FINAL PROJECT: HTML 5 CANVAS

Group:

1351004 – Huỳnh Gia Bảo

1351008 – Cao Khắc Lê Duy

Overview

Talking about Canvas in HTML 5, it is a HTML tag similar to the `<div>`, `<a>` or `<img>` tag. The difference is that its content is rendered with `Javascript`. To control and integrate with canvas in our project, we need to put canvas somewhere in the html document first, then in the `Javascript` files connected to this document, we utilize the `HTML5 Canvas API` to draw visualizations.

Furthermore, in canvas, there exist a confusion between canvas context and canvas element. Canvas element is the actual `DOM` node that's embedded in the HTML page. The canvas context is an object having properties, and methods that we can use to express graphic object inside the canvas element. The context can be 2d or webgl (3d), here we utilize the 2d context of canvas.

There is only one context for each canvas, it is a singleton pattern, everytime we access to the context via "`getContext()`" method, we always get the same context object.

About the project, it is a small demo that inherit the paint program incompletely to check the graphic ability of `HTML 5 Canvas`, so we focus on the concepts of drawing basic objects like `Line`, `Rectangle`, `Oval`, `Pencil`, `Eraser`, `Color`, etc...

Basic concepts

Mouse Detection

To detect the click event, we utilize the power of `Javascript` with their built-in event defined with strings: `‘mousemove’, ‘mousedown’, ‘mouseup’` and `‘mouseout’`. They are events that the mouse is moving, being clicked down, being clicked up, and moving outside the range of the canvas, respectively.

```
canvas.addEventListener("mouseout", function (e) {

    findxy('out', e)

  }, false);
```

The pieace of code above show us an example about attach a listener to the event `‘mouseout’`, when the mouse is moving out of the range of the canvas, the function `findxy(‘out’,e)` will be invoked.

Line

First, we need to know about drawing a line in canvas, this is the most basic concept of graphics because from this object, we can render the others which are more complicated. The basic functions and properties are:

`beginPath(), closePath():` these two methods of context are used not only in making lines but also in making the other shape object in canvas. `beginPath()` signal that a shape has been started to be drawn. `closePath()` signal that the shape has been completed drawing.

`moveTo``(`**`x1, y1`**`)`: the context move focus to a point, and this point become the starting point for the drawing action

`lineTo``(`**`x2, y2`**`):` the context identify the final point of the line with coordinates `x2, y2`, then create the line from `(x1, y1)` to `(x2, y2)` with **`stroke().`**

`lineWidth:` declare the thickness of the line or the border of a shape

`lineCap:` declare the type of the head of the line. It can be round, square or butt…

Curve

This is also a basic components of a painting, with HTML5 Canvas, we use quardarticCurveTo() instead lineTo() to draw a curve, the syntax is as follow:

`quadraticCurveTo(x2, y2, x3, y3)`: (x2, y2) is the middle point of the curve, or the peak of the curve; (x3, y3) is the last point of the curve.

Oval

There are two ways to draw an oval shape. We can use **`arc()`** or **`ellipse().`** The key difference here is that with `arc()`, we can draw incomplete circle but not an ellipse and with `ellipse()`, we can draw a full oval but no incomplete oval.

`arc(x, y, r, startAngle, endAngle, counterClockWise):` to draw with arc, we need to pass into the function these parameteres: x and y are the coordinates of the central point, `startAngle` is the angle where the arc is started to be drawn, `endAngle` is the angle where the arc is stopped being drawn. `counterClockWise` is for determining the direction of drawing.

`ellipse(x, y, r1, r2, rotation, startAngle, endAngle, anticlockwise):` drawing with ellipse will require us to inform the function the data of central point, first radius, second radius, rotation here is the angle of the rotation according to the clock-wise direction. `startAngle` and `endAngle` have the same meaning as in arc() function. `‘anticlockwise’` define the direction of drawing the ellipse: true means the counter-clockwise direction and vice versa.

Rectangle

The built-in function for drawing a rectangle is **`rect(leftTopX, leftTopY,width, height)`** where `leftTopX` and `leftTopY` is the coordinates of the top left point of the rectangle; width and height are the size of the rectangle.

Besides, we can draw 4 lines to form a rectangle with the line-drawing technique above.

Color

`Line color:` after drawing any shapes on the canvas, we can choose the color for the border or line by setting property `‘strokeStyle’` of the context.

`Color fill:` right after completing drawing a shape, the property `fillStyle` would be used to choose the background color for that shape then the shape will be filled with that color when the `fill()` function is called.

Project insights

To perform the feature of each object in paint application, instead of getting stuck with the structure of OOP characteristic, HTML5 Canvas have JavaScript as a powerful tool for doing back-end works and HTML5 and CSS for the UI. This separation makes the application clear and efficient.

Back-end components

To build `Paint Application`, first we build the structure of the “painting tools”. In `Javascript`, we can store mutable objects in just one array so, it is really convenient when we don’t need to make it more complicated with inheritance. Each class of object we set up the methods called “draw” which will called for drawing the instance of that class.

The Flow: when mouse clicking event is invoked, the coordinates will be identified then the “draw” function of current chosen tool will be invoked.

The menu bar will work like this: a tool or shape is chosen then a holder will hold the id of the chosen tool. This is also the working principle of choosing line color. Next time there is something drawn on the canvas, the appropriate behavior will be determined based on these holders. We can call it State Pattern.

3-layer technique:

- First layer is temporary layer which is used for draft, after users change their tool or shape (polygon) or after mouse is up back from down (other shapes).
- Second layer is for holding the painting.
- Third layer is for the grid, it is useful for accurate drawing.

Custom Shapes:

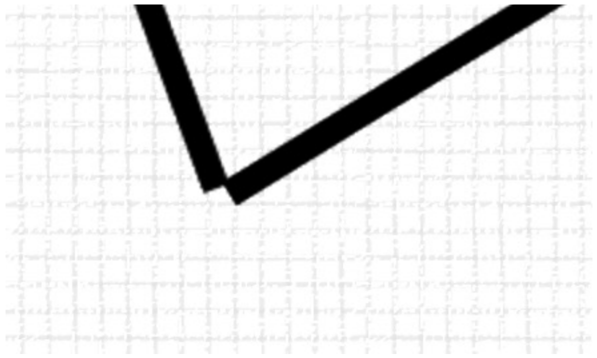
- Star and Equilateral Triangle: the key for drawing these shapes is the bounding circle, we separate the circle into multiple equal segment to put pixel at the right point.
- Pentagon: bounded by rectangle and put the pixel on this rectangle with a specific ratio.
- Arrow: put pixel at static key points, the longer the distance between the initial point and final point is, the bigger the arrow is. It means the arrow is scaled according to the distance from the initial point to the final point of the mouse-down motion.
- Polygon:
    - First Mouse-down action: polygon object is initialized, first line is added to the array of lines of the polygon (initial point = final point)
    - Mouse-down moving, the object update the last point of last line everytime position of mouse is changed. The temporary canvas is updated right after that.
    - Mouse-up action: save the drawing line, ,add new line to the array, this line has only the initial point, the last point will be updated later.
    - Click on other tools or shape : mark the current polygon as completed and finish the last line.
- Chatbox: combine multiple curves to create this shape, defining some fixed ratio then carry out the drawing.

Tools:

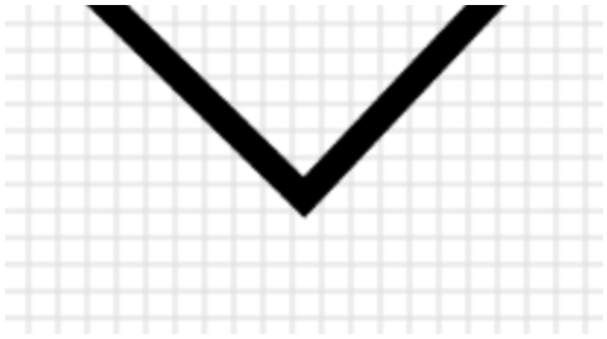
- Pencil: just put point on the temporary canvas first then when the mouse-up action is invoked, save the points to the below canvas and clear the temporary canvas.
- Eraser: we use the property `ctx.globalCompositeOperation` of the canvas context to erase the drawn pixels.
- Color filler: we use the flood fill algorithm to fill the shapes segment, it is slow.
- Color picker: we use the method `getImageData` to get the color of a the pixel the user point to.

Closed Shapes problem:

- When combine lines to make a shape, there would be a gap between two adjacent lines, as can be seen in the figure below:

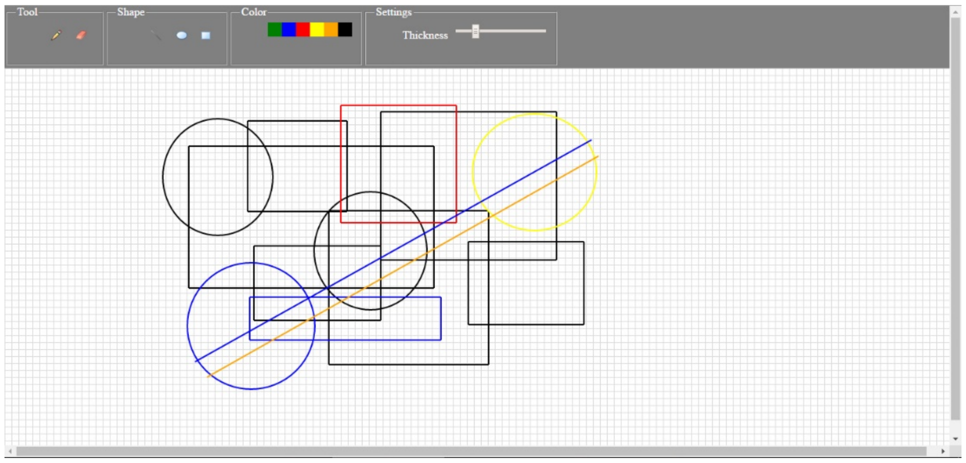


- Fortunately, HTML5 Canvas support us this one, instead of using moveTo everytime starting a new line, for a shape, we use moveTo only one time and then use lineTo until finishing the second last point, because the shape will automatically be closed when the function closePath() is invoked before the function stroke().



Front-end components

There are 4 modules: Toolbox, White board, Color boxes, and Setting area.



Thanks to CSS and HTML5, the frame of the paint application are well-formed instead of writing bunch of code to align the view.

Further information

HTML5 Canvas is a powerful tool for Web 2d Graphic, its feature far exceed what we mention above. The following is additional feature of HTML5 Canvas:

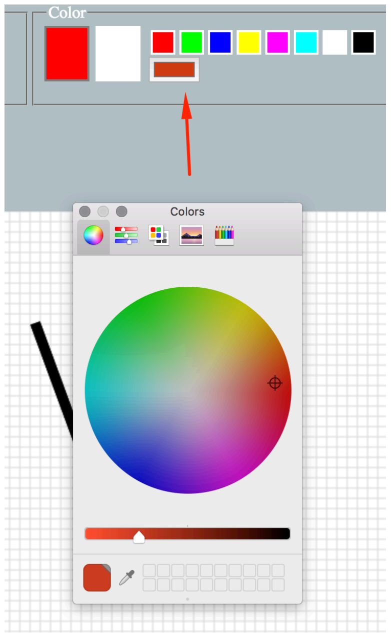
Transformations: it includes all basic transformation such as translation, scaling, rotation, shear, mirror, e.g. It also support custom transformation which works with matrix formulation.

Shadows: create shadow effect for the object

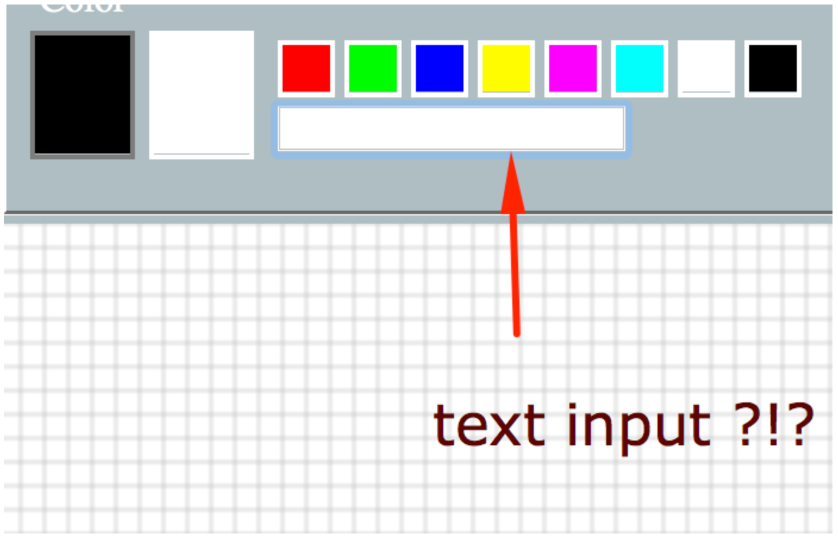
Animation: it also supports us to create animation, after some configuration, we can use a recursive function that changes the form of the shape to make it animated.

Drawbacks with multiple web browser: because html5 canvas works on web platform so it is a little bit difficult to adapt all kinds of web browser today. For example: the color input will not work on Safari and Internet Explorer:

Color input in Chrome



Color input in Safari



Conclusion

HTML5 is a powerful library for web graphics, it includes as many features as the other library such as GLUT. It is somehow more effective to the programmer as it utilize the advantages of JavaScript and HTML5/CSS to handle the control and the UI.