INFINITE RUNNER

Daniel Figueiredo
98498

Introduction to Computer Graphics – 2021/2022 - Project 1

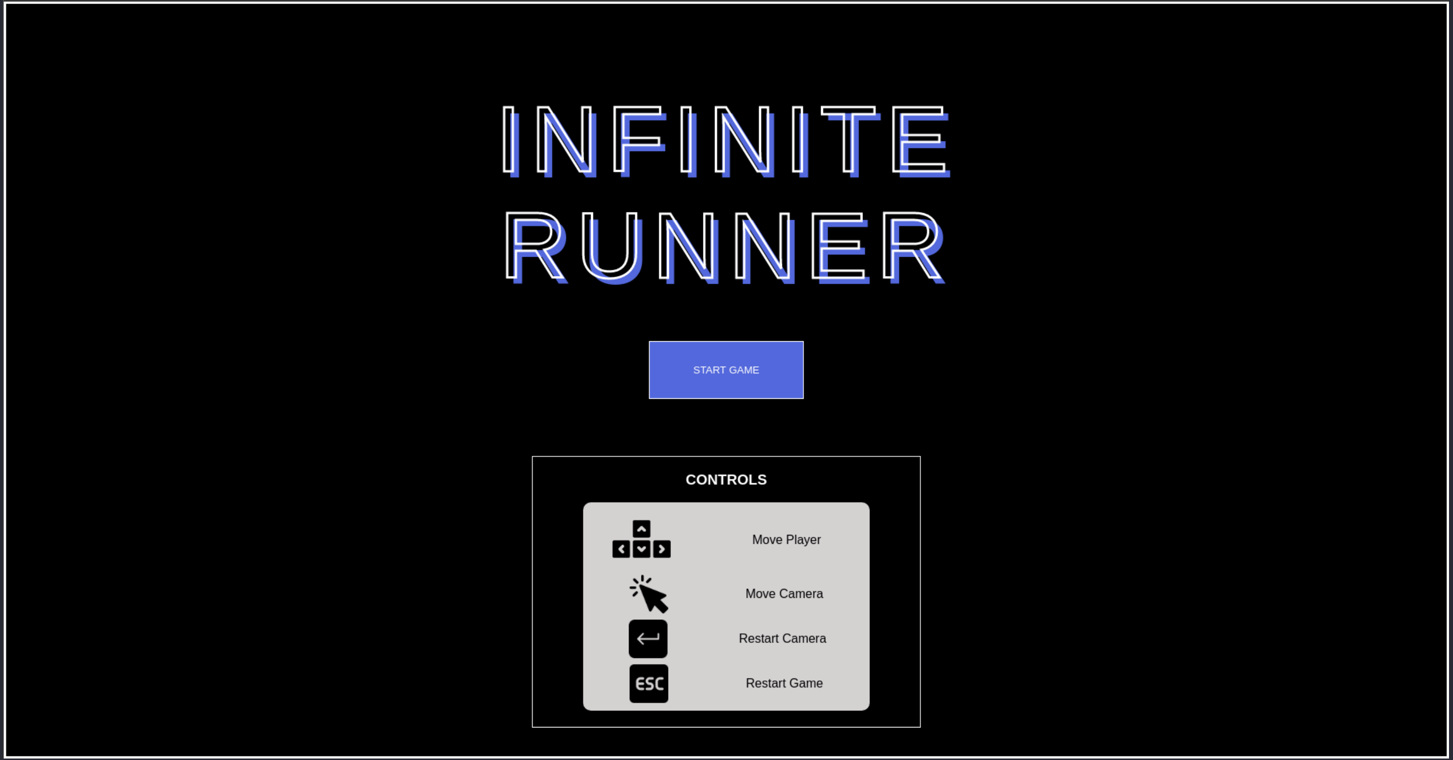
MAIN IDEAS

What is the project about?

1. Infinite Runner
2. Avoid Obstacles
3. Get Lifes
4. Levels

MAIN IDEAS

What can the user do?

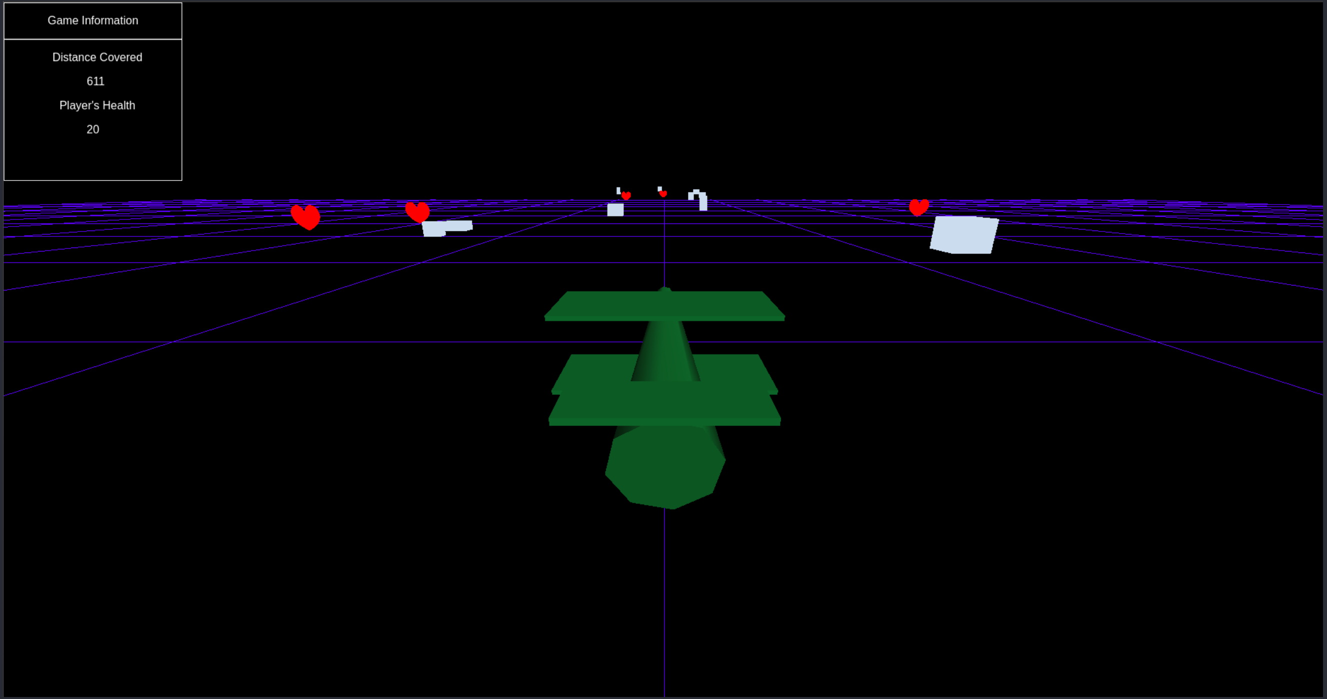


CONTROLS

↑ ← ↓ →	Move Player
	Move Camera
↵	Restart Camera
ESC	Restart Game

MAIN IDEAS

What modules were used?



MAIN IDEAS

Deployment

- Deployed on GitHub Pages
https://daniff15.github.io/icg_1project/

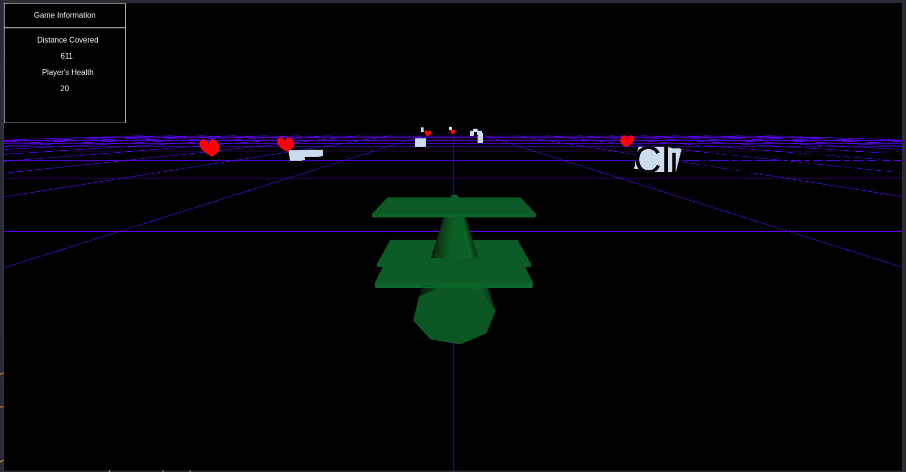
ANIMATION

1. Field Moving
2. Obstacles Approaching
3. Lifes Approaching
4. Level Increase
5. Player Moving

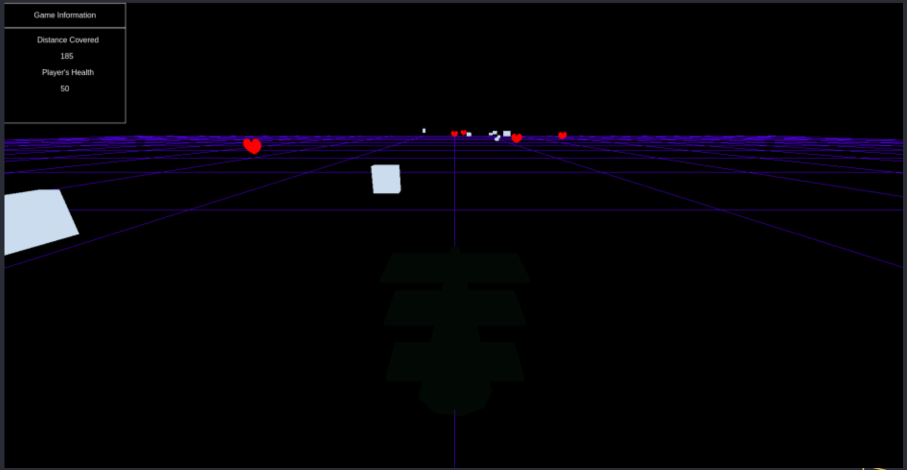
ILLUMINATION

1. Ambient Light on Scene

2. Directional Light



With Directional Light

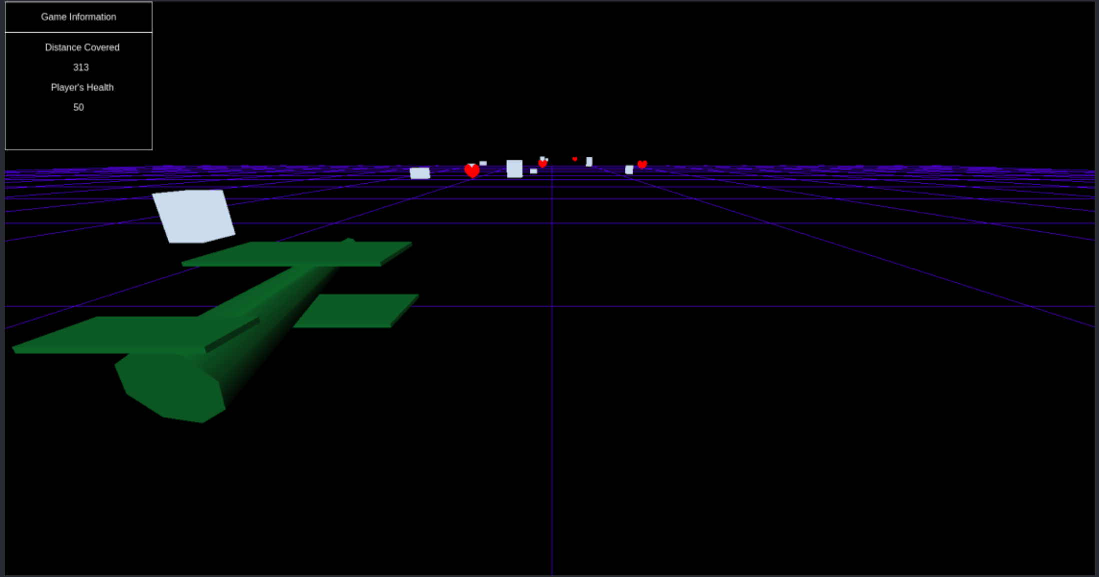


Without Directional Light

USER INTERACTION

ARROW KEYS

Move plane alongside X and Y axis


Game Information

Distance Covered

313

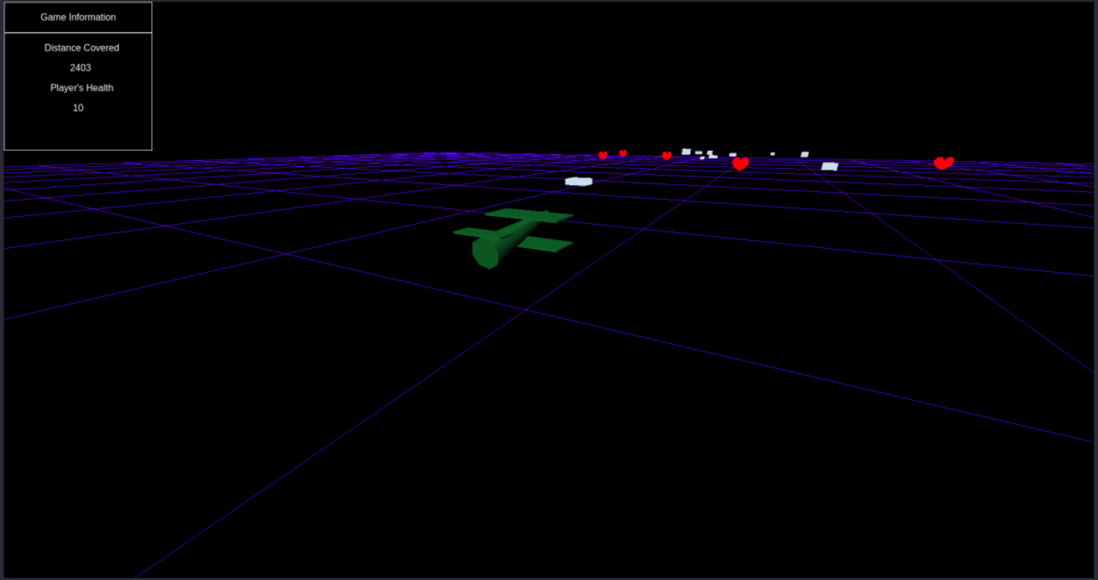
Player's Health

50

USER INTERACTION

MOUSE INPUT

Move camera around the plane



Game Information

Distance Covered

2403

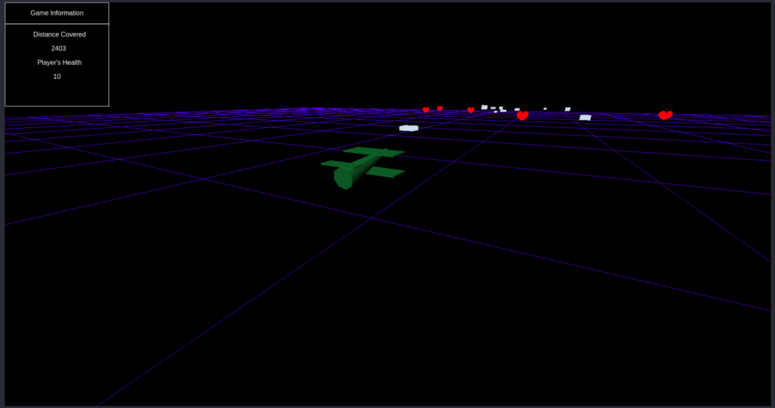
Player's Health

10

USER INTERACTION

ENTER KEY

Reset camera to initial position


Game Information

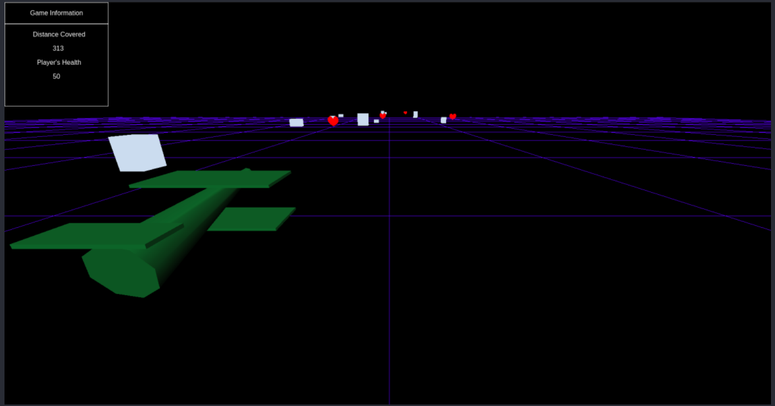
Distance Covered

2403

Player's Health

10

Before clicking ENTER


Game Information

Distance Covered

313

Player's Health

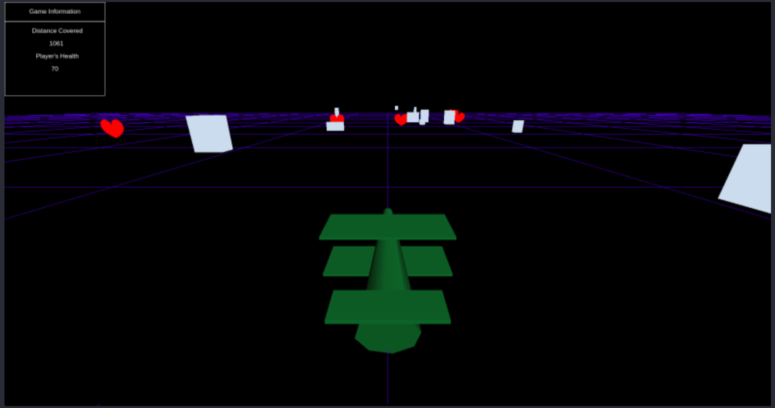
50

After clicking ENTER

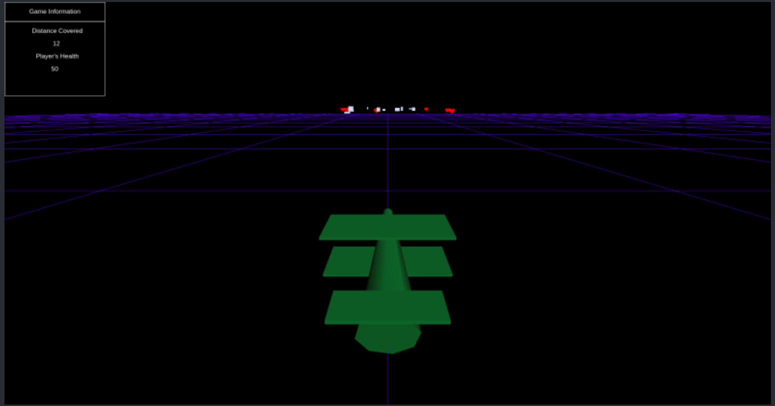
USER INTERACTION

ESCAPE KEY

Reset the game


Game Information
Distance Covered
1061
Player's Health
70

Before clicking ESCAPE


Game Information
Distance Covered
12
Player's Health
50

After clicking ESCAPE

DEVELOPMENT

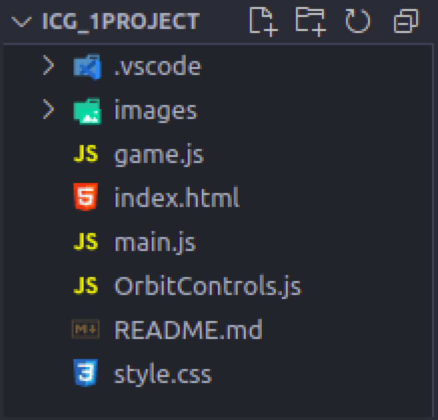
Code Organization

main.js

game.js

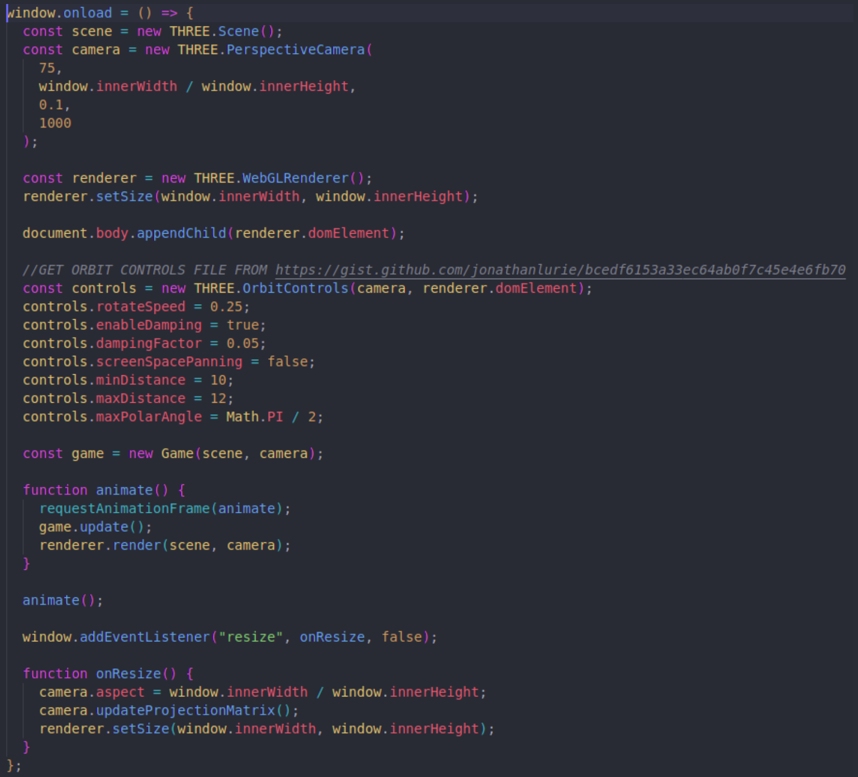
OrbitControls.js

index.html


ICG_1PROJECT
.vscode
images
game.js
index.html
main.js
OrbitControls.js
README.md
style.css

DEVELOPMENT

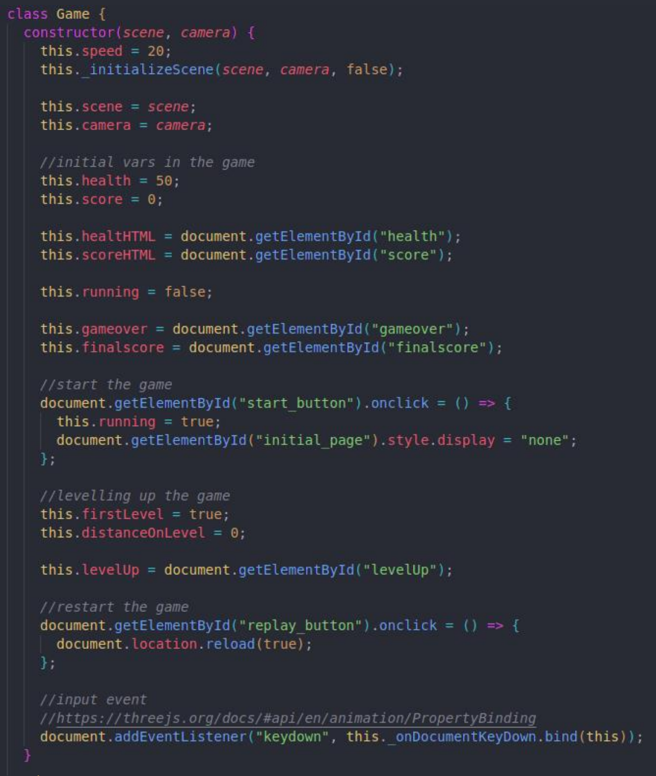
main.js



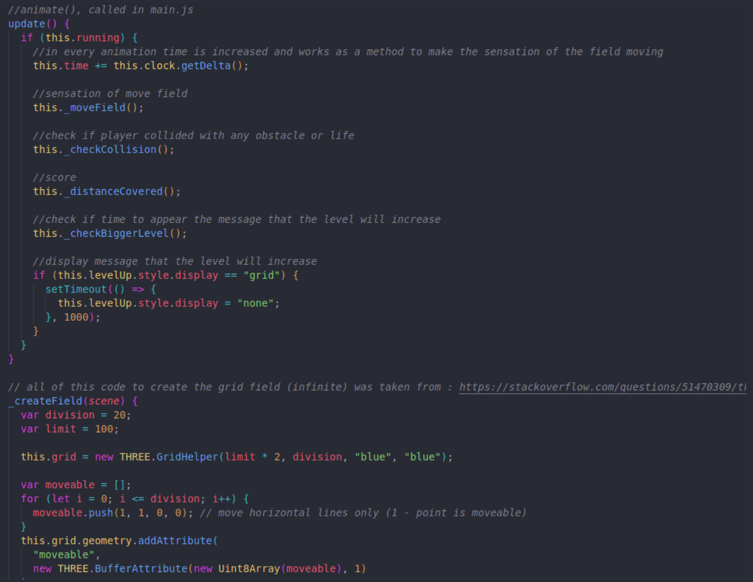
```javascript
window.onload = () => {
  const scene = new THREE.Scene();
  const camera = new THREE.PerspectiveCamera(
    75,
    window.innerWidth / window.innerHeight,
    0.1,
    1000
  );

  const renderer = new THREE.WebGLRenderer();
  renderer.setSize(window.innerWidth, window.innerHeight);

  document.body.appendChild(renderer.domElement);

  //GET ORBIT CONTROLS FILE FROM https://gist.github.com/jonathanlurie/bcedf6153a33ec64ab0f7c45e4e6fb70
  const controls = new THREE.OrbitControls(camera, renderer.domElement);
  controls.rotateSpeed = 0.25;
  controls.enableDamping = true;
  controls.dampingFactor = 0.05;
  controls.screenSpacePanning = false;
  controls.minDistance = 10;
  controls.maxDistance = 12;
  controls.maxPolarAngle = Math.PI / 2;

  const game = new Game(scene, camera);

  function animate() {
    requestAnimationFrame(animate);
    game.update();
    renderer.render(scene, camera);
  }

  animate();

  window.addEventListener("resize", onResize, false);

  function onResize() {
    camera.aspect = window.innerWidth / window.innerHeight;
    camera.updateProjectionMatrix();
    renderer.setSize(window.innerWidth, window.innerHeight);
  }
};
```

DEVELOPMENT

game.js



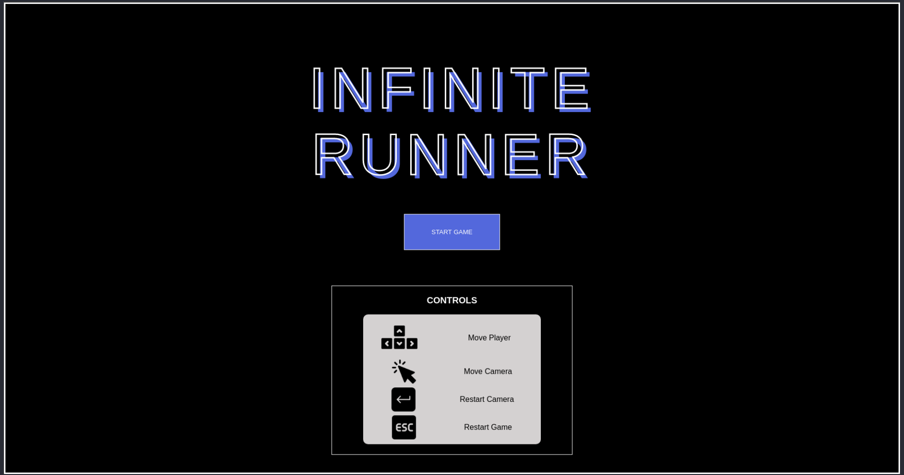
```javascript
class Game {
  constructor(scene, camera) {
    this.speed = 20;
    this._initializeScene(scene, camera, false);

    this.scene = scene;
    this.camera = camera;

    //initial vars in the game
    this.health = 50;
    this.score = 0;

    this.healtHTML = document.getElementById("health");
    this.scoreHTML = document.getElementById("score");

    this.running = false;

    this.gameover = document.getElementById("gameover");
    this.finalscore = document.getElementById("finalscore");

    //start the game
    document.getElementById("start_button").onclick = () => {
      this.running = true;
      document.getElementById("initial_page").style.display = "none";
    };

    //levelling up the game
    this.firstLevel = true;
    this.distanceOnLevel = 0;

    this.levelUp = document.getElementById("levelUp");

    //restart the game
    document.getElementById("replay_button").onclick = () => {
      document.location.reload(true);
    };

    //input event
    //https://threejs.org/docs/#api/en/animation/PropertyBinding
    document.addEventListener("keydown", this._onDocumentKeyDown.bind(this));
  }
```



```javascript
//animate(), called in main.js
update() {
  if (this.running) {
    //in every animation time is increased and works as a method to make the sensation of the field moving
    this.time += this.clock.getDelta();

    //sensation of move field
    this._moveField();

    //check if player collided with any obstacle or life
    this._checkCollision();

    //score
    this._distanceCovered();

    //check if time to appear the message that the level will increase
    this._checkBiggerLevel();

    //display message that the level will increase
    if (this.levelUp.style.display == "grid") {
      setTimeout(() => {
        this.levelUp.style.display = "none";
      }, 1000);
    }
  }
}

// all of this code to create the grid field (infinite) was taken from : https://stackoverflow.com/questions/51470309/t
_createField(scene) {
  var division = 20;
  var limit = 100;

  this.grid = new THREE.GridHelper(limit * 2, division, "blue", "blue");

  var moveable = [];
  for (let i = 0; i <= division; i++) {
    moveable.push(1, 1, 0, 0); // move horizontal lines only (1 - point is moveable)
  }
  this.grid.geometry.addAttribute(
    "moveable",
    new THREE.BufferAttribute(new Uint8Array(moveable), 1)
```

DEVELOPMENT

index.html

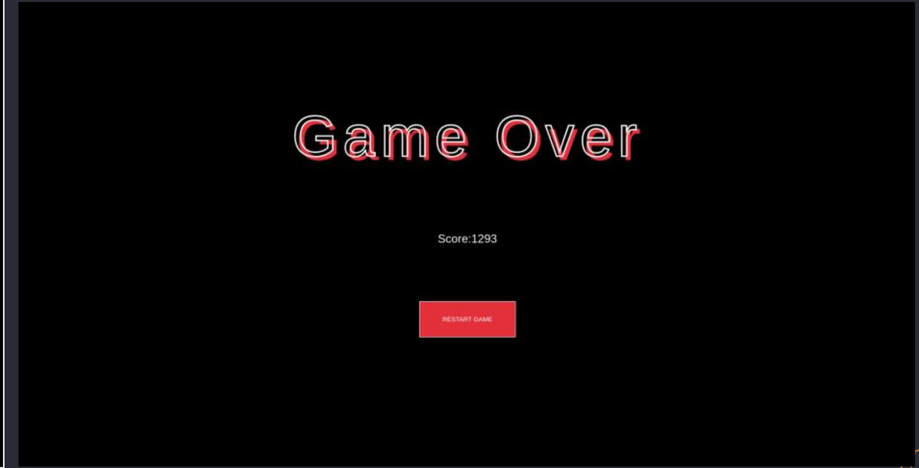


INFINITE RUNNER

START GAME

CONTROLS

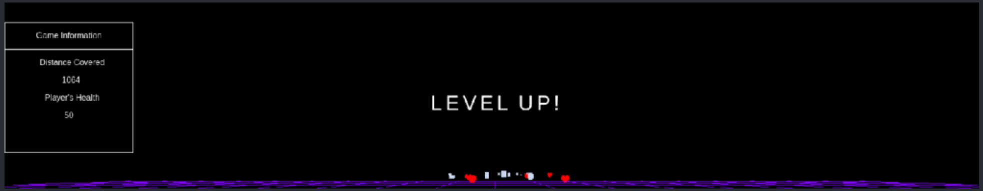
	Move Player
	Move Camera
	Restart Camera
ESC	Restart Game



Game Over

Score:1293

RESTART GAME



Game Information

Distance Covered

1064

Player's Health

50

LEVEL UP!

DEVELOPMENT

Major Difficulties

Make the field and give it the sensation of it moving

Make the objects collide with the player

CONCLUSION

Three.js is not a game engine (and never will be). However I find it to be in a really sweet spot of abstraction where you can build one on top of it.

Three.js is a good documented framework that has a lot of example code in the web.

REFERENCES

Books

- Learning Three.js - the JavaScript 3D Library for WebGL

Sites

- https://gameprogrammingpatterns.com/object-pool.html
- https://threejs.org/
- https://threejs.org/docs/#api/en/animation/PropertyBinding
- https://threejs.org/docs/#api/en/extras/core/Shape
- https://stackoverflow.com/questions/51470309/three-js-and-infinite-forward-grid-movement