Unit Tests:

Feature One:

* Description: The first feature to be unit tested is the detectWin() method. This is an essential feature of the game because this method controls whether Player is still in a game, wins a game, loses a game based on different conditions.
* Test case/class: To test this feature, a class called TestDetectWin is created with three methods that each contains different test cases. The first method is testStillInGame(), which has two test cases that test two scenarios where Player is still in a game. The second method is testLose(), which has three test cases that test three scenarios where Player loses a game. The third method is tests whether Player wins a game. It only has one test case, which is when Player reaches the exit cell and has collected all the regular rewards.

Feature Two:

* Description

Integration Tests:

* Interaction 1 between components, components involved, and test case/class
* Interaction 2 between components, components involved, and test case/class

Test Quality

Findings

1. The four private int methods for wall detection originally resided in the Panel class and returned an integer variable that only the Panel has. However, as we created unit tests for the wall detection methods, we discovered that private methods generally are not tested, and if they are so complex that they need testing, the methods should deserve their own class. Therefore, we created a separate class called WallDetection that hold these four methods and changed them to Boolean. By doing this, the code is more reusable because not only the Panel class can use the wall detection methods, other classes can also use them by creating a WallDetection object and calling the methods. It is also easier to do testing on its methods.