Bilkent University

Department of Computer Engineering

Senior Design Project

Project Analysis Report

Deepgame

Students

Betül Reyhan Uyanık
Mert Alp Taytak
Ömer Faruk Geredeli

Supervisor

Dr. Uğur Güdükbay

Jury Members

Prof. Dr. Özgür Ulusoy
Asst. Prof. Dr. Shervin Rahimzadeh Arashloo

Innovation Expert

Cem Çimenbiçer

March 23, 2020

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

1 Introduction

Modern times offer people various kinds of video games through a multitude of platforms. In some genres of games the user plays as a character. Since the appearance of the player character is important be it for immersion or role playing purposes, games have been offering character customization options. Some of these customization options take the form of accessories or clothing that changes the player character superficially. Another option is customization of character models. Some games, with the help of improving technology, offer extensive and elaborate character creation menus. Which makes it possible to create a visually diverse range of characters. However, customization is not limited to visuals. Character voices are another part of the customization process. Due to its nature, human speech is not fit for modeling and customization in the way the human body is. Many games we have seen in the market use a limited set of recorded, generic voice lines for characters. Some do not voice act player character's lines at all. Perhaps, due to how expensive it gets to offer various voices for extensive dialogue options.

We would like to develop a software tool that can be integrated into games themselves to allow the user to put themselves, or maybe others, into the game visually and auditorily. While there exist games with similar features, they require expensive hardware. Using techniques developed in recent years, Deepgame will work with photographs and audio recordings. Easily acquirable through hardware common to almost every computer user.

2 Current System

Currently, there is only one game that does anything similar to Deepgame's purpose. That is the game Kinect Sports Rivals [1]. Kinect Sports Rivals is an XBOX One game and utilizes the console's Kinect motion-sensing camera to take a depth image of the player and create an in game model with player's likeness.

3 Proposed System

3.1 Overview

Deepgame will be a software tool that can be integrated into games by the game developers with the purpose of putting humans' visual and auditory likeness into the game. It will be made up of two main modules: *Visual Engine* and *Auditory Engine*. Visual Engine will take photographs of the target person and train a model to match the target's appearance. Auditory Engine will take sound recordings of the target and train a model to match the target's speech and voice. After the training process these modules will act to transform the in-game player character into the target's likeness.

3.2 Functional Requirements

3.2.1 Visual Feature Transfer

- User is able to train an in game model of a target's likeness by feeding Deepgame images of the target.
- This model can be used to change the appearance of an in game character to match the target's appearance.
- As an advanced feature, videos can be processed to extract mimics as animations.

3.2.2 Auditory Feature Transfer

- User is able to customize a default voice package using recordings of a target's voice such that lines in the package sound like they are spoken by the target.
- Customized packages should be applicable to any in game character.

3.2.3 Feature Packages

- Results of visual and auditory feature transfers can be packaged and shared with others in a secure and private manner.
- These packages should be reusable between games.

3.3 Nonfunctional Requirements

3.3.1 Performance

- Training process for each feature should take less than an hour with the ideal target of less than ten minutes.
- Resulting models should be usable in real time with no major performance cost.
- Input lag should be less than 50 milliseconds.

3.3.2 Characteristics of Feature Transfer

- Resulting models should appear and sound reasonably convincing to be the target's looks and voice.
- Resulting models should match the art style of the game they are in with relatively little effort on developers' and user's end.

3.3.3 Portability

- Underlying software of Deepgame should be portable between platforms.
- Feature packages should be reusable between games and platforms.

3.3.4 Privacy

- User data should only be accessible and usable through their consent.

- Malicious use of the product that can undermine people's privacy should be prevented.

3.3.5 Security

- Any sensitive data should be encrypted if permanence is necessary, deleted after use if not.
- Only the user owned hardware should access user data during operation, i.e. no cloud processing.

3.4 Pseudo Requirements

3.4.1 Implementation Requirements

We will develop Deepgame as a software tool that can be integrated into games and we will develop a video game to showcase its features. For this purpose, we will use Unity as the game engine to develop our game. Deepgame will be a tool to be used with Unity projects like others in the Unity Asset Store [2].

3.4.2 Development Process Requirements

During our development process, we will use version control and issue tracking. For this purpose we will use git and the GitHub platform.

3.5 System Models

3.5.1 Scenarios

Use Case #1

Unique Name: Create New Game
Participating Actor/s: Player
Entry Condition: Player is directed to creation new game menu by clicking "Create New Game" button on the main menu of game.
Exit Condition: The game will be started by clicking "Play Game" button.
Flow of Events:

1. After clicking Create New Game button, the page will appear that include options which are Create New Character, Use Registered Character and Back to Main Menu.
2. Player will pass successfully the next menu by clicking "Create New Character"
3. After creation of new character, Player will choose its newly created character by clicking "Use Registered Character" and then Player will be directed to Use Registered Character menu.
4. Player will choose its character by clicking "Select Character" button.
5. After selection of character, Player will be able to choose game modes that include classic mode and fear mode. Player will click one of them and "Play Game" button will be activated.

Use Case #2

Unique Name: Edit Options
Participating Actor/s: Player
Entry Condition: Player is directed to options menu by clicking “Edit Options” on the main menu of the game.
Exit Condition: Player will turn back to main menu by clicking “Back to Main Menu”.
Flow of Events:

1. After clicking Edit Options button, new menu will appear that include game settings.
2. Player can save the settings that he/she changes if he/she wants by clicking “Save Settings”.
3. Player will back to the main menu by clicking “Back to Main Menu”.

Use Case #3

Unique Name: Create New Character
Participating Actor/s: Player
Entry Condition: After clicking “Create New Character” button the player is directed to Customize New Character menu.
Exit Condition: Player will create character by clicking “Create Character” button.
Flow of Events:

1. After clicking Customize New Character button, the new page will appear.
2. Player will upload photographs for the visual customization by clicking “Upload Photographs” button.
3. Player will upload audio recordings for the auditory customization by clicking “Upload Audio Recordings” button.
4. Player will create character after uploadings by clicking “Create Character” button and player will be redirected to Create New Game menu.

Use Case #4

Unique Name: Quit Game
Participating Actor/s: Player
Entry Condition: Player will press the escape button on the keyboard to see the Pause menu.
Exit Condition: Player will be directed to the Main Menu.
Flow of Events:

1. Player will see the Pause Menu and press the button quit game.

Use Case #5

Unique Name: Edit Character
Participating Actor/s: Player

Entry Condition: Player is directed to Customize Character menu that includes “Create New Character”, “Edit Character” and “Delete Character” buttons and table registered characters by clicking the “Customize Character” button on the Main Menu.
Exit Condition: Player will be back to the Main Menu.
Flow of Events:

1. Player will choose the character that he wants to edit by clicking the “Edit Character” button.
2. Edit Menu will be appeared, Player will change eye color, hair color and body features.
3. Player will save the changes.
4. Player will turn back to the Main Menu.

Use Case #6

Unique Name: Delete Character
Participating Actor/s: Player
Entry Condition: Player is directed to Customize Character menu that includes “Create New Character”, “Edit Character” and “Delete Character” buttons and table registered characters by clicking the “Customize Character” button on the Main Menu.
Exit Condition: Player will turn back to Main Menu
Flow of Events:

1. Player will choose any character that was created before in the character list.
2. Player will delete the selected character by clicking the “Delete Character” button.
3. Player will turn back to the Main Menu.

3.5.2 Use Case Models

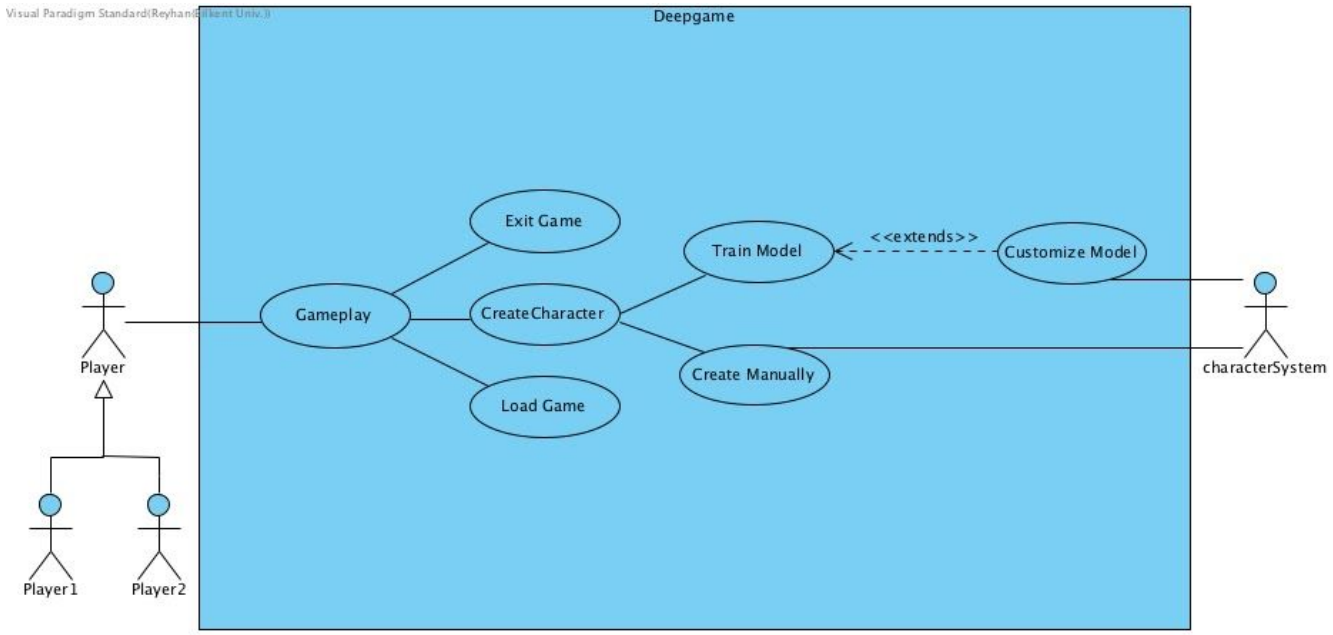

Figure 1: Use Case Diagram

3.5.3 Object and Class Models

Class models are separated into two packages for ease of reading. These two packages normally work independent of each other.

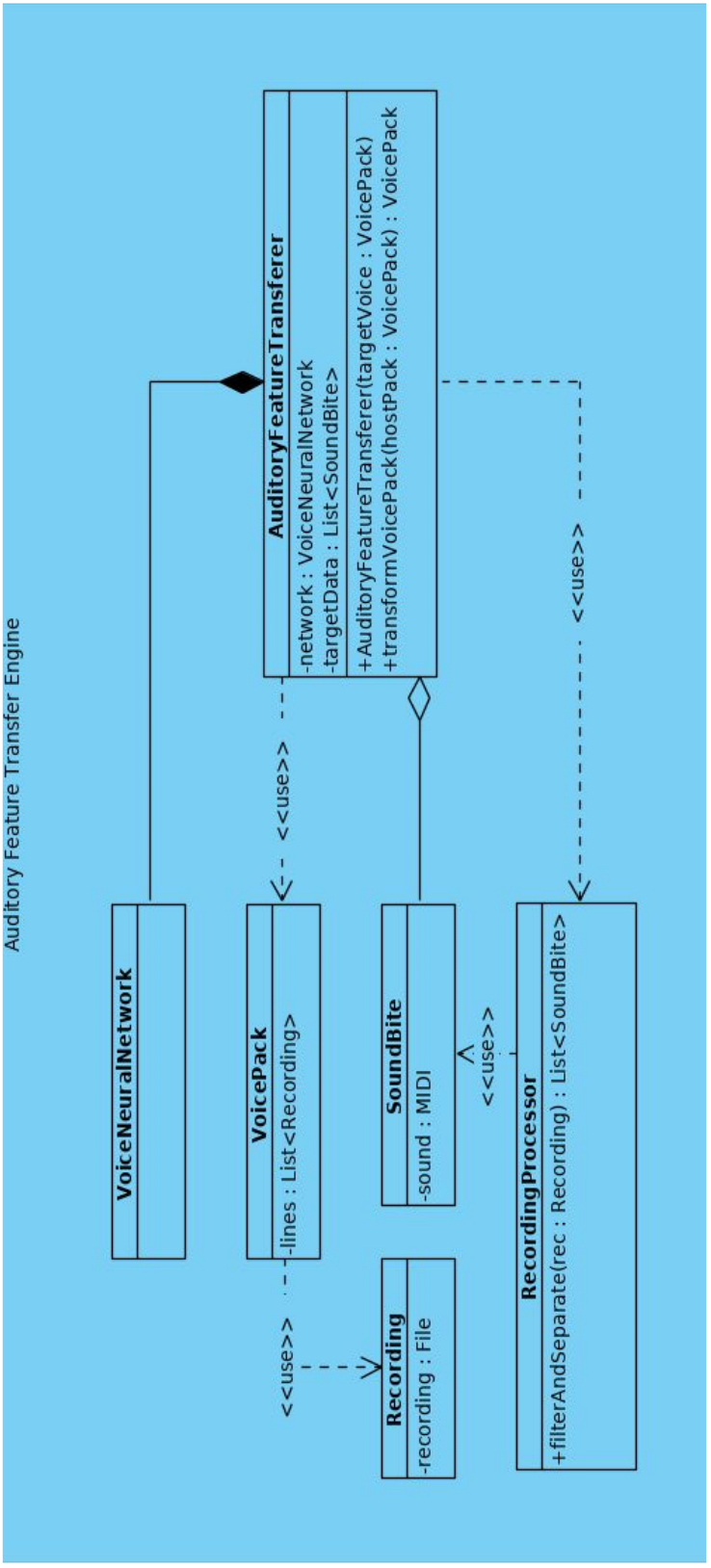

Figure 2: Class Diagram for Auditory Feature Transfer Engine

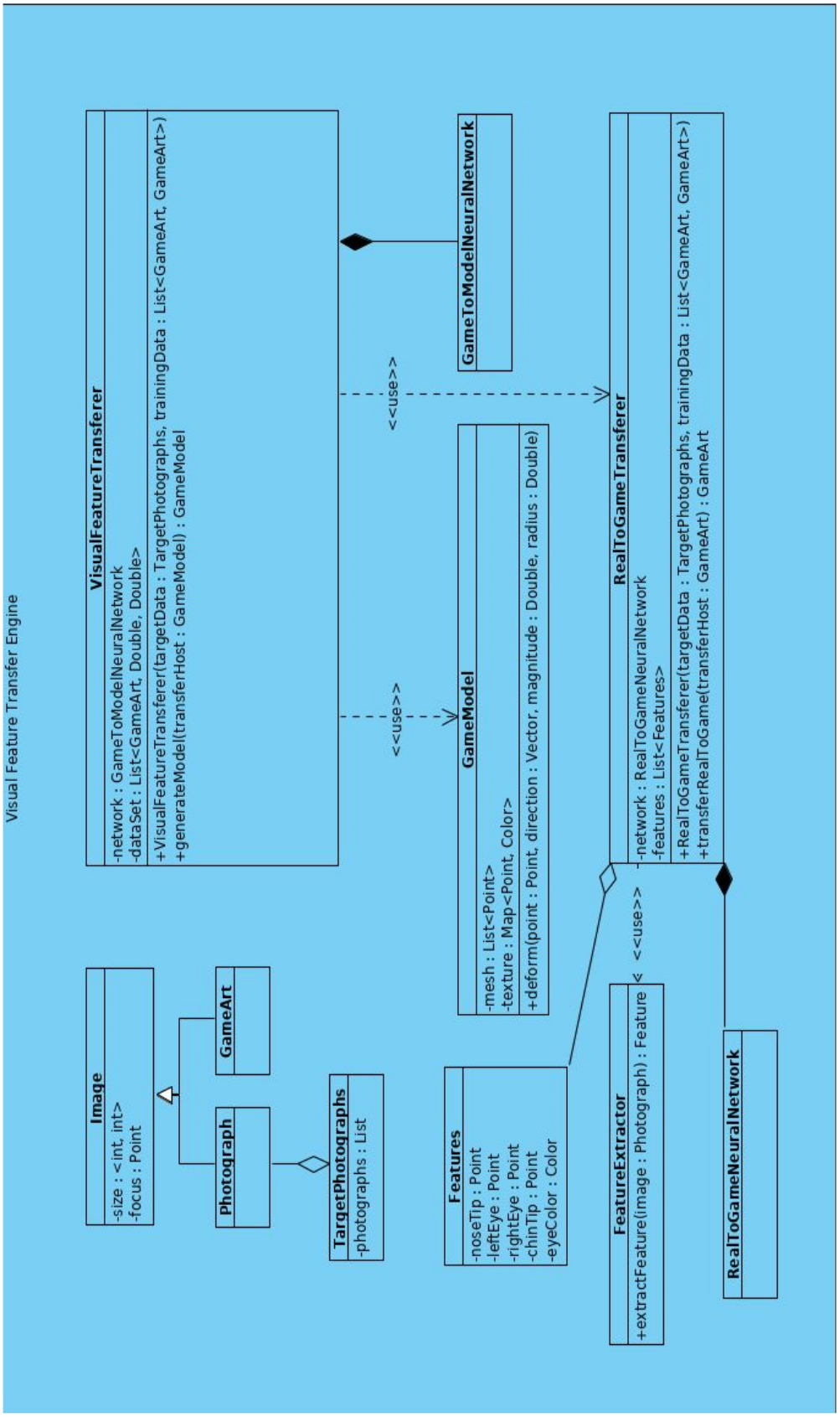

Figure 3: Class Diagram of Visual Feature Transfer Engine

3.5.4 Dynamic Models

3.5.4.1 Activity Diagram

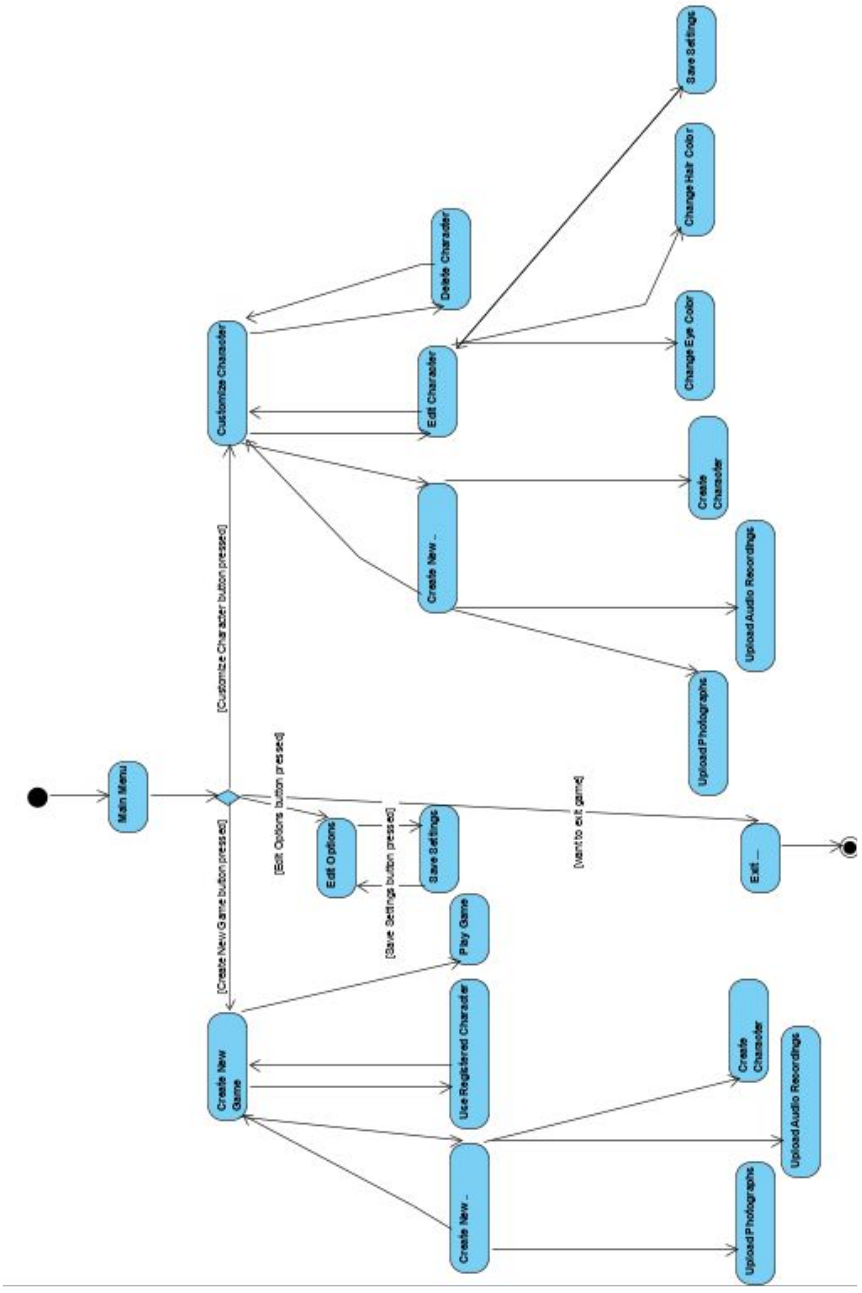

Figure 4: Activity Diagram for Deepgame

3.5.4.2 State Diagrams

3.5.4.2.1 Create New Game

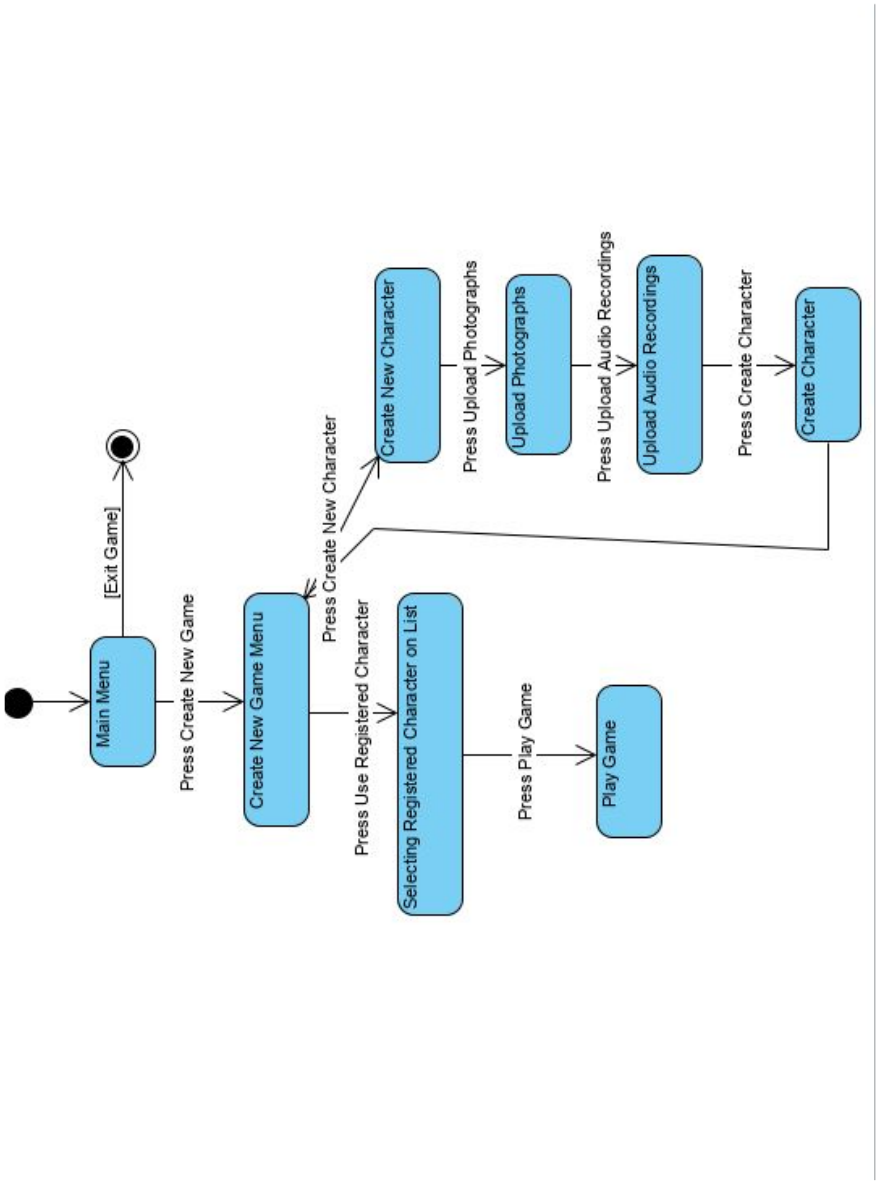

Figure 5: New Game Creation States

3.5.4.2.2 Customize Character

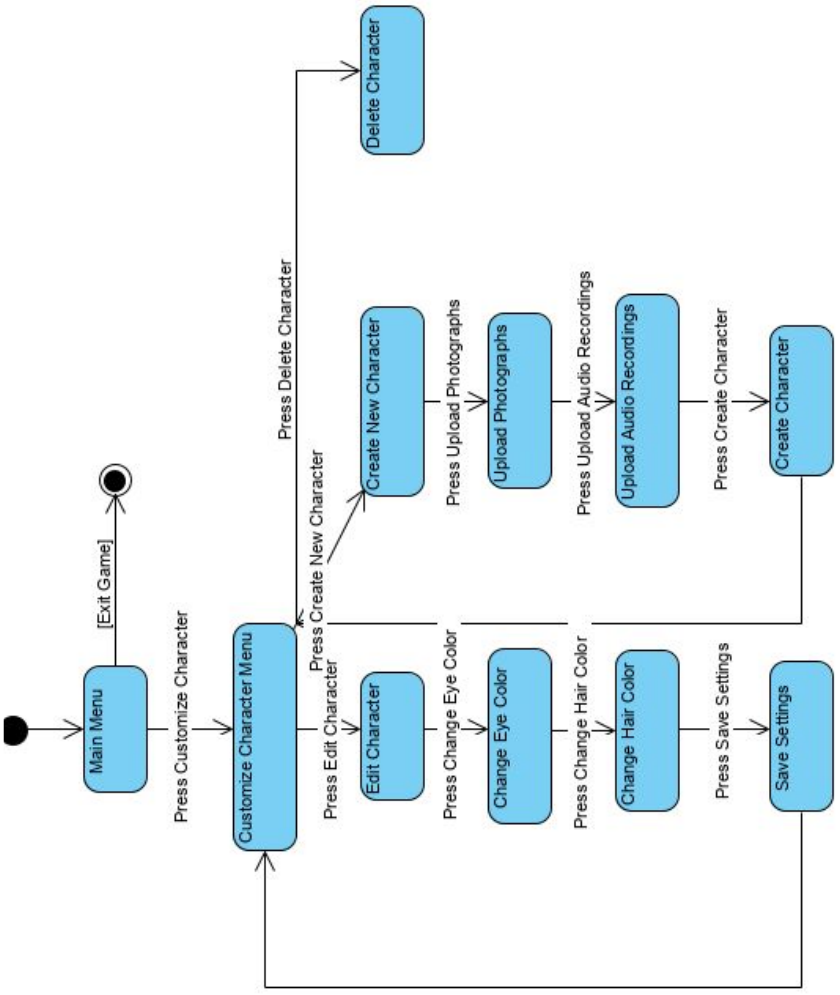

Figure 6: Character Customization States

3.5.5 User Interface - Navigational Paths and Screen Mock-ups

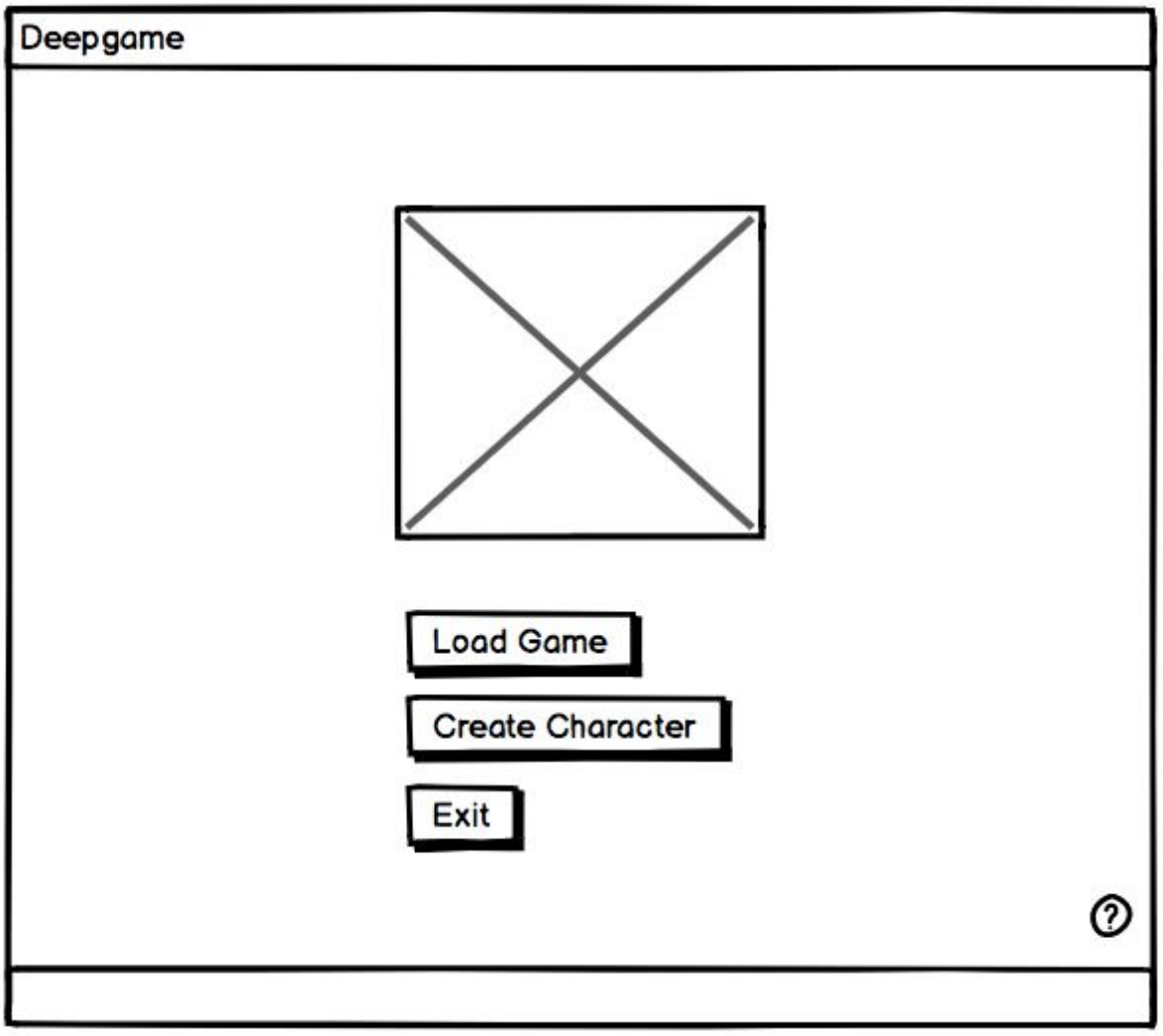

Figure 7: Main Menu

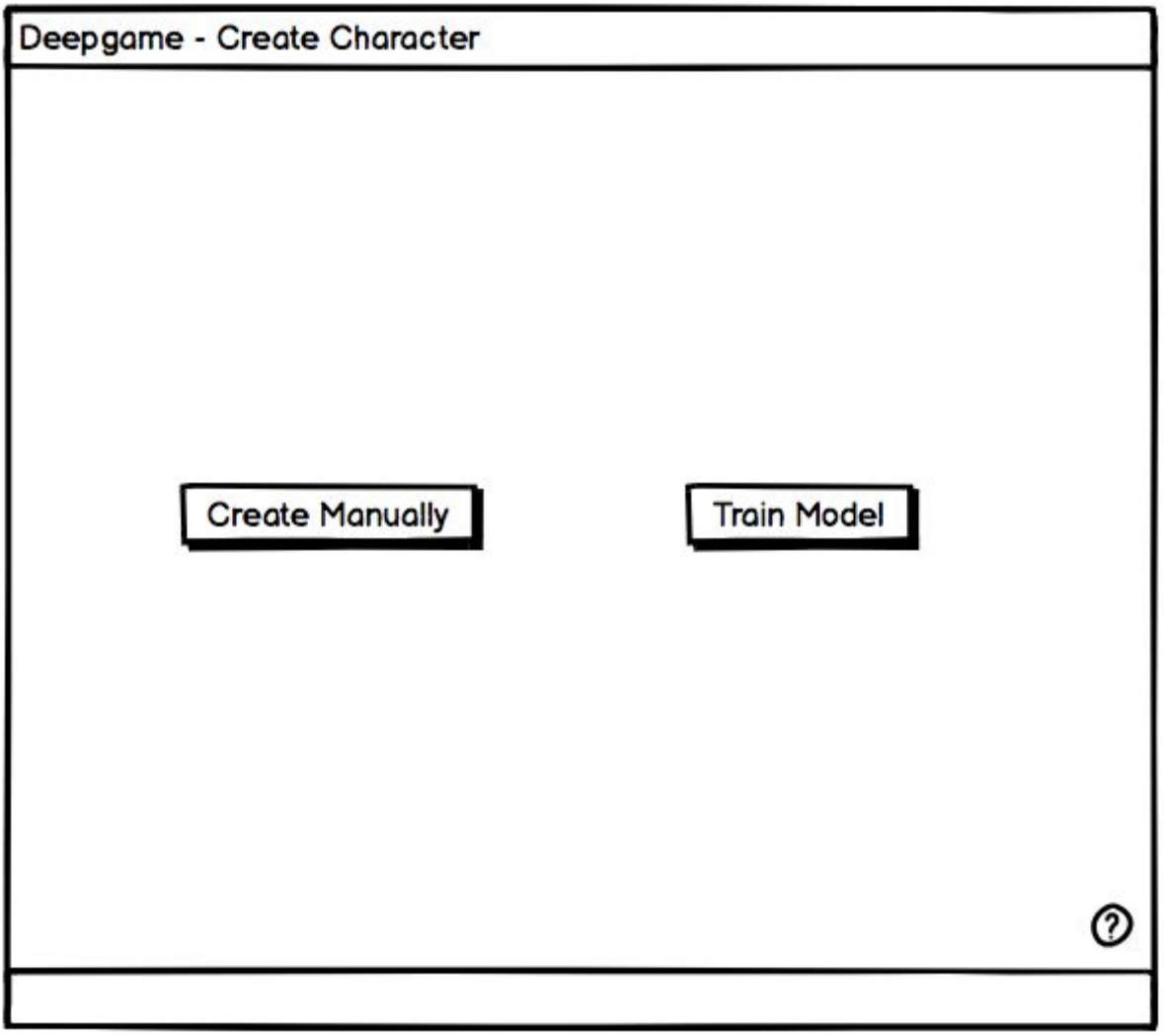

Figure 8: Creating Creation

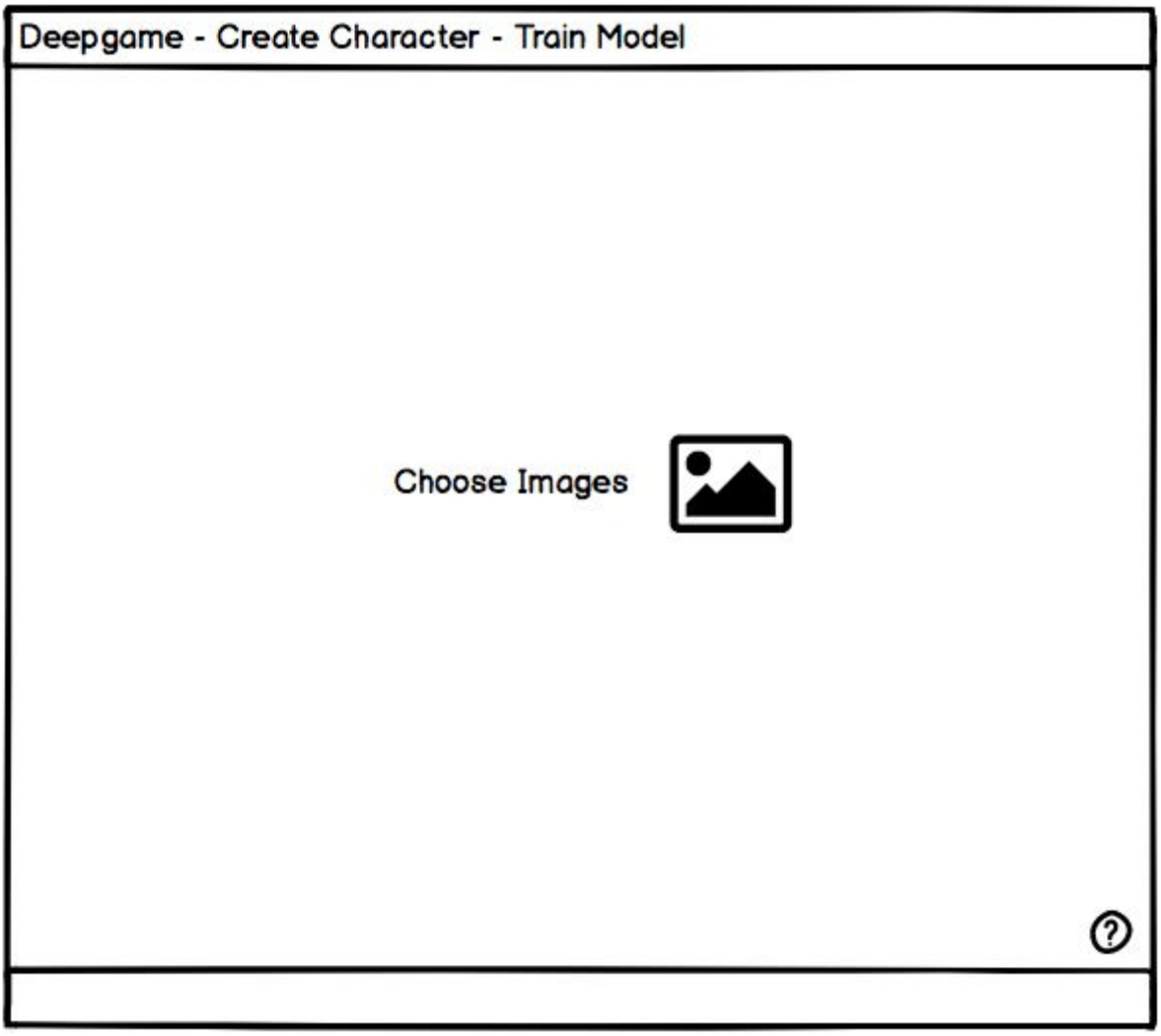

Figure 9: Uploading Images for Model Training

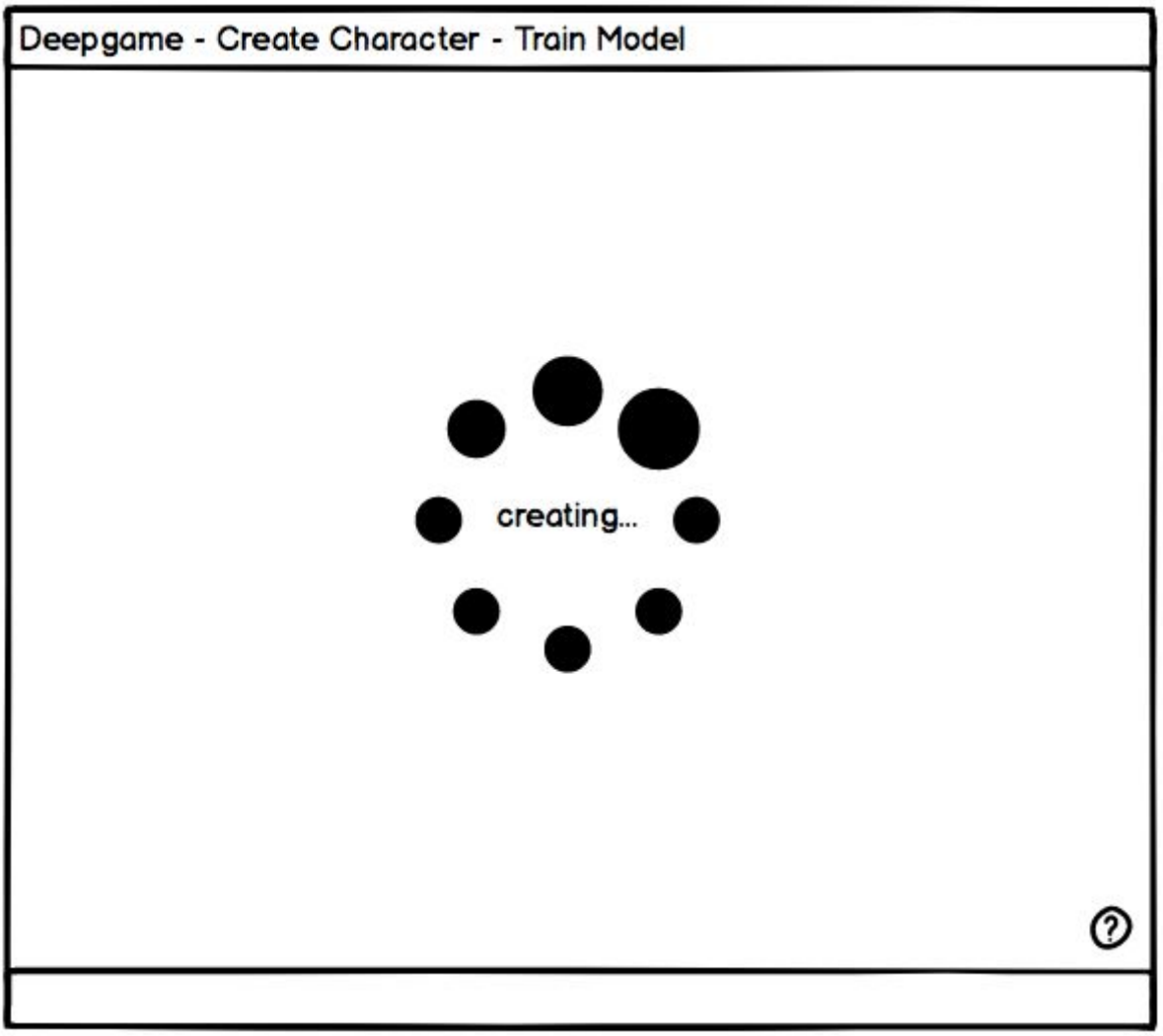

Figure 10: Creation of the Character using Deep Fake Algorithms

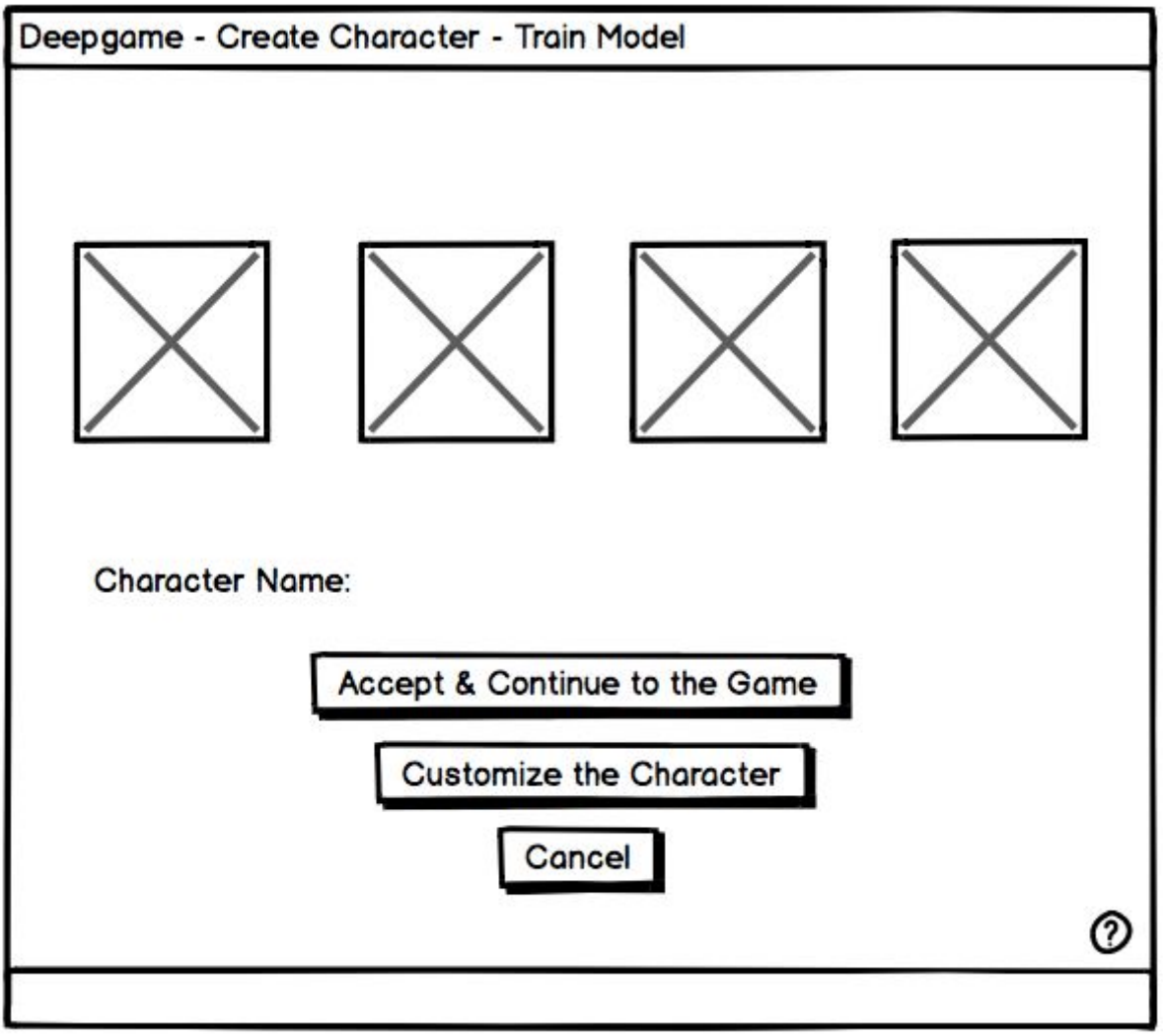

Figure 11: Resulting Character

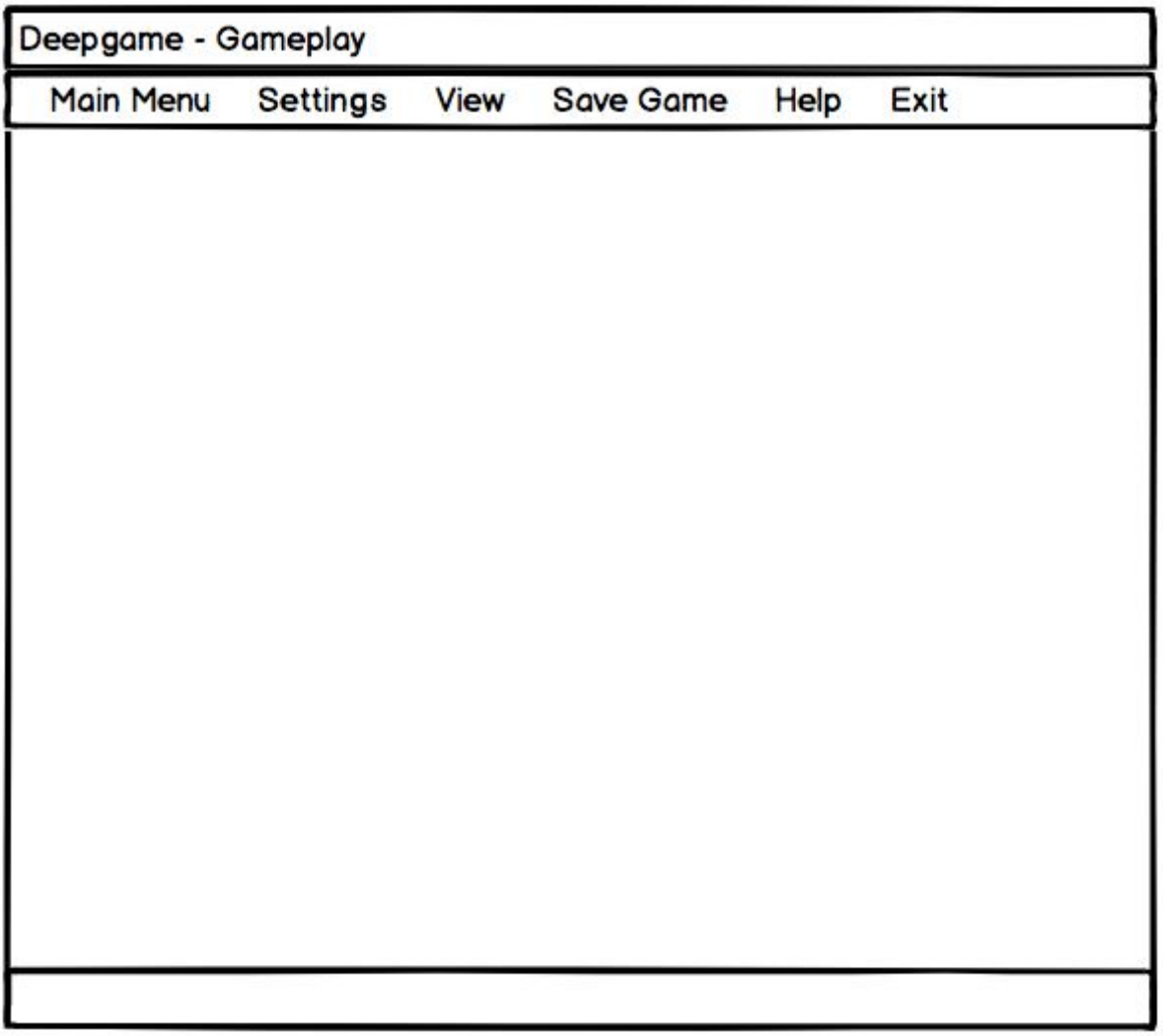

Figure 12: Gameplay

4 Other Analysis Elements

4.1 Consideration of Various Factors

4.1.1 Entertainment

Our project has the potential to bring a new breath to the entertainment industry. In the game that should be developed with any character that can be played with the personalization of the game characters, the more people feel themselves in the game will make the player feel the reality more. The character customization techniques that are tried to be obtained in many games will be at a different level with the realization of our project. With the development of this method at a sufficient level, it may be aimed to increase the interest in the entertainment sector more.

4.1.2 Socialization

With the multiplayer mode, we aim for the players to spend more time together by competing with each other with their personalized game characters. With the increased sense of reality, people will be able to challenge each other and test the strength of each other's characters within the game. As a result, it can directly affect people spending more time with each other.

4.2 Risks and Alternatives

First of all, entire group members will learn Deepfake technology. Since this technology is very new to us, we will start from the zero. Learning is a process which may have different responses on the every single person. Realizing and comprehensive understanding will take significant amount of time to implement the project. Because of that , we may encounter a risky situation that may prevents us completing the project before the deadline. In terms of the syllabus we have 2 semesters to implement the project. That is why, we should start the coding section of the project at the very beginning of the second semester. Moreover, the motivations and responsibilities of the each member will be determinant factor for risks. At the beginning of the each stage of the project, the workload is being allocated along team members fairly. In order to succeed the project, each member is responsible of doing his/her work within the specified time interval. Moreover, the project may be implemented in many different ways. Those implementation and design styles will construct alternatives of the project. The first is to adapt deepfake to games and change the character's face in real time like post-processing effect. In this method, there is no change in in-game materials. Second is to change the character model using Deepfake. we'll have a default character model at hand, and we'll look at it from different angles. By applying Deepfake algorithm, we will get the image transfer applied. Then we will take this picture and change a new 3D model to look like it. Both methods have deepfake application from real human characteristics to game characters. But in the first method, we do this continuously in real time. There are

animations and facial expressions that change while the game is playing. In the second method, we use the 3D model I obtained in the game.

4.3 Project Plan

4.3.1 Work Packages

Deepgame itself is a software tool. But, we will make a video game that integrates Deepgame for demonstration purposes. Therefore, we have the following work packages:

- Visual Feature Transfer Engine – Photographs to Game Art
- Visual Feature Transfer Engine – Game Art to Game Model
- Auditory Feature Transfer Engine
- Deepgame Intregratable Tool
- Deepgame Integrated Video Game

In the following section, there will be a detailed breakdown of these packages.

4.3.2 Breakdown of Work Packages

Package 1: Visual Feature Transfer Engine – Photographs to Game Art
Start Date: 15/05/2020
End Date: 01/09/2020
Leader: Mert Alp Taytak
Assigned Members: - Mert Alp Taytak - Reyhan Uyanık
Objectives: Create part of the Visual Feature Transfer Engine that handles deepfake-style transferring of visual features from the target's photographs to default game models with proper art style.
Tasks: **Task 1.1 Real-to-Real Still Image:** Achieve visual feature transfer from photographs of real people to photographs of real people. Basically the deepfake technology for single frames. **Task 1.2 Real-to-Game Still Image:** Achieve visual feature transfer from photographs of real people to images of games with proper game art style. Basically the deepfake technology that targets game art instead of real life images.
Deliverables: **Deliverable 1.1:** Real-to-Art Deepfake Tool

Package 2: Visual Feature Transfer Engine – Game Art to Game Model
Start Date: 15/05/2020
End Date: 01/09/2020
Leader: Reyhan Uyanık
Assigned Members: - Reyhan Uyanık - Mert Alp Taytak
Objectives: Create part of the Visual Feature Transfer Engine that handles creating 3D game models that are game-appropriate from images of game characters with proper art style.
Tasks: **Task 2.1 Game Model Editor:** Write the basis software that takes a generic 3D game model and is capable of making fine adjustments to the model. **Task 2.2 Image to Model Generator:** Achieve generation of 3D game model from 2D game-appropriate images of the target model that is input images from any angle asked by the generator.
Deliverables: **Deliverable 2.1:** Art-to-Model Generator Tool

Package 3: Auditory Feature Transfer Engine
Start Date: 15/05/2020
End Date: 01/09/2020
Leader: Ömer Faruk Geredeli
Assigned Members: - Ömer Faruk Geredeli - Reyhan Uyanık
Objectives: Create part of Deepgame that handles applying target's sound recordings to pre-existing voice packs to create new voice packs that sound like their lines are spoken by the target.
Tasks: **Task 3.1 Exact Line Transfer:** Achieve auditory transfer that takes a voice line and makes it sound like spoken by someone else. **Task 3.2 General Voice Transfer:** Achieve auditory transfer that takes multiple sound recordings of a target person and that can make any voice line sound spoken by the target.

Deliverables: **Deliverable 3.1:** Auditory Feature Transfer Engine

Package 4: Deepgame Integratable Tool
Start Date: 01/09/2020
End Date: 01/10/2020
Leader: Ömer Faruk Geredeli
Assigned Members: - Ömer Faruk Geredeli - Mert Alp Taytak
Objectives: Integrate parts of the Visual Feature Transfer Engine and the Auditory Feature Transfer Engine to make a tool that can be integrated into any game via an interface.
Tasks: **Task 4.1 Visual Engine Parts Integration:** Integrate two parts of the Visual Feature Transfer Engine to complete the engine and make sure it works seamlessly. **Task 4.2 Integrate Engines into Final Tool:** Integrate both engines to create the final Deepgame tool and create the integration interface.
Deliverables: **Deliverable 4.1:** Deepgame Integratable Tool **Deliverable 4.2:** Integration Interface **Deliverable 4.3:** Documentation

Package 5: Deepgame Integrated Video Game
Start Date: 01/08/2020
End Date: 01/12/2020
Leader: Mert Alp Taytak
Assigned Members: - Mert Alp Taytak - Ömer Faruk Geredeli - Reyhan Uyanık
Objectives: Create a video game integrated with Deepgame to demonstrate Deepgame's features.
Tasks: **Task 5.1 Game Design:** Design a video game that is sufficiently complex and can demonstrate Deepgame's features. **Task 5.2 Game Development:** Develop the game following the agreed upon design.

Task 5.3 Deepgame Integration: Integrate Deepgame Integratable Tool into the game. **Task 5.4 Playtest:** Test the game from both technical and game design perspectives and fix found flaws until desired quality is reached.
Deliverables: **Deliverable 5.1:** Deepgame Integrated Video Game **Deliverable 5.2:** Documentation **Deliverable 5.3:** User's Manual

4.3.3. Gantt Chart

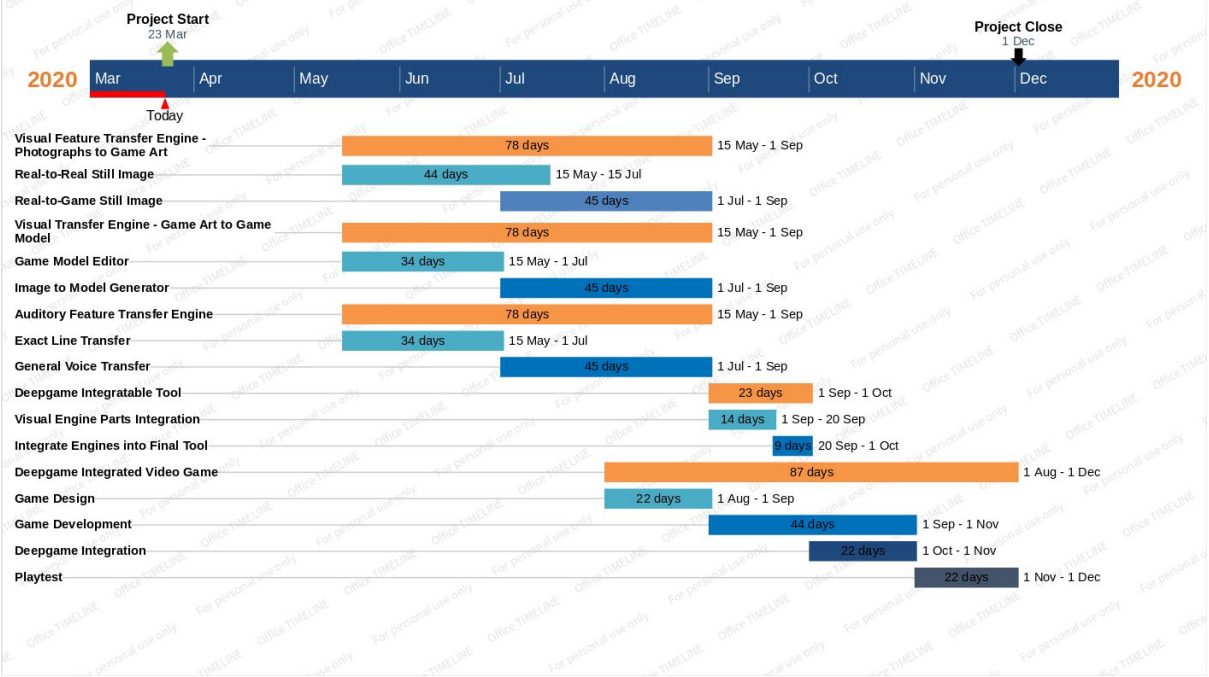

Figure 13: Gantt Chart (Format: DD/MM/YYYY)

4.4 Ensuring Proper Teamwork

In order to create a successful tool that can be integrated into the games, every member should be engaged in the project. In order to coordinate tasks among group members equally and keep track of each other's work we are using online tools. Currently we are using google drive for project related documents. We have decided to use google docs for our questions/concerns about the project, and to have regular online meetings, face to face meetings whenever possible. Our main tool to keep track of the contribution of each member will be Github.

4.5 Ethics and Professional Responsibilities

Deepgame will collect personal data(images) from the players only if they are consented to do so. Personal information about players also may contain sensitive information like location and name. It is very important for us to use this information in total confidentiality

and on the grounds of player's consent and agreement. Player can cancel the process of training a model. Any sensitive data will be encrypted, and can be deleted as the player wishes.

4.6 New Knowledge and Learning Strategies

In order to complete this project, we need to learn about new fields, algorithms and technologies. Our deliverables can be categorized into two parts:

1. A visual and auditory feature transfer technology in the form of Deepgame,
2. A video game integrated with Deepgame for demonstration.

For the first part, we need to study deepfake algorithms for both visuals and sounds. Then, we need to learn how they can be applied to games. Since existing deepfake algorithms target real life photographs, it will be challenging to create a variant that can match a game's art style. An algorithm capable of this will need to be designed by us. In order to achieve this, we will need to study advanced machine learning topics such as deep neural networks or convolutional neural networks.

For the second part, we need to learn about game design and game development. For game design, we need to learn about level design, concepts and general principles about what makes a good game. For game development, we need to learn about a game engine of our choice and how to use it. Most likely, this will require learning a new programming language as well.

As a general requirement, both Deepgame and the video game we will develop needs to be efficient, consistent and performant. We will have to learn high performance computing techniques in order to achieve this.

For the sources; first part requires academic style learning and second part requires engineering style learning while third part requires a combination of both. Academic style learning will be done by researching academic papers and text books. This can be done by finding computer scientists working in relevant fields and looking through their research history along with the courses they teach. Engineering style learning will be done through online courses, tutorials and forums.

Finally, we will consult with our project advisor and other faculty members for guidance as needed.

5 References

[1] X. W. Staff, “Creating a Champion in Kinect Sports Rivals,” Xbox Wire, 20-Oct-2015. [Online]. Available: https://news.xbox.com/en-us/2014/04/01/games-ksr-developer-qa/. [Accessed: 23-Feb-2020].

[2] “Unity Asset Store - The Best Assets for Game Making,” Unity Asset Store - The Best Assets for Game Making. [Online]. Available: https://assetstore.unity.com/. [Accessed: 23-Mar-2020].