Sheet: Sheet1

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
POSITION_010	PositionTest	testEqualsReturnsFalseForDifferentClass	Position	equals	Verify equals returns false for different class	position.equals("not a position")	Returns false	Pass
POSITION_011	PositionTest	testHashCodeConsistency	Position	hashCode	Verify hashCode is consistent for equal positions	Position (3,5) and Position (3,5)	Both have same hashCode	Pass
PUZZLE_001	PuzzleTest	testConstructorSetsPuzzleId	Puzzle	Puzzle (constructor)	Verify constructor sets puzzleId	puzzleId="P001", type="MAZE", difficulty="EASY", title="Test Maze", description, data	getPuzzleId() returns P001	Pass
PUZZLE_002	PuzzleTest	testConstructorSetsPuzzleType	Puzzle	Puzzle (constructor)	Verify constructor sets puzzleType	Full constructor parameters	getPuzzleType() returns MAZE	Pass
PUZZLE_003	PuzzleTest	testConstructorSetsDifficulty	Puzzle	Puzzle (constructor)	Verify constructor sets difficulty	Full constructor parameters	getDifficulty() returns EASY	Pass
PUZZLE_004	PuzzleTest	testConstructorSetsTitle	Puzzle	Puzzle (constructor)	Verify constructor sets title	Full constructor parameters	getTitle() returns Test Maze	Pass
PUZZLE_005	PuzzleTest	testConstructorSetsDescription	Puzzle	Puzzle (constructor)	Verify constructor sets description	Full constructor parameters	getDescription() returns A simple test maze	Pass
PUZZLE_006	PuzzleTest	testConstructorSetsData	Puzzle	Puzzle (constructor)	Verify constructor sets data	Full constructor parameters with data Map	getData() equals input Map	Pass
PUZZLE_007	PuzzleTest	testSetPuzzleId	Puzzle	setPuzzleId	Verify setPuzzleId updates puzzleId	setPuzzleId(P002)	getPuzzleId() returns P002	Pass
PUZZLE_008	PuzzleTest	testSetPuzzleType	Puzzle	setPuzzleType	Verify setPuzzleType updates puzzleType	setPuzzleType(MATCHING)	getPuzzleType() returns MATCHING	Pass
PUZZLE_009	PuzzleTest	testSetDifficulty	Puzzle	setDifficulty	Verify setDifficulty updates difficulty	setDifficulty(HARD)	getDifficulty() returns HARD	Pass
PUZZLE_010	PuzzleTest	testSetTitle	Puzzle	setTitle	Verify setTitle updates title	setTitle(New Title)	getTitle() returns New Title	Pass
PUZZLE_011	PuzzleTest	testSetDescription	Puzzle	setDescription	Verify setDescription updates description	setDescription(New Description)	getDescription() returns New Description	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
PUZZLE_012	PuzzleTest	testSetData	Puzzle	setData	Verify setData updates data	setData(newDataMap)	getData() equals newData	Pass
PUZZLE_013	PuzzleTest	testToStringContainsPuzzleId	Puzzle	toString	Verify toString contains puzzleId	toString()	Contains P001	Pass
PUZZLE_014	PuzzleTest	testToStringContainsTitle	Puzzle	toString	Verify toString contains title	toString()	Contains Test Maze	Pass
USERPROG_001	UserProgressTest	testDefaultConstructorInitializesCompletedPuzzles	UserProgress	UserProgress (constructor)	Verify default constructor initializes completedPuzzles list	Default constructor	getCompletedPuzzles() is non-null	Pass
USERPROG_002	UserProgressTest	testDefaultConstructorInitializesPuzzleScores	UserProgress	UserProgress (constructor)	Verify default constructor initializes puzzleScores map	Default constructor	getPuzzleScores() is non-null	Pass
USERPROG_003	UserProgressTest	testDefaultConstructorSetsTotalScoreToZero	UserProgress	UserProgress (constructor)	Verify default constructor sets totalScore to zero	Default constructor	getTotalScore() equals 0	Pass
USERPROG_004	UserProgressTest	testDefaultConstructorCreatesEmptyCompletedPuzzles	UserProgress	UserProgress (constructor)	Verify default constructor creates empty completedPuzzles	Default constructor	getCompletedPuzzles().isEmpty() is true	Pass
USERPROG_005	UserProgressTest	testConstructorWithUserIdSetsUserId	UserProgress	UserProgress (constructor)	Verify constructor with userId sets userId	userId=user1	getUserId() returns user1	Pass
USERPROG_006	UserProgressTest	testConstructorWithUserIdInitializesCompletedPuzzles	UserProgress	UserProgress (constructor)	Verify constructor with userId initializes completedPuzzles	userId=user1	getCompletedPuzzles() is non-null	Pass
USERPROG_007	UserProgressTest	testAddCompletedPuzzleMarksPuzzleAsCompleted	UserProgress	addCompletedPuzzle	Verify addCompletedPuzzle marks puzzle as completed	puzzleId=puzzle1, score=100	isPuzzleCompleted(puzzle1) returns true	Pass
USERPROG_008	UserProgressTest	testAddCompletedPuzzleUpdatesTotalScore	UserProgress	addCompletedPuzzle	Verify addCompletedPuzzle updates totalScore	puzzleId=puzzle1, score=100	getTotalScore() equals 100	Pass
USERPROG_009	UserProgressTest	testAddCompletedPuzzleIncrementsCount	UserProgress	addCompletedPuzzle	Verify addCompletedPuzzle increments completed count	puzzleId=puzzle1, score=100	getCompletedCount() equals 1	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
USERPROG_010	UserProgressTest	testAddCompletedPuzzleStoresScore	UserProgress	addCompletedPuzzle	Verify addCompletedPuzzle stores score in map	puzzleId=puzzle1, score=100	getPuzzleScores()[puzzle1] equals 100	Pass
USERPROG_011	UserProgressTest	testAddCompletedPuzzleDoesNotAddDuplicates	UserProgress	addCompletedPuzzle	Verify addCompletedPuzzle rejects duplicate puzzles	Add puzzle1 twice with scores 100 and 150	getCompletedCount() equals 1	Pass
USERPROG_012	UserProgressTest	testAddCompletedPuzzleDoesNotUpdateScoreForDuplicate	UserProgress	addCompletedPuzzle	Verify addCompletedPuzzle doesn't update score for duplicates	Add puzzle1 twice with scores 100 and 150	getTotalScore() equals 100 (not 250)	Pass
USERPROG_013	UserProgressTest	testAddMultipleCompletedPuzzlesIncrementsCount	UserProgress	addCompletedPuzzle	Verify multiple addCompletedPuzzle calls increment count	Add puzzle1 (100), puzzle2 (75), puzzle3 (50)	getCompletedCount() equals 3	Pass
USERPROG_014	UserProgressTest	testAddMultipleCompletedPuzzlesUpdatesTotalScore	UserProgress	addCompletedPuzzle	Verify multiple addCompletedPuzzle calls update totalScore	Add puzzle1 (100), puzzle2 (75), puzzle3 (50)	getTotalScore() equals 225	Pass
USERPROG_015	UserProgressTest	testIsPuzzleCompletedReturnsFalseForUncompletedPuzzle	UserProgress	isPuzzleCompleted	Verify isPuzzleCompleted returns false for uncompleted puzzle	puzzleId=puzzle1 (not completed)	Returns false	Pass
USERPROG_016	UserProgressTest	testSaveGameStateSetsCurrentPuzzleId	UserProgress	saveGameState	Verify saveGameState sets currentPuzzleId	puzzleId=puzzle1, gameState with level=5	getCurrentPuzzleId() equals puzzle1	Pass
USERPROG_017	UserProgressTest	testSaveGameStateSavesGameState	UserProgress	saveGameState	Verify saveGameState saves gameState map	puzzleId=puzzle1, gameState with level=5	getGameState() equals saved gameState	Pass
USERPROG_018	UserProgressTest	testSaveGameStateSetsHasGameInProgress	UserProgress	saveGameState	Verify saveGameState sets hasGameInProgress to true	puzzleId=puzzle1, gameState with level=5	hasGameInProgress() returns true	Pass
USERPROG_019	UserProgressTest	testClearGameStateClearsCurrentPuzzleId	UserProgress	clearGameState	Verify clearGameState clears currentPuzzleId	Save game state, then clear	getCurrentPuzzleId() returns null	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
USERPROG_020	UserProgressTest	testClearGameStateClearsGameState	UserProgress	clearGameState	Verify clearGameState clears gameState map	Save game state, then clear	getGameState() returns null	Pass
USERPROG_021	UserProgressTest	testClearGameStateClearsHasGameInProgress	UserProgress	clearGameState	Verify clearGameState sets hasGameInProgress to false	Save game state, then clear	hasGameInProgress() returns false	Pass
USERPROG_022	UserProgressTest	testHasGameInProgressReturnsFalseWhenNoPuzzleId	UserProgress	hasGameInProgress	Verify hasGameInProgress returns false when no puzzleId	Set gameState but currentPuzzleId=null	Returns false	Pass
USERPROG_023	UserProgressTest	testHasGameInProgressReturnsFalseWhenNoGameState	UserProgress	hasGameInProgress	Verify hasGameInProgress returns false when no gameState	Set currentPuzzleId but gameState=null	Returns false	Pass
USERPROG_024	UserProgressTest	testGetCompletedCountReturnsCorrectCount	UserProgress	getCompletedCount	Verify getCompletedCount returns correct count	Add puzzle1 and puzzle2	Returns 2	Pass
USERPROG_025	UserProgressTest	testSetUserId	UserProgress	setUserId	Verify setUserId updates userId	setUserId(newUser)	getUserId() returns newUser	Pass
USERPROG_026	UserProgressTest	testSetTotalScore	UserProgress	setTotalScore	Verify setTotalScore updates totalScore	setTotalScore(500)	getTotalScore() returns 500	Pass
WORDPUZZ_001	WordPuzzleGameTest	testInitializeSetsUpGame	WordPuzzleGame	initialize	Verify initialize sets up game properly	puzzleData with prompt and answer	isGameOver() returns false	Pass
WORDPUZZ_002	WordPuzzleGameTest	testInitializeDoesNotSetGameOverState	WordPuzzleGame	initialize	Verify initialize doesn't set game over state	puzzleData with prompt and answer	isGameOver() returns false	Pass
WORDPUZZ_003	WordPuzzleGameTest	testGetGameTypeReturnsRiddle	WordPuzzleGame	getGameType	Verify getGameType returns RIDDLE	setPuzzleType(RIDDLE)	Returns RIDDLE	Pass
WORDPUZZ_004	WordPuzzleGameTest	testSetPuzzleTypeSetsCipher	WordPuzzleGame	setPuzzleType	Verify setPuzzleType sets CIPHER	setPuzzleType(CIPHER)	getGameType() returns CIPHER	Pass
WORDPUZZ_005	WordPuzzleGameTest	testSetPuzzleTypeSetsAnagram	WordPuzzleGame	setPuzzleType	Verify setPuzzleType sets ANAGRAM	setPuzzleType(ANAGRAM)	getGameType() returns ANAGRAM	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
WORDPUZZ_006	WordPuzzleGame Test	testProcessInput WithCorrectAnsw erReturnsTrue	WordPuzzleGame	processInput	Verify processInput returns true for correct answer	input=KEYBOAR D (correct answer)	Returns true	Pass
WORDPUZZ_007	WordPuzzleGame Test	testProcessInput WithCorrectAnsw erSetsGameOver	WordPuzzleGame	processInput	Verify correct answer sets game over	input=KEYBOAR D (correct answer)	isGameOver() returns true	Pass
WORDPUZZ_008	WordPuzzleGame Test	testProcessInputI sCaseInsensitive	WordPuzzleGame	processInput	Verify processInput is case-insensitive	input=keyboard (lowercase)	Returns true	Pass
WORDPUZZ_009	WordPuzzleGame Test	testProcessInput WithLeadingWhite space	WordPuzzleGame	processInput	Verify processInput handles whitespace	input=" KEYBOARD "	Returns true	Pass
WORDPUZZ_010	WordPuzzleGame Test	testProcessInput WithIncorrectAns werReturnsTrue	WordPuzzleGame	processInput	Verify processInput returns true for incorrect answer	input=MOUSE (incorrect)	Returns true	Pass
WORDPUZZ_011	WordPuzzleGame Test	testProcessInput WithPartialAnswer	WordPuzzleGame	processInput	Verify processInput handles partial answer	input=KEY (partial)	Returns true	Pass
WORDPUZZ_012	WordPuzzleGame Test	testProcessInput WithEmptyString	WordPuzzleGame	processInput	Verify processInput handles empty string	input=empty string	Returns true	Pass
WORDPUZZ_013	WordPuzzleGame Test	testProcessInput WithHintComman dReturnsTrue	WordPuzzleGame	processInput	Verify processInput returns true for HINT command	input=HINT	Returns true	Pass
WORDPUZZ_014	WordPuzzleGame Test	testProcessInputH intCommandIsCa seInsensitive	WordPuzzleGame	processInput	Verify HINT command is case-insensitive	input=hint (lowercase)	Returns true	Pass
WORDPUZZ_015	WordPuzzleGame Test	testProcessInputH intCommandWith Whitespace	WordPuzzleGame	processInput	Verify HINT command handles whitespace	input=" HINT "	Returns true	Pass
WORDPUZZ_016	WordPuzzleGame Test	testIsGameOverR eturnsFalseInitiall y	WordPuzzleGame	isGameOver	Verify isGameOver returns false initially	Initialize game	Returns false	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
WORDPUZZ_017	WordPuzzleGameTest	testIsGameOverAfterCorrectAnswer	WordPuzzleGame	isGameOver	Verify isGameOver returns true after correct answer	input=KEYBOARD	Returns true	Pass
WORDPUZZ_018	WordPuzzleGameTest	testIsGameOverAfterMaxAttempts	WordPuzzleGame	isGameOver	Verify isGameOver returns true after max attempts	Input WRONG1, WRONG2, WRONG3	Returns true	Pass
WORDPUZZ_019	WordPuzzleGameTest	testGetGameStateReturnsNonNull	WordPuzzleGame	getGameState	Verify getGameState returns non-null	Call getGameState()	Non-null Map	Pass
WORDPUZZ_020	WordPuzzleGameTest	testGetGameStateContainsPuzzleType	WordPuzzleGame	getGameState	Verify getGameState contains puzzleType	Call getGameState()	State contains puzzleType key	Pass
WORDPUZZ_021	WordPuzzleGameTest	testGetGameStateContainsPrompt	WordPuzzleGame	getGameState	Verify getGameState contains prompt	Call getGameState()	State contains prompt key	Pass
WORDPUZZ_022	WordPuzzleGameTest	testGetGameStateContainsCategory	WordPuzzleGame	getGameState	Verify getGameState contains category	Call getGameState()	State contains category key	Pass
WORDPUZZ_023	WordPuzzleGameTest	testGetGameStateContainsAttemptsUsed	WordPuzzleGame	getGameState	Verify getGameState contains attemptsUsed	Call getGameState()	State contains attemptsUsed key	Pass
WORDPUZZ_024	WordPuzzleGameTest	testGetGameStateContainsMaxAttempts	WordPuzzleGame	getGameState	Verify getGameState contains maxAttempts	Call getGameState()	State contains maxAttempts key	Pass
WORDPUZZ_025	WordPuzzleGameTest	testGetGameStateContainsRemainingAttempts	WordPuzzleGame	getGameState	Verify getGameState contains remainingAttempts	Call getGameState()	State contains remainingAttempts key	Pass
WORDPUZZ_026	WordPuzzleGameTest	testGetGameStateContainsGuesses	WordPuzzleGame	getGameState	Verify getGameState contains guesses list	Call getGameState()	State contains guesses key	Pass
WORDPUZZ_027	WordPuzzleGameTest	testGetGameStateContainsRevealedHints	WordPuzzleGame	getGameState	Verify getGameState contains revealedHints	Call getGameState()	State contains revealedHints key	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
WORDPUZZ_028	WordPuzzleGameTest	testGetGameStateContainsAvailableHintsCount	WordPuzzleGame	getGameState	Verify getGameState contains availableHintsCount	Call getGameState()	State contains availableHintsCount key	Pass
WORDPUZZ_029	WordPuzzleGameTest	testGetResultReturnsNonNull	WordPuzzleGame	getResult	Verify getResult returns non-null	Call getResult()	Non-null Map	Pass
WORDPUZZ_030	WordPuzzleGameTest	testGetResultContainsWon	WordPuzzleGame	getResult	Verify getResult contains won status	Call getResult()	Result contains won key	Pass
WORDPUZZ_031	WordPuzzleGameTest	testGetResultContainsTime	WordPuzzleGame	getResult	Verify getResult contains time	Call getResult()	Result contains time key	Pass
WORDPUZZ_032	WordPuzzleGameTest	testGetResultContainsMoves	WordPuzzleGame	getResult	Verify getResult contains moves	Call getResult()	Result contains moves key	Pass
WORDPUZZ_033	WordPuzzleGameTest	testGetResultContainsHintsUsed	WordPuzzleGame	getResult	Verify getResult contains hintsUsed	Call getResult()	Result contains hintsUsed key	Pass
WORDPUZZ_034	WordPuzzleGameTest	testGetResultWonIsTrueAfterCorrectAnswer	WordPuzzleGame	getResult	Verify getResult shows won=true after correct answer	input=KEYBOARD then getResult()	Result[won] equals true	Pass
WORDPUZZ_035	WordPuzzleGameTest	testGetResultWonIsFalseAfterLoss	WordPuzzleGame	getResult	Verify getResult shows won=false after loss	Input 3 wrong answers, getResult()	Result[won] equals false	Pass
WORDPUZZ_036	WordPuzzleGameTest	testGetResultContainsAnswerWhenLost	WordPuzzleGame	getResult	Verify getResult contains answer when lost	Input 3 wrong answers, getResult()	Result contains answer key	Pass
WORDPUZZ_037	WordPuzzleGameTest	testGetResultDoesNotContainAnswerWhenWon	WordPuzzleGame	getResult	Verify getResult doesn't contain answer when won	input=KEYBOARD then getResult()	Result doesn't contain answer key	Pass
WORDPUZZ_038	WordPuzzleGameTest	testSaveStateReturnsNonNull	WordPuzzleGame	saveState	Verify saveState returns non-null	Call saveState()	Non-null Map	Pass
WORDPUZZ_039	WordPuzzleGameTest	testSaveStateContainsPuzzleType	WordPuzzleGame	saveState	Verify saveState contains puzzleType	Call saveState()	SavedState contains puzzleType key	Pass
WORDPUZZ_040	WordPuzzleGameTest	testSaveStateContainsAttemptsUsed	WordPuzzleGame	saveState	Verify saveState contains attemptsUsed	Call saveState()	SavedState contains attemptsUsed key	Pass
WORDPUZZ_041	WordPuzzleGameTest	testSaveStateContainsGuesses	WordPuzzleGame	saveState	Verify saveState contains guesses	Input WRONG, saveState()	SavedState contains guesses key	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
WORDPUZZ_042	WordPuzzleGame Test	testSaveStateContainsRevealedHints	WordPuzzleGame	saveState	Verify saveState contains revealedHints	Input HINT, saveState()	SavedState contains revealedHints key	Pass
WORDPUZZ_043	WordPuzzleGame Test	testSaveStateContainsWon	WordPuzzleGame	saveState	Verify saveState contains won status	Call saveState()	SavedState contains won key	Pass
WORDPUZZ_044	WordPuzzleGame Test	testSaveStateContainsStartTime	WordPuzzleGame	saveState	Verify saveState contains startTime	Call saveState()	SavedState contains startTime key	Pass
WORDPUZZ_045	WordPuzzleGame Test	testRestoreStateRestoresAttempts	WordPuzzleGame	restoreState	Verify restoreState restores attemptsUsed	Input WRONG, save, restore to new game	New game attemptsUsed equals saved value	Pass
WORDPUZZ_046	WordPuzzleGame Test	testRestoreStateRestoresGuesses	WordPuzzleGame	restoreState	Verify restoreState restores guesses list	Input WRONG, save, restore to new game	New game guesses size equals 1	Pass
WORDPUZZ_047	WordPuzzleGame Test	testRestoreStateRestoresPuzzleType	WordPuzzleGame	restoreState	Verify restoreState restores puzzleType	Save state, restore to new game	New game getGameType() returns RIDDLE	Pass
WORDPUZZ_048	WordPuzzleGame Test	testResetClearsGameOverState	WordPuzzleGame	reset	Verify reset clears game over state	Input correct answer, reset	isGameOver() returns false	Pass
WORDPUZZ_049	WordPuzzleGame Test	testResetClearsGuesses	WordPuzzleGame	reset	Verify reset clears guesses list	Input WRONG, reset	getGameState()[guesses] is empty	Pass
WORDPUZZ_050	WordPuzzleGame Test	testResetClearsRevealedHints	WordPuzzleGame	reset	Verify reset clears revealed hints	Input HINT, reset	getGameState()[revealedHints] is empty	Pass
WORDPUZZ_051	WordPuzzleGame Test	testResetResetsAttemptsUsed	WordPuzzleGame	reset	Verify reset resets attemptsUsed to zero	Input WRONG, reset	getGameState()[attemptsUsed] equals 0	Pass
WORDPUZZ_052	WordPuzzleGame Test	testSetHintsAllowsHintRevealing	WordPuzzleGame	setHints	Verify setHints allows hint revealing	Set 2 hints, input HINT	getGameState()[revealedHints] is not empty	Pass
WORDPUZZ_053	WordPuzzleGame Test	testSetHintsUpdatesAvailableCount	WordPuzzleGame	setHints	Verify setHints updates available HintsCount	Set 2 hints	getGameState()[availableHintsCount] equals 2	Pass
WORDPUZZ_054	WordPuzzleGame Test	testSetPuzzleIdStoresId	WordPuzzleGame	setPuzzleId	Verify setPuzzleId stores puzzleId	setPuzzleId(TEST_001)	saveState()[puzzleId] equals TEST_001	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
POSITION_009	PositionTest	testEqualsReturnsFalseForNull	Position	equals	Verify equals returns false for null	position.equals(null)	Returns false	Pass
POSITION_010	PositionTest	testEqualsReturnsFalseForDifferentClass	Position	equals	Verify equals returns false for different class	position.equals("not a position")	Returns false	Pass
POSITION_011	PositionTest	testHashCodeConsistency	Position	hashCode	Verify hashCode is consistent for equal positions	Position (3,5) and Position (3,5)	Both have same hashCode	Pass
PUZZLE_001	PuzzleTest	testConstructorSetsPuzzleId	Puzzle	Puzzle (constructor)	Verify constructor sets puzzleId	puzzleId="P001", type="MAZE", difficulty="EASY", title="Test Maze", description, data	getPuzzleId() returns P001	Pass
PUZZLE_002	PuzzleTest	testConstructorSetsPuzzleType	Puzzle	Puzzle (constructor)	Verify constructor sets puzzleType	Full constructor parameters	getPuzzleType() returns MAZE	Pass
PUZZLE_003	PuzzleTest	testConstructorSetsDifficulty	Puzzle	Puzzle (constructor)	Verify constructor sets difficulty	Full constructor parameters	getDifficulty() returns EASY	Pass
PUZZLE_004	PuzzleTest	testConstructorSetsTitle	Puzzle	Puzzle (constructor)	Verify constructor sets title	Full constructor parameters	getTitle() returns Test Maze	Pass
PUZZLE_005	PuzzleTest	testConstructorSetsDescription	Puzzle	Puzzle (constructor)	Verify constructor sets description	Full constructor parameters	getDescription() returns A simple test maze	Pass
PUZZLE_006	PuzzleTest	testConstructorSetsData	Puzzle	Puzzle (constructor)	Verify constructor sets data	Full constructor parameters with data Map	getData() equals input Map	Pass
PUZZLE_007	PuzzleTest	testSetPuzzleId	Puzzle	setPuzzleId	Verify setPuzzleId updates puzzleId	setPuzzleId(P002)	getPuzzleId() returns P002	Pass
PUZZLE_008	PuzzleTest	testSetPuzzleType	Puzzle	setPuzzleType	Verify setPuzzleType updates puzzleType	setPuzzleType(MATCHING)	getPuzzleType() returns MATCHING	Pass
PUZZLE_009	PuzzleTest	testSetDifficulty	Puzzle	setDifficulty	Verify setDifficulty updates difficulty	setDifficulty(HARD)	getDifficulty() returns HARD	Pass
PUZZLE_010	PuzzleTest	testSetTitle	Puzzle	setTitle	Verify setTitle updates title	setTitle(New Title)	getTitle() returns New Title	Pass
PUZZLE_011	PuzzleTest	testSetDescription	Puzzle	setDescription	Verify setDescription updates description	setDescription(New Description)	getDescription() returns New Description	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
PUZZLE_012	PuzzleTest	testSetData	Puzzle	setData	Verify setData updates data	setData(newDataMap)	getData() equals newData	Pass
PUZZLE_013	PuzzleTest	testToStringContainsPuzzleId	Puzzle	toString	Verify toString contains puzzleId	toString()	Contains P001	Pass
PUZZLE_014	PuzzleTest	testToStringContainsTitle	Puzzle	toString	Verify toString contains title	toString()	Contains Test Maze	Pass
AUTH_001	AuthenticationServiceTest	testRegisterWithValidDataReturnsTrue	AuthenticationService	register	Verify registration with valid data returns true	userId="usr01", password="pass123", firstName="John", lastName="Doe", email="john@example.com"	true	Pass
AUTH_002	AuthenticationServiceTest	testRegisterCreatesUser	AuthenticationService	register	Verify user is created after registration	userId="usr02", password="pass123", firstName="Jane", lastName="Doe", email="jane@example.com"	User exists in system	Pass
AUTH_003	AuthenticationServiceTest	testRegisterWithNullUserIdReturnsFalse	AuthenticationService	register	Verify registration with null userId returns false	userId=null, password="pass123", firstName="Test", lastName="User", email="test@example.com"	false	Pass
AUTH_004	AuthenticationServiceTest	testRegisterWithEmptyUserIdReturnsFalse	AuthenticationService	register	Verify registration with empty userId returns false	userId="", password="pass123", firstName="Test", lastName="User", email="test@example.com"	false	Pass
AUTH_005	AuthenticationServiceTest	testRegisterWithShortUserIdReturnsFalse	AuthenticationService	register	Verify registration with short userId (< 5 chars) returns false	userId="usr", password="pass123", firstName="Test", lastName="User", email="test@example.com"	false	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
AUTH_006	AuthenticationServiceTest	testRegisterWithLongUserIdReturnsFalse	AuthenticationService	register	Verify registration with long userId (> 5 chars) returns false	userId="user01", password="pass123", firstName="Test", lastName="User", email="test@example.com"	false	Pass
AUTH_007	AuthenticationServiceTest	testRegisterWithShortPasswordReturnsFalse	AuthenticationService	register	Verify registration with short password returns false	userId="usr03", password="abc", firstName="Test", lastName="User", email="test@example.com"	false	Pass
AUTH_008	AuthenticationServiceTest	testRegisterWithNullPasswordReturnsFalse	AuthenticationService	register	Verify registration with null password returns false	userId="usr04", password=null, firstName="Test", lastName="User", email="test@example.com"	false	Pass
AUTH_009	AuthenticationServiceTest	testRegisterWithDuplicateUserIdReturnsFalse	AuthenticationService	register	Verify registration with duplicate userId returns false	Register usr05 twice with different details	false on second attempt	Pass
AUTH_010	AuthenticationServiceTest	testLoginWithValidCredentialsReturnsTrue	AuthenticationService	login	Verify login with valid credentials returns true	Register usr06, then login with userId="usr06", password="pass123"	true	Pass
AUTH_011	AuthenticationServiceTest	testLoginSetsLoggedInStatus	AuthenticationService	login	Verify login sets logged in status to true	Register usr07, login with valid credentials	isLoggedIn() returns true	Pass
AUTH_012	AuthenticationServiceTest	testLoginSetsCurrentUser	AuthenticationService	login	Verify login sets current user	Register usr08, login with valid credentials	getCurrentUser() returns non-null User	Pass
AUTH_013	AuthenticationServiceTest	testLoginWithWrongPasswordReturnsFalse	AuthenticationService	login	Verify login with wrong password returns false	Register usr09, login with wrong password	false	Pass
AUTH_014	AuthenticationServiceTest	testLoginWithNonexistentUserReturnsFalse	AuthenticationService	login	Verify login with non-existent user returns false	userId="usr99", password="pass123"	false	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
AUTH_015	AuthenticationServiceTest	testLoginDoesNotSetCurrentUserOnFailure	AuthenticationService	login	Verify failed login does not set current user	Register usr10, login with wrong password	getCurrentUser() returns null	Pass
AUTH_016	AuthenticationServiceTest	testLoginReturnsCorrectUser	AuthenticationService	login	Verify login returns correct user object	Register usr11, login successfully	getCurrentUser().getUserId() equals usr11	Pass
AUTH_017	AuthenticationServiceTest	testLogoutClearsCurrentUser	AuthenticationService	logout	Verify logout clears current user	Register, login, then logout	getCurrentUser() returns null	Pass
AUTH_018	AuthenticationServiceTest	testLogoutSetsLoggedOutStatus	AuthenticationService	logout	Verify logout sets logged out status	Register, login, then logout	isLoggedIn() returns false	Pass
AUTH_019	AuthenticationServiceTest	testLogoutWithoutLoginDoesNotFail	AuthenticationService	logout	Verify logout without login does not fail	Call logout() without logging in	isLoggedIn() returns false	Pass
AUTH_020	AuthenticationServiceTest	testIsLoggedInReturnsFalseInitially	AuthenticationService	isLoggedIn	Verify initial logged in status is false	Create new AuthenticationService	isLoggedIn() returns false	Pass
AUTH_021	AuthenticationServiceTest	testGetCurrentUserReturnsNullInitially	AuthenticationService	getCurrentUser	Verify initial current user is null	Create new AuthenticationService	getCurrentUser() returns null	Pass
AUTH_022	AuthenticationServiceTest	testMultipleLoginsUpdateCurrentUser	AuthenticationService	login	Verify multiple logins update current user correctly	Register usr14 and usr15, login usr14, logout, login usr15	getCurrentUser().getUserId() equals usr15	Pass
CARD_001	CardTest	testConstructorSetsId	Card	Card (constructor)	Verify constructor sets card ID	id="C001", value="■", name="Star"	getId() returns C001	Pass
CARD_002	CardTest	testConstructorSetsValue	Card	Card (constructor)	Verify constructor sets card value	id="C001", value="■", name="Star"	getValue() returns ■	Pass
CARD_003	CardTest	testConstructorSetsName	Card	Card (constructor)	Verify constructor sets card name	id="C001", value="■", name="Star"	getName() returns Star	Pass
CARD_004	CardTest	testSetId	Card	setId	Verify setId updates card ID	setId(C002)	getId() returns C002	Pass
CARD_005	CardTest	testSetValue	Card	setValue	Verify setValue updates card value	setValue(■)	getValue() returns ■	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
CARD_006	CardTest	testSetName	Card	setName	Verify setName updates card name	setName(Balloon)	getName() returns Balloon	Pass
CARD_007	CardTest	testToStringContainsId	Card	toString	Verify toString contains card ID	toString()	Contains C001	Pass
CARD_008	CardTest	testToStringContainsName	Card	toString	Verify toString contains card name	toString()	Contains Star	Pass
CARD_009	CardTest	testToStringContainsValue	Card	toString	Verify toString contains card value	toString()	Contains ■	Pass
CERT_001	CertificateServiceTest	testAwardCertificateReturnsNonNull	CertificateService	awardCertificate	Verify awarding certificate returns non-null object	userId="usr01", puzzle=testPuzzle, score=100	Non-null Certificate object	Pass
CERT_002	CertificateServiceTest	testAwardCertificateHasCorrectUserId	CertificateService	awardCertificate	Verify awarded certificate has correct userId	userId="usr02", puzzle=testPuzzle, score=95	Certificate.getUserId() equals usr02	Pass
CERT_003	CertificateServiceTest	testAwardCertificateHasCorrectPuzzleId	CertificateService	awardCertificate	Verify awarded certificate has correct puzzleId	userId="usr03", puzzle=testPuzzle, score=88	Certificate.getPuzzleId() equals puzzle1	Pass
CERT_004	CertificateServiceTest	testAwardCertificateHasCorrectScore	CertificateService	awardCertificate	Verify awarded certificate has correct score	userId="usr04", puzzle=testPuzzle, score=92	Certificate.getScoreAchieved() equals 92	Pass
CERT_005	CertificateServiceTest	testAwardCertificateHasCorrectDifficulty	CertificateService	awardCertificate	Verify awarded certificate has correct difficulty	userId="usr05", puzzle=testPuzzle (EASY), score=100	Certificate.getDifficulty() equals EASY	Pass
CERT_006	CertificateServiceTest	testAwardCertificateGeneratesUniqueId	CertificateService	awardCertificate	Verify each awarded certificate has unique ID	Award 2 certificates to same user for same puzzle	Different certificateIds	Pass
CERT_007	CertificateServiceTest	testAwardCertificateStoresInSystem	CertificateService	awardCertificate	Verify awarded certificate is stored in system	Award certificate to usr07, check hasCertificate	hasCertificate returns true	Pass
CERT_008	CertificateServiceTest	testGetUserCertificatesReturnsNonNull	CertificateService	getUserCertificates	Verify getUserCertificates returns non-null list	userId=usr08	Non-null List	Pass
CERT_009	CertificateServiceTest	testGetUserCertificatesReturnsEmptyForNewUser	CertificateService	getUserCertificates	Verify new user has empty certificate list	userId=usr09	Empty List	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
CERT_010	CertificateServiceTest	testGetUserCertificatesReturnsAwardedCertificates	CertificateService	getUserCertificates	Verify getUserCertificates returns awarded certificates	Award 1 certificate to usr10, get certificates	List size equals 1	Pass
CERT_011	CertificateServiceTest	testGetUserCertificatesReturnsMultipleCertificates	CertificateService	getUserCertificates	Verify getUserCertificates returns multiple certificates	Award 2 certificates to usr11, get certificates	List size equals 2	Pass
CERT_012	CertificateServiceTest	testGetCertificateStatsReturnsNonNull	CertificateService	getCertificateStats	Verify getCertificateStats returns non-null map	userId=usr12	Non-null Map	Pass
CERT_013	CertificateServiceTest	testGetCertificateStatsInitializesAllDifficulties	CertificateService	getCertificateStats	Verify stats map contains all difficulty levels	userId=usr13	Map contains keys: EASY, MEDIUM, HARD	Pass
CERT_014	CertificateServiceTest	testGetCertificateStatsCountsCorrectly	CertificateService	getCertificateStats	Verify stats counts certificates correctly	Award 1 EASY certificate to usr14, get stats	Stats[EASY] equals 1	Pass
CERT_015	CertificateServiceTest	testGetCertificateStatsCountsMultipleDifficulties	CertificateService	getCertificateStats	Verify stats counts multiple difficulties	Award EASY and MEDIUM certificates to usr15, get stats	Stats[EASY]=1, Stats[MEDIUM]=1	Pass
CERT_016	CertificateServiceTest	testHasCertificateReturnsFalseForNoCertificate	CertificateService	hasCertificate	Verify hasCertificate returns false when no certificate exists	userId=usr16, puzzleId=puzzle1	false	Pass
CERT_017	CertificateServiceTest	testHasCertificateReturnsTrueAfterAwarding	CertificateService	hasCertificate	Verify hasCertificate returns true after awarding	Award certificate to usr17 for puzzle1, check hasCertificate	true	Pass
CERT_018	CertificateServiceTest	testHasCertificateReturnsFalseForDifferentPuzzle	CertificateService	hasCertificate	Verify hasCertificate returns false for different puzzle	Award certificate for puzzle1 to usr18, check puzzle2	false	Pass
CERT_019	CertificateServiceTest	testGetCertificateCountReturnsZeroForNewUser	CertificateService	getCertificateCount	Verify new user has zero certificates	userId=usr19	0	Pass
CERT_020	CertificateServiceTest	testGetCertificateCountReturnsCorrectCount	CertificateService	getCertificateCount	Verify count returns correct number	Award 1 certificate to usr20, get count	1	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
CERT_021	CertificateServiceTest	testGetCertificateCountReturnsCorrectCountForMultiple	CertificateService	getCertificateCount	Verify count returns correct number for multiple	Award 2 certificates to usr21, get count	2	Pass
CERTENT_001	CertificateTest	testConstructorSetsCertificateId	Certificate	Certificate (constructor)	Verify constructor sets certificateId	certificateId="CERT_001", userId="user1", puzzleId="puzzle1", description="Completed Test Puzzle", difficulty="MEDIUM", scoreAchieved=100	getCertificateId() returns CERT_001	Pass
CERTENT_002	CertificateTest	testConstructorSetsUserId	Certificate	Certificate (constructor)	Verify constructor sets userId	Full constructor parameters	getUserId() returns user1	Pass
CERTENT_003	CertificateTest	testConstructorSetsPuzzleId	Certificate	Certificate (constructor)	Verify constructor sets puzzleId	Full constructor parameters	getPuzzleId() returns puzzle1	Pass
CERTENT_004	CertificateTest	testConstructorSetsDescription	Certificate	Certificate (constructor)	Verify constructor sets description	Full constructor parameters	getDescription() returns Completed Test Puzzle	Pass
CERTENT_005	CertificateTest	testConstructorSetsDifficulty	Certificate	Certificate (constructor)	Verify constructor sets difficulty	Full constructor parameters	getDifficulty() returns MEDIUM	Pass
CERTENT_006	CertificateTest	testConstructorSetsScore	Certificate	Certificate (constructor)	Verify constructor sets score	Full constructor parameters	getScoreAchieved() returns 100	Pass
CERTENT_007	CertificateTest	testConstructorSetsEarnedAt	Certificate	Certificate (constructor)	Verify constructor sets earnedAt timestamp	Full constructor parameters	getEarnedAt() is non-null	Pass
CERTENT_008	CertificateTest	testDefaultConstructorSetsEarnedAt	Certificate	Certificate (constructor)	Verify default constructor sets earnedAt	Default constructor	getEarnedAt() is non-null	Pass
CERTENT_009	CertificateTest	testSetCertificateId	Certificate	setCertificateId	Verify setCertificateId updates ID	setCertificateId(CERT_002)	getCertificateId() returns CERT_002	Pass
CERTENT_010	CertificateTest	testSetUserId	Certificate	setUserId	Verify setUserId updates userId	setUserId(user2)	getUserId() returns user2	Pass
CERTENT_011	CertificateTest	testSetPuzzleId	Certificate	setPuzzleId	Verify setPuzzleId updates puzzleId	setPuzzleId(puzzle2)	getPuzzleId() returns puzzle2	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
CERTENT_012	CertificateTest	testSetDescription	Certificate	setDescription	Verify setDescription updates description	setDescription(New Description)	getDescription() returns New Description	Pass
CERTENT_013	CertificateTest	testSetDifficulty	Certificate	setDifficulty	Verify setDifficulty updates difficulty	setDifficulty(HARD)	getDifficulty() returns HARD	Pass
CERTENT_014	CertificateTest	testSetScoreAchieved	Certificate	setScoreAchieved	Verify setScoreAchieved updates score	setScoreAchieved (150)	getScoreAchieved () returns 150	Pass
CERTENT_015	CertificateTest	testSetEarnedAt	Certificate	setEarnedAt	Verify setEarnedAt updates timestamp	setEarnedAt(testTime)	getEarnedAt() equals testTime	Pass
CERTENT_016	CertificateTest	testToStringContainsCertificateId	Certificate	toString	Verify toString contains certificateId	toString()	Contains CERT_001	Pass
CERTENT_017	CertificateTest	testToStringContainsDifficulty	Certificate	toString	Verify toString contains difficulty	toString()	Contains MEDIUM	Pass
FACADE_001	GameDataFacadeTest	testGetInstanceReturnsSameInstance	GameDataFacade	getInstance	Verify singleton returns same instance	Call getInstance() twice, compare instances	Same instance	Pass
FACADE_002	GameDataFacadeTest	testGetInstanceReturnsNonNull	GameDataFacade	getInstance	Verify getInstance returns non-null	Call getInstance()	Non-null instance	Pass
FACADE_003	GameDataFacadeTest	testResetInstanceClearsInstance	GameDataFacade	resetInstance	Verify resetInstance clears singleton	Get instance, call resetInstance()	Instance is reset	Pass
FACADE_004	GameDataFacadeTest	testAddUserAddsUserToSystem	GameDataFacade	addUser	Verify addUser adds user to system	user with userId="tst01"	Returns true	Pass
FACADE_005	GameDataFacadeTest	testAddUserMakesUserRetrievable	GameDataFacade	addUser	Verify added user can be retrieved	Add user tst02, check userIdExists	userIdExists returns true	Pass
FACADE_006	GameDataFacadeTest	testAddUserReturnsFalseForDuplicateUserId	GameDataFacade	addUser	Verify duplicate userId is rejected	Add tst03 twice with different emails	Second add returns false	Pass
FACADE_007	GameDataFacadeTest	testAddUserReturnsFalseForDuplicateEmail	GameDataFacade	addUser	Verify duplicate email is rejected	Add tst04 and tst05 with same email	Second add returns false	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
FACADE_008	GameDataFacadeTest	testGetUserReturnsEmptyForNonexistentUser	GameDataFacade	getUser	Verify getUser returns empty for non-existent user	userId=nonexistent	Optional.empty()	Pass
FACADE_009	GameDataFacadeTest	testGetUserReturnsUserWhenExists	GameDataFacade	getUser	Verify getUser returns user when exists	Add user tst06, get user	Optional.isPresent() is true	Pass
FACADE_010	GameDataFacadeTest	testGetUserReturnsCorrectUser	GameDataFacade	getUser	Verify getUser returns correct user data	Add user tst07, get user	User.getUserId() equals tst07	Pass
FACADE_011	GameDataFacadeTest	testUpdateUserReturnsTrueForExistingUser	GameDataFacade	updateUser	Verify updateUser returns true for existing user	Add user tst08, update firstName, call updateUser	Returns true	Pass
FACADE_012	GameDataFacadeTest	testUpdateUserReturnsFalseForNonexistentUser	GameDataFacade	updateUser	Verify updateUser returns false for non-existent user	Update user with userId=nonexistent	Returns false	Pass
FACADE_013	GameDataFacadeTest	testUpdateUserActuallyUpdatesData	GameDataFacade	updateUser	Verify updateUser persists changes	Add user tst09, update firstName to Updated, retrieve user	Retrieved user has firstName=Updated	Pass
FACADE_014	GameDataFacadeTest	testUserIdExistsReturnsTrueForExistingUser	GameDataFacade	userIdExists	Verify userIdExists returns true for existing user	Add user tst10, check exists	Returns true	Pass
FACADE_015	GameDataFacadeTest	testUserIdExistsReturnsFalseForNonexistentUser	GameDataFacade	userIdExists	Verify userIdExists returns false for non-existent user	userId=nonexistent	Returns false	Pass
FACADE_016	GameDataFacadeTest	testEmailExistsReturnsTrueForExistingEmail	GameDataFacade	emailExists	Verify emailExists returns true for existing email	Add user tst11 with email test11@example.com, check exists	Returns true	Pass
FACADE_017	GameDataFacadeTest	testEmailExistsReturnsFalseForNonexistentEmail	GameDataFacade	emailExists	Verify emailExists returns false for non-existent email	email=nonexistent@example.com	Returns false	Pass
FACADE_018	GameDataFacadeTest	testGetUsersReturnsNonNull	GameDataFacade	getUsers	Verify getUsers returns non-null list	Call getUsers()	Non-null List	Pass
FACADE_019	GameDataFacadeTest	testGetUsersReturnsCorrectCount	GameDataFacade	getUsers	Verify getUsers returns correct count	Add 2 users (tst12, tst13), call getUsers()	List size equals 2	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
FACADE_020	GameDataFacadeTest	testGetAllPuzzlesReturnsNonNull	GameDataFacade	getAllPuzzles	Verify getAllPuzzles returns non-null list	Call getAllPuzzles()	Non-null List	Pass
FACADE_021	GameDataFacadeTest	testGetAllPuzzlesReturnsCorrectCount	GameDataFacade	getAllPuzzles	Verify getAllPuzzles returns correct count	Test data has 2 puzzles	List size equals 2	Pass
FACADE_022	GameDataFacadeTest	testGetPuzzleReturnsEmptyForNonexistent	GameDataFacade	getPuzzle	Verify getPuzzle returns empty for non-existent puzzle	puzzleId=nonexistent	Optional.empty()	Pass
FACADE_023	GameDataFacadeTest	testGetPuzzleReturnsPuzzleWhenExists	GameDataFacade	getPuzzle	Verify getPuzzle returns puzzle when exists	puzzleId=puzzle1	Optional.isPresent() is true	Pass
FACADE_024	GameDataFacadeTest	testGetPuzzleReturnsCorrectPuzzle	GameDataFacade	getPuzzle	Verify getPuzzle returns correct puzzle data	puzzleId=puzzle1	Puzzle.getPuzzleId() equals puzzle1	Pass
FACADE_025	GameDataFacadeTest	testGetPuzzlesByTypeReturnsCorrectType	GameDataFacade	getPuzzlesByType	Verify getPuzzlesByType filters correctly	type=MAZE	Returns 1 puzzle	Pass
FACADE_026	GameDataFacadeTest	testGetPuzzlesByTypeIgnoresCase	GameDataFacade	getPuzzlesByType	Verify getPuzzlesByType is case-insensitive	type=word (lowercase)	Returns 1 puzzle	Pass
FACADE_027	GameDataFacadeTest	testGetPuzzlesByTypeReturnsEmptyForNonexistentType	GameDataFacade	getPuzzlesByType	Verify getPuzzlesByType returns empty for unknown type	type=NONEXISTENT	Empty list	Pass
FACADE_028	GameDataFacadeTest	testGetPuzzlesByDifficultyReturnsCorrectPuzzles	GameDataFacade	getPuzzlesByDifficulty	Verify getPuzzlesByDifficulty filters correctly	type=MAZE, difficulty=EASY	Returns 1 puzzle	Pass
FACADE_029	GameDataFacadeTest	testGetPuzzlesByDifficultyReturnsEmptyForWrongDifficulty	GameDataFacade	getPuzzlesByDifficulty	Verify getPuzzlesByDifficulty returns empty for wrong difficulty	type=MAZE, difficulty=HARD	Empty list	Pass
FACADE_030	GameDataFacadeTest	testGetAvailablePuzzleTypesReturnsNonNull	GameDataFacade	getAvailablePuzzleTypes	Verify getAvailablePuzzleTypes returns non-null set	Call getAvailablePuzzleTypes()	Non-null Set	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
FACADE_031	GameDataFacadeTest	testGetAvailablePuzzleTypesReturnsCorrectCount	GameDataFacade	getAvailablePuzzleTypes	Verify getAvailablePuzzleTypes returns correct count	Test data has MAZE and WORD types	Set size equals 2	Pass
FACADE_032	GameDataFacadeTest	testGetAvailablePuzzleTypesContainsMaze	GameDataFacade	getAvailablePuzzleTypes	Verify getAvailablePuzzleTypes contains MAZE	Call getAvailablePuzzleTypes()	Set contains MAZE	Pass
FACADE_033	GameDataFacadeTest	testGetAvailablePuzzleTypesContainsWord	GameDataFacade	getAvailablePuzzleTypes	Verify getAvailablePuzzleTypes contains WORD	Call getAvailablePuzzleTypes()	Set contains WORD	Pass
FACADE_034	GameDataFacadeTest	testGetHintsForPuzzleReturnsNonNull	GameDataFacade	getHintsForPuzzle	Verify getHintsForPuzzle returns non-null list	puzzleId=puzzle1	Non-null List	Pass
FACADE_035	GameDataFacadeTest	testGetHintsForPuzzleReturnsCorrectCount	GameDataFacade	getHintsForPuzzle	Verify getHintsForPuzzle returns correct count	puzzleId=puzzle1 has 2 hints	List size equals 2	Pass
FACADE_036	GameDataFacadeTest	testGetHintsForPuzzleReturnsEmptyForNonexistent	GameDataFacade	getHintsForPuzzle	Verify getHintsForPuzzle returns empty for non-existent puzzle	puzzleId=nonexistent	Empty list	Pass
FACADE_037	GameDataFacadeTest	testGetHintsForPuzzleReturnsOrderedByPriority	GameDataFacade	getHintsForPuzzle	Verify getHintsForPuzzle returns hints ordered by priority	puzzleId=puzzle1	First hint has priority 1	Pass
FACADE_038	GameDataFacadeTest	testGetUserProgressReturnsNonNull	GameDataFacade	getUserProgress	Verify getUserProgress returns non-null	Add user tst14, get progress	Non-null UserProgress	Pass
FACADE_039	GameDataFacadeTest	testGetUserProgressCreatesNewForNewUser	GameDataFacade	getUserProgress	Verify getUserProgress creates new progress for new user	userId=newuser	UserProgress.getUserId() equals newuser	Pass
FACADE_040	GameDataFacadeTest	testGetUserProgressReturnsSameInstanceForSameUser	GameDataFacade	getUserProgress	Verify getUserProgress returns same instance for same user	Call getUserProgress(tst15) twice, compare instances	Same instance	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
FACADE_041	GameDataFacadeTest	testCompletePuzzleMarksAsCompleted	GameDataFacade	completePuzzle	Verify completePuzzle marks puzzle as completed	Add user tst16, complete puzzle1, check isPuzzleCompleted	isPuzzleCompleted returns true	Pass
FACADE_042	GameDataFacadeTest	testIsPuzzleCompletedReturnsFalseForIncomplete	GameDataFacade	isPuzzleCompleted	Verify isPuzzleCompleted returns false for incomplete puzzle	Add user tst17, check puzzle1	Returns false	Pass
FACADE_043	GameDataFacadeTest	testCompletePuzzleStoresInList	GameDataFacade	completePuzzle	Verify completePuzzle stores puzzle in completed list	Add user tst18, complete puzzle1, get progress	CompletedPuzzles contains puzzle1	Pass
FACADE_044	GameDataFacadeTest	testCompletePuzzleStoresScore	GameDataFacade	completePuzzle	Verify completePuzzle stores score	Add user tst19, complete puzzle1 with score 95, get progress	PuzzleScores[puzzle1] equals 95	Pass
FACADE_045	GameDataFacadeTest	testSaveUserProgressPersistsData	GameDataFacade	saveUserProgress	Verify saveUserProgress persists data	Add user tst20, add completed puzzle, save, retrieve progress	Retrieved progress has totalScore=88	Pass
FACADE_046	GameDataFacadeTest	testAddCertificateAddsToSystem	GameDataFacade	addCertificate	Verify addCertificate adds to system	Add user tst21, add certificate, get certificates	Certificates list size equals 1	Pass
FACADE_047	GameDataFacadeTest	testGetUserCertificatesReturnsEmptyForNewUser	GameDataFacade	getUserCertificates	Verify getUserCertificates returns empty for new user	userId=newuser	Empty list	Pass
FACADE_048	GameDataFacadeTest	testGetUserCertificatesReturnsNonNull	GameDataFacade	getUserCertificates	Verify getUserCertificates returns non-null	Add user tst22, get certificates	Non-null List	Pass
FACADE_049	GameDataFacadeTest	testGetCertificateStatsReturnsNonNull	GameDataFacade	getCertificateStats	Verify getCertificateStats returns non-null	Add user tst23, get stats	Non-null Map	Pass
FACADE_050	GameDataFacadeTest	testGetCertificateStatsInitializesAllDifficulties	GameDataFacade	getCertificateStats	Verify getCertificateStats initializes all difficulties	Add user tst24, get stats	Map contains EASY key	Pass
FACADE_051	GameDataFacadeTest	testGetCertificateStatsCountsCorrectly	GameDataFacade	getCertificateStats	Verify getCertificateStats counts correctly	Add user tst25, add 1 EASY certificate, get stats	Stats[EASY] equals 1	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
FACADE_052	GameDataFacade Test	testHasCertificate ReturnsFalseWhe nNoCertificate	GameDataFacade	hasCertificate	Verify hasCertificate returns false when no certificate	Add user tst26, check puzzle1	Returns false	Pass
FACADE_053	GameDataFacade Test	testHasCertificate ReturnsTrueWhen CertificateExists	GameDataFacade	hasCertificate	Verify hasCertificate returns true when certificate exists	Add user tst27, add certificate for puzzle1, check	Returns true	Pass
FACADE_054	GameDataFacade Test	testGetLeaderboa rdReturnsNonNull	GameDataFacade	getLeaderboard	Verify getLeaderboard returns non-null	getLeaderboard(1 0)	Non-null List	Pass
FACADE_055	GameDataFacade Test	testGetLeaderboa rdRespectsLimit	GameDataFacade	getLeaderboard	Verify getLeaderboard respects limit	getLeaderboard(5)	List size <= 5	Pass
FACADE_056	GameDataFacade Test	testGetLeaderboa rdReturnsEmptyIn itially	GameDataFacade	getLeaderboard	Verify getLeaderboard returns empty initially	getLeaderboard(1 0)	Empty list	Pass
FACADE_057	GameDataFacade Test	testGetUserRank ReturnsNegativeF orNonexistentUse r	GameDataFacade	getUserRank	Verify getUserRank returns -1 for non-existent user	userId=nonexisten t	Returns -1	Pass
FACADE_058	GameDataFacade Test	testLeaderboardU pdatedOnUserPro gress	GameDataFacade	getUserRank	Verify leaderboard updates on user progress	Add user tst28, complete puzzle, save progress, get rank	Rank >= 0	Pass
FACADE_059	GameDataFacade Test	testLeaderboardM akesUserRankabl e	GameDataFacade	getLeaderboard	Verify completing puzzle makes user rankable	Add user tst29, complete puzzle, get leaderboard	Leaderboard size equals 1	Pass
LOADER_001	GameDataLoader Test	testReadUsersRet urnsNonNull	GameDataLoader	readUsers	Verify readUsers returns non-null list	Empty users.json file	Non-null List	Pass
LOADER_002	GameDataLoader Test	testReadUsersRet urnsListType	GameDataLoader	readUsers	Verify readUsers returns List type	Empty users.json file	Returns instanceof List	Pass
LOADER_003	GameDataLoader Test	testReadUsersRet urnsEmptyListWh enFileEmpty	GameDataLoader	readUsers	Verify readUsers returns empty list for empty file	users.json contains []	Empty list	Pass
LOADER_004	GameDataLoader Test	testReadUsersRet urnsSingleUser	GameDataLoader	readUsers	Verify readUsers returns single user	users.json with 1 user	List size equals 1	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
LOADER_005	GameDataLoader Test	testReadUsersReturnsMultipleUsers	GameDataLoader	readUsers	Verify readUsers returns multiple users	users.json with 2 users	List size equals 2	Pass
LOADER_006	GameDataLoader Test	testReadUsersHandlesMissingFile	GameDataLoader	readUsers	Verify readUsers handles missing file gracefully	Delete users.json file	Non-null List	Pass
LOADER_007	GameDataLoader Test	testReadUsersReturnsEmptyListOnMissingFile	GameDataLoader	readUsers	Verify readUsers returns empty list for missing file	Missing users.json	Empty list	Pass
LOADER_008	GameDataLoader Test	testReadGameDataReturnsNonNull	GameDataLoader	readGameData	Verify readGameData returns non-null	Empty gamedata.json with structure	Non-null GameData	Pass
LOADER_009	GameDataLoader Test	testReadGameDataReturnsGameDataType	GameDataLoader	readGameData	Verify readGameData returns GameData type	Empty gamedata.json	Returns instanceof GameData	Pass
LOADER_010	GameDataLoader Test	testReadGameDataInitializesPuzzles	GameDataLoader	readGameData	Verify readGameData initializes puzzles list	Empty gamedata.json	getPuzzles() is non-null	Pass
LOADER_011	GameDataLoader Test	testReadGameDataInitializesHints	GameDataLoader	readGameData	Verify readGameData initializes hints list	Empty gamedata.json	getHints() is non-null	Pass
LOADER_012	GameDataLoader Test	testReadGameDataLoadsPuzzles	GameDataLoader	readGameData	Verify readGameData loads puzzles	gamedata.json with 1 puzzle	getPuzzles().size() equals 1	Pass
LOADER_013	GameDataLoader Test	testReadGameDataLoadsHints	GameDataLoader	readGameData	Verify readGameData loads hints	gamedata.json with 1 hint	getHints().size() equals 1	Pass
LOADER_014	GameDataLoader Test	testReadGameDataHandlesMissingFile	GameDataLoader	readGameData	Verify readGameData handles missing file	Delete gamedata.json	Non-null GameData	Pass
LOADER_015	GameDataLoader Test	testReadGameDataCreatesEmptyDataOnMissingFile	GameDataLoader	readGameData	Verify readGameData creates empty data for missing file	Missing gamedata.json	getPuzzles() is non-null	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
GAMEDATA_001	GameDataTest	testDefaultConstructorInitializesPuzzlesList	GameData	GameData (constructor)	Verify default constructor initializes puzzles list	Default constructor	getPuzzles() is non-null	Pass
GAMEDATA_002	GameDataTest	testDefaultConstructorInitializesHintsList	GameData	GameData (constructor)	Verify default constructor initializes hints list	Default constructor	getHints() is non-null	Pass
GAMEDATA_003	GameDataTest	testDefaultConstructorInitializesUserProgressList	GameData	GameData (constructor)	Verify default constructor initializes userProgress list	Default constructor	getUserProgress() is non-null	Pass
GAMEDATA_004	GameDataTest	testDefaultConstructorInitializesCertificatesList	GameData	GameData (constructor)	Verify default constructor initializes certificates list	Default constructor	getCertificates() is non-null	Pass
GAMEDATA_005	GameDataTest	testDefaultConstructorInitializesLeaderboardList	GameData	GameData (constructor)	Verify default constructor initializes leaderboard list	Default constructor	getLeaderboard() is non-null	Pass
GAMEDATA_006	GameDataTest	testDefaultConstructorCreatesEmptyPuzzlesList	GameData	GameData (constructor)	Verify default constructor creates empty puzzles list	Default constructor	getPuzzles().isEmpty() is true	Pass
GAMEDATA_007	GameDataTest	testDefaultConstructorCreatesEmptyHintsList	GameData	GameData (constructor)	Verify default constructor creates empty hints list	Default constructor	getHints().isEmpty() is true	Pass
GAMEDATA_008	GameDataTest	testDefaultConstructorCreatesEmptyUserProgressList	GameData	GameData (constructor)	Verify default constructor creates empty userProgress list	Default constructor	getUserProgress().isEmpty() is true	Pass
GAMEDATA_009	GameDataTest	testDefaultConstructorCreatesEmptyCertificatesList	GameData	GameData (constructor)	Verify default constructor creates empty certificates list	Default constructor	getCertificates().isEmpty() is true	Pass
GAMEDATA_010	GameDataTest	testDefaultConstructorCreatesEmptyLeaderboardList	GameData	GameData (constructor)	Verify default constructor creates empty leaderboard list	Default constructor	getLeaderboard().isEmpty() is true	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
GAMEDATA_011	GameDataTest	testParameterizedConstructorSetsPuzzles	GameData	GameData (constructor)	Verify parameterized constructor sets puzzles	List with 1 puzzle	getPuzzles() equals input list	Pass
GAMEDATA_012	GameDataTest	testParameterizedConstructorSetsHints	GameData	GameData (constructor)	Verify parameterized constructor sets hints	List with 1 hint	getHints() equals input list	Pass
GAMEDATA_013	GameDataTest	testParameterizedConstructorSetsUserProgress	GameData	GameData (constructor)	Verify parameterized constructor sets userProgress	List with 1 userProgress	getUserProgress() equals input list	Pass
GAMEDATA_014	GameDataTest	testParameterizedConstructorSetsCertificates	GameData	GameData (constructor)	Verify parameterized constructor sets certificates	List with 1 certificate	getCertificates() equals input list	Pass
GAMEDATA_015	GameDataTest	testParameterizedConstructorSetsLeaderboard	GameData	GameData (constructor)	Verify parameterized constructor sets leaderboard	List with 1 leaderboardEntry	getLeaderboard() equals input list	Pass
GAMEDATA_016	GameDataTest	testParameterizedConstructorHandlesNullPuzzles	GameData	GameData (constructor)	Verify parameterized constructor handles null puzzles	puzzles=null	getPuzzles() is non-null	Pass
GAMEDATA_017	GameDataTest	testParameterizedConstructorHandlesNullHints	GameData	GameData (constructor)	Verify parameterized constructor handles null hints	hints=null	getHints() is non-null	Pass
GAMEDATA_018	GameDataTest	testParameterizedConstructorHandlesNullUserProgress	GameData	GameData (constructor)	Verify parameterized constructor handles null userProgress	userProgress=null	getUserProgress() is non-null	Pass
GAMEDATA_019	GameDataTest	testParameterizedConstructorHandlesNullCertificates	GameData	GameData (constructor)	Verify parameterized constructor handles null certificates	certificates=null	getCertificates() is non-null	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
GAMEDATA_020	GameDataTest	testParameterizedConstructorHandlesNullLeaderboard	GameData	GameData (constructor)	Verify parameterized constructor handles null leaderboard	leaderboard=null	getLeaderboard() is non-null	Pass
GAMEDATA_021	GameDataTest	testSetPuzzles	GameData	setPuzzles	Verify setPuzzles updates puzzles list	List with 1 puzzle	getPuzzles() equals input list	Pass
GAMEDATA_022	GameDataTest	testSetHints	GameData	setHints	Verify setHints updates hints list	List with 1 hint	getHints() equals input list	Pass
GAMEDATA_023	GameDataTest	testSetUserProgress	GameData	setUserProgress	Verify setUserProgress updates userProgress list	List with 1 userProgress	getUserProgress() equals input list	Pass
GAMEDATA_024	GameDataTest	testSetCertificates	GameData	setCertificates	Verify setCertificates updates certificates list	List with 1 certificate	getCertificates() equals input list	Pass
GAMEDATA_025	GameDataTest	testSetLeaderboard	GameData	setLeaderboard	Verify setLeaderboard updates leaderboard list	List with 1 leaderboardEntry	getLeaderboard() equals input list	Pass
WRITER_001	GameDataWriterTest	testWriteUsersWithEmptyList	GameDataWriter	writeUsers	Verify writeUsers handles empty list	Empty user list	Returns true	Pass
WRITER_002	GameDataWriterTest	testWriteUsersCreatesFile	GameDataWriter	writeUsers	Verify writeUsers creates file	Empty user list	File exists	Pass
WRITER_003	GameDataWriterTest	testWriteUsersWithNonEmptyList	GameDataWriter	writeUsers	Verify writeUsers handles non-empty list	List with 1 user	Returns true	Pass
WRITER_004	GameDataWriterTest	testWriteUsersWithMultipleUsers	GameDataWriter	writeUsers	Verify writeUsers handles multiple users	List with 2 users	Returns true	Pass
WRITER_005	GameDataWriterTest	testWriteUsersCreatesValidJson	GameDataWriter	writeUsers	Verify writeUsers creates valid JSON	List with usr01	File content contains usr01	Pass
WRITER_006	GameDataWriterTest	testWriteGameDataWithEmptyData	GameDataWriter	writeGameData	Verify writeGameData handles empty data	Empty GameData	Returns true	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
WRITER_007	GameDataWriterTest	testWriteGameDataCreatesFile	GameDataWriter	writeGameData	Verify writeGameData creates file	Empty GameData	File exists	Pass
WRITER_008	GameDataWriterTest	testWriteGameDataWithPuzzles	GameDataWriter	writeGameData	Verify writeGameData handles puzzles	GameData with 1 puzzle	Returns true	Pass
WRITER_009	GameDataWriterTest	testWriteGameDataWithHints	GameDataWriter	writeGameData	Verify writeGameData handles hints	GameData with 1 hint	Returns true	Pass
WRITER_010	GameDataWriterTest	testWriteGameDataCreatesValidJson	GameDataWriter	writeGameData	Verify writeGameData creates valid JSON	GameData with Test Puzzle	File content contains Test Puzzle	Pass
WRITER_011	GameDataWriterTest	testWriteGameDataPreservesPuzzles	GameDataWriter	writeGameData	Verify writeGameData preserves multiple puzzles	GameData with 2 puzzles	File contains both Puzzle 1 and Puzzle 2	Pass
WRITER_012	GameDataWriterTest	testWriteGameDataPreservesHints	GameDataWriter	writeGameData	Verify writeGameData preserves multiple hints	GameData with 2 hints	File contains both Hint 1 and Hint 2	Pass
FACTORY_001	GameFactoryTest	testCreateGameReturnsMazeGameForMazeType	GameFactory	createGame	Verify createGame returns MazeGame for MAZE type	type=MAZE	Returns instanceof MazeGame	Pass
FACTORY_002	GameFactoryTest	testCreateGameReturnsMatchingGameForMatchingType	GameFactory	createGame	Verify createGame returns MatchingGame for MATCHING type	type=MATCHING	Returns instanceof MatchingGame	Pass
FACTORY_003	GameFactoryTest	testCreateGameReturnsWordPuzzleGameForCipherType	GameFactory	createGame	Verify createGame returns WordPuzzleGame for CIPHER type	type=CIPHER	Returns instanceof WordPuzzleGame	Pass
FACTORY_004	GameFactoryTest	testCreateGameReturnsWordPuzzleGameForAnagramType	GameFactory	createGame	Verify createGame returns WordPuzzleGame for ANAGRAM type	type=ANAGRAM	Returns instanceof WordPuzzleGame	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
FACTORY_005	GameFactoryTest	testCreateGameReturnsWordPuzzleGameForRiddleType	GameFactory	createGame	Verify createGame returns WordPuzzleGame for RIDDLE type	type=RIDDLE	Returns instanceof WordPuzzleGame	Pass
FACTORY_006	GameFactoryTest	testCreateGameIsCaseInsensitive	GameFactory	createGame	Verify createGame is case-insensitive	type=maze (lowercase)	Returns instanceof MazeGame	Pass
FACTORY_007	GameFactoryTest	testCreateGameThrowsExceptionForNullType	GameFactory	createGame	Verify createGame throws exception for null type	type=null	Throws IllegalArgumentException	Pass
FACTORY_008	GameFactoryTest	testCreateGameThrowsExceptionForUnknownType	GameFactory	createGame	Verify createGame throws exception for unknown type	type=UNKNOWN	Throws IllegalArgumentException	Pass
PROGRESS_001	GameProgressServiceTest	testGetUserProgressReturnsNonNull	GameProgressService	getUserProgress	Verify getUserProgress returns non-null	userId=usr01	Non-null UserProgress	Pass
PROGRESS_002	GameProgressServiceTest	testGetUserProgressCreatesNewForNewUser	GameProgressService	getUserProgress	Verify getUserProgress creates new for new user	userId=usr02	UserProgress.getUserId() equals usr02	Pass
PROGRESS_003	GameProgressServiceTest	testGetUserProgressReturnsSameInstanceForSameUser	GameProgressService	getUserProgress	Verify getUserProgress returns same instance	Call twice with usr03, compare instances	Same instance	Pass
PROGRESS_004	GameProgressServiceTest	testCompletePuzzleMarksAsCompleted	GameProgressService	completePuzzle	Verify completePuzzle marks puzzle as completed	Complete puzzle1 for usr04, check isPuzzleCompleted	isPuzzleCompleted returns true	Pass
PROGRESS_005	GameProgressServiceTest	testCompletePuzzleDoesNotAffectOtherPuzzles	GameProgressService	completePuzzle	Verify completePuzzle doesn't affect other puzzles	Complete puzzle1 for usr05, check puzzle2	isPuzzleCompleted(puzzle2) returns false	Pass
PROGRESS_006	GameProgressServiceTest	testCompletePuzzleStoresScore	GameProgressService	completePuzzle	Verify completePuzzle stores score	Complete puzzle1 with score 95 for usr06, get progress	PuzzleScores[puzzle1] equals 95	Pass
PROGRESS_007	GameProgressServiceTest	testIsPuzzleCompletedReturnsFalseForNewUser	GameProgressService	isPuzzleCompleted	Verify isPuzzleCompleted returns false for new user	New user usr07, check puzzle1	Returns false	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
PROGRESS_008	GameProgressServiceTest	testIsPuzzleCompletedReturnsTrueAfterCompletion	GameProgressService	isPuzzleCompleted	Verify isPuzzleCompleted returns true after completion	Complete puzzle1 for usr08, check	Returns true	Pass
PROGRESS_009	GameProgressServiceTest	testGetAvailablePuzzlesReturnsNonNull	GameProgressService	getAvailablePuzzles	Verify getAvailablePuzzles returns non-null	userId=usr09	Non-null List	Pass
PROGRESS_010	GameProgressServiceTest	testGetAvailablePuzzlesReturnsAllForNewUser	GameProgressService	getAvailablePuzzles	Verify getAvailablePuzzles returns all for new user	New user usr10	List size equals 3	Pass
PROGRESS_011	GameProgressServiceTest	testGetAvailablePuzzlesExcludesCompleted	GameProgressService	getAvailablePuzzles	Verify getAvailablePuzzles excludes completed	Complete puzzle1 for usr11, get available	List size equals 2	Pass
PROGRESS_012	GameProgressServiceTest	testGetAvailablePuzzlesReturnsEmptyWhenAllCompleted	GameProgressService	getAvailablePuzzles	Verify getAvailablePuzzles returns empty when all completed	Complete all 3 puzzles for usr12, get available	Empty list	Pass
PROGRESS_013	GameProgressServiceTest	testGetCompletedPuzzlesReturnsNonNull	GameProgressService	getCompletedPuzzles	Verify getCompletedPuzzles returns non-null	userId=usr13	Non-null List	Pass
PROGRESS_014	GameProgressServiceTest	testGetCompletedPuzzlesReturnsEmptyForNewUser	GameProgressService	getCompletedPuzzles	Verify getCompletedPuzzles returns empty for new user	New user usr14	Empty list	Pass
PROGRESS_015	GameProgressServiceTest	testGetCompletedPuzzlesReturnsCompleted	GameProgressService	getCompletedPuzzles	Verify getCompletedPuzzles returns completed puzzles	Complete puzzle1 for usr15, get completed	List size equals 1	Pass
PROGRESS_016	GameProgressServiceTest	testGetCompletedPuzzlesReturnsAllCompleted	GameProgressService	getCompletedPuzzles	Verify getCompletedPuzzles returns all completed	Complete puzzle1 and puzzle2 for usr16, get completed	List size equals 2	Pass
PROGRESS_017	GameProgressServiceTest	testGetProgressStatsReturnsNonNull	GameProgressService	getProgressStats	Verify getProgressStats returns non-null	userId=usr17	Non-null Map	Pass
PROGRESS_018	GameProgressServiceTest	testGetProgressStatsContainsTotalPuzzles	GameProgressService	getProgressStats	Verify getProgressStats contains totalPuzzles	userId=usr18	Stats[totalPuzzles] equals 3	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
PROGRESS_019	GameProgressServiceTest	testGetProgressStatsContainsCompleted	GameProgressService	getProgressStats	Verify getProgressStats contains completed count	Complete puzzle1 for usr19, get stats	Stats[completed] equals 1	Pass
PROGRESS_020	GameProgressServiceTest	testGetProgressStatsContainsRemaining	GameProgressService	getProgressStats	Verify getProgressStats contains remaining count	Complete puzzle1 for usr20, get stats	Stats[remaining] equals 2	Pass
PROGRESS_021	GameProgressServiceTest	testGetProgressStatsContainsTotalScore	GameProgressService	getProgressStats	Verify getProgressStats contains totalScore	Complete puzzle1 (100) and puzzle2 (95) for usr21, get stats	Stats[totalScore] equals 195	Pass
PROGRESS_022	GameProgressServiceTest	testGetProgressStatsCalculatesCompletionPercentage	GameProgressService	getProgressStats	Verify getProgressStats calculates completion percentage	Complete 1 of 3 puzzles for usr22, get stats	Stats[completionPercentage] equals 33	Pass
PROGRESS_023	GameProgressServiceTest	testGetProgressStatsCalculatesFullCompletion	GameProgressService	getProgressStats	Verify getProgressStats calculates 100% completion	Complete all 3 puzzles for usr23, get stats	Stats[completionPercentage] equals 100	Pass
HINT_001	HintTest	testConstructorSetsHintText	Hint	Hint (constructor)	Verify constructor sets hintText	hintText="Look for the exit on the right side", puzzleId="P001", hintPriority=1	getHintText() returns correct text	Pass
HINT_002	HintTest	testConstructorSetsPuzzleId	Hint	Hint (constructor)	Verify constructor sets puzzleId	Full constructor parameters	getPuzzleId() returns P001	Pass
HINT_003	HintTest	testConstructorSetsHintPriority	Hint	Hint (constructor)	Verify constructor sets hintPriority	Full constructor parameters	getHintPriority() returns 1	Pass
HINT_004	HintTest	testSetHintText	Hint	setHintText	Verify setHintText updates hint text	setHintText(New hint)	getHintText() returns New hint	Pass
HINT_005	HintTest	testSetPuzzleId	Hint	setPuzzleId	Verify setPuzzleId updates puzzleId	setPuzzleId(P002)	getPuzzleId() returns P002	Pass
HINT_006	HintTest	testSetHintPriority	Hint	setHintPriority	Verify setHintPriority updates priority	setHintPriority(2)	getHintPriority() returns 2	Pass
HINT_007	HintTest	testCompareToOrdersByPriority	Hint	compareTo	Verify compareTo orders by priority ascending	hint1 priority=1, hint2 priority=2	compareTo returns negative	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
HINT_008	HintTest	testCompareToReturnsZeroForSamePriority	Hint	compareTo	Verify compareTo returns zero for same priority	hint1 priority=1, hint2 priority=1	compareTo returns 0	Pass
HINT_009	HintTest	testToStringContainsHintText	Hint	toString	Verify toString contains hint text	toString()	Contains hint text	Pass
HINT_010	HintTest	testToStringContainsPriority	Hint	toString	Verify toString contains priority	toString()	Contains priority value	Pass
LEADER_001	LeaderboardEntryTest	testConstructorSetsUserId	LeaderboardEntry	LeaderboardEntry (constructor)	Verify constructor sets userId	userId="user1", userName="John Doe", totalScore=500, puzzlesCompleted=5	getUserId() returns user1	Pass
LEADER_002	LeaderboardEntryTest	testConstructorSetsUserName	LeaderboardEntry	LeaderboardEntry (constructor)	Verify constructor sets userName	Full constructor parameters	getUserName() returns John Doe	Pass
LEADER_003	LeaderboardEntryTest	testConstructorSetsTotalScore	LeaderboardEntry	LeaderboardEntry (constructor)	Verify constructor sets totalScore	Full constructor parameters	getTotalScore() returns 500	Pass
LEADER_004	LeaderboardEntryTest	testConstructorSetsPuzzlesCompleted	LeaderboardEntry	LeaderboardEntry (constructor)	Verify constructor sets puzzlesCompleted	Full constructor parameters	getPuzzlesCompleted() returns 5	Pass
LEADER_005	LeaderboardEntryTest	testConstructorSetsLastUpdated	LeaderboardEntry	LeaderboardEntry (constructor)	Verify constructor sets lastUpdated	Full constructor parameters	getLastUpdated() is non-null	Pass
LEADER_006	LeaderboardEntryTest	testDefaultConstructorSetsLastUpdated	LeaderboardEntry	LeaderboardEntry (constructor)	Verify default constructor sets lastUpdated	Default constructor	getLastUpdated() is non-null	Pass
LEADER_007	LeaderboardEntryTest	testSetUserId	LeaderboardEntry	setUserId	Verify setUserId updates userId	setUserId(user2)	getUserId() returns user2	Pass
LEADER_008	LeaderboardEntryTest	testSetUserName	LeaderboardEntry	setUserName	Verify setUserName updates userName	setUserName(Jane Smith)	getUserName() returns Jane Smith	Pass
LEADER_009	LeaderboardEntryTest	testSetTotalScore	LeaderboardEntry	setTotalScore	Verify setTotalScore updates totalScore	setTotalScore(1000)	getTotalScore() returns 1000	Pass
LEADER_010	LeaderboardEntryTest	testSetPuzzlesCompleted	LeaderboardEntry	setPuzzlesCompleted	Verify setPuzzlesCompleted updates puzzlesCompleted	setPuzzlesCompleted(10)	getPuzzlesCompleted() returns 10	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
LEADER_011	LeaderboardEntryTest	testSetLastUpdated	LeaderboardEntry	setLastUpdated	Verify setLastUpdated updates lastUpdated	setLastUpdated(testTime)	getLastUpdated() equals testTime	Pass
LEADER_012	LeaderboardEntryTest	testToStringContainsTotalScore	LeaderboardEntry	toString	Verify toString contains totalScore	toString()	Contains 500	Pass
LEADERSVC_001	LeaderboardServiceTest	testGetTopPlayersReturnsNonNull	LeaderboardService	getTopPlayers	Verify getTopPlayers returns non-null	limit=10	Non-null List	Pass
LEADERSVC_002	LeaderboardServiceTest	testGetTopPlayersReturnsEmptyInitially	LeaderboardService	getTopPlayers	Verify getTopPlayers returns empty initially	limit=10	Empty list	Pass
LEADERSVC_003	LeaderboardServiceTest	testGetTopPlayersRespectsLimit	LeaderboardService	getTopPlayers	Verify getTopPlayers respects limit	Create 10 users, get top 5	List size equals 5	Pass
LEADERSVC_004	LeaderboardServiceTest	testGetTopPlayersReturnsSingleEntry	LeaderboardService	getTopPlayers	Verify getTopPlayers returns single entry	Create 1 user with progress, get top 10	List size equals 1	Pass
LEADERSVC_005	LeaderboardServiceTest	testGetTopPlayersReturnsMultipleEntries	LeaderboardService	getTopPlayers	Verify getTopPlayers returns multiple entries	Create 3 users with progress, get top 10	List size equals 3	Pass
LEADERSVC_006	LeaderboardServiceTest	testGetTopPlayersReturnsSortedByScore	LeaderboardService	getTopPlayers	Verify getTopPlayers returns sorted by score descending	Create users with scores 70, 100, 85, get top 10	First entry has highest score	Pass
LEADERSVC_007	LeaderboardServiceTest	testGetUserRankReturnsNegativeForNonexistentUser	LeaderboardService	getUserRank	Verify getUserRank returns -1 for non-existent user	userId=nonexistent	Returns -1	Pass
LEADERSVC_008	LeaderboardServiceTest	testGetUserRankReturnsOneForFirstPlace	LeaderboardService	getUserRank	Verify getUserRank returns 1 for first place	Create 1 user usr08 with score 100, get rank	Returns 1	Pass
LEADERSVC_009	LeaderboardServiceTest	testGetUserRankReturnsCorrectRankForMultipleUsers	LeaderboardService	getUserRank	Verify getUserRank returns correct rank	Create 3 users with scores 100, 95, 88, get rank of second user	Returns 2	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
LEADERSVC_010	LeaderboardServiceTest	testGetUserRankReturnsThirdPlace	LeaderboardService	getUserRank	Verify getUserRank returns 3 for third place	Create 3 users with scores 100, 95, 88, get rank of third user	Returns 3	Pass
LEADERSVC_011	LeaderboardServiceTest	testGetUserRankUpdatesAfterNewScore	LeaderboardService	getUserRank	Verify getUserRank updates after new score	Create usr15 with score 90, then add usr16 with score 100, get usr15 rank	Rank increases (number goes up)	Pass
LEADERSVC_012	LeaderboardServiceTest	testGetUserEntryReturnsNullForNonexistentUser	LeaderboardService	getUserEntry	Verify getUserEntry returns null for non-existent user	userId=nonexistent	Returns null	Pass
LEADERSVC_013	LeaderboardServiceTest	testGetUserEntryReturnsEntryAfterProgress	LeaderboardService	getUserEntry	Verify getUserEntry returns entry after progress	Create usr17 with progress, get entry	Non-null LeaderboardEntry	Pass
LEADERSVC_014	LeaderboardServiceTest	testGetUserEntryReturnsCorrectUserId	LeaderboardService	getUserEntry	Verify getUserEntry returns correct userId	Create usr18 with progress, get entry	Entry.getUserId() equals usr18	Pass
LEADERSVC_015	LeaderboardServiceTest	testGetUserEntryReturnsCorrectScore	LeaderboardService	getUserEntry	Verify getUserEntry returns correct score	Create usr19 with score 95, get entry	Entry.getTotalScore() equals 95	Pass
LEADERSVC_016	LeaderboardServiceTest	testGetUserEntryReturnsCorrectPuzzleCount	LeaderboardService	getUserEntry	Verify getUserEntry returns correct puzzle count	Create usr20 with 1 completed puzzle, get entry	Entry.getPuzzlesCompleted() equals 1	Pass
LEADERSVC_017	LeaderboardServiceTest	testGetFullLeaderboardReturnsNonNull	LeaderboardService	getFullLeaderboard	Verify getFullLeaderboard returns non-null	Call getFullLeaderboard()	Non-null List	Pass
LEADERSVC_018	LeaderboardServiceTest	testGetFullLeaderboardReturnsEmptyInitially	LeaderboardService	getFullLeaderboard	Verify getFullLeaderboard returns empty initially	Call getFullLeaderboard()	Empty list	Pass
LEADERSVC_019	LeaderboardServiceTest	testGetFullLeaderboardReturnsAllEntries	LeaderboardService	getFullLeaderboard	Verify getFullLeaderboard returns all entries	Create 2 users with progress, get leaderboard	List size equals 2	Pass
LEADERSVC_020	LeaderboardServiceTest	testGetFullLeaderboardReturnsManyEntries	LeaderboardService	getFullLeaderboard	Verify getFullLeaderboard returns many entries	Create 20 users with progress, get leaderboard	List size equals 20	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
LEADERSVC_021	LeaderboardServiceTest	testGetFullLeaderboardReturnsSortedEntries	LeaderboardService	getFullLeaderboard	Verify getFullLeaderboard returns sorted entries	Create 3 users with scores 60, 100, 80, get leaderboard	Sorted in descending order	Pass
MATCHING_001	MatchingGameTest	testInitializeCreatesBoard	MatchingGame	initialize	Verify initialize creates game board	puzzleData with 2x2 grid	GameState contains board	Pass
MATCHING_002	MatchingGameTest	testInitializeCreatesMatchedArray	MatchingGame	initialize	Verify initialize creates matched array	puzzleData with 2x2 grid	GameState contains matched array	Pass
MATCHING_003	MatchingGameTest	testInitializeSetsGameType	MatchingGame	initialize	Verify initialize sets game type to MATCHING	puzzleData with 2x2 grid	getGameType() returns MATCHING	Pass
MATCHING_004	MatchingGameTest	testInitializeSetsZeroMoveCount	MatchingGame	initialize	Verify initialize sets move count to zero	puzzleData with 2x2 grid	GameState[moveCount] equals 0	Pass
MATCHING_005	MatchingGameTest	testInitializeNoCardsSelected	MatchingGame	initialize	Verify initialize has no cards selected	puzzleData with 2x2 grid	firstCard and secondCard are null	Pass
MATCHING_006	MatchingGameTest	testInitializeNotShowingPair	MatchingGame	initialize	Verify initialize sets showingPair to false	puzzleData with 2x2 grid	isShowingPair() returns false	Pass
MATCHING_007	MatchingGameTest	testProcessInputWithValidCoordinates	MatchingGame	processInput	Verify processInput accepts valid coordinates	Initialize game, input "0 0"	Returns true	Pass
MATCHING_008	MatchingGameTest	testProcessInputSelectsFirstCard	MatchingGame	processInput	Verify processInput selects first card	Initialize game, input "0 0"	GameState[firstCard] is non-null	Pass
MATCHING_009	MatchingGameTest	testProcessInputSelectsSecondCard	MatchingGame	processInput	Verify processInput selects second card	Initialize game, input "0 0" then "0 1"	GameState[secondCard] is non-null	Pass
MATCHING_010	MatchingGameTest	testProcessInputWithNullReturnsFalse	MatchingGame	processInput	Verify processInput rejects null input	Initialize game, input null	Returns false	Pass
MATCHING_011	MatchingGameTest	testProcessInputWithInvalidFormatReturnsFalse	MatchingGame	processInput	Verify processInput rejects invalid format	Initialize game, input "abc"	Returns false	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
MATCHING_012	MatchingGameTest	testProcessInputWithOutOfBoundsReturnsFalse	MatchingGame	processInput	Verify processInput rejects out of bounds coordinates	Initialize game, input "5 5"	Returns false	Pass
MATCHING_013	MatchingGameTest	testProcessInputWithNegativeCoordinatesReturnsFalse	MatchingGame	processInput	Verify processInput rejects negative coordinates	Initialize game, input "-1 0"	Returns false	Pass
MATCHING_014	MatchingGameTest	testProcessInputWithSameCardTwiceReturnsFalse	MatchingGame	processInput	Verify processInput rejects same card selection twice	Initialize game, select "0 0" twice	Second input returns false	Pass
MATCHING_015	MatchingGameTest	testProcessInputSecondCardSetsShowingPair	MatchingGame	processInput	Verify second card selection sets showingPair	Initialize game, select "0 0" then "0 1"	isShowingPair() returns true	Pass
MATCHING_016	MatchingGameTest	testProcessInputSecondCardIncrementsMoveCount	MatchingGame	processInput	Verify second card selection increments move count	Initialize game, select "0 0" then "0 1"	GameState[moveCount] equals 1	Pass
MATCHING_017	MatchingGameTest	testProcessInputWithMatchedCardReturnsFalse	MatchingGame	processInput	Verify processInput rejects already matched card	Initialize game, match a pair, try to select matched card	Returns false	Pass
MATCHING_018	MatchingGameTest	testClearSelectionClearsFirstCard	MatchingGame	clearSelection	Verify clearSelection clears first card	Select 2 cards, call clearSelection	GameState[firstCard] is null	Pass
MATCHING_019	MatchingGameTest	testClearSelectionClearsSecondCard	MatchingGame	clearSelection	Verify clearSelection clears second card	Select 2 cards, call clearSelection	GameState[secondCard] is null	Pass
MATCHING_020	MatchingGameTest	testClearSelectionClearsShowingPair	MatchingGame	clearSelection	Verify clearSelection clears showingPair	Select 2 cards, call clearSelection	isShowingPair() returns false	Pass
MATCHING_021	MatchingGameTest	testIsGameOverReturnsFalseInitially	MatchingGame	isGameOver	Verify isGameOver returns false initially	Initialize game	Returns false	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
MATCHING_022	MatchingGameTest	testIsGameOverReturnsTrueWhenAllMatched	MatchingGame	isGameOver	Verify isGameOver returns true when all matched	Match all pairs	Returns true	Pass
MATCHING_023	MatchingGameTest	testGetGameStateReturnsNonNull	MatchingGame	getGameState	Verify getGameState returns non-null	Initialize game	Non-null Map	Pass
MATCHING_024	MatchingGameTest	testGetGameStateContainsBoard	MatchingGame	getGameState	Verify getGameState contains board	Initialize game	State contains board key	Pass
MATCHING_025	MatchingGameTest	testGetGameStateContainsMatched	MatchingGame	getGameState	Verify getGameState contains matched array	Initialize game	State contains matched key	Pass
MATCHING_026	MatchingGameTest	testGetGameStateContainsMoveCount	MatchingGame	getGameState	Verify getGameState contains moveCount	Initialize game	State contains moveCount key	Pass
MATCHING_027	MatchingGameTest	testGetResultReturnsNonNull	MatchingGame	getResult	Verify getResult returns non-null	Initialize game	Non-null Map	Pass
MATCHING_028	MatchingGameTest	testGetResultContainsWonStatus	MatchingGame	getResult	Verify getResult contains won status	Initialize game	Result contains won key	Pass
MATCHING_029	MatchingGameTest	testGetResultContainsTime	MatchingGame	getResult	Verify getResult contains time	Initialize game	Result contains time key	Pass
MATCHING_030	MatchingGameTest	testGetResultContainsMoves	MatchingGame	getResult	Verify getResult contains moves	Initialize game	Result contains moves key	Pass
MATCHING_031	MatchingGameTest	testGetResultShowsNotWonInitially	MatchingGame	getResult	Verify getResult shows not won initially	Initialize game	Result[won] equals false	Pass
MATCHING_032	MatchingGameTest	testResetClearsMoveCount	MatchingGame	reset	Verify reset clears move count	Initialize game, make moves, reset	GameState[moveCount] equals 0	Pass
MATCHING_033	MatchingGameTest	testResetClearsFirstCard	MatchingGame	reset	Verify reset clears first card	Initialize game, select card, reset	GameState[firstCard] is null	Pass
MATCHING_034	MatchingGameTest	testResetClearsSecondCard	MatchingGame	reset	Verify reset clears second card	Initialize game, select 2 cards, reset	GameState[secondCard] is null	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
MATCHING_035	MatchingGameTest	testResetClearsShowingPair	MatchingGame	reset	Verify reset clears showingPair	Initialize game, select 2 cards, reset	isShowingPair() returns false	Pass
MATCHING_036	MatchingGameTest	testSaveStateReturnsNonNull	MatchingGame	saveState	Verify saveState returns non-null	Initialize game	Non-null Map	Pass
MATCHING_037	MatchingGameTest	testSaveStateContainsRows	MatchingGame	saveState	Verify saveState contains rows	Initialize game	SavedState contains rows key	Pass
MATCHING_038	MatchingGameTest	testSaveStateContainsCols	MatchingGame	saveState	Verify saveState contains cols	Initialize game	SavedState contains cols key	Pass
MATCHING_039	MatchingGameTest	testSaveStateContainsBoard	MatchingGame	saveState	Verify saveState contains board	Initialize game	SavedState contains board key	Pass
MATCHING_040	MatchingGameTest	testSaveStateContainsMatched	MatchingGame	saveState	Verify saveState contains matched array	Initialize game	SavedState contains matched key	Pass
MATCHING_041	MatchingGameTest	testSaveStateContainsMoveCount	MatchingGame	saveState	Verify saveState contains moveCount	Initialize game	SavedState contains moveCount key	Pass
MATCHING_042	MatchingGameTest	testSaveStatePreservesMoveCount	MatchingGame	saveState	Verify saveState preserves moveCount	Initialize game, make 1 move, save	SavedState[moveCount] equals 1	Pass
MATCHING_043	MatchingGameTest	testRestoreStateRestoresMoveCount	MatchingGame	restoreState	Verify restoreState restores moveCount	Make 1 move, save, restore to new game	New game moveCount equals 1	Pass
MATCHING_044	MatchingGameTest	testRestoreStateRestoresBoard	MatchingGame	restoreState	Verify restoreState restores board	Save state, restore to new game	New game has non-null board	Pass
MATCHING_045	MatchingGameTest	testRestoreStatePreservesGameType	MatchingGame	restoreState	Verify restoreState preserves game type	Save state, restore to new game	New game getGameType() returns MATCHING	Pass
MATCHING_046	MatchingGameTest	testRestoreStateWithNullFirstCard	MatchingGame	restoreState	Verify restoreState handles null firstCard	Save state with no cards selected, restore	New game firstCard is null	Pass
MATCHING_047	MatchingGameTest	testRestoreStateAllowsContinuingGame	MatchingGame	restoreState	Verify restoreState allows continuing game	Select 1 card, save, restore, select 2nd card	Second selection returns true	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
MATCHING_048	MatchingGameTest	testGetGameTypeReturnsMatching	MatchingGame	getGameType	Verify getGameType returns MATCHING	Initialize game	Returns MATCHING	Pass
MAZE_001	MazeGameTest	testInitializeCreatesMaze	MazeGame	initialize	Verify initialize creates maze	puzzleData with 6x6 maze	GameState contains maze	Pass
MAZE_002	MazeGameTest	testInitializeCreatesPlayer	MazeGame	initialize	Verify initialize creates player	puzzleData with 6x6 maze	GameState contains player	Pass
MAZE_003	MazeGameTest	testInitializeSetsGameType	MazeGame	initialize	Verify initialize sets game type to MAZE	puzzleData with 6x6 maze	getGameType() returns MAZE	Pass
MAZE_004	MazeGameTest	testInitializeSetsZeroMoveCount	MazeGame	initialize	Verify initialize sets move count to zero	puzzleData with 6x6 maze	GameState[moveCount] equals 0	Pass
MAZE_005	MazeGameTest	testInitializePlacesPlayerAtStart	MazeGame	initialize	Verify initialize places player at start position	puzzleData with start (1,1)	Player at row=1, col=1	Pass
MAZE_006	MazeGameTest	testInitializeNotGameOver	MazeGame	initialize	Verify initialize sets game over to false	puzzleData with 6x6 maze	isGameOver() returns false	Pass
MAZE_007	MazeGameTest	testProcessInputWithValidMoveDown	MazeGame	processInput	Verify processInput accepts valid down move	Initialize game, input "S"	Returns true	Pass
MAZE_008	MazeGameTest	testProcessInputWithValidMoveRight	MazeGame	processInput	Verify processInput accepts valid right move	Initialize game, input "D"	Returns true	Pass
MAZE_009	MazeGameTest	testProcessInputMovesPlayerDown	MazeGame	processInput	Verify processInput moves player down	Initialize game at (1,1), input "S"	Player at row=2, col=1	Pass
MAZE_010	MazeGameTest	testProcessInputMovesPlayerRight	MazeGame	processInput	Verify processInput moves player right	Initialize game at (1,1), input "D"	Player at row=1, col=2	Pass
MAZE_011	MazeGameTest	testProcessInputWithLowercaseCommand	MazeGame	processInput	Verify processInput accepts lowercase commands	Initialize game, input "d"	Returns true	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
MAZE_012	MazeGameTest	testProcessInputIncrementsMoveCount	MazeGame	processInput	Verify processInput increments move count	Initialize game, input "D"	GameState[moveCount] equals 1	Pass
MAZE_013	MazeGameTest	testProcessInputWithMultipleMoves	MazeGame	processInput	Verify processInput handles multiple moves	Initialize game, input "D" twice	Player at row=1, col=3	Pass
MAZE_014	MazeGameTest	testProcessInputWithNullReturnsFalse	MazeGame	processInput	Verify processInput rejects null input	Initialize game, input null	Returns false	Pass
MAZE_015	MazeGameTest	testProcessInputWithInvalidCommandReturnsFalse	MazeGame	processInput	Verify processInput rejects invalid command	Initialize game, input "X"	Returns false	Pass
MAZE_016	MazeGameTest	testProcessInputOutOfBoundsUpReturnsFalse	MazeGame	processInput	Verify processInput rejects out of bounds up	Initialize game at (1,1), input "W" (wall above)	Returns false	Pass
MAZE_017	MazeGameTest	testProcessInputOutOfBoundsLeftReturnsFalse	MazeGame	processInput	Verify processInput rejects out of bounds left	Initialize game at (1,1), input "A" (wall left)	Returns false	Pass
MAZE_018	MazeGameTest	testProcessInputDoesNotChangeMoveCountOnInvalidMove	MazeGame	processInput	Verify invalid move doesn't increment move count	Initialize game, input "W" (invalid)	GameState[moveCount] equals 0	Pass
MAZE_019	MazeGameTest	testProcessInputDoesNotMovePlayerOnInvalidMove	MazeGame	processInput	Verify invalid move doesn't move player	Initialize game at (1,1), input "W" (invalid)	Player remains at row=1, col=1	Pass
MAZE_020	MazeGameTest	testProcessInputIntoWallReturnsFalse	MazeGame	processInput	Verify processInput rejects wall collision	Move to (2,1), try to move right into wall at (2,2)	Returns false	Pass
MAZE_021	MazeGameTest	testProcessInputIntoWallDoesNotMovePlayer	MazeGame	processInput	Verify wall collision doesn't move player	Move to (2,1), try wall collision	Player remains at row=2, col=1	Pass
MAZE_022	MazeGameTest	testProcessInputIntoWallDoesNotIncrementMoveCount	MazeGame	processInput	Verify wall collision doesn't increment move count	Move once to (2,1), try wall collision	GameState[moveCount] equals 1 (not 2)	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
MAZE_023	MazeGameTest	testMultipleWallCollisionsDoNotAffectState	MazeGame	processInput	Verify multiple wall collisions don't affect state	Move to (2,1), try wall collision 3 times	Player at row=2, col=1, moveCount=1	Pass
MAZE_024	MazeGameTest	testCanNavigateAroundWalls	MazeGame	processInput	Verify player can navigate around walls	Navigate (1,1) -> (1,2) -> (1,3)	Player at row=1, col=3, moveCount=2	Pass
MAZE_025	MazeGameTest	testIsGameOverReturnsFalseInitially	MazeGame	isGameOver	Verify isGameOver returns false initially	Initialize game	Returns false	Pass
MAZE_026	MazeGameTest	testIsGameOverReturnsTrueWhenAtEnd	MazeGame	isGameOver	Verify isGameOver returns true when at end	Navigate to end position (4, 4)	Returns true	Pass
MAZE_027	MazeGameTest	testIsGameOverReturnsFalseNearEnd	MazeGame	isGameOver	Verify isGameOver returns false near end	Navigate to (3, 4) - one step before end	Returns false	Pass
MAZE_028	MazeGameTest	testGetGameStateReturnsNonNull	MazeGame	getGameState	Verify getGameState returns non-null	Initialize game	Non-null Map	Pass
MAZE_029	MazeGameTest	testGetGameStateContainsMaze	MazeGame	getGameState	Verify getGameState contains maze	Initialize game	State contains maze key	Pass
MAZE_030	MazeGameTest	testGetGameStateContainsPlayer	MazeGame	getGameState	Verify getGameState contains player	Initialize game	State contains player key	Pass
MAZE_031	MazeGameTest	testGetGameStateContainsMoveCount	MazeGame	getGameState	Verify getGameState contains moveCount	Initialize game	State contains moveCount key	Pass
MAZE_032	MazeGameTest	testGetResultReturnsNonNull	MazeGame	getResult	Verify getResult returns non-null	Initialize game	Non-null Map	Pass
MAZE_033	MazeGameTest	testGetResultContainsWonStatus	MazeGame	getResult	Verify getResult contains won status	Initialize game	Result contains won key	Pass
MAZE_034	MazeGameTest	testGetResultContainsTime	MazeGame	getResult	Verify getResult contains time	Initialize game	Result contains time key	Pass
MAZE_035	MazeGameTest	testGetResultContainsMoves	MazeGame	getResult	Verify getResult contains moves	Initialize game	Result contains moves key	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
MAZE_036	MazeGameTest	testGetResultShowsNotWonInitially	MazeGame	getResult	Verify getResult shows not won initially	Initialize game	Result[won] equals false	Pass
MAZE_037	MazeGameTest	testGetResultShowsWonAfterReachingEnd	MazeGame	getResult	Verify getResult shows won after reaching end	Navigate to end position	Result[won] equals true	Pass
MAZE_038	MazeGameTest	testGetResultShowsCorrectMoveCount	MazeGame	getResult	Verify getResult shows correct move count	Make 2 moves, get result	Result[moves] equals 2	Pass
MAZE_039	MazeGameTest	testResetClearsMoveCount	MazeGame	reset	Verify reset clears move count	Make 2 moves, reset	GameState[moveCount] equals 0	Pass
MAZE_040	MazeGameTest	testResetMovesPlayerToStart	MazeGame	reset	Verify reset moves player to start	Move to (2,1), reset	Player at row=1, col=1	Pass
MAZE_041	MazeGameTest	testResetSetsNotGameOver	MazeGame	reset	Verify reset sets game over to false	Navigate to end, reset	isGameOver() returns false	Pass
MAZE_042	MazeGameTest	testSaveStateReturnsNonNull	MazeGame	saveState	Verify saveState returns non-null	Initialize game	Non-null Map	Pass
MAZE_043	MazeGameTest	testSaveStateContainsMaze	MazeGame	saveState	Verify saveState contains maze	Initialize game	SavedState contains maze key	Pass
MAZE_044	MazeGameTest	testSaveStateContainsPlayer	MazeGame	saveState	Verify saveState contains player	Initialize game	SavedState contains player key	Pass
MAZE_045	MazeGameTest	testSaveStateContainsMoveCount	MazeGame	saveState	Verify saveState contains moveCount	Initialize game	SavedState contains moveCount key	Pass
MAZE_046	MazeGameTest	testSaveStateContainsStartTime	MazeGame	saveState	Verify saveState contains startTime	Initialize game	SavedState contains startTime key	Pass
MAZE_047	MazeGameTest	testSaveStatePreservesMoveCount	MazeGame	saveState	Verify saveState preserves moveCount	Make 2 moves, save	SavedState[moveCount] equals 2	Pass
MAZE_048	MazeGameTest	testSaveStatePreservesPlayerPosition	MazeGame	saveState	Verify saveState preserves player position	Move to (2,1), save	SavedState player at row=2, col=1	Pass
MAZE_049	MazeGameTest	testRestoreStateRestoresMoveCount	MazeGame	restoreState	Verify restoreState restores moveCount	Make 2 moves, save, restore to new game	New game moveCount equals 2	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
MAZE_050	MazeGameTest	testRestoreStateRestoresPlayerPosition	MazeGame	restoreState	Verify restoreState restores player position	Move to (2,1), save, restore to new game	New game player at row=2, col=1	Pass
MAZE_051	MazeGameTest	testRestoreStatePreservesGameType	MazeGame	restoreState	Verify restoreState preserves game type	Save state, restore to new game	New game getGameType() returns MAZE	Pass
MAZE_052	MazeGameTest	testRestoreStateAllowsContinuingGame	MazeGame	restoreState	Verify restoreState allows continuing game	Make 1 move, save, restore, make another move	Second move returns true	Pass
MAZE_053	MazeGameTest	testRestoreStatePreservesGameOverStatus	MazeGame	restoreState	Verify restoreState preserves game over status	Navigate to end, save, restore to new game	New game isGameOver() returns true	Pass
MAZE_054	MazeGameTest	testGetGameTypeReturnsMaze	MazeGame	getGameType	Verify getGameType returns MAZE	Initialize game	Returns MAZE	Pass
MAZEENT_001	MazeTest	testDefaultConstructorCreatesInstance	Maze	Maze (constructor)	Verify default constructor creates instance	Default constructor	Non-null Maze	Pass
MAZEENT_002	MazeTest	testParameterizedConstructorSetsWidth	Maze	Maze (constructor)	Verify parameterized constructor sets width	width=5, height=5, mazeData, start, end	getWidth() returns 5	Pass
MAZEENT_003	MazeTest	testParameterizedConstructorSetsHeight	Maze	Maze (constructor)	Verify parameterized constructor sets height	Full constructor parameters	getHeight() returns 5	Pass
MAZEENT_004	MazeTest	testParameterizedConstructorSetsMazeData	Maze	Maze (constructor)	Verify parameterized constructor sets mazeData	Full constructor parameters	getMazeData() equals input array	Pass
MAZEENT_005	MazeTest	testParameterizedConstructorSetsStart	Maze	Maze (constructor)	Verify parameterized constructor sets start	Full constructor parameters	getStart() equals start Position	Pass
MAZEENT_006	MazeTest	testParameterizedConstructorSetsEnd	Maze	Maze (constructor)	Verify parameterized constructor sets end	Full constructor parameters	getEnd() equals end Position	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
MAZEENT_007	MazeTest	testGetWidthReturnsCorrectValue	Maze	getWidth	Verify getWidth returns correct value	width=5	Returns 5	Pass
MAZEENT_008	MazeTest	testGetHeightReturnsCorrectValue	Maze	getHeight	Verify getHeight returns correct value	height=5	Returns 5	Pass
MAZEENT_009	MazeTest	testGetMazeDataReturnsCorrectArray	Maze	getMazeData	Verify getMazeData returns correct array	5x5 maze array	Returns same array	Pass
MAZEENT_010	MazeTest	testGetMazeDataReturnsNonNull	Maze	getMazeData	Verify getMazeData returns non-null	Valid maze	Non-null array	Pass
MAZEENT_011	MazeTest	testGetStartReturnsCorrectPosition	Maze	getStart	Verify getStart returns correct position	start Position (0, 0)	Returns same Position	Pass
MAZEENT_012	MazeTest	testGetStartReturnsNonNull	Maze	getStart	Verify getStart returns non-null	Valid maze	Non-null Position	Pass
MAZEENT_013	MazeTest	testGetEndReturnsCorrectPosition	Maze	getEnd	Verify getEnd returns correct position	end Position (4, 4)	Returns same Position	Pass
MAZEENT_014	MazeTest	testGetEndReturnsNonNull	Maze	getEnd	Verify getEnd returns non-null	Valid maze	Non-null Position	Pass
MAZEENT_015	MazeTest	testSetWidthUpdatesWidth	Maze	setWidth	Verify setWidth updates width	setWidth(10)	getWidth() returns 10	Pass
MAZEENT_016	MazeTest	testSetWidthAcceptsZero	Maze	setWidth	Verify setWidth accepts zero	setWidth(0)	getWidth() returns 0	Pass
MAZEENT_017	MazeTest	testSetWidthAcceptsLargeValue	Maze	setWidth	Verify setWidth accepts large value	setWidth(100)	getWidth() returns 100	Pass
MAZEENT_018	MazeTest	testSetHeightUpdatesHeight	Maze	setHeight	Verify setHeight updates height	setHeight(10)	getHeight() returns 10	Pass
MAZEENT_019	MazeTest	testSetHeightAcceptsZero	Maze	setHeight	Verify setHeight accepts zero	setHeight(0)	getHeight() returns 0	Pass
MAZEENT_020	MazeTest	testSetHeightAcceptsLargeValue	Maze	setHeight	Verify setHeight accepts large value	setHeight(100)	getHeight() returns 100	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
MAZEENT_021	MazeTest	testSetMazeDataUpdatesData	Maze	setMazeData	Verify setMazeData updates data	2x2 new array	getMazeData() equals new array	Pass
MAZEENT_022	MazeTest	testSetMazeDataAcceptsNull	Maze	setMazeData	Verify setMazeData accepts null	setMazeData(null)	getMazeData() returns null	Pass
MAZEENT_023	MazeTest	testSetMazeDataAcceptsEmptyArray	Maze	setMazeData	Verify setMazeData accepts empty array	0x0 array	getMazeData().length equals 0	Pass
MAZEENT_024	MazeTest	testSetStartUpdatesStartPosition	Maze	setStart	Verify setStart updates start position	new Position (1, 1)	getStart() equals new Position	Pass
MAZEENT_025	MazeTest	testSetStartAcceptsNull	Maze	setStart	Verify setStart accepts null	setStart(null)	getStart() returns null	Pass
MAZEENT_026	MazeTest	testSetStartAcceptsDifferentPosition	Maze	setStart	Verify setStart accepts different position	Position (2, 3)	getStart().getRow() equals 2	Pass
MAZEENT_027	MazeTest	testSetEndUpdatesEndPosition	Maze	setEnd	Verify setEnd updates end position	new Position (3, 3)	getEnd() equals new Position	Pass
MAZEENT_028	MazeTest	testSetEndAcceptsNull	Maze	setEnd	Verify setEnd accepts null	setEnd(null)	getEnd() returns null	Pass
MAZEENT_029	MazeTest	testSetEndAcceptsDifferentPosition	Maze	setEnd	Verify setEnd accepts different position	Position (4, 2)	getEnd() equals new Position	Pass
MAZEENT_030	MazeTest	testMazeDataContainsCorrectPathAtStart	Maze	getMazeData	Verify mazeData contains correct path at start	5x5 maze	mazeData[0][0] equals 0	Pass
MAZEENT_031	MazeTest	testMazeDataContainsCorrectWallAt01	Maze	getMazeData	Verify mazeData contains correct wall at (0, 1)	5x5 maze	mazeData[0][1] equals 1	Pass
MAZEENT_032	MazeTest	testMazeDataHasCorrectNumberOfRows	Maze	getMazeData	Verify mazeData has correct number of rows	5x5 maze	mazeData.length equals 5	Pass
MAZEENT_033	MazeTest	testMazeDataHasCorrectNumberOfCols	Maze	getMazeData	Verify mazeData has correct number of columns	5x5 maze	mazeData[0].length equals 5	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
MAZEENT_034	MazeTest	testStartAndEndAreDifferentObjects	Maze	getStart/getEnd	Verify start and end are different objects	Valid maze	start != end (different references)	Pass
MAZEENT_035	MazeTest	testStartPositionRowIsValid	Maze	getStart	Verify start position row is valid	start Position (0, 0)	getStart().getRow() >= 0	Pass
MAZEENT_036	MazeTest	testStartPositionColIsValid	Maze	getStart	Verify start position col is valid	start Position (0, 0)	getEnd().getCol() >= 0	Pass
MAZEENT_037	MazeTest	testEndPositionRowIsValid	Maze	getEnd	Verify end position row is valid	end Position (4, 4)	getStart().getRow() >= 0	Pass
MAZEENT_038	MazeTest	testEndPositionColIsValid	Maze	getEnd	Verify end position col is valid	end Position (4, 4)	getEnd().getCol() >= 0	Pass
MAZEENT_039	MazeTest	testMultipleSetWidthCalls	Maze	setWidth	Verify multiple setWidth calls	setWidth(7), then setWidth(12)	getWidth() returns 12	Pass
MAZEENT_040	MazeTest	testMultipleSetHeightCalls	Maze	setHeight	Verify multiple setHeight calls	setHeight(8), then setHeight(15)	getHeight() returns 15	Pass
MAZEENT_041	MazeTest	testMultipleSetStartCalls	Maze	setStart	Verify multiple setStart calls	setStart (1,1), then setStart (2,2)	getStart() equals last Position	Pass
MAZEENT_042	MazeTest	testMultipleSetEndCalls	Maze	setEnd	Verify multiple setEnd calls	setEnd (3,3), then setEnd (4,4)	getEnd() equals last Position	Pass
PLAYER_001	PlayerTest	testConstructorSetsRow	Player	Player (constructor)	Verify constructor sets row	row=2, col=3	player.row equals 2	Pass
PLAYER_002	PlayerTest	testConstructorSetsCol	Player	Player (constructor)	Verify constructor sets col	row=2, col=3	player.col equals 3	Pass
PLAYER_003	PlayerTest	testSetRow	Player	row (field)	Verify row field can be set	player.row = 5	player.row equals 5	Pass
PLAYER_004	PlayerTest	testSetCol	Player	col (field)	Verify col field can be set	player.col = 7	player.col equals 7	Pass
POSITION_001	PositionTest	testConstructorSetsRow	Position	Position (constructor)	Verify constructor sets row	row=3, col=5	getRow() returns 3	Pass
POSITION_002	PositionTest	testConstructorSetsCol	Position	Position (constructor)	Verify constructor sets col	row=3, col=5	getCol() returns 5	Pass
POSITION_003	PositionTest	testSetRow	Position	setRow	Verify setRow updates row	setRow(7)	getRow() returns 7	Pass
POSITION_004	PositionTest	testSetCol	Position	setCol	Verify setCol updates col	setCol(9)	getCol() returns 9	Pass

TEST ID	TEST CLASS	TEST METHOD	TARGET CLASS	TARGET METHOD	DESCRIPTION	TEST INPUT	EXPECTED OUTPUT	STATUS
POSITION_005	PositionTest	testEqualsReturnsTrueForSamePosition	Position	equals	Verify equals returns true for same position	Position (3,5) equals Position (3,5)	Returns true	Pass
POSITION_006	PositionTest	testEqualsReturnsFalseForDifferentRow	Position	equals	Verify equals returns false for different row	Position (3,5) equals Position (4,5)	Returns false	Pass
POSITION_007	PositionTest	testEqualsReturnsFalseForDifferentCol	Position	equals	Verify equals returns false for different col	Position (3,5) equals Position (3,6)	Returns false	Pass
POSITION_008	PositionTest	testEqualsReturnsTrueForSameObject	Position	equals	Verify equals returns true for same object	position.equals(position)	Returns true	Pass
POSITION_009	PositionTest	testEqualsReturnsFalseForNull	Position	equals	Verify equals returns false for null	position.equals(null)	Returns false	Pass