

Arash Nassirinia and Fatih Ak

Command Prompt Monopoly README

- Python 3.5
- N/A
- Open up the file in interactive mode in command prompt (python -i (name of file))

Complexity: The project was relatively complex, in part because we did it in python and had to learn a lot of features in order to be able to implement and display everything. In it, we use lists, dictionaries, variables (global,script), while loops, and recursion in order to make the project all work.

The project is called Command Prompt Monopoly because the user opens up command prompt, runs the python file, and is presented with this textual representation of the board:

Free Parking	Kentucky \$240 X 0H	Indiana \$240 X 0H	Chance	Atlantic \$280 X 0H	Ventnor \$300 X 0H	GO TO JAIL
New York \$200 X 0H	P1 Money:\$1500				P2 Money:\$1500	Pacific \$350 X 0H
St.James \$190 X 0H	WELCOME TO MONOPOLY BY Fatih Ak and Arash Nassirinia					N Carolina \$360 X 0H
Com. Chest						Com. Chest
Virginia \$160 X 0H						Park Place \$370 X 0H
States \$140 X 0H						Boardwalk \$400 X 0H
Visitng	Vermont \$110 X 0H	Oriental \$100 X 0H	Chance	Baltic \$60 X 0H	Medit. \$60 X 0H	GO Get \$200

This is the monopoly board we implemented in python. Each property name is listed along with its price. The number of houses owned are indicated by the number preceding the H

(this is a picture of the beginning of the game so each property has 0 houses). The player that owns the house is also presented. If there is an X, it means neither of the two players own the property and if one of the players owns the property, the X is replaced by either a 1 or a 2 depending on which player bought it. The balance of each player is also displayed in the top left and right hand corners of the inner rectangle.

Each player starts at GO (position 0) with \$1500 in their balance (stored in p1). Players start their turn by rolling a set of dice by pressing r and enter (roll and its variations are also acceptable):



After rolling the dice, the player is presented with two randomly chosen values from 1 to 6 which are also displayed on the dice (stored in the global variables d1 and d2). After the player has rolled the dice, they are told where they have landed. If this is a property, it tells them that either it is unowned, giving them the option to buy it, or that it is owned and they have to pay rent to the other player (the rent is deducted from the player's balance). There is a list that stores the properties that are owned by the players (p1[3]). If they land on chance or community chest, a card is chosen at random and presented to them (from the dictionary "chance" that has multiple cards stored). Players can lose or gain money (their money is stored in a list for each player, p1[1] and p2[1]), or be sent to GO based on the card they got, which are based on the original Monopoly cards. If they land on jail, they become inactive for three turns. They must roll doubles or pay \$50 dollars to leave jail, like in the classic Monopoly game. If they are in

Free Parking, Visiting, or Go (in which case they receive \$200), nothing really happens and they are told where they are and that they cannot buy the property. After all of this, the players have a set of options that they can choose from:

```
Would you like to Mortgage one of your properties (m), Unmortgage a property(u), Build a house(h), or End your turn(e)?
```

This pops up at the end of each turn and gives the player the option to mortgage and unmortgage their properties or build houses in any of the properties they own. Mortgaging checks the list of the properties you own and asks which of those you would like to mortgage. If you mortgage a property, it adds it to an additional list (p1[4] and p2[4]), allowing you to also unmortgage these properties. When you mortgage, you gain half of the original price of the property but the other player does not pay rent if they land there. When you unmortgage, you have to pay the full price of the property, but you are able to keep your houses. If they choose to end their turn, the next player goes. Another function that we implemented is doubles, which allows you to roll again if you rolled a double (it checks the d1 and d2 and runs again if they are equal). Otherwise, the next player goes and has the same options while they play. A function called other_player is used to proceed to the next player's turn using the same function used for the first player's turn. The Player function uses recursion and keeps calling itself until a player reaches negative balance at the end of their turn. If a player has negative balance at the end of their turn, they are told you can mortgage properties to increase money. If the player has no properties to mortgage or chooses not to, they lose and the other player wins, as displayed in the command prompt.