**Project Proposal**

Folashade Okunubi, Nishita Muhnot, and Srinivasan Vijayaraghavan

**Application Description**

This application is a spinoff of a touch screen game Fruit Ninja. The goal of the game is to pop the balloons to get water to the respective reservoirs either positioned below the screen or to the side. For our current implementation instead of “reservoirs” we are trying to supply water to Neighborhoods. Each neighborhood is signified by a different color, so you will need to go back and forth between the neighborhoods to keep all their levels high. Switch between the neighborhoods by tapping them on the screen, and then only slash balloons that have the same color as the neighborhood. Popping the wrong color balloon will have negative consequences and affect water levels.

**Purpose**

This derivative of Fruit Ninja that is novel is the language learning aspect of the game. Throughout the game there will be special balloons (either white or multicolored) that when swiped will give a question to answer. If answered correctly, the neighborhood currently selected.

This application will assist the learning process by including a question and answer component as a supplement to the English classes they are attending. Content (questions and answers) will be provided by an API accessed by the server side of our application.

**Target Demographic**

This game will be open source for use by *TechBridgeWorld*. This game is targeted to be gender-neutral, but will be used mostly by male emigrant workers in Qatar attempting to learn English. With the current implementation, it should be easy to learn many languages by connecting it to different question databases. There are currently hopes to extend this application to help language learning in sub-Saharan Africa.

**Challenges**

There are several unique challenges to the development of this app. The target population is located in several other countries, so it will be hard to receive direct feedback during development. In addition to the time difference, our game faces the technical challenge of having to support two types of content servers. Another challenging aspect of the game is that the user experience should be targeted towards adults and developing the new languages, while still being engaging and interesting. Arguably the most important aspect and most obvious problem, the application needs to be as bug-free as possible as real people who want to learn will use it and the code will be shared for others to use.

**Goals**

Primary goals

The primary objective is to create a game akin to Fruit Ninja with the following characteristics:

* Active, engaging gameplay aspect apart from the questions
* Need to answer questions correctly to go far enough into the game.
* High replay value, which we’ll achieve through a never-ending game which becomes increasingly difficult
* Easy to pick up and play anytime
* Gestures for control are intuitive, whether or not the user has used touchscreens before
* Graphics are realistic and appropriate for a game meant for adults.
* It is also required for this game to work with the existing TechBridgeWorld content server, which it already does.

TechBridgeWorld is currently working on a new central content server with richer features and more modern technologies. Our secondary goal would be to have our project (client + game server) work with the new content server for forward compatibility. This goal is contingent upon their new server being finished, or at least finalized, in time.

Optional Development

We hope to also implement (time permitting):

* A high-scores table using mongoDB, which is updated when someone finishes playing
* A login system

**Technical implementation**

Since we are targeting multiple screen sizes, we will be using percentages and ems instead of pixels as the unit of measurement in the DOM.

The 1-2 menu screens before the game will use CSS transitions and/or jQuery to handle transitions.

-------------

Our game canvas will be fixed-size. We will be developing it for the iPhone 4 retina display size (960x640). The canvas will be scaled maintaining aspect ratio to fit the user’s screen. For phones which have other display sizes, they will see the website background on either side of the canvas.

Since our target platform is mobile, we will be using requestAnimationFrame() with fallback, as documented at http://paulirish.com/2011/requestanimationframe-for-smart-animating/

------------

We will be using PhoneGap to lock the screen in landscape mode during the game.

Since playing multiple audio files at the same time on HTML5 is difficult, we will not be including any background music; we will only have sound effects.

**Game Details**

To begin gameplay, the user will have to select which instructor they want to answer questions for. This was decided as opposed to game level difficulty selection, since this will be used first and foremost as a classroom learning supplement.  Instead, the gameplay simply becomes harder as you go on. The three ways gameplay difficulty will be increased are:

* increasing the speed of the water balloons
* rendering the balloons closer together
* increasing the number of undesirable balloon colors (more balloons to avoid)

The game will have a panel on the left to select which neighborhood the water is supplied to, and a large game area where water balloons can be popped. The balloons will be “thrown” onto the gameplay area to move in arclike trajectories similar to the Fruit Ninja gameplay. To “pop” a balloon, the user will have to slash it and then water will fall out of the balloon.

There are also special larger balloons with question marks on them. When they are slashed, a question is asked to the player (the game is paused behind it). The player gets 30 seconds to answer the question. If answered correctly, water levels of all neighborhoods will be increased.

For this current implementation, our game will have infinite gameplay. The scoring for the game will be as follows:

* 10 points for popping single balloons
* If multiple balloons are popped on one swipe, the user will be awarded 10\*n^2 amount of points.
  + ex: 2 balloons = 40 points, 3 balloons: 90 points
* Question balloons will be 50 points

**List of technologies**

Server

Our server is RESTful and runs on NodeJS. It currently also uses the ‘request’, ‘express’, and ‘bodyparser’ modules for NodeJS. Most of the server has been implemented already.

If we implement highscores, we will be using mongoDB

Client

Our client is still under development. However, we have decided to use Canvas for our game. We will also be using the following:

* jQuery: for AJAX requests, as well as for DOM manipulation for the menus
* CSS: transitions for the menus + general style
* Javascript: We will be using prototype and inheritance, and other advanced techniques as necessary/required.

General

* We are using Github.  
  Our repo is currently located at <https://github.com/Folashade/social-good>
* We have not decided on hosting yet, but we will be using either heroku or dotCloud.
* Since our game is not real-time, we are **not** using socket.io or any other Websockets implementation.

**File Structure**

As is typical for node applications, our client-side files stored inside the static folder, while the server-side is stored at the top-level directory

- readme.md – for github

- app.js

- docs/ -- for documents such as this one

- node\_modules/ -- has the installed modules

- static/css/ -- CSS files

- static/js/ -- client-side javascript for pages

- static/game/ -- javascript for the game

**Proposed Timeline**

4/5 through 4/11: Did competitive analysis, initial proposal, and finished functioning server.

4/12: Finish final proposal including timeline, file structure, and technical details.

4/13 through 4/17: Work on the front-end exclusively.

4/17 through 4/21 (carnival): Catch up + implement changes from class mini-user-study-a-thon

4/22 through 4/26: General work + Polish UI + Implement changes from class user-study-a-thon

4/27 and 4/28: Last weekend before project deadline.

4/30: Deadline