CSC309H5 Winter 2016

Assignment 3: Remote Controlled Multi-Player Online Game

Work in groups of 2-3 students

Important dates:

- ➔ March 8th, 10:00pm: Proposal due (worth 10%)
- ➔ March 18th, 10:00pm: Checkpoint 1 due (worth 20%)
- ➔ March 29th, lecture time: Showcase (worth 20%)
- ➔ April 1st, 10:00pm: Final submission due (worth 50%)

You will work in groups of 2-3 students. Working individually is NOT recommended. Please consult with Larry if you really want to work individually.

Outside the allowed libraries, all code in this assignment must be your own. That is, you must write from scratch you HTML, CSS, Javascript, PHP, schema, etc. You must consult with Larry first if you are considering using any libraries other than the allowed ones.

In this assignment we are going to implement a multi-player online game that is remote-controlled by smartphones. The setup of the game is depicted as the following: multiple players will share the same computer screen where the main part of the game is displayed; each player uses a smartphone to visit a web page where the game controls are defined. The multiple players will control using their smartphone and interact with each other on the big computer screen.

An example of this type of game that we showed earlier in this course: https://lightsaber.withgoogle.com/ (but it's not satisfying our assignment requirement because it is not multi-player).

The implementation of this game will include the following components:

- ➔ The **game display web page** for the big screen. This part can be implemented using PHP, or Express.js if you like.
- ➔ The **game control web page** for the phone. This part can also be implemented using PHP, or Express.js if you like.
- ➔ A **Node.js server** that maintains the shared game states. The game display web page and the game control web page communicate with the Node.js server via **WebSocket**, i.e., each of the game display web page and the game control web page is a WebSocket client of the Node.js server.
- ➔ Other components such as databases, if it is needed by your game.

Basic requirements:

- ➔ The game must be a **multi-player** game, i.e., it need to be played by at least **two** players at the same time (more is better). The multiple player could play against each other, or cooperatively.
- ➔ You must use WebSocket as the communication mechanism.

In general, the product we want to see is a polished and fun-to-play game with a nice interface and responsive game control. The main criterion of the evaluation will be the amount of work that's put into the product, i.e., we expected to see at least 3 weeks' worth of work put into your product.

Features that will get higher marks:

- ➔ Game play of more than 2 players.
- ➔ Extensive and sophisticated use of HTML Canvas
- ➔ Game control using the motion and orientation API of the mobile phone
- ➔ Impressive visual appearance
- ➔ Responsive and smooth game play
- ➔ Other features that exhibit technical complexity and creativity.

Allowed libraries and tools (any other libraries/tools must be approved by the instructor):

- ➔ HTML/CSS/JavaScript
- ➔ Canvas and Mobile API
- ➔ WebGL (for 3D graphics)
- ➔ JQuery

- ➔ Node.js
- ➔ Express.js
- ➔ WebSocket
- ➔ PHP
- ➔ PostgreSQL

Proposal Requirement:

- ➔ Submit to MarkUs as “proposal.pdf”.
- ➔ A one-page (at most two pages if having pictures) description of the game that you want to implement. You may include mock-up picture of how your game will look like.
- ➔ You must provide a catchy name of your game.
- ➔ Evaluation: You get full mark for the proposal as long as you submit it on time.

Checkpoint 1 Requirement:

- ➔ Checkpoint 1 will be submitted to MarkUs as “a3ch1.zip”.
- ➔ You should finish most of your front-end implementation for the game display web page.
- ➔ You should have some temporary local controls on the game display web page, which is can be used to play the game as a local offline game. The appearance of the game should be very close to the final product.
- ➔ Backend features such as WebSocket are not required for Checkpoint 1.
- ➔ Evaluation: Marks will be given according to the following two criteria
    - ◆ The technical complexity of the frontend of the game
    - ◆ The completeness of the frontend implementation
    - ◆ The overall quality of the game that is exhibited by the frontend.

Evaluation of the final submission: In general, your product should show that you have put in a sufficient amount of work (**at least 3 weeks of work**) into it. Criteria include:

- ➔ The technical complexity of the game
- ➔ The completeness of the game
- ➔ The quality of the visual display
- ➔ The quality of the game control
- ➔ The quality of the code
- ➔ Additional creative features, wow factors...

More details of the assignment will be posted on Piazza. Please ask question whenever there is something you are not sure about.