all.js

```
/*
** Ball Constructor Function
** Jakob Hachigian-Kreutzer
** 10/4/18
*/

//function creating balls, utilized through abstraction
function Ball(loc, vel, rad, col, sp){
  // Instance variables
  this.loc = loc;
  this.vel = vel;
  this.rad = rad;
  this.col = col;
  this.sp = sp;
  this.acc = createVector(0, .1);
  //this function calls other functions
  this.run = function(){
    this.checkEdges();
    this.update();
    this.render();
    this.checkPaddle();
  }
  //This function changes the location of the ball
  //by adding speed to x and y
  this.update = function(){
    this.loc.add(this.vel);
    this.vel.add(this.acc);
    this.loc.add(this.vel);
    this.loc.mag();
  }
  //checkEdges() reverses speed when the ball touches an edge
  //keeps shit from going off the edge
  this.checkEdges = function(){
    if(this.loc.x < 0) this.vel.x = -this.vel.x;
    if(this.loc.x > width) this.vel.x = -this.vel.x;
    if(this.loc.y < 0) this.vel.y = -this.vel.y;
    if(this.loc.y > height) this.vel.y = -this.vel.y;
  }
  //render() draws the ball at the new location
  this.render = function(){
    fill(this.col);
    ellipse(this.loc.x, this.loc.y, rad, rad);
  }

  //checking when the ball hits the paddle
  this.checkPaddle = function(){
    //takes location of center of ball - paddle y + 1/2(paddle's length)
    var distY = abs(this.loc.y - 560)
    //looking for if the ball is hitting the top of the bottom of the paddle
    if((distY < 10) && (this.loc.x > mouseX - 125) && (this.loc.x < mouseX + 125) &&
(this.vel.y > 0)){
      this.sp = 1
    }
    if((distY < 10) && (this.loc.x > mouseX - 125 ) && (this.loc.x < mouseX + 125) &&
(this.vel.y < 0)){
      this.sp = 2
    }
  }
}
```

paddle.js

```
/*
** Paddle
** Jakob Hachigian-Kreutzer
**
*/

function Paddle(loc, vel, width, length, col){
  // Instance variables
  this.loc = loc;
  this.vel = vel;
  this.w = width;
  this.l = length;
  this.col = col;
  //this function calls other functions
  this.run = function(){
    this.checkEdges();
    this.update();
    this.render();
  }
//lerp -- paddle follows mouse
  this.update = function(){
    //make paddle lerp to middle of rectangle instead of corner
    paddleLength = width/2
    this.loc.x = lerp(this.loc.x, mouseX-paddleLength, .15)
  }
  //checkEdges() reverses speed when the rectangle touches an edge
  this.checkEdges = function(){
    if(this.loc.x < 0) this.vel.x = -this.vel.x;
    if(this.loc.x > width) this.vel.x = -this.vel.x;
    if(this.loc.y < 0) this.vel.y = -this.vel.y;
    if(this.loc.y > height) this.vel.y = -this.vel.y;
  }

  //render() draws the paddle at the new location
  this.render = function(){
    fill(this.col);
    rect(this.loc.x, this.loc.y, this.w, this.l);
  }
}
```

sketch.js

```
//Global variables
var Balls = [];
var paddle;
var score = 0;
//setup canvas
function setup(){
  var cnv = createCanvas(800, 800);
  cnv.position((windowWidth-width)/2, 30);
  background(20, 20, 20);
  //# of balls loaded
  numBalls = 20;
  loadBalls(numBalls);
  //
  //creating the lerping paddle
  //
  var loc = createVector(400, 550)
  var vel = createVector(0, 0);
  var width = 250;
  var length = 20;
  var col = color(random(0, 255), random(0, 255), random(0, 255))
  paddle = new Paddle(loc, vel, width, length, col);
}
//
//load balls
//
function loadBalls(numBalls){
  for(var i = 0; i < numBalls; i++){
    //where the balls are spawned in
    var loc = createVector(random(100, 600), 20);
    var vel = createVector(random(-3, 3), random(-3, 3));
    var rad = 25
    var col = color(random(0, 255), random(0, 255), random(0, 255));
    var sp = 3
    var b = new Ball(loc, vel, rad, col, sp);
    //add balls to the array
    Balls.push(b);
  }
}


//draw balls + mouse controlled paddle
function draw(){
  background(20, 20, 20, 6000);
  //control the score
  textSize(32);
  fill(random(0,255), random(0,255), random(0,255));
  text("score = " + score, 50, 50);
  //instructions
  if(score < 50){
    fill(random(0,255), random(0,255), random(0,255));
    text("Collect 50 Balls!", 500, 50);
  }
```

```
//if instructions are completed
if(score >= 50 && score <= 100){
  fill(random(0,255), random(0,255), random(0,255));
  textSize(120);
  text("You Win!", 150, 400);
}
//if instructions are completed
if(score == 50){
  score = score + 1
  //prize
  var numBalls = 1000;
  Balls = []
  loadBalls(numBalls);
  for(var i = 0; i < numBalls; i++){
    Balls[i].run();
  }
}
//get rid of outlines
noStroke();
paddle.run();
for(var i = 0; i < Balls.length; i++){
  Balls[i].run();
  var aBalls = Balls[i];
  //splice the balls if they have touched the top of the paddle
  if(aBalls.sp == 1){
    Balls.splice(i,1);
    //adds to score for every ball
    score = score + 1;
  }
  //"reset" the balls if a ball hits the buttom
  if(aBalls.sp == 2){
    //decides how many balls are going to be in the next "reset"
    var numBalls = Balls.length + 25;
    //resets the array (deleted all the current balls)
    Balls = []
    loadBalls(numBalls)
    for(var i = 0; i < Balls.length; i++){
      Balls[i].run();
    }
```

```
      }
    }
  }
```