



ENCRYPTOPEDIA

PROCESS DOCUMENT

TABLE OF CONTENTS

Concept	2
Research	4
Project Design/Preparation	19
Asset Production	30
Wireframe	54
Final Prototype	68
Reflection	70



ENCRYPTOPEDIA

CONCEPT

Basic Concept

I wanted to make a 3D interactive experience where the user could move around a physical space in order to learn about ciphertext instead of only using an interface

Some questions I had to ask myself were things like "Okay, what do you want the environment to look like?", "You don't know exactly how to program this kind of thing. How long will it take to learn?", and most importantly, "Can you learn on the fly while delivering on the project every week?"

At this point I decided to spend my spring break working on asset development and building up some of the skills I'd need, but before moving on with anything else, I needed to see what was already out there and how I could build upon it.



20 years of XBOX anniversary Microsoft



ENCRYPTOPEDIA

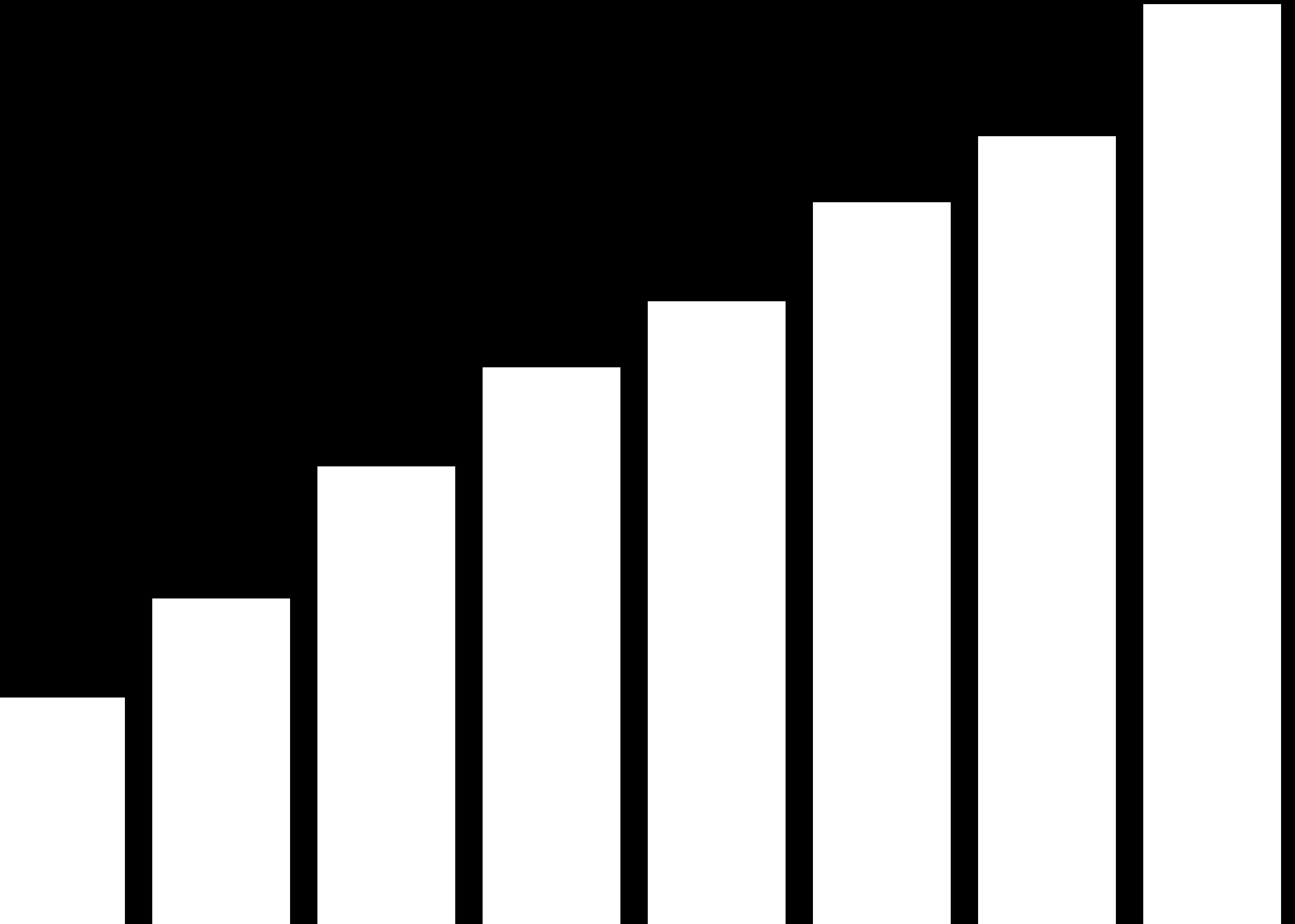
RESEARCH

Research Introduction

So I decided to start by coming up with a name for the project: Encrytopedia. The project is about ciphertexts and has it's style based off of an encryption animation I did, so I think it fits

After this I had to think about a concrete project summary and goal. What exactly will I be teaching the user? How will I do it?

Identifying content sources, themes, and opportunities related to the subject of ciphertext in general was also another thing on the to do list...



Topic Description/Summary

The project will focus on teaching its user what ciphertext is and about the different types that exist. This will be done using a 3d environment that the user can wander through in exhibit-style fashion, with interactive objects containing info about the aforementioned topics. In order to access secondary level information, the user must decrypt information

Teaching about the general topic is what I decided to be the best option since it would require more broad research, thus giving me ample room to research and develop the project at the same time

Possible Content Sources

Since ciphertext is a concept and not an event or phenomena, the sources of information are mostly going to be articles, documentation(e.g web documentation of terminology, functions, etc.), and/or working models/systems that use or relate to ciphertext

There is already a plethora of articles and blogs that cover the topic, so I'll mostly try to look into documentation and working systems as sources of information for the project content.

My next move was to look at some projects/information sources on ciphertext to see what I was up against and what was "missing" in the field

Problems with Current Ciphertext Teaching

The common thread that I've noticed amongst the content I found is the lack of some kind of laymen display of information. Information that's easy to digest and that's isolated in a vacuum that makes as much sense as possible. However, there a few more specific problems that I've noticed

Niche target audience despite being useful and more general content.

Limited teaching styles. Most of the resources that I see out there for learning about ciphertext right now are blogs, articles, or websites that aren't engaging

Solution

I think that gamifying the learning experience can help both make learning about ciphertext fun and make it a bit more mainstream since games appeal to mostly everyone. I won't be making an actual full-fledged game, but the way I'll deliver information on the topic will be done through prompting the user to decrypt a prompt, so there's more engagement and a sort of reward system

One huge goal of the project is to make learning about ciphertext more fun, if not, interesting

Another is to simplify the ciphertext concepts to make them easier to remember and understand

Target Audience

Even though the project is designed to let anyone learn about ciphertext and find it interesting, users that are more experienced with ciphertext might only find entertainment as their main goal for using the game. Because of this, they're a low priority target. My main two audiences are:

Laymen: people who know nothing about ciphertext and are indifferent to it. Basically the only way they'd learn is if they encountered a pleasant experience with the topic or stumbled upon the topic by happenstance

Curious People: People that have a strong sense of curiosity and that tend to go down rabbit holes. Nothing is needed for them to want to learn about a topic since they find interest in most things. For these users, the experience just needs to sate their curiosity in a fulfilling way

Target Devices

Ideally, the game could be played on desktop and consoles. There are going to be camera controls and simple key controls that will probably sync well with those two devices. Not to mention the aspect ration of the game. I wasn't sure of the exact number at this point in time, but I knew that I wanted the screen to be fairly large

Target Environment

It will exist as an isolated user experience, but If I had to concoct a situation in which the project would be displayed, it'd most likely be during an installation or presentation on the topic. More specifically, during a segment of the event where viewers are encouraged to go through the interaction. Another environment would be on a website dedicated to the experience that a user could find when looking up topics related to ciphertext

I spent enough time looking at other content and articles at this point, so my next step was to analyze some UI that I wanted to take inspiration from for my project

UI/UX Trends and Inspiration

UI isn't my strong suit, so I made to sure to have a decent portion of the project and research developed prior to this point. I looked around online and remembered some of my favorite games, experiences, etc. in order to gather some UI inspiration. A few notable sources I came across are:

東亞重工 · TOA Heavy Industries

<https://toahi.net/en/>

I like the visual aesthetic of this site because of the particles and monochrome look, but the navigation isn't intuitive

Space Website Design

<https://dribbble.com/shots/15488061-Space-Website-concept>

A lot of UI and design relating to technology follow certain patterns like the use of interlocked lines(resembles a computer motherboard), sans serif fonts, and some color choices. This design is a good example of what I mean

Nier Automata

<https://www.platinumgames.com/official-blog/article/9624>

The UI in this game follows some of the patterns I listed earlier, but also has small nuances in color and iconography that I want to try. It's also functionally different since it's for a game as opposed to an app, but that's what I wanted to observe since I will be using a bit of game-based information display

UI/UX Trends and Inspiration

There were other sources of inspiration I had as well, but listing all of them would make the document extremely long(more than it needs to be). Those three sources were the main ones I looked at anyway

After this, the last step in the research was to identify some sources that i'd be using for the information in my project

General Topic References/Project Details

There are too many sources of information I have listed, so instead of putting them all in these slides, I decided that it'd be easier to just link my dropbox file that has more detailed project research information and the general topic references I used

[Dropbox paper link](#)



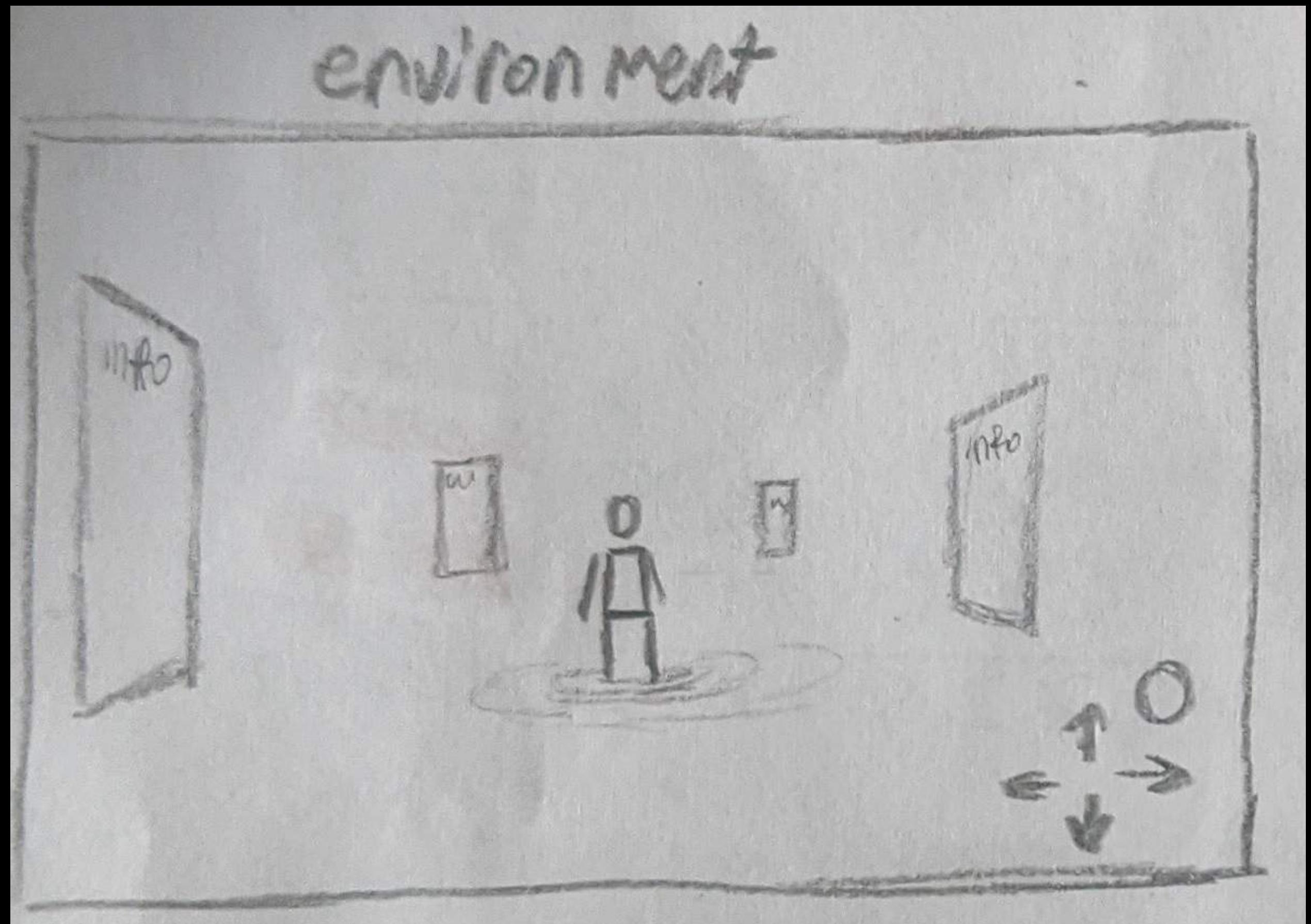
ENCRYPTOPEDIA

PROJECT DESIGN

Sketches I

It was difficult to relay how I wanted the environment and game's overall feel to be without some visuals. I was still developing at this point, so I made sketches to get my ideas on paper

This is how I envisioned the basic function of the game's exploration. Bear in mind that I was coding the game at this point already, so the idea was in my head, but not fully realized as I was deciding while creating

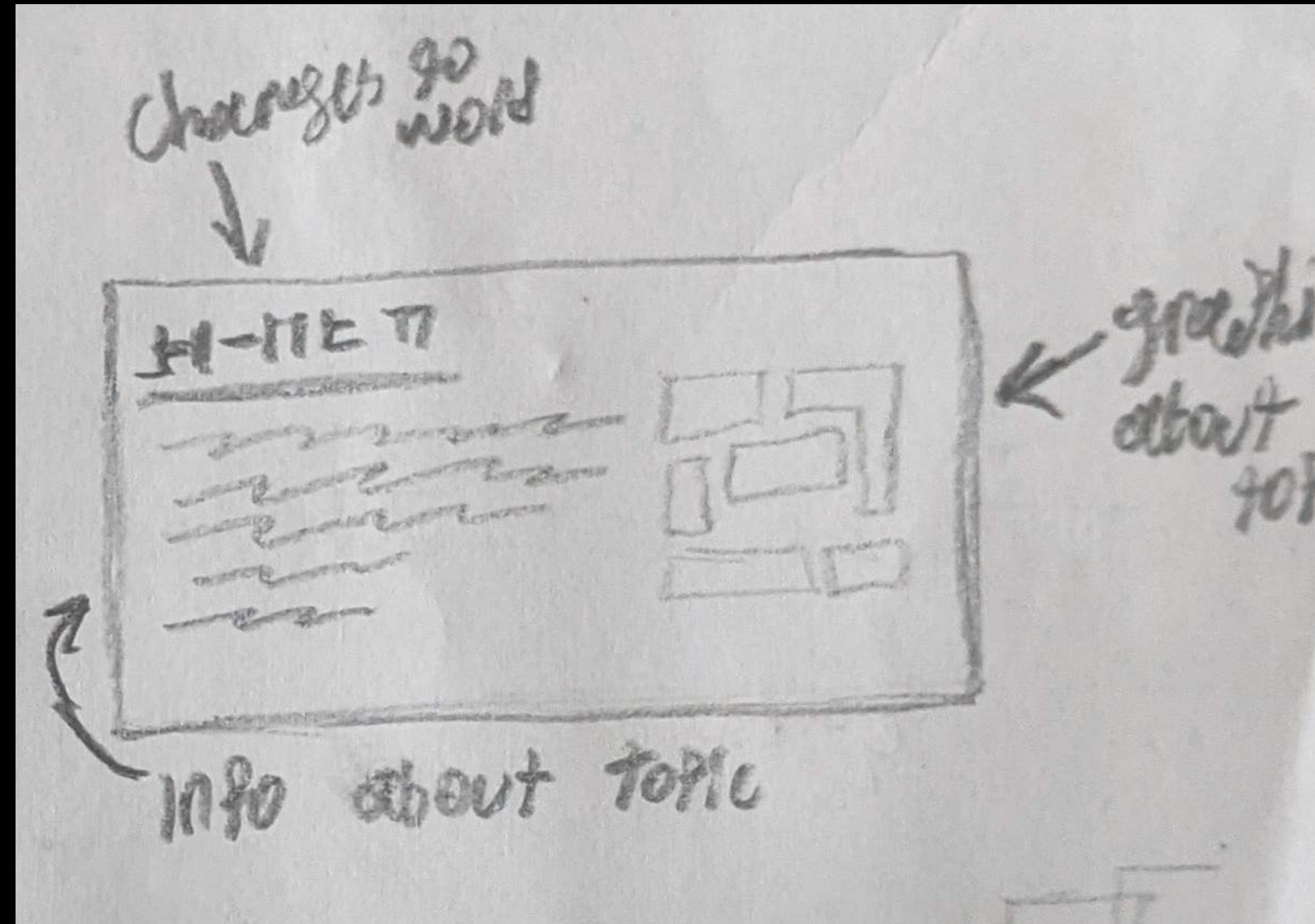


crude sketch of the 3d space

Sketches II

This is how I wanted the information to be displayed. The user will get pop-up windows that contain written information on the cipher they just decrypted

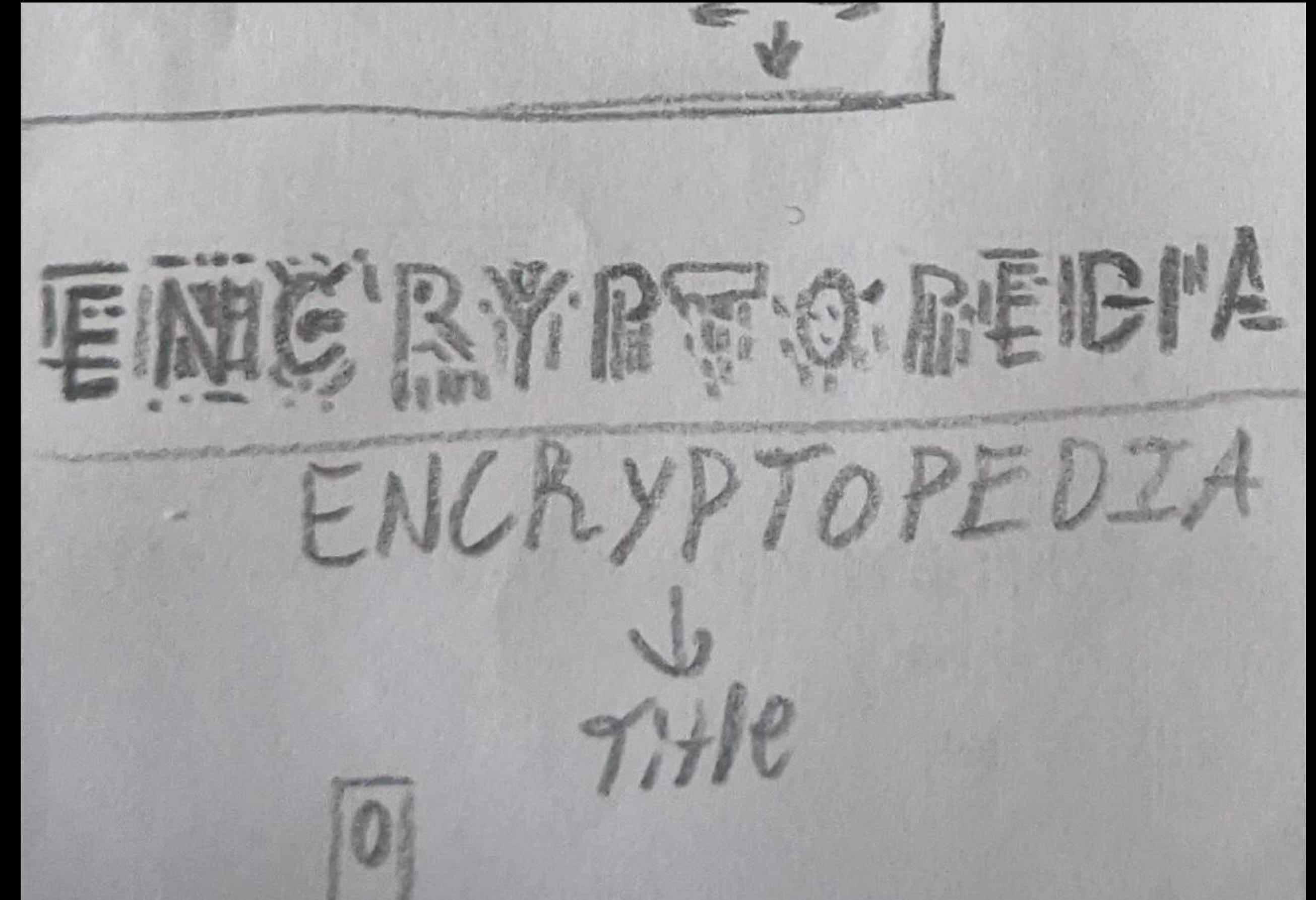
I made sure to note that I'd be using some graphics to take up the white space left by the text. The graphics are also cool in my opinion, so I added them in because I thought that the users would like them too



Sketches III

This is what the title's style will be like. I already mentioned this earlier briefly, but I aimed to use the motifs and style of the language and motion project I made to create some of the style of this project

Coming up with the graphics for the last couple of letters was pretty fun and interesting

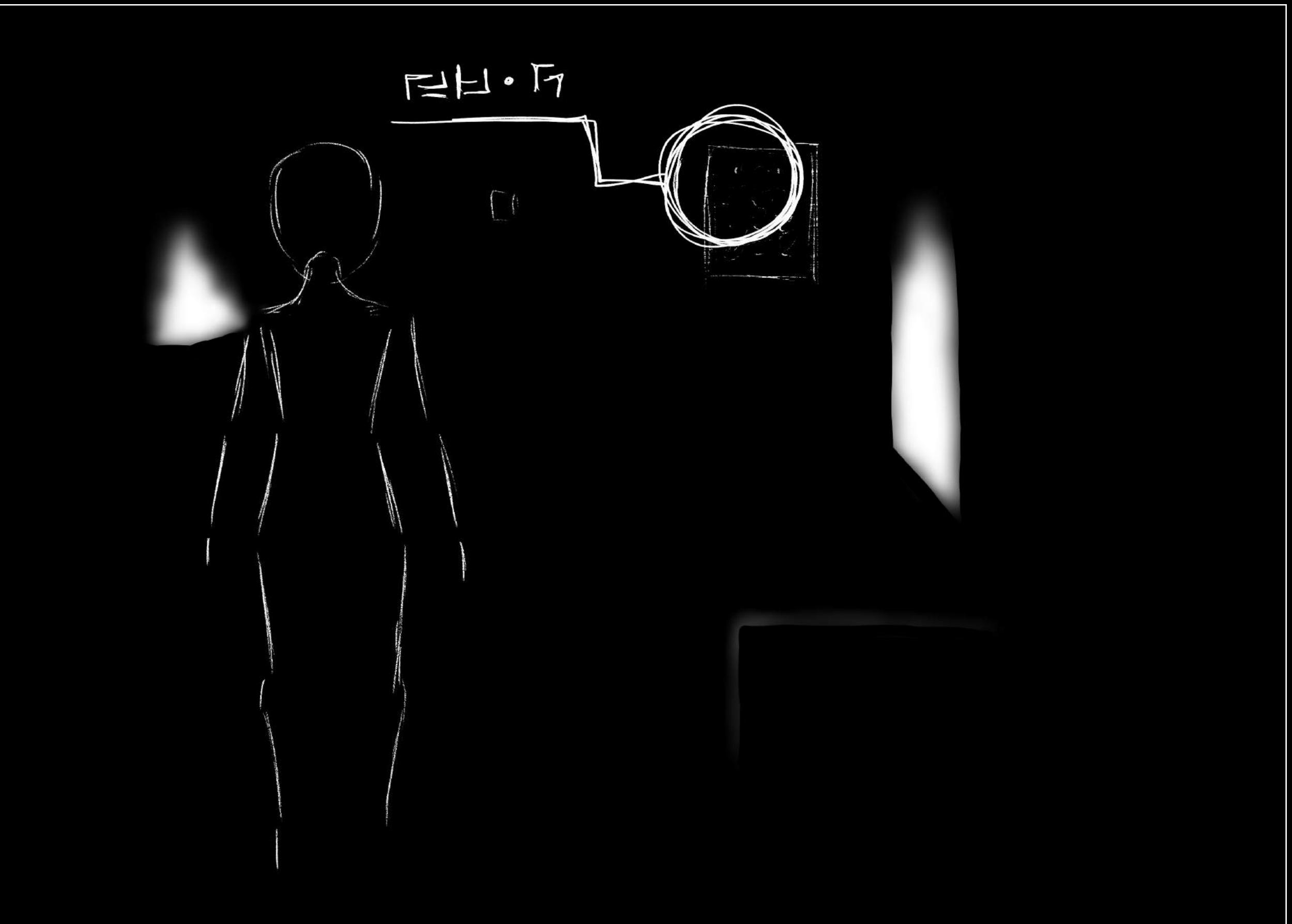


I really like logo/vector graphic design for these kinds of projects

Sketches IV

This is more for figuring out the atmosphere and lighting that I want. The final product still didn't quite become what I originally envisioned, but it's heavily based on this digital sketch

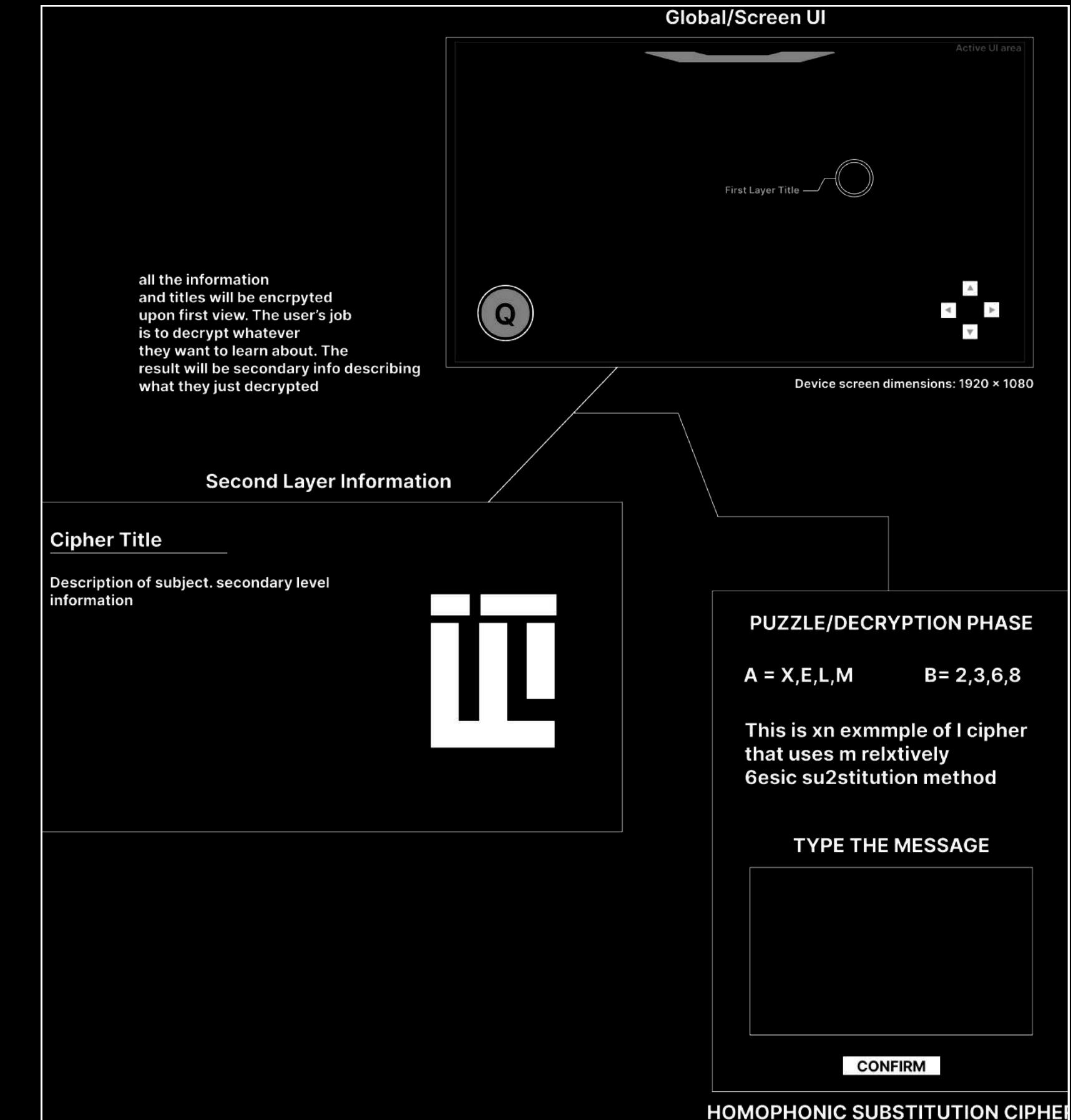
I wanted the environment to be mostly dark, using lighting to indicate the importance of objects



UI and Navigation Ideas

I mainly wanted to capture the vibe or feel that I wanted the UI to have. These are far from the final product, but the futuristic/techno style is what I wanted to relay with this

Because this is a game, there wasn't really too much to design in terms of UI. But for that same reason, I had to make sure to really get the UI down since the user would be looking at the same screens repeatedly in the game



Inspiration and Style Boards

I already had a style and mood in mind when I started the project, but still decided to experiment with 2 directions that fit under the same style just to see if there was anything interesting I could incorporate into the ideas I already had

BLOCKY DIGITAL

KEYWORDS

- Structured
- Solid
- Heavy

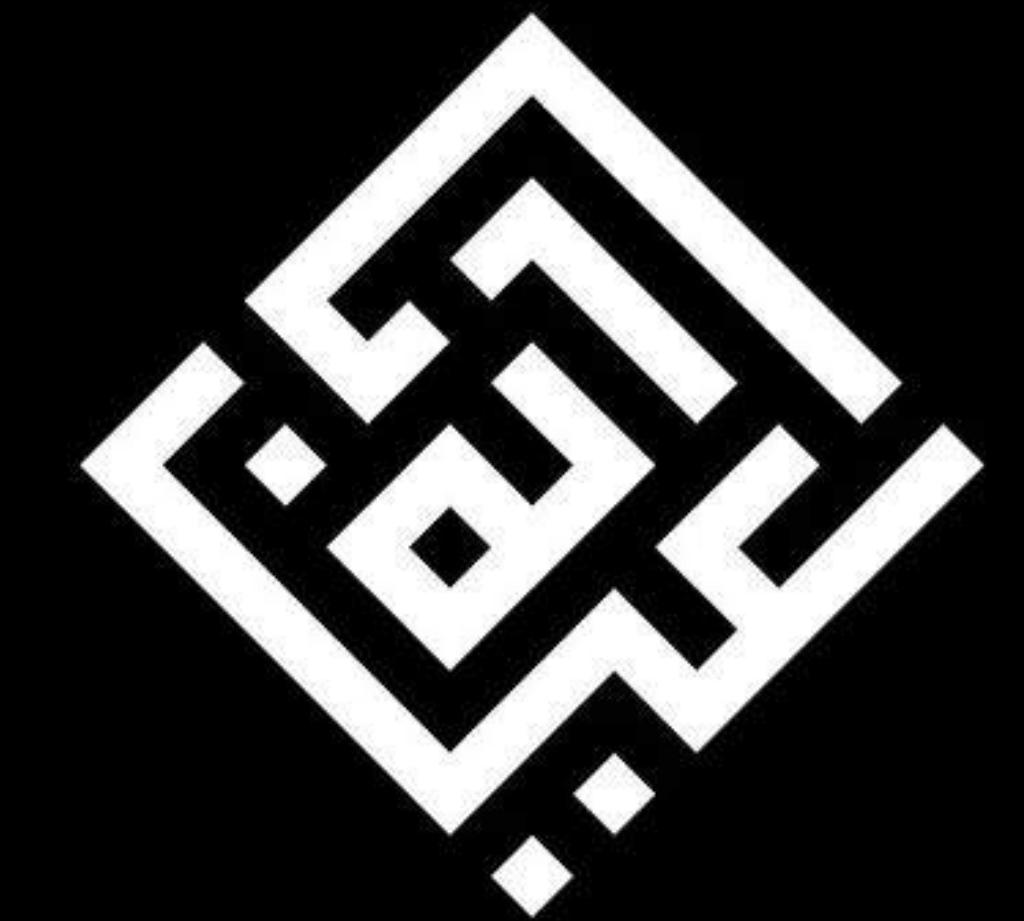
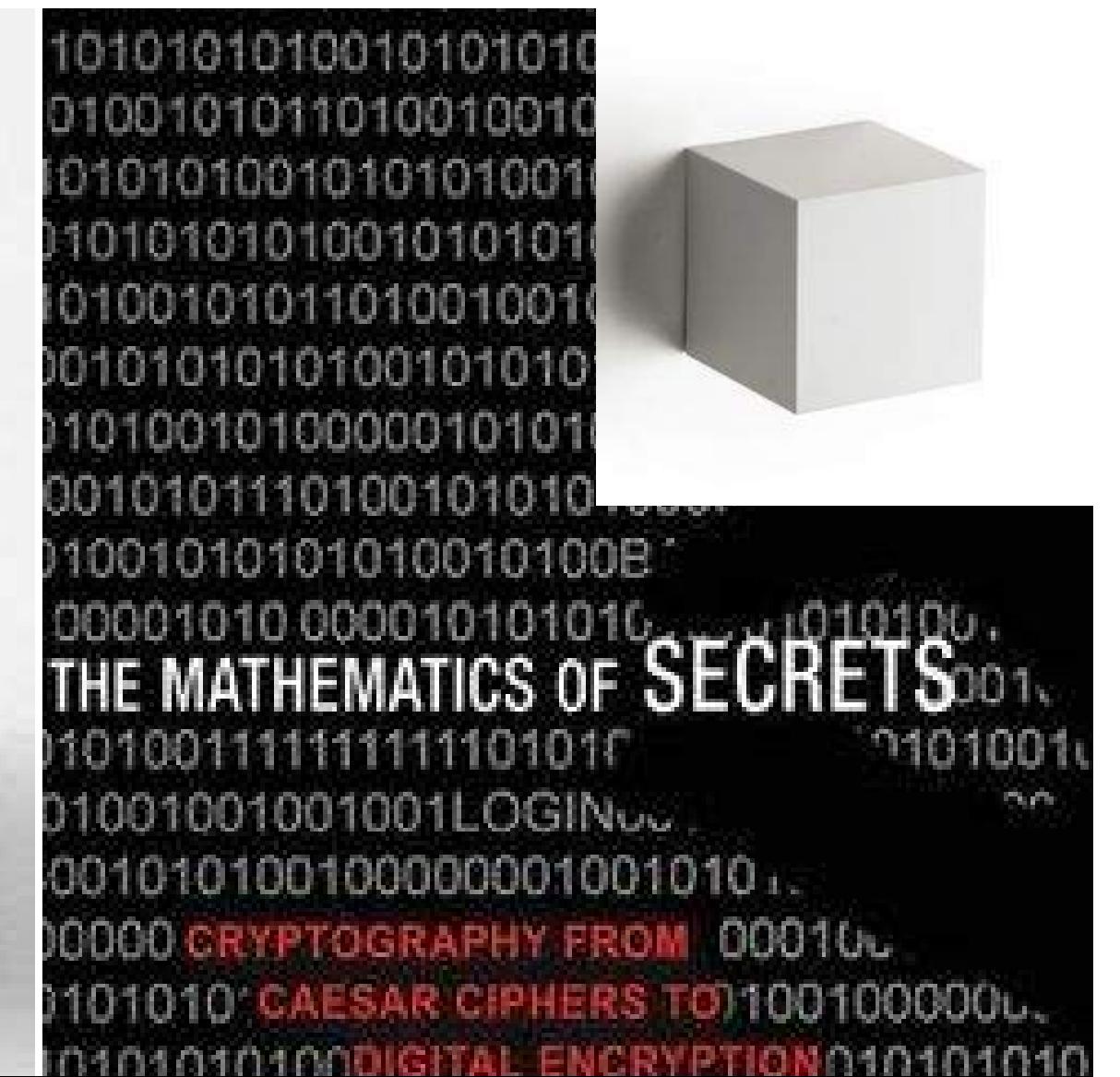
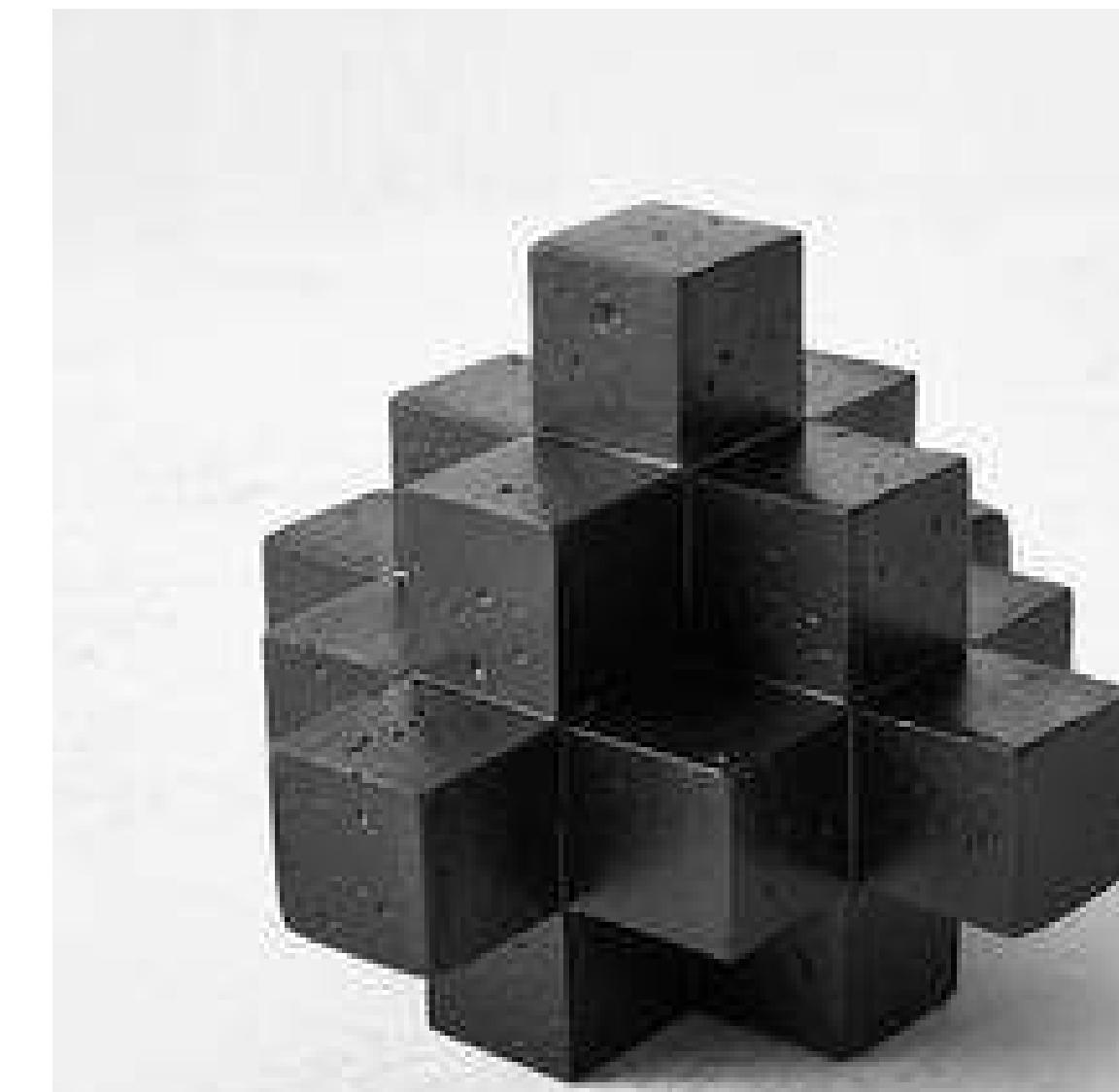
TYPEFACE

INTER BOLD 96px

INTER BOLD 64px

INTER BOLD 48px

INTER REGULAR 48PX



LESSUIK

TECHNO

KEYWORDS

- Orbital
- Future
- Alien

TYPEFACE

ORBITRON 96px

ORBITRON BOLD 64px

ORBITRON 48px

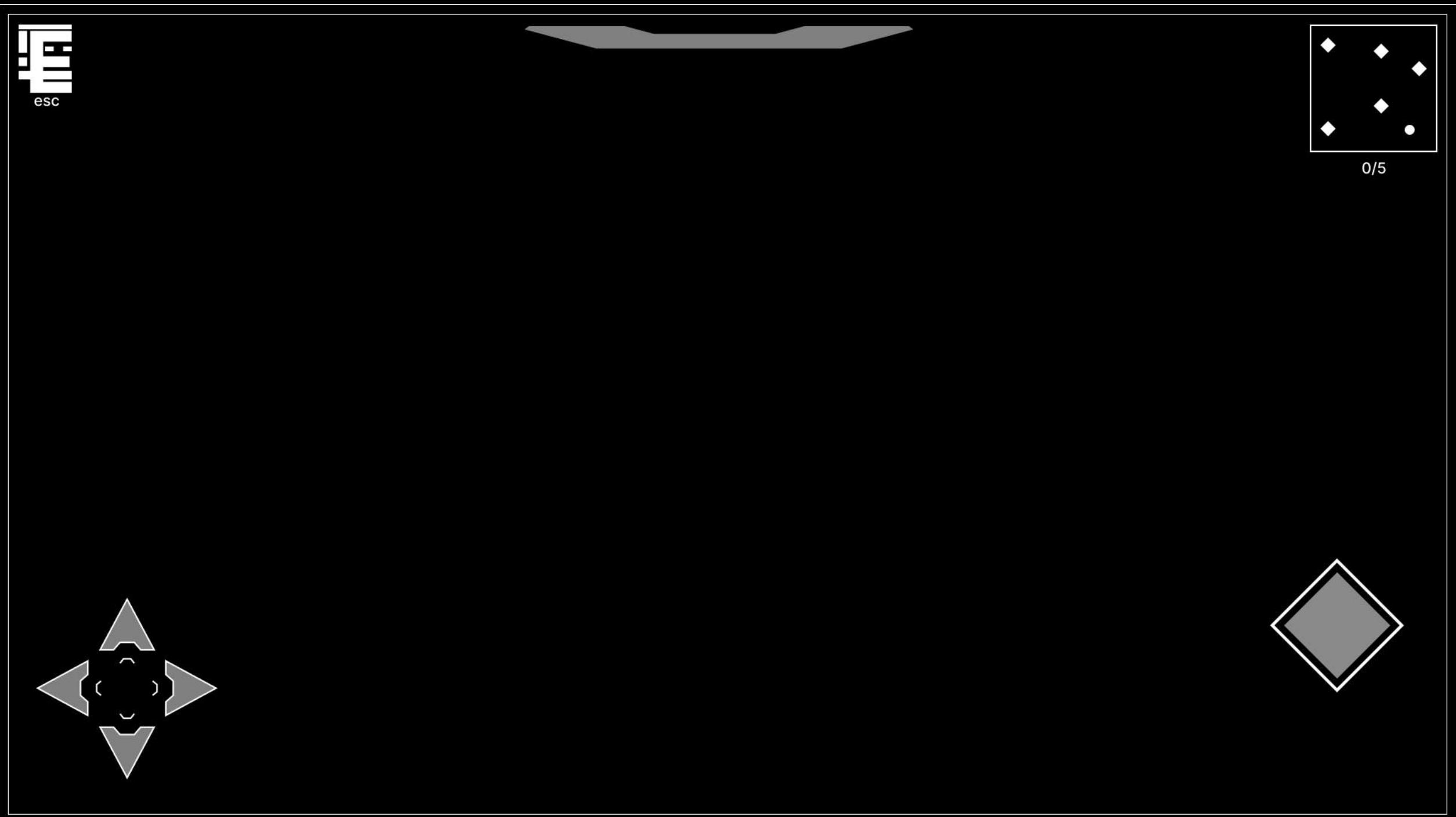
NOTO SANS 48PX



Visual Composition 1

The first visual composition had some branding in it since that's what was said to be needed in the instructions, but it just felt out of place no matter how I included it...

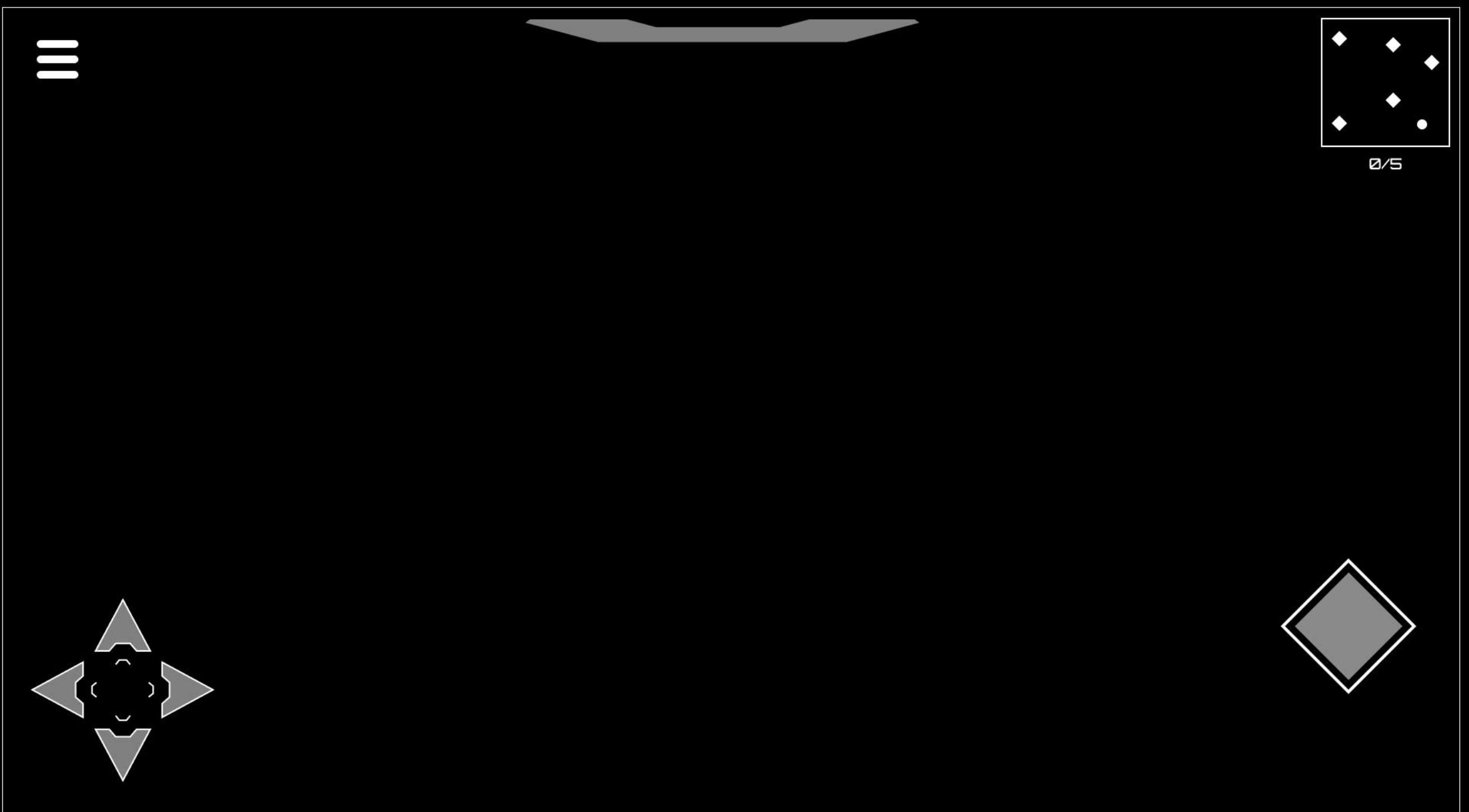
Since this exploration screen will be the one the user looks at 80% of the time, I thought that making my visual compositions be the screen itself would make the most sense



Visual Composition 2

This direction of the UI was one I liked more. Even though it lacked branding, it fit more with the icon based UI I originally wanted. It still has flaws in terms of unity and spacing, but was a step in the right direction

I figured that this was enough UI iteration for now since I still needed an actual environment to implement it in.





ENCRYPTOPEDIA

ASSET PRODUCTION

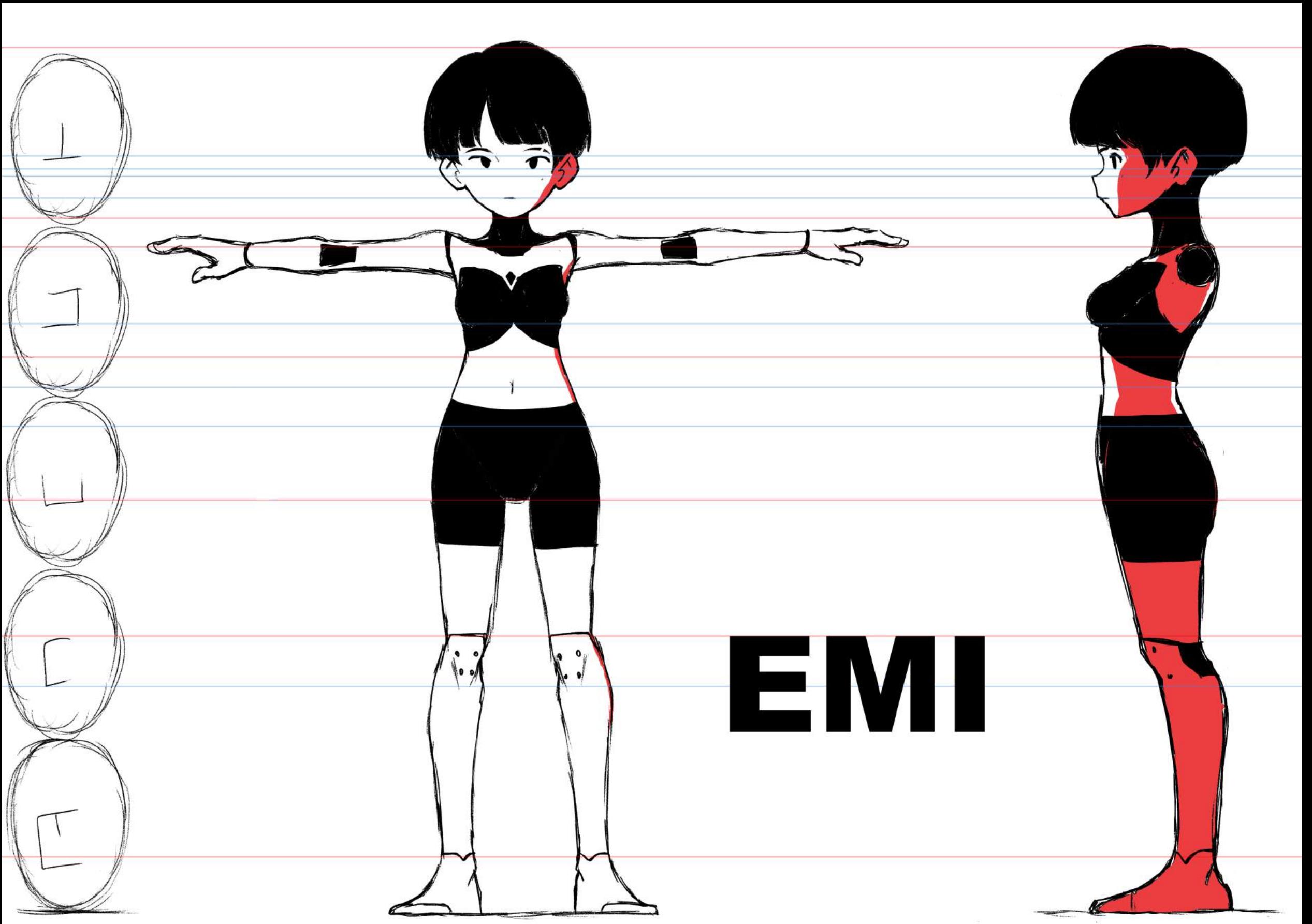
Overview

There was a lot of work that went into making the assets for the project. I'll have to divide the work into some sections to organize it all. I basically went through 3 stages: Modeling, Programming, Implementation/Testing

Modeling I : Character Design

Before even heading into my 3D software of choice, I had to draw a character sheet for the model that the users would be controlling. I outlined proportions and kept her design simple to save time

The lore is that she's an AI model made to decrypt information.

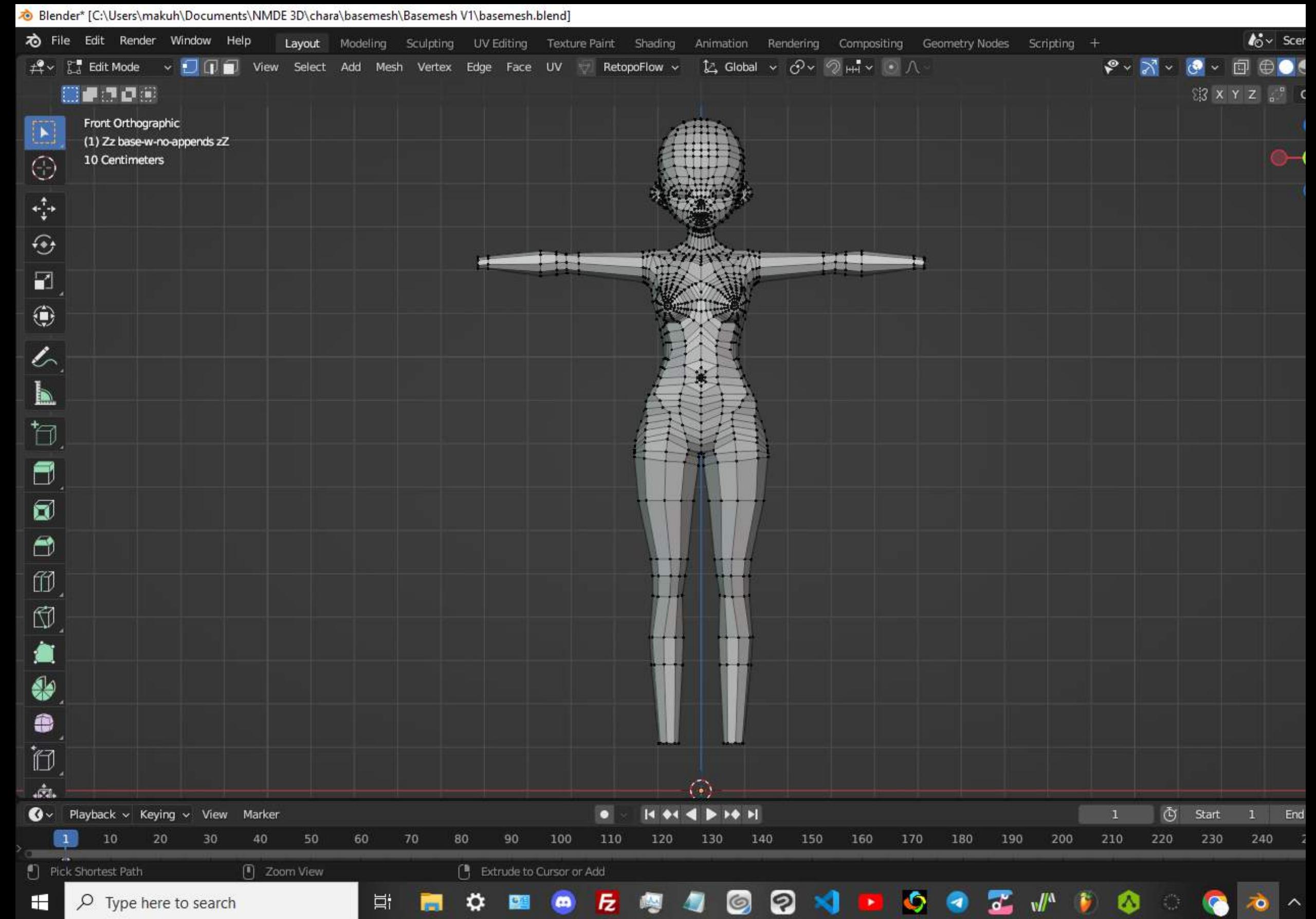


Modeling II : Basemesh Part 1

After sketching the character and measuring out the anatomy I wanted, I hoped into Blender and began modeling a base mesh that I could then turn into my character

I started out by modeling large, fairly simple parts of the body first. The hands and feet are complex, so they come later

Nothing is subdivided yet, but already, I noticed that the topology around the waist was a little weird...

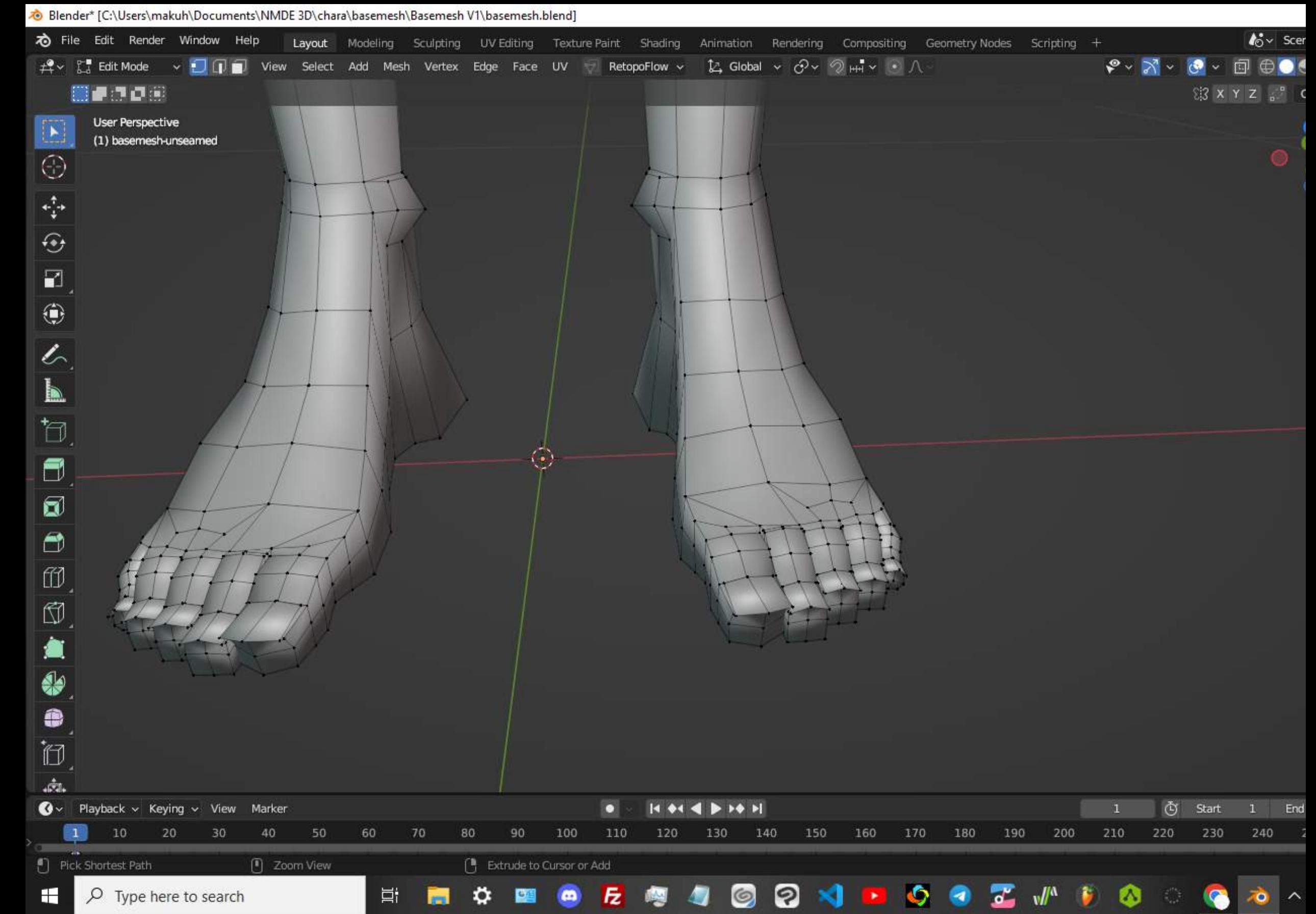


There were many failed attempts before even getting to this point.

Modeling III : Basemesh Part 2

I began modeling the feet by extruding the ankles and subdividing where needed.

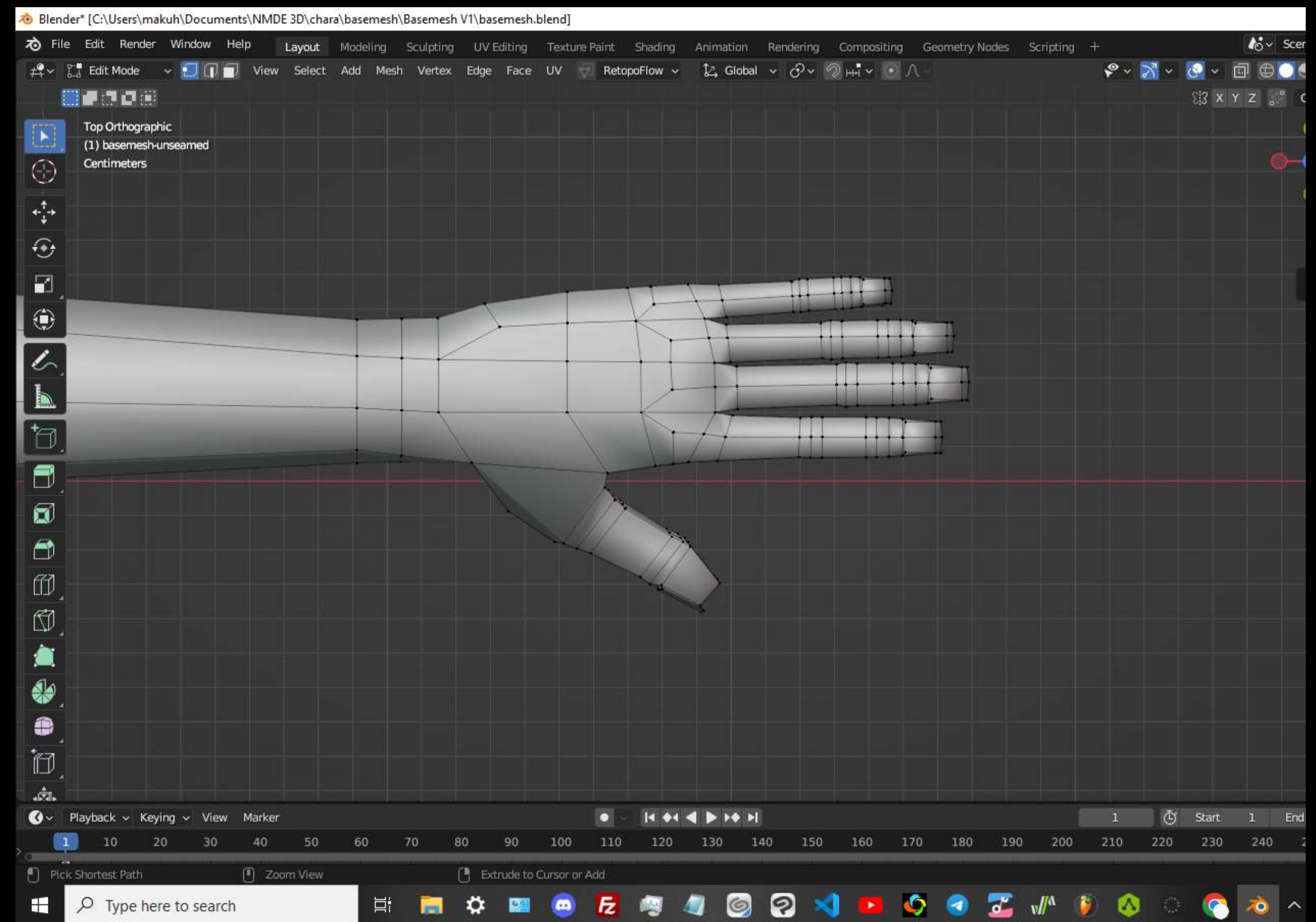
This process decreases the chances of error when compared to modeling the feet separately because you don't have to bridge any geometry (thus, no worrying about vert count in edge loops, recalculating inside/outside faces, etc.).



Modeling IV : Basemesh Part 3

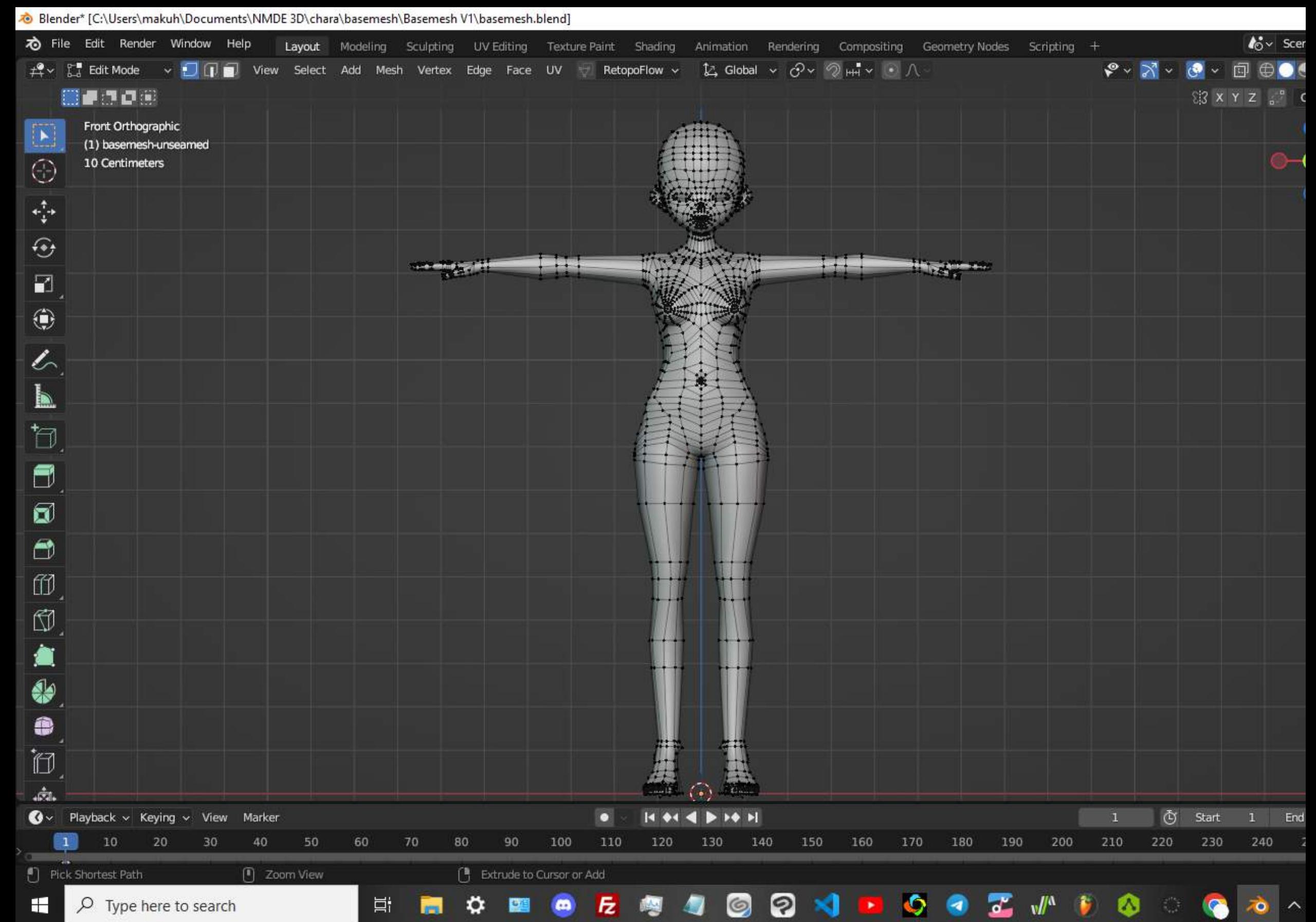
The hands were tricky because I couldn't find a way to utilize the 8 verts I had on the wrist edge loop to create the fingers and thumbs. I got it down eventually, but the topology could be better...

I was having a lot of fun learning about character modeling at this time. I also had a friend who was pretty good at hard surface modeling give me some pointers. No pun intended.



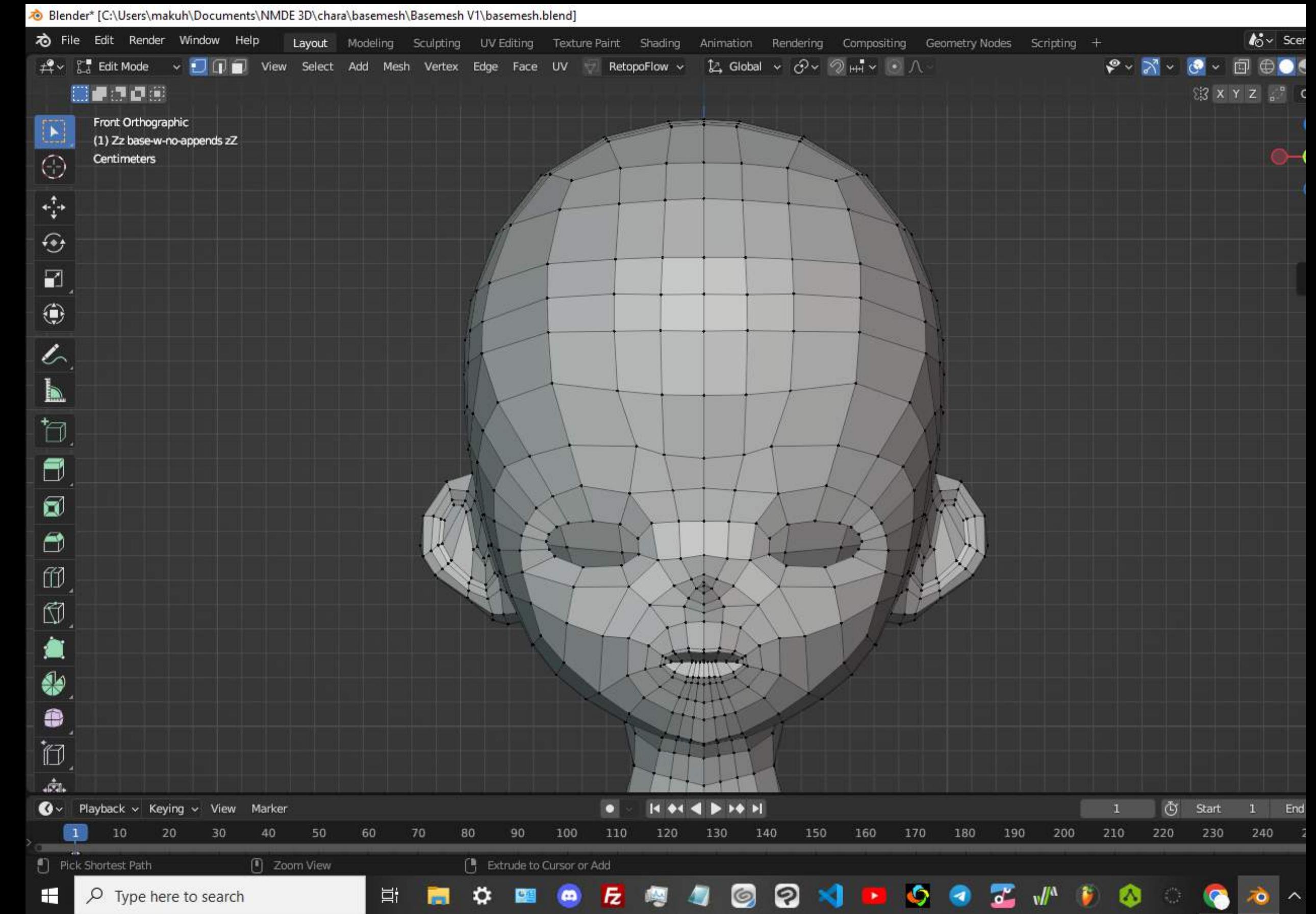
Modeling V : Basemesh Part 4

With the appendages out of the way, the base mesh is about done. From here, I decided to start adding seams to the mesh and began to plan how I would change the face normals to get an anime kind of face lighting style



Modeling VI : Normal Editing Part 1

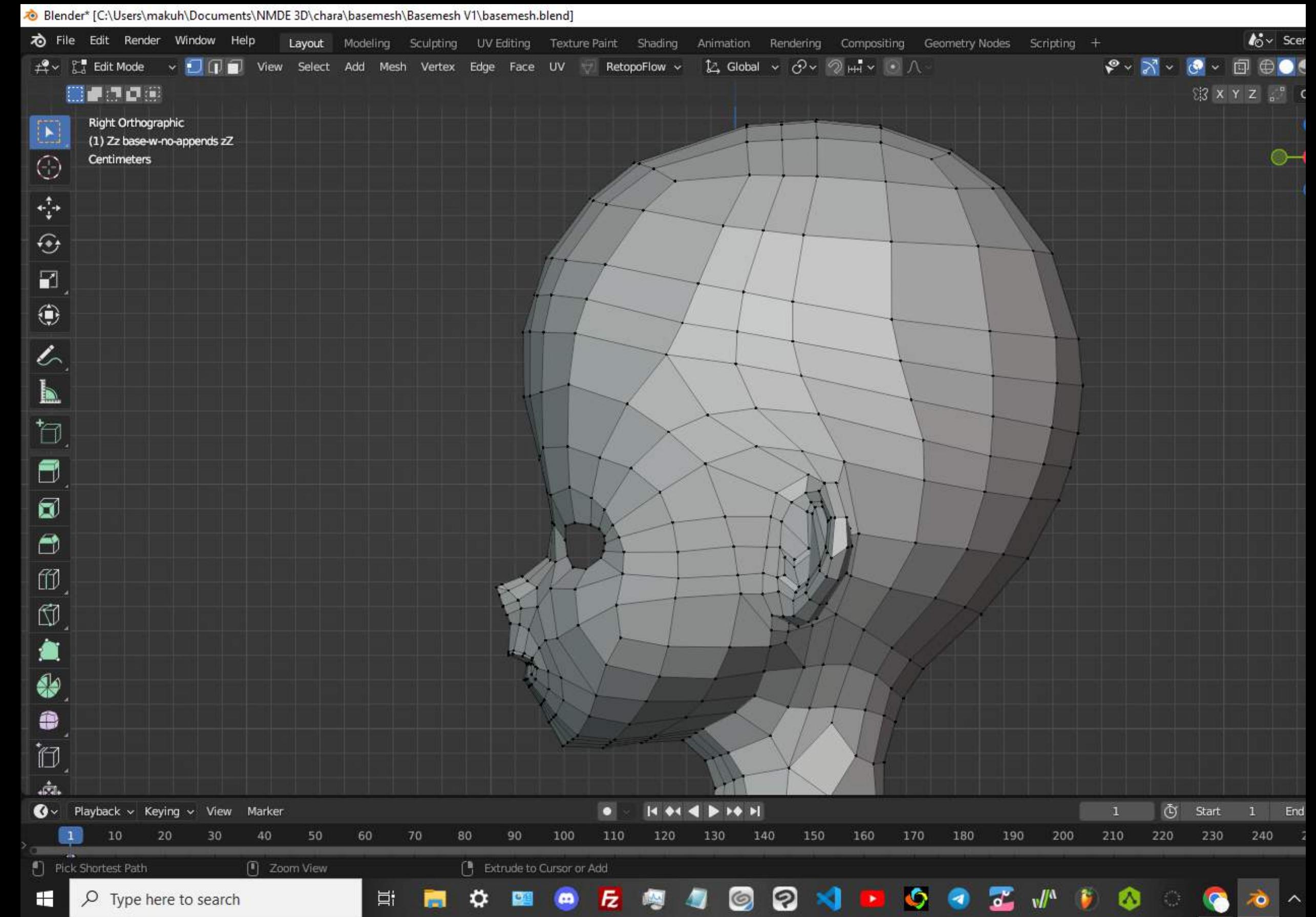
Normals are basically the direction that each face on a given mesh is facing. On normal 3D objects in the real world, you'd have to move an actual object to get a face of it looking in a direction you want, but in 3d modeling, this is different



Modeling VII : Normal Editing Part 2

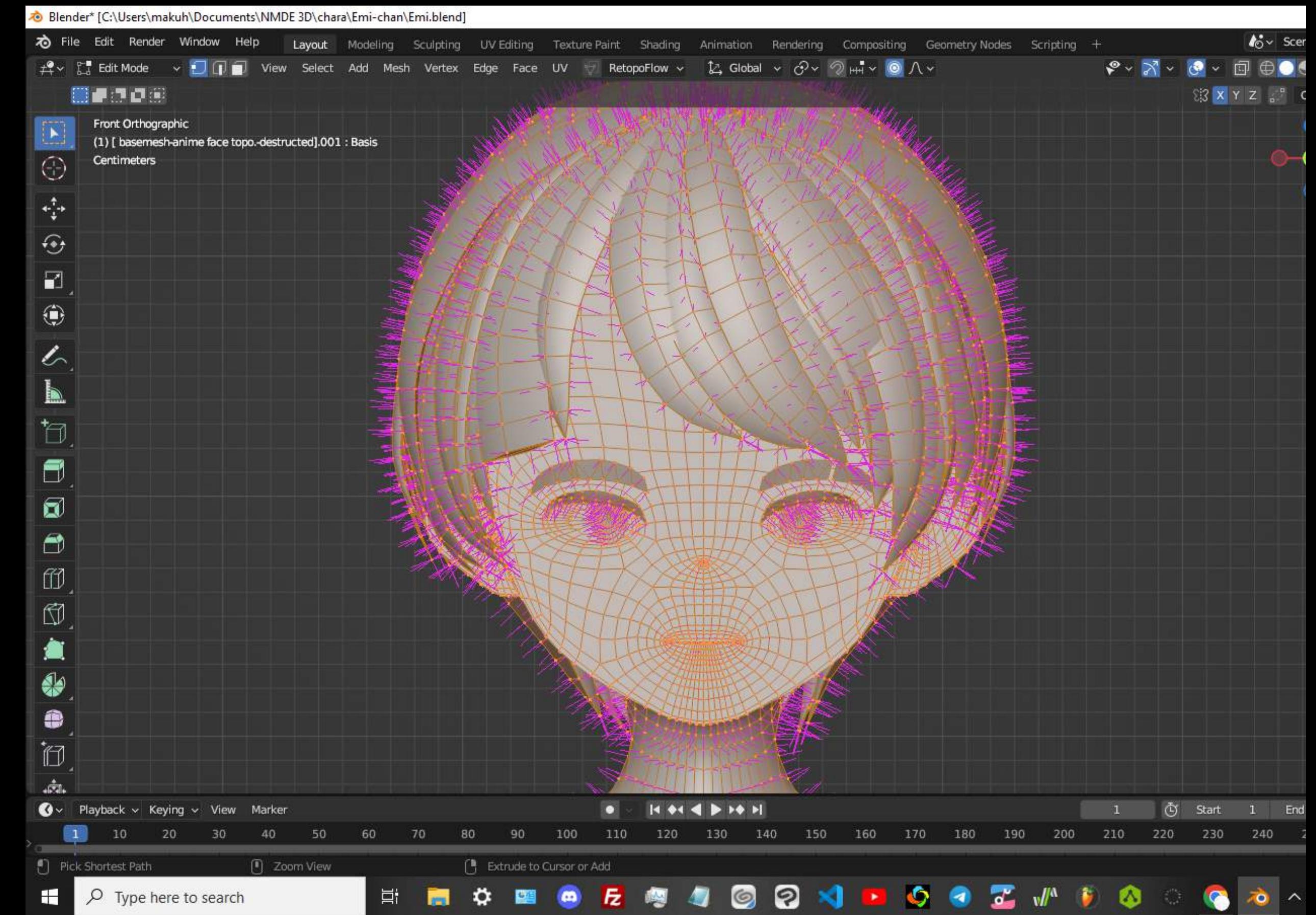
I basically wanted the normals of the head to all face forward

I also decided to move on and add more personality to the model via hair, eyes, etc.



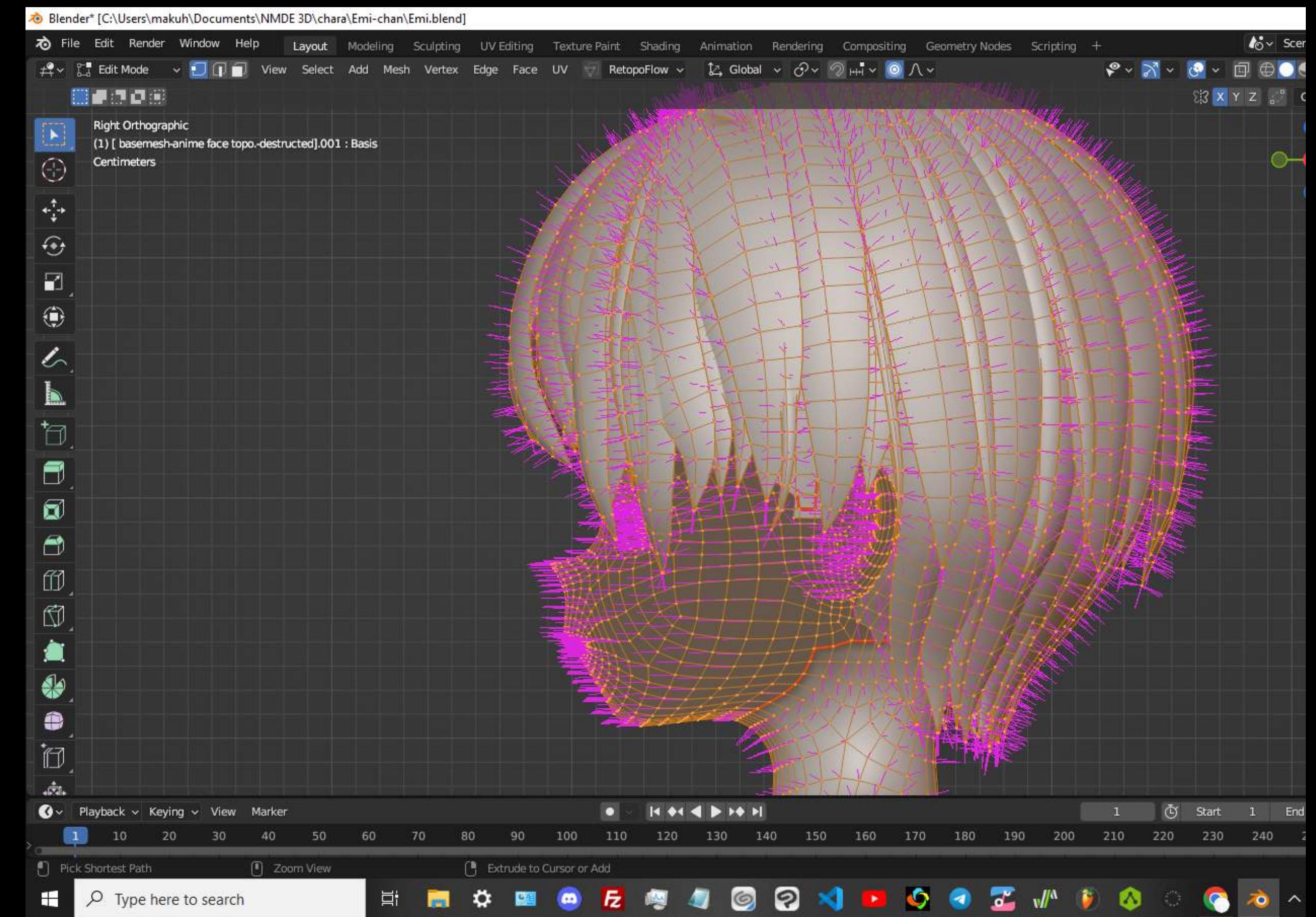
Modeling VIII : Normal Editing Part 3

I added extra facial details and began to edit the face normals. They're represented by the purple lines



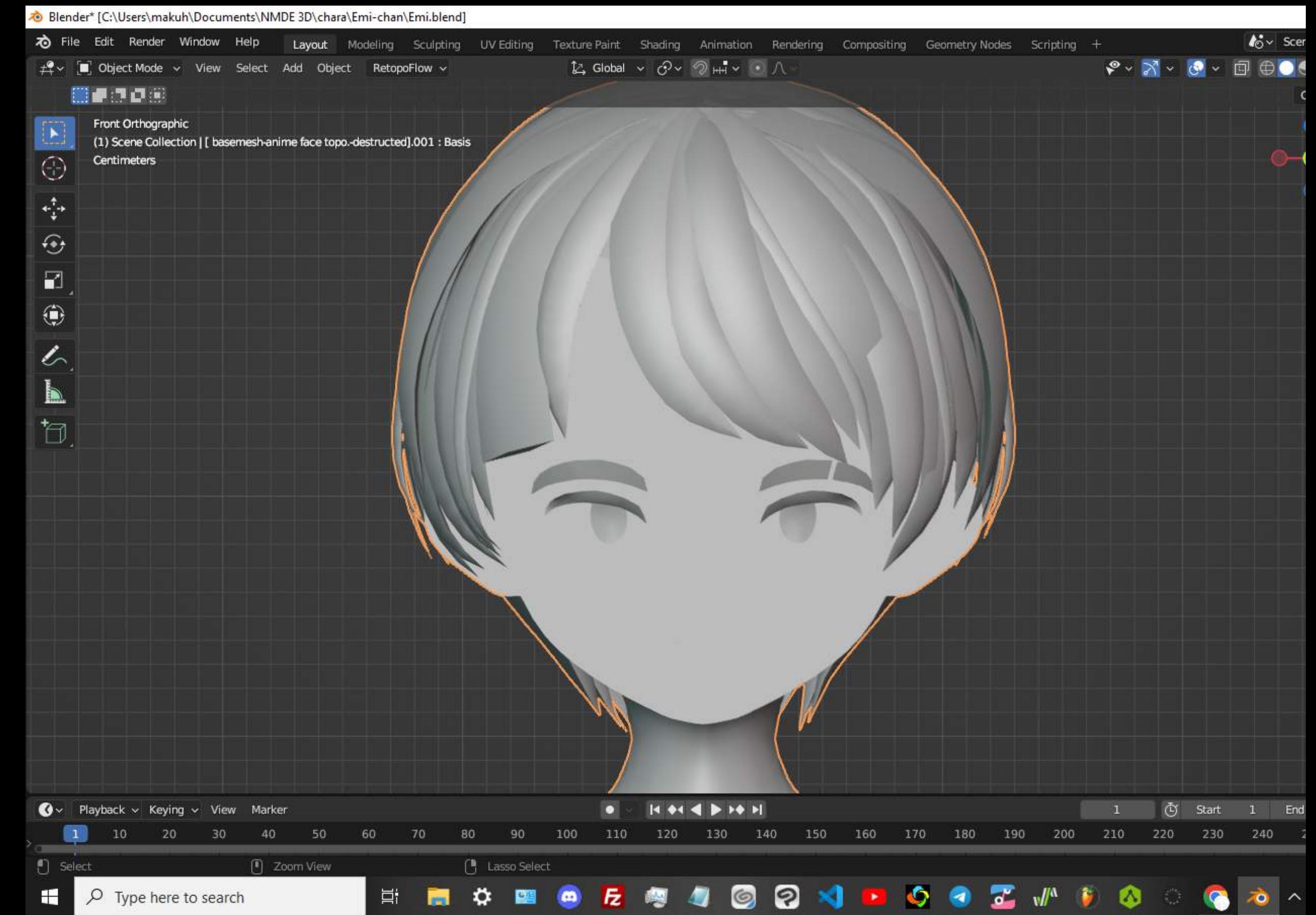
Modeling IX : Normal Editing Part 4

Here all of the normals of the face are facing forwards, resulting in a weird shadow from the side



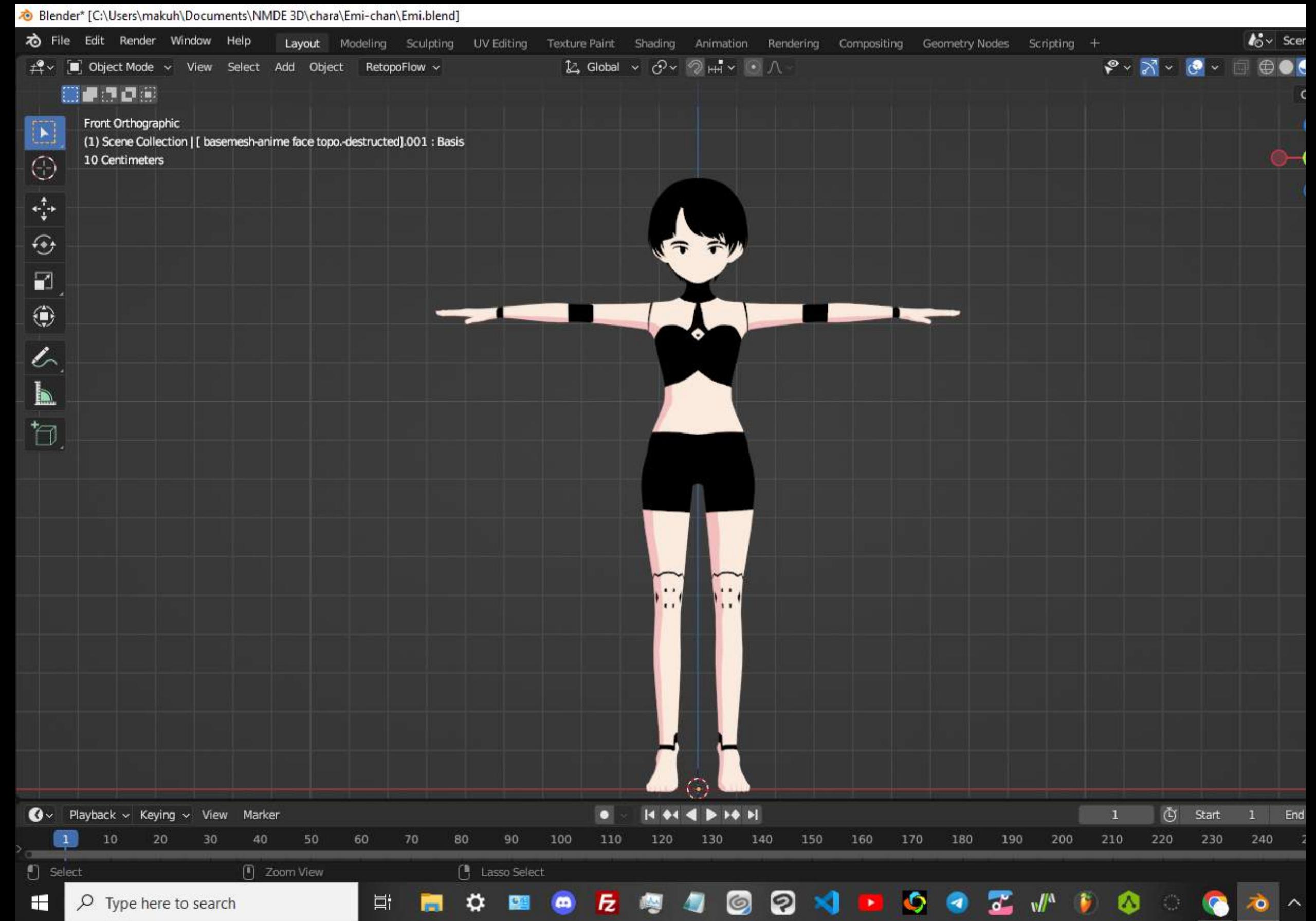
Modeling X : Normal Editing Part 5

The end result is this kind of look. The face looks undetailed right now, but when I apply materials and lighting, the shading effect I was looking for will occur



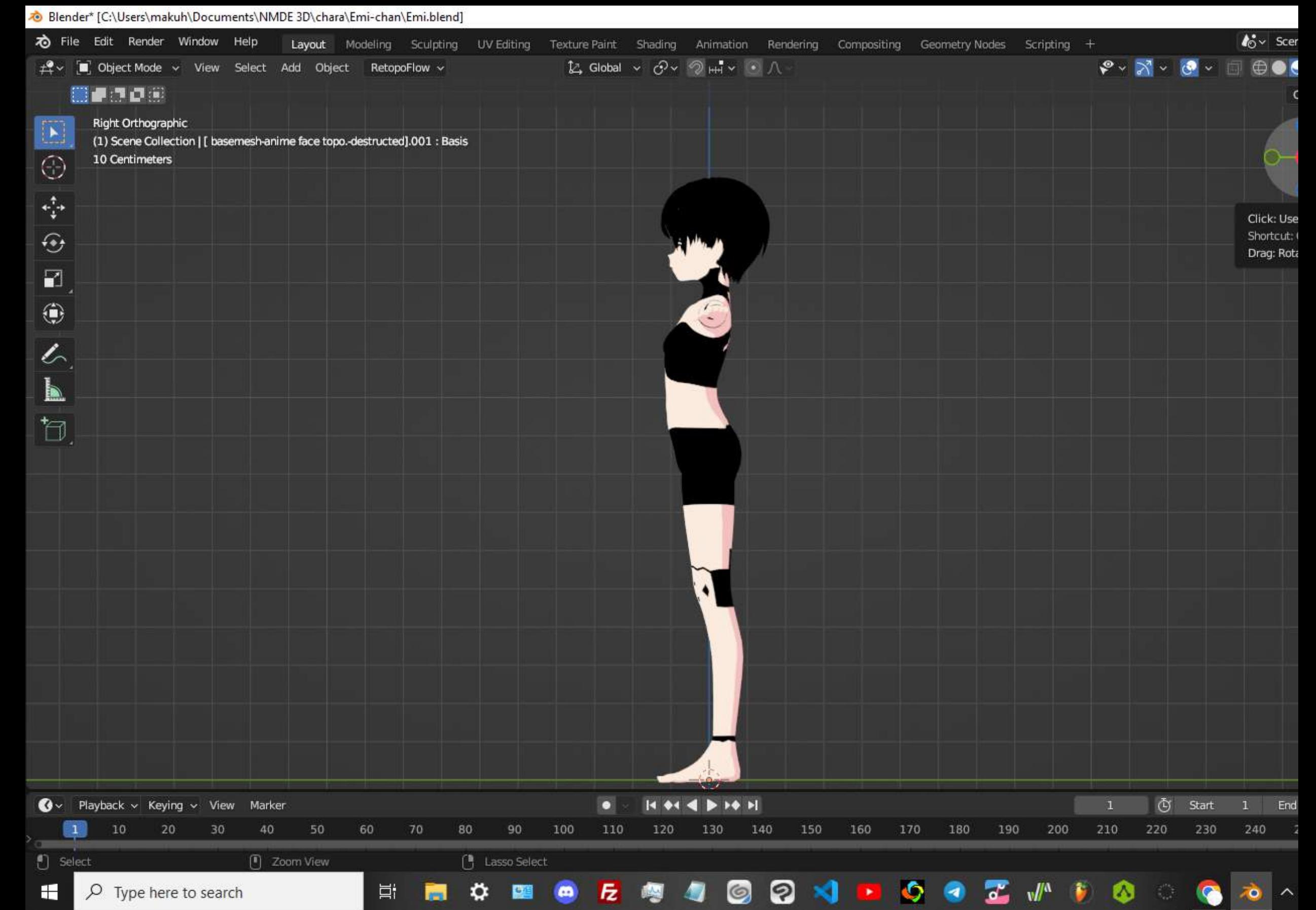
Modeling XI : Materials

I originally wanted the material to be a kind of skin color-based texture, but I later found out that doing so was a mistake for exporting...



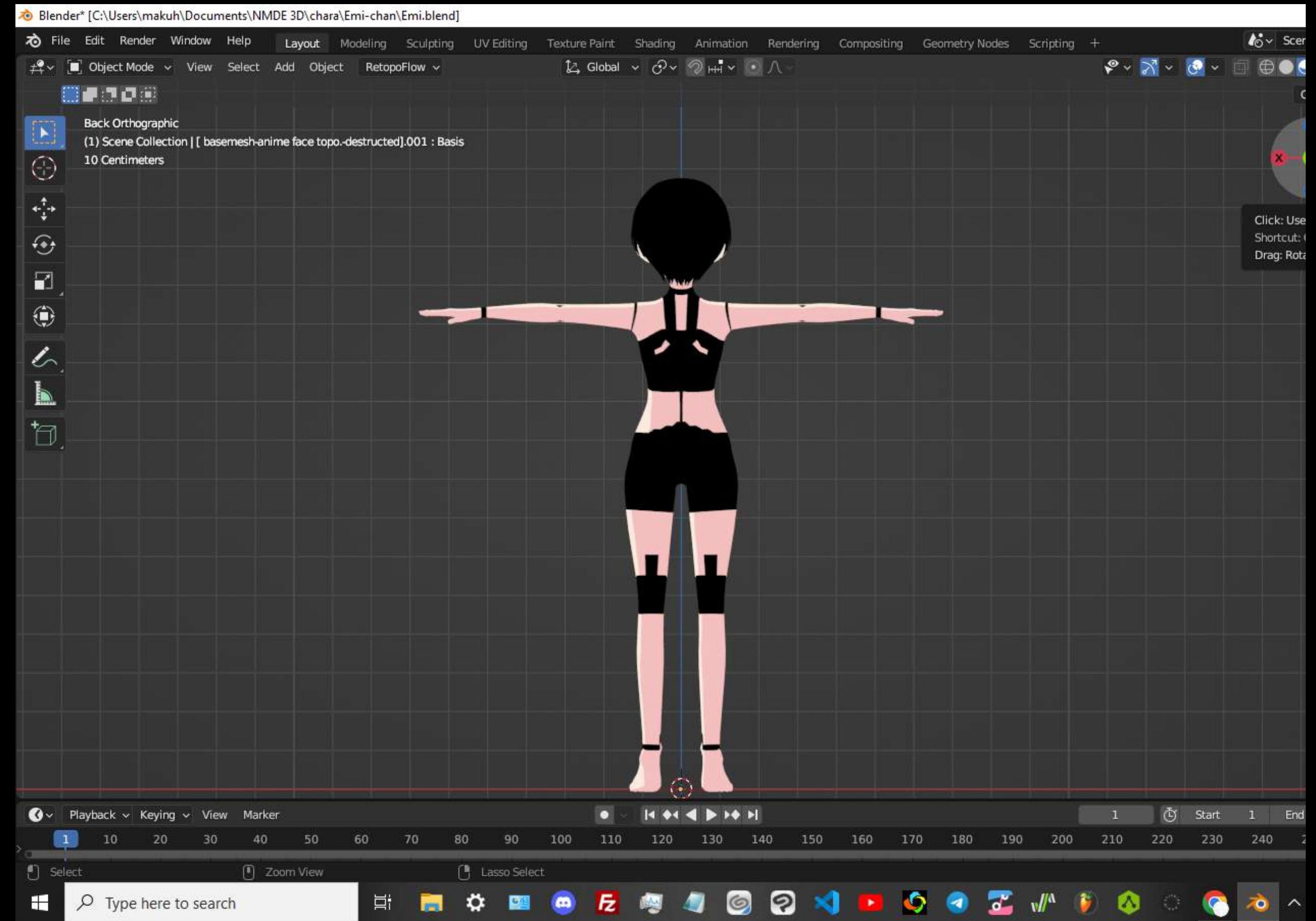
Modeling XI : Materials Cont.

Just a side view of the character



Modeling XI : Materials Cont.

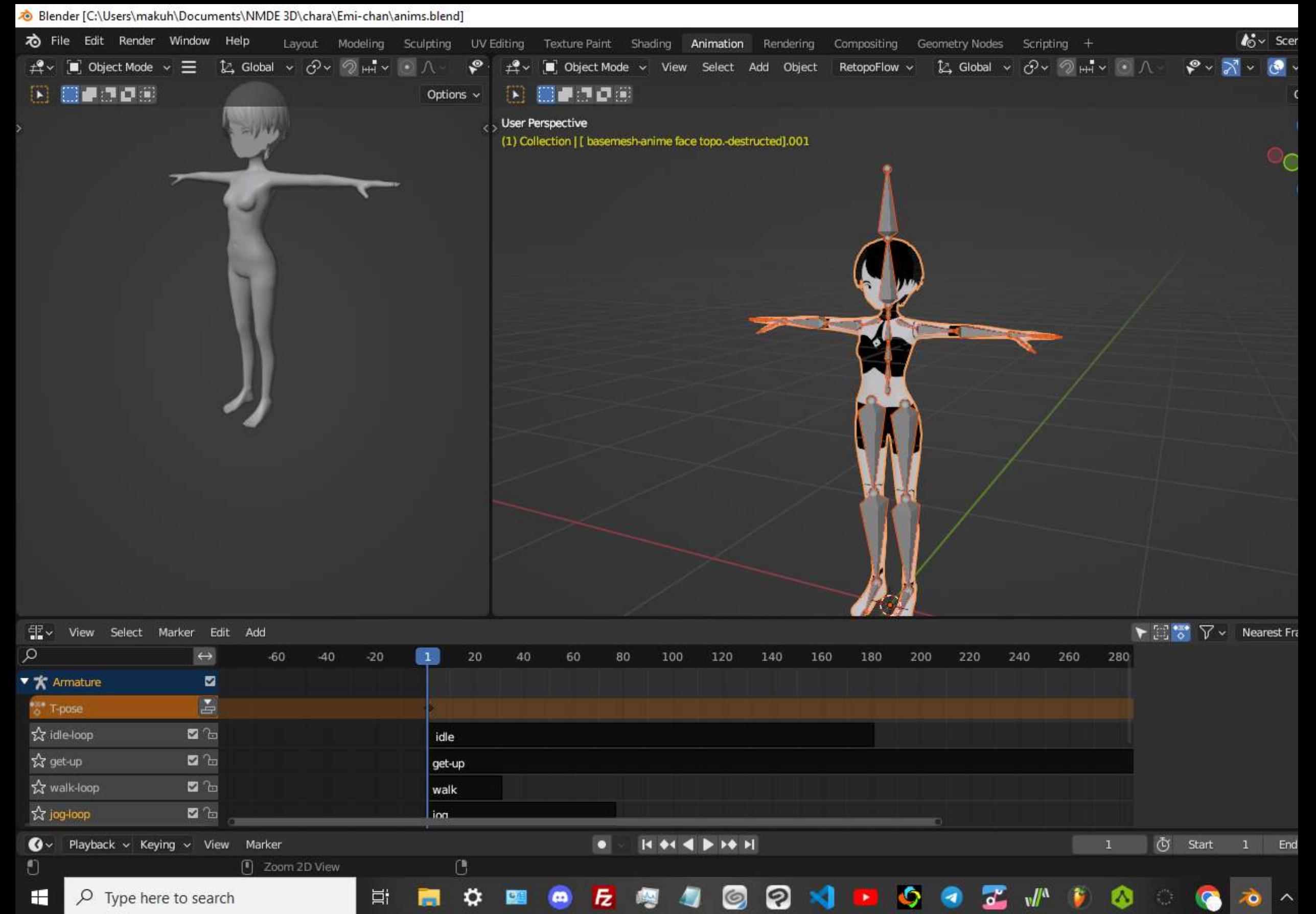
Back view



Modeling XII : Animations

After the character was completed, I went to Mixamo and downloaded some animations for the model. I then went back into Blender and loaded the animations into the model's data

I also just switched back to the basic grey material here. It's more robot-like anyways

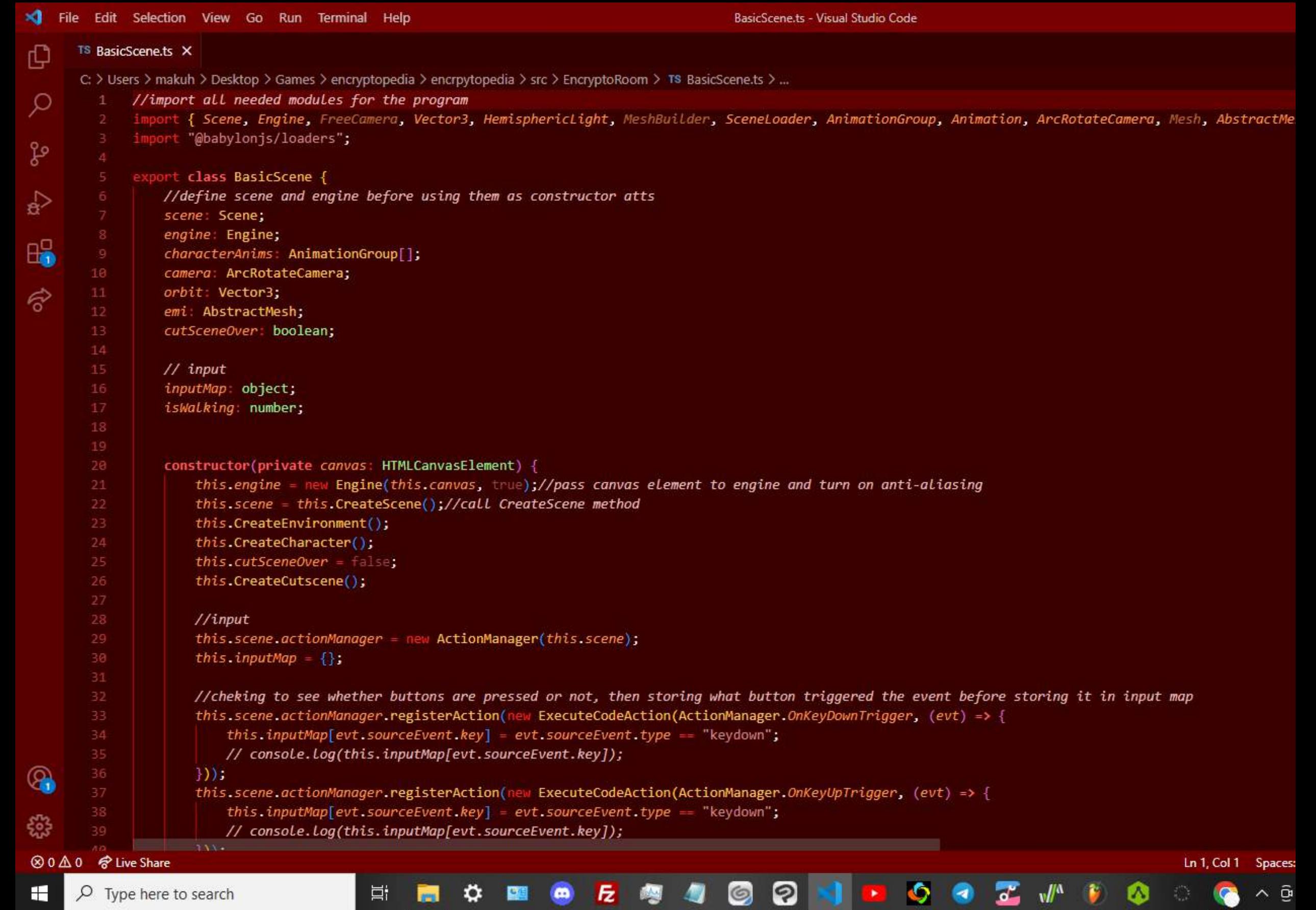


Programming I : Babylon JS and TypeScript “Fun”

So I decided that it'd be fun to try making the game as a Vue JS web application built in the Babylon JS framework. It was for a bit until I began to run into problems that warranted more time to solve than they were worth

I still got pretty far in the JS approach though, I figure I'd still show that process

I started by first importing all necessary packages and utilities from their respective libraries and with npm(nodes package manager)



```

//import all needed modules for the program
import { Scene, Engine, FreeCamera, Vector3, HemisphericLight, MeshBuilder, SceneLoader, AnimationGroup, Animation, ArcRotateCamera, Mesh, AbstractMesh } from "@babylonjs/loaders";
import { ExecuteCodeAction } from "babylonjs";

export class BasicScene {
    //define scene and engine before using them as constructor attrs
    scene: Scene;
    engine: Engine;
    characterAnimations: AnimationGroup[];
    camera: ArcRotateCamera;
    orbit: Vector3;
    emi: AbstractMesh;
    cutSceneOver: boolean;

    // input
    inputMap: object;
    isWalking: number;

    constructor(private canvas: HTMLCanvasElement) {
        this.engine = new Engine(this.canvas, true); //pass canvas element to engine and turn on anti-aliasing
        this.scene = this.CreateScene(); //call CreateScene method
        this.CreateEnvironment();
        this.CreateCharacter();
        this.cutSceneOver = false;
        this.CreateCutscene();

        //input
        this.scene.actionManager = new ActionManager(this.scene);
        this.inputMap = {};

        //cheching to see whether buttons are pressed or not, then storing what button triggered the event before storing it in input map
        this.scene.actionManager.registerAction(new ExecuteCodeAction(ActionManager.OnKeyDownTrigger, (evt) => {
            this.inputMap[evt.sourceEvent.key] = evt.sourceEvent.type == "keydown";
            // console.log(this.inputMap[evt.sourceEvent.key]);
        }));
        this.scene.actionManager.registerAction(new ExecuteCodeAction(ActionManager.OnKeyUpTrigger, (evt) => {
            this.inputMap[evt.sourceEvent.key] = evt.sourceEvent.type == "keydown";
            // console.log(this.inputMap[evt.sourceEvent.key]);
        }));
    }
}

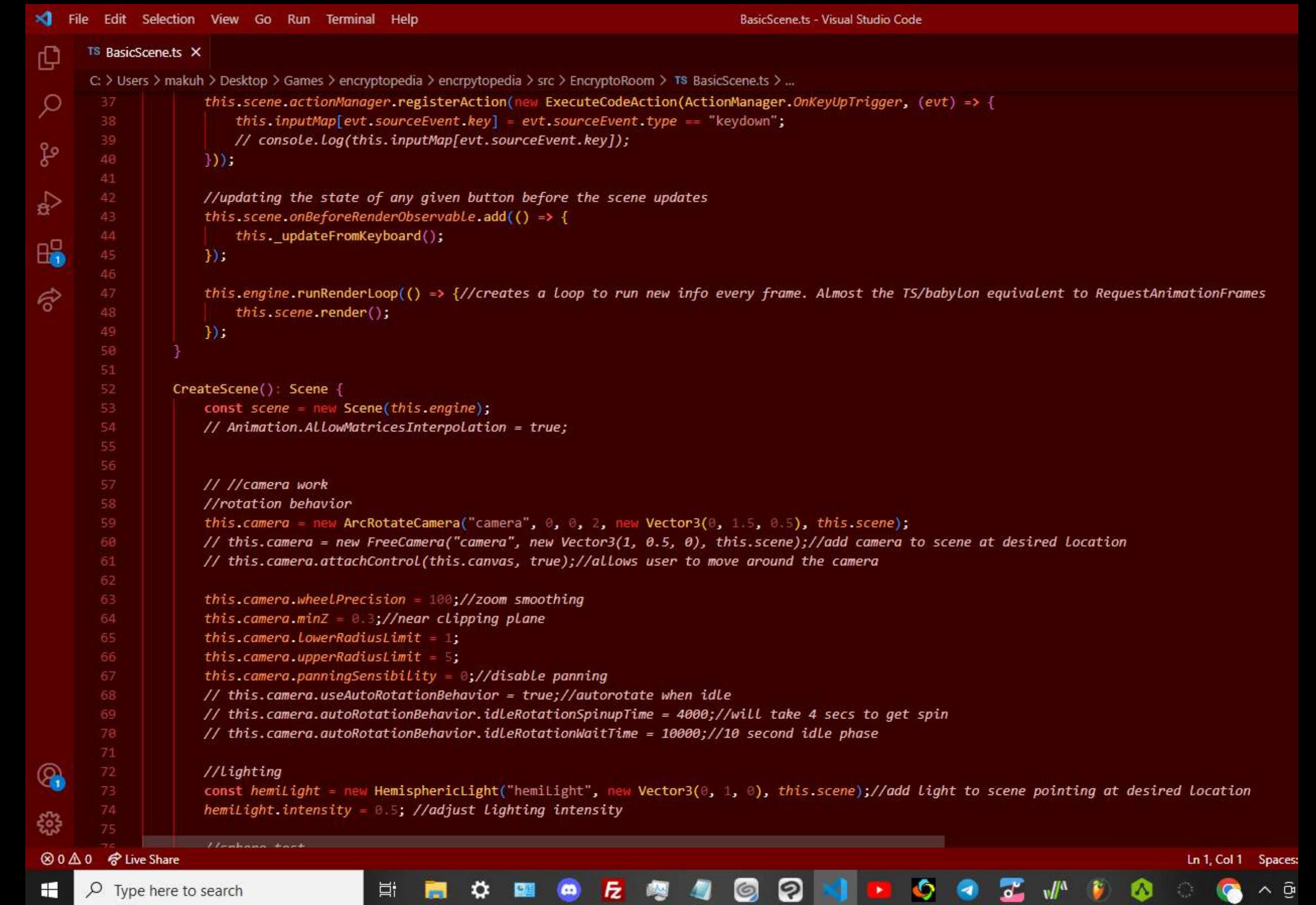
```

Programming II : Constructor and Methods

After creating a constructor that made the camera, engine, and renderer for the app, I began creating some important methods for the overall scene class

Create scene does what it says for the most part, but also instantiates the camera I declared in the constructor and the lighting of the environment

I made an event listener for keyboard presses so that the app could know when the user was pressing a key. This would be useful later on for WASD movement and other key controls



```

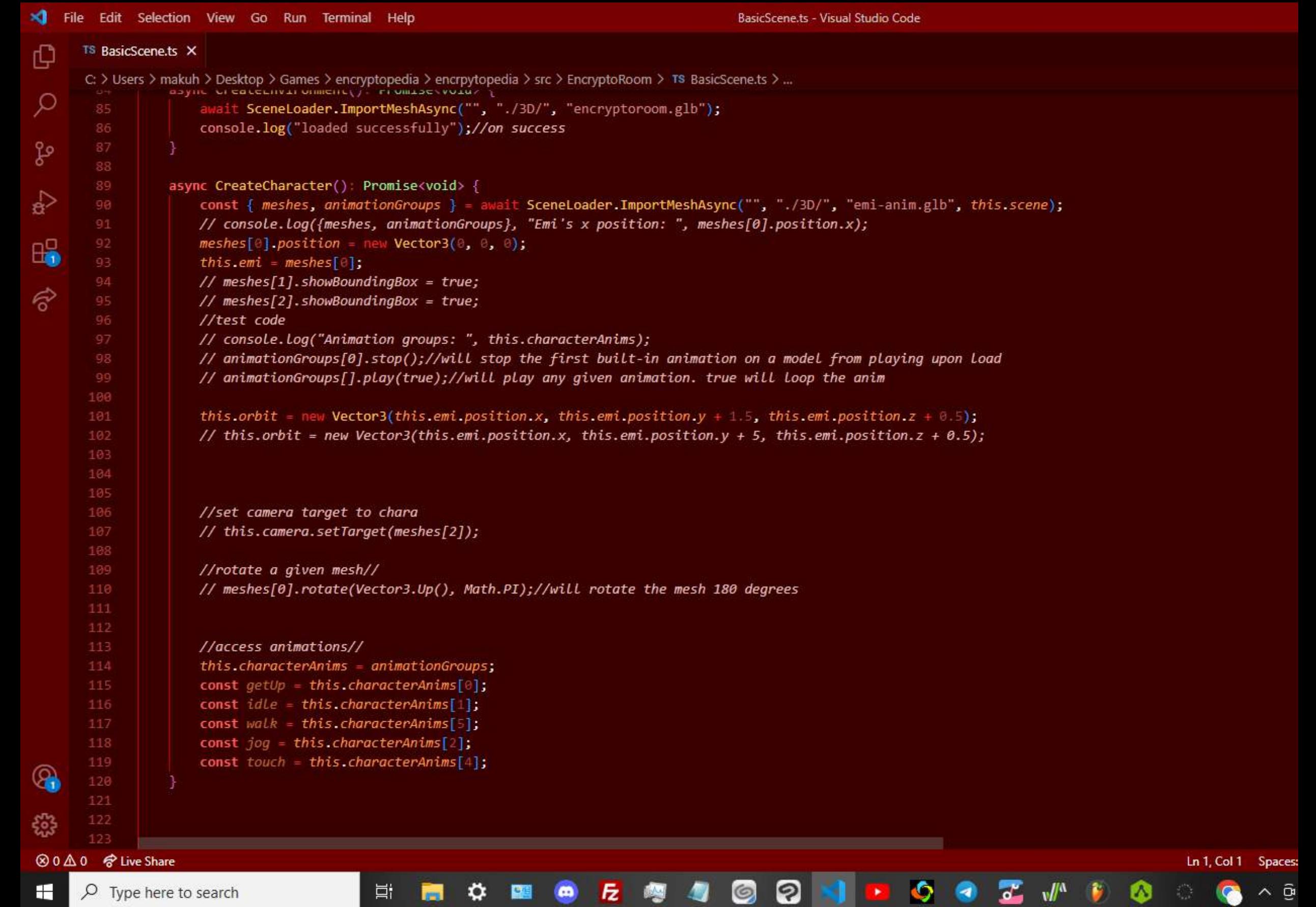
File Edit Selection View Go Run Terminal Help
BasicScene.ts - Visual Studio Code
C: > Users > makuh > Desktop > Games > encryptopedia > src > CryptoRoom > TS BasicScene.ts > ...
37   this.scene.actionManager.registerAction(new ExecuteCodeAction(ActionManager.OnKeyUpTrigger, (evt) => {
38     this.inputMap[evt.sourceEvent.key] = evt.sourceEvent.type == "keydown";
39     // console.log(this.inputMap[evt.sourceEvent.key]);
40   }));
41
42   //updating the state of any given button before the scene updates
43   this.scene.onBeforeRenderObservable.add(() => {
44     this._updateFromKeyboard();
45   });
46
47   this.engine.runRenderLoop(() => {//creates a Loop to run new info every frame. Almost the TS/babylon equivalent to RequestAnimationFrame
48     this.scene.render();
49   });
50
51
52   CreateScene(): Scene {
53     const scene = new Scene(this.engine);
54     // Animation.AllowMatricesInterpolation = true;
55
56
57     // //camera work
58     //rotation behavior
59     this.camera = new ArcRotateCamera("camera", 0, 0, 2, new Vector3(0, 1.5, 0.5), this.scene);
60     // this.camera = new FreeCamera("camera", new Vector3(1, 0.5, 0), this.scene); //add camera to scene at desired location
61     // this.camera.attachControl(this.canvas); //allows user to move around the camera
62
63     this.camera.wheelPrecision = 100; //zoom smoothing
64     this.camera.minZ = 0.3; //near clipping plane
65     this.camera.lowerRadiusLimit = 1;
66     this.camera.upperRadiusLimit = 5;
67     this.camera.panningSensibility = 0; //disable panning
68     // this.camera.useAutoRotationBehavior = true; //autorotate when idle
69     // this.camera.autoRotationBehavior.idleRotationSpinupTime = 4000; //will take 4 secs to get spin
69     // this.camera.autoRotationBehavior.idleRotationWaitTime = 10000; //10 second idle phase
70
71
72     //lighting
73     const hemiLight = new HemisphericLight("hemiLight", new Vector3(0, 1, 0), this.scene); //add light to scene pointing at desired location
74     hemiLight.intensity = 0.5; //adjust lighting intensity
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

Programming III : Loading the Model

I then spent the next couple of days trying to figure out how to load my 3D model into the app. After some tinkering and resaving the model as a glb file instead of an fbx one, I managed to get it working...

I loaded the model asynchronously, so that the model would only show up after everything else was loaded. This prevents weird events like the user walking around in space while waiting for the room to load



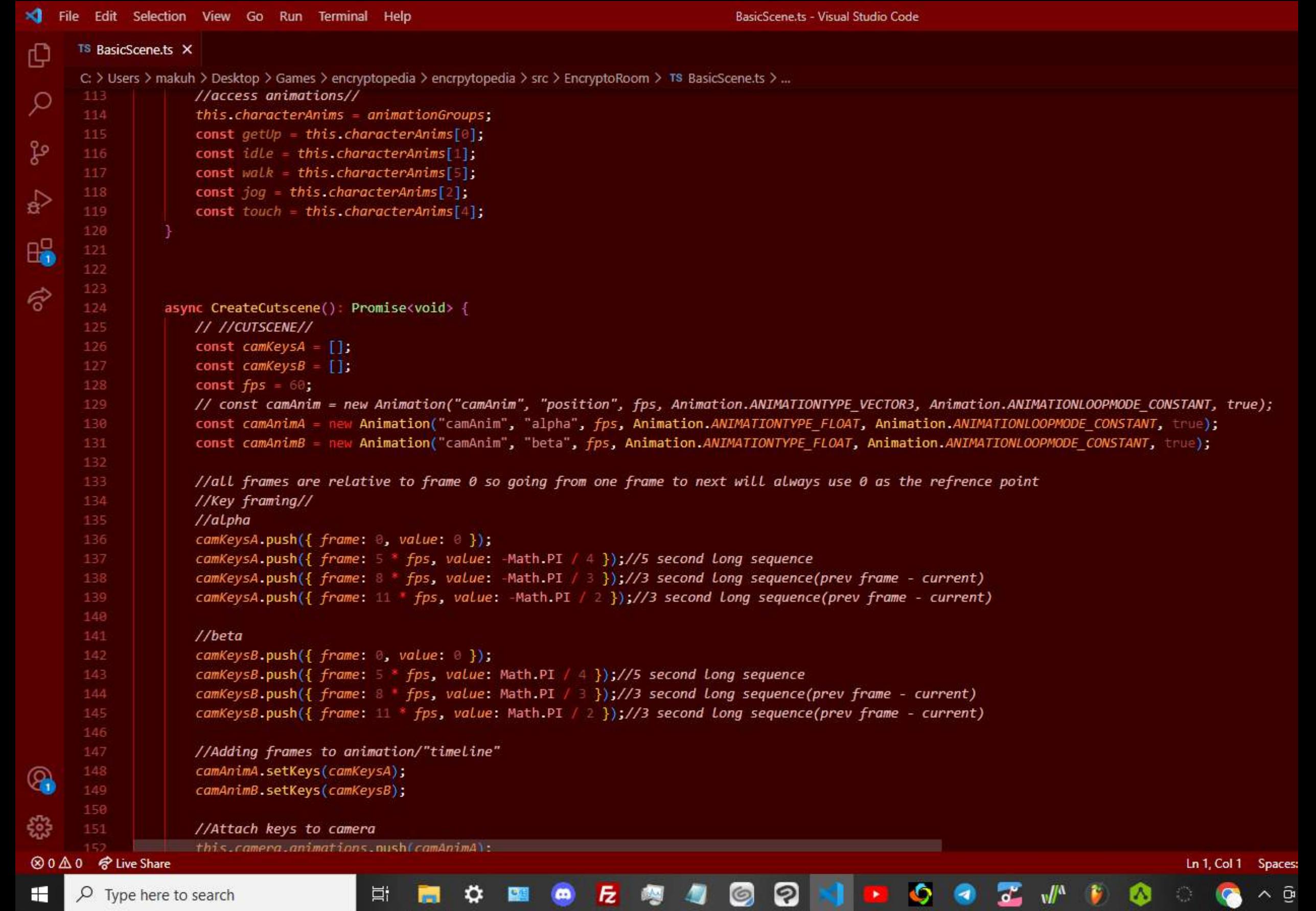
```

TS BasicScene.ts X
C:\Users\makuh\Desktop\Games\encyclopedia\src\EncryptionRoom> TS BasicScene.ts > ...
85     await SceneLoader.ImportMeshAsync("", "./3D/", "encrytoroom.glb");
86     console.log("loaded successfully");//on success
87 }
88
89 async CreateCharacter(): Promise<void> {
90     const { meshes, animationGroups } = await SceneLoader.ImportMeshAsync("", "./3D/", "emi-anime.glb", this.scene);
91     // console.log({ meshes, animationGroups }, "Emi's x position: ", meshes[0].position.x);
92     meshes[0].position = new Vector3(0, 0, 0);
93     this.emi = meshes[0];
94     // meshes[1].showBoundingBox = true;
95     // meshes[2].showBoundingBox = true;
96     // test code
97     // console.log("Animation groups: ", this.characterAnims);
98     // animationGroups[0].stop(); // will stop the first built-in animation on a model from playing upon Load
99     // animationGroups[].play(true); // will play any given animation. true will loop the anim
100
101    this.orbit = new Vector3(this.emi.position.x, this.emi.position.y + 1.5, this.emi.position.z + 0.5);
102    // this.orbit = new Vector3(this.emi.position.x, this.emi.position.y + 5, this.emi.position.z + 0.5);
103
104
105    // set camera target to chara
106    // this.camera.setTarget(meshes[2]);
107
108    // rotate a given mesh//
109    // meshes[0].rotate(Vector3.Up(), Math.PI); // will rotate the mesh 180 degrees
110
111
112    // access animations//
113    this.characterAnims = animationGroups;
114    const getUp = this.characterAnims[0];
115    const idle = this.characterAnims[1];
116    const walk = this.characterAnims[5];
117    const jog = this.characterAnims[2];
118    const touch = this.characterAnims[4];
119
120 }
121
122
123
Ln 1, Col 1 Spaces:

```

Programming IV : Miscellaneous Methods

I started making methods for a bunch of other stuff like a cutscene that starts when the game is loaded up, a key board key identifier, and a cutscene ender



```

File Edit Selection View Go Run Terminal Help
BasicScene.ts - Visual Studio Code
C: > Users > makuh > Desktop > Games > encryptopedia > src > CryptoRoom > TS BasicScene.ts > ...
113 //access animations//
114 this.characterAnims = animationGroups;
115 const getUp = this.characterAnims[0];
116 const idle = this.characterAnims[1];
117 const walk = this.characterAnims[5];
118 const jog = this.characterAnims[2];
119 const touch = this.characterAnims[4];
120 }
121
122
123
124 async CreateCutscene(): Promise<void> {
125 // //CUTSCENE//
126 const camKeysA = [];
127 const camKeysB = [];
128 const fps = 60;
129 // const camAnim = new Animation("camAnim", "position", fps, Animation.ANIMATIONTYPE_VECTOR3, Animation.ANIMATIONLOOPMODE_CONSTANT, true);
130 const camAnimA = new Animation("camAnim", "alpha", fps, Animation.ANIMATIONTYPE_FLOAT, Animation.ANIMATIONLOOPMODE_CONSTANT, true);
131 const camAnimB = new Animation("camAnim", "beta", fps, Animation.ANIMATIONTYPE_FLOAT, Animation.ANIMATIONLOOPMODE_CONSTANT, true);
132
133 //all frames are relative to frame 0 so going from one frame to next will always use 0 as the refrence point
134 //Key framing//
135 //alpha
136 camKeysA.push({ frame: 0, value: 0 });
137 camKeysA.push({ frame: 5 * fps, value: -Math.PI / 4 });//5 second Long sequence
138 camKeysA.push({ frame: 8 * fps, value: -Math.PI / 3 });//3 second Long sequence(prev frame - current)
139 camKeysA.push({ frame: 11 * fps, value: -Math.PI / 2 });//3 second Long sequence(prev frame - current)
140
141 //beta
142 camKeysB.push({ frame: 0, value: 0 });
143 camKeysB.push({ frame: 5 * fps, value: Math.PI / 4 });//5 second Long sequence
144 camKeysB.push({ frame: 8 * fps, value: Math.PI / 3 });//3 second Long sequence(prev frame - current)
145 camKeysB.push({ frame: 11 * fps, value: Math.PI / 2 });//3 second Long sequence(prev frame - current)
146
147 //Adding frames to animation/"timeline"
148 camAnimA.setKeys(camKeysA);
149 camAnimB.setKeys(camKeysB);
150
151 //Attach keys to camera
152 this.camera.animations.push(camAnimA);

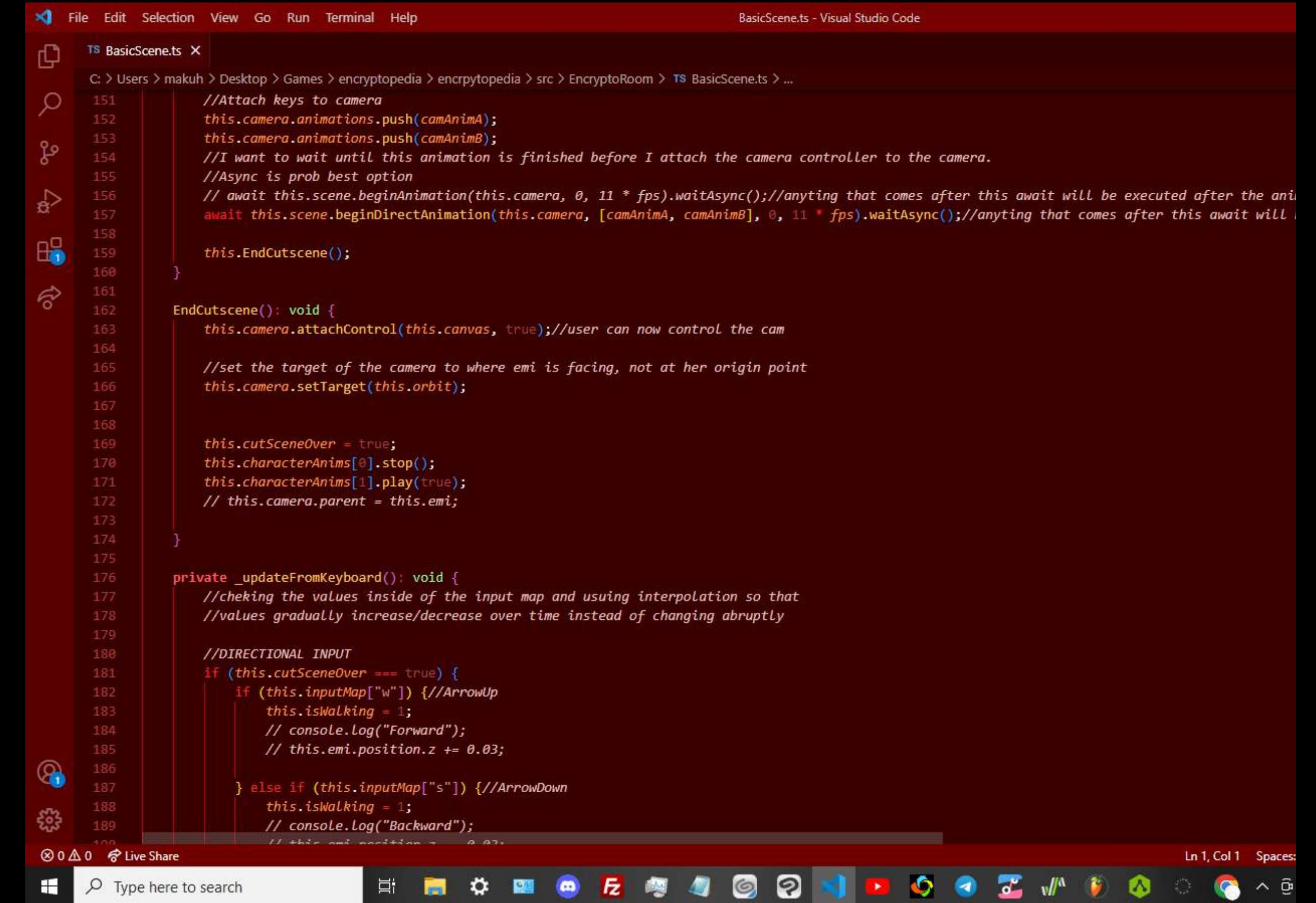
```

Programming V : Time Limits and Complex Problems

At this point, I began to run into problems with moving the character and loading the right animations for the character movement. There was also the issue of blending the animations so that they fit well together

Problems went from taking about 1-2 days to research and fix, to like a week or so. At this rate, I was going to fall behind, so I began to think of alternative programming solutions and frameworks to use

The fact that I knew nothing about game development programming beforehand also handicapped my ability to solve complex problems in a timely manner...



```

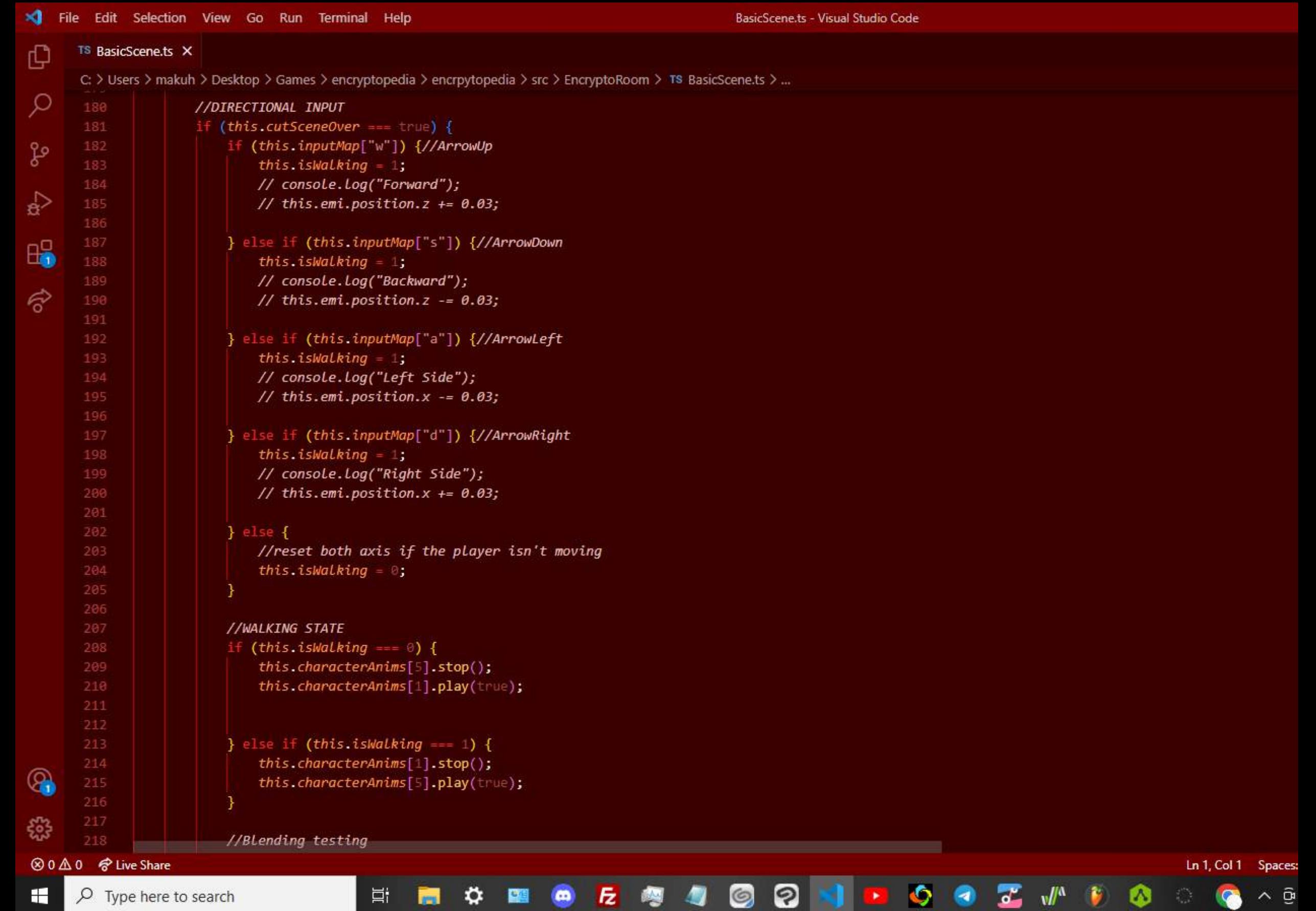
TS BasicScene.ts X
C: > Users > makuh > Desktop > Games > encryptopedia > encryptopedia > src > CryptoRoom > TS BasicScene.ts > ...
151 //Attach keys to camera
152 this.camera.animations.push(camAnimA);
153 this.camera.animations.push(camAnimB);
154 //I want to wait until this animation is finished before I attach the camera controller to the camera.
155 //Async is prob best option
156 // await this.scene.beginAnimation(this.camera, 0, 11 * fps).waitAsync();//anyting that comes after this await will be executed after the ani
157 await this.scene.beginDirectAnimation(this.camera, [camAnimA, camAnimB], 0, 11 * fps).waitAsync();//anyting that comes after this await will
158
159 this.EndCutscene();
160 }
161
162 EndCutscene(): void {
163   this.camera.attachControl(this.canvas, true); //user can now control the cam
164
165   //set the target of the camera to where emi is facing, not at her origin point
166   this.camera.setTarget(this.orbit);
167
168   this.cutSceneOver = true;
169   this.characterAnims[0].stop();
170   this.characterAnims[1].play(true);
171   // this.camera.parent = this.emi;
172
173 }
174
175 private _updateFromKeyboard(): void {
176   //cheking the values inside of the input map and ususing interpolation so that
177   //values gradually increase/decrease over time instead of changing abruptly
178
179   //DIRECTIONAL INPUT
180   if (this.cutSceneOver === true) {
181     if (this.inputMap["w"]) { //ArrowUp
182       this.isWalking = 1;
183       // console.log("Forward");
184       // this.emi.position.z += 0.03;
185
186     } else if (this.inputMap["s"]) { //ArrowDown
187       this.isWalking = 1;
188       // console.log("Backward");
189     }
190   }
}

```

Programming VI : Turning Point

The farthest I got after trying to stick out the JS approach was a working function that moved the character and played animations at the same time. After that, I got tired of not having enough time to overcome walls I ran into

I figured it'd be quicker to jump ship and start using another framework. I was still on schedule, so I could manage to do it too



```

//DIRECTIONAL INPUT
if (this.cutSceneOver === true) {
    if (this.inputMap["w"]) { //ArrowUp
        this.isWalking = 1;
        // console.log("Forward");
        // this.emi.position.z += 0.03;
    } else if (this.inputMap["s"]) { //ArrowDown
        this.isWalking = 1;
        // console.log("Backward");
        // this.emi.position.z -= 0.03;
    } else if (this.inputMap["a"]) { //ArrowLeft
        this.isWalking = 1;
        // console.log("Left Side");
        // this.emi.position.x -= 0.03;
    } else if (this.inputMap["d"]) { //ArrowRight
        this.isWalking = 1;
        // console.log("Right Side");
        // this.emi.position.x += 0.03;
    } else {
        //reset both axis if the player isn't moving
        this.isWalking = 0;
    }
}

//WALKING STATE
if (this.isWalking === 0) {
    this.characterAnims[5].stop();
    this.characterAnims[1].play(true);
}

else if (this.isWalking === 1) {
    this.characterAnims[1].stop();
    this.characterAnims[5].play(true);
}

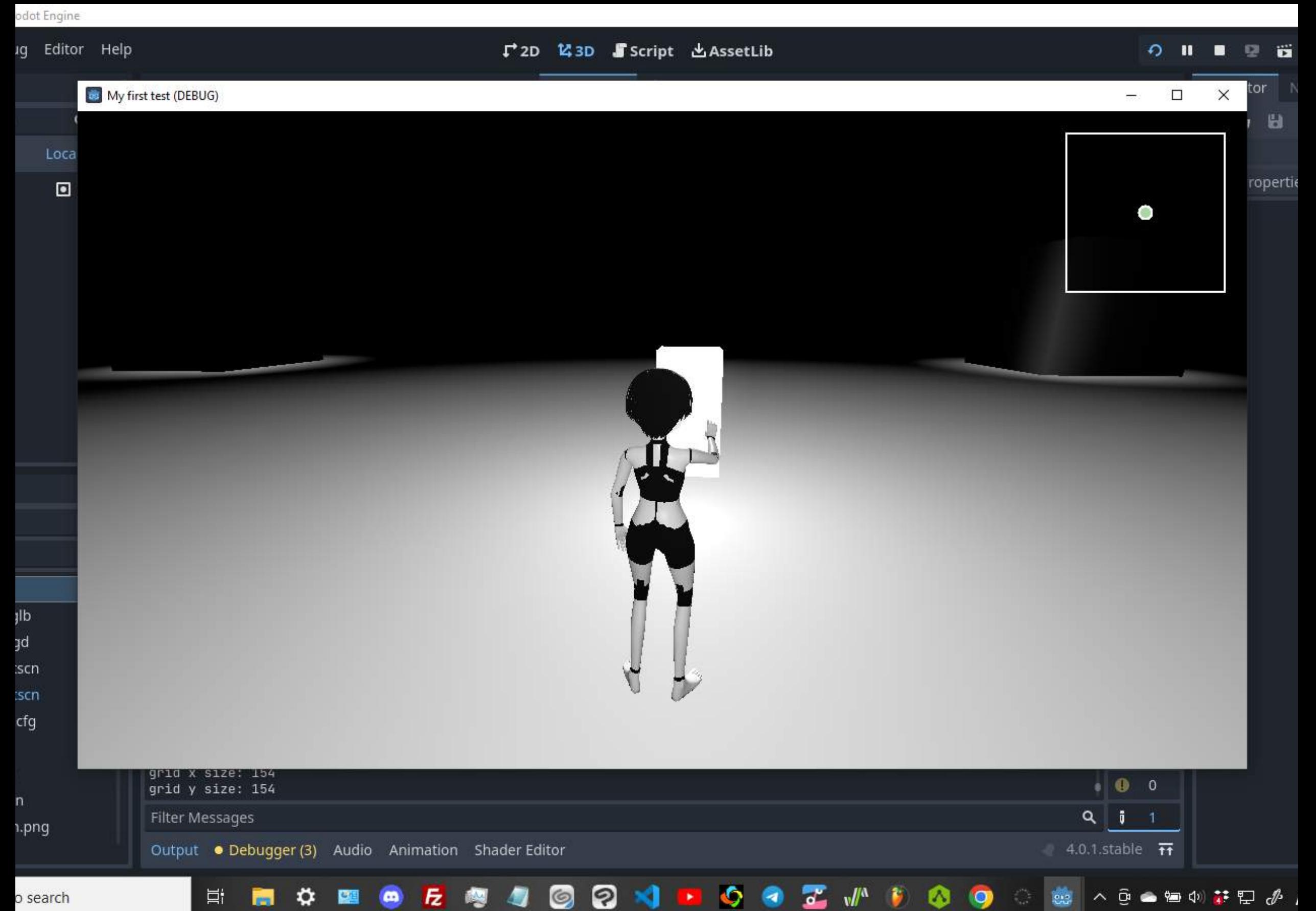
//Blending testing

```

Testing and Implementation I : Godot Engine

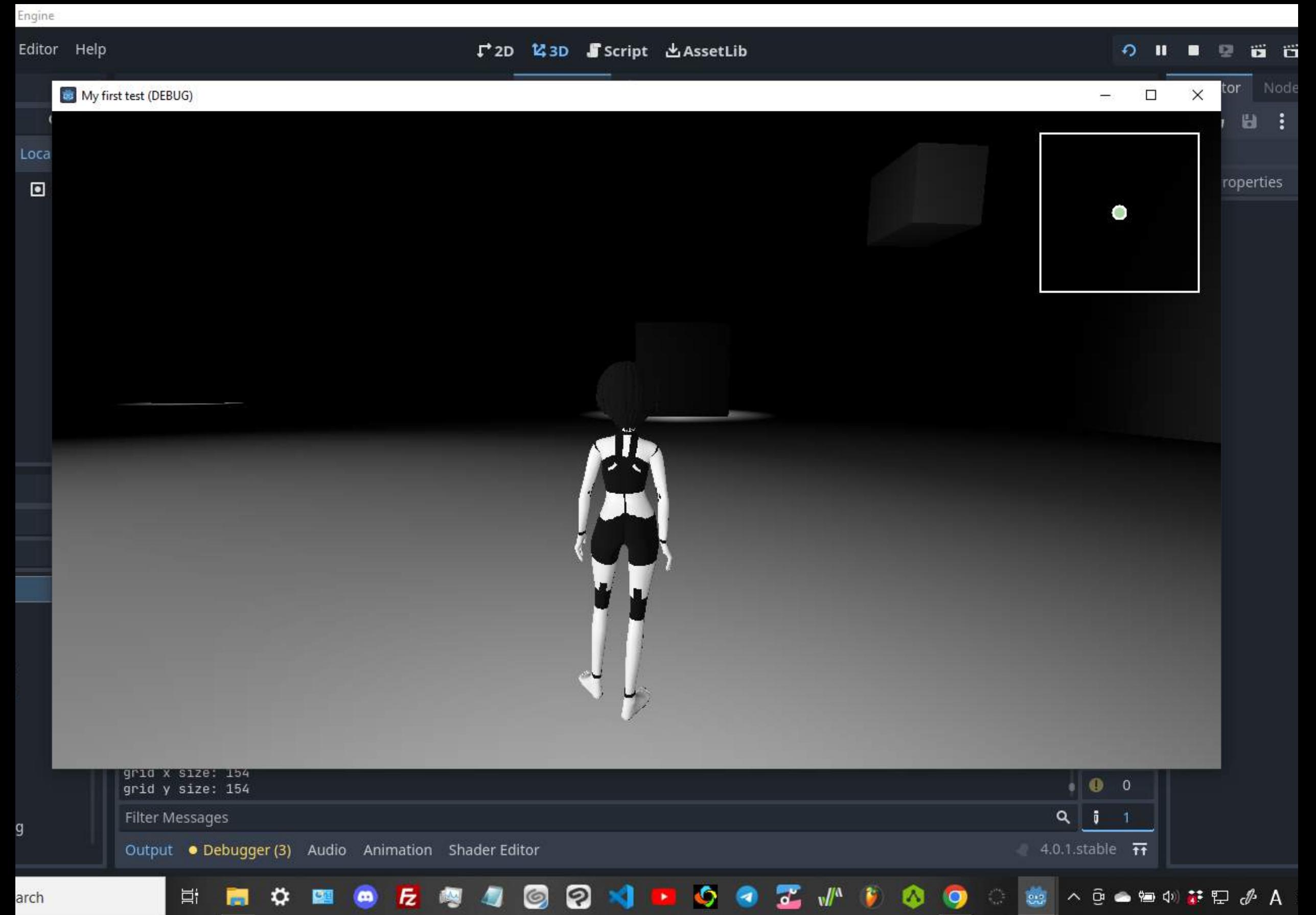
I chose to switch to using Godot 4.0, a free game development software that uses an ECS based framework much like Unity does

The process then went by extremely quickly. I was able to manage the states and animations of my character, model the entire “room”, and work on the lighting and minimap features all within Godot in a matter of days



Testing and Implementation II : Wrapping Up

After getting the basic foundation for the game done, I decided that I did enough developing and moved on to solidify my UI before bringing all of my pieces together

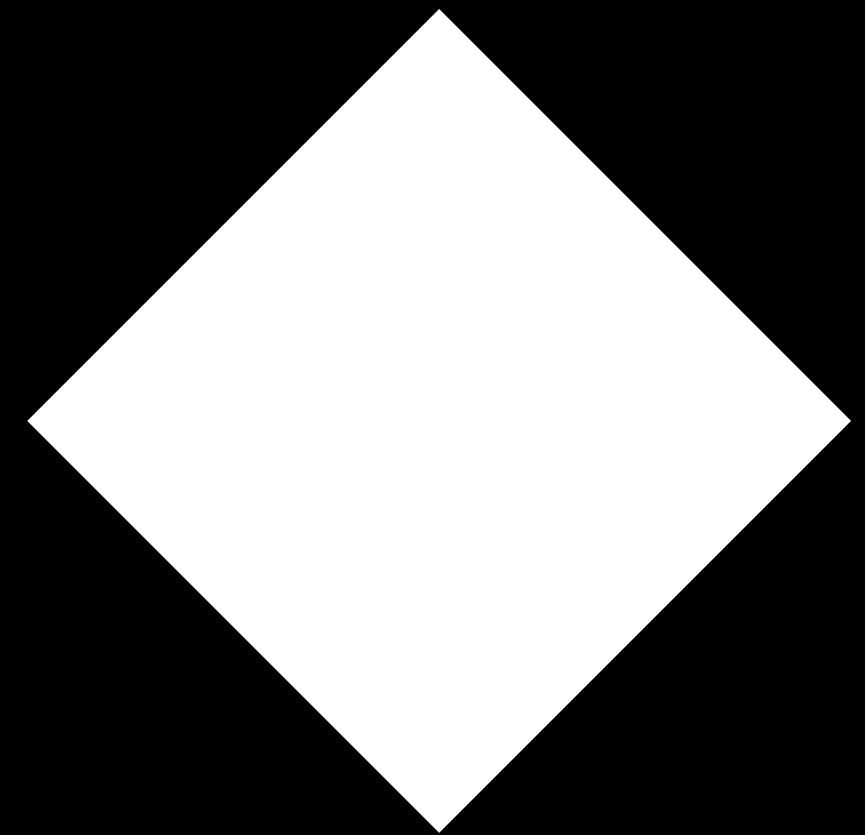




ENCRYPTOPEDIA

WIREFRAME

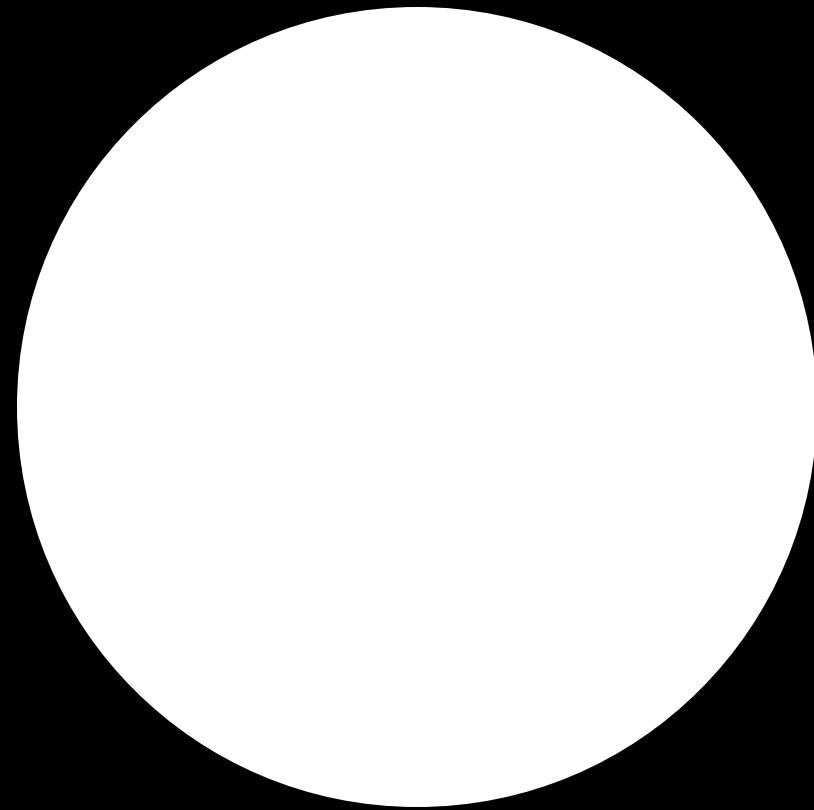
Design Choices: Shapes



Rhombus shape will be used for interactions and objects that require user influence. It also represents a “goal” or target.



Squares are used for structure and stability. Walls, physical structures, and glyphs will use this shape. The blockiness of the shape will be used in the type as well



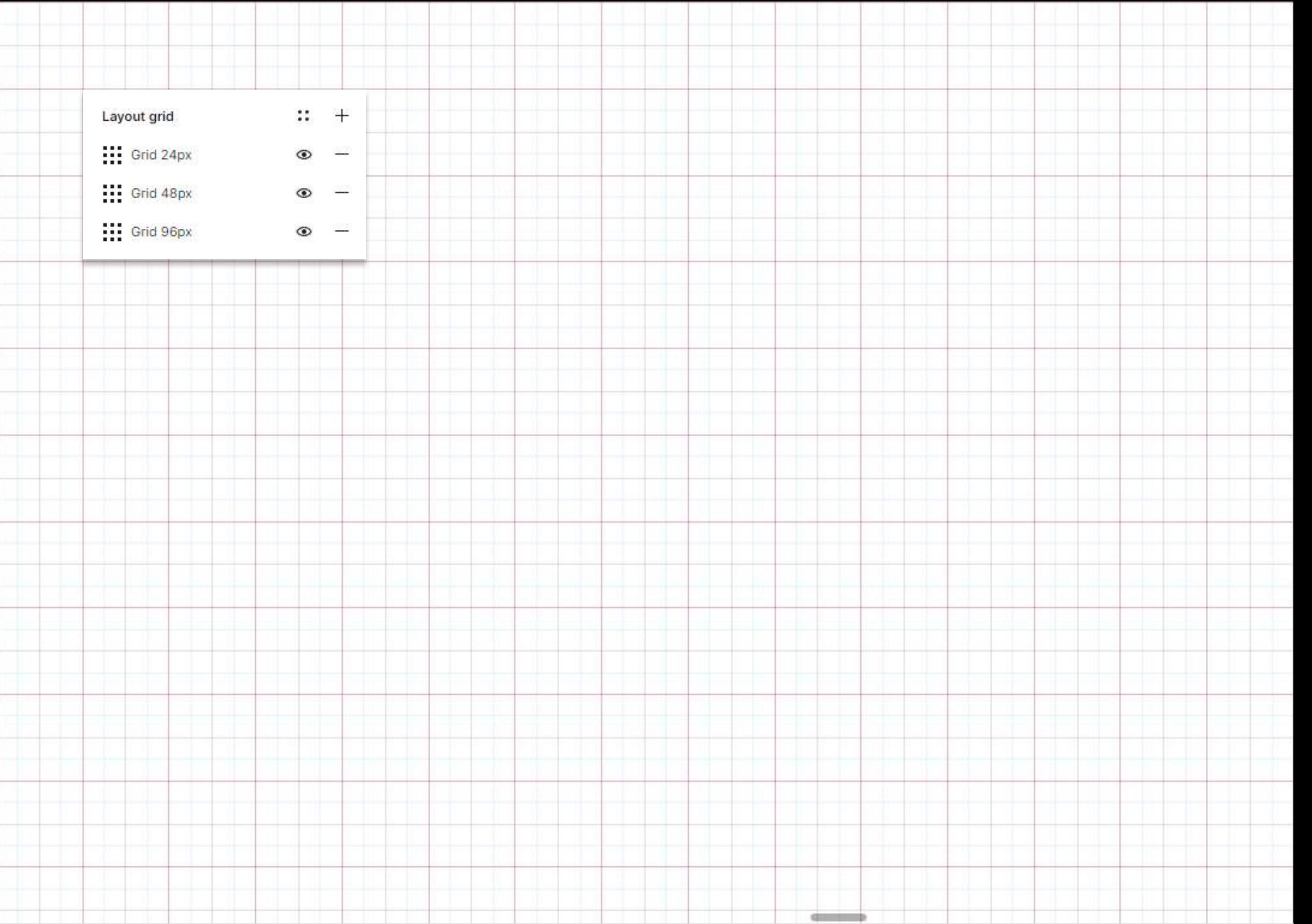
Circles will be used to represent the user, their location, and/or movement

Grid Layout

On the web, 96 pixels is equal to an inch, so the base 96px grid divides the screen into 1" by 1" squares

The 96 pixel squares are then divided into halves by the 48 pixel grid, which is then divided in half by the 24 pixel grid

The end result is a screen divided into 0.25" squares all over. This kind of spacing is what I believe to be beneficial to my project



Onboarding I

The user will be presented with a bit of context before delving into the experience. The onboarding will be simple and more story/context heavy

Navigation element to let user know where they are in the process

Content will be a combination of text and graphics that relate to it

What is ciphertext?

Ciphertext is encrypted text transformed from regular text(plaintext) via an encryption algorithm. Ciphertext is extremely difficult to read, if not impossible, unless it's decrypted into plaintext

BNDVKN KZ
ARBCMDKN
GIKAKFD K
KNMIDJHLR
LLGKNKKK



Onboarding II

The user will be presented with a bit of context before delving into the experience. The onboarding will be simple and more story/context heavy

Main call to action here is to click the button on the screen to enter the 3d experience

Notice that the action button and user location(end of the onboarding) are both represented by a rhombus icon

**enter
ENCRYPTOPEDIA**

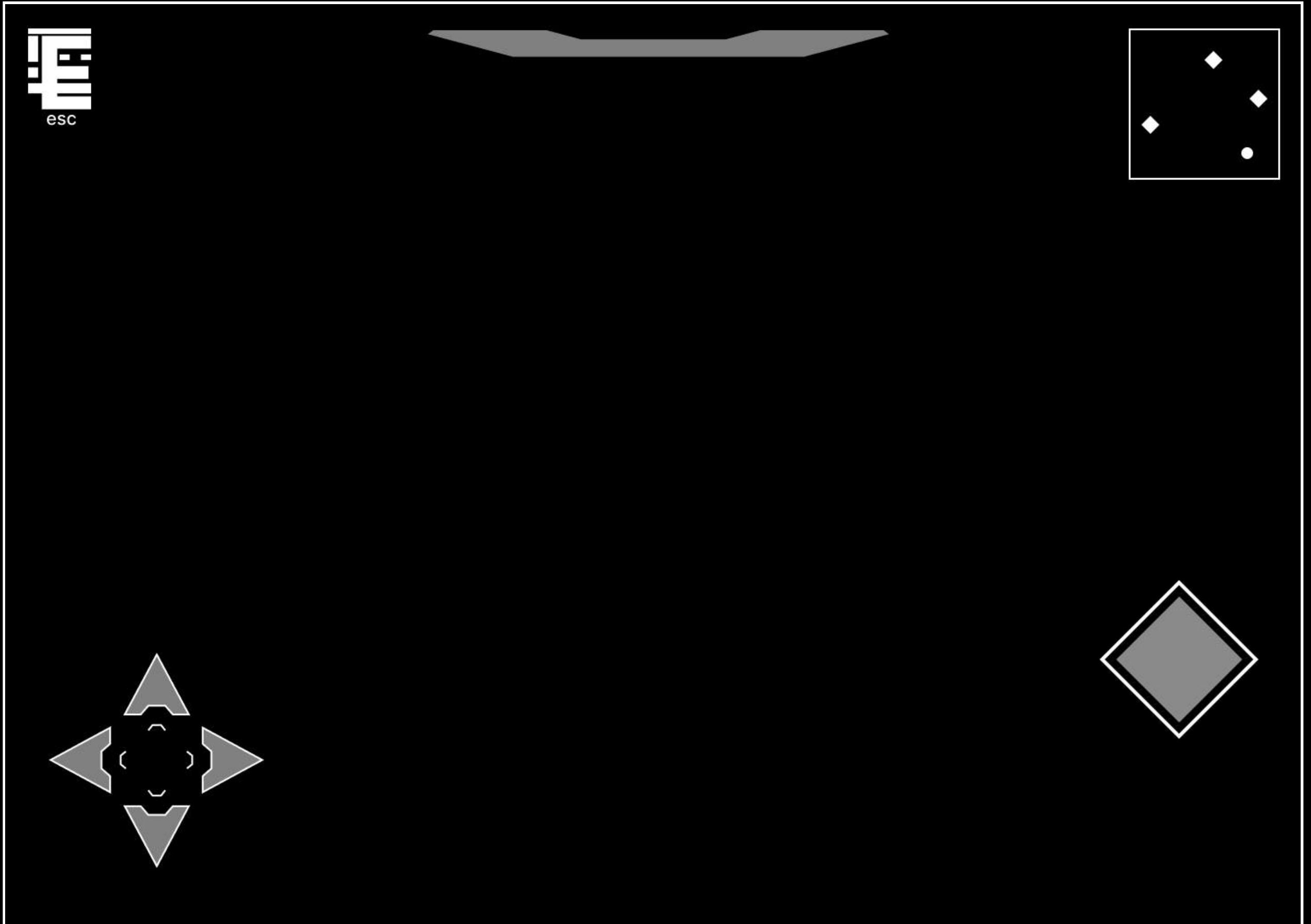


Exploration

This will be the most common screen viewed by the user. The escape button and map are higher level navigation elements

The arrows will correspond with user WASD input

The rhombus is going to signal to the user that an item is interactable

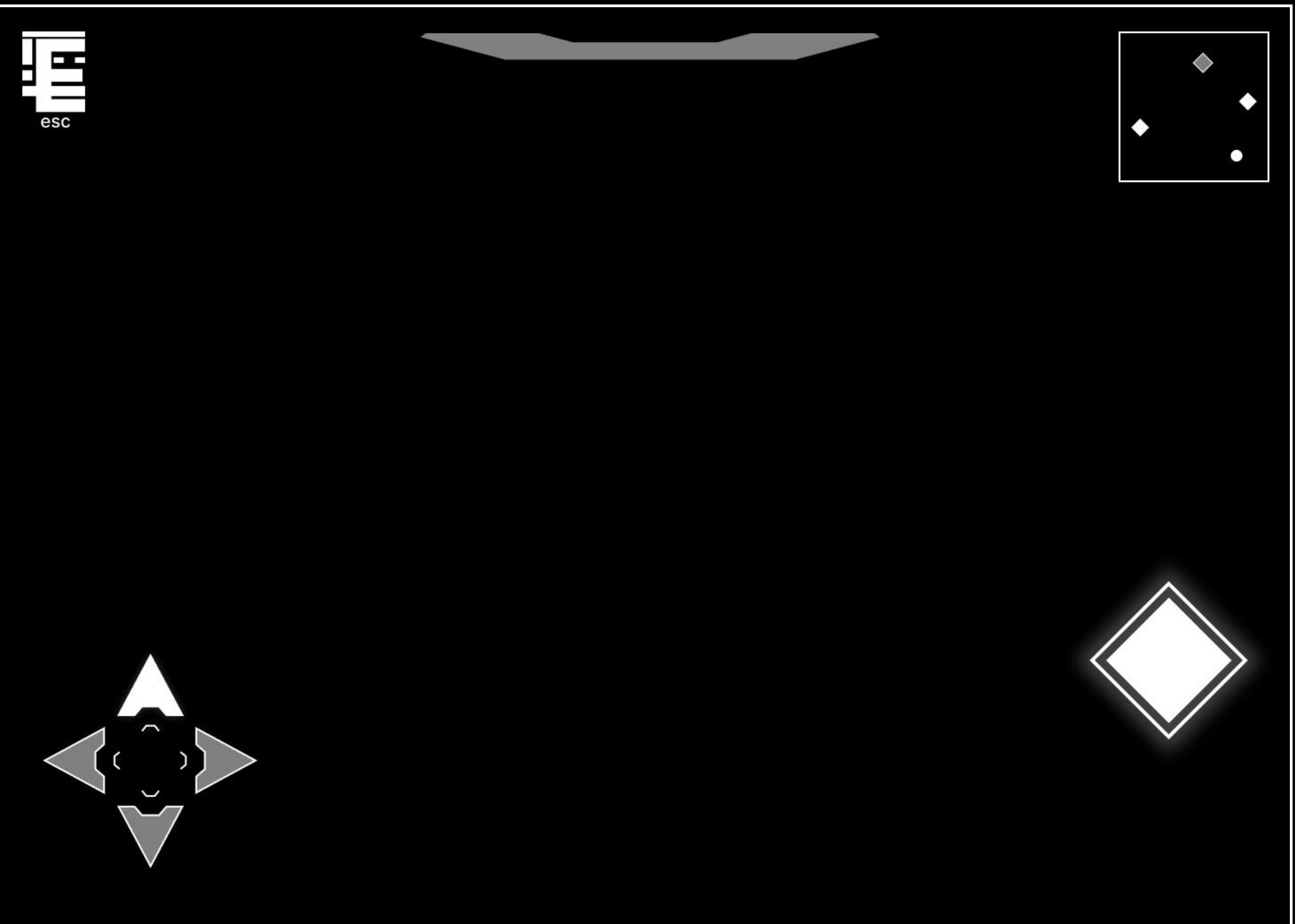


First Layer Information

Different UI elements will respond to user actions upon finding first level information. Notice how the map also displays which items have already been interacted with

The directional arrows light up with the user input direction

The rhombus glows upon looking at an interactable item, signaling that the item can be “used” per say. Some extra animation will function as the sort of reaction to the user actually clicking the button

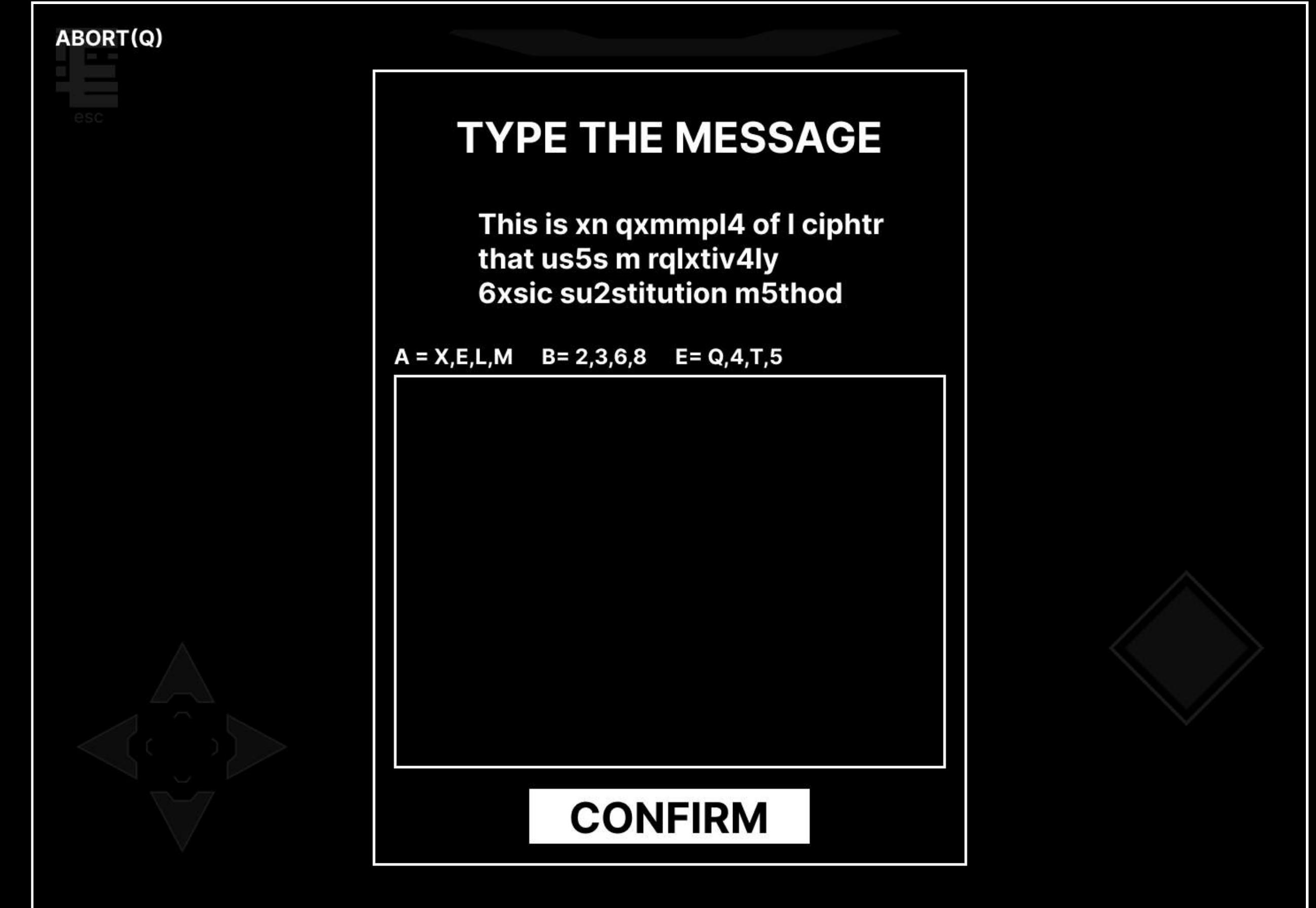


Decryption

The screen will get a black overlay, allowing the decryption pop-up to have its space. The user can abort a decryption phase without completing it.

Users will be given 2 elements: a key to the cipher and the cipher itself. They will have to use these 2 in order to complete the decryption phase and gain access to second level information

For now I'm thinking of making the ciphers be mostly text based, so the decryption phase will look the same most of the time



Second Layer Information

After completing the decryption phase, the user is presented with the secondary information about the cipher they just solved

Similar text and graphic layout to the onboarding section

User can close out of this pop-up as well

CLOSE(Q)



esc

Substitution Cipher

This cipher encrypts plaintext by swapping each letter or symbol in the plaintext for a different symbol as directed by a given key.

This is probably the most common cipher due to its simplicity. Humans from long ago used to refer to man named Caesar when talking about this cipher...

HR3J
CAR



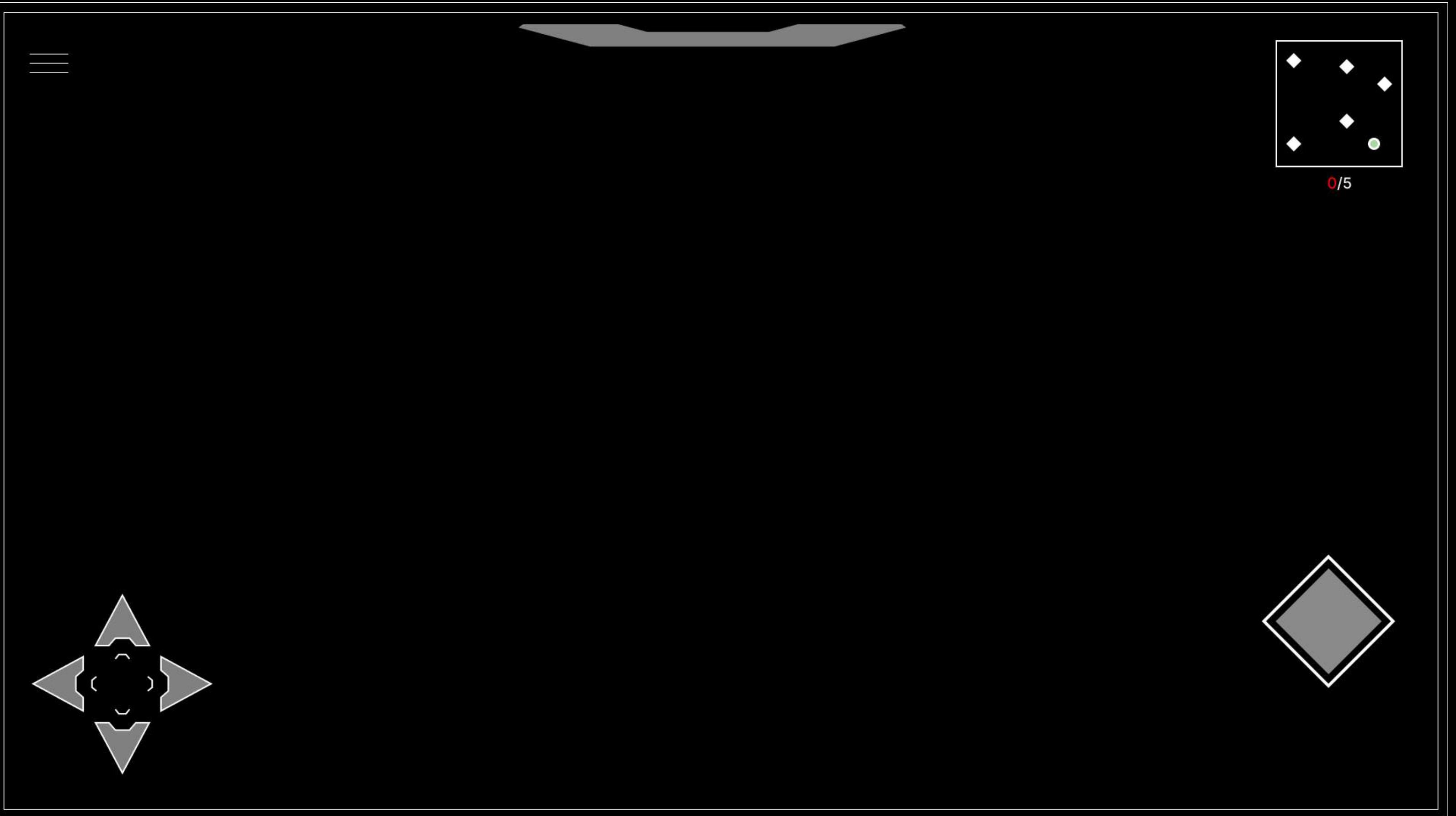
ENCRYPTOPEDIA

VISUAL COMPOSITION

Exploration Screen

This is the final version of the exploration screen

Compared to visual composition 2, this screen has a more thin and sharp feel to it that matches with the vector graphics

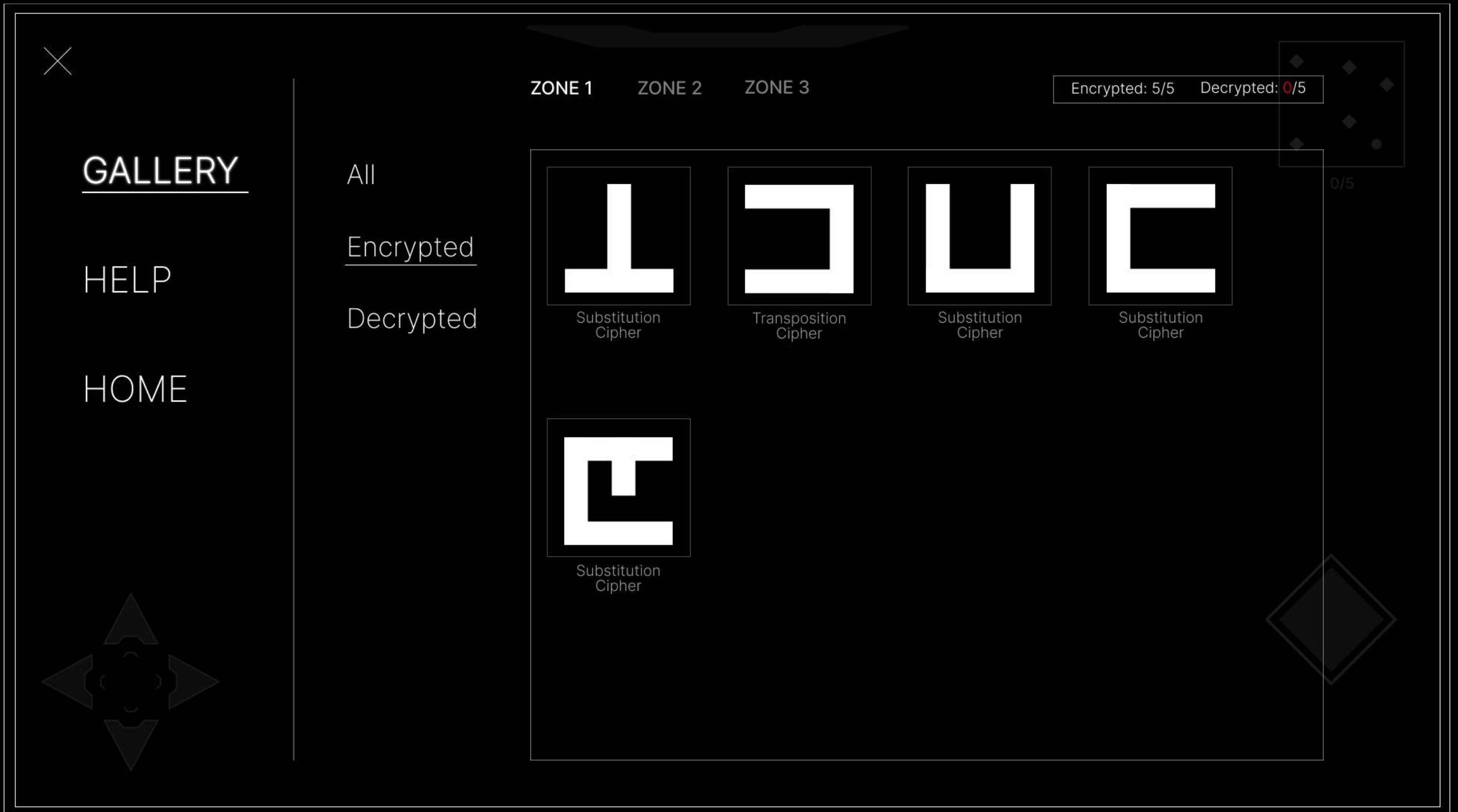


Pause Menu UI

This is another screen that the user will be frequently seeing. I tried to make the type and layout fit the aesthetic of the project while being easy to skim through

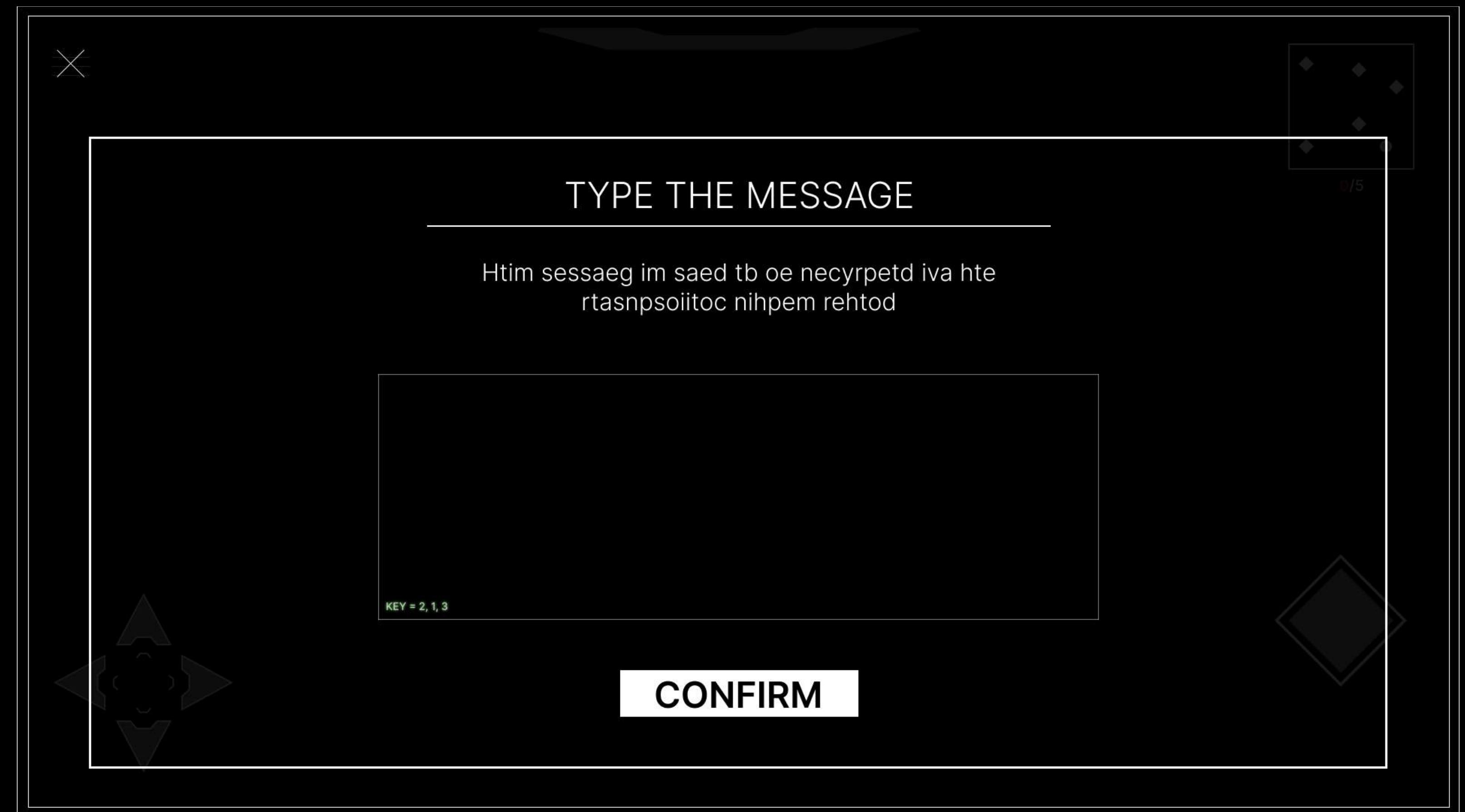
Similar text and graphic layout to the onboarding section

User can close out of this pop-up as well



Decryption

This is a pop up screen fo the decrypting



Demo

ENCRYPTOPEDIA DEMO



ENCRYPTOPEDIA

FINAL PROTOTYPE

Final

ENCRYPTOPEDIA DEMO



ENCRYPTOPEDIA

REFLECTION

Final Thoughts

In all honesty, I'm just burned out. I spent the entire project being either on or ahead of schedule, which was good, but resulted in a huge crash at the end of the project. I had originally intended for this project to be turned into a full game that people could actually download and play, but for now, I'll just shelf it as a proof of concept for a game so that if I ever want to return to it (or if someone wants to fund it) I can fully complete development

I did end up realizing that I like game UI/UX more than normal UI/UX. That, and that creating stuff is more fun for me than designing stuff. I would've never had the drive to complete this project in the time I did if it was purely a design-based one

In the end, there was a lot of content that I wanted to implement, create, and go over, but my mental limit has got me stopping here for now. It still resulted in a decent portfolio piece though

Miscellaneous Content

I wasn't able to cover a lot of stuff like the language and music I made and other visual assets, so I'll just leave my figma file link here for anyone that cares enough to dig through it

[Figma file link](#)