Detailed Project Specification: Run! Quaver

Team 302
Team member: gbian xiaok

The original project proposal is “Vocal Run” from the accepted proposals of other groups’. This project is aimed at to offering users a voice-control parkour game. Players can use their voice to control the character to run or jump until reach the finish line. Then the score of the player will be recorded and presented on a public scoreboard. Instead of just implementing a single-player mode, the game provides two different control mode in all playing mode: 1) User can use keyboard to control their character; 2) User can also use voice to control their character. Also, instead of offering two-player mode, all public players play in the same global scene, where they could view other characters with different transparency levels in the background. And each user can also review their history matches with sound. All users whether they are login or not could view the global match stream with every user playing currently.

Technologies:
* Languages: Python (version: 2.7), JavaScript
* Frameworks: Django, JQuery, Ajax, Bootstrap
* APIs: TBD

List of Functionality

Sprint 1: Complete the web pages design and basic framework construction. Also, have a fully-working implementation for a single player with only keyboard control mode by Sprint 1, with the game state being stored in the database.

Functionalities:

* User register/login/logout:

1. Login: Users can use their username and password to login. And login page also provides a password reset link in case that users forget their passwords which allow users reset their passwords through email.
2. Registration: Unique username, password and email address are required while registration.

* Basic game view: The game view contains several features:

1. Character
2. Map or game scene
3. Scoreboard
4. Timer

   [See the draft for the game implementation in appendix (Fig 1)]

* Keyboard input stream: Direction keys: “↑” “↓” “←” “→” will perform as one of the action control streams.

* Keyboard control mode: Use keyboard input stream to control characters’ action, like walk, run, jump and stop.

Primarily responsible team member: Ge Bian (Andrew ID: gbian)

Sprint 2:
Functionalities: Have a fully-working implementation for a single player with voice control mode by Sprint 2, with the voice file being stored in the database.
* Voice input stream: Analyse volume and frequency of the voice stream
* Voice control mode: Use voice input to control character's action, similar to the keyboard control mode instead of using keyboard stream.
* Voice storage: Store voice track of each match with related game info to database
Primarily responsible team member: Xiao Kuang (Andrew ID: xiaok)

Sprint 3:
Functionalities: Implement the real-time interactions between players, and also complete the game data fetching and displaying from database.
* Public scoreboard: Global ranking of highest score of each user's (Ajax update). Each score is related not only the distance finished, but also the time using which depend on the game accomplish state. An algorithm is designed for the scoring.
* Game match view: Depend on the basic game view completed in Sprint 1, a concurrent final game view will be fully implemented which allow the current user view matches of other user playing at the same time in the background.
* Game viewer view: Both unlogin and login users are able to enter the viewer view to watch an ongoing match stream with sound.

[See the draft for the game implementation in appendix (Fig 2)]

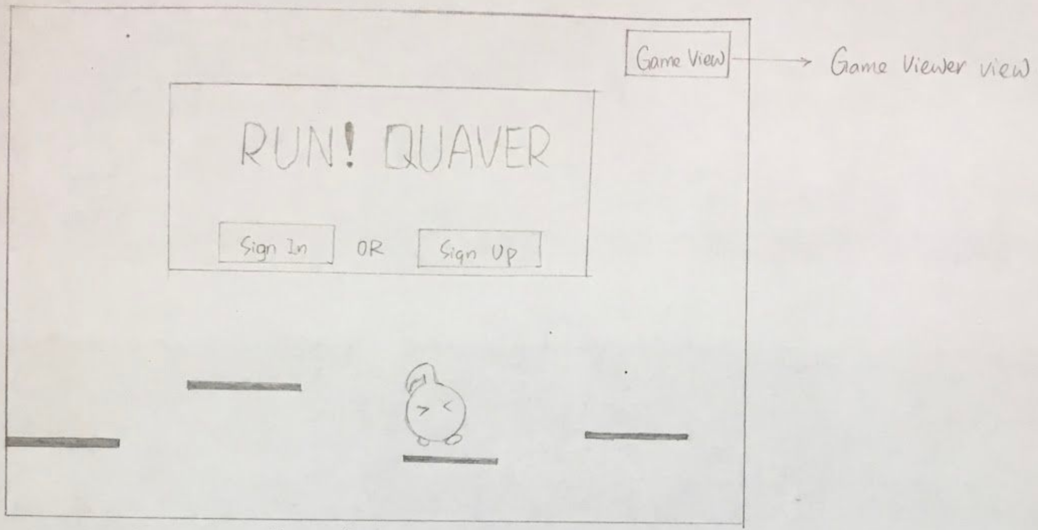
* Game playback: Logged in user can review their history matches
Primarily responsible team member: Both Team members

Implementation of the data models

```
CREATE TABLE User (
        Username VARCHAR(25) PRIMARY KEY,
        Password VARCHAR(25),
        HighestScore Double,
);
CREATE TABLE Matches (
        User VARCHAR(25) FOREIGN KEY REFERENCE User(Username),
        Match Int FOREIGN KEY REFERENCE Match(Id)
);
CREATE TABLE Match (
        MatchId Int PRIMARY KEY,
        MatchTime DataTime,
        TimeSpend Long,
        Score Double,
        Distance Double,
);
```

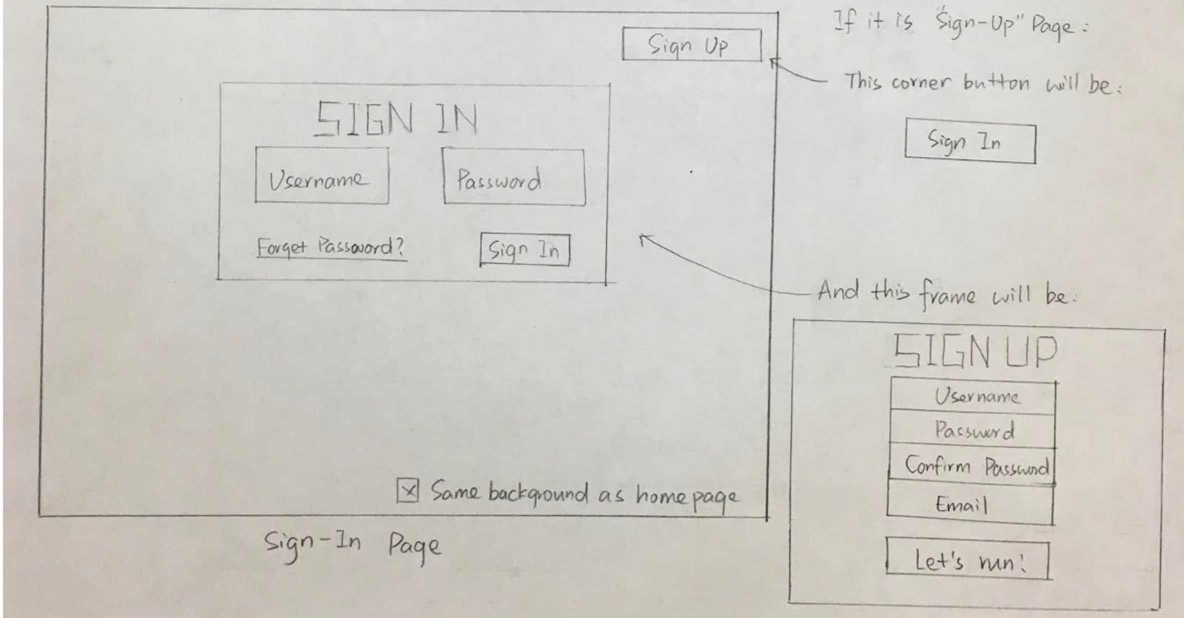
Wireframes for view

User Interface draft:



Game Viewer view

Home Page

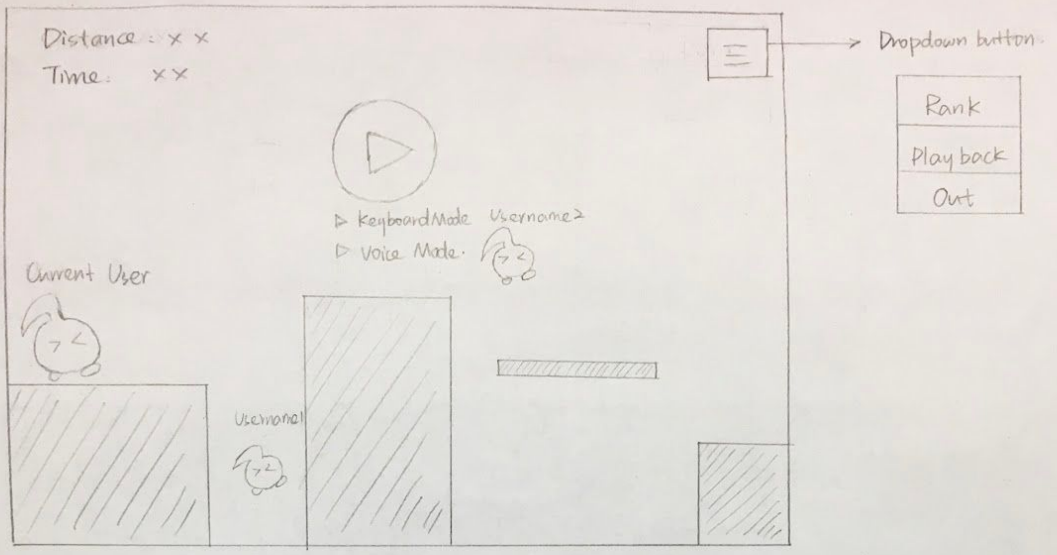


If it is "Sign-Up" Page:

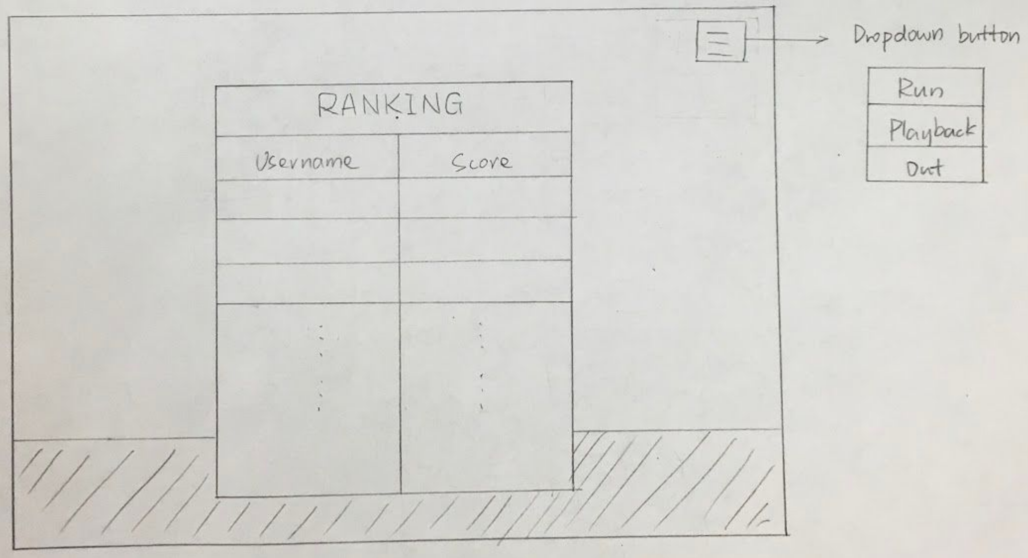
This corner button will be:

And this frame will be:

Sign-In Page



Game View



Public Scoreboard