Cryptomon

Logan Hoots, Collin Stratton

College of Science, Engineering, and Technology, Grand Canyon University

CST-461: Current Trends in Computer Science Lecture and Lab

Dr. Isac Artzi

April 20th, 2022

Cryptomon

The purpose of this assignment was to showcase the culminated knowledge gained in this course. For this project, we created a game that can access the blockchain to take advantage of its features.

# Project Description

Cryptomon is modeled after Pokemon, where each Cryptomon card is an NFT, and if a player loses a match, the ownership of the NFT is transferred. Players own unique Cryptomon cards and can battle them for ownership of other people’s cards or bet Ethereum. The game is a frontend that accesses the Cryptomon and smart contracts on the Ethereum blockchain. This improves security and validates interactions.

# Implementation Approach

The implementation approach for this assignment is as follows. First, the smart contracts for creating the Cryptomon NFTs were made. A couple of attempts were made the creation of these smart contracts. Initially, multiple contracts were created to create and store the Cryptomon and moves associated with the Cryptomon. Many issues arose with getting information to and from the multiple contracts. Thus, a single contract approach was utilized for creating and storing the Cryptomon and moves. Additionally, the contracts were written in an OOP format, but this was incompatible as Solidity does not follow all the OOP practices.

Once the functions required for creating and getting the Cyptomon and Cryptomon moves off the blockchain were created, the C# front end began to be developed. The game was straightforward to implement as the Pokemon format is pretty simple. Difficulties arose during linking the C# front end and the blockchain. First, research needed to be done to learn how to connect C# to the blockchain. This involved a testing phase in understanding how to connect them. Once research and testing were over to get the two connected, reading the data on the blockchain became the next step. This is where the connection implementation was ultimately scrapped due to the way information is gleaned off the blockchain by C#. Essentially, groups of data are encoded to the blockchain, and the tool used to get the information on the blockchain could not be decoded by C#. Thus, the smart contract writes to a file locally instead of reading off the blockchain, and the C# game reads the local file. Ultimately, this is not what would be implemented in the real world, but the blockchain is still being utilized for storing and trading NFTs, and the C# game can read that information.

# Tools Used

* Ganache: Local blockchain for testing
* Truffle: Testing and deploying smart contracts
* Nethereum: Connecting C# scripts to the blockchain
* Unity: Game creation

# Deployment

This project is deployed on a local network as we were having trouble working with C# and smart contracts talked about in a previous section. Furthermore, testing on the faucets on TestNets still yielded no coins to use for our project, thus making it difficult to be able to push the code onto the Ropsten TestNet. See in the Screenshots a photo of the compiled and deployed contract for this project.

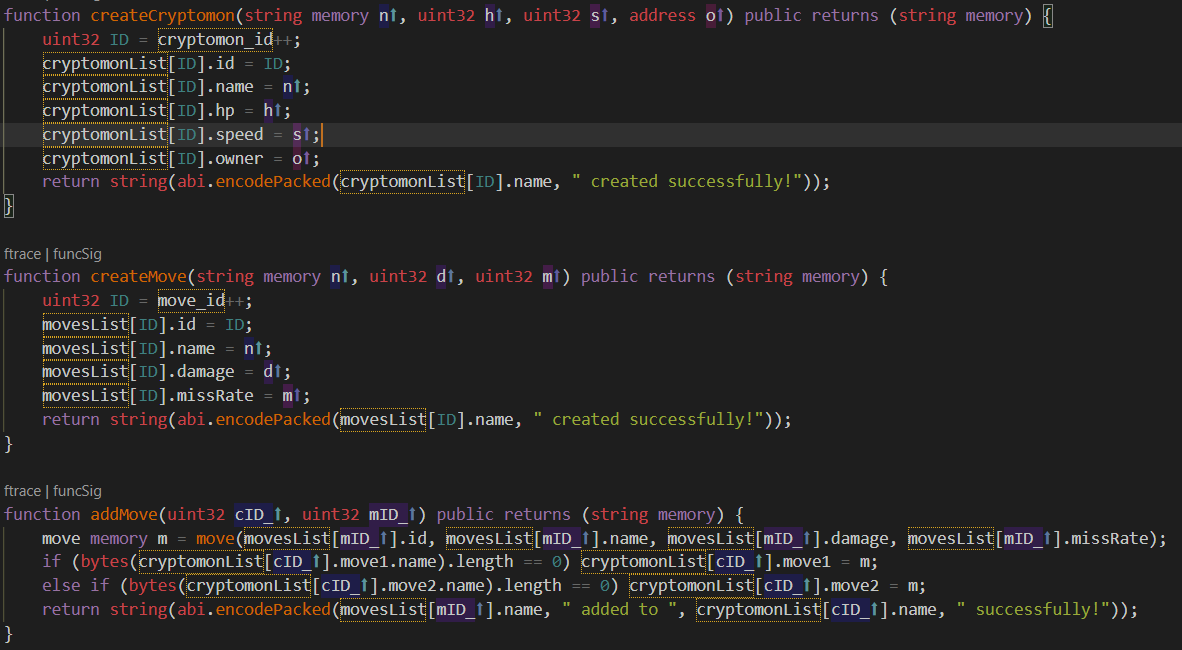
# How To Run

1. Launch Ganache
2. Update all Cryptomon Ownership in order to be deployed
3. Deploy Cryptomon contract
4. Launch Cryptomon Battle Hub
5. Input addresses associated with NFTs you wish to battle

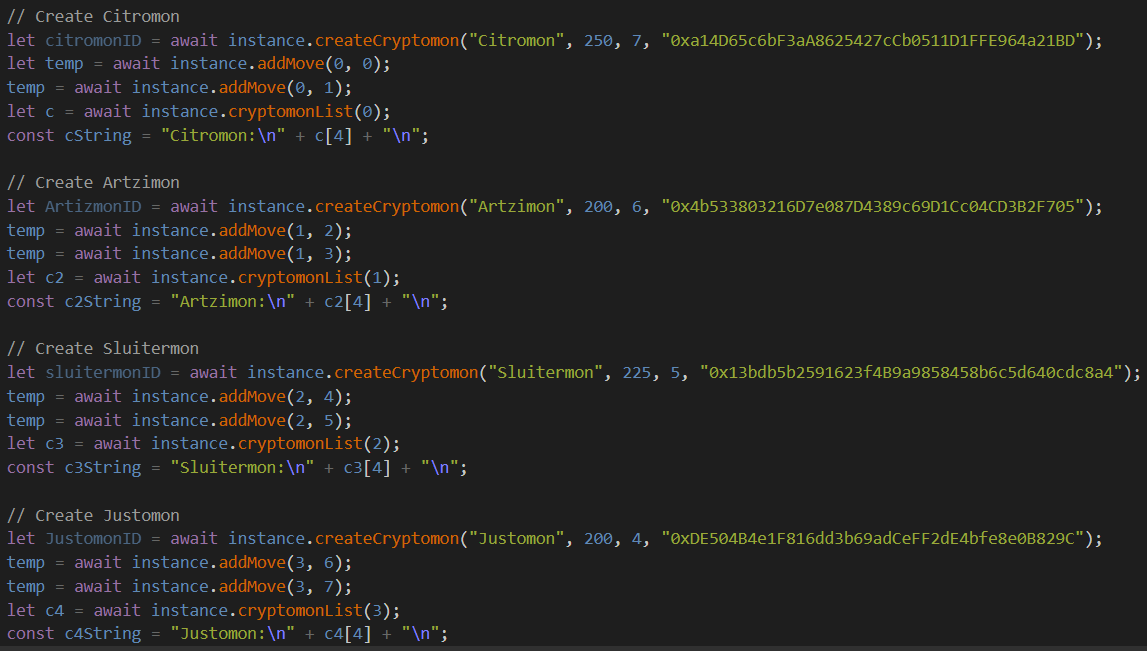
# How To Battle

1. Turn order is determined by the speed of the two Cryptomon that are battling. The faster a Cryptomon the sooner it will get a turn
2. On a Cryptomon’s turn, you are able to choose between the two available moves associated with said NFT. These are unique to each NFT.
3. Each move will have a damage number and a miss rate out of 100. When a move is selected, the move's miss rate is compared against a random number to determine if the move hits the opponent. If the move hits the opponent, the opponent’s hp is lowered by an equal number to the move’s damage.
4. When a Cryptomon’s HP is lowered to 0. The game ends and the Cryptomon existing with HP above 0 wins the battle.

# Screenshots



*1. Solidity Code Handling the Creation of Cryptomon and Moves*

**

*2. JavaScript Code Handling the Address Assignments to the Cryptomon*

**

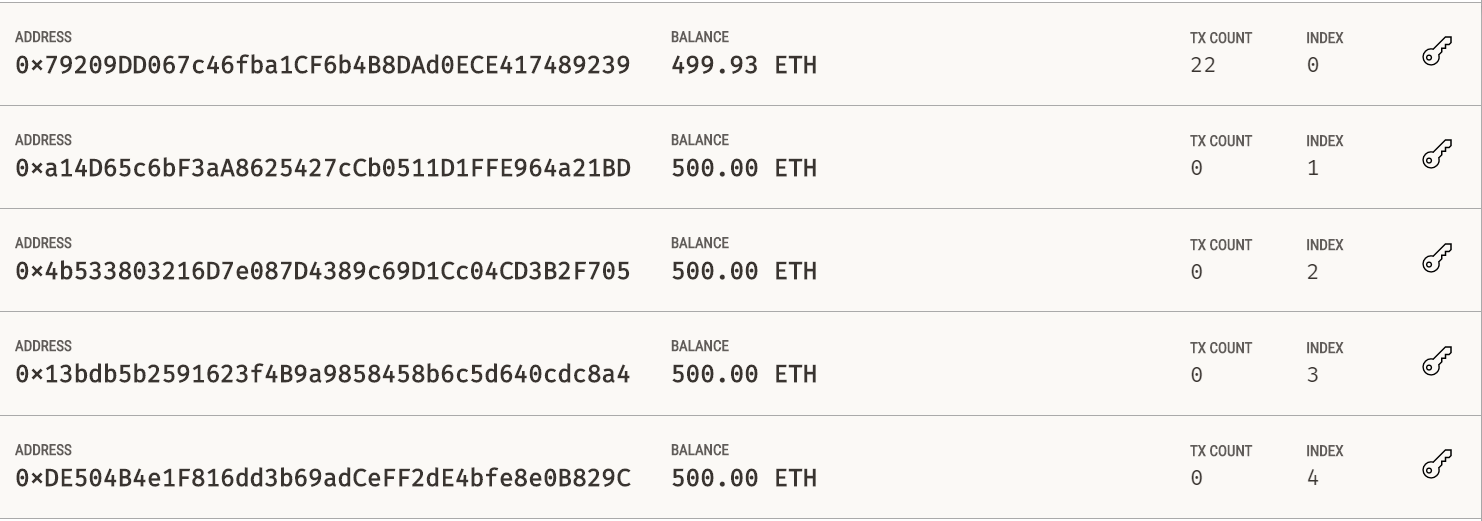
*3. C# Code to Read in Blockchain Information*

**

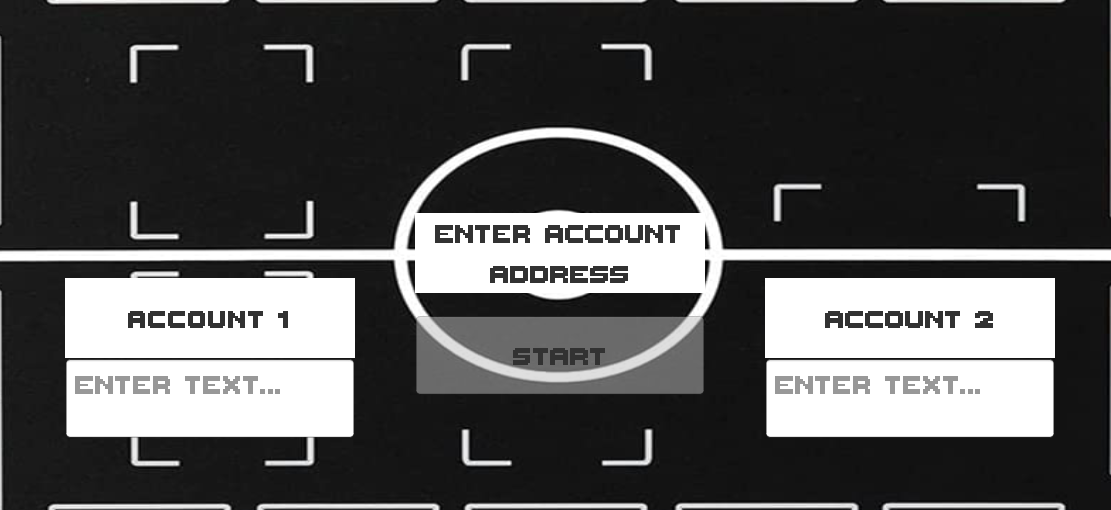
*4. C# Code to Interpret Blockchain Information*

# 

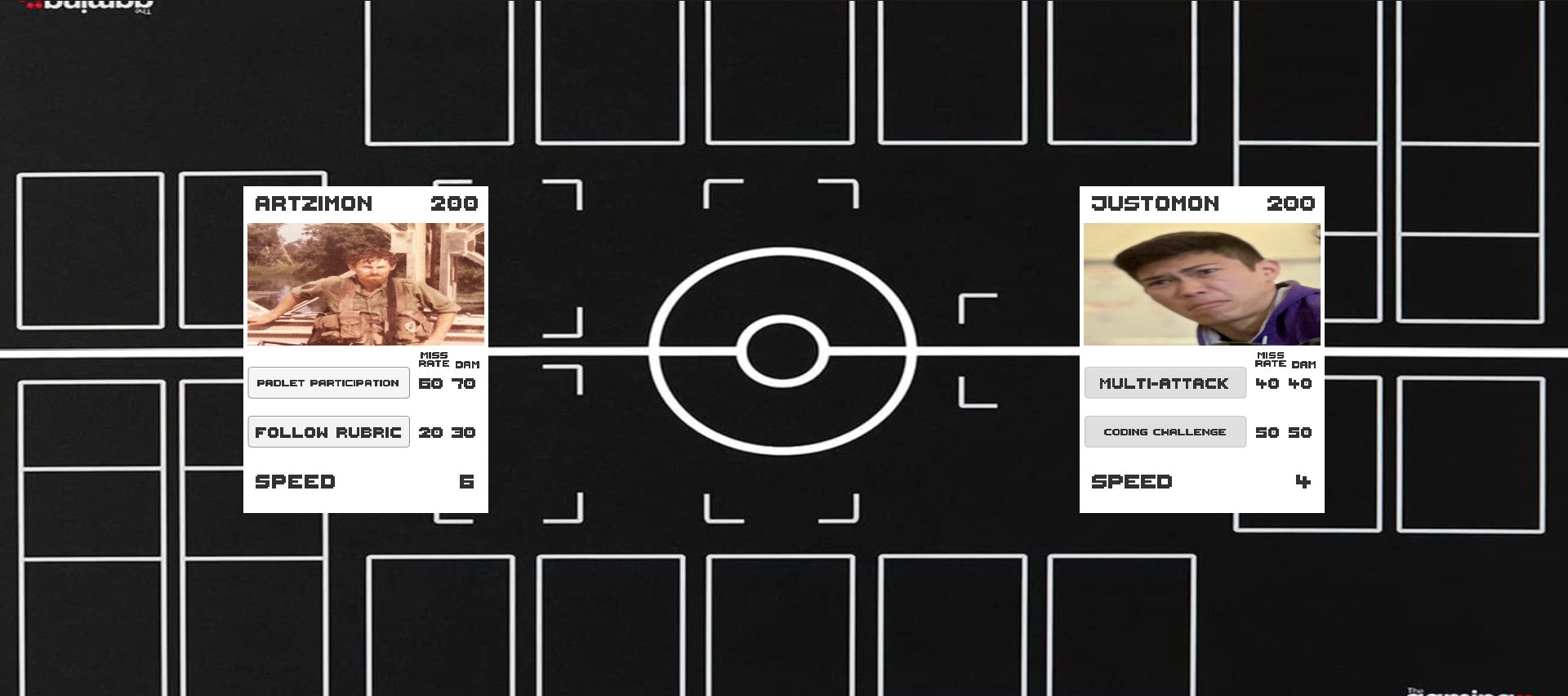
*5. Compilation and Deployment of Creator Smart Contract*



*6. Ganache Addresses that own the NFTs*



*7. Game Starting Screen*



*8. In-Game NFT Battle*



*9. Post-Game End Screen*

Resources

Blanco, J. (2018, June 13). *Nethereum 2.0.0 rc5 — Unity3d integration - Juan Blanco*. Medium. Retrieved April 16, 2022, from https://medium.com/@juanfranblanco/nethereum-2-0-0-rc5-unity3d-integration-eb7664664813

Community, N. (n.d.). *Calls, transactions, events, filters and topics (untyped) - Nethereum Documentation*. Nethereum. Retrieved April 16, 2022, from https://docs.nethereum.com/en/latest/contracts/calling-transactions-events/

Ethereum. (n.d.). *Ethereum for .NET developers*. Ethereum.Org. Retrieved April 16, 2022, from https://ethereum.org/en/developers/docs/programming-languages/dot-net/

Kenneth, 胡.H. (2021, December 11). *Interfacing .NET and Ethereum Blockchain Smart Contracts with Nethereum*. Medium. Retrieved April 16, 2022, from https://medium.com/my-blockchain-development-daily-journey/interfacing-net-and-ethereum-blockchain-smart-contracts-with-nethereum-2fa3729ac933

Small, K. (2018, November 9). *Unity and Ethereum: Why and How*. Raywenderlich.Com. Retrieved April 16, 2022, from https://www.raywenderlich.com/5509-unity-and-ethereum-why-and-how#toc-anchor-011

Truffle. (n.d.). *Running Migrations - Truffle Suite*. Truffle Suite. Retrieved April 16, 2022, from <https://trufflesuite.com/docs/truffle/getting-started/running-migrations/>

# Links

<https://www.loom.com/share/538f99be3cbc40698bc79ad481b61637>

<https://github.com/CStratton00/CST-461/tree/main/Cryptomon>