

eScape  
Interactive graphics final project

Paoluzzi Fabrizio  
1743773

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Gameplay</b>	<b>3</b>
2.1	Commands . . . . .	3
<b>3</b>	<b>Menu</b>	<b>4</b>
<b>4</b>	<b>Three.js</b>	<b>5</b>
4.1	Implementations . . . . .	5
<b>5</b>	<b>Models</b>	<b>6</b>
<b>6</b>	<b>Animations</b>	<b>8</b>
6.1	Idle . . . . .	8
6.2	Walking . . . . .	8
6.3	Crouch . . . . .	8
6.4	Crouch moving . . . . .	8
6.5	Lamp . . . . .	8
<b>7</b>	<b>Lightning</b>	<b>9</b>
7.1	Shadows . . . . .	9
<b>8</b>	<b>Collisions</b>	<b>10</b>
<b>9</b>	<b>Camera</b>	<b>10</b>

# 1 Introduction

eScape is a stealth game thematically inspired by the Matrix film series and with an animated film aesthetic.

The goal of the user is to reach the glitch to let the agent escape the simulation.

## 2 Gameplay

The gameplay is pretty straightforward, the user has to move the agent across the simulation to reach the glitch.

To reach the goal there will be some obstacles, those obstacles are represented by some lights that the agent has to dodge to avoid being caught.

### 2.1 Commands

The user can interact with the agent and the camera using:

- **WASD KEYS:** The user can move the agent using WASD keys;
- **LEFT/RIGHT ARROWS:** The user can move the camera with the arrows;
- **LEFT SHIFT:** The agent can crouch with the LEFT SHIFT, this will come in handle to avoid being caught in a particular situation;
- **E KEY:** Once the agent reaches the glitch, he can escape by pressing the E KEY.

### 3 Menu

The user is firstly faced with a start menu, the background is rendered in the scene along with the other models, from the menu the user can chose to either start the game, or to enter the instruction tab.

When the user lose or win the game, it will be faced with a menu that asks if the user wants to exit the game or to replay it, the background, as in the main menu, is rendered in the scene, actually it's just a close up of the glitch at the end of the level.

All of the models are loaded at the loading of the page, so the user will probably wait a bit before being able to play for the first time (within the current page), every other time the user will be able to start the game almost immediately.

## 4 Three.js

The game is entirely built using Three.js.

Three.js is a javascript library used to create and display 3D objects on the browser using WebGL, allowing the developers to implements their objects, animations and so on in a simpler way compared to basic WebGL.

### 4.1 Implementations

In the game are implemented some elements of Three.js, such as:

- **The scene:** This is the container of all the elements that need to appear in the game (e.g. the agent, the alley, ...);
- **The camera:** This represents the point of view of the user;
- **The loader:** To load and render the models, I needed a GLTFLoader and a DRACOLoader, the GLTFLoader is needed to load all the 3D models, the DRACOLoader is needed to load the alley model since to reduce its size, it's been DRACO compressed;
- **The light:** The light is used to illuminate the scene and to spot the agent.

## 5 Models

All of the 3D Models have been downloaded on Sketchfab.com :

- **The agent:** The model used for the agent, the main character of the game

This work is based on "Bad Guy"  
(<https://sketchfab.com/3d-models/bad-guy-bed56c2c69a04e55aee44ffc45897e04>)  
by Stan ([https://sketchfab.com/Stas\\_SayHallo](https://sketchfab.com/Stas_SayHallo))

- **The alley:** The model used for the alley, the main environment of the game

This work is based on "environment ally with bar/strip club"  
(<https://sketchfab.com/3d-models/environment-ally-with-barstrip-club-a98c2e9748b24b7fb781d452814304ef>)  
by anthonydpc (<https://sketchfab.com/tonydpc>)

- **The car:** The model used for the car placed in the middle of the alley, its purpose is to act as an hiding object to avoid the light if the agent is crouching behind it

This work is based on "Stylized Rusty Car"  
(<https://sketchfab.com/3d-models/stylized-rusty-car-19dccc6b924adea72411d0dac3c24f>)  
by Renafox (<https://sketchfab.com/kryik1023>)

- **The wall:** The model used for the brick wall in the main menu, this wall is actually placed far away from the main alley and is not present in the game (except for the menu)

This work is based on "Low-Poly Brick Wall"  
(<https://sketchfab.com/3d-models/low-poly-brick-wall-8b3eb0618fb64e818a48e7713828174a>)  
by NoobiePie (<https://sketchfab.com/NoobiePie>)

- **The exclamation:** The model used for the exclamation, this is used when the agent is spotted by the light to let the user know that the agent is in danger

This work is based on "Exclamation Mark 3D icon"  
(<https://sketchfab.com/3d-models/exclamation-mark-3d-icon-35fcb8285f134554989f822ab90ee974>)  
by summer57 (<https://sketchfab.com/summer5717>)

There are some changes to the agent and to the alley models, in particular, the agent has now a changed hierarchical structure, meanwhile the alley has some additional buildings to have a closed space instead of an open alley. The models have been changed using Blender.

In the game the glitch is represented by a 3D Box that uses as a texture the text from Matrix, the animation is done using the elapsed time to scroll the image on the box.

## **6 Animations**

### **6.1 Idle**

When the agent stays still, it starts its idle animation, during this animation the character looks around, rotating its head.

### **6.2 Walking**

When the agent is moving, it starts its walk animation, during this animation it rotates its chest, arms (lower parts), thighs and shins, this results in a plausible way of walking.

### **6.3 Crouch**

When the agent is crouching, it starts its crouch idle animation, with this the agent lowers itself.

### **6.4 Crouch moving**

When the agent is crouching and moving, it starts its crouch walking animation, starting from its crouch position, it starts to rotate its arms, thighs and shins.

### **6.5 Lamp**

The first lamp that the agent will encounter is moving in a basic pattern, the source of the light stays the same but the target is moving, to render this a bit more realistically, the alley lamp placed at the source of the spotlight is rotating accordingly with the target.



## 7 Lightning

In the game I use two types of light sources, the ambient light, used for the illumination of the environment, and the spotlights.

The spotlights are used as the main obstacle of the game and for the menu, there are three spotlights that acts as obstacle and one for the menu.

The agent's health bar depletes when he's directly under the light, at each rendition there's a check on which if the agent is too close to the target object of the light, he'll be spotted, the distance is calculated using the distance between target and spotlight and the aperture angle of the light.

When the agent is crouching behind the car, there is a special hitbox that when in it, the agent cannot be spotted, if he stands up he'll be directly under the light and he'll be spotted.

One spotlight is used to light the wall in the starting menu.

### 7.1 Shadows

Every object outside of the agent casts a shadow when its under a light, this can be seen with the car at the center of the scene and with the trashcan under the moving spotlight.

The agent itself doesn't cast a shadow since when the agent goes under the light, the game will stutter a bit.

Since the camera is placed behind the character, the shadow of the agent could be visible in a small set of cases, so it wouldn't balance out the stuttering.

## 8 Collisions

Each object in the scene has an hitbox, when the character could collide with one of the hitboxes, that movement is removed, blocking the player from going outside the environment and in unwanted places.

Other than the hitboxes, there are two special boxes, the hiding hitbox that is placed behind the car and the exit hitbox placed near the glitch, allowing the agent to interact with it (and escape).

## 9 Camera

The user can move the camera freely around the Y-axis.

The camera is placed behind the target at a fixed distance and it will always look at the agent.

The camera itself has an hitbox, this is used to avoiding unwanted shots, like taking a look from behind the wall.

To avoid blocking the camera in certain places, the camera will be automatically moved when it'll be too close to the wall.