Group 2

Kaibo Ma (32400129)
David Henderson (19475128)
Shivanshu Goyal (33245127)

Exercise 1.2

Test Charter 1: Highscore of Current Session
Actor: A normal game user
Purpose: To investigate if there is any feature that will display the highest score achieved in the current JPacman instance. This would requesting the game have some sort of persistence storage only a memory of what the highest score was for the current instance of the JPacman opened.
Setup: A computer that has a instance of the JPacman game on it.
Priority: Low, because it is a non essential feature.
Exploratory Tour: 1) Open an instance of the JPacman game.
2) Start a round of the game.
3) Collect any number of points.
4) Either get eaten by ghost or collect all the tokens.
5) Start a new JPacman game in current instance.
6) Collect more points that you did in the first game.
7) Either get eaten or ghost or collect all the tokens.
8) There is **no** indication of getting the highscore.

Test Charter 2: Pacman Power Pellets
Actor: A normal game user
Purpose: To investigate if the game includes the classic feature of most Pacman games where the user can eat a certain pellet and then be invincible to any of the ghosts.
Setup: A computer that has a instance of the JPacman game on it.
Priority: Low, because it is a non essential feature.
Exploratory Tour: 1) Open an instance of the JPacman game.
2) Start a round of the game.
3) Eat all the pellets (winning the game)
4) There will be no time that the user will be invisible from the ghosts.

Test Charter 3: Undoable Button for current Progress in the game
Actor: A normal game user
Purpose: To investigate if there is a functionality that allows a user to press a button to undo all previous saved states. This will require the game to have some sort of stack storage system for all previous moves to be pushed and popped out.
Setup: A computer that has a instance of the JPacman game on it.

Priority: Low, it is not a required feature but will make the game more user friendly in winning.
Exploratory Tour: 1) Open JPacman game.
2) Press the Start button to begin playing.
3) Start collecting points in the map.
4) Then accidentally get eaten by a ghost.
5) There is no way to undo the steps so you can redo your moves and not lose.

Exercise 1.3

- The JPacman have 10 classes
- There are 39 JUnit tests
- When a test fails the green bar will become red and also the test will have a red X beside it in the JUnit window.

Exercise 1.4

```
88    /**
89     * Test that the ghost causes the player's death if the ghost bumps into the player.
90     */
91    @Test
92    public void test_S3_4_GhostKillsPlayer() {
93        // given
94        getEngine().start();
95        // when
96        getUI().getGame().moveGhost(theGhost(), Direction.LEFT);
97        // then
98        assertFalse(getPlayer().isAlive());
99    }
```

GhostKillsPlayer()
We picked the test case where it asserts if a Ghost is able to kill a player. The test validates that the player will die whenever a ghost bumps into them. This requires the test to begin the game's engine and set the state to start and have the player's location directly left of the ghost. Then it will get the ghost to move from the starting cell to the cell on the left where the player is residing. At the end it will validate if the player has died by checking if they are not alive.

We think this test case is not complete. There should be more test cases to supplement this test case. For example, This test case only considers if the Ghost moves to the left. There also should be test cases where the ghost moves up, down, and right. Also it should check what if the player bumps into a ghost too. Lastly, it should also check if the player or ghost collide with each other if they move through the tunnels on the left and right side of the pacman map.