



<b>Class</b>			
<b>GameState</b>			
	Method	Input	Expected Value
	testGetPlayer	getPlayer() Player.getName()	"Player"
	testIsGameOver	-  movePlayer(Direction.WEST) movePlayer(Direction.SOUTH) movePlayer(Direction.EAST) movePlayer(Direction.EAST) movePlayer(Direction.EAST)	FALSE    TRUE
	testGetAvailableDirectionFromPlayerPosition	getAvailableDirectionFromPlayerPosition()	"WEST"
	testGetPlayerStartLocation	getPlayer() Player.getCurrentLocation()	getPlayerStartLocation()
	testGetPlayerEndLocation	getPlayer() Player.getPlayerEndLocation()	"(3, 4)"
	testMovePlayer	Player.getPlayerStartLocation()  movePlayer(Direction.WEST)  movePlayer(Direction.SOUTH)	"(2, 2)"  "(2, 1)"  "(3, 1)"
	testPlayerCurrentLocation	getPlayerCurrentLocation()	"(2, 2)"
	testInterconnectivity	getDungeon()	getNeighbours().size()
	testTreasureInRightLocation	getDungeon() Dungeon.isCave() Dungeon.getTreasure	TRUE
	testTreasurePercentage	getDungeon() Dungeon.isCave() Dungeon.getTreasure	
	testPlayerTravelStatus	movePlayer(Direction.WEST) movePlayer(Direction.SOUTH)	"(2, 2), (2, 1), (3, 1)"
	testPlayerPickingTreasure	movePlayer(Direction.WEST) movePlayer(Direction.SOUTH) movePlayer(Direction.EAST) movePlayer(Direction.EAST) movePlayer(Direction.EAST)	"(2, 2) (2, 1) (3, 1), (3, 2) (3, 3) (3, 4)"
	testTreasureNullAfterPicking	getDungeon() locationInfo() movePlayer(Direction.WEST) movePlayer(Direction.SOUTH) movePlayer(Direction.WEST) movePlayer(Direction.NORTH) Player.travelStatus()	"(2, 2) (2, 1, (3, 1) (3, 0) (2, 0)"
	testIllegalMove	movePlayer(Direction.NORTH)	IllegalArgumentException
	testDungeonCreation	getDungeon() getNeighbours().size()	
	testPathLength	getPlayerStartLocation() getPlayerEndLocation()	distance = getPlayerStartLocation() - getPlayerEndLocation() distance >= 5

<b>GameTestWrapping</b>			
	testGameStateWrapping	GameState(-10, -10, -1, "wrapping", 20, null)	IllegalArgumentException
	testGetPlayer	GameState.getPlayer().getName()	"Player"
		GameState.isGameOver()	FALSE
		GameState.movePlayer(Direction.NORTH); GameState.movePlayer(Direction.NORTH); GameState.movePlayer(Direction.NORTH); GameState.movePlayer(Direction.NORTH); GameState.movePlayer(Direction.EAST); GameState.movePlayer(Direction.EAST); GameState.movePlayer(Direction.EAST);	TRUE
	testIsGameOver		
		GameState.getAvailableDirectionsFromPlayerPosition().toString()	"[SOUTH, EAST, WEST]"
	testGetAvailableDirectionsFromPlayerPosition	GameState.getAvailableDirectionsFromPlayerPosition().size()	3
	testGetPlayerStartLocation	GameState.getPlayerStartLocation().toString()	"(2, 4)"
	testGetPlayerEndLocation	GameState.getPlayerEndLocation().toString()	"(4, 7)"
		GameState.getDungeon() GameState.getPlayer().getCurrentLocation().toString()	"(2, 4)"
		GameState.movePlayer(Direction.EAST) GameState.getPlayer().getCurrentLocation().toString()	"(2, 5)"
		GameState.movePlayer(Direction.NORTH) GameState.movePlayer(Direction.NORTH) GameState.movePlayer(Direction.EAST) GameState.movePlayer(Direction.NORTH) GameState.getPlayer().getCurrentLocation().toString()	"(9, 6)"
	testMovePlayer		
	TestTreasureInRightLocation	GameState.getDungeon() isCave() getTreasure()	TRUE
		GameState.getDungeon() isCave() getTreasure() toStringPlayerTravelStatus()	"Player has traveled to the following locations: [(2, 4) ].\n" + "Treasures: {}"
	testToStringTravelStatus		
		GameState.movePlayer(Direction.EAST); GameState.movePlayer(Direction.EAST); GameState.movePlayer(Direction.SOUTH); GameState.movePlayer(Direction.SOUTH); GameState.movePlayer(Direction.WEST); GameState.movePlayer(Direction.WEST); GameState.movePlayer(Direction.SOUTH); GameState.movePlayer(Direction.EAST); GameState.movePlayer(Direction.EAST); toStringPlayerTravelStatus()	"Player has traveled to the following locations: [(2, 4) (2, 5) (2, 6) " + "(3, 6) (4, 6) (4, 5) (4, 4) (5, 4) (5, 5) (5, 6) ].\n" + "Treasures: {DIAMONDS=14, RUBIES=9, SAPPHIRES=15}"
	testPlayerPickingTreasure		
		GameState.movePlayer(Direction.EAST) GameState.movePlayer(Direction.EAST) GameState.movePlayer(Direction.SOUTH) printPlayerTravelStatus()  movePlayer(Direction.WEST) movePlayer(Direction.EAST) printPlayerTravelStatus()	"Player has traveled to the following locations: [(2, 4) (2, 5) " + "(2, 6) (3, 6) ].\n" + Treasures: {DIAMONDS=7, RUBIES=2, SAPPHIRES=9}"  Player has traveled to the following locations: [(2, 4) (2, 5) " + "(2, 6) (3, 6) (3, 5) (3, 6) ].\n" + "Treasures: {DIAMONDS=7, RUBIES=2, SAPPHIRES=9}"
	testTreasureNullAfterPick		
	testIllegalMove	movePlayer(Direction.NORTH)	IllegalArgumentException
		getPlayerStartLocation() getPlayerEndLocation()	distance = getPlayerStartLocation() - getPlayerEndLocation() distance >= 5
	testPathLength		

		GameState(6, 10, 1000, "wrapping", 20, random) getAvailableDirectionsFromPlayerPosition().toString()  movePlayer(Direction.NORTH) getPlayerCurrentLocation().toString()  getAvailableDirectionsFromPlayerPosition().toString() movePlayer(Direction.SOUTH) getPlayerCurrentLocation().toString()  getAvailableDirectionsFromPlayerPosition().toString() movePlayer(Direction.EAST) getPlayerCurrentLocation().toString()  getAvailableDirectionsFromPlayerPosition().toString() movePlayer(Direction.WEST) getPlayerCurrentLocation().toString()	"[SOUTH, WEST, EAST, NORTH]"  "(3, 8)"  "[EAST, WEST, SOUTH, NORTH]"  "(4, 8)"  "[WEST, EAST, SOUTH, NORTH]"  "(4, 9)"  "[WEST, EAST, SOUTH, NORTH]"  "(4, 8)"
	testPlayerMovementsAllDirections		
<b>Location</b>			
	testIncorrectLocation	new Location()	IllegalArgumentException
	testGetName	getName()	"(0, 0)"
	testGetRowCoordinate	getRowCoordinate()	0
	testGetColCoordinate	getColCoordinate()	0
	testJoinLocationToNorthDirection	Location.joinLocationToNorthDirection(Location()) Location.getNeighbours().get(Direction.NORTH)	Location(0, 2, rand)
	testJoinLocationToSouthDirection	Location.joinLocationToSouthDirection(Location()) Location.getNeighbours().get(Direction.SOUTH)	Location(1, 2, rand)
	testJoinLocationToEastDirection	Location.joinLocationToEastDirection(Location()) Location.getNeighbours().get(Direction.EAST)	Location(1, 2, rand)
	testJoinLocationToWestDirection	Location.joinLocationToWestDirection(Location()) Location.getNeighbours().get(Direction.WEST)	Location(1, 2, rand)
	testIsCave	Location.joinLocationToNorthDirection(Location()) Location.joinLocationToSouthDirection(Location()) Location.joinLocationToEastDirection(Location()) Location.joinLocationToWestDirection(Location()) Location.isCave()	TRUE
	testSetTreasure	Location.joinLocationToNorthDirection(Location()) Location.joinLocationToSouthDirection(Location()) Location.joinLocationToEastDirection(Location()) Location.joinLocationToWestDirection(Location()) Location.setTreasure()	null
	testSetTreasureInTunnel	Location.joinLocationToNorthDirection(Location()) Location.joinLocationToSouthDirection(Location()) Location.setTreasure()	null
	testGetTreasure	Location.joinLocationToNorthDirection(Location()) Location.joinLocationToSouthDirection(Location()) Location.joinLocationToEastDirection(Location()) Location.joinLocationToWestDirection(Location()) Location.setTreasure() Location.getTreasure()	"{SAPPHIRES=4, RUBIES=8, DIAMONDS=9}"

	testGetNeighbours	Location.joinLocationToNorthDirection(Location()) Location.joinLocationToSouthDirection(Location()) Location.joinLocationToEastDirection(Location()) Location.joinLocationToWestDirection(Location()) Location.getNeighbours().size()  Location.getNeighbours().get(Direction.NORTH)  Location.getNeighbours().get(Direction.SOUTH)	4  Location(0, 2, rand)  Location(2, 2, rand)
	testLocationInfo	Location.joinLocationToNorthDirection(Location()) Location.setTreasure() Location.LocationInfo()	"(0, 0), treasure: {SAPPHIRES=4, RUBIES=8, DIAMONDS=9}, " + "neighbours: {NORTH=(0, 2)}"
<b>Character</b>			
	testPlayerNull	Player(null)	IllegalArgumentException
	testGetName	Player.getName()	"Player"
	testaddTreasure	Treasure.getTreasure()	
	testGetTreasure	Player.getTreasures()	
	testAddTwoTreasures	Player.addTreasure(Treasure) Player.addTreasure(Treasure)	
	testGetCurrentLocation	Player.setCurrentLocation(Location(0, 0, random))	Location(0, 0, random))
	testGetCurrentLocationEmpty	Player.getCurrentLocation()	null
	testSetCurrentLocation	Player.getCurrentLocation()	Player.setCurrentLocation(Location(0, 0, random))
	testMovePlayer	Player.getCurrentLocation() Player.getCurrentLocation()	Player.setCurrentLocation(Location(0, 0, random)) Player.setCurrentLocation(Location(0, 1, random))
	testTravelStatus	Player.setCurrentLocation(Location(0, 0, random)) Player.setCurrentLocation(Location(1, 0, random)) Player.setCurrentLocation(Location(2, 0, random)) Player.toStringTravelStatus()	"Player has traveled to the following locations: [(0, 0) (1, 0) (2, 0) ].\n" + "Treasures: {}"
	testTravelStatus	getDungeon() locationInfo() movePlayer(Direction.WEST) movePlayer(Direction.SOUTH) movePlayer(Direction.WEST) movePlayer(Direction.NORTH) Player.toStringTravelStatus()	"Player has traveled to the following locations: [].\n" + "Treasures: {}"
<b>Treasure</b>			
	testIncorrectTreasure	Treasure(null)	IllegalArgumentException
	testGetTreasure	Treasure.getTreasure()	expected.put(TreasureType.RUBIES, 4); expected.put(TreasureType.SAPPHIRES, 4); expected.put(TreasureType.DIAMONDS, 1);