F09 – Group 3

Members: Darren, Wei Guang, Baarath, Melodee, Jayati

Description:

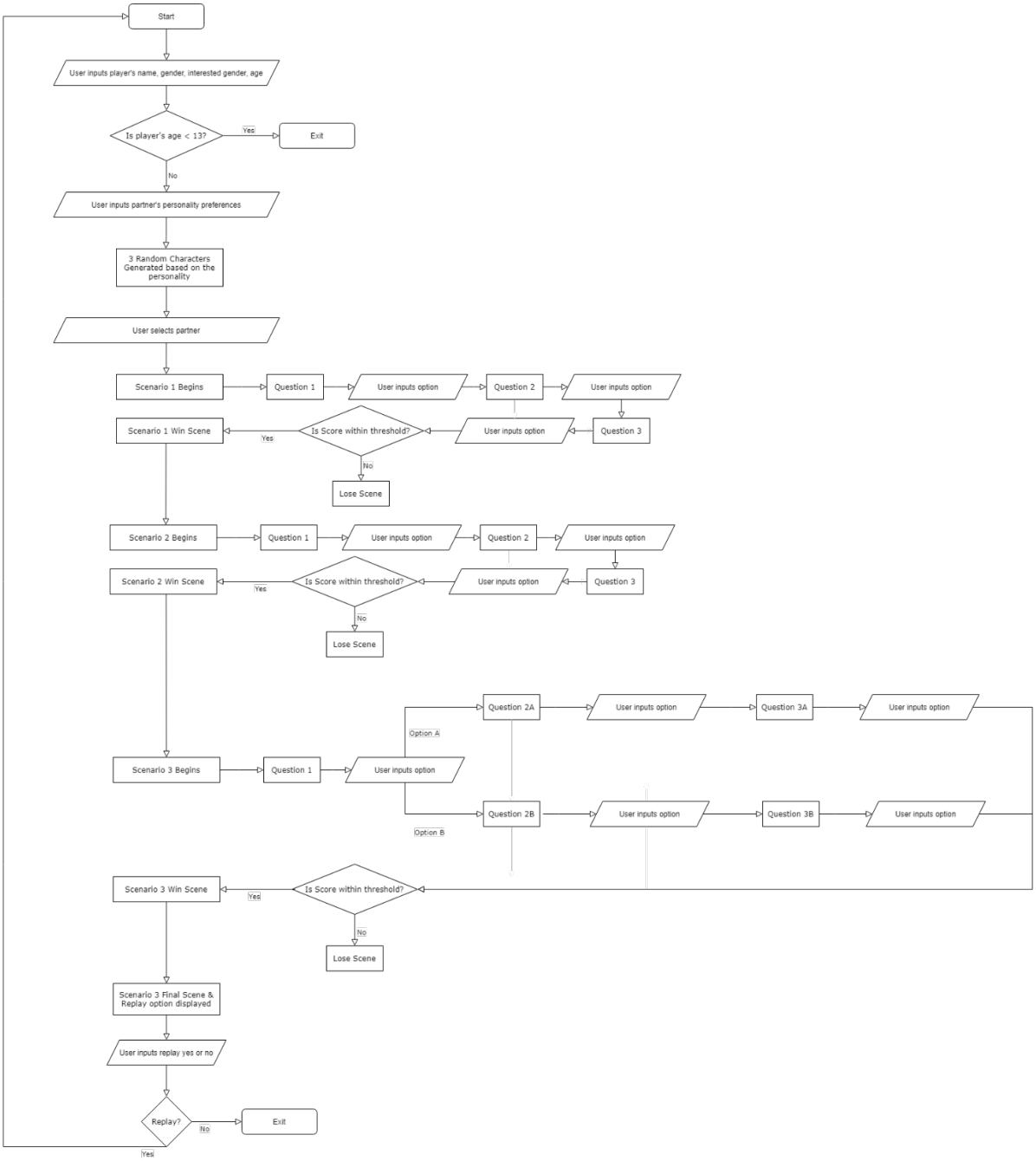
Our game is a simplified version of a popular interactive dating video game called Episode(https://www.episodeinteractive.com/), where the player must make his/her own choices throughout the game which will affect the eventual outcome.

The game first starts by asking some basic information about the user(e.g., gender, age) and then proceeds to ask the user for his/her personality preferences in choosing his/her partner.

Subsequently, 3 customised characters, with different characteristics(e.g., occupation, personality), will appear on the screen and the player makes his/her 1st choice to pick the character to engage in a text conservation with.

Thereafter, during the conservation with the chosen character, he/she will have to pick between 2 possible responses. Depending on the response that he/she picks, it will either increase their compatibility or make the relationship go downhill. The game ends when the player goes through all the scenarios and the partner decides to marry(?) him/her or when a series of incompatible decisions and he/she decides to end the relationship.

The flow of our game can also be summarised using the flow chart below:



Documentation:

Imported libraries:

```
import random
import sys
import matplotlib.pyplot as plt
```

data_config.py

This file contains a data_config class which contains many attributes, such as data_config.list_of_jobs, which is a list of occupations that the character may possibly have.

This .py file mainly acts as the database for our game, containing information such as personality types, ages, scenario questions.

main.py

```
def choose_options(qns_and_options):
```

This function makes the user choose the available options regarding his/her preferences and using an if statement, forces the player to retype the option if it is an invalid option.

```
def randomize_partners(current_player: player(), game_config: data_config() ,
num_of_partners: int() = 3 )
```

This function is mainly responsible for generating the 3 character profiles. From the lists in the data_config class, the function is able to generate 3 random profiles. A very significant feature in this function is the ability to generate “imperfect” profiles. Using the *random.choice()* and then the *remove()* function, we are able to generate a personality profile where it is impossible to have a character that fits the preferences of the user completely.

This thus forces the user to make a choice between what she/he thinks will be the most compatible partner moving forward.

```
def end_game():
```

This function simply asks the user if he/she would like to restart or quit the game upon its completion.

```
def string_validation(question)
def age_validation(question):
```

These 2 functions are meant to make sure that the user types in valid input when answering the questions. They contain the *try* and *except* keywords to aid error handling.

```
def display_partners(partners: []):
```

This function acts to print out the 3 character profiles that we have generated for the user. It makes use of f-string formatting to make sure that the strings are aligned.

```
def main():
```

This function contains all the above functions and executes them in the desired chronological order.