LAB REPORT

PONG

Video Game Design
Explore Engineering

Weiji “Victor” Li
Mentor: Kai

TABLE OF CONTENT

INTRODUCTION

1.1 PONG

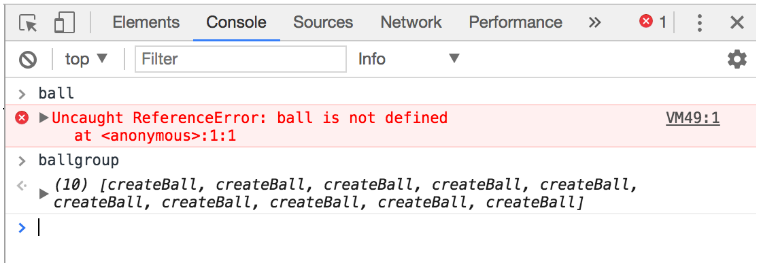
“Pong is a two-dimensional sports game that simulates table tennis. The player controls an in-game paddle by moving it vertically across the left side of the screen, and can compete against either a computer-controlled opponent [*not implemented yet*] or another player controlling a second paddle on the opposing side. Players use the paddles to hit a ball back and forth. The aim is for each player to reach eleven [or any threshold] points before the opponent; points are earned when one fails to return the ball to the other” [1].

My goal is to build this game on the personal computer and add more features.

1.2 programming language and software

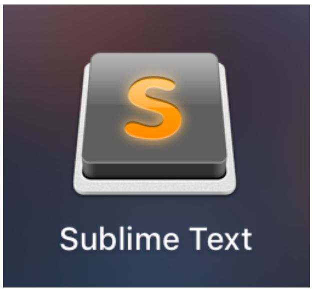
My main programming language is JavaScript and I have also used HTML and CSS to create the website. The JavaScript can achieve many programming goals that can help me to build, draw and change the balls. HTML is a kind of frame that helps me to put this program inside a web page; CSS beautifies my user interface.

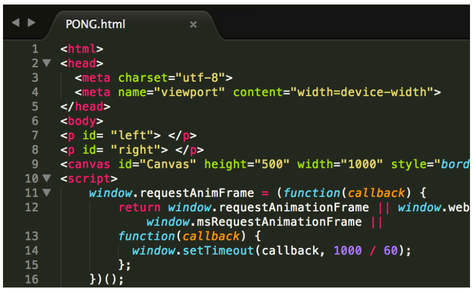
I use sublime to code since it can highlight keywords with different colors. Also I use the Chrome browser’s console to help me debug and assists me to test my code in the website. These two pictures below are their icons and the working frames.



```
> ball
✖ ▶Uncaught ReferenceError: ball is not defined                VM49:1
      at <anonymous>:1:1
> ballgroup
<· ▶ (10) [createBall, createBall, createBall, createBall, createBall,
   createBall, createBall, createBall, createBall, createBall]
>
```

[2]





```
<html>
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width">
</head>
<body>
<p id= "left"> </p>
<p id= "right"> </p>
<canvas id="Canvas" height="500" width="1000" style="bord
<script>
    window.requestAnimFrame = (function(callback) {
        return window.requestAnimationFrame || window.web
            window.msRequestAnimationFrame ||
        function(callback) {
          window.setTimeout(callback, 1000 / 60);
        };
    })();
```

[3]

DESCRIPTION

2.1 FILES

- PONG.html (main)
- Acc.png, small.png, big.png, dec.png

2.2.1 INPUT

- The mode of the games (classic or extreme)
- The number of the balls

2.2.2 OUTPUT

- The balls
- Some small items
- The amazing visual effects
- The score of each side

2.3 VARIABLES

1. The object of ball
    a. x,y *//Describes position of the ball*
    b. vx,vy *//Describes velocities of the ball*
    c. radius, mass, color *//Describes the properties of the ball*
    d. last collided balls *//Used for collide function, to avoid joint debugs*
2. The object of paddles
    a. x,y *//Describes left top position of the paddle*
    b. width, height *//Describes the whole width and height*
    c. up, down,left,right *//Used for moving the paddles*
3. The object of item
    a. x,y *//Describes position of the item*
    b. radius *//Describes radius of the item*
    c. isExist *//Used for generating new ones*

METHODOLOGY

3.1 Algorithm

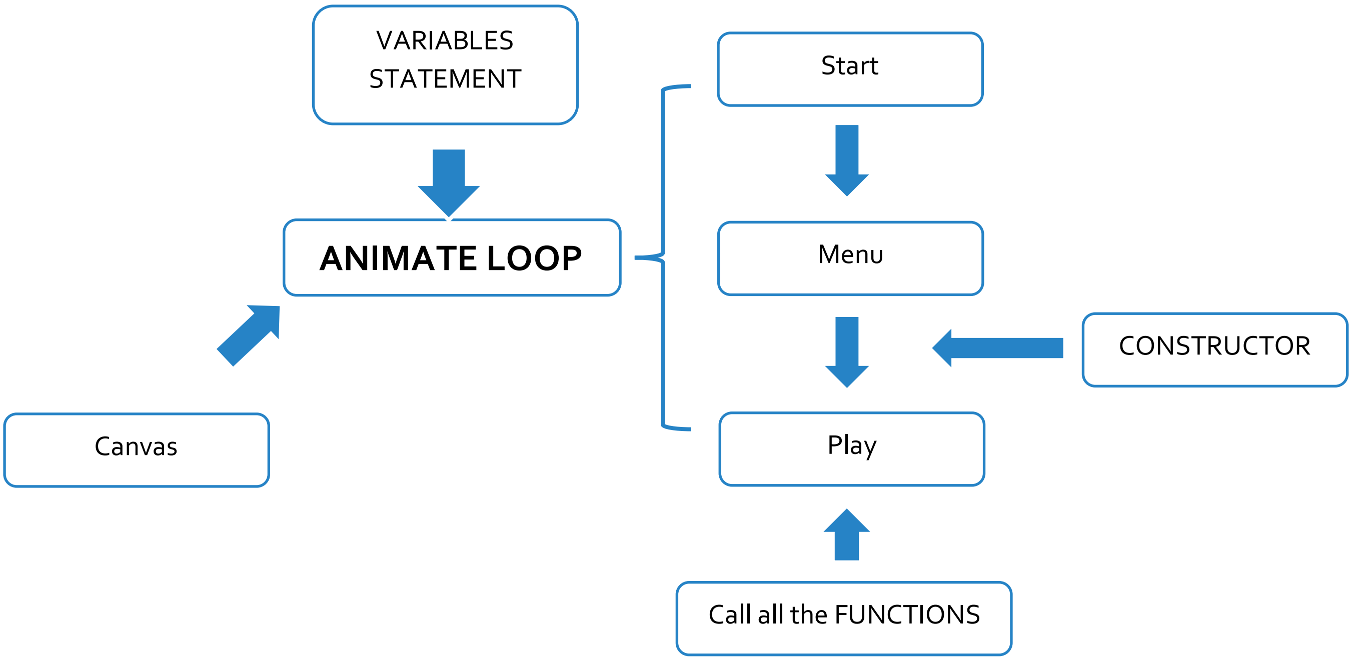
1. main function (animate): used for refreshing the whole screen

   including calling all the functions, drawing canvas and decide whether some events will happen

2. constructors
   a. balls
   b. paddles
   c. items

3. function
   a. the moving of the balls
   b. the collision
   c. the explode
   d. the start
   e. the keyboard and the mouse detetion

3.2 STRUCTURE

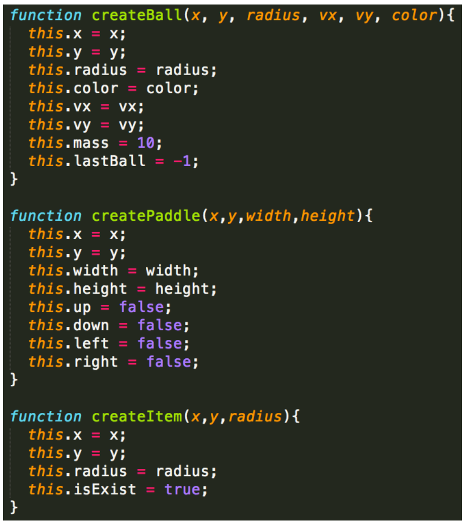


SPECIFIC FUNCTIONS

4.1 JavaScript

1. Constructors

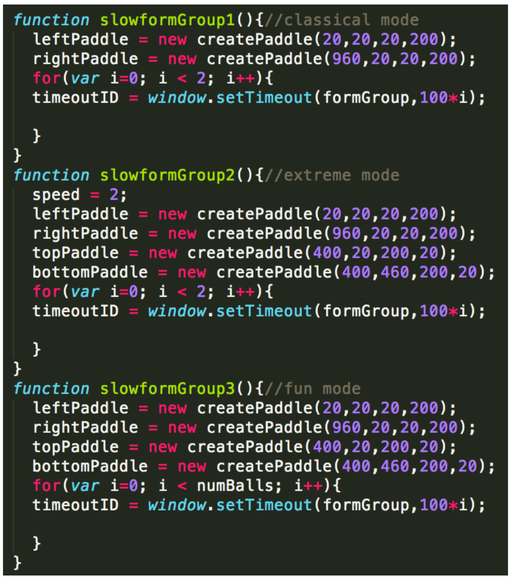
I use these three constructors to form these three different objects easily since I can just call them later very easily. For example, I can generate different numbers of balls easily by just calling the constructors of balls for a number of times.



```
function createBall(x, y, radius, vx, vy, color){
  this.x = x;
  this.y = y;
  this.radius = radius;
  this.color = color;
  this.vx = vx;
  this.vy = vy;
  this.mass = 10;
  this.lastBall = -1;
}

function createPaddle(x,y,width,height){
  this.x = x;
  this.y = y;
  this.width = width;
  this.height = height;
  this.up = false;
  this.down = false;
  this.left = false;
  this.right = false;
}

function createItem(x,y,radius){
  this.x = x;
  this.y = y;
  this.radius = radius;
  this.isExist = true;
}
```

2. Functions of slowFormGroup

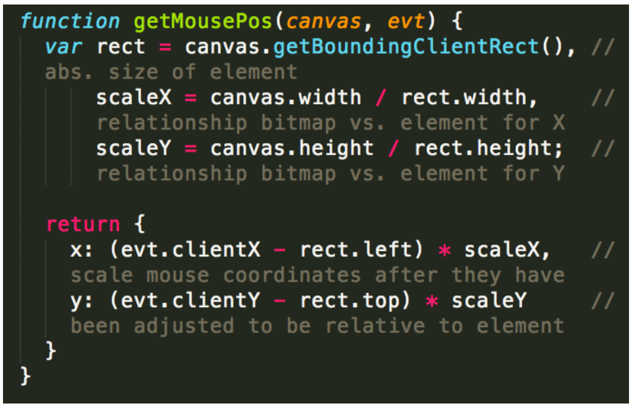
These functions are used for generate the arrays of balls and different functions correspond to different modes since they have different numbers and setting. I use a function called setTimeOut to generate the balls one by one. At first the time is so small since it is in millisecond and also it is calculated from the start so I set them to 100*i at last to fix the bugs in this function.



```
function slowformGroup1(){//classical mode
  leftPaddle = new createPaddle(20,20,20,200);
  rightPaddle = new createPaddle(960,20,20,200);
  for(var i=0; i < 2; i++){
  timeoutID = window.setTimeout(formGroup,100*i);

  }
}
function slowformGroup2(){//extreme mode
  speed = 2;
  leftPaddle = new createPaddle(20,20,20,200);
  rightPaddle = new createPaddle(960,20,20,200);
  topPaddle = new createPaddle(400,20,200,20);
  bottomPaddle = new createPaddle(400,460,200,20);
  for(var i=0; i < 2; i++){
  timeoutID = window.setTimeout(formGroup,100*i);

  }
}
function slowformGroup3(){//fun mode
  leftPaddle = new createPaddle(20,20,20,200);
  rightPaddle = new createPaddle(960,20,20,200);
  topPaddle = new createPaddle(400,20,200,20);
  bottomPaddle = new createPaddle(400,460,200,20);
  for(var i=0; i < numBalls; i++){
  timeoutID = window.setTimeout(formGroup,100*i);

  }
}
```

3. Function of getMousePos

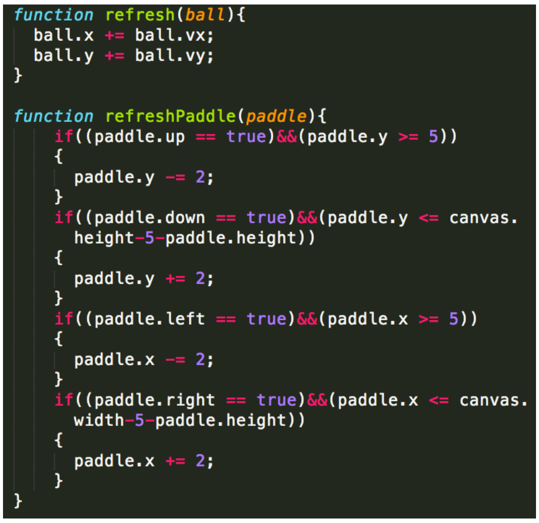
This function is used for get the position of the mouse. Since my canvas for the main game doesn't start at the left top of the whole website so I met the problems of the dislocation of the mouse at first. Finally, I changed this function to get the relative position of the mouse to the canvas.



```
function getMousePos(canvas, evt) {
  var rect = canvas.getBoundingClientRect(), //
  abs. size of element
      scaleX = canvas.width / rect.width,    //
      relationship bitmap vs. element for X
      scaleY = canvas.height / rect.height;  //
      relationship bitmap vs. element for Y

  return {
    x: (evt.clientX - rect.left) * scaleX,   //
    scale mouse coordinates after they have
    y: (evt.clientY - rect.top) * scaleY     //
    been adjusted to be relative to element
  }
}
```

4. Functions of refresh

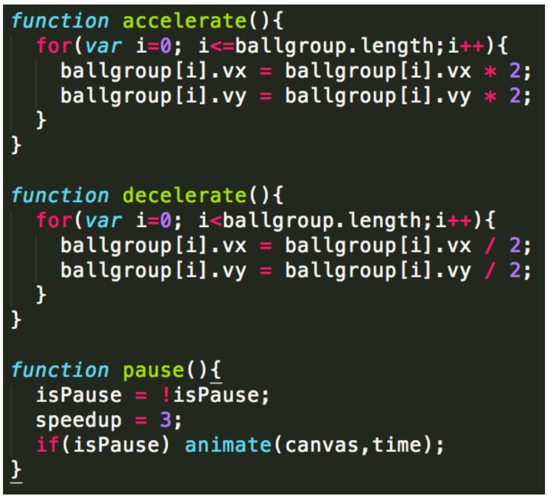
These two functions are used for the refresh of the paddle and the ball since I should refresh their positions every frame so that they look like they are moving. I set the time interval at first but as the interval is recorded in millisecond so the speed cannot be controlled well so at last I remove this additional setting.



```
function refresh(ball){
  ball.x += ball.vx;
  ball.y += ball.vy;
}

function refreshPaddle(paddle){
    if((paddle.up == true)&&(paddle.y >= 5))
    {
      paddle.y -= 2;
    }
    if((paddle.down == true)&&(paddle.y <= canvas.
      height-5-paddle.height))
    {
      paddle.y += 2;
    }
    if((paddle.left == true)&&(paddle.x >= 5))
    {
      paddle.x -= 2;
    }
    if((paddle.right == true)&&(paddle.x <= canvas.
      width-5-paddle.height))
    {
      paddle.x += 2;
    }
}
```

5. Functions of controlling the whole game

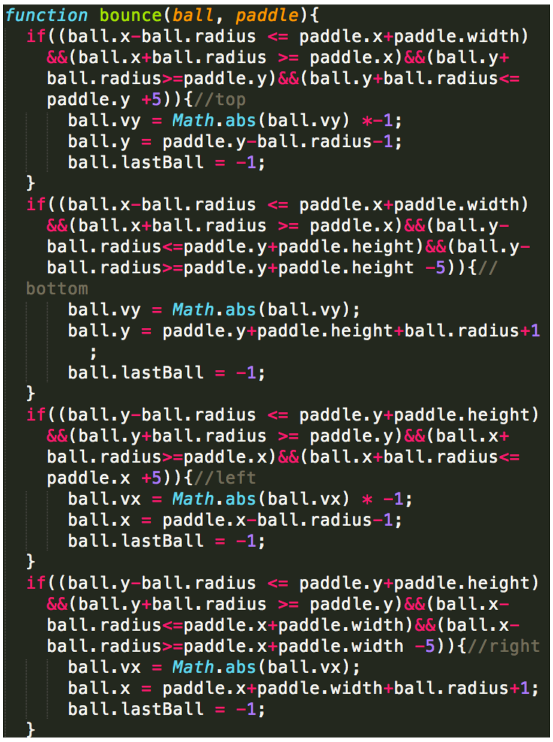
There are three functions in all which can control the whole game: accelerate, decelerate and pause. First two functions are quite simple since I just change the speed of each balls. But the last functions cost me much time because I can't put them inside the animate. As a result, I choose to make a signal called "isPause" and check it every frame.



```
function accelerate(){
  for(var i=0; i<=ballgroup.length;i++){
    ballgroup[i].vx = ballgroup[i].vx * 2;
    ballgroup[i].vy = ballgroup[i].vy * 2;
  }
}

function decelerate(){
  for(var i=0; i<ballgroup.length;i++){
    ballgroup[i].vx = ballgroup[i].vx / 2;
    ballgroup[i].vy = ballgroup[i].vy / 2;
  }
}

function pause(){
  isPause = !isPause;
  speedup = 3;
  if(isPause) animate(canvas,time);
}
```

6. Function of bounce

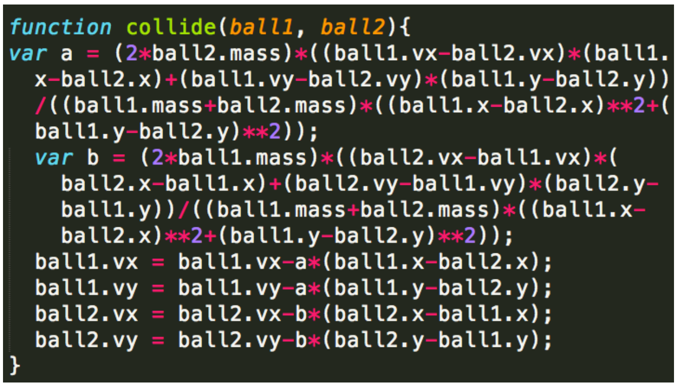
All the paddles should bounce the balls back and this function is used for this. I divide the surface of the paddles into four pieces which correspond to four edges so that I can check the bounce accurately. At first, some really fast balls even entered and went through the paddle so I forced to be out of the paddle after colliding to solve this problem effectively.



```
function bounce(ball, paddle){
  if((ball.x-ball.radius <= paddle.x+paddle.width)
    &&(ball.x+ball.radius >= paddle.x)&&(ball.y+
    ball.radius>=paddle.y)&&(ball.y+ball.radius<=
    paddle.y +5)){//top
      ball.vy = Math.abs(ball.vy) *-1;
      ball.y = paddle.y-ball.radius-1;
      ball.lastBall = -1;
  }
  if((ball.x-ball.radius <= paddle.x+paddle.width)
    &&(ball.x+ball.radius >= paddle.x)&&(ball.y-
    ball.radius<=paddle.y+paddle.height)&&(ball.y-
    ball.radius>=paddle.y+paddle.height -5)){//
  bottom
      ball.vy = Math.abs(ball.vy);
      ball.y = paddle.y+paddle.height+ball.radius+1
        ;
      ball.lastBall = -1;
  }
  if((ball.y-ball.radius <= paddle.y+paddle.height)
    &&(ball.y+ball.radius >= paddle.y)&&(ball.x+
    ball.radius>=paddle.x)&&(ball.x+ball.radius<=
    paddle.x +5)){//left
      ball.vx = Math.abs(ball.vx) * -1;
      ball.x = paddle.x-ball.radius-1;
      ball.lastBall = -1;
  }
  if((ball.y-ball.radius <= paddle.y+paddle.height)
    &&(ball.y+ball.radius >= paddle.y)&&(ball.x-
    ball.radius<=paddle.x+paddle.width)&&(ball.x-
    ball.radius>=paddle.x+paddle.width -5)){//right
      ball.vx = Math.abs(ball.vx);
      ball.x = paddle.x+paddle.width+ball.radius+1;
      ball.lastBall = -1;
  }
```

7. Function of collide

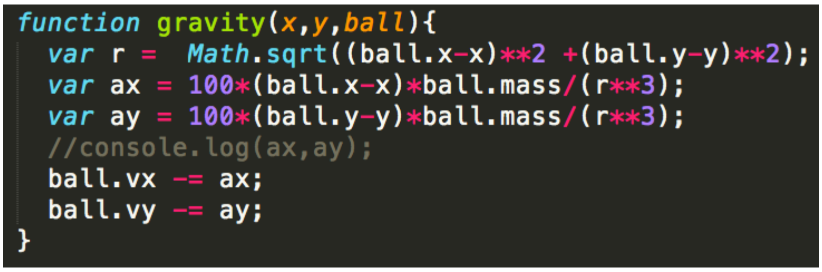
When dealing with the collision between balls, I use a vector formula which is quite difficult to express in JavaScript so I write a quite long code to achieve this. Also there is a big problem that two balls may stick together and keep colliding. Consequently, I use a kind of signal to point them so that they won't collide together before hitting a wall.



```
function collide(ball1, ball2){
var a = (2*ball2.mass)*((ball1.vx-ball2.vx)*(ball1.
  x-ball2.x)+(ball1.vy-ball2.vy)*(ball1.y-ball2.y))
  /((ball1.mass+ball2.mass)*((ball1.x-ball2.x)**2+(
  ball1.y-ball2.y)**2));
  var b = (2*ball1.mass)*((ball2.vx-ball1.vx)*(
    ball2.x-ball1.x)+(ball2.vy-ball1.vy)*(ball2.y-
    ball1.y))/((ball1.mass+ball2.mass)*((ball1.x-
    ball2.x)**2+(ball1.y-ball2.y)**2));
  ball1.vx = ball1.vx-a*(ball1.x-ball2.x);
  ball1.vy = ball1.vy-a*(ball1.y-ball2.y);
  ball2.vx = ball2.vx-b*(ball2.x-ball1.x);
  ball2.vy = ball2.vy-b*(ball2.y-ball1.y);
}
```

8. Function of gravity

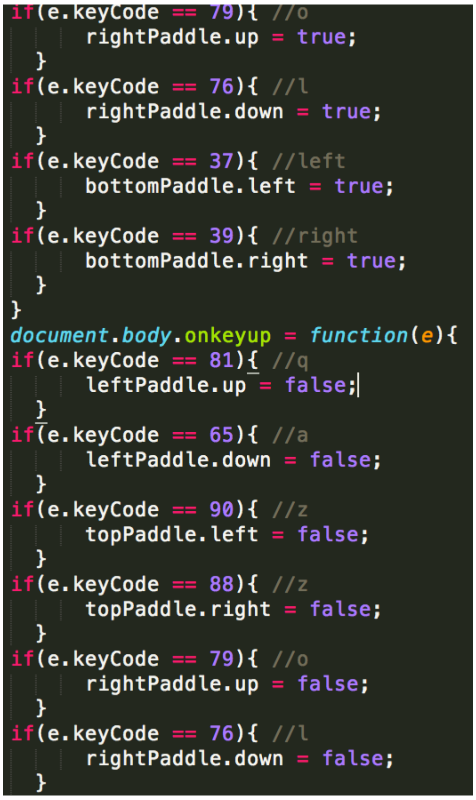
This function is very interesting and it is also the parts in the program that I am proud most with. When you click the mouse and move them, the balls will go towards the mouse as if there is a gravity there. And the movement of the balls follow the rules of circular moving in real world! I use the formula of Universal Gravitation and use the getMousePos I mentioned before to follow the mouse. As a result, the mouse is just like a black hole that can attract all the moving balls.



```
function gravity(x,y,ball){
  var r =  Math.sqrt((ball.x-x)**2 +(ball.y-y)**2);
  var ax = 100*(ball.x-x)*ball.mass/(r**3);
  var ay = 100*(ball.y-y)*ball.mass/(r**3);
  //console.log(ax,ay);
  ball.vx -= ax;
  ball.vy -= ay;
}
```

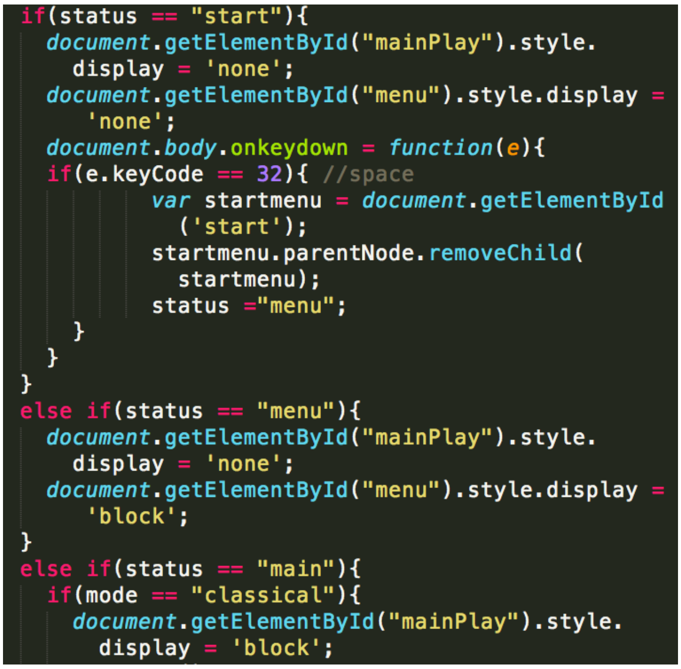
9. Function of keyboard

This function is used for get the press of keyboard to execute some commands. I thought it was quite easy at first since I thought I just need to get the code of each key. However, I met a problem that the movement of the paddle is completely not smooth so I make several signals and the keydown() and keyup() to make the movement consistent. When you press any key, the signal of this equals true, which means the paddle contiunes to do in one direction. And when your fingers move away, this siganl equals false again, stop the movement immediately.



```
if(e.keyCode == 79){ //o
      rightPaddle.up = true;
  }
if(e.keyCode == 76){ //l
      rightPaddle.down = true;
  }
if(e.keyCode == 37){ //left
      bottomPaddle.left = true;
  }
if(e.keyCode == 39){ //right
      bottomPaddle.right = true;
  }
}
document.body.onkeyup = function(e){
if(e.keyCode == 81){ //q
      leftPaddle.up = false;
  }
if(e.keyCode == 65){ //a
      leftPaddle.down = false;
  }
if(e.keyCode == 90){ //z
      topPaddle.left = false;
  }
if(e.keyCode == 88){ //z
      topPaddle.right = false;
  }
if(e.keyCode == 79){ //o
      rightPaddle.up = false;
  }
if(e.keyCode == 76){ //l
      rightPaddle.down = false;
  }
```

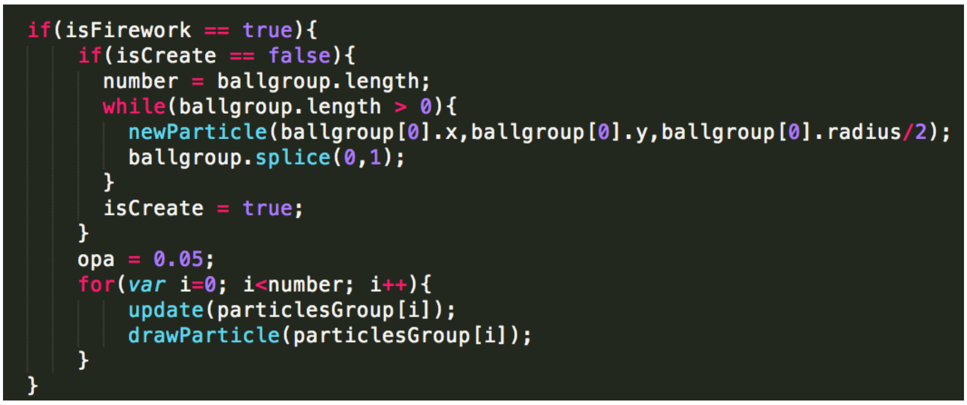
10. Function of status changing



```
if(status == "start"){
  document.getElementById("mainPlay").style.
    display = 'none';
  document.getElementById("menu").style.display =
     'none';
  document.body.onkeydown = function(e){
  if(e.keyCode == 32){ //space
          var startmenu = document.getElementById
            ('start');
          startmenu.parentNode.removeChild(
            startmenu);
          status ="menu";
    }
  }
}
else if(status == "menu"){
  document.getElementById("mainPlay").style.
    display = 'none';
  document.getElementById("menu").style.display =
     'block';
}
else if(status == "main"){
  if(mode == "classical"){
    document.getElementById("mainPlay").style.
      display = 'block';
```

These lines are used for changing the status---“start”, “menu” and “main”. I set their display to “none” at first and set them back when I need them. In this way, I can easily change the status in frames.

11. Function of firework

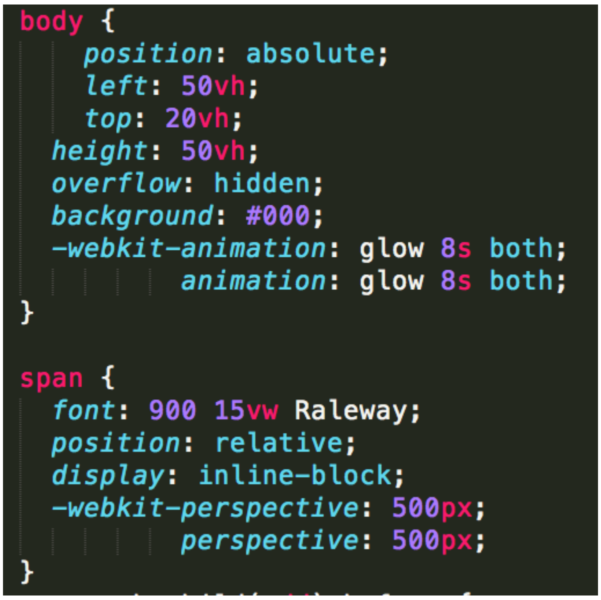


```
if(isFirework == true){
    if(isCreate == false){
      number = ballgroup.length;
      while(ballgroup.length > 0){
        newParticle(ballgroup[0].x,ballgroup[0].y,ballgroup[0].radius/2);
        ballgroup.splice(0,1);
      }
      isCreate = true;
    }
    opa = 0.05;
    for(var i=0; i<number; i++){
      update(particlesGroup[i]);
      drawParticle(particlesGroup[i]);
    }
}
```

We have made a single firework program before so I want to apply this to my PONG game. But actually it was not a easy job. The objects in different programs are completely different. Thanks to Kai, I connected them in an easy way since I have a great format of constructors. They all have position of x and y in common. At last, I successfully made it and it has a wonderful effect with Gravity.

4.2 CSS

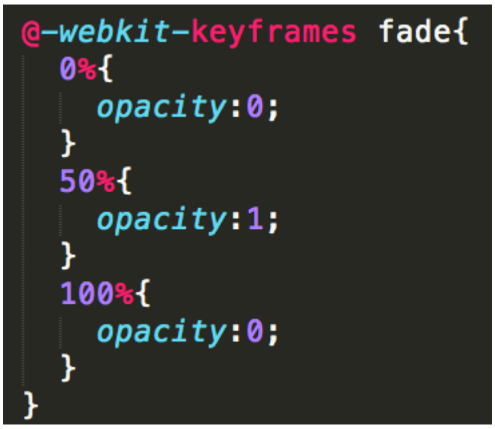
1. The "PONG" font
   This class is used for setting the "PONG" is the start status. I use absolute postion so that it can always display in the center regardless of other elements. These vh and vw mean the percentage of the whole screen. For example, 50vh is just the half of the frame. Also the number of font determines the thickness of it so that I set it to 900 to make it large enough.



```
body {
    position: absolute;
    left: 50vh;
    top: 20vh;
  height: 50vh;
  overflow: hidden;
  background: #000;
  -webkit-animation: glow 8s both;
          animation: glow 8s both;
}

span {
  font: 900 15vw Raleway;
  position: relative;
  display: inline-block;
  -webkit-perspective: 500px;
          perspective: 500px;
}
```

2. The fade effect

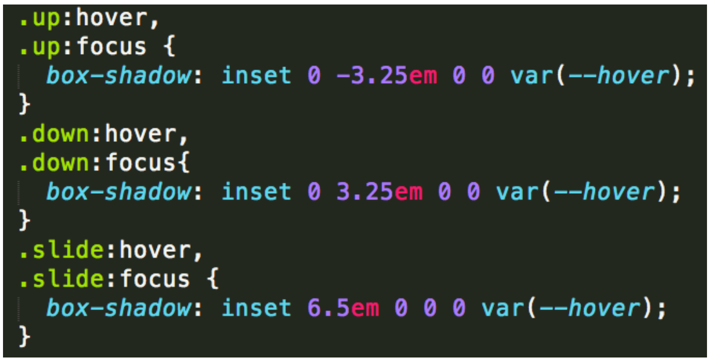
   This animation is used for the fading in and out of the font. At first, I just set it to 0 at 0% and 1 to 100%. But it can just fade in but not fade out. So I set 100% to be a full cycle to make it fade in and out smoothly.



```
@-webkit-keyframes fade{
  0%{
    opacity:0;
  }
  50%{
    opacity:1;
  }
  100%{
    opacity:0;
  }
}
```

3. The button effect

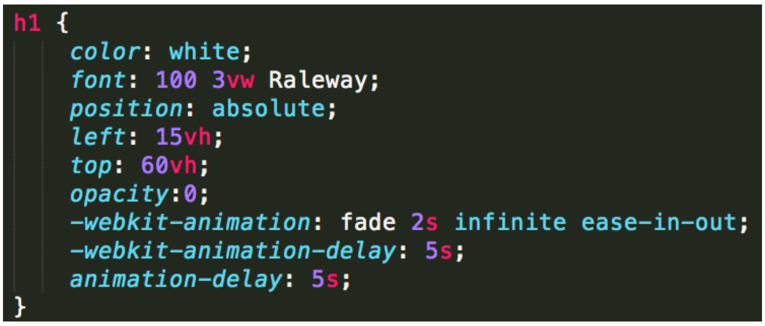
   This class is used for my button animation at second status. When you hover the mouse over the button, it will have the effect of covering from different direction. And these three different classes are connected with three different directions.



```
.up:hover,
.up:focus {
  box-shadow: inset 0 -3.25em 0 0 var(--hover);
}
.down:hover,
.down:focus{
  box-shadow: inset 0 3.25em 0 0 var(--hover);
}
.slide:hover,
.slide:focus {
  box-shadow: inset 6.5em 0 0 0 var(--hover);
}
```

4. The display of the font

   This is how I set the "press space to start". I also set its position to be absolute and make the animation of fading begin after 5 seconds, after the animation of the "PONG".



```
h1 {
    color: white;
    font: 100 3vw Raleway;
    position: absolute;
    left: 15vh;
    top: 60vh;
    opacity:0;
    -webkit-animation: fade 2s infinite ease-in-out;
    -webkit-animation-delay: 5s;
    animation-delay: 5s;
}
```

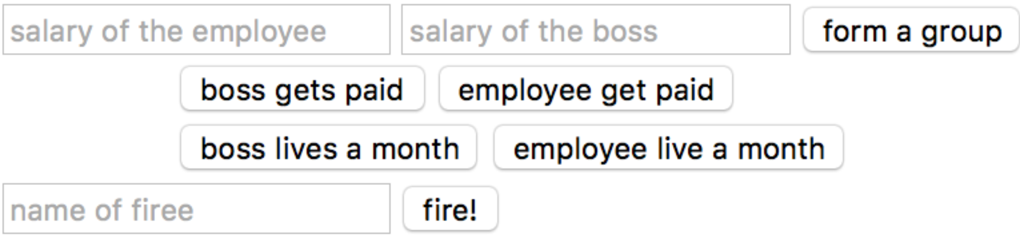
ADDITIONAL PROGRAM

When I am learning the JavaScript and how to make the game, I also did several small programs to learn, understand and apply some useful concepts.

5.1 The Hard Life of Employees

In the first week, we learnt about the basic elements about JavaScript and I apply one of them---prototype---the Hard Life of Employees. I made a group of our four mentees and the manager is Kai. My goal is to apply the prototype function. For example, the employee is a kind of person and the manager is a kind of manager. Therefore, employee, for instance, can have all the functions of the person.

Also, I made the input part in this program. It can only detect the number in the input string so that the users can have a better user experience.



5.2 The Drawing

When practicing the eventListeners and getMousePos, we made a program to draw on the screen. The function is not hard to accomplish and the result is pretty good. The most importantly, I learnt about the mouse function in this program which I applied to my PONG game later.

5.3 The Firework

This is an additional part of our project and we can also use this in our PONG game. In this program, I use many kinds of different objects such as the point or the particles. Also, I met many bugs during the process and how I fixed them is also a valuable experience. At last, mine, Jordan's and Kai's were evaluated by other people and I won because my smooth appearance. This is quite useful for me since I successful accomplished that at last. Also, with Gravity, the visual effect is fantastic and awesome.

IMPROVEMENT

6.1 Back button

I don't have a button to return to the main menu in the game so that people have to refresh the game if they want to change the mode. If I have more time, I will do this to improve the user experience.

6.2 Audio

I want to add music to make the game more fun but indeed I don't find the appropriate sound for this LOL.

6.3 Setting

I can change all the setting such as the speed or the number of the balls in my code but the users can change them so maybe I can set a setting for the game in the future when I have time for this because I want to set everything to be controlled.

6.4 Server

It will be more fun to put this game on the server since two players on different computers can play with each other. But the knowledge about server is kind of difficult to learn so I can't finish it in the last week.

EVALUATION

7.1 Code

I learn how to code in JavaScript. Although I have the experience of coding in C, Java and Python, I still found many differences and met several difficulties. But with the help of Kai, I got familiar with this programming language and also know something about HTML and CSS to apply my JavaScript program into the real website.

More importantly, the process of debugging teaches me a lot. Like one of the Chinese saying, "The failure is the mother of the success." Bug is dispensable for a successful program. I have experienced the time when I have spent several hours just working on several lines but didn't get any progress. This is computer science. We should be patient, devote ourselves completely to the code and this is why I love computer science.

7.2 With mentor

The happiness of working with my mentor, Kai! We spend much time in the lab of UCLA and work and study together. We learn together by googling the information online and testing them by ourselves. Also, we work together even by spending several hours finding and fixing the bugs. It is a very valuable experience for me to study with a student majors in CS in UCLA and it can help me a lot in my admission and the future university life.

What's more, we also talk so much outside the CS. We will eat together and talk on the way since we have many same interests and many things in common. The most importantly, when he invited all of us to have sushi dinner in the restaurants. I "pay" the bill for all of us so that we have built such strong relationship. Maybe after many years, we can still work together (of course I work for him).

7.3 Teamwork

This project can be successful without our team. Although we all do our own project, we all help each other when learning and debugging. We tried to fix many bugs together, following our mentor, Kai. And we all give suggestions to each other about presentation or lab report. We listened or read from each other and make our project to be more successful.

7.4 Life Experience

At last, I also learn some important life experience about life. When talking with Kai, I found that a goal is very important for man. For example, his goal is simple that he just wants to make more money so that he can have his free time to do whatever he wants. Also, computer science is a field that you should devote yourself completely in. I choose it and I love it so that I can gain the happiness of making a great program but I also should undergo the process of debugging heavily.

MEMBER CONTRIBUTION

1. Luke Lu helps me with moving the paddles smoothly. He inspired me to make a signal to mark it so that the paddle can move continuously.

2. Jordan Sun helps me a lot about the menu selecting. He used a array to mark it and I chose to use a string to mark it from his idea.

3. Frost Fu didn’t do very fast but I also learnt a lot when helping him. I helped him about the eventlisteners and the getMousePos. During this time, I had a deeper understanding, making me debug easily later.

REFERENCES

[1]: “Pong.” Wikipedia, Wikimedia Foundation, 31 July 2017, en.wikipedia.org/wiki/Pong.

[2]: “Chrome Web Browser.” Google, Google, www.google.com/chrome/.

[3]: “Sublime Text.” Sublime Text: The Text Editor You'll Fall in Love With, www.sublimetext.com/.