

Ady

1. The problem the group member set out to solve:

I was looking to solve the problem of how the program could find whether or not a player has a monopoly on a specific color of properties (or railroads or utilities) and then return the appropriate rent multiplier to be assigned to the other player.

2. A few sentences about the main things you figured out (be concise):

I figured out how to better integrate attributes from other classes into a function outside of the class; for example, in my code, I had to use specific attributes from the Property and Player class in my mp_check function outside of those classes. I also learned how to improve my utilization of dictionaries, as I realized I could just use one instead of a text file to represent the total numbers of properties per color. I also worked through quite a bit of logic in the if-statements to ascertain the appropriate rent multipliers for each specific situation. Finally, I also learned how to use the `if __name__ == "__main__"` 'magic method' more practically in my testing.

3. Which file contains your solution and any instructions on how to run the solution:

File: checkin_1.py

I created incomplete, temporary classes to test this specific method (e.g., Player, Property, etc.)

Instructions: Control+A the entirety of the file, and then shift+enter.

Anshu

1. The problem the group member set out to solve:

Determine whether or not the computer player should pay to get out of jail.

2. A few sentences about the main things you figured out (be concise):

First, my program rolls a pair of dice three times, and if any of them are matching pairs, then the program stops and the computer player gets out of jail for free. If after three rolls, none of the outcomes were a matching pair, then there are different outcomes based on the difficulty. If the game is intelligently played, then \$50 bail will only be paid if the player's balance is over \$50. Otherwise, with a difficulty of 0, the decision to pay bail is random.

3. Which file contains your solution and any instructions on how to run the solution:

Check_in_1.py I created a temporary class to test the method. The only module needed is import random.

Emily

1. The problem the group member set out to solve:

The problem I set out to solve was how the computer would decide if it should buy the property it lands on or not if the property is available.

2. A few sentences about the main things you figured out (be concise):

We decided to do a low difficulty and a high difficulty for the computer player. For low difficulty I decided for it to be random whether the computer purchases the property or not. For high difficulty, I decided to evaluate how much money the computer has before purchasing the property.

3. Which file contains your solution and any instructions on how to run the solution:

The check_in1.py python file has my solution. The method will need the random module which is imported.

Brian

1. The problem the group member set out to solve:

The problem that I set out to solve is how a computer player decides which properties to mortgage or unmortgage.

2. A few sentences about the main things you figured out (be concise):

I figured out, for the most part, how a computer player is going to decide which properties they will mortgage and unmortgage. I also learned that a dictionary is not going to be ideal for holding each property's names and values. Pandas might be a much better option.

3. Which file contains your solution and any instructions on how to run the solution:

My solution is in the file called "checkin_mortgage.py" and it imports the random module. I created a temporary class, so the functions can run as if they were in our program. This file can run by itself, to test it create an instance of the class, specifying the difficulty.