

The Painful Paint Program

Dr. Bradford A. Towle Jr.
Graphics

1. Objectives

- A. Outcomes – Using JavaScript and OOC, Managing Complex Scenes, Interactive Graphical Applications
 - a. To understand Mouse Inputs
 - b. To dynamically create objects at runtime.
 - c. To demonstrate what must happen to change vertices during run time.

2. Equipment

- A. A computer that runs Notepad ++ and a modern web browser
- B. Several pots of coffee.

3. Lab Preparation

- A. Understand Mouse Input and examples we went over in class

4. Lab Instruction

- A. Phase 1 – Drawing the shapes
 - a. In HTML, there will be five different options to choose from:
 - i. Box
 - ii. Line
 - iii. Circle
 - iv. Triangle
 - v. N-Polygon
 - b. In HTML, there will also be a way to change the current color.
 - i. This can be done through color selector or buttons.
 - ii. There must be at least 5 different colors.
 - iii. You do not have to worry about changing colors while you are currently drawing a shape.
 - c. Inside HTML, you will also allow the user to select between outline or fill.
 - d. Inside the main class you will have an empty list of shapes which you will populate.
 - e. When I click on the canvas and I am not currently drawing a shape, then I will create a new shape (Of whatever type I currently have selected) with the current color I have selected.
 - i. The first point will be added to the vertices.
 - f. If I am drawing a square, the first point will be one corner and the 2nd click will be the opposite corner, (you will have to calculate the other two points and add them when you do the 2nd click).
 - g. If you are drawing a circle, the first click will be the center and the 2nd click will set the radius. You will have to calculate at least 10 points to get a good circle.
 - h. If you are drawing a line, you can either end the line on the 2nd click, (thus finishing the shape) OR you can allow the user to click as much as they want until they press the right mouse button.
 - i. If you are drawing the triangle you will need the user to click two more points before the shape is finished.

- j. If you are drawing an N-Polygon the user can continuously add points until they press the right mouse button.
 - k. The shapes must be rendered either as an outline or filled, based on whatever was selected when the shape was created.
- B. Phase 2 – The cursor
- a. In this phase, we are going to add a call back that will be called whenever the mouse moves
 - b. The call back will go through a global function and then to a function in main.
 - c. If the last shape in the list is being drawn, the current location of the mouse will be *temporarily* added to the shape as if it was the 2nd click.
 - d. The vertices will be re-buffered (for this shape only)
 - e. The screen will be cleared
 - f. All the shapes will be rendered, the last shape will be rendered as a LINE_LOOP.
 - g. All of the vertices that were added to the shape at step c will be removed. (Hint: JS lists have pop()).
 - h. This will give you the effect of having a cursor as you draw your shape.

5. Lab Rubric

	Perfect	Logical or visual errors but no syntax errors	Syntax errors or incomplete
Can select color	5	2	0
Can select shape	5	2	0
Can select fill or outline	5	2	0
Can draw square	5	3	0
Can draw triangle	5	3	0
Can draw line	5	3	0
Can draw circle	5	3	0
Can draw n-polygon	5	3	0
Shape cursors are accurate	20	10	0
Total	/60		

6. Lab Report Requirements

There is no lab report due, but there is a peer review.