Assignment 5

(due Tue, 5/2 at 11:59 pm)

CMSC 491 – Mobile Programming (iOS)
Spring 2017

In this assignment you will be building a multi-player quiz app (name your own app and use a custom app icon). **This assignment will be done in groups of two. Test your app, at least with two simulators and one actual device.** The app should run on every device type – portrait only.

Initial Screen:

There will be a choice for playing either single player or multi-player. If multi-player is selected and "Start Quiz" button is clicked without having connected to at least a peer, it should give error. On the other hand, as the maximum players is four for the quiz, if more than three peers connected, the game should not start. User needs to decrease the number of connections to start.

To connect other players, user needs to Click "Connect" and select whom to connect with through Multipeer Connectivity framework. You can use built in view controller to connect to nearby devices. Once the connections are over, when one of the peers clicks start quiz button, they all will switch to quiz screen and start quiz. Note that all players will need to switch to other screen at the same time. The player who clicks the "start quiz" button orders the switch by clicking the button but other players should be automatically switched once notified about this click.

Quiz Screen:

Depending on the number of peers connected (multi-player mode), there will be a display of peers connected (with different colors of person images), their scores, their short names (3-4 letters) – you can use first letters of peer IDs for this or p1, p2 etc. The first peer will always be the player of the current device. There can be at most 4 players (including the current device player). If the number of players is less than 4, the figures of remaining players should be accordingly designed with some placeholders (ex: gray color indicating inactivity)

The questions will be read from a url. Initial url is http://www.people.vcu.edu/~ebulut/jsonFiles/quiz1.json. An example json file is shown in the next section.

Once the quiz starts, there will be 20 seconds allowed for each question. The remaining time will be shown on the screen.

Each question has four options with one correct answer. At the first click, a player selects an option (which will not be submitted yet as final answer). He can change his selection by clicking other options. Once he re-clicks the selected option, the answer will be submitted and other players will also be notified.
The selected option should visually differ from the other options (background color can be different, feel free to use other ways). There can only be one selected option at a time. Once user submits, the submitted option can not be changed. It should also be different visually than the selected but not submitted option (another background color).

Users can also control their selections via Core Motion.

- By shaking the device, a random selection can be made (this also lets user do the first selection). Shaking should not cause submission, so consecutive random selection of same option should be prevented.
- By tilting the device to any of the four directions, the selected option can switch from the current one to other one in tilted direction. For example, if the current selection is C, then tilting the device upwards should change the selection to A. Or tilting the device to right side should change the selection to D (from C). Use some thresholds (ex: >1.0 or <-1.0) to get good amount of tilt to switch and prevent quick switches that may cause annoying user experience. Tilting should not cause submission.
- For submission of current selected option, player can use a good amount of acceleration in negative z direction (as you are moving the device away from your body). Second way for submission is to yaw good amount (>1.0 or <-1.0) in each direction. You need to implement both ways.

A B
C D

As the submissions made from users, the quiz screen with player figures should show the submissions in all players' devices. Selections will not be shown. Once all players submit their answers, no need to wait until 20 seconds end. Player scores will be updated and correct answer will be shown before switching to the next question. Here, do not use error alerts or anything that will require user clicking etc. The correct answer can be shown for 3 seconds, then automatic switching to next question can happen. You can show correct answer in your own way (by updating text in time label or by blinking the correct answer etc.)

At the end of quiz, the winner(s) and losers should be notified (ex: "You win", "You Lost", "You are one of the winners") and all players should be given an option to restart the quiz. Restarting the quiz will restart the quiz with same peers but it will read the next quiz file (quiz2.json, quiz3.json etc.). If it is not available, it should restart from quiz1 file.

Sample JSON File:


```
{
   "numberOfQuestions": 2,
   "questions": [
            {
            "number": 1,
            "questionSentence": "What is the capital of USA?",
            "options": {"A": "Rome", "B": "London", "C": "D.C.", "D": "New York"},
            "correctOption": "C"
            },
            {
            "number": 2,
            "questionSentence": "What was the age of Steve Jobs when he died?",
            "options": {"A": "59", "B": "56", "C": "49", "D": "54"},
            "correctOption": "B"
            }
            ],
   "topic": "General"
}
```

General Guidelines:

- Define your custom classes as needed.
- Do not make your images stretched. Use aspect fit or fill for the content mode of the image views.
- You can submit a README file to explain the differences in your design.
- Use your own image resources. Feel free to design your app UI as you wish. Custom designs with good user interfaces will get bonuses.

Submission Guidelines:

- **Make sure your app is running before submitting. Also, make sure your submission includes all files.**
- Submissions should be made in Blackboard before midnight on 5/2. Up to 3 submissions are allowed per group.
- Up to 2 days late submissions only (1 day 20% penalty, 2 days 40% penalty).
- Make sure all your images and other resources used are inside the project folder. When copying an image inside the Assets folder, always make sure "copy" is checked and select the target as the project.

General Grading Criteria:

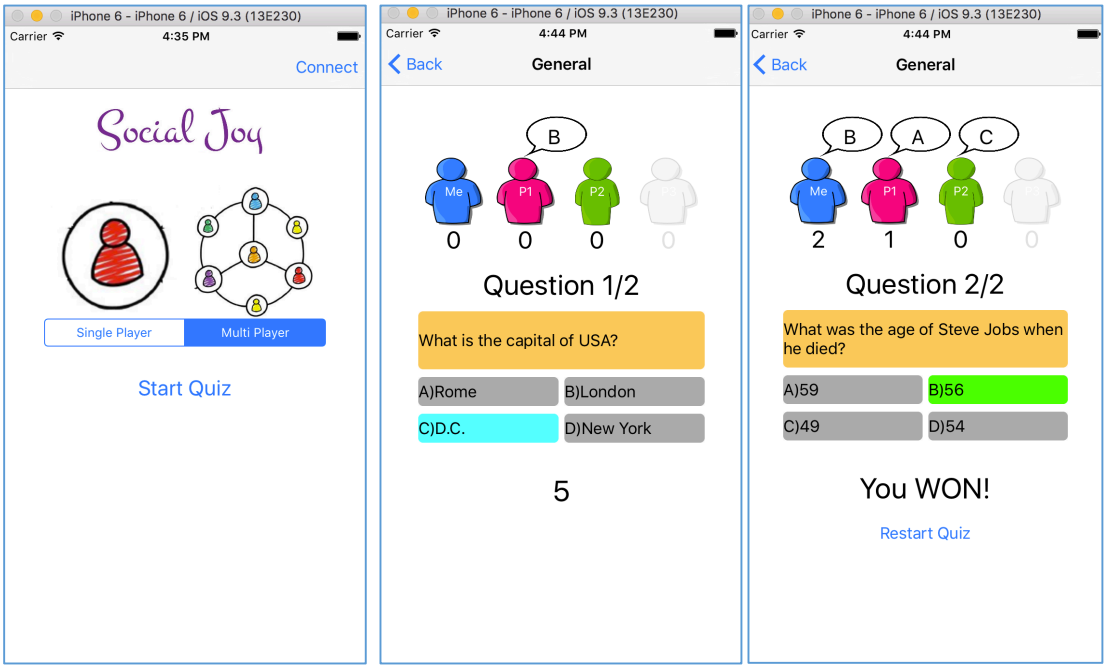
Appearance: (30%)

a. Initial Screen
b. Quiz Screen

Functionality: (70%)

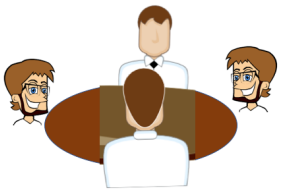
a. Single player gaming ability
b. Multi-player gaming ability (Will evaluate with 2, 3 and 4 players)
c. Reading of quiz files via JSON correctly and presenting them properly
d. User selections and submissions via clicks and taps
e. User selections and submissions via motion control
f. Showing user submissions in user panel in all screens
g. Showing scores in user panel in all screens
h. Game end – showing correct results and winner
i. Errors etc.

Some Screenshots:

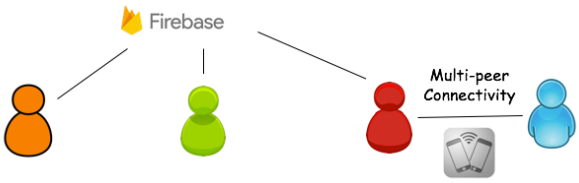


Bonus

- (10%) Design your UI as four people are sitting around a rectangular table (with the one whose back is seen is the current player and others are located based on their position with respect to the player. Other player on the left will be shown on the left etc. You can use magnetometer information for this.



- (15%) Allow a distant user to join the quiz (with necessary log in and Firebase database management). You need to connect only one user out of the users connected with multipeer connectivity framework to Firebase database. The app should work if some users only join with multipeer connectivity. Sample connection structure is shown on the right.



- (10%) **[This can only be done if the previous one is done]** Show players of the quiz on the map in another screen. They should be shown as custom annotations with profile images and their names under it. You can ask for profile photo and username if needed when they join. Show the users connected with multipeer connectivity over the same rectangular frame with the frame centered at their joint location.