

## Testing Design Project 4: Dungeons

### KruskalDungeonTest: Testing of functionality implemented as per Project 3.

Test Method Name	Description
<code>createPaths()</code>	<i>To test the creation of potential paths.</i>
<code>moveFourDirections()</code>	<i>To test the movement of player in all 4 directions.</i>
<code>movePlayerSouth()</code>	<i>To test the movement of player to south when possible.</i>
<code>movePlayerNorth()</code>	<i>To test the movement of player to north when possible.</i>
<code>movePlayerEast()</code>	<i>To test the movement of player to East when possible.</i>
<code>movePlayerWest()</code>	<i>To test the movement of player to West when possible.</i>
<code>testTreasure()</code>	<i>To test that treasure is available in the specified percentage of caves.</i>
<code>testGetPotentialPath()</code>	<i>To test that the total number of potential paths is correct and is same as expected.</i>
<code>describePlayer()</code>	<i>To test that description of a player whose name,treasures and location is displayed.</i>
<code>pickTreasure()</code>	<i>To test if a player picks treasure when available in the cave.</i>
<code>startEndDistance()</code>	<i>To test that the start and end are at least at a distance of 5 units.</i>
<code>getPlayerLocation()</code>	<i>To test that player location is accurate.</i>
<code>testAllNodesReachable()</code>	<i>To test if all nodes are connected. There exists at least one path from one node to another.</i>
<code>playerStartsFromDungeonStart()</code>	<i>o test that player starts at the Dungeon start location.</i>

PlayerControllerTest: to test functionality and methods of Controller which were implemented as part of project 4

Test Method Name	Description
testIllegalModel()	To test Controller with an invalid model throws <i>IllegalArgumentException</i> .
testChoosingMoveValid()	To test controller when choosing move and a valid move
testChoosingMoveInvalid()	To test controller when choosing move and an invalid move.
testMoveToEnd()	To test controller when player moves reaches the end and eaten by monster.
winGame()	To test controller when player moves,shoots and kill Otyugh at end to win the game.
injuresOtyugh()	To test controller when player moves,shoots and injures Otyugh.
shootMissOtyugh()	o test controller when player moves,shoots and Misses Otyugh.
playerDescription()	To test controller when checking player description at the moment.
testQuit()	o test controller when quitting the game.
testInvalidMainOption()	To test controller when choosing an invalid option.
testPickingArrowValid()	To test controller when picking arrows when available.
testPickingInvalid()	To test Controller when picking arrow/Treasure when not available.
shootBlocked()	Test controller when shooting in blocked direction.
invalidChoice()	Test controller when user enters an invalid choice.
treasurePickAvailable()	To test controller when treasure is available and up for picking.

TestProject4: to test the methods and functionality which were added part of project 4.

Test Method Name	Description
testNumberOfOtyughs ()	To test the number of Otyughs in the dungeon.
testOtyughAtEnd()	To test that End always has an Otyugh.
invalidNumberOfOtyughs ()	To test Illegal number of otyughs.
invalidNumberOfOtyughs2 ()	To test illegal number of Otyughs i.e. more than the nodes.
smellOtyughLevel0 ()	To test the pungency when otyugh is not in the vicinity(2 units) of the player.
smellOtyughLevel1 ()	To test the pungency when one otyugh is in the vicinity of the player at a distance of 2 units.
smellOtyughLevel2 ()	To test the pungency when one otyugh is in the vicinity of the player at a distance of 1 units.
smellOtyughLevelMultipleOtyughs ()	To test for smell when multiple otyughs present at 2 units. In this case Otyughs present at Node 34 and 36. We try to find the pungency of smell from location 27 which is 2 units from both.
testPlayerStartsWith3Arrows ()	To test that player starts with 3 arrows.
testPlayerLosesArrowOnShooting()	To test that player loses an arrow on shooting.
testPlayerPicksArrow ()	To test tha player picks Arrow.
testPlayerCannotPickArrows ()	Trying to pick arrows when not available.
testPlayerInjuringOtyugh ()	Test to check if one arrow injures the Otyugh. * shoot() function returns 1 if the arrow injured otyugh. 2 if the arrow kills the otyugh.

<code>testPlayerKillingOtyugh()</code>	<p><i>Test to check if one arrow Kills the Otyugh.</i></p> <p><i>* shoot() function returns 1 if the arrow injured otyugh. 2 if the arrow kills the otyugh.</i></p>
<code>testArrowBlocked()</code>	<p><i>To test that arrow gets blocked in cave. Returns 0 if No harm done or arrow gets blocked.</i></p>
<code>testArrowTunnel()</code>	<p><i>To test that arrow doesn't get blocked in a tunnel.</i></p> <pre>*      █ \$ \$ \$ \$ \$ *  ↯   ↯ ↯ ↯   ↯ *      M\$ *</pre> <p><i>M is the monster and when we shoot in the direction of South the arrow travels in</i></p> <p><i>* the same direction as of the Tunnel and shoots the otyugh.</i></p>