



PROJECT

Classic Arcade Game Clone

A part of the Front-End Web Developer Nanodegree Program

PROJECT REVIEW

CODE REVIEW 5

NOTES

▼ js/app.js 4

```
1 // Enemies our player must avoid
```

SUGGESTION

It is recommended to use `strict` mode:

<https://stackoverflow.com/questions/1335851/what-does-use-strict-do-in-javascript-and-what-is-the-reasoning-behind-it>
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Strict_mode

```
2 var Enemy = function(x, y) {
3   // Variables applied to each of our instances go here,
4   // we've provided one for you to get started
5
6   // The image/sprite for our enemies, this uses
7   // a helper we've provided to easily load images
8   this.sprite = 'images/enemy-bug.png';
9   // enemy coordinate (x,y)
10  this.x = x;
11  this.y = y;
12  // enemy moving speed
13  this.speed = Math.floor(100 + Math.random()*100);
```

AWESOME

Nice work randomizing this enemy parameter!!

```
14 };
15
16 // Update the enemy's position, required method for game
17 // Parameter: dt, a time delta between ticks
18 Enemy.prototype.update = function(dt) {
19   // You should multiply any movement by the dt parameter
20   // which will ensure the game runs at the same speed for
21   // all computers.
22   if (this.x <= 505) this.x += this.speed * dt;
23   else this.x = -10;
24 };
25
26 // Draw the enemy on the screen, required method for game
27 Enemy.prototype.render = function() {
28   ctx.drawImage(Resources.get(this.sprite), this.x, this.y);
29 };
30
31 // Now write your own player class
32 // This class requires an update(), render() and
33 // a handleInput() method.
34 var Player = function() {
35   this.sprite = 'images/char-boy.png'
36   this.x = 200;
37   this.y = 400;
38 };
39
40 // reset player's location
41 Player.prototype.reset = function() {
```

^
AWESOME

Great job on the usage of prototype functions for the class definition!!

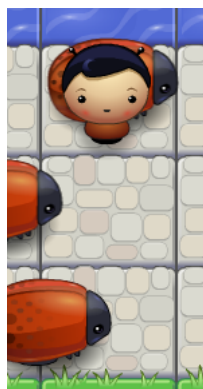
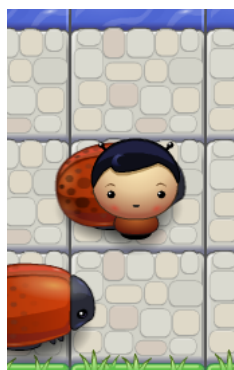
```

42     this.x = 200;
43     this.y = 400;
44 };
45 // Update the player's position based on keyboard input
46 // if collision with an enemy, reset player's position
47 Player.prototype.update = function(dt) {
48     if (this.pressedKey == 'left' && this.x > 0) this.x -= 101;
49     if (this.pressedKey == 'right' && this.x < 400) this.x += 101;
50     if (this.pressedKey == 'up' && this.y > 0) this.y -= 83;
51     if (this.pressedKey == 'down' && this.y < 400) this.y += 83;
52     // reset the pressedKey
53     this.pressedKey = null;
54     if (this.y < -10) {
55         alert("You won!");
56         this.reset();
57     }
58     // test collision with Enemy
59     for (let i = 0; i < 3; i++) {
60         if (this.x >= allEnemies[i].x - 70 && this.x <= allEnemies[i].x + 70) {
61             if (this.y >= allEnemies[i].y - 10 && this.y <= allEnemies[i].y + 10) {

```

REQUIRED

Please re-check this collision logic; the player is able to pass through the enemy on the second and third lanes (from bottom):



```

62         this.reset();
63     }
64 }
65 }
66 };
67 // Draw the player on the screen, required method for game
68 Player.prototype.render = function() {
69     ctx.drawImage(Resources.get(this.sprite), this.x, this.y);
70 };
71
72 Player.prototype.handleInput = function(command) {
73     this.pressedKey = command;
74 };
75
76
77
78 // Now instantiate your objects.
79 // Place all enemy objects in an array called allEnemies
80 // Place the player object in a variable called player
81
82 var player = new Player();

```

```
83 var allEnemies = [];  
84 allEnemies.push(new Enemy(-10, 50));  
85 allEnemies.push(new Enemy(-10, 140));  
86 allEnemies.push(new Enemy(-10, 230));  
87  
88  
89  
90  
91 // This listens for key presses and sends the keys to your  
92 // Player.handleInput() method. You don't need to modify this.  
93 document.addEventListener('keyup', function(e) {  
94   var allowedKeys = {  
95     37: 'left',  
96     38: 'up',  
97     39: 'right',  
98     40: 'down'  
99   };  
100  
101   player.handleInput(allowedKeys[e.keyCode]);  
102 });  
103
```

► README.md 1

► js/resources.js

► js/engine.js

► index.html

► css/style.css

Learn the [best practices for revising and resubmitting your project](#).

[RETURN TO PATH](#)

[Student FAQ](#)