PARKinetics

Group Name: TANKER
Group Number 6
https://ksackvil.github.io/CMPT-275/

Armaan Bandali
Rachel Djauhari
Negar Hariri
Yuxiang Huang
Takunda Mwinjilo
Kai Sackvile-Hii

Table of Contents

Table of Figures

Revision History

Table 1: Detailed Revision History of Document

Revision	Status	Publication/Revision Date	Author
1.0	First Draft	October 10, 2019	Rachel Djauhari
1.1	Added Guidelines and Data Requirements from Draft	October 12, 2019	Rachel Djauhari
1.2	Feature Priority	October 14, 2019	Rachel Djauhari
1.3	Modified Data Requirements	October 15, 2019	Yuxiang Huang
1.4	System Diagrams Edited	October 16, 2019	Kai Sackville-Hii
2.0	Linked Feature Priority to Features in Requirements Document	October 16, 2019	Rachel Djauhari

Guidelines

Development of the app will be done using Swift on Xcode IDE version 10.3 (10G8). There will be data collected from the users that will only have their progress recorded; no personal or unique identification will be required that would lead to unsafe handling of the user's information.

System Diagrams

In this section our systems process and classes will be explained using system diagrams.

Figure 1 is the process view of our app, which describes each process of our system and how they interact. When the app is launched, the system process in figure 1 begins at the "Main Screen" box. From here a user can either load a game or view their progress. When a game is loaded the users view is segued to the "Game Screen". Inside this screen only one out of the three possible games will be loaded. Each game inherits from the "Game Screen", in figure 1 this is visualized by the "Game Screen" box encapsulating each game box. From the "Game Screen" a user can either exit back to the "Main Screen", or successfully complete the game. After completing a game, the results are written to our database. The users view is then segued to the "Progress Screen" where the user can see their updated progress.

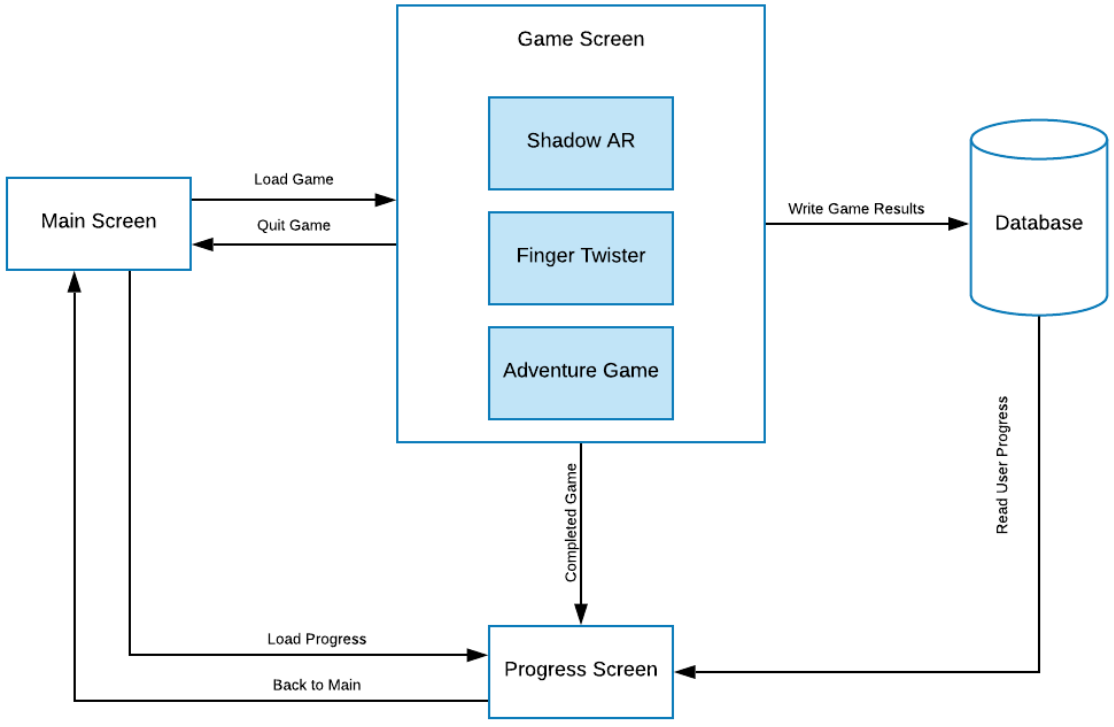

Figure 1: System Process Diagram

Figure 2 is the UML class diagram, which shows each object class and its associations. Figure 2 shows our systems process in a class specific lens. When the app is launched, the "ViewController" initialized. The "ViewController" class handles "Main Screen" view, as well as loading the "Game Screen" and "Progress Screen" seen in figure 1. The "Game Screen" is controlled by three classes: "GameViewController", "GameScene", and "GameOverScene". The "GameViewController" is called when the "Game Screen" is loaded and handles all game logistics. "GameScene" is called from the "GameViewController" to begin one of the child class' gameplay. When a game is over, the "GameOverScene" is called to handle the exit process. If a game was not completed, the "GameOverScene" returns to the "Main Screen" controlled by "ViewController". If a game was successfully completed the "GaneOverScene" updates our database and segues to the "Progress Screen" controlled by "ProgressVC".

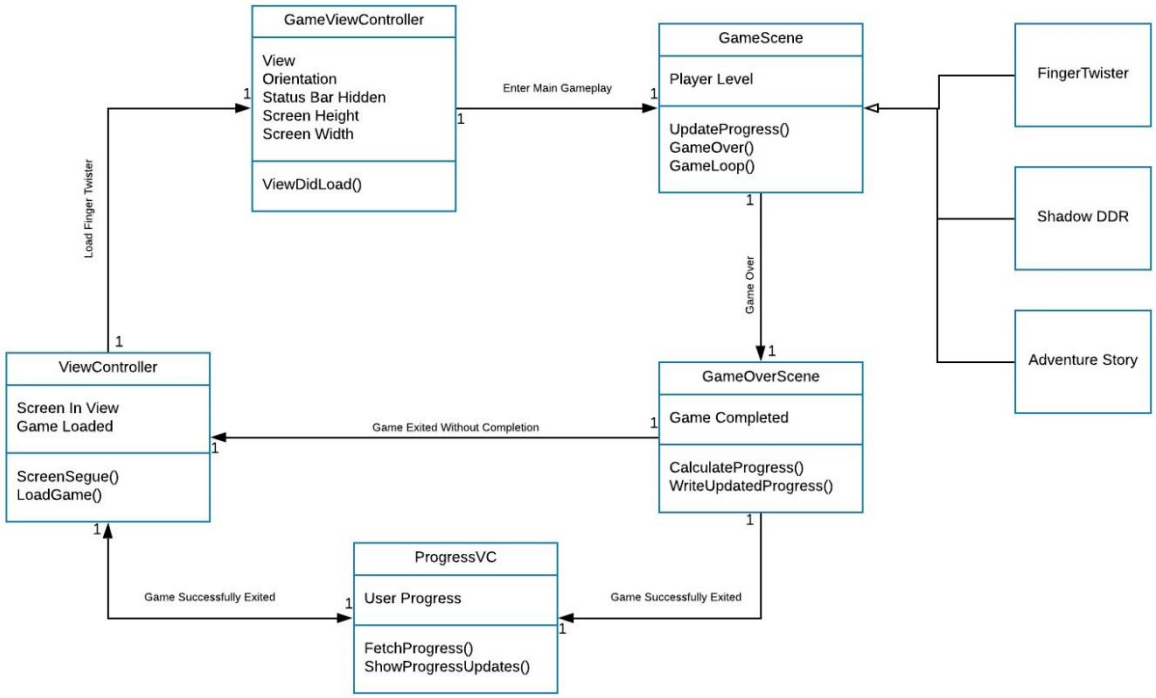

Figure 2: UML Class Diagram

Data Requirements

Overall, the inputs of our iOS app will cover text input, audio input and video input. Setting data is read from text input (users drag sliders and press radio buttons on touch screen). Profile information is also read from text input (users enter text by typing on a virtual, on-screen keyboard). The settings and profile data will be written to and read from iOS key-value local storage. Raw data from the provided games will be collected from the following inputs: touchscreen, microphone and front camera. All raw game data will be analyzed and 5 kinds of "PARKinetics" parameters will be calculated: balance, facial rigidity, speech, digit dexterity, and posture. New "PARKinetics" values will be uploaded to Firebase **[1]**, a remote real-time database (NoSQL schema), and a history of their gameplay will be read from the database at the end of each game. I/O between client and server is relied upon Firebase Realtime Database SDK.

```
{
    "$schema": "http://json-schema.org/draft-07/schema",
    "type": "array",
    "items": {
        "type": "object",
        "properties": {
            "uid": { "type": "string" },
            "username": { "type": "string" },
            "email": { "type": "string" },
            "games": {
                "type": "array",
                "items": {
                    "type": "object",
                    "properties": {
                        "gid": { "type": "string" },
                        "type": { "type": "string" },
                        "balance": { "type": "number", "minimum": 0, "maximum": 10 },
                        "dexterity": { "type": "number", "minimum": 0, "maximum": 10 },
                        "facial": { "type": "number", "minimum": 0, "maximum": 10 },
                        "posture": { "type": "number", "minimum": 0, "maximum": 10 },
                        "speech": { "type": "number", "minimum": 0, "maximum": 10 },
                        "time": { "type": "string" }
                    },
                    "required": ["gid", "type", "balance", "dexterity", "facial", "posture", "speech", "time"],
                    "additionalProperties": false
                }
            }
        },
        "required": ["uid", "username", "email", "games"],
        "additionalProperties": false
    }
}
```

Figure 3: Firebase Database Schema

Feature Priority

Version 1

1. Minimal UI
2. Homepage
3. Profile Page
    a. At least showing the radial chart
4. “Finger Twister”
    a. Complete working game with a few levels/difficulty modes to choose from

Version 2

1. Refine what was done in Version 1 as deemed necessary
    a. Complete Profile Page
2. Partially working “Shadow DDR” (AR game)
    a. The basic framework of the game
3. Partially working “Adventure Story” (speech recognition)
    a. Have a basic speech recognition going so that the game is playable
    b. Have a couple of scenes available for the storyline

Version 3

1. Refine what was done in Version 2 as deemed necessary
    a. Complete “Shadow DDR”
        i. AR is working properly
        ii. Have multiple routines set up from already known therapy for PD
    b. Complete “Adventure Story”
        i. Working speech recognition and accounts for errors in speech inputs
        ii. A couple of storylines available
2. Complete Settings Page

References

[1] Firebase Inc., "Firebase," Google, September 2011. [Online]. Available: https://firebase.google.com/docs/database. [Accessed September 2019].