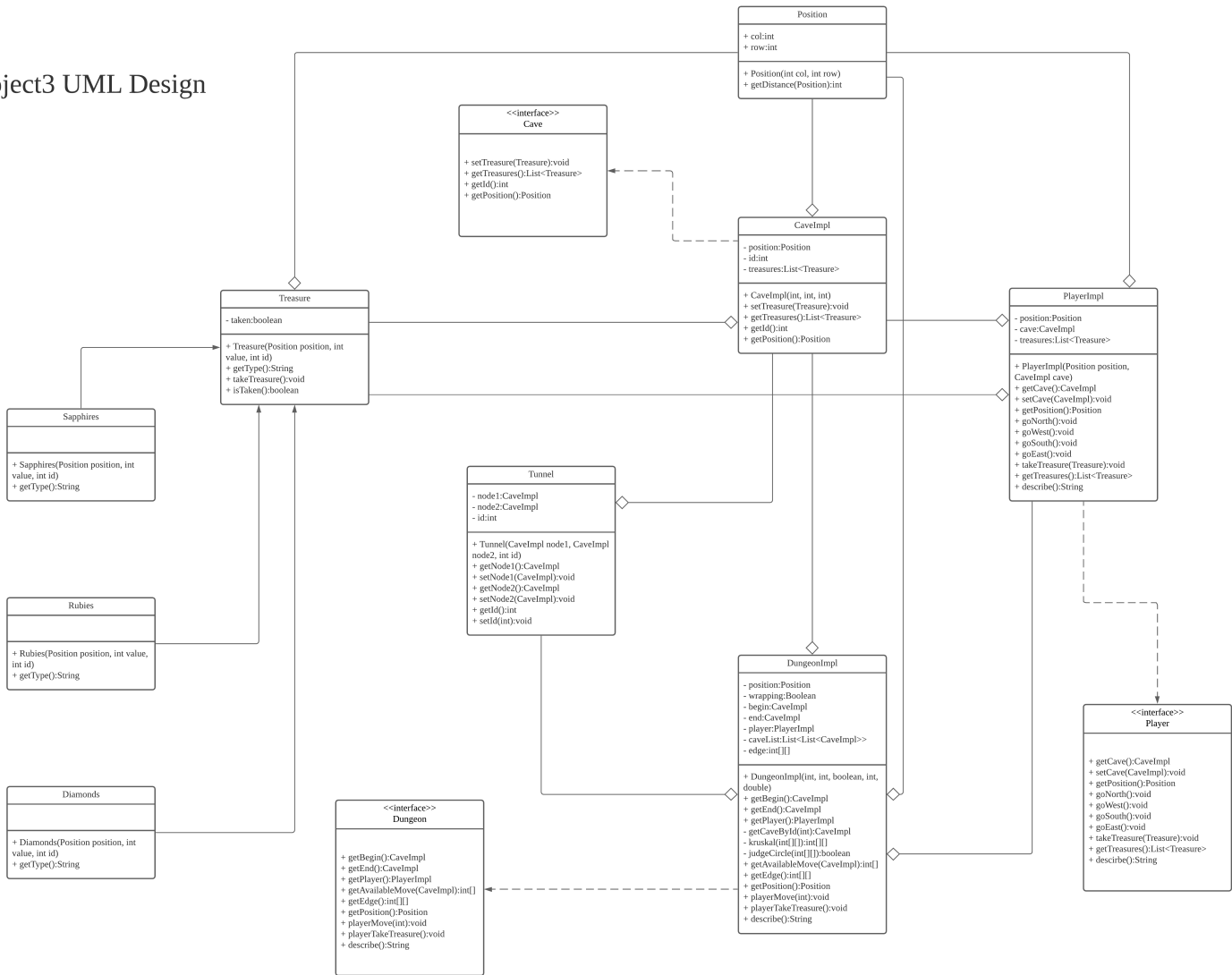


# Project3 UML Design



## Project 3 Test Plan

### Test Form

CaveTest	PlayerTest	TreasureTest	TunnelTest	DungeonTest	PositionTest
testSetGetTreasure()	testSetGetCave()	testGetType()	testSetGetNode()	testGetPosition()	testGetDistance()
testGetId()	testTakeTreasure()	testTakeTreasure()	testSetGetId()	testInvalidIntercon()	testInvalidRow()
testGetPosition()	testGetPosition()	testTakeSameTreasure()		testInvalidTreasure()	testInvalidCol()
testInvalidId()	testMove()				

### CaveTest

public void testInvalidId()

Test whether the id is supported.

public void testSetTreasure()

Test if you can set Treasure to the Cave correctly.

public void testGetTreasure()

After setting Treasure to the Cave, test whether it can be found in the Cave.

public void testGetId()

Test whether the id of the cave is correct.

public void testGetPosition()

Test whether the coordinates of the Cave are correct. And after creating a Cave, test whether its coordinates match expectations.

### PlayerTest

public void testSetCave()

Test whether the Player is properly bound to the Cave.

public void testGetCave()

After setting a Cave to Player, verify that the coordinates of Player and Cave are the same.

public void testTakeTreasure()

Test whether the Player can pick Treasure. And after passing in a Treasure, verify that the Player's pick list contains the Treasure.

public void testGetPosition()

Test whether the player is in the correct position.

public void testGoNorth()

Test whether the player can move to north normally. After moving, verify that the Player's coordinates change as expected.

public void testGoSouth()

Test whether the player can move to south normally. After moving, verify that the Player's coordinates change as expected.

public void testGoWest()

Test whether the player can move to west normally. After moving, verify that the Player's coordinates change as expected.

public void testGoEast()

Test whether the player can move to east normally. After moving, verify that the Player's coordinates change as expected.

## **TreasureTest**

public void testGetType()

Test whether the three types of treasures are correct.

public void testTakeTreasure()

Test whether the treasure is successfully taken by the player.

public void testTakeSameTreasure()

Test whether the player can pick up the same type of treasure multiple times.

## **TunnelTest**

public void testSetNode()

Test whether the node can be successfully set up in the tunnel.

public void testGetNode()

After setting the node, test whether the node can be found in the tunnel successfully.

public void testSetId()

Test whether the id can be successfully set up in the tunnel.

public void testGetId()

After setting the id, test whether the id can be found in the tunnel successfully.

## **DungeonTest**

public void testGetPosition()

Test whether the position (col and row) of the dungeon meets expectations.

public void testInvalidInterconnectivity()

Test whether the interconnectivity is supported.

public void testInvalidTreasureRandom ()

Test whether the probability of treasure is supported.

## **PositionTest**

public void testInvalidRow()

Test whether the row of position is supported.

public void testInvalidCol()

Test whether the column of position is supported.

public void testGetDistance()

Test whether Position can measure the distance correctly. And after passing in a Position, verify that the distance between the two positions as expected.