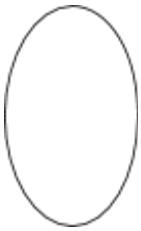
Sketch.js

```
var snake;

var food = [];
var numSeg = 1;
var start = "true"
var font;
var score = 0;
var timeRemaining;

//standarn setup
function setup(){
  textAlign(CENTER, CENTER);
  //new framerate
  frameRate(10);
  var cnv = createCanvas(800, 800);
  cnv.position((windowWidth-width)/2, 30);
  background(200, 244, 66);
  loadSnake();
  loadFood(100);
  img = loadImage("corgi.png");
  fSlider = createSlider(0, 50, 10)
  fSlider.position(780, 5);
  frameRate(frames);
  //load in 100 food and have it become depleted
}
//draw functions
function draw(){
  frames = fSlider.value();
  background(200, 244, 66);
  snake.run();
  //score count
  textSize(50);
  text("score: " + score, 120, 50)
  noStroke();
  Score();
  //calls for food function
  for(var i = 0; i < food.length; i++){
    food[i].run();
    //timer function
    textAlign(700, 100);
```

```
      textSize(50);
  }
//fucntions
  checkLoc();
  deadGame();
  gameStart();
  Score();
  snake.timer();
}
//checking location of the food
function checkLoc(){
```



```
      for(var i = 0; i < food.length; i++){
    var distX = food[i].loc.x - snake.loc.x;
    var distY = food[i].loc.y - snake.loc.y;
    if(distX == (0) && distY == (0)){
      food.splice(i, 1);
      //removes the food
      //would add in a new food if that was the way I wanted it to be
      loadFood(0);
      snake.segments.push(createVector(0, 0));
      console.log(snake.segments.length)
      score++;
    }
  }
}
//snake function call
function loadSnake(){
  var loc = createVector(200, 200);
  var vel = createVector(0, 0);
  snake = new Snake(loc, vel);
}
//loading food into the canvas
function loadFood(numFood){
  for(var i = 0; i < numFood; i++){
    var min = 1;
    //40 * 20 = 800
    var max = 39;
```

```
    var locX = (Math.floor(Math.random() * (max - min + 1) + min)) *
20;
    var locY = (Math.floor(Math.random() * (max - min + 1) + min)) *
20;
    var loc = createVector(locX, locY);
    var f = new Food(loc);
    food.push(f);
  }
}
//controls for the snakes direction
function keyPressed(){
  start = "false"
  if(keyCode === 38){
    snake.vel = createVector(0, -20)
  }
  if(keyCode === 40){
    snake.vel = createVector(0, 20)
  }
  if(keyCode === 39){
    snake.vel = createVector(20, 0)
  }
  if(keyCode === 37){
    snake.vel = createVector(-20, 0)
  }
}
```

```
//game over function
function deadGame(){
  if(snake.status == "true"){
    snake = 0
    score = 0;
    text("Good try bud, refresh for more!", 400, 400);
    loadSnake();
    gameStart();
    gameover();

  }
}
```

```
//pop up page of the beginning of the game
```

```
function gameStart(){
  if(start == "true"){
    textFont()
    fill(5, 225, 15);
    rect(225, 300, 350, 200);
    fill(0, 0, 0);
    rect(240, 315, 320, 170)
    fill(150, 200, 70);
    textAlign(CENTER);
    textSize(40);
    text("Corgi Chowdown", 400, 435)
  }
}
//score function with win function as well.
function Score(){
  if (score > 99){
  fill(255, 0, 5);
  textAlign(CENTER);
  text("good job", 400, 400);
  }
  snake.timer();
  text(snake.timeRemaining, 0, 100, 1450);
  noStroke();
  if (snake.timeRemaining === 0){
    deadGame();
  }
}
```

Snake.js

```
function Snake(loc, vel){
//what the snake needs to know
  this.loc = loc;
  this.vel = vel;
  this.segments = [];
  this.status = "false";
  this.timeRemaining = 100

//other functions of the snake
  this.run = function(){
    this.update();
    this.render();
```

```
    this.dead();
    this.timer();
  }
//snakes movement
  this.update = function(){
    for(var i = this.segments.length - 1; i >= 0; i--){
      if(i > 0){
        this.segments[i].x = this.segments[i-1].x;
        this.segments[i].y = this.segments[i-1].y;
      }else{
        this.segments[0].x = this.loc.x;
        this.segments[0].y = this.loc.y;
      }
    }
    this.loc.add(this.vel);
    this.loc.x = constrain(this.loc.x, 0, 800-20)
    this.loc.y = constrain(this.loc.y, 0, 800-20)
  }
//render function of the snake
  this.render = function(){
    for(var i = 0; i < this.segments.length; i++){
      fill(69, 68, 89);
      stroke(121, 139, 19);
      rect(this.segments[i].x, this.segments[i].y, 20, 20)
    }
    fill(195, 206, 224);
    rect(this.loc.x, this.loc.y, 20, 20);
  }

//the snakes death function
  this.dead = function(){
    for(var i = 0; i < this.segments.length; i++){
      var distX = this.loc.x - this.segments[i].x;
      var distY = this.loc.y - this.segments[i].y;
      if((distX == 0) && (distY == 0)){
        this.status = "true";
        console.log(this.status);
      }
    }
  }
  this.timer = function () {
```

```
     if (frameCount % 40 === 0 && this.timeRemaining > 0) { // if the
frameCount is divisible by 60, then a second has passed. it will stop
at 0
       this.timeRemaining --;
     }
     if (this.timeRemaining === 0) {
    this.dead();
     }

 }

}
```

Food.js

```
//food function
function Food(loc){
//gives the food a random location
  this.loc = loc;

  this.run = function(){
    this.render();
  }
//renders the food on the screen
  this.render = function(){
    fill(255, 0, 0);
    stroke(255);
    image(img, this.loc.x, this.loc.y, 50, 50);
    stroke(121, 139, 19);
  }

}
```