#### 

Software Test Report

Red Opal Innovations

Version 1

# Overview

This document is the software test report of the testing phase of the Simple Game software development project. It contains the results of tests, which were executed during the testing phase.

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Test case ID | Test case name | Summary | Expected results |
| I01 | Magic Number | Running the program 5 times and interrogating (printing) the random number between 1, 100 | The number should fluctuate each time as a number between 1 and 100 (inclusive) |
| I02 | Roll | Running program 5 times and interrogating (printing) random number of guesses between 1, 6 | The number of guesses should change within the bounds of 1, 6 (inclusive) each time the program is run |
| TC01 | Loss | The user should play the game to lose until the number of guesses runs out | The number of guesses should decrease after each number is entered, until the number of guesses equals zero. Then the users name should be displayed next to their outcome (loss) and number of guesses |
| TC02 | Win | In this run, the random number should be printed out so it is visible, and this number should be input as the guess | The program should end the game and display the users name next to the outcome (win) and the number of guesses they originally had |
| TC03 | Non-number | Regardless of the number of guesses, the input will be a letter or special character | The program should give a ValueError exception as the value given should be of type int, but is not |
| TC04 | Outside of range | The input will always be a number outside of the range 1, 100 (no previous games played) | The number of guesses should decrease until the number of guesses runs out, then the outcome should be a loss, and this will be displayed |
| TC05 | Statistics test | Before playing 5 games should be entered into the statistics file. The program will then be run (outcome irrelevant) | The game should end after the number of guesses displayed equals zero, and after should display all previous games as well as the most recent one |

# Test process

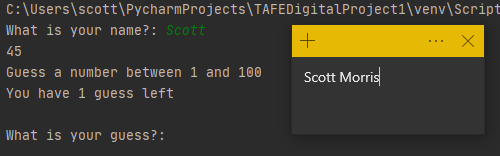
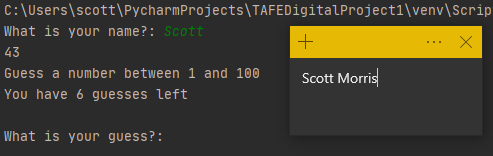
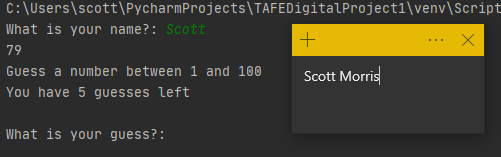
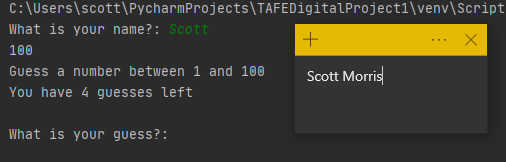
The interrogation tests were run first to ensure that the program was delivering correct starting values. If this part of the program were not running as per the specification, the other tests would also be affected by this.

After variable interrogation, the test cases were run to ensure that the game produced the expected outputs for various cases.

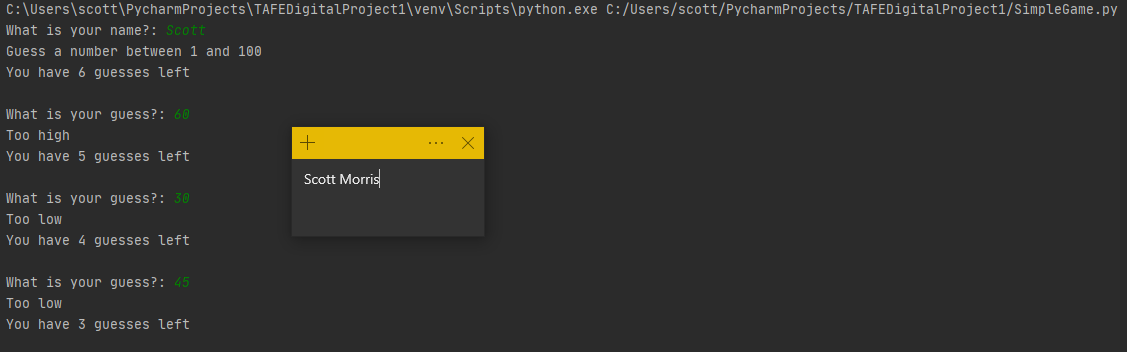
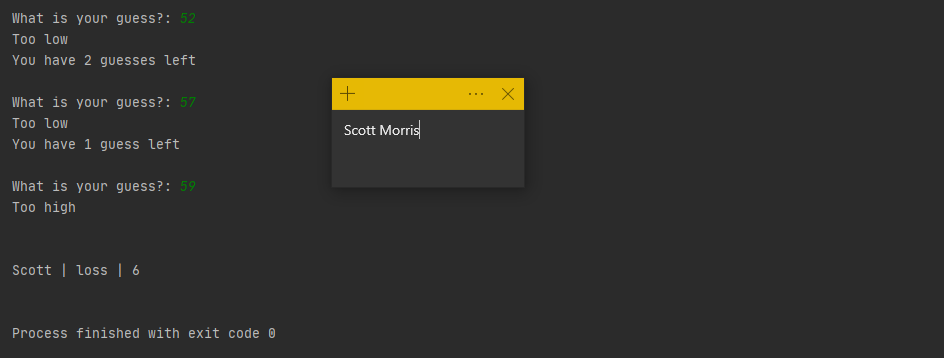
Several testers part of the development team were used to run the test cases and interrogations concurrently while developers from the development team observed and analysed the responses of the program and the feedback from the testers.

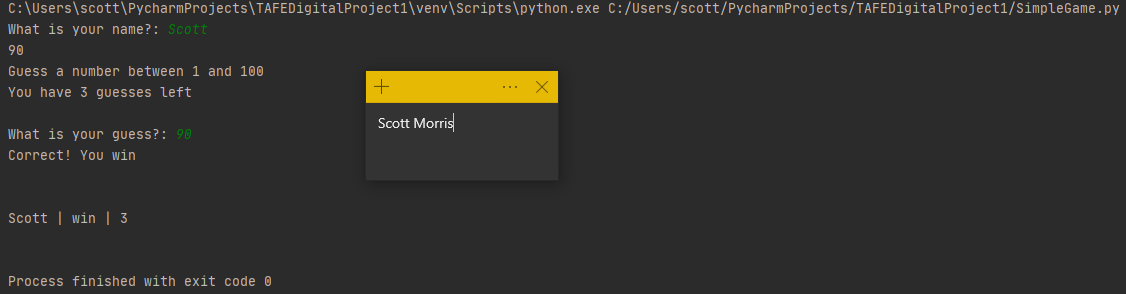
Python 3.7.6 was used as the software to create the program and the platform that it run off was PyCharm.

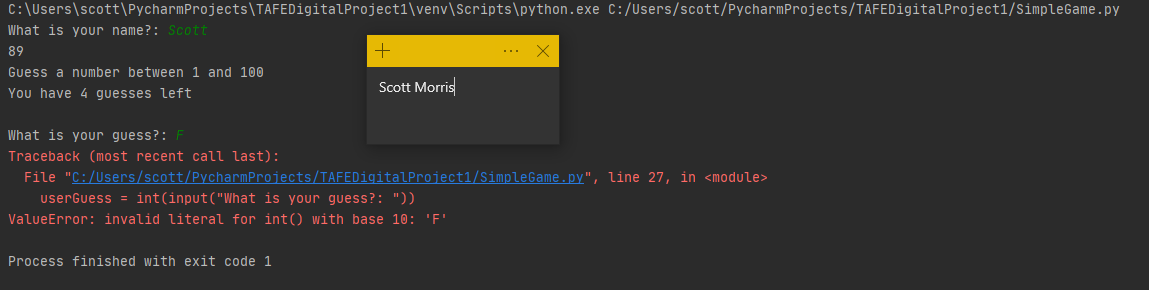
# Test results

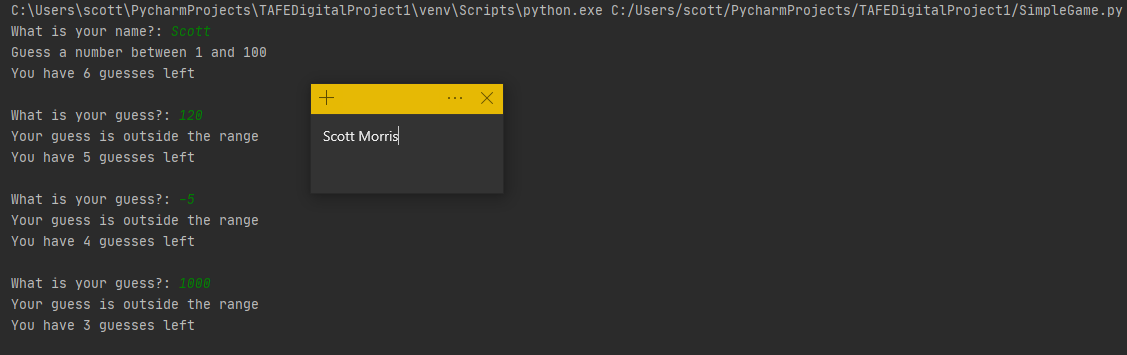
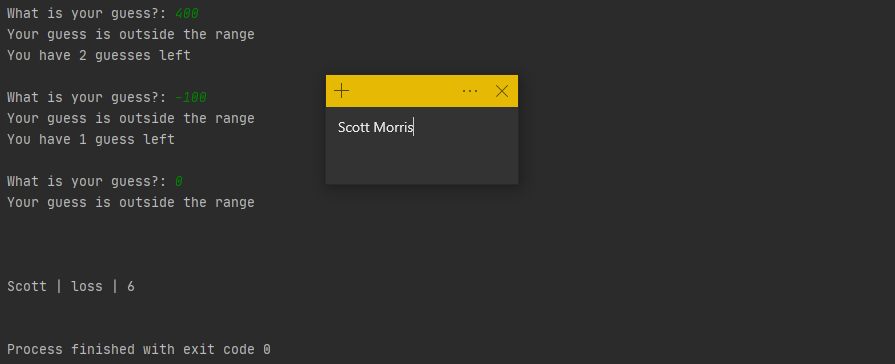
A screenshot of a cell phone

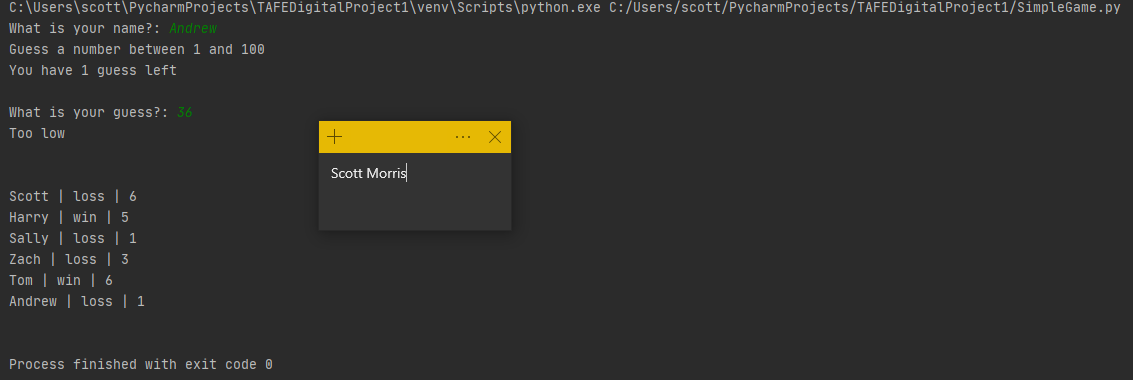
Description automatically generatedThe first round of testing involved interrogating the magic number and the number of guesses. The magic number would have to be printed out using a print statement while the number of guesses is already displayed.

The test cases were then started. The first test (TC01) case was to play the game, losing guesses until the game ended. There were no previous games in the statistics file.

The second test (TC02) was to win the game on the first guess. To achieve this the magic number was printed to the screen so that the first guess would be correct. There were no previous games stored in the statistics file.

The third test (TC03) had the tester input the character ‘F’ into their first guess. There were no previous games in the statistics file.

TC04 had the tester put in numbers that were outside the range of 1 and 100 until the number of guesses reached 0. There were no previous games stored in the statistics file.

The final test (TC05) had previous games inserted into the statistics file. The game was then played until the output was displayed.

# Software evaluation

|  |  |
| --- | --- |
| Software specification | Evaluation |
| A number between 1 and 100 (inclusive) is generated and assigned as the ‘magic number’ | I01 proved that a random number was being generated (between 1 and 100) each time the program was run. It appeared different numbers were produced each time |
| A number between 1 and 6 (inclusive) is rolled and assigned as the number of guesses | I02 also proved that the number of guesses changed each time that the program was run |
| If a number outside the range is input, a guess is ‘lost’ | TC04 had inputs below 1 and above 100 and each reduced 1 guess in the program for the input outside the correct range |
| If a number inside the range is input and is not equal to the ‘magic number’, a guess is lost | TC01 and TC05 both demonstrated that a number within the range but not equal to the magic number would reduce the number of guesses remaining as per the specification |
| If a number is input that is equal to the magic number, then the game ends and a win is assigned to that user (displayed in format [user name | win | number of guesses]) | TC02, the win test case had the user input the correct number from the start. The game then ended which aligned with the specification and the outcome of the game was displayed correctly as a win |
| If the number of guesses equals zero then the game ends and a loss is assigned to the user (displayed in format [user name | loss | number of guesses]) | TC01, TC04 and TC05 all ended in the number of guesses each being reduced to 0. This resulted in the game end output being displayed correctly and with a loss for the most recent game. |
| When the game ends, previous games should be read from statistics.txt and displayed to the screen in format: [user name | outcome | number of guesses] each on a new line | TC05 demonstrated this as the 5 games that were artificially inserted in at the start of TC05 were displayed along with the current game when the game of TC05 ended. |

# Software approval

Software is approved for installation by:

Name: John Smith

Role: Head Of Development

Signature: John Smith

Date: 31/05/20