

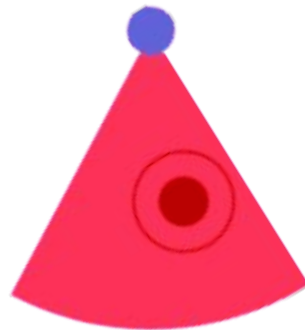
Stealth

Canvas-based web game

Web games design and development

Adriana Bastera

27/12/2020



Index

1. Overview

HTML
CSS

2. Files

3. Main .js files

utils.js
MapCreator.js
code.js

4. Features

5. Highlights

6. Problems

Mild
Headache

Overview: goals

- Level (2.5):
 - Box
 - Level loader
- Player (2.5):
 - Movement
 - Hide
 - Hack
 - Stealth bar
- Enemies (2.5):
 - Fixed
 - Rotating
 - Moving
- Extras:
 - Guided missiles
 - Mines
 - Key

Files

HTML

There is no structure. Calls function.

All elements are rendered in canvas created in runtime.

CSS

Defines background of web page.

The rest of elements' style defined in JavaScript files.

```
<html>
  <head>
    <title>Stealth</title>
    <link rel="stylesheet" type="text/css" href="style.css">
    <script type="text/javascript" src="js/code.js"></script>
    <script type="text/javascript" src="js/utils.js"></script>
    <script type="text/javascript" src="js/MapCreator.js"></script>
  </head>js/

  <body onload="Init()">

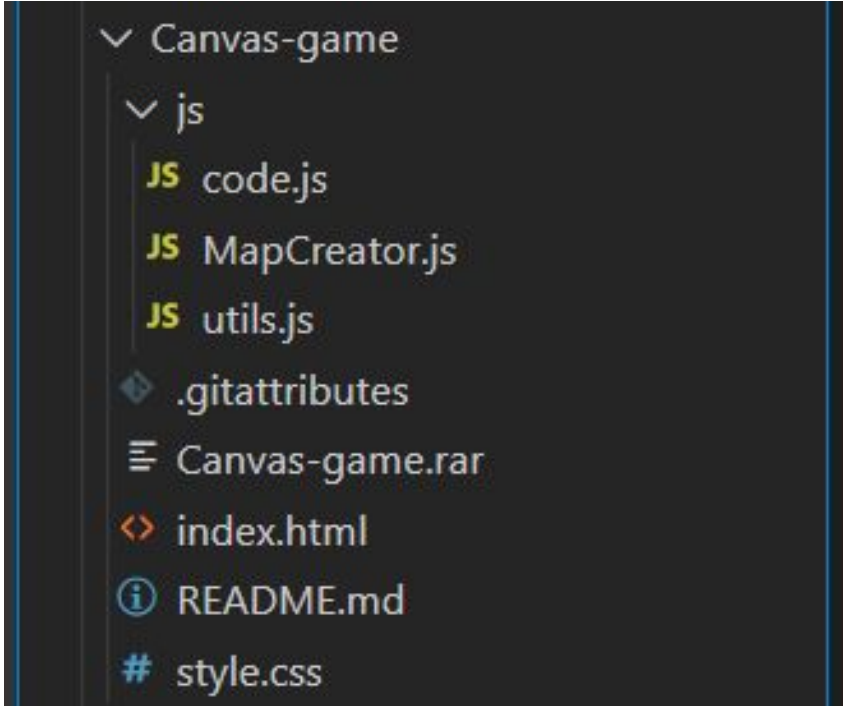
</body>
</html>
```

HTML

CSS

```
body{
  margin: 0;
  position: absolute;
  background-color: #0F0FFF;
}
```

Project files



A screenshot of a file explorer interface with a dark background. The 'Canvas-game' folder is expanded, showing its contents. The 'js' subfolder is also expanded, listing three JavaScript files: 'code.js', 'MapCreator.js', and 'utils.js'. Other files in the root of 'Canvas-game' include '.gitattributes' (with a diamond icon), 'Canvas-game.rar' (with a folder icon), 'index.html' (with a code icon), 'README.md' (with an information icon), and 'style.css' (with a hash icon).

- Canvas-game
 - js
 - code.js
 - MapCreator.js
 - utils.js
 - .gitattributes
 - Canvas-game.rar
 - index.html
 - README.md
 - style.css

.js Files

utils.js, MapCreator.js, code.js

Defines *level1* with the following elements:

- Grid
- Player
- Configuration
- Boxes
- Turrets
- Enemies
- Key
- Stealth bar
- Finish line
- End screens

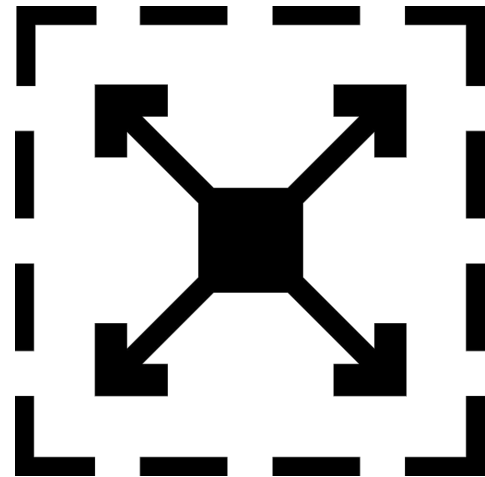
Config defines all common attributes for same class objects.

```
config : {  
  boxes : {  
    color : "#69056f",  
    fill : true,  
    size : 20  
  },  
  balls : {  
    radius : 10,  
    color : "#6262d6"  
  },  
  arcs : {  
    color : "#c1e1ffa0"  
  },  
  screens : {  
    background : "#0F0F0FF0",  
    color : "#FFFFFF",  
  }  
},
```

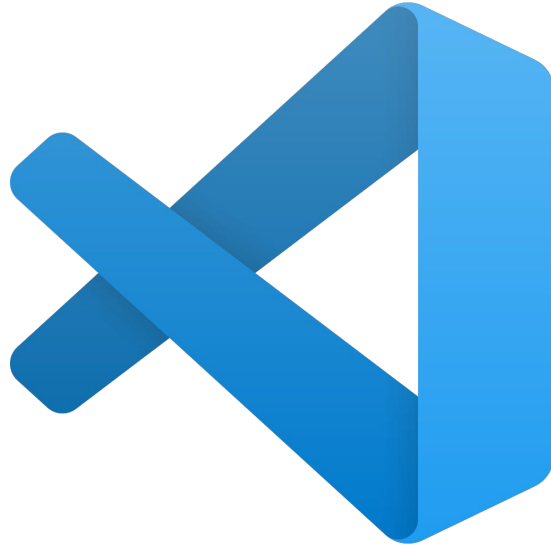
1. Sets game size (width & height)
2. Creates canvas

```
const gameHeight = 710;  
const gameWidth = 1510;
```

Based on my laptop's dimensions

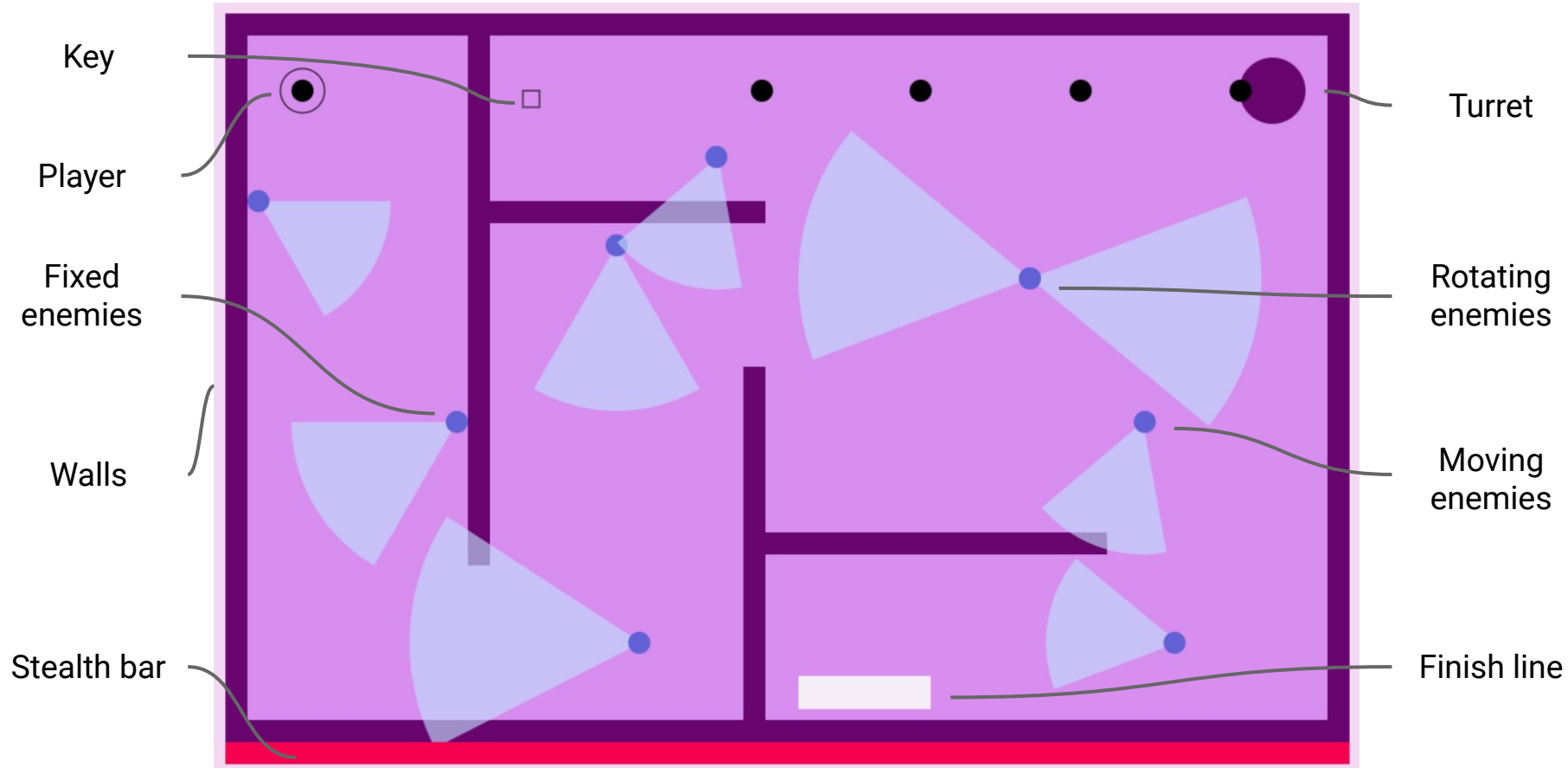


Let's see the structure on code



Features

Classes, achieved goals...



Elements in game

Rather see a demo of it



Note: I'm not good at playing videogames. Sorry for the inconvenience.

Highlights

End screens, canvas size

End screens

Screens for:

- Winning
- Loosing

Game over

Press space to play again

Win

Press space to play again

Reload game when
pressing space

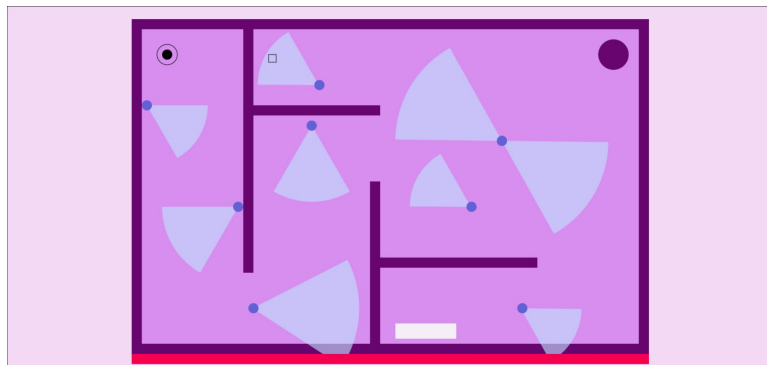
Canvas size

Fixed size:

- Width: 710px
- Height: 1510px

Position:

- Centered in the middle



Problems

Mild and headache problems

Mild problems

- Typographical errors:

`object.lenght != object.length`

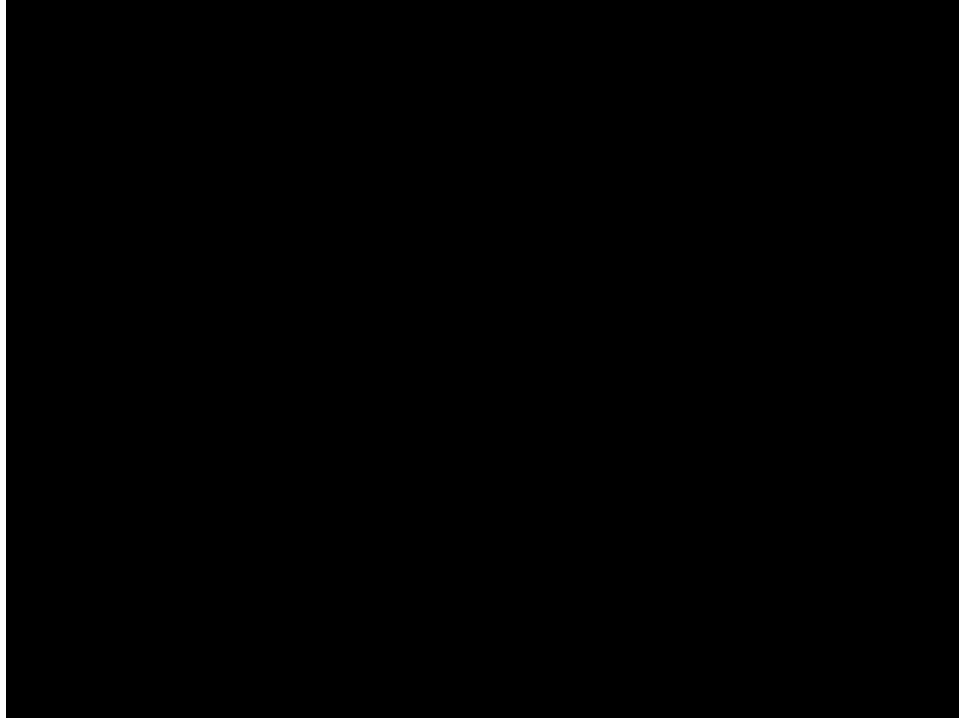
- Not calling functions and expect them to work.

- Define:

`this.collision = { up: false, down: false, left: false, right: false };`

`and use: this.collision.top;`

Headache problems



```
class Vec2 {  
    constructor(x, y) {  
        this.x = x;  
        this.y = y;  
    }  
  
> subtract(v) { ...  
    }  
  
> normalized() { ...  
    }  
  
> add(other) { ...  
    }  
}
```

The end

Thanks for your attention!