Assignment 5: Spaceship

Due: Friday, February 7 at 11:59 PM

In this assignment, you will create a simple spaceship game.

There is no starter code, but you are free to start from any code from this year’s labs or lectures. You can see a video demo of an example solution here: https://youtu.be/ghGr5iIN_9E

Requirements and Grading

You will create two sketches:

In A5_basic, you will implement as much required functionality as you can without any enhancements. Your goal is to match the video demonstration. This is the sketch the markers will evaluate for required functionality marks. Do this first.

In A5_enhanced, you can extend and deviate somewhat from the required functionality to show off your programming skill and design creativity. Do this second.
[17 marks] Correctness

**Required Functionality**

These are roughly ordered by difficulty. The last bullet is very challenging.

- Use a 400 by 600 canvas.

- [1] Draw a “spaceship” at the bottom of the canvas (see demo or screenshot). The rectangle used as the body of the spaceship is 40 by 40. The center of the rectangle is 70 pixels from the bottom of the canvas. **You may not use rectMode(CENTER).**

- [1] Draw three lines under the spaceship (see demo or screenshot). The lines are 20 pixels long. Two lines are 10 pixels from the rectangle, and one line 20 pixels.

- [1] Make the spaceship and three lines move left and right with the x position of the mouse.

- [1] Draw two “stars” (circles of two different sizes). You can choose the size for these stars.

- [1] Make the stars fall from the top of the canvas to the bottom of the canvas. They **must** move at different speeds. You can choose the speed of each.

- [1] Once the stars move past the bottom end of the canvas, they must reappear from the top of the canvas and repeat the falling behavior. They **must** appear at random x position and at random times.

  **HINT:** Randomize the start and endpoint of the stars to make it appear at random times.

- [1] Draw “enemy ship” (red rectangle) as shown on the demo or screenshot. The top left corner of the rectangle is 50 pixels from the top of the canvas. It is 50 by 30.

- [1] The score is displayed in the top-left area of the canvas. The number updates only when the missile hits the enemy ship (see below).

- [1] The number of available missiles is shown as vertical white lines in the top-right area of the canvas. It should display five missiles when the game starts. The vertical lines are 10 pixels apart from each other. The rightmost line is 30 pixels from the right side of the canvas.

- [1] When the mouse button is pressed, a missile (short white line) is launched from the tip of the spaceship (see demo), and it moves towards the top of the canvas. The missile is not launched when there is no missile left in the spaceship.

- [1] When the missile launches, the number of available missiles decreases by one. (That is, one fewer number of vertical white lines are drawn.)

  **HINT:** Use variable and loop to control the number of lines being drawn.
• [1] The missile disappears when it reaches the top of the canvas. No points are scored.

• [1] The missile disappears when it hits the enemy ship. One point is scored.

• [1] When the missile hits the enemy ship, a new enemy ship appears near the top of the canvas (50 pixels from top), at a random x location (see demo). The ship is always the same size at 50 by 30.

• [1] A new missile can only be launched when no other missile is currently in the air.

• [1] When the missile is in the air, it does not move with the mouse or spaceship, but the spaceship still moves with the mouse.

  HINT: Use variable to check whether the missile is in the air.

• [1] Pressing ‘x’ resets the score and the number of available missiles, that is, puts the score back to 0 and the number of missiles to five (see demo).

**General Correctness**

• One mark will be deducted for files or directories named incorrectly (the zip file, etc.)

• One mark will be deducted if the wrong canvas size is used.

• One or more marks will be deducted if variable names were declared, but not used.

• One or more marks will be deducted if the program crashes (depending on the severity).

• **Assignments that do not run may receive a grade of 0.** Even if you don't complete the entire assignment, don't leave it in a broken state. Make sure it runs so we can find ways to give you part marks.

**[2 marks] Coding Style and Efficiency**

Coding style is evaluated in both sketches.

Follow the course coding style for whitespace and comments. Consult the “Code Style Guide” on Learn. For example:

• Comment your code appropriately. Avoid superfluous comments.
• Correctly and consistently indent your code blocks.
• Use correct inline spacing for variable declaration and assignment.
• Use good line spacing to chunk sections of your code.
• Pay special attention to inline spacing for your conditional statements.
• Use semicolons.
• Use `let` or `const`, never `var`. 
One or more marks may be deducted for solutions that have obvious inefficiencies.

- Variables that are declared or assigned, but not used.
- Unnecessary variables that are duplicates of other variables.
- Unnecessarily repeating the same code in multiple places.
- Too many “magic numbers”: the same number appears in many places indicating a variable should have been used instead.

[ 2 marks ] Functionality or Visual Design Enhancements

Once you have basic functionality working (the “correctness” requirements above), enhance the functionality or the visual design in a sketch called A5_enhanced.

These basic requirements must still be followed:

- The canvas size must be the same size as A5_basic.
- It should still be launching something from the bottom.

Some ideas:

- Make a more exciting looking display (e.g., more stars), maybe with more randomness, better shapes (e.g., for enemy ship), or different background.
- Create more enemy ships.
- Allow spaceship to shoot multiple missiles while launched missiles are still in the air.
- Show an animation when the missile hits the enemy ship.
- Make it more challenging and fun. The game ends if the spaceship is struck by a star.

Remember to describe how your enhanced version works in a comment at the top of your code. The TAs need to know how to use your program so they can see how cool it is!

It’s ok to not add enhancements or be extra creative if you’re running out of time: a correct basic solution with excellent coding style will still achieve a grade of 88%.

Restrictions

In general, you may not use any functions, libraries, or statements not covered in lecture or labs unless specifically exempted below or in a post by a TA or instructor on this assignment discussion board. For example:

- NO circle or square functions
- NO translate(), rotate(), or scale() functions
- NO arrays and no classes
- You MAY use bezier, arc, and other standard drawing functions

If in doubt, make a post to ask about using a specific statement of function.

Functionality marks will be deducted for using forbidden functions/statements.
Submitting

Zip your assignment sketch folders (A5_basic and A5_enhanced) into one zip file, and submit it to the correct assignment dropbox. Consult “How to Save and Submit” on Learn for more information on how to create a ZIP.

It is your responsibility to submit to the correct dropbox with the correct files before the deadline. Otherwise, you will receive a mark of 0.