CS2011 Intermediate programming and problem solving I

Project (15% of your total marks)

Due: 25th November, 5pm (Monday, Week 12)

Note this is an individual project

For your project, you will be a GUI application the Iowa Gambling Task, containing some functionality. In the Iowa Gambling Task, participants are presented with four decks (Deck A, Deck B, Deck C, and Deck D). Behind each deck is a winning amount and a losing amount. The goal of the game is to win as much money as possible. Some Decks are "good", meaning the winning amount is greater than the losing amount overall. However, some decks are "bad", meaning the losing amount outweighs the winning amount. To see how the Iowa Gambling Task works, visit: https://www.psytoolkit.org/experiment-library/experiment_igt.html

This GUI application uses event-driven programming:

- When a user presses on the 'a' character, the winning and losing amount associated with Deck A applies to the participant's overall winnings;
- When a user presses on the 'b' character, the winning and losing amount associated with Deck B applies to the participant's overall winnings;
- When a user presses on the 'c' character, the winning and losing amount associated with Deck C applies to the participant's overall winnings; and
- When a user presses on the 'd' character, the winning and losing amount associated with Deck D applies to the participant's overall winnings;

Overall, there are 16 possible winning and losing amounts for each Deck (as supplied in Decks.csv). Once all the winning and losing amounts from any one Deck have been applied to the winnings, the game should end. The game should also end if the participant clicks on the 'q' key. See video located on the CS2011 canvas page (under modules – Assignments - Project)

INSTRUCTIONS

It is only required to submit this project on canvas (psychopy is not working on repl.it). Please follow these instructions carefully. You will need four files, which should be saved in one folder named in the format ID_CS2011_Project. For example, if your ID is 123456, then you should name this folder *123456_CS2011_Project* upon submission. **You will need to zip the folder before submitting.**

- *participant.py* – a modification of lab assignment 1
- *filemanager.py* – from lab assignment 2
- *main.py* – creating from scratch, and we will use the psychopy library to create the GUI components
- *Decks.csv* – copy this file from the canvas assignment page or repl.it.
- *Report* – a short, 1 page document that details what you did and why.

participant.py:

In this file, you will be dividing the Participant class you created in Lab assignment 1 into Two classes: `Participant` and `TrialParticipant`.

1. Class `Participant`:
    - This class contains two variables: the participant's first name and the participant's last name.
    - Create getters and setters for these values and declare properties for these.
    - This class also contains a string representation – i.e. when the object is printed to the screen, it displays reader friendly information about the object to the user. The read friendly information should include the participant's full name and number.
2. Class `TrialParticipant`: `TrialParticipant` inherits all of the public methods and variables of Participant. Additionally, it contains the following:
    - Upon instantiation, this class also contains a dictionary, which is used to record the amount of times the characters 'a', 'b', 'c' and 'd' have been pressed.
    - This class should also include the following methods you created for the Participant class in Lab Assignment 1: `getWinnings(self)`, `setWinnings(self, winnings_loses)`, getKeyPressInfo(self), recordKeyPress(self, char_pressed), getMaxKeyPress(self), getMinKeyPress(self)

filemanager.py:

- This file should include The FileManager and DeckFileManager classes you created in lab assignment 2. No additional methods are required for this class.

main.py:

This is the main python file from which you will be running code and creating the main GUI components. To create this file, you will need to do the following:

- Import your TrialParticipant and DeckFileManager class
- Import the appropriate PsychoPy modules:

```
from psychopy.visual import Window, TextStim
import psychopy.visual
from psychopy import event
from psychopy import core
```

- As this trial will only be for one participant, you only need to create one instance of TrialParticipant. However, since there are four Decks, you will need to create four instances of the Deck class.
- Create a window to display Text and GUI components. **Note: everything the participant needs to see should be displayed in this window (as shown in video). All other information can be printed to the console window:**
    - The characters with the max number of key presses
    - The characters with the minimum number of key presses
- **`runningTrial(key_pressed):`** your main code should have **at least one function** (called runningTrial), which sends in a list representing the initial key pressed by the user. You can create other functions if you wish. Note that this is not a recursive function.

a. A loop, which is used to keep the window and trial running. You should continuously check to see if a user has pressed a character inside the loop. This loop is only broken if:
    i. The user chooses to quit the trial (by clicking on the 'q' key); or
    ii. If there are no more winning and losing amounts to read in one or more of the Decks. For instance, if all 16 win and lose amounts have been read for Deck A, the trial should end, even if there are still winning and losing amounts in Deck B, Deck C and Deck D.
b. Inside the while loop, write code that checks to see if a user has pressed the 'a' character (i.e. indicating the user has chosen Deck A). If the user has pressed the 'a' character, the following actions apply:
    i. Obtain the winning and losing amount from Deck A
    ii. Obtain the amount won as display text
    iii. Obtain the amount lost as display text
    iv. Recalculate the participants winnings
    v. Pass the character 'a' into the `recordKeyPress()` function of your instance of `TrialParticipant` class, so that the character count is logged
c. Inside the while loop, write code that checks to see if a user has pressed the 'b' character. If the user has pressed the 'b' character, the steps in part (b) apply, but for Deck B in this case
d. Inside the while loop, write code that checks to see if a user has pressed the 'c' character. If the user has pressed the 'c' character, the steps in part (b) apply, but for Deck C in this case
e. Inside the while loop, write code that checks to see if a user has pressed the 'c' character. If the user has pressed the 'c' character, the steps in part (b) apply, but for Deck D in this case

Report

As a supplement to your code, you are also requested to write a report (1 page max) that describes how you implemented the code (e.g. how are you obtaining the winning and losing amount for each deck, and how is that being applied to the participants winnings?)

Your project will be marked according to the following, so you should demonstrate your understanding of these in your report:

- Classes and inheritance (5 marks)
- Problem solving as per instructions provided (5 marks)
- Creating GUI components and events using PsychoPy (5 marks)

Once you have completed the project, zip the folder containing all of your files and submit it on canvas at the following location: *Assignments -> CS2011 - Project*

Plagiarism:

The Python code will be checked for plagiarism following the UCC guidelines. Plagiarism detection in a computer program is a simple process and experienced programmers can easily see if two programs are "modified" versions of the same one.