**Fishing Game Test Plan** (1CWK50)

|  |  |  |  |
| --- | --- | --- | --- |
| **Test** | **Expected Outcome** | **Actual Outcome** | **Corrective Action** |
| Swordfish moves along the y-axis at a speed dependent on the ‘swordfishSpeedY’ variable. | The Swordfish gradually moves towards the Fisherman at a speed dependent on the ‘swordfishSpeedY’ variable. | The code was not performed as the chosen value for the ‘swordfishSpeedY’ variable was not an integer. | Converted ‘swordfishSpeedY’ variable from an ‘int’ type to ‘float’ type to allow for decimals. |
| Swordfish moves along the x-axis at a speed dependent on the ‘swordfishSpeedX’ variable. | The Swordfish gradually moves following the Fisherman at a speed dependent on the ‘swordfishSpeedX’ variable. | The code was not performed as the chosen value for the ‘swordfishSpeedX’ variable was not an integer. | Converted ‘swordfishSpeedX’ variable from an ‘int’ type to ‘float’ type to allow for decimals. |
| Fishes’ y value gradually decreases as they move along the x-axis. | The fishes’ y value gradually decreases; moving towards the Fisherman as they move along the x-axis. | The fishes’ y value gradually decreases; moving towards the Fisherman as they move along the x-axis. However, the fishes move towards the Fisherman too quick although the lowest integer was chosen. | The fishes move towards the Fisherman each time they bounce off the wall. This was added to the ‘bounce()’ function:  y = y - fishSpeedY; |
| ‘gameOver()’ function | The ‘gameOver()’ function is performed when the game ends (by losing all 3 hearts or by hooking all the fish). If the player loses all 3 hearts, 0 hearts are shown and the ‘draw()’ function and displays “G A M E O V E R” text on screen. If all the fish are hooked, the ‘draw()’ function and displays “Y O U W I N !” text on screen. | The ‘gameOver()’ function is performed when the game ends (by losing all 3 hearts or by hooking all the fish). If the player loses all 3 hearts, 0 hearts are shown and the ‘draw()’ function and displays “G A M E O V E R” text on screen. If all the fish are hooked, the ‘draw()’ function and displays “Y O U W I N !” text on screen. | N/A |
| Losing a life | One life is lost when the swordfish collides with the fisherman's boat. | One life is lost when the swordfish collides with the fisherman's boat. | N/A |
| An image of the Fisherman shows when the code runs | The Fisherman is shown facing either left or right. | The Fisherman image does not show until the left or right key is pressed and unloads if a different key is pressed other than left or right. | if (fishermanDirection == 1)  image (fishermanImage, x, y);  else if (fishermanDirection == 0)  image (fisherman2Image, x, y);  Now, an image is always loaded and shown on-screen. |
| Left arrow key for Fisherman direction | When the ‘left’ arrow key is pressed the Fisherman image faces left and the hook & fishing line is positioned correctly. | The Fisherman faces left; the fishing line begins at the end of the Fisherman’s pole and the hook is positioned at the bottom of the fishing line. | N/A |
| Right arrow key for Fisherman direction | When the ‘right’ arrow key is pressed the Fisherman image faces right and the hook & fishing line is positioned correctly. | The Fisherman faces right; the fishing line begins at the end of the Fisherman’s pole and the hook is positioned at the bottom of the fishing line. | N/A |
| Left arrow key for Fisherman movement | When the ‘left’ arrow key is pressed the Fisherman moves left along the x-axis. | The Fisherman moves left by 10-pixels along the x-axis when the ‘left’ arrow key is pressed. | N/A |
| Right arrow key for Fisherman movement | When the ‘right’ arrow key is pressed the Fisherman moves right along the x-axis. | The Fisherman moves right by 10-pixels along the x-axis when the ‘right’ arrow key is pressed. | N/A |
| Right arrow key for hook & fishing line movement | When the ‘right’ arrow key is pressed the hook and fishing line move right along the x-axis in accordance to the Fisherman’s position. | The Fisherman’s hook and fishing line move right along the x-axis by 10-pixels when the ‘right’ arrow key is pressed in accordance with the Fisherman’s position. | N/A |
| Left arrow key for hook & fishing line movement | When the ‘left’ arrow key is pressed the hook and fishing line move left along the x-axis in accordance to the Fisherman’s position. | The Fisherman’s hook and fishing line move left along the x-axis by 10-pixels when the ‘left’ arrow key is pressed in accordance with the Fisherman’s position. | N/A |
| Right arrow key for swordfish x-axis movement | When the ‘right’ arrow key is pressed the swordfish move right along the x-axis in accordance to the Fisherman’s position. | The Fisherman’s hook and fishing line move right along the x-axis by 5-pixels when the ‘right’ arrow key is pressed in accordance with the Fisherman’s position. | N/A |
| Left arrow key for swordfish x-axis movement | When the ‘left’ arrow key is pressed the swordfish move left along the x-axis in accordance to the Fisherman’s position. | The Fisherman’s hook and fishing line move left along the x-axis by 5-pixels when the ‘left’ arrow key is pressed in accordance with the Fisherman’s position. | N/A |
| Down arrow key to drop hook | When the ‘down’ arrow key is pressed the hook moves down along the y-axis via an ‘if’ statement. | The Fisherman’s hook moves along the y-axis by 10-pixels when the ‘down’ arrow key is pressed. | N/A |
| Hook returns to its original position | When the ‘down’ arrow key is released the hook moves up along the y-axis by 1-pixel whilst the hook does not equal its original position via the ‘keyReleased()’ function and a ‘while’ statement. | The Fisherman’s hook moves back up to its original position along the y-axis until it reaches its original position. | N/A |
| ‘livesCounter’ variable decreases | When a life is lost (‘livesCounter’ variable decreases) the relevant number of hearts/lives is shown in the top-left of the screen via several ‘if’ statements. | The relevant number of hearts/lives is shown in the top-left of the screen dependent upon the ‘livesCounter’ variable. This was tested via setting the ‘livesCounter’ at different integers and causing the boat to collide with the Swordfish which results in a life lost. | N/A |
| The title & instruction strings function correctly. | The title & instruction strings (‘instructions1’, ‘instructions2’ & ‘instructions3’) show on the splash screen when called. | The title & instruction strings (‘instructions1’, ‘instructions2’ & ‘instructions3’) show on the splash screen when called. | N/A |
| The score statement is functioning correctly within the ‘collisions’ function:  score = score + 100; | When the hook collides with a fish belonging to the fishArray, 100 score is added to the score tally in the top-right of the screen. | When the hook collides with a fish belonging to the fishArray, 100 score is added to the score tally in the top-right of the screen. | N/A |
| The swordfish’s x value is randomly generated in the desired area. | The swordfish’s x value is randomly generated. Between:  (random(F1.x - 150, (F1.x + 125) + 150)) | The swordfish’s x value is randomly generated. Between:  (random(F1.x - 150, (F1.x + 125) + 150)) | N/A |
| The game initiates once a key is pressed. | Once a key is pressed on the keyboard, the game begins and images load. | Once a key is pressed on the keyboard, the game begins and images load. | N/A |
| Fish are created and loaded into the game dependent on the values in the variables. | Fish are created and loaded into the game dependent on the values in the ‘fishColumns’ & ‘fishRows’ variables via a nested for loop. | Fish are created and loaded into the game dependent on the values in the ‘fishColumns’ & ‘fishRows’ variables via a nested for loop. | N/A |
| The ‘bounce()’ function. | When the fish approach the lowest point of the x-axis (left wall) they go in the opposing direction and move towards the Fisherman. | The fish bounce off the left, however, they bounce slightly before the wall. | Changed from:  (x < 0)  To:  (x <= 0) |
| The ‘bounce()’ function. | When the fish approach the highest point of the x-axis (right wall) they go in the opposing direction and move towards the Fisherman. | The fish bounce off the right, however, they bounce slightly before the wall. | Changed from:  (x >= width)  To:  (x >= width - 80) |
| Fish disappear once the hook has collided with that fish. | The fish disappears once they have collided with the hook by setting the Fish variable: fishArray[x][y].visible to false. | The fish disappears once they have collided with the hook by setting the Fish variable: fishArray[x][y].visible to false. | N/A |
| Animated Fishes. | A slightly different Fish image is shown dependent on their x values to appear animated. | A slightly different Fish image is shown dependent on their x values to appear animated. | N/A |
| Animated Swordfish. | A slightly different Swordfish image is shown dependent on its y value to appear animated. | A slightly different Swordfish image is shown dependent on its y value to appear animated. | N/A |
| Game ends when Fish reach the boat. | When a fish’s y value is equal to or greater than the Fisherman’s y value plus its height; the game ends. | When a fish’s y value is equal to or greater than the Fisherman’s y value plus its height; the game ends. | N/A |