MMLAB BLOCKCHAIN GAME

Odyssefs Diamantopoulos Pantaleon

Ilias Marios Stogiannidis

Thesis Report

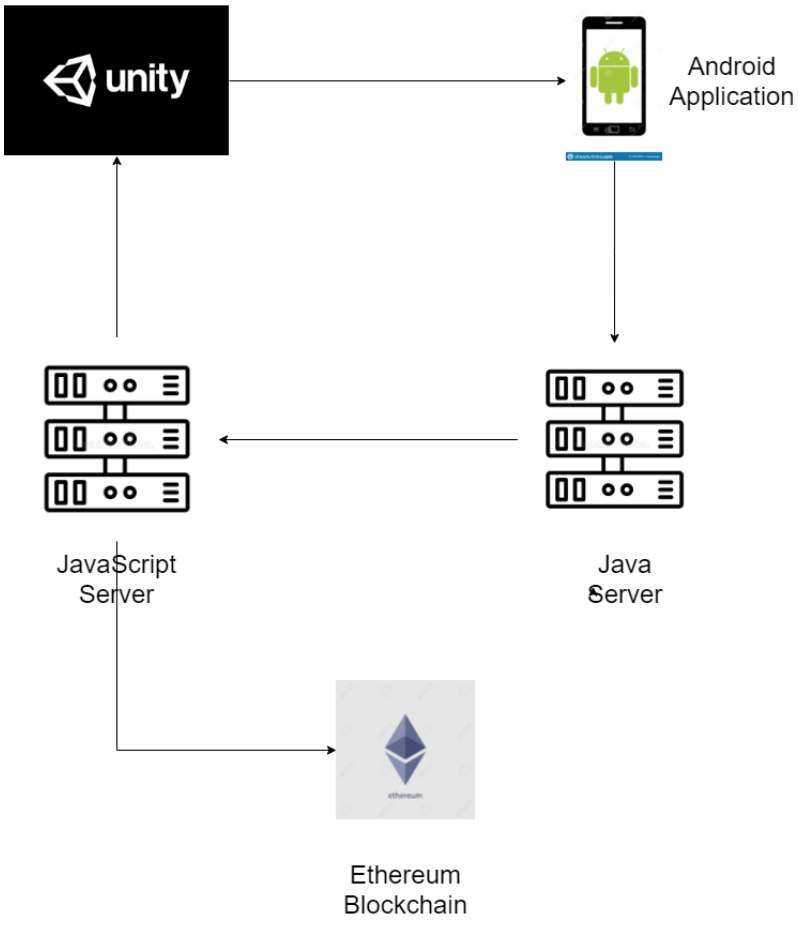
Dr. Georgios Polyzos

Athens University of Economic and Business

Fall 2021

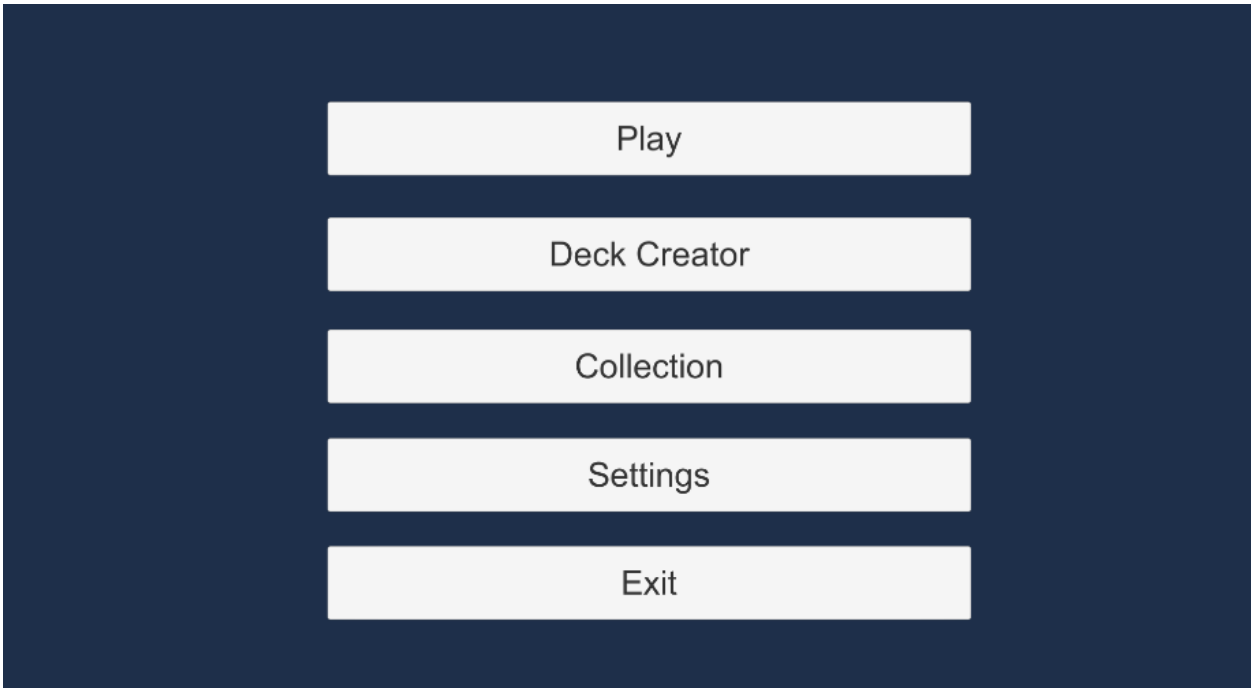
INTRODUCTION

The past year Ethereum and other crypto currencies have seen a dramatic increase in both their popularity and their value which has inspired many people to develop numerous applications based on the blockchain technology. We decided to explore the unknown field of Game development that utilizes this new tech and its features and create our own Mobile Card Game. The Card Game is developed on the unity game engine and is embedded inside an android application. The android application communicates directly with a java server that passes the user information to a JavaScript server. The role of the latter is to establish a connection with the Ethereum Blockchain and gather all the necessary information which it will then relay to the Card Game running on the android application.



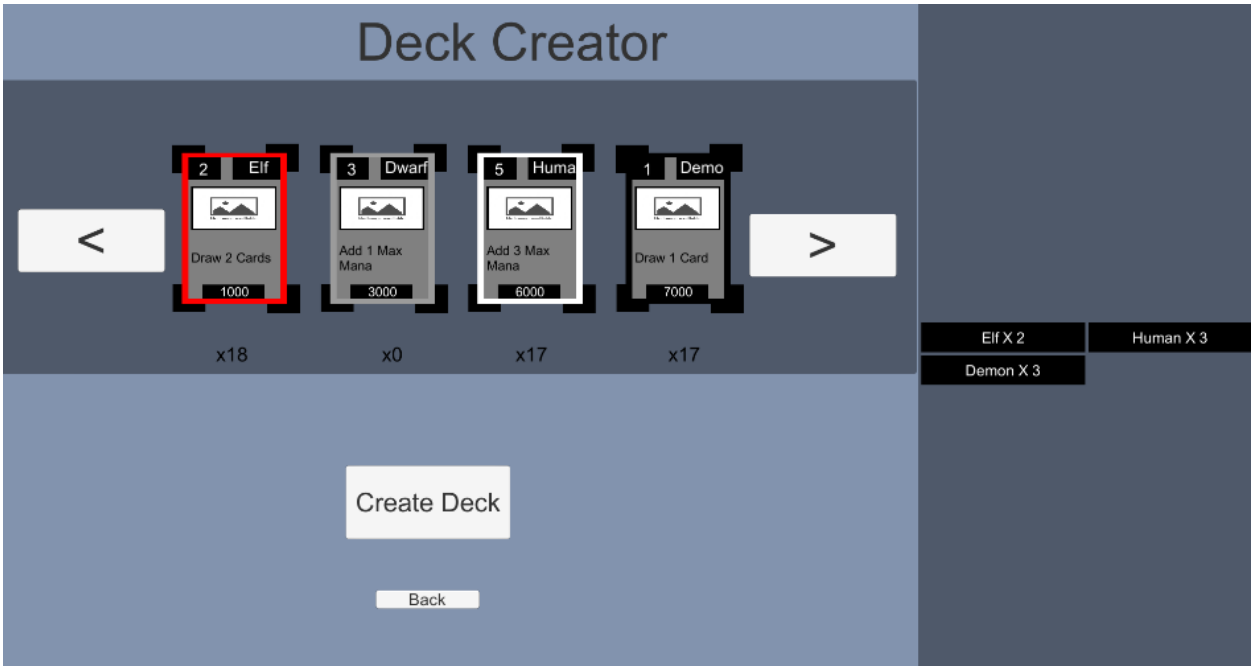
Card Game

The Card Game was made using the Unity engine and the C# programming language. Its contents are a Menu screen where you can choose what your next action.



In the background a connection is established with the JavaScript server and information about the cards that the user has is being transferred to Unity. If the user chooses the Collection option, he will be transferred to the following screen and will be able to view all of his available cards and browse through them. If a card is not owned, then its color is turned to Grey.

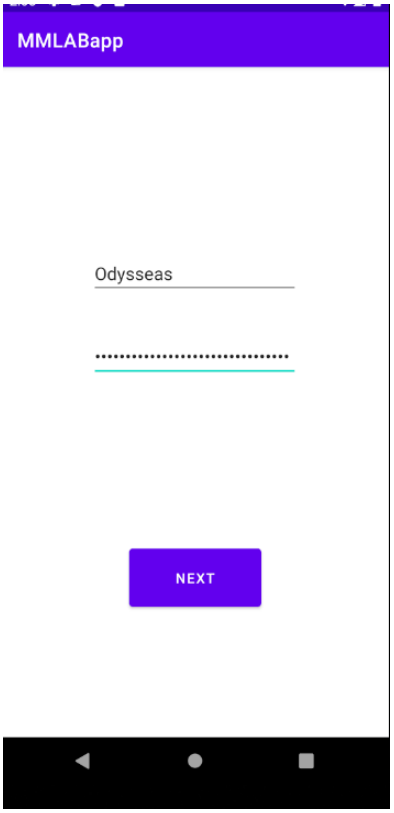
If the user chooses the Deck Creator option, he will be redirected to the Creation screen that will let him view his cards and drag them in order to create a 40-card deck. After he is satisfied with his choice he can click the create deck button in order to save his deck.



The last choice is to play a Game. The user will be able to play against an AI using its own deck. The goal is to eliminate the enemy life points while protecting your own. The deck used is the one created in the previous screen.

ANDROID APPLICATION

The Android Application contains the Unity Game and collects useful user information. More specifically, it collects his username and his 12-word mnemonic phrase that is used to connect to his Ethereum wallet. After collecting this information, it opens a socket towards the Java server and sends the information. Finally, it starts Unity and the games begin.



Java and JavaScript Servers

The Java server acts as an intermediate between the Android Application and the JavaScript server. The data sent by the Android Application is processed by the Java server and is forwarded to the JavaScript server that has created the server using the “net” module. The JavaScript server unpacks the information and isolates the user mnemonic phrase. Using that mnemonic phrase, it connects to the user’s wallet and interacts with the smart contracts on the Ethereum blockchain that give it all the necessary information regarding the cards that are in possession to the user. This information is processed and is turned into a dictionary that has all the card data collected. This dictionary is then sent to the Unity game to load the correct cards for the user. In order to establish this connection, we make use of the socket.io module.

Ethereum Blockchain

On the blockchain we have deployed smart contracts that resemble the in-game cards. More specifically, each contract is associated with a specific card and resembles a token. The type of the token depends on the rarity of the card. For example, common cards use the ERC-20 token interface which allows multiple copies of the same card to exist. On the other hand, rare cards use the ERC-721 interface which permits the existence of only one copy. These assets are also known as NFTs. The JavaScript server uses a view function that the contract contains in order to obtain the number of copies of a specific card/token.