Team Members – Madhav Maheshwari (2022MT61975)
Yashu Agarwal (2022ME22066)

The goal of our project is to create a working business-game which can be played via the terminal and is built by our team (AltF4) using python using the concepts of Object-Oriented Programming.

Our code includes the concepts of OOPS like inheritance, creating classes and building their instances.

The rules and description of the game can be found in the link provided below: -

https://drive.google.com/file/d/13jPje4dAEnF1cwIXQVL64QhupPGfjkLR/view?usp=sharing

The image of the business-game board can be found in the link provided below: -

https://drive.google.com/file/d/1V730eOay0SWmHHxH8RrKKFyxC3mPJPLp/view?usp=sharing

Below are the steps which we followed:

Step 1: Define Game Player Class. This class encapsulates a player's properties like name, id, player money, properties and its current position on board.

Step 2: Define a Parent class 'Site' to create children classes like Properties(cities), Transportation and Services (using inheritance). These classes include methods like buying a property, buying a house if possible, whether it is owned or not, who is the owner and finally, interact which helps us to interact with the users.

Step 3: Defining classes for Jail, resthouse, club,etc

Step 3: Defining board class:

1. The board contains all Sites which would be the instances of the above mentioned classes.
2. Chance and Community Chest classes to be created and used in the board as four positions on the board are filled by these only.

Step 4: Defining all the possibilities that the Chance and Community Classes can have and the necessary functions they must have in order to change the game environment according to that possibilities.

Step 5: Define the Two Dices and their random rolling, for loop which will iterate to give each player their turn until a player gets bankrupt and this maintained by a while loop.

BONUS:-

Step 6: Define attributes of notes in the player class in a dictionary or list, Making the functions for different possible denominations,
PositiveDenomination -> Bank to Player
NegativeDenomination -> Player to Bank
PlayerPlayer -> Player to Player

Step 7: Adding the combination of payment when the game starts and whenever the player passes the “Start” as per mentioned in the game rules.

Challenges we faced:

Challenge 1: Including the condition of first roll only when greater than 9 in Step 5.

Challenge 2: Linking all the classes to make it function properly

Challenge 3 : Including some conditions as mentioned like the player numbers would be between 2 and 4, and if two players has the same name then providing them a unique ID.

BONUS:-

Challenge 4: Including the condition of max possible denomination in Player-to-Bank transaction, condition of min possible denomination in Bank-to-Player transaction.

Challenge 5: Including the condition of max possible denomination in Player-to-Player transaction which is very different as we have to change currency notes of both players and if the player do not have the required combination of notes, then bank has to interrupt and fulfil the transaction by giving smaller currency notes by taking larger currency notes.