Math121 Project 2 instructions Spring 2023

Submit materials to “Project 2” on Gradescope.

## Files to submit

* poker\_solitaire.py (and any scripts that it imports, not counting graphics.py)
  + Python program for the poker\_solitaire game described below, written in an Object-Oriented Programming (OOP) design.
* test\_poker\_solitaire.py
  + Python script to test classes that you have defined in poker\_solitaire.py.
  + Test every non-trivial method in your class. (A trivial method is where it’s pretty obvious that it works just by looking at the code. For example, if the goal of the method is to return the value of an instance variable. You also don’t need to separately test the \_\_str\_\_ method since you’ll use it in your other tests.
  + The output of this script should be self-evident and informative. Each test should specify what is being tested, how it’s being tested, and the result of the test.
* Proj2\_reflection.pdf
  + A text file describing your process for designing, implementing, and testing your program. Review Chapter 12 for useful frameworks on the subject.

## Learning objectives

To successfully complete this project, you need to be proficient in Course Learning Objectives (CLOs) 1-4. The main learning objective for this project line up with the fifth Course Learning Objective:

CLO-5. Develop and implement programs using the principles of object-oriented design

For both Python scripts that you’ll hand in, here are the basic requirements:

* Introductory comment specifying the author and the intent of the script.
* All **import** statements happen right after the introductory comment. (Whatever modules or libraries you are importing: those import statements should happen right at the beginning of your script.) I expect that you wouldn’t need more than the random module for this project.
* After the introductory comment, the organization of your script is as follows:
  + All import statements that appear in the script
  + All class definitions
  + All function definitions, including the main() function
  + And then, at the very end of the script, the only other top-level executable command:

if \_\_name\_\_ == ‘\_\_main\_\_’: main()

* No use of global variables. All variables in your functions must either be class variables, instance variables, passed in as parameters, or be locally defined within a function or method.

I expect the following from your code:

* Code is clear and easy to follow. Uses reasonably descriptive variable and function names and informative comments.
* Good use of classes. It should be clear from your design why it makes sense to consider some process as an object, and I expect you to elaborate on that in your reflection.
* The constructor method specifies all the instance variables, even if they are fully determined in a different method later.
* Clear class methods. Keep each method's code easy to understand and at reasonable length. The purpose of each method is made clear through an informative comment at the beginning of that method. All variables used within the method are either passed as parameters or are defined locally.
* Reasonable use of self-defined functions. In an Object-Oriented Programming design, much of the action happens via class methods. When you find yourself writing a function to do something with an object, ask yourself if that’s something the object should know how to do (that is, whether that should be a class method). If the answer is no, then by all means write a function to do the work (and include your thoughts in the reflection).

# Poker Solitaire

This game is a solitaire (one-person game) version of Texas hold’em poker. From Wikipedia (edited for our use):

Two cards, known as “hole” cards, are dealt face down to each player, and then five community cards are dealt face up in three stages. The stages consist of a series of three cards ("the flop"), later an additional single card ("the turn"), and a final card ("the river"). Each player seeks the best five card poker hand from any combination of the seven cards: the five community cards and their two “hole” cards. Rounds of betting take place before the flop is dealt and after each subsequent deal. The player who has the best hand and has not folded by the end of all betting rounds wins the round.

The game is played with a standard deck of 52 cards, which in text we represent as:

2♣ 3♣ 4♣ 5♣ 6♣ 7♣ 8♣ 9♣ T♣ J♣ Q♣ K♣ A♣

2♦ 3♦ 4♦ 5♦ 6♦ 7♦ 8♦ 9♦ T♦ J♦ Q♦ K♦ A♦

2♥ 3♥ 4♥ 5♥ 6♥ 7♥ 8♥ 9♥ T♥ J♥ Q♥ K♥ A♥

2♠ 3♠ 4♠ 5♠ 6♠ 7♠ 8♠ 9♠ T♠ J♠ Q♠ K♠ A♠

(The suit symbols are the same we did in class; Microsoft Word adds color, but a simpler text-based display will not. That’s fine.)

When you play one game, the deck is shuffled so that the cards are randomly dealt out, and the same card cannot be dealt twice in the same game.

## Rules of the game:

The objective for each game is to predict whether you’ll win against one (automated) dealer. If you are going to lose, then the objective is to “fold” as soon as possible. If you are going to win, then the objective is to “stay” until the end.

The game proceeds as follows:

* (Betting round 1) First, you and the dealer both get two cards. Those are your two “hole” cards. You can see your cards (the are “face up), but not the dealer’s cards (they are “face down”). You get a chance to bid (“stay” or “fold”). If you stay:
* (Betting round 2) Three more cards are dealt “face up”. Those are “the flop”. You get a chance to bid (“stay” or “fold”). If you stay:
* (Betting round 3) One more card is dealt “face up”. That’s “the turn”. You get a chance to bid (“stay” or “fold”). If you stay:
* (Betting round 4) One last card is dealt “face up”. That’s “the river”. You get a chance to bid (“stay” or “fold”). If you stay:
* The dealer’s “hole” cards are revealed, and your hand is compared to the dealer’s hand according to the standard rules of poker. Whoever has the best hand wins. In an unlikely event of a tie, nobody wins.

If you stay all four betting rounds:

* If you win, you get 100 points—all your bets were correct for the occasion!
* If you lose, you get -100 points—all your bets were not correct this time around.

If you fold on any of the betting stages, then the dealer’s “hole” cards are revealed, the rest of the five community cards are dealt, and the would-be winner gets determined.

* If you would have lost, you get 100 points if you folded on the 1st betting round, 75 points if you folded on the 2nd betting round, 50 points if you folded on the 3rd betting round, and 25 points if you folded on the 4th betting round. The sooner you realized that you should fold, the more points you get!
* If you would have won, you get -25 points if you folded on the 4th round (and stayed before then), -50 points if you folded on the 3rd round (and stayed before then), -75 points if you folded on the 2nd round (and stayed before then), and -100 points if you folded on the 1st round.

After each game, the player’s average score gets displayed, and the player gets a chance to play again. The cards that got played get put back into the deck, and the deck is reshuffled before each new game.

## Comparing hands

Here are the categories, from highest to lowest:

|  |  |  |
| --- | --- | --- |
| **Category name** | **Description** | **Example** |
| Straight flush | The hand contains five cards that are of the same suit and whose ranks are in order | 8♦ 9♦ T♦ J♦ Q♦ |
| Four of a kind | The hand contains four cards of the same rank | 2♣ 2♦ 2♥ 2♠ |
| Full house | The hand contains three cards of one rank and two cards of another rank | 2♣ 2♦ 2♥ Q♣ Q♦ |
| Flush | The hand contains five cards of the same suit | 3♠ 6♠ 7♠ T♠ K♠ |
| Straight | The hand contains five cards whose ranks are in order | 8♥ 9♠ T♦ J♣ Q♠ |
| Three of a kind | The hand contains three cards of one rank | 2♣ 2♦ 2♥ |
| Two pair | The hand contains two cards of one rank and two cards of another rank | 2♣ 2♦ Q♣ Q♦ |
| One pair | The hand contains two cards of one rank | 2♣ 2♦ |
| High card | None of the above. |  |

This game uses simplified rules. Whichever hand achieves the higher category wins. With exception of “One pair” and “High card”, if both hands have the same category, the game is a draw. If that same category is “One pair”, then the hand whose pair has higher rank wins (e.g., Q♣ Q♦ wins over 2♣ 2♦). If that same category is “High card”, then the hand with the highest-ranked card wins.

# Suggestions for OOP design

There are many good approaches you can take to developing this app in OOP paradigm. Here are some suggestions for classes:

|  |  |
| --- | --- |
| Card | We wrote a graphic version of Card class, and you are welcome to use it and modify it. |
| Deck | A Deck object should start with all 52 cards, should be able to shuffle itself, and should be able to deal one card at random. Once a card is dealt, it should no longer be in the deck. |
| PokerHand | A PokerHand object is a collection of Card objects. It should know how to add a card. It should know its highest Poker category. If its highest category is “One pair”, it should know how to return the rank of that pair. If its highest category is “High card”, it should know how to return the highest rank among its cards. |
| Button | Our textbook has a Button class, and you are welcome to use it and modify it. |
| PSGame | A PSGame is a way of thinking of the game as an object in its own right. While I will accept versions of Project 2 that don’t design the game itself as an object, I encourage you to try designing the app this way.  As an object, the game knows the dealt cards (and which need to be “face-up” or “face-down”). It knows which betting round it is. It knows what to do when various control buttons are clicked. Et cetera. |

## App visualization

**Start of a new game:** **Betting Round 1**

The two cards in the player’s hole are face-up, the rest are face-down. In the CONTROLS box, the STAY and FOLD buttons are active, the DEAL and QUIT buttons are inactive. In the RESULTS box, only the statistics are shown (average score of the games played, and the number of games played).

Graphical user interface

Description automatically generated with medium confidence

**Betting Round 2:**

If the player “stays” on the 1st round of betting, reveal the first three common cards.

Graphical user interface, application

Description automatically generated

**Betting round 3:**

If the player “stays” on betting round 2, reveal the fourth card:

Graphical user interface, application

Description automatically generated

**Betting round 4:**

If the player “stays” on betting round 3, reveal the last card:

Graphical user interface, application, PowerPoint

Description automatically generated

**Completed game: player “folds”, or player “stays” through the four rounds:**

All cards get revealed. The highest category for each hand gets determined and displayed in the RESULTS box. The winner is indicated with a checkmark (chr(10003)) and red color, the loser with blue color. The player’s actions are presented, and the points given. The average points and the number of games get updated. In the CONTROLS box, the STAY and FOLD buttons are deactivated, the DEAL and QUIT buttons are activated. A new game starts if the player clicks the DEAL button, and the program ends if the player clicks on the QUIT button.

For example, suppose the player “Stays” on 1st and 2nd round, and “Folds” on 3rd round. Here’s the result:

Graphical user interface, application, PowerPoint

Description automatically generated