

# UPMpoly

– Blockchain and Services for Fintech Enterprise Integration –

## Introduction

Students from two different faculties from the UPM want to play **UPMpoly**. This game is a simplification of the famous board game **Monopoly**, in which you have UPM **faculties** instead of streets. When a **player** visits a faculty, there are two options:

- a) The faculty is **free**, and the player can opt to buy it.
- b) The faculty has already been **bought** and the player must pay a rental fee for the visit.

Each player owns an amount of money with which he/she can buy new faculties and pay rental fees. While a player still has some money, he/she is considered to be **playing**, while when he/she runs out of money, he/she will not be able to do anything else and will be considered as an **eliminated** player.

As the competitiveness among the players (and even more among players from different faculties) is very high and they do not trust each other, they have proposed to store the status of the game in a distributed ledger by using **Blockchain** technology.

Therefore, they have asked you to develop a smart contract to play UPMpoly in a Hyperledger Fabric environment.

## Development

This that in mind, we are asked to develop the Java classes that you need to model this game, in which there will be **two assets** to store in the ledger: players and faculties.

Moreover, you need to implement **one single smart contract** which must include the following methods (exactly with the same names and following exactly the same order for the parameters as the one specified here):

- **newPlayer** – It creates a new player of the game. It needs the player number, the name, and the initial amount of money as parameters.
- **newFaculty** – It creates a new faculty. It needs the faculty ID, the name, the sale price, and the rental fee as parameters.
- **buyFaculty** – It changes the ownership of a faculty and it consequently reduces the owned amount of money of the buyer. It needs the player number and the faculty ID as parameters.
- **payRental** – It subtracts the rental fee of a faculty to the visitor and adds it to the owner. It needs the faculty ID, the player number of the owner, and the player of the visitor as parameters.
- **facultySale** – It changes the ownership of a faculty to a new owner by paying a specified sale price. It needs the faculty ID, the player number of the old player, the player number of the new owner, and the sale price as parameters.

- **printMoney** – It requests the amount of money of a given player from the ledger and prints it in the prompt. It needs the player number as parameter.
- **printOwner** – It request the owner of a specific faculty from the ledger and prints his/her name. If the faculty is free, it must print “Free”. It needs the faculty ID as parameter.
- **printPlayers** – It request all the names of the players that have not been eliminated from the ledger and prints them. It does not need parameters.

You must take into account that when a **player runs out of money**, you must set all his/her faculties as free and he/she will not be able to operate any more.

## Deployment

You must deploy your smart contract using the **test-network** of the “fabric-samples” software repository, with two organizations, one peer per organization, and an orderer.

You must also prepare two scripts:

- One that includes the deployment of the smart contract (it must bring down the network and then, bring it up again, create the channel, and deploy the chaincode).
- The second script must provide several examples of calls to the methods of the smart contract, demonstrating the functionality of all the functionalities.

## Other details

The assignment must be done in **groups of 2 students**.

A single ZIP file that contains the source code, the deployment scripts and a short report that explains your work (no more than 5 pages). The file must be uploaded to the task that will be settled up on Moodle before **January 8th, 2021 at 23:55**.

As you need to present to work, the presentation will be done the same day of the final exam, before starting it (January 13th, 2021, at 15:00).

You can ask any doubt to Antonio Jesús Díaz on Teams or by e-mail: [antoniojesus.diaz@upm.es](mailto:antoniojesus.diaz@upm.es)