**Program Testing**

**Testing Plan:**

Game Functions:

Buttons: Distribute Items

Movement: Forward

Backwards

Left

Right

Combination

Mouse click in game

Items: SQL table to List conversion  
 Item to Treeview

Random item selection   
 Random Item placement

Timer: Duration selection   
 Timer Countdown   
 Game Over

Object Collision: Collision with barrier

Contact with item

Sorting Algorithms: Bubble Sort  
 Insertion Sort  
Search Algorithms:   
 Linear Search

Item Table: item information added

|  |  |  |  |
| --- | --- | --- | --- |
| **Test No.** | **Test Expected Value** | **Pass/ Fail** | **Comment** |
| **1** | When Distribute items is pressed the game items & robot should randomly placed in arena with the timer starting at 1:00 | Pass | The game started correctly and the timer started counting down from 1 minute. |
| **2** | When the robot is in the arena it should move in the corresponding direction as the arrow key press by users. | Partial Pass | The robot was able to move forward, backwards & Left when its related key was pressed. It didn’t respond to right arrow key presses |
| **3** | When the robot is in the arena it should move to the area mouse clicked by the user | Pass | The robot moved to the location where the user clicked |
| **4** | The robot should ‘bounce’ off a wall when contact is made between the 2 objects. | Fail | The robot doesn’t respond to walls and goes through them. |
| **5** | When an item is collected it should disappear from the arena and a random item from a random list of 9 (from the SQL Database) should be added to the Treeview | Partial Pass | When the robot makes contact with the item a random product is placed into the treeview but it doesn’t disappear from the arena |
| **6** | When the timer reaches 00:00 the game should end and user interaction with the arena should be disabled | Fail | The timer reaches 00:00 but users can still move & collect items |
| **7** | When the user selects a timer duration the game duration should change to the duration they selected | Pass | The timer changes to the corresponding duration |
| **8** | The products in the basket should be sorted into ascending/ descending order (based on product name) depending on what way the product arrow is facing (up = ascending etc.) when the user selects the arrow | Pass | The products are sorted by product name in ascending or descending order depending on what was selected by the user. |
| **9** | The products in the basket should be sorted into ascending/ descending order (based on price) depending on what way the price arrow is facing (up = ascending etc.) when the user selects the arrow | Pass | The products are sorted by price when the arrow is selected. |
| **10** | When the user types a search in the search bar a message stating whether or not the product is in the basket. (Found = “<product name> was found in the basket, Not Found =”<product name> was not found”) | Fail | Even if the product is in the basket the linear search fails to find the product and outputs the ‘Not found’ message. |
| **11** |  |  |  |
| **12** |  |  |  |
| **13** |  |  |  |
| **14** |  |  |  |
| **15** |  |  |  |
| **16** |  |  |  |
| **17** |  |  |  |
| **18** |  |  |  |
| **19** |  |  |  |
| **20** |  |  |  |
| **21** |  |  |  |
| **22** |  |  |  |
| **23** |  |  |  |
| **24** |  |  |  |
| **25** |  |  |  |