Web Animation

Week2

Plan of Action

- WEEK1
  - Intro to animation and canvas: Setup our constructors and other game functions
- WEEK2
  - Set up our Bird character and gravity
- WEEK3
  - Obstacles and Collision
- WEEK4
  - Score and End Screen
- WEEK5
  - Wrap up

Recap on Animations

Time
Layers
Frames (FPS)

Where did we use this techniques in last weeks code?

Time and FPS

setInterval calls a function at a set interval or moment in time (milli seconds)

Look at the below code

- updateGameArea is the function being called
- Every 20 milliseconds
- 1000/20 = 50. Hence this function is called 50 times in 1 sec. **50FPS**
- This simulates our frames.
- The updateGameArea function is in charge of clearing away old images and redrawing in updated position

```
this.interval = setInterval(updateGameArea, 20);
```

Frames

We also have a variable called frameNo

This increments every time our updateGameArea function is called

Therefore it counts each frame and lets us know the current frame we are on

We will use this idea more next week

```
//setting up frame no
this.frameNo = 0;
```

Layers

We did not use any method or function to place the images on a layer

Instead they layer is inferred by the order in which we draw our objects

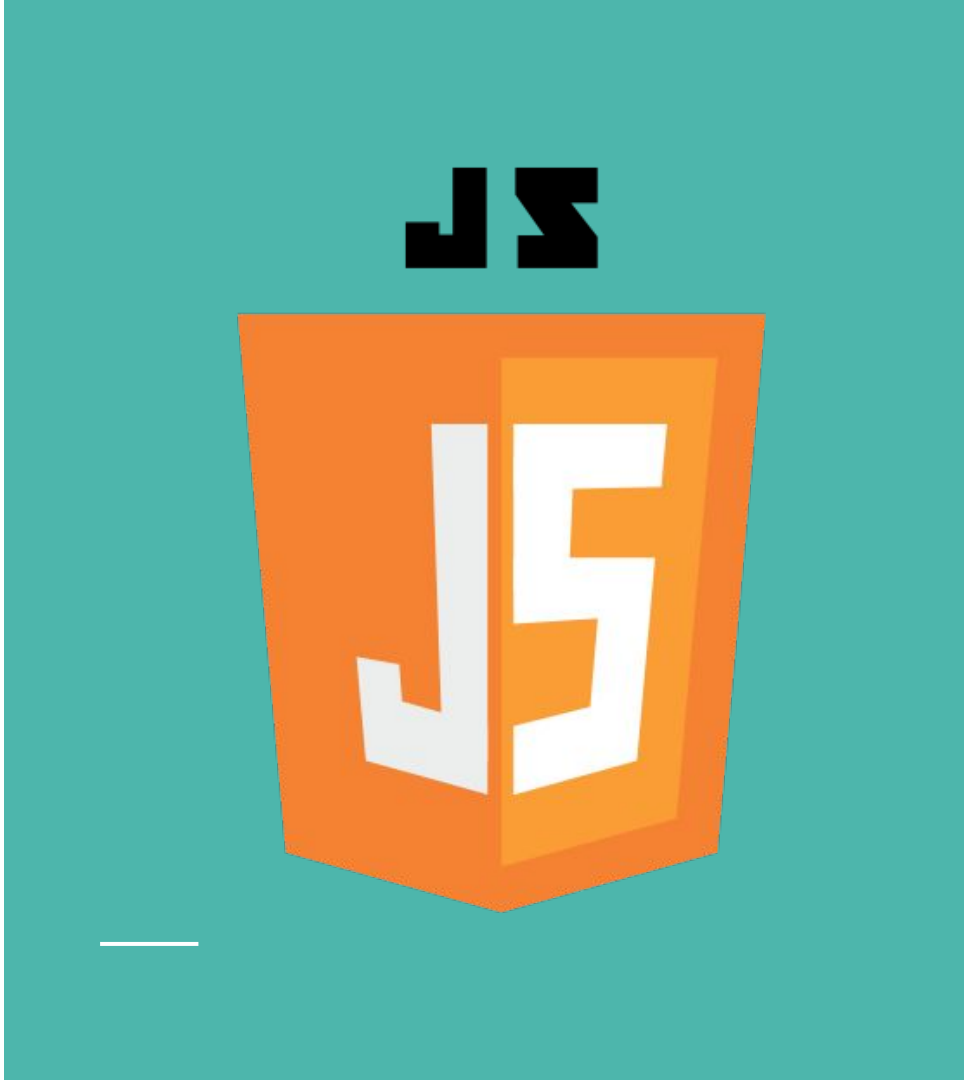
Remember these lines of code are executed one after another

What would happen if we wrote myGameArea.clear(); last?

```
//clear canvas (so we dont have thousands of images)
myGameArea.clear();
// increment our Frame count
myGameArea.frameNo += 1;
// update our bg image
myBackGround.update();
```

Recap on OOP

Object
Constructor



Objects

- name : value Pairs
- Separated by ,
- Can also have methods
- No comma on last line

```
var person = {
    firstName:"John",
    lastName:"Doe",
    age:50,
    eyeColor:"blue",
    sayHello: function(){
        alert("Hello");
    }
};
```

Constructor

- Making objects as on previous screen limits what we can do
- So we can use a constructor to declare an object types and make multiple objects of that type

```
function person(first, last, age, eye) {
    this.firstName = first;
    this.lastName = last;
    this.age = age;
    this.eyeColor = eye;
}
var myFather = new person("John", "Doe", 50, "blue");
var myMother = new person("Sally", "Rally", 48, "green");
```

Where did we use Objects and Constructors in last weeks code?

myGameArea Object

- 1 property and 2 methods
- What do they do and where can they be used?

```
var myGameArea = {
    canvas : document.createElement("canvas"),
    start : function() {
        this.canvas.width = 480;
        this.canvas.height = 270;
        this.context = this.canvas.getContext("2d");
        document.body.insertBefore(this.canvas, document.body.childNodes[0]);
        this.frameNo = 0;
        this.interval = setInterval(updateGameArea, 20);
        },

    clear : function() {
        this.context.clearRect(0, 0, this.canvas.width, this.canvas.height);
    }

}
```

Component Constructor

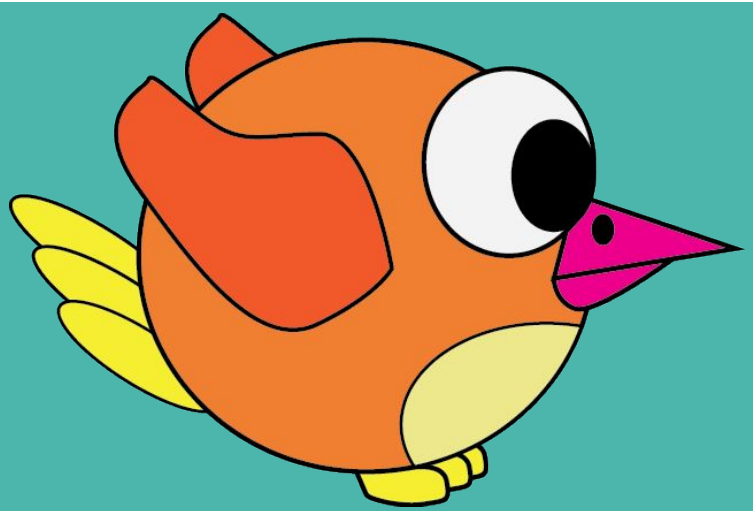
What objects are being constructed?

What are the methods?

What do they do?

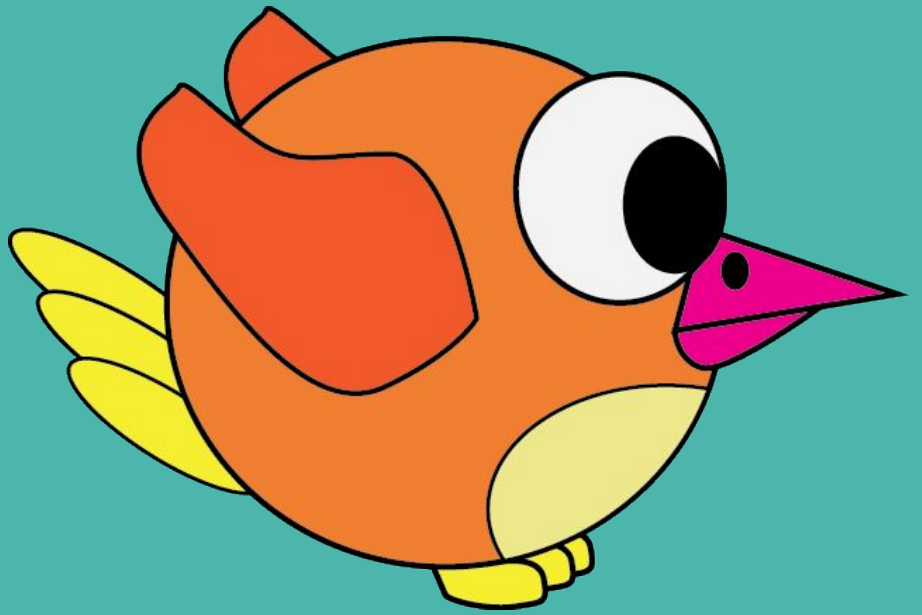
```
function component(width, height, look, x, y, type) {
    this.type = type;
    this.width = width;
    this.height = height;
    this.x = x;
    this.y = y;

      if (type == "image") {
        this.image = new Image();
        this.image.src = look;
      }

    this.update = function() {
        ctx = myGameArea.context;
        if (type == "image") {
          ctx.drawImage(this.image,
            this.x,
            this.y,
            this.width, this.height);
        }
        else {
            ctx.fillStyle = look;
            ctx.fillRect(this.x, this.y, this.width, this.height);
        }
    }
}
```

This Week
We will work on the bird component



What Methods does this component need?

Think back to the Flappy Bird Game



Today's Lab

- Component Methods
  - New position
  - Hit bottom
  - flip
- Function
  - accelerate
- Input
  - Update html with button
- Make new component
  - Bird