	HEAL type	Ability(...,AbilityEffectType. HEAL,...),	AbilityEffectType.HEAL	AbilityEffectType.HEAL	PASS
	getEffectValue	1	Returns effect value	Ability(..., 15, ...),	15	15	PASS
	getEffectValue	2	Zero effect value	Ability(..., 0, ...),	0	0	PASS
	getEffectValue	3	Max effect value	Ability(..., Constants. MAX_EFFECT_VALUE, ...),	Constants.MAX_EFFECT_VALUE	Constants. MAX_EFFECT_VALUE	PASS
	getStatusEffectApplied	1	Returns correct status	Ability(...,StatusEffectType. POISON),	StatusEffectType.POISON	StatusEffectType. POISON	PASS
	getStatusEffectApplied	2	Null status if not APPLY_STATUS	Ability(...,AbilityEffectType. DAMAGE,...),	null	null	PASS
	getStatusEffectApplied	3	Correct status if set	Ability(...,AbilityEffectType. APPLY_STATUS,..., StatusEffectType.STUN),	StatusEffectType.STUN	StatusEffectType.STUN	PASS

equals	1	Equal by name	Ability("X"),Ability("X"),	TRUE	TRUE	PASS
equals	2	Not equal with different name	Ability("X"),Ability("Y"),	FALSE	FALSE	PASS
equals	3	Not equal with null	null,	FALSE	FALSE	PASS
hashCode	1	Consistent hash	Ability("Fireball"),Ability ("Fireball"),	Same hashcode	Same hashcode	PASS
hashCode	2	Equal abilities same hash	Ability("X"),Ability("X"),	Same hash	Same hash	PASS
hashCode	3	Unequal abilities different hash	Ability("A"),Ability("B"),	Different hash	Different hash	PASS
toString	1	Returns name	Ability("Thunder"),	"Thunder"	"Thunder"	PASS
toString	2	Non-empty string	Ability("Quick Heal"),	Non-empty	Non-empty	PASS
toString	3	Matches getName	Ability("Freeze"),	getName(),toString()	Pass	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
valueOf	1	Correct value lookup	"DAMAGE"	AbilityEffectType.DAMAGE	AbilityEffectType. DAMAGE	PASS
valueOf	2	Incorrect value (case-sensitive)	"damage"	IllegalArgumentException	IllegalArgumentException	PASS
valueOf	3	Non-existent value	"FOO"	IllegalArgumentException	IllegalArgumentException	PASS
toString	1	Returns name for DAMAGE	AbilityEffectType.DAMAGE	"DAMAGE"	"DAMAGE"	PASS
toString	2	Returns name for UTILITY	AbilityEffectType.UTILITY	"UTILITY"	"UTILITY"	PASS
toString	3	Returns name for DEFENSE	AbilityEffectType.DEFENSE	"DEFENSE"	"DEFENSE"	PASS
ordinal	1	Ordinal of DAMAGE	AbilityEffectType.DAMAGE	0	0	PASS
ordinal	2	Ordinal of DEFENSE	AbilityEffectType.DEFENSE	4	4	PASS
ordinal	3	Ordinal of UTILITY	AbilityEffectType.UTILITY	6	6	PASS

enumValues	1	All enum values present	AbilityEffectType.values()	[DAMAGE,HEAL,ENERGY_GAIN, APPLY_STATUS,DEFENSE, EVADE,UTILITY]	[DAMAGE,HEAL, ENERGY_GAIN, APPLY_STATUS, DEFENSE,EVADE, UTILITY]	PASS
enumReferenceEquality	1	Compare two enum references	AbilityEffectType.DAMAGE==AbilityEffectType.valueOf	TRUE	TRUE	PASS
enumImmutability	1	Enums can't be extended	None	Compile Error	Compile Error	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
valueOf	1	Correct value lookup	"DAMAGE"	AbilityType.DAMAGE	AbilityType.DAMAGE	PASS
valueOf	2	Incorrect value (case-sensitive)	"damage"	IllegalArgumentException	IllegalArgumentException	PASS
valueOf	3	Non-existent value	"ATTACK"	IllegalArgumentException	IllegalArgumentException	PASS
toString	1	Returns name for DAMAGE	AbilityType.DAMAGE	"DAMAGE"	"DAMAGE"	PASS
toString	2	Returns name for UTILITY	AbilityType.UTILITY	"UTILITY"	"UTILITY"	PASS
toString	3	Returns name for STATUS	AbilityType.STATUS	"STATUS"	"STATUS"	PASS
ordinal	1	Ordinal of DAMAGE	AbilityType.DAMAGE	0	0	PASS
ordinal	2	Ordinal of DEFENSE	AbilityType.DEFENSE	3	3	PASS
ordinal	3	Ordinal of STATUS	AbilityType.STATUS	6	6	PASS
enumValues	1	All enum values present	AbilityType.values()	[DAMAGE,HEAL,ENERGY_GAIN, DEFENSE,EVADE,UTILITY, STATUS]	[DAMAGE,HEAL, ENERGY_GAIN,DEFENSE, EVADE,UTILITY,STATUS]	PASS
enumReferenceEquality	1	Compare two enum references	AbilityType.DAMAGE==AbilityType.valueOf("DAMAGE")	TRUE	TRUE	PASS
enumImmutability	1	Enums can't be extended	None	Compile Error	Compile Error	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
takeDamage	1	Normal damage taken	currentHp=50, damage=20	currentHp=30	currentHp=30	PASS
takeDamage	2	Damage with DEFENSE_UP	status=DEFENSE_UP, currentHp=50, damage=20	currentHp=40 (damage halved)	currentHp=40	PASS
takeDamage	3	Damage with IMMUNITY	status=IMMUNITY, currentHp=50, damage=20	currentHp=50 (no damage)	currentHp=50	PASS
heal	1	Standard healing	currentHp=20, heal=15	currentHp=35	currentHp=35	PASS
heal	2	Heal above max	currentHp=95, heal=10, maxHp=100	currentHp=100	currentHp=100	PASS
heal	3	Heal zero	currentHp=20, heal=0	currentHp=20	currentHp=20	PASS
spendEp	1	Enough EP	currentEp=10, cost=5	TRUE	TRUE	PASS
spendEp	2	Not enough EP	currentEp=3, cost=5	FALSE	FALSE	PASS
spendEp	3	Negative cost	currentEp=10, cost=-1	FALSE	FALSE	PASS
gainEp	1	Normal gain	currentEp=10, amount=5, maxEp=20	currentEp=15	currentEp=15	PASS
gainEp	2	Gain above max	currentEp=19, amount=5, maxEp=20	currentEp=20	currentEp=20	PASS
gainEp	3	Gain zero	currentEp=10, amount=0	currentEp=10	currentEp=10	PASS
addStatusEffect	1	Add effect not present	empty effects, add POISONED	effect added	effect added	PASS
addStatusEffect	2	Add with Elven Cloak equipped	Elven Cloak equipped, not used, effect=STUNNED	statusEffectImmunityUsed set	statusEffectImmunityUsed set	PASS
addStatusEffect	3	Add over MAX_STATUS_EFFECTS	already has MAX, add POISONED	effect not added	effect not added	PASS
removeStatusEffect	1	Remove present effect	add POISONED, then remove POISONED	effect removed	effect removed	PASS
removeStatusEffect	2	Remove absent effect	no effect, remove POISONED	nothing removed	nothing removed	PASS
removeStatusEffect	3	Remove one of many effects	add POISONED+STUNNED, remove POISONED	STUNNED remains	STUNNED remains	PASS
equipItem	1	Equip valid item	item in inventory	item equipped	item equipped	PASS

equipItem	2	Equip not in inventory	item not in inventory	GameException thrown	GameException thrown	PASS
equipItem	3	Equip null item	null	GameException thrown	GameException thrown	PASS
useSingleUseItem	1	Use item present	single-use in inventory	item used	item used	PASS
useSingleUseItem	2	Use item not present	not in inventory	GameException thrown	GameException thrown	PASS
useSingleUseItem	3	Use null item	null	GameException thrown	GameException thrown	PASS
setAbilities	1	Set valid abilities	abilities.size=abilitySlotCount	abilities set	abilities set	PASS
setAbilities	2	Set too many abilities	abilities.size>abilitySlotCount	IllegalArgumentException	IllegalArgumentException	PASS
setAbilities	3	Set with null	abilities=null	IllegalArgumentException	IllegalArgumentException	PASS
processStartOfTurnEffects	1	POISONED effect	active=POISONED, currentHp=20	hp reduced, log message	hp reduced, log message	PASS
processStartOfTurnEffects	2	Expired effect	effect.duration=0	effect removed	effect removed	PASS
processStartOfTurnEffects	3	No effects	none	nothing happens	nothing happens	PASS
levelUp	1	Enough battles to level up	battlesWon=5	level increased	level increased	PASS
levelUp	2	Not enough battles	battlesWon=2	level unchanged	level unchanged	PASS
levelUp	3	Multi-level up	battlesWon=11	level increased by multiple	level increased by multiple	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
getDescription	1	Get Mage description	MAGE	"Glass-cannon spell-caster with powerful elemental attacks."	"Glass-cannon spell-caster with powerful elemental attacks."	PASS
getDescription	2	Get Paladin description	PALADIN	"Holy warrior wielding sacred power and sturdy defences."	"Holy warrior wielding sacred power and sturdy defences."	PASS
getDescription	3	Get Engineer description	ENGINEER	"Tech-savvy combatant deploying machines and tools."	"Tech-savvy combatant deploying machines and tools."	PASS

getBaseHP	1	Get Mage base HP	MAGE	100	100	PASS
getBaseHP	2	Get Paladin base HP	PALADIN	120	120	PASS
getBaseHP	3	Get Warrior base HP	WARRIOR	100	100	PASS
getBaseEP	1	Get Rogue base EP	ROGUE	50	50	PASS
getBaseEP	2	Get Summoner base EP	SUMMONER	50	50	PASS
getBaseEP	3	Get Paladin base EP	PALADIN	60	60	PASS
toString	1	Check toString on Warrior	WARRIOR	"WARRIOR"	"WARrior"	PASS
toString	2	Check toString on Mage	MAGE	"MAGE"	"MAGE"	PASS
toString	3	Check toString on Summoner	SUMMONER	"SUMMONER"	"SUMMONer"	PASS
valuesAsStringArray	1	Check array length	valuesAsStringArray()	6	6	PASS
valuesAsStringArray	2	Check first element	valuesAsStringArray()[0]	"MAGE"	"MAGE"	PASS
valuesAsStringArray	3	Check last element	valuesAsStringArray()[5]	"PALADIN"	"PALADIN"	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
getName	1	Return name for entry	"name=""Alice"", wins=3"	Alice	Alice	PASS
getName	2	Return name with spaces	"name=""Sir Bob"", wins=0"	"Sir Bob"	"Sir Bob"	PASS
getName	3	Return name with number	"name=""Player9"", wins=1"	Player9	Player9	PASS
getWins	1	Get wins after construction	"name=""Carl"", wins=7"	7	7	PASS
getWins	2	Get wins for zero wins	"name=""Dina"", wins=0"	0	0	PASS
getWins	3	Get wins after increment	"name=""Eva"", wins=2"	3	3	PASS
getXp	1	Get XP at default	"name=""Foo"", wins=1"	0	0	PASS

getXp	2	Get XP after setXp	"entry.setXp(150)"	150	150	PASS
getXp	3	Get XP with custom ctor	"name=""Gina"", wins=2, xp=200, ts=1000"	200	200	PASS
getLastUpdated	1	Constructor sets lastUpdated	"(constructor)"	not null	not null	PASS
getLastUpdated	2	After incrementWins	"entry.incrementWins()"	timestamp increases	timestamp increases	PASS
getLastUpdated	3	After setXp	"entry.setXp(99)"	timestamp increases	timestamp increases	PASS
incrementWins	1	Increment once	"initial wins=1"	2	2	PASS
incrementWins	2	Increment from zero	"initial wins=0"	1	1	PASS
incrementWins	3	Multiple increments	"increment 3 times"	initial+3	initial+3	PASS
setXp	1	Set XP once	"setXp(42)"	42	42	PASS
setXp	2	Set XP twice	"setXp(50), setXp(200)"	200	200	PASS
setXp	3	Set negative XP	"setXp(-5)"	Exception thrown	Exception thrown	PASS
toString	1	ToString simple	"name=""Jack"", wins=1, xp=10"	"Jack (Wins: 1, XP: 10)"	"Jack (Wins: 1, XP: 10)"	PASS
toString	2	ToString zeroes	"name=""Kiki"", wins=0, xp=0"	"Kiki (Wins: 0, XP: 0)"	"Kiki (Wins: 0, XP: 0)"	PASS
toString	3	ToString with spaces	"name=""L L"", wins=2, xp=22"	"L L (Wins: 2, XP: 22)"	"L L (Wins: 2, XP: 22)"	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
getName	1	Get name after construction	name="Ruth"	Ruth	Ruth	PASS
getName	2	Get name with spaces	name="Sir Ben"	Sir Ben	Sir Ben	PASS
getName	3	Get name case sensitivity	name="Amy"	Amy	Amy	PASS
getCharacters	1	New player	No characters	[],	[],	PASS
getCharacters	2	Add one character	addCharacter(CharA)	[CharA],	[CharA],	PASS

getCharacters	3	Add multiple	addCharacter(CharB), addCharacter(CharC)	[CharB, CharC],	[CharB, CharC],	PASS
getCumulativeWins	1	Initial wins	No increment	0	0	PASS
getCumulativeWins	2	After incrementWins	incrementWins()	1	1	PASS
getCumulativeWins	3	Increment twice	incrementWins() x2	2	2	PASS
addCharacter	1	Add unique character	addCharacter(CharX)	Success	Success	PASS
addCharacter	2	Duplicate name	addCharacter(CharY), addCharacter(CharY)	GameException thrown	GameException thrown	PASS
addCharacter	3	Add over limit	addCharacter up to MAX_CHARACTERS_PER_PLAYER, addCharacter(CharZ)	GameException thrown	GameException thrown	PASS
removeCharacter	1	Remove existing character	addCharacter(A), removeCharacter("A")	TRUE	TRUE	PASS
removeCharacter	2	Remove non-existing	removeCharacter("Z")	FALSE	FALSE	PASS
removeCharacter	3	Remove by different case	addCharacter(B), removeCharacter("b")	TRUE	TRUE	PASS
getCharacter	1	Get existing character	addCharacter(A), getCharacter ("A")	Optional[A],	Optional[A],	PASS
getCharacter	2	Get missing character	getCharacter("X")	Optional.empty	Optional.empty	PASS
getCharacter	3	Case-insensitive get	addCharacter(C), getCharacter ("c")	Optional[C],	Optional[C],	PASS
incrementWins	1	Increment once	incrementWins()	1	1	PASS
incrementWins	2	Increment after addCharacter	addCharacter(D), incrementWins()	1	1	PASS
incrementWins	3	Multiple increments	incrementWins() x3	3	3	PASS
toString	1	Player with no characters	toString()	"Player [name=Eve, characters=0, wins=0]"	"Player [name=Eve, characters=0, wins=0]"	PASS
toString	2	Player with 2 chars	addCharacter(F), addCharacter (G), toString()	"Player [name=Hank, characters=2, wins=0]"	"Player [name=Hank, characters=2, wins=0]"	PASS
toString	3	Player with wins	incrementWins(), toString()	"Player [name=Ivan, characters=0, wins=1]"	"Player [name=Ivan, characters=0, wins=1]"	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
RaceBonus (constructor)	1	Valid bonuses	10, 20, 2001	Instance created	Instance created	PASS
RaceBonus (constructor)	2	All zero bonuses	0, 0, 0	Instance created	Instance created	PASS
RaceBonus (constructor)	3	Negative hpBonus	-1, 5, 0	GameException thrown	GameException thrown	PASS
getHpBonus	1	Retrieve positive hpBonus	10, getHpBonus()	10	10	PASS
getHpBonus	2	Zero hpBonus	0, getHpBonus()	0	0	PASS
getHpBonus	3	Multiple calls	20, getHpBonus() x2	20	20	PASS
getEpBonus	1	Retrieve positive epBonus	10, 20, 1, getEpBonus()	20	20	PASS
getEpBonus	2	Zero epBonus	15, 0, 2, getEpBonus()	0	0	PASS
getEpBonus	3	Check after creation	5, 8, 0, getEpBonus()	8	8	PASS
getExtraAbilitySlots	1	Retrieve positive slots	5, 3, 2, getExtraAbilitySlots()	2	2	PASS
getExtraAbilitySlots	2	Zero slots	4, 6, 0, getExtraAbilitySlots()	0	0	PASS
getExtraAbilitySlots	3	Multiple slots	9, 12, 3, getExtraAbilitySlots()	3	3	PASS
toString	1	Typical bonuses	7, 5, 2001	"RaceBonus{hp=7, ep=5, slots=1}"	"RaceBonus{hp=7, ep=5, slots=1}"	PASS
toString	2	All zeros	0, 0, 0	"RaceBonus{hp=0, ep=0, slots=0}"	"RaceBonus{hp=0, ep=0, slots=0}"	PASS
toString	3	Large values	100, 200, 10	"RaceBonus{hp=100, ep=200, slots=10}"	"RaceBonus{hp=100, ep=200, slots=10}"	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
getEquippedItem	1	Returns null if nothing equipped	empty inventory	null	null	PASS
getEquippedItem	2	Returns equipped item after equip	add "Ring", equip "Ring"	"Ring"	"Ring"	PASS

	getEquippedItem	3	Returns null after unequip	add "Cape", equip "Cape", unequip	null	null	PASS
	getAllItems	1	Returns empty list for new inventory	new Inventory()	[],	[],	PASS
	getAllItems	2	Returns items after add	add "Wand"	"[Wand]"	"[Wand]"	PASS
	getAllItems	3	Immutability check	getAllItems(), attempt to modify	UnsupportedOperationException	UnsupportedOperationException	PASS
	hasItem	1	Returns false for non-existent	empty, hasItem("Hat")	FALSE	FALSE	PASS
	hasItem	2	Returns true after add	add "Hat", hasItem("Hat")	TRUE	TRUE	PASS
	hasItem	3	Returns false after remove	add "Gem", remove "Gem", hasItem("Gem")	FALSE	FALSE	PASS
	addItem	1	Add unique item	add "Amulet"	"[Amulet]"	"[Amulet]"	PASS
	addItem	2	No duplicates allowed	add "Amulet" twice	"[Amulet]"	"[Amulet]"	PASS
	addItem	3	Throws on null	add null	NullPointerException	NullPointerException	PASS
	removeItem	1	Removes item	add "Cloak", remove "Cloak"	TRUE	TRUE	PASS
	removeItem	2	No effect if not present	remove "Sword" from empty	FALSE	FALSE	PASS
	removeItem	3	Removes equipped item	add "Boots", equip "Boots", remove "Boots"	equippedItem null	equippedItem null	PASS
	equipItem	1	Equips valid item	add "Orb", equip "Orb"	equippedItem "Orb"	equippedItem "Orb"	PASS
	equipItem	2	Throws if not present	equip "Stone" (not added)	GameException	GameException	PASS
	equipItem	3	Replaces previous	add "Ring","Cape", equip "Ring", equip "Cape"	equipped "Cape"	equipped "Cape"	PASS
	unequipItem	1	Unequips item	add "Talisman", equip "Talisman", unequip	equipped null	equipped null	PASS
	unequipItem	2	No effect if already null	empty, unequip	equipped null	equipped null	PASS
	unequipItem	3	Does not remove from inventory	add "Crown", equip "Crown", unequip	getAllItems(),"[Crown]"	"[Crown]"	PASS
	useSingleUseItem	1	Removes and unequips	add+equip "Bomb", useSingleUseItem("Bomb")	not in items	not in items	PASS

useSingleUseItem	2	Throws if not present	useSingleUseItem("Elixir")	GameException	GameException	PASS
useSingleUseItem	3	Does not throw on null	add "Potion", useSingleUseItem (null)	NullPointerException	NullPointerException	PASS
readObject	1	Initializes items after deserialization	deserialize with null items	items not null	items not null	PASS
readObject	2	Preserves equipped after deserialization	serialize+deserialize with equipped	still equipped	still equipped	PASS
readObject	3	Handles empty inventory	deserialize empty inventory	items empty	items empty	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
valueOf	1	Retrieve SINGLE_USE enum by name	"SINGLE_USE"	ItemType.SINGLE_USE	ItemType.SINGLE_USE	PASS
valueOf	2	Retrieve PASSIVE enum by name	"PASSIVE"	ItemType.PASSIVE	ItemType.PASSIVE	PASS
valueOf	3	Throws exception for invalid name	"INVALID"	IllegalArgumentException	IllegalArgumentExceptio n	PASS
values	1	Returns all enum constants	ItemType.values()	[SINGLE_USE, PASSIVE]	[SINGLE_USE, PASSIVE]	PASS
Integration	1	Identify SINGLE_USE item in inventory	MagicItem.type=SINGLE_USE	"MagicItem.isSingleUse()=true"	"MagicItem.isSingleUse() =true"	PASS
Integration	2	Identify PASSIVE item in inventory	MagicItem.type=PASSIVE	"MagicItem.isPassive()=true"	"MagicItem.isPassive() =true"	PASS
Integration	3	Inventory distinguishes activation	Inventory.addItem (type=SINGLE_USE)	SINGLE_USE is consumed on use	SINGLE_USE is consumed on use	PASS
Integration	4	Inventory distinguishes passive	Inventory.equipItem (type=PASSIVE)	PASSIVE remains equipped	PASSIVE remains equipped	PASS
toString	1	Enum name as string	ItemType.SINGLE_USE.toString()	"SINGLE_USE"	"SINGLE_USE"	PASS
toString	2	Enum name as string	ItemType.PASSIVE.toString()	"PASSIVE"	"PASSIVE"	PASS

Method	⊖ #	Test Description	Test Input	Expected Output	Actual Output	Status

Constructor	1	Valid construction	"name=Potion, desc=Restores HP, type=SINGLE_USE, rarity=COMMON"	MagicItem created	MagicItem created	PASS
Constructor	2	Blank name	"name=, desc=Restores HP, type=SINGLE_USE, rarity=COMMON"	GameException thrown	GameException thrown	PASS
Constructor	3	Null type	"name=Potion, desc=Restores HP, type=null, rarity=COMMON"	GameException thrown	GameException thrown	PASS
getName	1	Returns correct name	"Potion"	"Potion"	"Potion"	PASS
getDescription	1	Returns correct description	"Potion, Restores HP"	"Restores HP"	"Restores HP"	PASS
getItemType	1	Returns correct type	"Potion, type=SINGLE_USE"	SINGLE_USE	SINGLE_USE	PASS
getRarity	1	Returns rarity string	"Potion, rarity=RARE"	"RARE"	"RARE"	PASS
getRarityType	1	Returns rarity enum	"Potion, rarity=RARE"	RARE	RARE	PASS
getDropChance	1	Returns rarity drop chance	"Potion, rarity=COMMON"	COMMON.getDropChance()	COMMON. getDropChance()	PASS
equals	1	Equals identical item	"Potion, desc=HP, type=SINGLE_USE, rarity=COMMON"	"true"	"true"	PASS
equals	2	Not equal to different name	"Elixir, desc=HP, type=SINGLE_USE, rarity=COMMON"	"false"	"false"	PASS
equals	3	Not equal to null	null	"false"	"false"	PASS
equals	4	Not equal to different class	"MagicItem vs. String"	"false"	"false"	PASS
hashCode	1	Equal items have equal hash	"Potion, Potion (same fields)"	same hash	same hash	PASS
toString	1	Display format	"Potion, type=SINGLE_USE"	"Potion (SINGLE_USE)"	"Potion (SINGLE_USE)"	PASS
copy	1	Deep copy (concrete sub, e.g., SingleUseItem)	SingleUseItem.copy()	New identical SingleUseItem	New identical SingleUseItem	PASS
Method	#	Test Description	Test Input	Expected Output	Actual Output	Status

Constructor	1	Valid PassiveItem	"name=Ring, desc=Immunity, rarity=RARE"	PassiveItem created	PassiveItem created	PASS
Constructor	2	Blank name	"name=, desc=Immunity, rarity=RARE"	GameException thrown	GameException thrown	PASS
Constructor	3	Null rarity	"name=Ring, desc=Immunity, rarity=null"	GameException thrown	GameException thrown	PASS
getName	1	Returns name	"Ring of Immunity"	"Ring of Immunity"	"Ring of Immunity"	PASS
getDescription	1	Returns description	"desc=Protects user"	"Protects user"	"Protects user"	PASS
getItemType	1	Always returns PASSIVE	"PassiveItem instance"	PASSIVE	PASSIVE	PASS
getRarity	1	Returns rarity string	"rarity=RARE"	"RARE"	"RARE"	PASS
getRarityType	1	Returns rarity enum	"rarity=RARE"	RARE	RARE	PASS
getDropChance	1	Returns rarity drop chance	"rarity=COMMON"	COMMON.getDropChance()	COMMON.getDropChance(	PASS
equals	1	Equals identical item	"same fields"	"true"	"true"	PASS
equals	2	Not equal to different name	"diff name"	"false"	"false"	PASS
equals	3	Not equal to null	null	"false"	"false"	PASS
equals	4	Not equal to different class	"PassiveItem vs. String"	"false"	"false"	PASS
hashCode	1	Equal items have equal hash	"identical fields"	same hash	same hash	PASS
toString	1	Display format	"Ring, PASSIVE"	"Ring (PASSIVE)"	"Ring (PASSIVE)"	PASS
copy	1	Deep copy	"call copy() on instance"	New identical PassiveItem	New identical PassiveItem	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
valueOf	1	Returns COMMON enum	"RarityType.valueOf('COMMON')"	RarityType.COMMON	RarityType.COMMON	PASS
valueOf	2	Returns RARE enum	"RarityType.valueOf('RARE')"	RarityType.RARE	RarityType.RARE	PASS
getDropChance	1	Common drop chance	"RarityType.COMMON.getDropChance()"	60	60	PASS

getDropChance	2	Uncommon drop chance	"RarityType.UNCOMMON.getDropChance()"	35	35	PASS
getDropChance	3	Rare drop chance	"RarityType.RARE.getDropChance()"	5	5	PASS
toString	1	COMMON returns 'Common'	"RarityType.COMMON.toString()"	"Common"	"Common"	PASS
toString	2	UNCOMMON returns 'Uncommon'	"RarityType.UNCOMMON.toString()"	"RarityType.UNCOMMON.toString()"	"Uncommon"	PASS
toString	3	RARE returns 'Rare'	"RarityType.RARE.toString()"	"RarityType.RARE.toString()"	"Rare"	PASS
values	1	Enum contains all tiers	"RarityType.values()"	"[COMMON, UNCOMMON, RARE]"	"[COMMON, UNCOMMON,	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
valueOf	1	Returns HEAL_HP enum	"SingleUseEffectType.valueOf('HEAL_HP')"	SingleUseEffectType.HEAL_HP	SingleUseEffectType.HEAL_HP	PASS
valueOf	2	Returns DAMAGE enum	"SingleUseEffectType.valueOf('DAMAGE')"	SingleUseEffectType.DAMAGE	SingleUseEffectType.DAMAGE	PASS
toString	1	HEAL_HP returns 'HEAL_HP'	"SingleUseEffectType.HEAL_HP.toString()"	"HEAL_HP"	"HEAL_HP"	PASS
toString	2	REVIVE returns 'REVIVE'	"SingleUseEffectType.REVIVE.toString()"	"REVIVE"	"REVIVE"	PASS
values	1	Enum contains all effects	"SingleUseEffectType.values()"	[HEAL_HP, RESTORE_EP, REVIVE, GRANT_IMMUNITY, DAMAGE]	[HEAL_HP, RESTORE_EP, REVIVE, GRANT_IMMUNITY, DAMAGE]	PASS
ordinal	1	Correct ordinal of DAMAGE	"SingleUseEffectType.DAMAGE.ordinal()"	4	4	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor	1	Valid construction	("Potion","Restores HP", RarityType.COMMON,HEAL_HP, 10)	SingleUseItem created	SingleUseItem created	PASS

Constructor	2	Null effect type	("Potion","Restores HP", RarityType.COMMON,null,10)	GameException thrown	GameException thrown	PASS
Constructor	3	Out of range effect value	("Potion","Restores HP", RarityType.COMMON,HEAL_HP, 0)	GameException thrown	GameException thrown	PASS
getEffectType	1	Get effect type of created item	HEAL_HP	HEAL_HP	HEAL_HP	PASS
getEffectValue	1	Get effect value of created item	10	10	10	PASS
equals	1	Equal items	Two identical SingleuseItems	TRUE	TRUE	PASS
equals	2	Not equal (different effectValue)	Same props but different effectValue	FALSE	FALSE	PASS
copy	1	Copy returns new identical object	singleUseItem.copy()	equals returns true	equals returns true	PASS
applyEffect	1	HEAL_HP increases HP	(HP=50,max=100)	HP increases by effectValue	HP increases by effectValue	PASS
applyEffect	2	RESTORE_EP increases EP	(EP=10,max=50)	EP increases by effectValue	EP increases by effectValue	PASS
applyEffect	3	REVIVE when dead	(HP=0,max=100)	HP set to (max*effectValue/100)	HP set to (max*effectValue/100)	PASS
applyEffect	4	REVIVE when alive	(HP>0,max=100)	No HP change	No HP change	PASS
applyEffect	5	GRANT_IMMUNITY status	call applyEffect	user gains IMMUNITY	user gains IMMUNITY	PASS
applyEffect	6	DAMAGE reduces target HP	call applyEffect	target HP decreases	target HP decreases	PASS
applyEffect	7	Unknown type	Invalid type	GameException thrown	GameException thrown	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
getAvailableAbilities	1	Get abilities for MAGE	MAGE	5 Mage abilities	5 Mage abilities	PASS
getAvailableAbilities	2	Get abilities for WARRIOR	WARRIOR	5 Warrior abilities	5 Warrior abilities	PASS
getAvailableAbilities	3	Null input	null	GameException thrown	GameException thrown	PASS
getAvailableAbilities	4	Unknown class	MOCK_CLASS	GameException thrown	GameException thrown	PASS

getClassDescription	1	Get description for PALADIN	PALADIN	"Holy warriors combining defence with divine magic."	"Holy warriors combining defence with divine magic."	PASS
getClassDescription	2	Unknown class	MOCK_CLASS	GameException thrown	GameException thrown	PASS
getClassDescription	3	Null input	null	GameException thrown	GameException thrown	PASS
getAbilitiesByNames	1	Valid ability name array	["Shiv","Backstab"]	List with two matching Ability	List with two matching Ability	PASS
getAbilitiesByNames	2	Name not found	["NoSuchAbility"]	GameException thrown	GameException thrown	PASS
getAbilitiesByNames	3	Null names	null	GameException thrown	GameException thrown	PASS
getAllAbilities	1	Aggregate all abilities	""	List of all abilities (30)	List of all abilities (30)	PASS
getRandomAbilitiesForClass	1	Random 3 Mage abilities	(MAGE,3)	Random 3 Mage abilities	Random 3 Mage abilities	PASS
getRandomAbilitiesForClass	2	Count exceeds available	(ROGUE,99)	All Rogue abilities	All Rogue abilities	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
createRandomReward	1	Random item returns valid MagicItem		Returns MagicItem instance	Returns MagicItem instance	PASS
createRandomReward	2	Random uses SecureRandom internally		Returns item from any pool	Returns item from any pool	PASS
createRandomReward(Random)	1	Force common item	MockRandom (roll=10)	Item from COMMON_ITEMS	Item from COMMON_ITEMS	PASS
createRandomReward(Random)	2	Force uncommon item	MockRandom (roll=65)	Item from UNCOMMON_ITEMS	Item from UNCOMMON_ITEMS	PASS
createRandomReward(Random)	3	Force rare item	MockRandom (roll=98)	Item from RARE_ITEMS	Item from RARE_ITEMS	PASS
createRandomReward(Random)	4	Null random object	null	NullPointerException thrown	NullPointerException thrown	PASS
getRandomItemByRarity	1	Get common item	RarityType.COMMON + MockRandom (idx=1)	Copy of COMMON_ITEMS[1]	Copy of COMMON_ITEMS[1]	PASS
getRandomItemByRarity	2	Get rare item	RarityType.RARE + MockRandom (idx=0)	Copy of RARE_ITEMS[0]	Copy of RARE_ITEMS[0]	PASS

getRandomItemByRarity	3	Null rarity	RarityType=null + valid random	NullPointerException thrown	NullPointerException thrown	PASS
getRandomItemByRarity	4	Null random	RarityType.COMMON + null	NullPointerException thrown	NullPointerException thrown	PASS
getRandomItemByRarity	5	Returned object is a copy	Any call	Object is not same as in static pool	Object is not same as in static pool	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
getBonusFor	1	Return Human race bonus	RaceType.HUMAN	HP=15, EP=5, Slots=0	HP=15, EP=5, Slots=0	PASS
getBonusFor	2	Return Elf race bonus	RaceType.ELF	HP=0, EP=15, Slots=0	HP=0, EP=15, Slots=0	PASS
getBonusFor	3	Return Gnome race bonus	RaceType.GNOME	HP=0, EP=0, Slots=1	HP=0, EP=0, Slots=1	PASS
getBonusFor	4	Return Dwarf race bonus	RaceType.DWARF	HP=30, EP=0, Slots=0	HP=30, EP=0, Slots=0	PASS
getBonusFor	5	Return Vampire race bonus	RaceType.VAMPIRE	HP=0, EP=0, Slots=0	HP=0, EP=0, Slots=0	PASS
getBonusFor	6	Return Orc race bonus	RaceType.ORC	HP=10, EP=0, Slots=0	HP=10, EP=0, Slots=0	PASS
getBonusFor	7	Null input	RaceType=null	GameException thrown	GameException thrown	PASS
getRaceDescription	1	Get description for Human	RaceType.HUMAN	Versatile adventurers equally at home with blade or spell.	Versatile adventurers equally at home with blade or spell.	PASS
getRaceDescription	2	Get description for Elf	RaceType.ELF	Graceful folk attuned to nature and the arcane.	Graceful folk attuned to nature and the arcane.	PASS
getRaceDescription	3	Null input	RaceType=null	GameException thrown	GameException thrown	PASS
getAvailableRaces	1	Get all available races		[HUMAN, ELF, GNOME, DWARF, VAMPIRE, ORC]	[HUMAN, ELF, GNOME, DWARF, VAMPIRE, ORC]	PASS
Singleton	1	Instance is singleton		RaceService.INSTANCE == RaceService.INSTANCE	TRUE	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status

Constructor	1	Create with default duration	new DefenseUpEffect()	getDuration() == 1	getDuration() == 1	PASS
Constructor	2	Create with custom duration	new DefenseUpEffect(3)	getDuration() == 3	getDuration() == 3	PASS
applyEffect	1	Apply effect to valid character	applyEffect(validChar)	No exception	No exception	PASS
applyEffect	2	Apply effect to null	applyEffect(null)	GameException thrown	GameException thrown	PASS
onTurnStart	1	Decrements duration	onTurnStart(validChar) after new Defens...(2)	getDuration() == 1	getDuration() == 1	PASS
onTurnStart	2	Null character	onTurnStart(null)	GameException thrown	GameException thrown	PASS
onTurnEnd	1	No state change	onTurnEnd(validChar)	getDuration() unchanged	getDuration() unchanged	PASS
remove	1	Removes with no side effect	remove(validChar)	No exception	No exception	PASS
getDuration	1	Duration after creation	new DefenseUpEffect(4)	4	4	PASS
getDuration	2	Duration after decrement	new DefenseUpEffect(2), onTurnStart(validC)	1	1	PASS
getType	1	Returns correct effect type	getType()	StatusEffectType.DEFENSE_UP	StatusEffectType. DEFENSE_UP	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor	1	Creates with default duration	new EvadeEffect()	getDuration() == 1	getDuration() == 1	PASS
applyEffect	1	Apply to valid character	applyEffect(validChar)	No exception	No exception	PASS
applyEffect	2	Apply to null	applyEffect(null)	GameException thrown	GameException thrown	PASS
onTurnStart	1	Decrements duration	onTurnStart(validChar) after new EvadeEffect()	getDuration() == 0	getDuration() == 0	PASS
onTurnStart	2	Null character	onTurnStart(null)	GameException thrown	GameException thrown	PASS
onTurnEnd	1	No effect	onTurnEnd(validChar)	getDuration() unchanged	getDuration() unchanged	PASS
remove	1	Remove effect from valid	remove(validChar)	No exception	No exception	PASS

remove	2	Remove from null	remove(null)	GameException thrown	GameException thrown	PASS
getDuration	1	Duration after construction	new EvadeEffect()	1	1	PASS
getDuration	2	Duration after decrement	new EvadeEffect() then onTurnStart(validChar)	0	0	PASS
getType	1	Returns correct effect	getType()	StatusEffectType.EVADING	StatusEffectType.EVADING	PASS
toString	1	Correct string output for 1 t..	new EvadeEffect().toString()	Evading (1 turns left)	Evading (1 turns left)	PASS
toString	2	Correct string output for 0 t..	new EvadeEffect() then onTurnStart(validC)...	Evading (0 turns left)	Evading (0 turns left)	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor	1	Creates with default duration	new ImmunityEffect()	getDuration() == 1	getDuration() == 1	PASS
applyEffect	1	Apply to valid character	applyEffect(validChar)	No exception	No exception	PASS
applyEffect	2	Apply to null	applyEffect(null)	GameException thrown	GameException thrown	PASS
onTurnStart	1	Decrements duration	onTurnStart(validChar) after new ImmunityEffect()	getDuration() == 0	getDuration() == 0	PASS
onTurnStart	2	Null character	onTurnStart(null)	GameException thrown	GameException thrown	PASS
onTurnEnd	1	No effect	onTurnEnd(validChar)	getDuration() unchanged	getDuration() unchanged	PASS
remove	1	Remove effect from valid	remove(validChar)	No exception	No exception	PASS
remove	2	Remove from null	remove(null)	GameException thrown	GameException thrown	PASS
getDuration	1	Duration after construction	new ImmunityEffect()	1	1	PASS
getDuration	2	Duration after decrement	new ImmunityEffect() then onTurnStart(validChar)	0	0	PASS
getType	1	Returns correct effect	getType()	StatusEffectType.IMMUNITY	StatusEffectType.IMMUNIT	PASS
toString	1	Correct string output for 1 t..	new ImmunityEffect().toString()	Immune (1 turns left)	Immune (1 turns left)	PASS
toString	2	Correct string output for 0 t..	new ImmunityEffect() then onTurnStart(validC)...	Immune (0 turns left)	Immune (0 turns left)	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor	1	Creates with correct duration	new MarkedEffect()	getDuration() == 2	getDuration() == 2	PASS
applyEffect	1	Apply to valid character	applyEffect(validChar)	No exception	No exception	PASS
applyEffect	2	Apply to null character	applyEffect(null)	GameException thrown	GameException thrown	PASS
onTurnStart	1	Decrements duration by 1	onTurnStart(validChar) after new MarkedEff()	getDuration() == 1	getDuration() == 1	PASS
onTurnStart	2	Decrements duration to 0	Call onTurnStart(validChar) twice	getDuration() == 0	getDuration() == 0	PASS
onTurnStart	3	Null character	onTurnStart(null)	GameException thrown	GameException thrown	PASS
onTurnEnd	1	No operation	onTurnEnd(validChar)	No change to duration	No change to duration	PASS
remove	1	No operation	remove(validChar)	No exception	No exception	PASS
getDuration	1	Returns initial duration	new MarkedEffect()	2	2	PASS
getDuration	2	Returns decremented duration	new MarkedEffect(); onTurnStart (validChar)	1	1	PASS
getType	1	Returns correct status type	getType()	StatusEffectType.MARKED	StatusEffectType.MARKED	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor	1	Default duration	new PoisonEffect()	getDuration() == 3	getDuration() == 3	PASS
applyEffect	1	Apply to valid character	applyEffect(validChar)	No exception	No exception	PASS
applyEffect	2	Apply to null	applyEffect(null)	GameException thrown	GameException thrown	PASS
onTurnStart	1	Deals 5 damage	onTurnStart(char with 20 HP)	char HP decreases by 5	char HP decreases by 5	PASS
onTurnStart	2	Decrements duration	onTurnStart(char)	getDuration() decreases by 1	getDuration() decreases by 1	PASS
onTurnStart	3	Null character	onTurnStart(null)	GameException thrown	GameException thrown	PASS

onTurnEnd	1	No action	onTurnEnd(char)	No effect on char	No effect on char	PASS
remove	1	Remove from valid character	remove(validChar)	No exception	No exception	PASS
remove	2	Remove from null	remove(null)	GameException thrown	GameException thrown	PASS
getDuration	1	Initial duration after construction	new PoisonEffect()	3	3	PASS
getDuration	2	After all decrements	new PoisonEffect(), then onTurnStart() x3	0	0	PASS
getType	1	Returns correct type	getType()	StatusEffectType.POISONED	StatusEffectType. POISONED	PASS
toString	1	Correct format for logging	toString()	Contains "Poisoned (X turns left)"	Contains "Poisoned (X turns left)"	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor	1	Initial state after creation	new ShieldEffect()	getDuration() == 1	getDuration() == 1	PASS
applyEffect	1	Apply to valid character	applyEffect(validChar)	No exception	No exception	PASS
applyEffect	2	Apply to null	applyEffect(null)	GameException thrown	GameException thrown	PASS
onTurnStart	1	No change on turn start	onTurnStart(char)	getDuration() unchanged	getDuration() unchanged	PASS
onTurnEnd	1	No change on turn end	onTurnEnd(char)	getDuration() unchanged	getDuration() unchanged	PASS
remove	1	Mark as used	remove(validChar)	getDuration() == 0	getDuration() == 0	PASS
remove	2	Remove from null	remove(null)	No exception	No exception	PASS
absorb	1	Absorb less than BLOCK_AMOUNT	absorb(10)	returns 0	returns 0	PASS
absorb	2	Absorb equal to BLOCK_AMOUNT	absorb(15)	returns 0	returns 0	PASS
absorb	3	Absorb more than BLOCK_AMOUN	absorb(25)	returns 10	returns 10	PASS
absorb	4	Second use after being used	absorb(any) after used	returns original value	returns original value	PASS
getDuration	1	After absorb()	getDuration() == 0	getDuration() == 0	getDuration() == 0	PASS

getType	1	Returns correct type	getType()	StatusEffectType.SHIELDED	StatusEffectType.SHIELDEI	PASS

Method	⊖ #	Test Description	Test Input	Expected Output	Actual Output	Status
Constructor	1	Initial state	new StunEffect()	getDuration()==1	getDuration()==1	PASS
applyEffect	1	Apply to valid character	applyEffect(validChar)	validChar.isStunned()==true	validChar.isStunned()==tru	PASS
applyEffect	2	Apply to null	applyEffect(null)	GameException thrown	GameException thrown	PASS
onTurnStart	1	Decrements duration	onTurnStart(char)	getDuration()==0	getDuration()==0	PASS
onTurnStart	2	Clears stun when expired	onTurnStart(char with stunned=true)	char.isStunned()==false	char.isStunned()==false	PASS
onTurnStart	3	Null safety	onTurnStart(null)	GameException thrown	GameException thrown	PASS
onTurnEnd	1	No-op	onTurnEnd(char)	no state change	no state change	PASS
remove	1	Removes stun	remove(char with stunned=true)	char.isStunned()==false	char.isStunned()==false	PASS
remove	2	Null safety	remove(null)	GameException thrown	GameException thrown	PASS
getDuration	1	After decrement	getDuration()	matches internal counter	matches internal counter	PASS
getType	1	Type is STUNNED	getType()	StatusEffectType.STUNNED	StatusEffectType.STUNNEI	PASS
toString	1	Correct format	toString()	contains "Stunned"	contains "Stunned"	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
getAbilityByName	1	Valid Mage ability returns correct object	"Arcane Bolt"	Ability object with name "Arcane Bolt"	Ability{name='Arcane Bolt', ...}	PASS
getAbilityByName	2	Valid Warrior ability returns correct object	"Bloodlust"	Ability object with name "Bloodlust"	Ability{name='Bloodlust', ...}	PASS
getAbilityByName	3	Nonexistent ability returns null	"Fake Ability"	null	null	PASS

getAbilityByName	4	Case-sensitive: wrong case returns null	"arcane bolt"	null	null	PASS
getAbilityByName	5	Valid Rogue ability returns correct object	"Shiv"	Ability object with name "Shiv"	Ability{name='Shiv', ...}	PASS
getAllAbilities	1	Returns unmodifiable map	N/A	Map instance (unmodifiable, throws UnsupportedOperationException on put())	UnsupportedOperationException thrown when put attempted	PASS
getAllAbilities	2	Map contains all expected keys	N/A	Map contains 15 keys (all defined abilities)	15 keys present: [Arcane Bolt, Arcane Blast, Mana Channel, Lesser Heal, Arcane Shield, Shiv, Backstab, Focus, Smoke Bomb, Sneak Attack, Cleave, Shield Bash, Ironclad Defense, Bloodlust, Rallying Cry]	PASS
getAllAbilities	3	Abilities in map are not null	N/A	All map values are non-null Ability objects	All 15 values checked: All non-null	PASS
getAllAbilities	4	Returns same map instance on repeated call	N/A	Reference equality (map1 == map2)	TRUE	PASS
getAllAbilities	5	Returned map is not null	N/A	Map != null	Map instance returned	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
decideMove	1	RandomStrategy: Valid input returns a Move	"botCharacter: Mage (40 EP), opponentCharacter: Warrior (50 HP)"	Returns non-null Move object	Move{name='Arcane Bolt', ...}	PASS
decideMove	2	AggressiveStrategy: Selects most damaging move	"botCharacter: Warrior (full EP, has Cleave & Shield Bash), opponentCharacter: Rogue"	Returns Move corresponding to highest damage ability	Move{name='Shield Bash', damage=35, ...}	PASS
decideMove	3	No valid moves (bot has 0 EP)	"botCharacter: Mage (0 EP), opponentCharacter: Rogue"	Throws GameException	GameException thrown	PASS
decideMove	4	Null botCharacter	null botCharacter	Throws GameException	GameException thrown	PASS
decideMove	5	Null opponentCharacter	botCharacter: Warrior (50 EP), null opponentCharacter	Throws GameException	GameException thrown	PASS

decideMove	6	Move targets correct opponent	"botCharacter: Rogue, opponentCharacter: Mage"	Returned Move targets opponentCharacter	Move targets Character {name='Mage'}	PASS
decideMove	7	DefensiveStrategy: Bot low HP chooses heal	"botCharacter: Mage (HP=10, EP=20, has Lesser Heal), opponentCharacter: Rogue"	Returns Move of heal ability	Move{name='Lesser Heal', effect=HEAL, ...}	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
private Constants()	1	Instantiation attempt throws exception	N/A	Throws UnsupportedOperationException	UnsupportedOperationException thrown	PASS
NUM_ABILITIES_PER_CHAR	1	Value is correct	N/A	3	3	PASS
MAX_CHARACTERS_PER_PLAYER	1	Value is correct	N/A	6	6	PASS
BASE_HP	1	Value is correct	N/A	100	100	PASS
BASE_EP	1	Value is correct	N/A	50	50	PASS
MAX_EP_COST	1	Value is correct	N/A	50	50	PASS
MAX_EFFECT_VALUE	1	Value is correct	N/A	100	100	PASS
MAX_STATUS_EFFECTS	1	Value is correct	N/A	5	5	PASS
DEFEND_EP_COST	1	Value is correct	N/A	5	5	PASS
RECHARGE_EP_GAIN	1	Value is correct	N/A	5	5	PASS
ROUND_EP_REGEN	1	Value is correct	N/A	5	5	PASS
WINS_PER_REWARD	1	Value is correct	N/A	3	3	PASS
ARCANE_BOLT_DMG	1	Value is correct	N/A	20	20	PASS
ARCANE_BOLT_COST	1	Value is correct	N/A	5	5	PASS
ARCANE_BLAST_DMG	1	Value is correct	N/A	65	65	PASS
ARCANE_BLAST_COST	1	Value is correct	N/A	30	30	PASS

	MANA_CHANNEL_GAIN	1	Value is correct	N/A	15	15	PASS
	MANA_CHANNEL_COST	1	Value is correct	N/A	0	0	PASS
	LESSER_HEAL_HP	1	Value is correct	N/A	40	40	PASS
	LESSER_HEAL_COST	1	Value is correct	N/A	15	15	PASS
	ARCANE_SHIELD_COST	1	Value is correct	N/A	12	12	PASS
	SHIV_DMG	1	Value is correct	N/A	20	20	PASS
	SHIV_COST	1	Value is correct	N/A	5	5	PASS
	BACKSTAB_DMG	1	Value is correct	N/A	35	35	PASS
	BACKSTAB_COST	1	Value is correct	N/A	15	15	PASS
	FOCUS_GAIN	1	Value is correct	N/A	10	10	PASS
	FOCUS_COST	1	Value is correct	N/A	0	0	PASS
	SMOKE_BOMB_COST	1	Value is correct	N/A	15	15	PASS
	SNEAK_ATTACK_DMG	1	Value is correct	N/A	45	45	PASS
	SNEAK_ATTACK_COST	1	Value is correct	N/A	25	25	PASS
	CLEAVE_DMG	1	Value is correct	N/A	20	20	PASS
	CLEAVE_COST	1	Value is correct	N/A	5	5	PASS
	SHIELD_BASH_DMG	1	Value is correct	N/A	35	35	PASS
	SHIELD_BASH_COST	1	Value is correct	N/A	15	15	PASS
	IRONCLAD_DEFENSE_COST	1	Value is correct	N/A	15	15	PASS
	BLOODLUST_HP	1	Value is correct	N/A	30	30	PASS
	BLOODLUST_COST	1	Value is correct	N/A	12	12	PASS
	RALLYING_CRY_GAIN	1	Value is correct	N/A	12	12	PASS

RALLYING_CRY_COST	1	Value is correct	N/A	0	0	PASS
SAVE_FILE_PATH	1	Value is correct	N/A	"ff_tactics_save.dat"	"ff_tactics_save.dat"	PASS
HALL_OF_FAME_SAVE_PATH	1	Value is correct	N/A	"save/hall_of_fame.dat"	"save/hall_of_fame.dat"	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
private InputValidator()	1	Instantiation attempt throws exception	N/A	Throws AssertionError	AssertionError thrown	PASS
requireNonNull	1	Accepts non-null object	"obj: new Object(), paramName: 'param'"	No exception thrown	No exception thrown	PASS
requireNonNull	2	Throws if null	"obj: null, paramName: 'value'"	Throws GameException		PASS
requireNonBlank	1	Accepts non-blank string	"text: 'hello', paramName: 'field'"	No exception thrown	No exception thrown	PASS
requireNonBlank	2	Throws on blank string	"text: ' ', paramName: 'field'"	Throws GameException		PASS
requireNonBlank	3	Throws on null	"text: null, paramName: 'field'"	Throws GameException	GameException thrown	PASS
requirePositive	1	Accepts positive integer	"value: 3, paramName: 'amount'"	No exception thrown	No exception thrown	PASS
requirePositive	2	Throws on zero	"value: 0, paramName: 'qty'"	Throws GameException	GameException thrown	PASS
requirePositive	3	Throws on negative	"value: -2, paramName: 'score'"	Throws GameException		PASS
requirePositiveOrZero	1	Accepts zero	"value: 0, fieldName: 'score'"	No exception thrown	No exception thrown	PASS
requirePositiveOrZero	2	Accepts positive	"value: 2, fieldName: 'score'"	No exception thrown	No exception thrown	PASS
requirePositiveOrZero	3	Throws on negative	"value: -1, fieldName: 'score'"	Throws GameException	GameException thrown	PASS
requireRange	1	Accepts value within range	"value: 5, min: 3, max: 7, paramName: 'level'"	No exception thrown	No exception thrown	PASS
requireRange	2	Throws if below range	"value: 2, min: 3, max: 7, paramName: 'level'"	Throws GameException	GameException thrown	PASS
requireRange	3	Throws if above range	"value: 10, min: 3, max: 7, paramName: 'level'"	Throws GameException		PASS

requireSize (int,int,String)	1	Accepts matching size	"size: 3, expectedSize: 3, message: 'must be 3'"	No exception thrown	No exception thrown	PASS
requireSize (int,int,String)	2	Throws if not matching	"size: 2, expectedSize: 3, message: 'must be 3'"	Throws GameException		PASS
requireSize (int,int,int,String)	1	Accepts size in range	"size: 4, minSize: 2, maxSize: 5, message: 'must be 2-5'"	No exception thrown	No exception thrown	PASS
requireSize (int,int,int,String)	2	Throws if below range	"size: 1, minSize: 2, maxSize: 5, message: 'must be 2-5'"	Throws GameException		PASS
requireSize (int,int,int,String)	3	Throws if above range	"size: 7, minSize: 2, maxSize: 5, message: 'must be 2-5'"	Throws GameException	GameException thrown	PASS
requireNonEmpty	1	Accepts non-empty string	"input: 'A', fieldName: 'Name'"	No exception thrown	No exception thrown	PASS
requireNonEmpty	2	Throws on null	"input: null, fieldName: 'Name'"	Throws GameException	GameException thrown	PASS
requireNonEmpty	3	Throws on blank	"input: ' ', fieldName: 'Name'"	Throws GameException		PASS
requireUniqueName	1	Accepts unique name	"input: 'Bob', existingPlayers: [Player('Alice')]"	No exception thrown	No exception thrown	PASS
requireUniqueName	2	Throws on duplicate (case-insensitive)	"input: 'alice', existingPlayers: [Player('Alice')]"	Throws GameException		PASS
requireDistinct(Collection)	1	Accepts distinct items	"items:, message: 'dup'"	No exception thrown	No exception thrown	PASS
requireDistinct(Collection)	2	Throws on duplicates	"items:, message: 'dup'"	Throws GameException		PASS
requireDistinct(Object[])	1	Accepts distinct array	"values: ['a','b','c'], message: 'dup'"	No exception thrown	No exception thrown	PASS
requireDistinct(Object[])	2	Throws on duplicate array	"values: ['a','b','a'], message: 'dup'"	Throws GameException		PASS

Method	#	#	Test Description	Test Input	Expected Output	Actual Output	Status
GameException(String)		1	Creates exception with valid message	"message: 'Error occurred'"	GameException created with message 'Error occurred'	GameException created, getMessage() == 'Error occurred'	PASS
GameException(String)		2	Throws IllegalArgumentException if message is null	"message: null"	Throws IllegalArgumentException	IllegalArgumentException thrown	PASS

GameException(String)	3	Throws IllegalArgumentException if message is blank	"message: ' '"	Throws IllegalArgumentException	IllegalArgumentException thrown	PASS
GameException(String, Throwable)	1	Creates exception with valid message and cause	"message: 'Failure', cause: Exception('X')"	GameException created, message == 'Failure', cause == Exception('X')	GameException created, getMessage() == 'Failure', getCause() instanceof Exception	PASS
GameException(String, Throwable)	2	Throws IllegalArgumentException if message is null	"message: null, cause: Exception ('X')"	Throws IllegalArgumentException	IllegalArgumentException thrown	PASS
GameException(String, Throwable)	3	Throws IllegalArgumentException if message is blank	"message: '', cause: null"	Throws IllegalArgumentException	IllegalArgumentException thrown	PASS
GameException(String, Throwable)	4	Accepts null as cause	"message: 'Err', cause: null"	GameException created, message == 'Err', cause == null	GameException created, getMessage() == 'Err', getCause() == null	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
generate	1	Generates valid character with random race/class	"name: 'AI_1'"	Character object with name 'AI_1', valid RaceType and ClassType	Character{name='AI_1', race=RaceType.HUMAN, class=ClassType.MAGE, abilities.size()=3}	PASS
generate	2	Abilities count includes race bonus	"name: 'AI_2', using race with 1 bonus slot"	Abilities.size() == NUM_ABILITIES_PER_CHAR + 1	Character{name='AI_2', abilities.size()=4}	PASS
generate	3	Race and class are randomly distributed (many runs)	"generate() called 100 times with different names"	Various races and classes observed	Observed all RaceTypes and ClassTypes after 100 runs	PASS
generate	4	Throws GameException for blank name	"name: ''"	Throws GameException	GameException thrown	PASS
generate	5	Throws GameException for null name	"name: null"	Throws GameException	GameException thrown	PASS
generate	6	Throws/fails gracefully if no races available	"RaceService mocked to return empty list"	Throws GameException	GameException thrown	PASS
generate	7	Throws/fails gracefully if classService returns not enough abilities	"ClassService returns empty list"	Throws GameException or invalid Character	GameException thrown	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
SimpleBot(Random)	1	Creates SimpleBot with valid Random	"random: new Random()"	SimpleBot object created	SimpleBot instance created	PASS
SimpleBot(Random)	2	Throws GameException for null Random	"random: null"	Throws GameException	GameException thrown	PASS
decideMove	1	Returns random AbilityMove if bot has EP and abilities	"botCharacter: 30 EP, 2 abilities (5 EP, 10 EP), no item; opponentCharacter: any"	Returns AbilityMove for one of available abilities	Returned AbilityMove for ability with epCost <= 30	PASS
decideMove	2	Returns Recharge if bot has no usable abilities or item	"botCharacter: 0 EP, abilities all cost >0, no item; opponentCharacter: any"	Returns Recharge	Returned Recharge	PASS
decideMove	3	Returns ItemMove if SingleUseItem equipped	"botCharacter: 20 EP, has SingleUseItem equipped, abilities cost > 20; opponentCharacter: any"	Returns ItemMove for equipped SingleUseItem	Returned ItemMove for SingleUseItem	PASS
decideMove	4	Throws GameException if botCharacter is null	"botCharacter: null; opponentCharacter: any"	Throws GameException	GameException thrown	PASS
decideMove	5	Throws GameException if opponentCharacter is null	"botCharacter: valid; opponentCharacter: null"	Throws GameException	GameException thrown	PASS
decideMove	6	Randomly chooses among all possible moves	"botCharacter: has 1 ability (usable) and 1 SingleUseItem; opponentCharacter: any"	Returns either AbilityMove or ItemMove	Observed both move types returned in repeated tests	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
SmartBot(Random)	1	Creates SmartBot with valid Random	"random: new Random()"	SmartBot object created	SmartBot instance created	PASS
SmartBot(Random)	2	Throws GameException for null Random	"random: null"	Throws GameException	GameException thrown	PASS
decideMove	1	Selects healing move when HP is low	"bot: HP = 20/100, has heal ability and EP, opponent: any"	Returns AbilityMove for healing	Returned AbilityMove for healing ability	PASS
decideMove	2	Selects energy move when EP is low	"bot: HP = 80/100, EP = 2, has ENERGY_GAIN ability, opponent: any"	Returns AbilityMove for ENERGY_GAIN	Returned AbilityMove for energy gain ability	PASS

decideMove	3	Selects highest damage move otherwise	"bot: HP = 100/100, EP sufficient, multiple damage abilities, opponent: HP=40"	Returns AbilityMove for highest damage	Returned AbilityMove for ability with highest damage	PASS
decideMove	4	Selects lethal move if it can KO opponent	"bot: has damage ability >= opponent HP, opponent: low HP"	Returns AbilityMove that can KO	Returned AbilityMove for lethal damage ability,P	PASS
decideMove	5	Selects defensive move if available and no better option	"bot: HP=50, EP sufficient, no healing/energy/damage, has defense ability, opponent: any"	Returns AbilityMove for DEFENSE or EVADE	Returned AbilityMove for defense ability	PASS
decideMove	6	Returns Recharge if no moves possible	"bot: 0 EP, no usable items, abilities all cost >0, opponent: any"	Returns Recharge	Returned Recharge	PASS
decideMove	7	Prefers item healing over ability if HP is low and both exist	"bot: HP=10, has SingleUseItem HEAL_HP and heal ability, both usable, opponent: any"	Returns ItemMove for SingleUseItem or AbilityMove	Observed both over repeated runs,P	PASS
decideMove	8	Throws GameException if bot is null	"bot: null, opponent: any"	Throws GameException	GameException thrown	PASS
decideMove	9	Throws GameException if opponent is null	"bot: valid, opponent: null"	Throws GameException	GameException thrown	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
applyEffect	1	Applies effect to valid character	"target: Character with no effect"	Effect applied	Character's status updated, effect applied	PASS
applyEffect	2	Throws GameException if effect cannot be applied	"target: Character already has effect"	Throws GameException	GameException thrown	PASS
applyEffect	3	Throws GameException for null target	"target: null"	Throws GameException	GameException thrown	PASS
onTurnStart	1	Triggers effect at turn start	"target: Character with effect"	Effect's onTurnStart logic triggered	Effect logic executed, character updated	PASS
onTurnStart	2	Throws GameException for null target	"target: null"	Throws GameException	GameException thrown	PASS
onTurnEnd	1	Triggers effect at turn end	"target: Character with effect"	Effect's onTurnEnd logic triggered	Effect logic executed, character updated	PASS
onTurnEnd	2	Throws GameException for null target	"target: null"	Throws GameException	GameException thrown	PASS
remove	1	Removes effect and cleans up	"target: Character with effect"	Effect removed	Character status cleaned, effect removed	PASS

remove		2	Throws GameException for null target		Throws GameException	GameException thrown	PASS
getDuration		1	Returns correct duration for effect	N/A	Returns non-negative integer	Returned 2	PASS
getType		1	Returns correct type for effect	N/A	Returns StatusEffectType enum	Returned StatusEffectType. POISONED	PASS

Method	#	#	Test Description	Test Input	Expected Output	Actual Output	Status
private StatusEffectFactory()		1	Instantiation attempt throws exception	N/A	Throws UnsupportedOperationException	UnsupportedOperationException thrown	PASS
create		1	Creates PoisonEffect for POISONED type	"type: StatusEffectType. POISONED"	Instance of PoisonEffect returned	PoisonEffect instance created	PASS
create		2	Creates StunEffect for STUNNED type	"type: StatusEffectType. STUNNED"	Instance of StunEffect returned	StunEffect instance created	PASS
create		3	Creates EvadeEffect for EVADING type	"type: StatusEffectType. EVADING"	Instance of EvadeEffect returned	EvadeEffect instance created	PASS
create		4	Throws GameException for null type	"type: null"	Throws GameException	GameException thrown	PASS
create		5	Throws GameException for unsupported type	"type: (UNSUPPORTED/unknown)"	Throws GameException	GameException thrown	PASS
create		6	Creates ImmunityEffect for IMMUNITY type	"type: StatusEffectType. IMMUNITY"	Instance of ImmunityEffect returned	ImmunityEffect instance created	PASS
create		7	Creates DefenseUpEffect for DEFENSE_UP type	"type: StatusEffectType. DEFENSE_UP"	Instance of DefenseUpEffect returned	DefenseUpEffect instance created	PASS
create		8	Creates ShieldEffect for SHIELDED type	"type: StatusEffectType. SHIELDED"	Instance of ShieldEffect returned	ShieldEffect instance created	PASS
create		9	Creates MarkedEffect for MARKED type	"type: StatusEffectType. MARKED"	Instance of MarkedEffect returned	MarkedEffect instance created	PASS

Method	#	#	Test Description	Test Input	Expected Output	Actual Output	Status
(private) StatusEffectFactory()		1	Instantiation attempt throws exception	N/A	Throws UnsupportedOperationException	UnsupportedOperationException thrown	PASS

create	1	Creates PoisonEffect for POISONED type	"type: StatusEffectType. POISONED"	Instance of PoisonEffect returned	PoisonEffect instance created	PASS
create	2	Creates StunEffect for STUNNED type	"type: StatusEffectType. STUNNED"	Instance of StunEffect returned	StunEffect instance created	PASS
create	3	Creates EvadeEffect for EVADING type	"type: StatusEffectType. EVADING"	Instance of EvadeEffect returned	EvadeEffect instance created	PASS
create	4	Creates ImmunityEffect for IMMUNITY type	"type: StatusEffectType. IMMUNITY"	Instance of ImmunityEffect returned	ImmunityEffect instance created	PASS
create	5	Creates DefenseUpEffect for DEFENSE_UP type	"type: StatusEffectType. DEFENSE_UP"	Instance of DefenseUpEffect returned	DefenseUpEffect instance created	PASS
create	6	Creates ShieldEffect for SHIELDED type	"type: StatusEffectType. SHIELDED"	Instance of ShieldEffect returned	ShieldEffect instance created	PASS
create	7	Creates MarkedEffect for MARKED type	"type: StatusEffectType. MARKED"	Instance of MarkedEffect returned	MarkedEffect instance created	PASS
create	8	Throws GameException for null type	"type: null"	Throws GameException	GameException thrown	PASS
create	9	Throws GameException for unsupported type	"type: (unsupported)"	Throws GameException	GameException thrown	PASS
(enum) StatusEffectType	1	All expected enum values present	N/A	Enum contains STUNNED, POISONED, DEFENSE_UP, EVADING, IMMUNITY, SHIELDED, MARKED, NONE	All values present	PASS
valueOf	1	Returns correct value for 'STUNNED'	"input: 'STUNNED'"	StatusEffectType.STUNNED	StatusEffectType. STUNNED	PASS
valueOf	2	Returns correct value for 'MARKED'	"input: 'MARKED'"	StatusEffectType.MARKED	StatusEffectType. MARKED	PASS
valueOf	3	Throws IllegalArgumentException for unknown literal	"input: 'UNKNOWN'"	Throws IllegalArgumentException	IllegalArgumentException thrown	PASS
ordinal	1	Correct ordinal for IMMUNITY	"StatusEffectType.IMMUNITY"	Ordinal is 4	4	PASS
toString	1	toString returns name	"StatusEffectType.POISONED"	Returns 'POISONED'	POISONED'	PASS
equals	1	Equality works for same instance	"StatusEffectType.SHIELDED == StatusEffectType.SHIELDED"	TRUE	TRUE	PASS
equals	2	Inequality works for different instance	"StatusEffectType.NONE != StatusEffectType.STUNNED"	TRUE	TRUE	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
GameData()	1	Default constructor initializes empty lists	N/A	allPlayers and hallOfFame are empty	allPlayers.isEmpty()==true, hallOfFame.isEmpty()==true	PASS
GameData(List,List)	1	Full constructor with valid lists	"allPlayers: [p1,p2], hallOfFame: [h1]"	GameData created with lists copied	Lists returned match input, no aliasing	PASS
GameData(List,List)	2	Throws GameException if allPlayers is null	"allPlayers: null, hallOfFame: [h1]"	Throws GameException	GameException thrown	PASS
GameData(List,List)	3	Throws GameException if hallOfFame is null	"allPlayers: [p1], hallOfFame: null"	Throws GameException	GameException thrown	PASS
getAllPlayers	1	Returns unmodifiable list of all players	"GameData with players=[p1]"	List returned, unmodifiable, getAllPlayers().contains(p1)	UnsupportedOperationException on add	PASS
getHallOfFame	1	Returns unmodifiable list of hall of fame	"GameData with hallOfFame=[h1]"	List returned, unmodifiable, getHallOfFame().contains(h1)	UnsupportedOperationException on clear	PASS
setAllPlayers	1	Sets all players to a new list	"players: [p2]"	allPlayers updated	getAllPlayers().contains(p2)	PASS
setAllPlayers	2	Throws GameException if players is null	"players: null"	Throws GameException	GameException thrown	PASS
setHallOfFame	1	Sets hall of fame to a new list	"entries: [h2]"	hallOfFame updated	getHallOfFame().contains(h2)	PASS
setHallOfFame	2	Throws GameException if entries is null	"entries: null"	Throws GameException	GameException thrown	PASS
readObject	1	Initializes null lists after deserialization	"serialized GameData with null lists"	allPlayers and hallOfFame initialized as empty	allPlayers!=null, hallOfFame!=null, both empty	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
HallOfFameData()	1	Creates empty HallOfFameData instance	N/A	HallOfFameData created, players and characters empty	HallOfFameData created, players.size()==0, characters.size()==0	PASS
HallOfFameData(List, List)	1	Creates HallOfFameData with valid lists	"players: [p1], characters: [c1]"	HallOfFameData created with correct data	Data correct, players.size()==1, characters.size()==1	PASS

HallOfFameData(List, List)	2	Throws GameException if players list is null	"players: null, characters: [c1]"	Throws GameException	GameException thrown	PASS
HallOfFameData(List, List)	3	Throws GameException if characters list is null	"players: [p1], characters: null"	Throws GameException	GameException thrown	PASS
getPlayers	1	Returns unmodifiable list of players	N/A	Unmodifiable list returned	List returned, unmodifiable	PASS
getCharacters	1	Returns unmodifiable list of characters	N/A	Unmodifiable list returned	List returned, unmodifiable	PASS
setPlayers	1	Updates players with valid list	"list: [p2, p3]"	Players updated, getPlayers().size()==2	Players updated, getPlayers().size()==2	PASS
setPlayers	2	Throws GameException if null	"list: null"	Throws GameException	GameException thrown	PASS
setCharacters	1	Updates characters with valid list	"list: [c2]"	Characters updated, getCharacters().size()==1	Characters updated, getCharacters().size()==1	PASS
setCharacters	2	Throws GameException if null	"list: null"	Throws GameException	GameException thrown	PASS
readObject	1	Initializes lists if null after deserialization	"in: mock object stream (sets fields to null)"	Fields players, characters initialized as empty	Fields initialized as empty lists	PASS

Method	#	Test Description	Test Input	Expected Output	Actual Output	Status
saveGame	1	Saves valid GameData to file	"gameData: valid GameData"	GameData saved, file created	GameData saved, file exists	PASS
saveGame	2	Throws GameException on IO error	"gameData: valid, but path unwritable"	Throws GameException	GameException thrown	PASS
loadGame	1	Loads GameData from file when present	N/A	Returns GameData from file	GameData loaded, data correct	PASS
loadGame	2	Returns empty GameData if file not found	N/A	Returns new GameData	GameData created, empty	PASS
loadGame	3	Throws GameException on corrupt file	N/A	Throws GameException	GameException thrown	PASS
saveHallOfFame	1	Saves valid HallOfFameData	"data: valid HallOfFameData"	HallOfFameData saved, file created	File created, HallOfFameData saved	PASS
saveHallOfFame	2	Throws GameException on IO error	"data: valid, but path unwritable"	Throws GameException	GameException thrown	PASS

loadHallOfFame	1	Loads HallOfFameData from file if present	N/A	Returns HallOfFameData from file	HallOfFameData loaded, data correct	PASS
loadHallOfFame	2	Returns empty HallOfFameData if file not found	N/A	Returns new HallOfFameData	HallOfFameData created, empty	PASS
loadHallOfFame	3	Handles legacy List<HallOfFameEntry> format	N/A	Converts list to HallOfFameData	saves upgraded,HallOfFam	PASS

Method	# #	Test Description	Test Input	Expected Output	Actual Output	Status
Main()	1	Instantiation attempt throws exception	N/A	Private constructor prevents instantiation	Instantiation prevented (private constructor)	PASS
main	1	Launches application via SwingUtilities	"args: empty or valid"	SceneManager instance created on EDT	SceneManager started on EDT	PASS
shutdown	1	Shutdown completes gracefully	N/A	No exceptions	Shutdown completed gracefully	PASS