Assignment 4: Shapes

**Due:** Friday, January 31 at 11:59 PM

You'll write code to create a little game where you select a shape that matches the given shape. There's a timer too, so if you do not select the correct shape and answer, the given shape changes. There's no starter code, but you're free to use any code from this year's labs or lectures.

You can see a video demo of my example solution here: [https://youtu.be/8MuzyOHG-S0](https://youtu.be/8MuzyOHG-S0)

Requirements and Grading

You will create two sketches:

In **A4_basic**, you 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 **A4_enhanced**, you can extend and deviate somewhat from the required functionality to show off your programming skill and design creativity. Do this second.

[ marks ] Correctness

*Required Functionality*

These are roughly ordered by difficulty. The final bullets are more challenging.

- Use a 400 by 400 canvas and light yellow background.
• [1] A 50 pixel diameter circle (i.e. selected shape) with a dark stroke, stays centred on the mouse cursor (i.e. it moves with the mouse).

• [1] A rectangle (i.e. answer box) with a dark stroke is drawn at the bottom of the canvas. The height should be 80 pixels, and it is as wide as the canvas.

• [1] A rectangle (i.e. container) with a dark stroke is drawn at the top right corner of the canvas with a 20 pixel margin from the top and right side of the canvas. The width and height are both 80 pixels.

• [1] The progress bar with a dark stroke is drawn at the top left with a 20 pixel margin from the top and left side of the canvas. It is 180 pixels high and 20 pixels wide.

• [1] In the progress bar, the white bar animates (shrinks) up by 1 pixel. When the time is up, the progress bar is simply an all-grey rectangle. This animation repeats. That is, once the white bar reaches the top, it starts to animate up again from the bottom.

  HINT: Use a variable to animate the change in progress bar. Check when white bar reaches the top.

• [1] Each time the program runs, one of the three shapes--circle, rectangle, or triangle--appear inside the rectangle (i.e., container) on the top right corner. The shape must be arbitrarily decided when the program runs. (This is a shape given to the user.)

  HINT: Use a variable to store one of three integers, each of which represents each shape; randomly choose one integer.

• [1] Every time the progress bar repeats its cycle, one of the three shapes chosen at random is shown inside the container rectangle.

• [1] For all three shapes drawn inside the container rectangle, the center of the shapes need to be at the center of the container rectangle as well.

• [1] When a key is pressed, the shape that follows the mouse (i.e., selected shape) should change its shape to one of the three shapes.

• [1] All three shapes that follow the mouse have dark stroke. The mouse is at the center of these shapes.

• [1] When the mouse is inside the answer box rectangle (i.e., the rectangle at the bottom), its color changes to green if the selected shape matches the given shape and to red if incorrect.

• [1] When the mouse is pressed inside the answer box rectangle, if the selected shape matches the given shape, it restarts the timer (progress bar) and randomly chooses a new shape inside the container rectangle. If it does not match, it does not do anything.
● 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.
- Unnecessarily 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 **A4_enhanced**.

These basic requirements must still be followed:

- The canvas size must be the same size as A4_basic.
- It must maintain the general idea of changing your shape to match the given shape.
- You can change the shape to be something else, change the way keys work, change colours, change the background, change the speed of timer, change how things move, ...

Some ideas:
• Allow users to set the speed of timer.
• Add a score counter to display the number of shapes users matched.
• Add more shapes or shapes with different colors.
• Allow users to transition to a more difficult level.
• Draw really detailed characters or backgrounds.

Think about game play. Full enhancement marks will only be given if the game is still playable and not impossible.

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 85% or more.

Restrictions

In general, you may not use any functions, libraries, or statements not covered in lecture or labs unless not 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 for or while loops
• 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 (A4_basic and A4_enhanced) into one zip file, and submit it 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.