UPMpoly

João Diogo Martins Romão, Marius Hauenstein, Niccolò Revel Garrone

Introduction

Students from the Universidad Politécnica de Madrid (UPM) are excited to play a new game called UPMpoly, which is based on the popular board game Monopoly. In this version of the game, players will be able to visit different faculties at UPM and have the option to purchase them or pay a rental fee if they have already been bought by another player. Each player starts with a certain amount of money which they can use to buy faculties or pay rental fees to other players. If a player runs out of money, they will no longer be able to participate in the game and will be eliminated. When a player visits a faculty, they have the option to purchase it if it is unowned or pay a rental fee if it has already been bought by another player. The goal of the game is to accumulate as many faculties as possible and earn money through rental fees. Players must carefully manage their assets and make strategic decisions in order to emerge as the ultimate winner. Given the high level of competitiveness and lack of trust among the players, they have requested that the game be stored on a distributed ledger using Blockchain technology. We developed a smart contract for UPMpoly in a Hyperledger Fabric environment to ensure a fair and secure gaming experience.

Smart Contract Assets

In the UPMpoly game, there are two main assets: Players and Faculties. Players are the participants in the game and are responsible for buying and selling faculties, paying rental fees, and trying to stay in the game as long as possible. Faculties are the equivalent of streets in the traditional Monopoly game and represent different departments or units within the UPM.

Player

A player in the UPMpoly game is represented by the following attributes:

- id: a unique identifier for each player
- name: the name of the player
- credit: the amount of money left in the player's account
- isEliminated: a boolean value indicating whether the player is still in the game (false) or has been eliminated (true)

The player's id and name are used to distinguish one player from another, while the credit attribute keeps track of the player's financial standing in the game. If a player's credit falls to zero or below, they will be eliminated from the game and there isEliminated attribute will be set to true. Players who are still in the game will have a credit greater than zero and an isEliminated attribute of false.

Faculty

A faculty in the UPMpoly game is represented by the following attributes:

- id: a unique identifier for each faculty
- name: the name of the faculty
- sale price: the price at which the faculty can be purchased by a player
- rental fee: the amount that must be paid by a player when they visit a faculty that is already owned by another player
- owner: the player who currently owns the faculty

The id and name attributes are used to distinguish one faculty from another, while the sale price and rental fee attributes determine the financial aspect of the game. The owner attribute keeps track of which player currently owns the faculty. If the owner is null, it means that the faculty is not currently owned by any player and is available for purchase. If the owner is set to a player's id, it means that the faculty has been bought and the player must pay the rental fee to visit it.

Smart Contract Methods

The smart contract for the UPMpoly game provides a number of methods that players can use to interact with the game.

Method	Description
InitLedger()	Initializes the ledger with sample players and faculties.
Player(id, name, credit)	Creates a new player.
Faculty(id, name, salePrice, rentalFee)	Creates a new Faculty.
buyFaculty(buyer, faculty)	Allows a player to purchase a faculty if they have sufficient funds and the faculty is not already owned by another player.
payRental(faculty, visitor)	The visiting player has to pay the rental fee for a faculty owned by another player.
tradeFaculty(faculty, buyer, price)	Transfers ownership of a faculty from one player to another if the buyer has enough money.
getOwner(faculty)	Returns the id of the player who owns a specific faculty.
getMoney(player)	Returns the amount of money a player has left in their account.
isEliminated(player)	Returns a boolean value indicating whether a player has been eliminated from the game.
getPlayers()	Returns a list of all players who are still in the game.

Usage

Deployment

The deployment script for the UPMpoly game is an extension of the existing script that is used for the sample network. It adds a parameter called "polyupm". You also have to specify the path to the chaincode of the java project. This is done using the "-ccp" flag, for example: "-ccp /vagrant/upmpoly". The script first brings down the existing network, then brings it back up again. It then creates a channel called "mychannel" and deploys the java chaincode on this channel. This allows players to start interacting with the game and managing their assets using the methods provided by the smart contract.

Console command for “deployment-upmpoly.sh”:

./deployment-upmpoly.sh upmpoly -ccp /vagrant/upmpoly

Interaction

To interact with the UPMpoly chaincode, you can use a custom bash script. At the beginning of the script, you can define some variables such as player and faculty names and id's. If no arguments are provided, the script will execute a series of actions. The predefined actions include:

- Initializing the ledger to create sample players and faculties
- Creating a new player using the Player method
- Creating a new faculty using the Faculty method
- Displaying a list of all players
- Displaying a list of all faculties
- Using the buyFaculty method to have player3 purchase faculty1
- Using the payRental method to have player2 pay rent for faculty1
- Displaying a list of all players again to see how their account balances have changed
- Using the payRental method to have player1 pay rent for faculty1, but player1 has insufficient funds and gets eliminated from the game
- Displaying a list of all players to see their credits and status
- Using the tradeFaculty method to have player2 buy faculty1 from player3
- Displaying the owner of faculty1 using the getOwner method
- Displaying the money of player2 using the getMoney method
- Displaying the status of player1 using the isEliminated method

This is just one example of how you can use the available methods to interact with the UPMpoly chaincode and play the game

Another way to use the script for interacting with the UPMpoly chaincode is to call the individual methods separately, rather than executing a series of pre-defined actions. This allows you to have more control over the game and make specific, targeted changes to the ledger. For example, you might use the Player method to create a new player, the Faculty method to create a new faculty, the

buyFaculty method to have a player purchase a faculty, or the payRental method to have a player pay rent for a faculty that is owned by another player.

Using the individual methods also allows you to check the current state of the ledger at any given time. For example, you can use the getPlayers method to see a list of all players who are still in the game, or the getOwner method to see who owns a specific faculty. You can also use the getMoney method to check how much money a player has left in their account, or the isEliminated method to see if a player has been eliminated from the game.

Method	Example
No Method	./upmpoly.sh
InitLedger()	./upmpoly.sh InitLedger
Player(id, name, credit)	./upmpoly.sh Player player5 Messi 20000
Faculty(id, name, salePrice, rentalFee)	./upmpoly.sh Faculty faculty5 Physics 2000000 200
buyFaculty(buyer, faculty)	./upmpoly.sh player3 faculty3
payRental(faculty, visitor)	./upmpoly.sh faculty3 player2
tradeFaculty(faculty, buyer, price)	./upmpoly.sh faculty1 player2 200000
getOwner(faculty)	./upmpoly.sh faculty1
getMoney(player)	./upmpoly.sh getMoney player2
isEliminated(player)	./upmpoly.sh isEliminated player1
getPlayers()	./upmpoly.sh getPlayers