

Homework 1 Summary

In my design I was focusing on implementing OOP concepts. It was the most interesting part and the hardest part because it's my first big project with pure OOP. Also I was trying to make my design modular so functions won't be very big and have undefined purpose.

I choose to implement classes Farkle, Board, Die and Hand because they represent objects from real life. It was the most reasonable way I found so every class will have it's purpose an

Farkle doesn't contain any specific logic except running the game. Die represents game dice and it's logic.

Hand is the main class. It keeps all info such as dice in the hand, dice in the bank, number of each value of dice in the hand and in the bank. Also, it has the logic to run all main processes in the game like rolling the hand, printing interface or calculating players score.

The design of the countBank function took the biggest time to complete. In the beginning I wasn't sure that I can fit it in reasonable amount of lines, but then I realized that I can combine most cases, so it will be much easier to implement and it will be more readable, so I splitted this function into one special case: straight, then I added functions to cover cases when player have single five or single one, then I made a function which is looking for a combination with three pairs, and finally I made a function to look for 3+ combinations, with an edge case when player has a combo of ones, because they have different formula to count the score.

The biggest issue was the scoring function which I described above. It took me some time, but I've managed to split it into separate functions, so I can cover all cases in relatively small functions.

Another issue was OOP in general. I'm used to goLang, where OOP is much simpler so I took this experience as base while I was designing my program.

If I had more time I definitely would transfer bank related functions to the Bank class and score to the User class. I'll return to this idea when I'll need to implement a second player and more rounds.

