Class Exercise for Testing

Do you need working code in order to design tests in the Test Phase? Why

You do not need working code to design tests. The reason behind this is that in some cases you can begin testing, such as verification testing, at the start with the requirements analysis. After this you can design tests for the project and then create the project based around these tests.

What is the difference between verification and validation?

1. Identify all the verification activities that occur in the Requirements, Design and Implementation phases of the SE Lifecycle
2. Identify all the validation activities that occur in the Requirements, Design and Implementation phases of the SE Lifecycle

Verification begins during the requirements analysis, design phase, and testing phase of the SE Lifecycle. During the design phase verification is design reviews and requirements traceability. During the testing phase verification is verifying that the solution implements the design of the project. This is done by starting with functional/low level integration testing, moving to subsystem or integration testing, and then ending with systems testing. In short, the verification tests trace the design specifications to make sure the entire design is covered.

Validation testing is trying to show that the solution/project does exactly what the customer/client had asked for. This is considered a subset of the Verification testing, so validation testing comes into play mostly in the implementation phase of the SE lifecycle. This testing involves the customer/client looking at the solution and identifying that it meets what they wanted. This means that the solution must be accepted by the customer/client before it is passed.

Solution: a game where the player shoots a missile at ships passing across the screen

1. The solution must support missiles launched by the user.
   1. The solution must initiate a missile launch when the user clicks the right mouse button.
      1. When playing the game have user right click, observe if a missile is launched.
   2. The solution must initiate a missile launch when the user clicks the space bar.
      1. When playing the game have user press space bar, observe if a missile is launched.
   3. 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)
      1. If missiles are successfully launched, then spam launch buttons to see if missile launches will stop if at most 5 missiles are on the user’s screen.
   4. The solution must remove the missile from being active if it goes off the screen.
      1. Fire 5 missiles, wait until one is off screen, see you can fire another missile, if you can then the missile was not active, if you can’t then failed test the missile is still active.
   5. The solution must maintain the same constant speed for all missiles
      1. For the test have 5 missiles automatically shot in the same direction at the same time and see if they are all moving at exactly the same speed.
   6. The solution must launch missiles from the center of the bottom of the screen when a launch is initiated.
      1. When solution is running have user launch missiles, observe if the missiles are launching from the desired location on the screen.
   7. The solution must maintain the same constant direction for all missiles to be vertically straight up from the launched position.
      1. When launching missiles, observe if the missiles stray off of their linear path established by launch position.
   8. The solution must detect when a missile “hits” a ship
      1. When a missile gets close to a ship, observe if the solution detects that the missile had hit the ship.
   9. The solution must display an explosion at the point where a missile “hits” a ship.
      1. If the solution is able to detect ships being hit, observe if an explosion is displayed where the ship that got hit is located at.
   10. The solution must remove the missile and ship after the missile “hits” the ship.
       1. If the solution is able to detect a hit and display an explosion then observe if the missile, explosion, and battleship are removed from the screen.
   11. The solution must keep a count of all “hits”
       1. If the solution detects hits the observe if the counter displayed to the user goes up by 1.
2. The solution must support ships launched by the solution
   1. The solution must initiate a ship launch when the system detects there are no active ships.
      1. Initiate the game with no ships and observe if a ship is automatically initiated because there are no ships.
      2. Have the user clear the screen of ships and observe if a ship is initiated when the screen is clear.
   2. The solution must support multiple types of ships based on a configurable value.
      1. Determine this value and then change this value to see which types of ships are initiated by the solution.
   3. The solution must display the appropriate image based on the type of ship when the ship is active
      1. Set the configurable value at the desired values to test if the appropriate image is displayed based on the type of ship that corresponds to the value.
   4. The solution must randomly initiate a ship launch based on a configurable rate where the default is 30% of the time
      1. The solution must randomly choose from available ship types when a launch is initiated, giving all types of equal chance of being launched.
         1. Have the configurable value be randomized giving each ship an equal chance of being picked then observe if all of the ships are displayed properly keeping count of how many times each type of ship is displayed. At the end of this take the average for each ship and determine if they are close enough to be considered equal.
      2. 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)
         1. Have 10 ships displayed on the screen and observe if an 11th ship gets displayed on the screen.
      3. 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
            1. On start of the solution observe if ships are coming from both the right and 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
            1. Determined on start of the solution if the solution is displaying ships from the left and right row in the top two-thirds of the screen. Also have the user shoot the ships to see if when more ships are initiated if those are also randomly placed in these rows.
         3. The system must assign the speed of the ship based on the type of the ship being initiated
            1. Assign all different types of ships different speeds. In order to test this, initiate all different types of ships and observe if they are moving at different speeds on the screen.
         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)
            1. When ship is initiated have it move in a linear direction based of initiation direction like the missiles direction system.
   5. The solution must remove the ship from being active if is goes off the screen.
      1. To test this initiate 10 different types of ships and allow them all to move off the screen, then see if more ships are initiated.
3. The solution must end the game when the “hit” count has reached ten (10)
   1. When the counter for the hits ends 10 see if the game ends or continues, so we need a check to see if the game is running.
4. The solution must end the game when the user clicks the left mouse button.
   1. Have the user left click the mouse and see if the game ends.
      1. Side note: I believe this would be a bad idea because we are all human and would left click to shoot a missile on accident, but it would keep ending the game.
5. The solution must end the game when the user clicks the esc button.
   1. Have the user click the esc key and determine if the game ends.
6. The solution must end the game if the user has not initiated a missile launch in the last 5 minutes.
   1. Have an internal timer that resets when a missile is launched, and then it hits 5 minutes end the game. To test this, have the user start the game, shoot some ships, and then stop shooting ships for 5 minutes to see if the game ends.