Sohee Cho
Week 7 Part 2

1. Who *first introduced* canvas and in what year? And what was it *used for*?

    1. It was first introduced by Apple for use in their Mac OS X WebKit component in the year 2004.

2. What is canvas 2d and what does 2d *stand for*?

    1. The canvas tag/element is an HTML5 element which can be used to draw graphics using JavaScript. This element is only a container for graphics. We need to use JavaScript to actually draw the graphics.

    2. 2d stands for two-dimensional graphics

3. How do you *define* the canvas element?

    1. The canvas element is defined by "<canvas></canvas>", which is added to the index.html file

4. The canvas element only *supports* the use of 2 other attributes *aside from* the id (or class) attributes. What *are they*?

    1. It also supports the width and height attributes, aside from the id or class.

5. What is the *default* canvas size (width x height)?

    1. The default canvas size is 300w x 150h. This is only if "ctx.fillRect(0,0,600,500) is set in main.js

6. If *no* width or height attribute is *added* to the canvas element in index.html, and *no* background, width, or height properties *within* the canvas element/id in the external stylesheet, or width or height in the JavaScript file, will *anything* render to the page?

    1. No, nothing will render to the page.

7. If the width and height are *set* in ctx.fillRect() only, and the width and height were set to *greater than* 300x x 150h, like 600w x 400h, for example, what width and height of canvas would *render* to the page?

    1. It would only render 300w x 150h. If ctx.fillRect( ) is set to anything below 300w x 150h, up to or equal to that size, that's the size that will render to the page. Above that size, it will only render 300w x 150h.

8. To *get access* to the actual drawing interface in canvas, what do we *first* need to create? And what is that *exactly*?

    1. In order to get access to the actual drawing interface in canvas, we need to create a **context**, which is an object whose methods provide the drawing interface.

9. What are the *2* main drawing styles? What are *each for* and what are they *called*?

1. The 2 main drawing styles are “2d” and “webgl”.

2. “2d” is two-dimensional graphics and “webgl” is three-dimensional graphics through the OpenGL interface

10. How can we *access* our drawing context once we have retrieved our canvas element by id? Which *built-in* HTML5 canvas method does that *for us* and what is it *set on*? *Give me* the variable name and the value it is *assigned*.

    1. We can access our drawing by using the HTML5 canvas built-in “.getContext( )” method.

    2. The variable name is canvas and it is given a value of “document.getElementById(“canvas”).

    3. Code example:

```
const canvas =
document.getElementById(“canvas”);
let ctx = canvas.getContext(“2d”);
if (canvas.getContext) {
        ctx = canvas.getContext(“2d”);
} else {
        const para =
document.querySelector(“.unsupported”);
        para.textContent = `Your browser does not
support HTML5 Canvas.`”;
```

```
}
```

11. What are the *two* primitive shapes canvas *supports*?

    1. The two primitive shapes supported by canvas are rectangles and paths. Paths are lists of points connected by lines, just like in Adobe Illustrator.

12. How are *all other shapes* (other than the primitive rectangle or path) created in canvas? And what kind of *functions* help us create complex shapes?

    1. All other shapes are created by combining one or more paths. There are also built-in canvas path drawing functions which help us create complex shapes.

13. What are the *3* built-in functions in canvas that *draw* rectangles in canvas? *Tell me* the function names and *give me* the names of the parameters *passed in* to them. Also, *describe* what each parameter *stands for*.

    1. The 3 built-in functions in canvas that draw rectangles are:

        1. fillRect ( )

            1. The parameters passed into this function are “x, y, width, height” which are the x-axis coordinate of the upper-left corner of the rectangle, the y-axis coordinate of the upper-left corner of the rectangle, the width of the rectangle (positive

values to the right and negative values to the left), the height of the rectangle (positive values are down, negative values are up)

2. strokeRect ( )

    1. The parameters in this function are “x, y, width, height”

    2. The x is the x-axis coordinate of the rectangle’s starting point, the y is the y-axis coordinate of the rectangles starting point. The width is the rectangle's width, where positive values are to the right and negative is to the left. The height is the rectangle’s height, where positive values are down and negative are up.

3. clearRect ( )

    1. The parameters in this function are also “x, y, width, height”

    2. The x is the x-axis coordinate of the rectangle’s starting point, the y is the y-axis coordinate of the rectangles starting point. The width is the rectangle's width, where positive values are to the right and negative is to the left. The height is the rectangle’s height, where positive values are down and negative are up.

14. *Describe* what a path is in canvas.

    1. A path in canvas is a list of points connected by segments of lines that can be different shapes. They can be curved or not, and of different widths and colors. A path/subpath can be closed.

15. What are the *extra steps* to drawing paths in canvas?

    1. First, draw a path.

    2. Use drawing commands to draw into the path.

    3. Then, once a path has been created, use stroke ( ) or fill ( ) to the path to render the drawing