# Testing

## Testing Method

Testing was done very simply: a Test class was created that initialised the objects it needed to perform each test, then ran the tests. Each test was run individually rather than running all tests in a batch, to make sure the program did not crash or produce any unusual errors.

Occasionally, random numbers were replaced with fixed values in some classes to provide deterministic tests.

All tests done were black box tests: doing white box testing would have taken too much time to justify the effort involved.

## 1. Unit Testing

## Card, CardManager and Related Classes

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic under test** | **Test** | **Expected result** | **Actual result** |
| Incorrectly formatted card input file | Delete flag (first character) in second line of chance card file | Error message | No message, segmentation fault when landing on chance (FAIL) |
| Player lands on community chest | Manually move a player to community chest tile. Try a second time to see if it is random. | Card is picked at random, action performed. Second time card is different. | Got two different community chest cards. (PASS) |
| Player lands on chance | Manually move a player to chance tile. Try a second time to see if it is random. | Card is picked at random, action performed. Second time card is different. |  |
| Player receives money from card | Manually assign “Won a crossword competition” action. | Player gets £100 |  |
| Player loses money from card | Manually assign “Drunk in charge” card | Player loses £20 |  |
| Player goes to jail | Manually assign “Go to jail” card | Player goes to jail |  |
| Player gets moved to a certain position | Manually assign “Advance to go” card | Player goes to “go” square |  |
| Test every card | Manually do every card's action on player | No bugs or errors |  |

## Player and PlayerManager Class

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic under test** | **Test** | **Expected result** | **Actual result** |
| Player lands on 'go to jail' tile | Manually move player to “go to jail” tile | Player goes to jail |  |
| Player buys a house | Give player all brown properties, land on Old Kent Road, select option to buy house | Player gets house on Old Kent Road, pays £60 |  |
| Player tries to buy a house, but doesn't have enough money | Set player's balance to 10, try to buy Pall Mall | Game prints “you can't afford that property” message |  |
| Player pays rent | Give Pall Mall to player A, send player B to Pall Mall | Player B loses £10, player A gets £10 |  |
| Player receives rent | Give Pall Mall to player A, move player B to Pall Mall | Player A gets £10, Player B loses £10 |  |
| Player rolls a double | Manually set dice to 1, 1 | Player gets another go |  |
| Test player landing on every tile | Manually move player to every tile | No bugs or errors |  |

## Property and Related Classes

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic under test** | **Test** | **Expected result** | **Actual result** |
| Mortgage property | Give player Pall Mall, player selects 'mortgage' option | Pall Mall is set to “mortgaged” |  |
| Pay rent on a station | Sell station to player A, manually send player B to that station | Player B pays £25 rent |  |
| Pay rent on a utility | Sell utility to player A, manually send player B to that utility | Player B pays four times second dice roll |  |
| Pay rent on 1 house | Give player A a house on Old Kent Road, send player B to Old Kent Road | Player B pays £10 to player A |  |
| Pay rent on 2 houses | Give player A two houses on Old Kent Road, send player B to Old Kent Road | Player B pays £30 to player A |  |
| Pay rent on 3 houses | Give player A three houses on Old Kent Road, send player B to Old Kent Road | Player B pays £90 to player A |  |
| Pay rent on 4 houses | Give player A four houses on Old Kent Road, send player B to Old Kent Road | Player B pays £160 to player A |  |
| Pay rent on a hotel | Give player A a hotel on Old Kent Road, send player B to Old Kent Road | Player B pays £250 to player A |  |

## 2. Integration Testing

## Game and Board Classes

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic under test** | **Test** | **Expected result** | **Actual result** |
| Incorrectly formatted board input file | Remove colour field from Pall Mall | Error message |  |
| Insufficent tiles in board input file | Remove Pall Mall | Error message |  |

## 3. System Testing

## Functional Testing

|  |  |  |  |
| --- | --- | --- | --- |
| **Characteristic under test** | **Test** | **Expected result** | **Actual result** |
| Finish game with 2 players | Keep playing the game until it finishes | Game finishes without errors or bugs |  |
| Finish game with 3 players | Keep playing the game until it finishes | Game finishes without errors or bugs |  |
| Finish game with 4 players | Keep playing the game until it finishes | Game finishes without errors or bugs |  |
| Finish game with 5 players | Keep playing the game until it finishes | Game finishes without errors or bugs |  |
| Finish game with 6 players | Keep playing the game until it finishes | Game finishes without errors or bugs |  |