1. The solution must support missiles launched by the user
    a. The solution must initiate a missile launch when the user clicks the right mouse button.
    b. The solution must initiate a missile launch when the user clicks the space bar.
    c. The solution must limit the number of active missiles on the screen to no more than five (5) at any given time. (an “active missile” is any missile currently being displayed on the screen)
    d. The solution must remove the missile from being active if it goes off the screen.
    e. The solution must maintain the same constant speed for all missiles
    f. The solution must launch missiles from the center of the bottom of the screen when a launch is initiated
    g. The solution must maintain the same constant direction for all missiles to be vertically straight up from the launched position.
    h. The solution must detect when a missile “hits” a ship
    i. The solution must display an explosion at the point where a missile “hits” a ship
    j. The solution must remove the missile and ship after the missile “hits” the ship
    k. The solution must keep a count of all “hits”
2. The solution must support ships launched by the solution
    a. The solution must initiate a ship launch when the system detects there are no active ships.
    b. The solution must support multiple types of ships based on a configurable value.
    c. The solution must display the appropriate image based on the type of ship when the ship is active
    d. The solution must randomly initiate a ship launch based on a configurable rate where the default is 30% of the time
        i. The solution must randomly choose from available ship types when a launch is initiated, giving all types equal chance of being launched.
        ii. The solution must limit the number of active ships on the screen to no more than ten (10) at any given time. (an “active ship ” is any ship currently being displayed on the screen)
        iii. The solution must randomly choose a location to launch a ship from when initiated
            1. The system must randomly choose to launch the ship from the left side of the screen or the right side of the screen
            2. The system must randomly choose a row in the top two-thirds of the screen to launch the ship from
            3. The system must assign the speed of the ship based on the type of the ship being initiated

        4. The system must assign the direction of the ship based on which side of the screen it is being launched from (if from the left, direction goes left to right; if from the right, direction goes right to left)

    e. The solution must remove the ship from being active if it goes off the screen.

3. The solution must end the game when the “hit” count has reached ten (10)
4. The solution must end the game when the user clicks the left mouse button
5. The solution must end the game when the user clicks the esc button
6. The solution must end the game if the user has not initiated a missile launch in the last 5 minutes.

Tests

1a;

Test 1: Does this solution launch a missile when the user clicks the right mouse button? Action: Press the right mouse button to verify and see if a missile is launched. Record results.

Test 2: Do any of the other mouse buttons launch a missile? Action: Press all mouse buttons except the right mouse button. Record results.

Test 3: Do any buttons on the keyboard launch a missile? Action: Press every key on the keyboard. Record results.

Test 4: What happens when the right mouse button is held? Action: Press and hold the right mouse button. Record results.

1b;

Test 1: Does this solution work? Action: Press the spacebar to verify and see if a missile is launched. Record results.

Test 2: Do any of the mouse buttons launch a missile? Action: Press all mouse buttons. Record results.

Test 3: Do any other buttons on the keyboard launch a missile? Action: Press every key on the keyboard except the spacebar. Record results.

Test 4: What happens when the spacebar is held? Action: Press and hold the right mouse button. Record results.

1c;

Test 1: Can one missile be active on the screen? Action: Press the missile launch button once. Record results.

Test 2: Can two missiles be active at the same time? Action: Press the missile launch button twice. Record results.

Test 3: Can three missiles be active at the same time? Action: Press the missile launch button three times. Record results.

Test 4: Can four missiles be active at the same time? Action: Press the missile launch button four times. Record results.

Test 5: Can five missiles be active at the same time? Action: Press the missile launch button five times. Record results.

Test 6: Can six missiles be active at the same time? Action: Press the missile launch button six times. Record results.

Test 7: If I press the missile launch button many times can there be more than five active missiles at once? Action: Press the missile launch button as many times as can be done in the shortest amount of time. Record results.

1d;

Test 1: Has the missile been removed from being active once it is off screen? Action: Bring up memory calculator for the computer being used, record initial memory usage. Fire one missile, record the difference between before the missile was launched, after the missile was launched and after the missile has left the screen on the memory calculator. Record results, repeat actions as necessary.

1e;

Test 1: Do the missiles maintain a constant speed? Action: Set up timer, launch missile, time how long the missile is on screen. Record results, repeat these actions as many times as necessary.

1f;

Test 1: Do all missiles launch from the bottom center of the screen? Action: Determine the bottom center of the screen. Launch a missile, does the missile come from the determined bottom center of the screen. Record results, repeat these actions as many times as necessary.

1g;

Test 1: Do all missiles maintain the same direction? Action: Launch missile, observe its direction, launch another missile does it maintain the same direction? Record results, repeat as many times as necessary.

Test 2: Do all missiles maintain a straight directory? Action: Launch a missile, is it moving in a straight line? Record results, repeat as many times as necessary.

Test 3: Do all missiles move vertically? Action: Launch missile, does it move vertically? Record results, repeat as many times as necessary.

1h;

Test 1: Do all missiles "hit" the ships? Action: Bring up memory calculator for the computer being used, record initial memory usage. launch missile at ship, record the difference between before the missile "hit," when the missile "hit" and after the missile "hit." Record results, repeat as many times as necessary.

1i;

Test 1: Do all missiles explode when they "hit" a ship? Action: launch missile at ship, observe the action of the missile "hitting" the ship. Record results, repeat as many times as necessary.

1j;

Test 1: Is the missile that "hit" the ship, and the ship removed once the missile "hits" the ships? Action: Bring up memory calculator for the computer being used, record initial memory usage. launch missile at ship, record the difference between before the missile "hit," when the missile "hit" and after the missile "hit." Record results, repeat as many times as necessary.

> Test 1A: Has the missile been removed? Action: Check screen and memory to see if the missile has been removed. Record results
>
> Test 1B: Has the ship been removed? Action: check screen and memory to see if the ship has been removed. Record results

1k;

Test 1: does the solution keep a count of all “hits?” Action: Bring up memory analyzer for the computer being used, record initial memory usage. Launch missiles at ships, record hits. Determine difference between memory and recorded hits. Record results.

2a;

Test 1: Does the solution initiate a ship launch if there are no active ships? Action: Bring up memory analyzer for the computer being used, record initial memory usage. Remove all ships on screen. Record results.

2b;

Test 1: Does the solution support all types of ships? Question: Is the configurable value configured by the programmer, the user, or through a process of the solution? Action: Get information on all ships, bring up memory analyzer for the computer being used, record memory.

Action 1A: If the value is configured by the programmer, record if the ship matches what the programmer says it should be.

Action 1B: If the value is configured by the user, configure the program for each ship. Record witch ship comes up with each configuration.

Action 1C: If the value is configured by a process of the solution, move through the solution and record all ships.

Record results.

2c;

Test 1: Does the ship displayed match the type of ship that is active? Action: Bring up memory analyzer for the computer being used, record memory. Compare memory of ship to ship on screen. Record results

2d;

Test 1: Does the solution initiate a ship launched based on the configurable rate? Question: How configurable is the rate? Action: set up a timer in seconds, time how long the solution is running record every time a ship is launched. Divide time by number of ships launched to see if this matches the rate. Record results

2di;

Test 1: Does the solution give all ship types an equal chance to be launched? Action: while the solution is running, set a count of every type of ship that is launched. Compare the count of each type of ship. Record results

Test 2: Does the solution choose randomly from the available ship types? Action: while the solution is running, set a count of every type of ship that is launched. Compare the count of each type of ship. Record results, repeat as necessary.

2dii;

Test 1: Does the solution limit the number of active ships to ten (10)? Action: while the solution is running count the number of active ships. Record results, repeat as necessary.

2diii;

Test 1: Does the solution randomly choose the launch location of the ships when initiated? Action: Bring up memory analyzer for the computer being used, record memory to determine the location of the ships to be launched. Record results.

2diii1;

Test 1: Does the solution randomly choose which side of the screen (left or right) that the ships are launched from? Action: Bring up memory analyzer for the computer being used, record memory to determine the location of the ships to be launched. Record results, repeat as necessary.

2diii2;

Test 1: Does the solution randomly choose a row to launch the ships from? Action: Bring up memory analyzer for the computer being used, record memory to determine the location of the ships to be launched. Record results, repeat as necessary.

Test 2: Are all of the rows chosen in the top two-thirds of the screen? Action: observe the screen while the solution is running, record the launch location of any active ships. Record results, repeat as necessary.

2diii3;

Test 1: Do all ships move at their assigned speed? Question: What are the Assigned speeds of each ship type? Action: get info on the speed of each ship type, time how long it takes for each ship type to cross the screen. Compare that time to the speed of each ship type. Record results.

2diii4;

Test 1: Do all ships follow the assigned path based on their initial launch point (if from the left, ship goes left to right, if from the right ship goes right to left)? Action: Bring up memory analyzer for the computer being used, record memory to determine ship launch points and which ships are active. Record which ships are active. Record results.

2e;

Test 1: Are all ships removed from being active once they are off screen? Action: Bring up memory analyzer for the computer being used, record memory to determine if a ship is inactive once it goes off the screen. Record results.

3;

Test 1: Does the solution end the game when the “hit” count reaches ten (10)? Acton: count “hits”. Record how many hits it takes before the game ends. Record results, repeat as necessary.

4;

Test 1: Does the solution end the game when the left mouse button is clicked? Action: Press the left mouse button. Record results.

Test 2: Do any of the other mouse buttons end the game? Action: Press all mouse buttons except the left mouse button. Record results.

Test 3: Do any buttons on the keyboard end the game? Action: Press every key on the keyboard. Record results.

Test 4: What happens when the left mouse button is held? Action: Press and hold the left mouse button. Record results.

5;

Test 1: Does the solution end the game when the esc button is clicked? Action: Press the esc button. Record results.

Test 2: Do any of the other buttons on the keyboard end the game? Action: Press all keyboard buttons except the esc button. Record results.

Test 3: Do any buttons on the mouse end the game? Action: Press every key on the mouse. Record results.

Test 4: What happens when the esc button is held? Action: Press and hold the esc button. Record results.

6;

Test 1: Does the solution end the game if left alone for 5 minutes? Action: Run solution but do not interact with the solution. Record how long it takes for the solution to end the game. Record results.

Test 2: does the solution end the game if left alone for 5 minutes after being played? Action: play the game for short while then leave it alone. Record how long it takes for the solution to end the game. Record results.

Points of concern.

Requirements 1a and 1b are both requirements to initiate the launch of a missile using different button presses. Is this accurate? Is this how the solution is supposed to work?

Requirement 2b talks about a configurable value, how is this value configured?

Requirement 2d talks about a configurable rate, what is the rate of change to this configurable rate?

Requirements 4 and 5 are both requirements for ending the game by pressing a button. Is this accurate? Is this how the solution is supposed to work?