

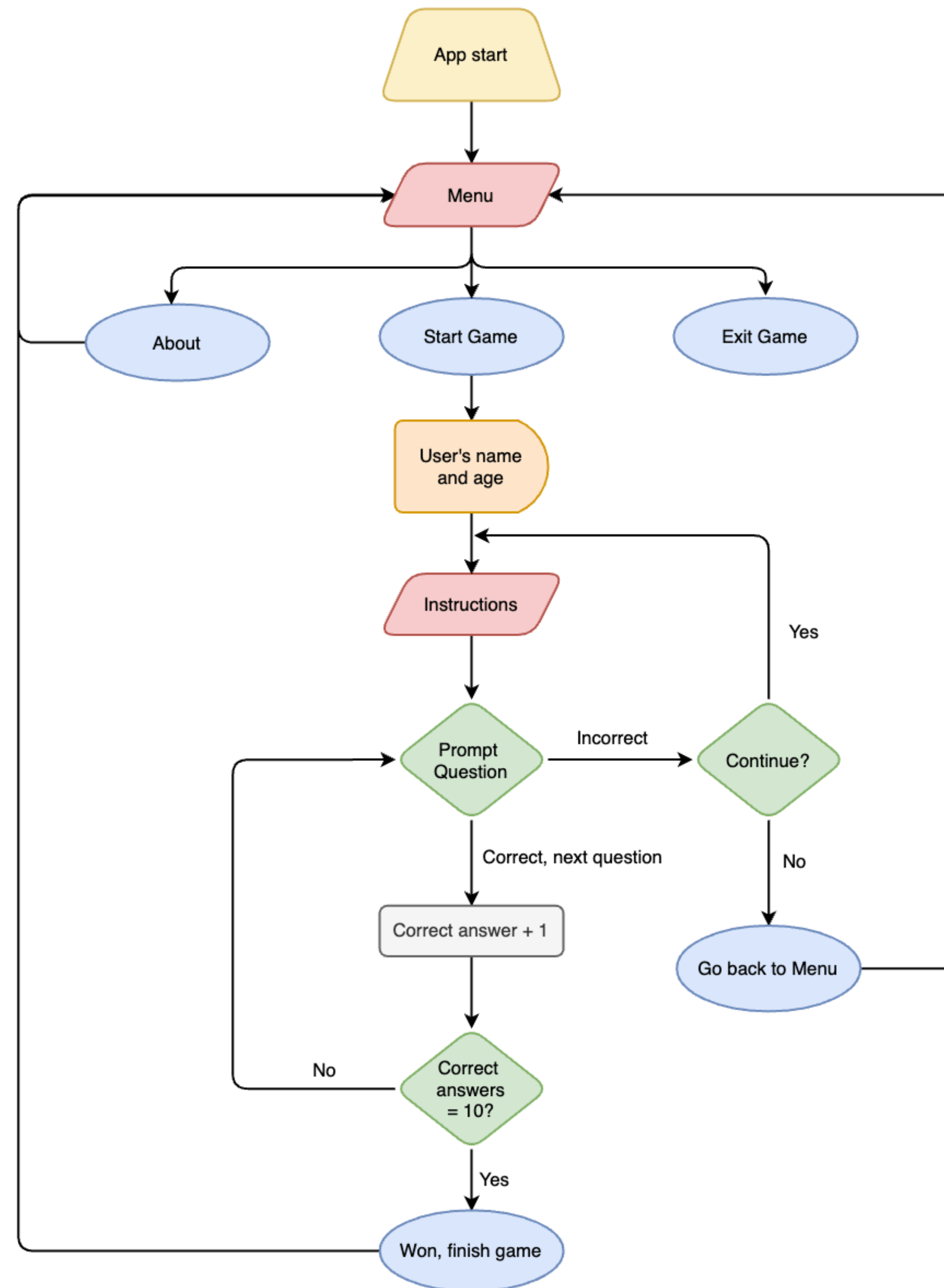
Are you smarter than a 5th grader?



Terminal app

Arturo Caceres

Control flow





Features

- User Input
- Different message depending on the age of the user
- Adding points per correct question

User Input

```
Please enter your name:
```

```
Please enter your age too:
```

```
Please enter your name: Mike
```

```
Please enter your age too: 42
```

User Input

The user will input 2 parameters, their name and age. This will be on the initialize method defined on the class Player.

Throughout the game, these 2 parameters(name, age) will be used to greet the user and depending on the age of the user, the user will get a message when they lose.

```
class Player
  def initialize(name, age)
    @name = name
    @age = age
  end
```


Message depending age group

Incorrect.

Wow, at your age you should know this. It's time to grab a book. Start again.

Game over

Message depending age group

When the user misses a question, depending on their age group, they will receive a personalized message. This is so that the user will feel a more personalized experience when playing the game.

The user should be preferably from 7 to 99 years old. Younger kids under 7 can participate and try, but usually at that age is when they start learning to write properly. In the other spectrum, someone older than 99 can also try, but it would be rare to see a 100-year-old person trying to a game in the terminal. In the class `Player`, we define the method `gest_loser_message` to call the messages that the user will see once they lose.

Adding points per correct question

```
Current points 10
```

```
In our solar system,
```

```
Neptune
```

```
Pluto
```

```
Saturn
```

```
Uranus
```

```
1
```

```
correct
```

```
Current points 20
```


Adding points per correct question

When the user starts the game, they will start with 0 points, and they will have to answer all answers correctly to win the game. There are 10 questions worth 10 points each. After every correct answer, the user will see how many points they have accumulated, knowing how close they are to victory.

For the requirement of accumulating the points, points are saved on a variable called `game_points`, which is within a loop that will run until the user goes through all questions or fails a question. When it breaks, you will encounter the option if you want to `Try again!` or `Exit` the game. As long as you select `Try again!`, you will stay in that loop until you win.

Challenges

- Making the code work first before adding design/cool gems
- Designing the Tests
- Documentation not clear for me, hence more research on examples.