**Player Config Tests**

1. Entering Invalid Player Name Gives Alert

* Entering a null string in the text field for “Name” gives an error
* Entering a string with only multiple empty spaces in the text field for “Name” gives an error
* Part of M2 implementation requirements for the player configuration on the “Welcome” screen

1. Player Starting Money is Different Based on Difficulty Levels

* Difficulty Level of 1 allows the player to start with $2000
* Difficulty Level of 2 allows the player to start with $1500
* Difficulty Level of 3 allows the player to start with $1000
* Checks to see that the starting money in the Player class is different when inputting different difficulty levels
* Part of M2 implementation requirements for the player configuration on the “Welcome” screen

1. Player Starting Health is Different Based on Difficulty Levels

* Difficulty Level of 1 allows the player to start with 500 health
* Difficulty Level of 2 allows the player to start with 400 health
* Difficulty Level of 3 allows the player to start with 300 health
* Checks to see that the starting health in the Base class is different when inputting different difficult levels
* Part of M2 implementation requirements for the player configuration on the “Welcome” screen

1. Entering null difficulty gives alert
   1. Causes method to return “Must choose a difficulty”
   2. Part of M2 implementation requirements for the player configuration on the “Welcome” screen

**Shop Tests**

1. Player purchase tower successfully
   1. checkPurchaseTower1()
      1. Checks to see if the Tower 1 bought at difficulty 1 correctly decreases the player’s money by the tower’s price.
   2. checkPurchaseTower2()
      1. Checks to see if the Tower 2 bought at difficulty 1 correctly decreases the player’s money by the tower’s price.
   3. checkPurchaseTower3()
      1. Checks to see if the Tower 3 bought at difficulty 1 correctly decreases the player’s money by the tower’s price.
2. Money remaining is different for same level different tower
   1. checkSameLevelDifTowerMoneyChange()
      1. Creates 3 towers of different types, and purchases them. Checks if each tower costs a different price by comparing the amount of money the player has left after purchasing.
3. Money remaining is different for different level same tower
   1. checkDifLevelSameTowerMoneyChange()
      1. Creates a Player that buys 3 towers, each at a different (upgraded) level. Checks whether the cost of each tower is different at each level.

## **Tower Place Tests**

1. Tower can not be placed onto the path
   1. The new tower cannot be placed onto the path
   2. Checks functionality:
      1. towerIsNotOnPathTest()
         1. Created an array of path rectangles and configured coordinates so that the new tower will not be in the same area as the path and tested if the method of checking isTowerOnPath results in False.
      2. towerIsOnPathTest()
         1. Created an array of path rectangles and configured coordinates so that the new tower will be on the same coordinates as the path and tested if the method of checking isTowerOnPath results in True.
2. A tower can not be placed onto another tower
   1. If current towers null, new tower should always be placed
   2. The new tower cannot be touching with current towers
   3. Checks functionality:
      1. towerIsNotOnTowerTest()
         1. Created an array of already placed towers and configured coordinates so that towers would not be near each other and tested if the method of checking isTowerOnTower results in False.
      2. towerIsOnTowerTest()
         1. Created an array of already placed towers and configured coordinates so that the new tower will be in the same place as an already placed tower and tested if the method of checking isTowerOnTower results in True.