# Blackjack Project

## Algorithms and python programming

Students: Abdelkoddous, Vasileios Skarleas

**Introduction:**

On the UE course of INF101 under the general idea of learning algorithms and programming in Python, we are forced to develop a blackjack game by using Python and following the different programming techniques as well as skills that we developed during this semester. Below are written the indications that we needed to follow, the procedure that we followed in order to resolve any problems, as well as a summary of the problems / difficulties that we met. Our goal through this project is to make it even more clear how to connect many different functions in a Python programming environment while we are staying focused on the indications given by the project’s paper.

**Indications:**

We need to create a blackjack game by using the Python programming language. The project is divided into 4 different sections, where, by combining them we have a completed result. More specifically:

1. Initialisations
2. Game Management
3. Some intelligence
4. UI (graphical interface)

We note that parts A and B are directly connected the one to other, while C provides more functionality onto the part B, and finally the part D is a complete separate task from all the others in terms of functionality.

As blackjack is a cards game, the creation of deck or decks of cards is essential, as well as the possibility to count the different points gained from every player according to the following criteria:

* Cards numbered from 2 to 10 are valued as many points as their number (so between 2 and 10 points)
* The ace is valued either 1 or 11 points, at the player’s choice
* Figures (jack, queen, king) are valued 10 points each

The data structure provides two different options so that we can play several games in a row. This includes the management of many different players’ scores. This is why it’s preferred to use lists or dictionaries for saving and retrieving data.

* Option 1: scores, number of victories, and money gains of each player are stored in lists, so they can be accessed from the player number (index in the corresponding list).
* Option 2: scores, number of victories, and money gains of players are stored in dictionaries whose keys are the players’ names, so they can be accessed with the player’s name instead of the player’s number.

**Procedure:**

As team we decided to work to the corresponding workflow below:

* Organization, transparency and communication are our standards for a good team work
* Around the above mantra, we used the softwares mentioned below:
  1. GitHub: Includes our private repository with all the code that has been submitted by the team members. It tracks the live versions that every team member is watching every time on his personal desktop workstation. Also, GitHub provides a great way to visualize any changes and to restore previous versions of the code.
  2. VSCode: Instead of the Python idle, we decided to load python and work with this programming language, in our TD, TP, Caseine and current project, on VSCode since it is a universal development app, with some great extensions to integrate onto the workflow and to visualize better the different commands. In addition to that, it offers a direct connection to our GitHub repository.
  3. PythonTutor: Online program service indicated by the UE that helps to the deeper understanding of the corresponding code that we are examining every single time.
* From the first view of the project, we knew that the part D would be the most difficult one and there was a possibility that we couldn’t be able to make it so far. So, our goal was set to complete as better as possible the rest of the tasks.
* For every step on the different four sections mentioned above, we decided to follow the “exams technique”. This means, that we will follow the exact steps by including the required elements mentioned on the specific task (function, returned value, required arguments, lists, etc.) as part of project’s understanding. The rest of the code will get completed according to the personal point of view through the understanding process of the task. This will allow us to have the biggest understanding that is possible, for this demanding project.

**Difficulties/problems:**

In this section we are presenting the different challenges that we met and the workflows around them in order to be solved:

1. The biggest difficulty that we had was the concept the idea of the game. Since we had never played that game, nor having experience with these types of games, it was extremely difficult to understand what we are expecting from the program to do, something that corresponds at how will combine all the functions that we initialized on part A and result at least at one game play on part B. On the other hand, the UNO game for instance that we have in our copies, was a game that we know how it works and what changes to do in the program in order to make it work correctly.
2. One of the first challenges was the introduction and use of dictionaries. The moment that we started the development of this project, we didn’t have done it in our course. Thanks to Python’s documentation, the different internal functions() of dictionaries in pythons and key words like “keys” and “values” were used as expected on the different sections (like initPlayers()).
3. Another challenge was the initialization of cards’ deck. According to different resources, to most direct way is to use classes in Python which will then also help to the UI (fourth part of the project) by calling the executing the specified task. But we wanted a more direct way that is closed to our recent knowledges in python, and that’s why in our workflow we used lists to initialize the deck.
4. It was challenging to write the drawPick() function as long as we don’t know how a real blackjack game is played. This is also connected with the gameTurn() and completeGame() functions. YouTube was a very helpful tools in this understanding and searching procedure but again, without the real-life experience, we think we did not succeed.

**Project’s code:**

We are attaching our project’s code divided into the different sections of the project mentioned on this document’s introduction. Unfortunately, we couldn’t realize a satisfying result for the section D of project’s parts even if we tried hard.