TEST REPORT

Testing Environment

The operating system that our unit tests are carried out on is Windows 8 64bit. Google tests and Codeblocks IDE are used to set up the testing framework. Testing has been done at every stage of the game to ensure that the basics as well as the major, minor and additional features all work as they should.

Test Overview

This section reviews the tests done on each class as well as the functionality that has not yet been tested by the current framework.

Basic Functionality Tests

The tests on the Galaxian class test the basic functionality of the game. The main "Run" function is tested first to ensure the operation of the game. The default setting of GameState is tested which should initially be set to Uninitialized. The other settings of GameState are tested such as Exiting, Playing, Paused, ShowingSplash and ShowingMenu; these tests ensure that the states functions as they should and that the game is only exited when the state has been set to Exiting.

Furthermore the game-loop, game-menu and set/get position functions are tested for this class. The positioning of objects on the screen is based on the getPosition() and setPosition() functions; hence the need for testing.

PowerUp Tests

A PowerUp is obtained when the spaceship collides with one of the PowerUp objects. The PowerUp functions are tested to ensure that they give the player extra functionality during the game. This includes the two different weapon projectiles, the shield and the wrap-around functionality.

SpaceShip Tests

The spaceship is required to move horizontally and vertically between the main screen boundaries. The wraparound PowerUp allows the spaceship to move beyond these boundaries and appear on the opposite side; however this feature should only occur when this powerUp is activated. This functionality of the game is tested by testing the default setting of wrap-around.

Brain Tests

The Brains represent the Galaxians which attack the Galaxip; the role of the Brains is to move horizontally and vertically on the screen while shooting randomly at the spaceship. The BrainMovement test checks whether the coordinates of the Brains are changing as they move; this is done by testing the Brain objects in the Galaxian class. The BrainFire test checks the shooting functionality of the Brain; this is done by checking that bullets are fired at random from the Brains.

Crash Tests

There are four types of collision in this version of Galaxian which include an overlap of boundaries between: the spaceship and a Brain; the spaceship and a Brain projectile; a Brain and spaceship projectile and the spaceship and a PowerUp. Each of these is tested individually to ensure they function correctly.

Scoring Tests

The initial score of the game is tested to make sure that it is set to zero before the game begins. The player's score increases with the number of Brains killed. High scores are saved in between games; this is tested by manually inserting the high score into the output text file and then comparing it to the current game score. If the game score is lower the high score in the text file it should remain as is.

Game End Tests

The aim of the game is to kill all Brains on the screen; however the spaceship is not allowed to collide with the Brains or any of its projectiles. If the spaceship collides with a Brain or one of its projectiles, the player loses the game; tests are done to ensure that the game ends when this occurs. An error test is done which checks for 'game over' while the player is alive. If Brains remain displayed on the screen, the game should not display a win. This is tested by setting the Brain life-arrays to false and checking for a game-win.

Untested Functionality

In order to ease the use of game testing, one can create get and set functions to retrieve private variables. Not all the necessary get and set functions have been created in this version and hence a complete testing of the program cannot be achieved. Certain positioning, such as the Brain's projectile position has not tested as the position is updated and cannot be retrieved dynamically. Furthermore, some Brain positions are constantly being set and therefore do not have a stationary position; this causes difficulty in the testing of the Brain's position.

Test Report Conclusion

Overall, the test framework as it stands is reasonably thorough as it tests the main aspects of the program's and class's functionality. All of the performed tests pass which indicates that the program's functionality being tested is operating as intended. There are however test cases that can still be added to make this framework more extensive.