**Test case explanation:**

**gameOver:**

* Shows that if the tile bag is empty and placing a tile causes a player’s hand to be empty, then the game will end and display the scores and the winner. If the scores are equal, then the game will call a draw.

**placeTile2:**

* Shows that placing a tile next to a segment in which the adjacent tile is valid, but already contains the specified tile, will fail (Y2 at G1). Also shows that placing a tile next to a segment in which the adjacent tile is valid, but the segment’s similarity type is different to the similarity between the specified tile and the adjacent tile, will fail (Y2 at H2).
* Shows that placing a tile between two segments that share a similar tile is invalid (R4 at B3) but placing a tile at the corner of two segments that share a similar tile is fine (R4 at B6). Also shows that placing a tile between two segments that don’t share a similar tile is also fine (Y5 at G4)
* Shows that placing a tile between two segments of different similarity types is invalid (R4 at H6), but placing a tile at the corner of two segments of a different similarity type in fine (R4 at H6)

**loadGame:**

* Player details included in the saved game file must follow the required format of a saved game. The name of both players must only be consisted of letters (no numbers or symbols). The score of both players can only contain a positive number or zero. Each player’s hand must not contain more than 6 tiles
* All the tiles found in the hands of the players and in the tile bag container must only contain valid tiles and are properly separated using the correct format which is by separating each tile using a comma and ending each line not with a space or comma but a valid tile.
* The board must follow the required format including the column and row co-ordinates. No tiles should be found out of bounds of the board and no invalid tiles should be accepted.
* Current player must match the name of one of the players.
* The tile bag contents should not be empty.

**Test cases yet to be created:**

**runGame:**

* Game end condition
* Stalemate condition

**placeTile:**

* Specifying an invalid tile (improper format, does not exist in the player’s hand) or an invalid position fail.

1. Invalid position fail -> place R5 at F26

2. Doesn’t exist in players hand -> place R6 at F10

3. Improper format of input -> plac R5 at G6

* Placing a tile next to one that shares no similarity fails -> place R5 at M5
* Placing a tile next to an (adjacent) duplicate fails -> place R5 at F10
* Placing a tile next to one that only shares one type of similarity succeeds (show both colour and shape successes)

1. Similar colour -> place R1 at G6 which is placed in a segment containing Red tiles

2. Similar shape -> place O1 at C11 which is placed next to R1

* Placing a tile in a segment adds one to the score for every tile in the segment (including the tile itself) -> place O6 at M10 depicts this behaviour
* Placing a tile as part of two segments adds to the score for every tile in each segment, plus an extra point for the tile itself (2 in total) -> place R6 at F10
* Creating a qwirkle results in 6 bonus points and "QWIRKLE!!!" being printed to the screen -> place )3 at M11

**replaceTile:**

* Attempting to replace a tile that doesn’t exist in the player’s hand will fail
* Attempting to replace a tile that does exist in the player’s hand will succeed
* Attempting to replace a tile while the tile bag is empty will fail

**Everything else:**

* getInput, boardToString and displayGameState are all tested implicitly in other test cases, we will display the fact that newGame and saveGame work on the day.