CSCI 4250-002 Testing Exercises 09/27/2023

Chicken Contenders

Trent Williams Tyler Burleson Tyson Bailey Bryson Brandon

Test Cases:

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.
        i. [TC:01]
            1. PRECOND: Game is running, no input
            2. ACTION: On click - right mouse button
            3. POSTCOND: a single missile appears
        ii. [TC:02]
            1. PRECOND: Game is running, no input
            2. ACTION: user is holding “w” and right clicks mouse
            3. POSTCOND: fires a single missile

    b. The solution must initiate a missile launch when the user clicks the space bar.
        i. [TC:03]
            1. PRECOND: Game is running, no input
            2. ACTION: user hits space bar
            3. POSTCOND: a single missile should fire
        ii. [TC:04]
            1. PRECOND: Game is running, no input
            2. ACTION: while user is holding “w”, they hit space
            3. POSTCOND: a single missile should fire

    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)
        i. [TC:05]
            1. PRECOND: 5 missiles on screen
            2. ACTION: user attempts to fire
            3. POSTCOND: no missiles will fire
        ii. [TC:06]
            1. PRECOND: 2 missiles on screen
            2. ACTION: user fires 2 missiles
            3. POSTCOND: 4 missiles on screen
        iii. [TC:06]
            1. PRECOND: 3 missiles on screen
            2. ACTION: user fires 3 missiles
            3. POSTCOND: 2 missiles fires and 1 doesn’t
        iv. [TC:07]
            1. PRECOND: 0 missiles on screen
            2. ACTION: user fires 8 times
            3. POSTCOND: only 5 missiles would appear on screen

d. The solution must remove the missile from being active if it goes off the screen.
    i. [TC:08]
        1. PRECOND: 4 missiles on screen
        2. ACTION: all 4 flies off screen
        3. POSTCOND: removes off screen missiles from screenArray[] and remove 4 from missileCount
    ii. [TC:09]
        1. PRECOND: 3 missiles on screen
        2. ACTION: 1 missile flies off screen
        3. POSTCOND: removes off screen missile from screenArray[] and remove 1 from missileCount

e. The solution must maintain the same constant speed for all missiles
    i. [TC:10]
        1. PRECOND: const setSpeed = "…"
        2. ACTION: missile fires at setSpeed
        3. POSTCOND: const missileSpeed = setSpeed

f. The solution must launch missiles from the center of the bottom of the screen when a launch is initiated
    i. [TC11]
        1. PRECOND: user inputted right click
        2. ACTION: missiles fire
        3. POSTCOND: missiles travel in an upwards straight line
    ii. [TC12]
        1. PRECOND: user inputted space
        2. ACTION: missiles fire
        3. POSTCOND: missiles travel in an upwards straight line
    iii. [TC13]
        1. PRECOND: user inputted right click 3 times
        2. ACTION: 3 missiles fire
        3. POSTCOND: 3 missiles travel in an upwards straight line
    iv. [TC14]
        1. PRECOND: user inputted space 3 times
        2. ACTION: 3 missiles fire
        3. POSTCOND: 3 missiles travel in an upwards straight line

g. The solution must maintain the same constant direction for all missiles to be vertically straight up from the launched position.
    i. [TC11]
        1. PRECOND: user inputted right click
        2. ACTION: missiles fire
        3. POSTCOND: missiles travel in an upwards straight line
    ii. [TC12]
        1. PRECOND: user inputted space
        2. ACTION: missiles fire
        3. POSTCOND: missiles travel in an upwards straight line
    iii. [TC13]
        1. PRECOND: user inputted right click 3 times
        2. ACTION: 3 missiles fire
        3. POSTCOND: 3 missiles travel in an upwards straight line
    iv. [TC14]
        1. PRECOND: user inputted space 3 times
        2. ACTION: 3 missiles fire
        3. POSTCOND: 3 missiles travel in an upwards straight line

h. The solution must detect when a missile “hits” a ship
    i. [TC15]
        1. PRECOND: missile has been fired and ship is on screen
        2. ACTION: missile graphic overlays ship graphic
        3. POSTCOND: ships are hit, remove active missile and active ship
        4. POSTCOND: hitCount++
        5. POST COND: display explosion
    ii. [TC16]
        1. PRECOND: two ships overlayed on each other
        2. ACTION: missile hits
        3. POSTCOND: ships are hit, remove active missile and top ship
        4. POSTCOND: hitCount++
        5. POST COND: display explosion

i. The solution must display an explosion at the point where a missile “hits” a ship
    i. [TC15]
        1. PRECOND: missile has been fired and ship is on screen
        2. ACTION: missile graphic overlays ship graphic
        3. POSTCOND: ships are hit, remove active missile and active ship
        4. POSTCOND: hitCount++
        5. POST COND: display explosion
    ii. [TC16]
        1. PRECOND: two ships overlayed on each other
        2. ACTION: missile hits
        3. POSTCOND: ships are hit, remove active missile and top ship
        4. POSTCOND: hitCount++
        5. POST COND: display explosion

j. The solution must remove the missile and ship after the missile “hits” the ship
    i. [TC15]
        1. PRECOND: missile has been fired and ship is on screen
        2. ACTION: missile graphic overlays ship graphic
        3. POSTCOND: ships are hit, remove active missile and active ship
        4. POSTCOND: hitCount++
        5. POST COND: display explosion
    ii. [TC16]
        1. PRECOND: two ships overlayed on each other
        2. ACTION: missile hits
        3. POSTCOND: ships are hit, remove active missile and top ship
        4. POSTCOND: hitCount++
        5. POST COND: display explosion

k. The solution must keep a count of all “hits”
    i. [TC15]
        1. PRECOND: missile has been fired and ship is on screen
        2. ACTION: missile graphic overlays ship graphic
        3. POSTCOND: ships are hit, remove active missile and active ship
        4. POSTCOND: hitCount++
        5. POST COND: display explosion
    ii. [TC16]
        1. PRECOND: two ships overlayed on each other
        2. ACTION: missile hits
        3. POSTCOND: ships are hit, remove active missile and top ship
        4. POSTCOND: hitCount++
        5. POST COND: display explosion

Chicken Contenders

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.
        i. [TC17]
            1. PRECOND: no ships on screen
            2. ACTION: system detects there are no ships
            3. POSTCOND: systems generate ship
        ii. [TC18]
            1. PRECOND: ship is on screen
            2. ACTION: system detects there are active ships
            3. POSTCOND: system doesn't generate ship

    b. The solution must support multiple types of ships based on a configurable value.
        i. [TC18]
            1. PRECOND: configVal = 2;
            2. ACTION: system pulls ship types from list
            3. POSTCOND: only displays the 2 ship types images
        ii. [TC19]
            1. PRECOND: configVal = 1;
            2. ACTION: system pulls ship type from list
            3. POSTCOND: only displays the 1 ship type image
        iii. [TC20]
            1. PRECOND: configVal = 5;
            2. ACTION: system pulls ship types from list
            3. POSTCOND: only displays the 5 ship types image

    c. The solution must display the appropriate image based on the type of ship when the ship is active
        i. [TC18]
            1. PRECOND: configVal = 2;
            2. ACTION: system pulls ship types from list
            3. POSTCOND: only displays the 2 ship types images
        ii. [TC19]
            1. PRECOND: configVal = 1;
            2. ACTION: system pulls ship type from list
            3. POSTCOND: only displays the 1 ship type image
        iii. [TC20]
            1. PRECOND: configVal = 5;
            2. ACTION: system pulls ship types from list
            3. POSTCOND: only displays the 5 ship types image

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.
        [TC21]
            PRECOND: multiple ship types = configShipType = 3
            ACTION: system chooses ship type randomly from 1-3
            POSTCOND: chosen ship type is displayed

    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)
        [TC22]
            PRECOND: 10 ships on screen
            ACTION: chanceToGenerate = false
            POSTCOND: do nothing
        [TC23]
            PRECOND: 5 ships on screen
            ACTION: chanceToGenerate = true
            POSTCOND: generate ship

    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.

Chicken Contenders

7. Combine missile test case; if no missiles are active, you click the right mouse button 6 times rapidly, you should see exactly 5 missiles.