

Worcester Polytechnic Institute

立命館大学（Ritsumeikan University）

MQP Project Center

Project Concept and Design

**Students**

Maxwell Perlman

Stefan Alexander

**Advisors**

Dr. Robert Lindeman

Dr. Haruo Noma

**Abstract**

Through the use of GPS and iBeacon technology, players will engage in a real world role-playing game experience. Through engaging and constant gameplay, players are encouraged to go out into the real world, exercise, and explore, searching for digital monsters to fight and treasure to discover. The technical aspects of this project, design decisions, and projected timeline of this project will be detailed below.

**Table of Contents**

1. Introduction
   1. Project Goal
   2. Experiential Goal
2. Technical Aspects
   1. Account Association
   2. iPhone and M7 Chip
   3. iBeacon Devices
   4. GPS
   5. Server and Database
   6. Languages and Development Environments
      1. Swift and Xcode 6.4
      2. iOS Frameworks
      3. Ubuntu Server 14.04.1 LTS with Lubuntu Desktop
      4. MySQL and phpMyAdmin
      5. PHP
      6. JSON
      7. GitHub
3. Inspirations and Related Works
   1. Ingress
   2. Dungeons and Dragons
   3. Find Mii
4. Gameplay Guide
   1. Gameplay Systems
      1. Players
      2. Enemies
      3. Treasure
      4. Leveling and Traits
      5. Abilities
      6. Combat
         1. Single Player
         2. Multiplayer
   2. First Gameplay Experience
   3. Recurring Gameplay Experiences
5. Timeline
6. References

**Introduction**

With the help from the Ritsumeikan University faculty and students, as well as the pieces of hardware that they provide to us, we will be developing an iOS application with the latest pieces of technology.

**Project Goal**

The main goal of our project is to develop a mobile application that encourages a healthy lifestyle through a heavily gamified experience. We plan to meet this goal using the latest pieces of technology such as Apple’s new programming language Swift, their Core Motion API, and some other iOS frameworks in order to keep track of the user’s fitness while letting them have a great time. Although there is a lot more documentation on how to program certain things in Objective-C, we decided to push ourselves into trying out the brand new programming language with the beta version of Xcode6.

**Experiential Goal**

Of course we would like to end up with a well functioning iOS application, but at the same time our goal is to make an application that is enjoyable by the targeted audience—people who do not necessarily enjoy walking. Using a traditional role-playing game style, our goal is to create an actual real life version of an RPG where the players are encouraged to walk around to different locations with their friends in order to get various treasures and fight various monsters on the way. Once we have successfully developed a very enjoyable game, our other goal of encouraging a healthy lifestyle should be achieved as well since walking and socializing are the main features of our application.

**Technical Aspects**

**Account Association**

In order for players to play the game, they must associate one of their iPhones accounts with our game. iPhone devices automatically record accounts for popular social media applications (Facebook, Twitter, Sina Weibo, and Tencent Weibo) and can be accessed via the Accounts framework. Using this, there will be no need for the user to type in their username and password since it is already stored in the iPhone. This information is used by our server and serves as the players account id. Additional information is available in the Client-Server Model and Database section below.

**iPhone and M7 Chip**

Through the use of the newest iPhone’s M7 coprocessor, which allows for the collection and processing of data while the phone is locked, players will be able to constantly play the game even while engaged in another task. These data are accessible through the Core Motion API available from iOS 7, which allows the iPhone application to determine whether the player is stationary, walking, running, or in a vehicle as well as the number of steps they have been taking in a certain period. The background calculations are integral to our gameplay, especially since they save some battery life compared to previous methods of data collection.

**iBeacon Devices**

iBeacon devices are USB devices that can be powered by battery or plugged into a wall or machine. These devices emit a bluetooth signal to all nearby active bluetooth devices within a range of 10 meters. Through these iBeacons, we are capable of setting up hotspot location where players can meet together in order to engage in cooperative gameplay. This will be done by setting up the GPS locations of the iBeacons into a database to show their relative locations on the iOS map for the user, and also from the Bluetooth signals emitted by the iBeacons. The iPhone can detect whether it is “Immediate,” “Near,” or “Far” from the iBeacons so the player should be able to determine the exact location of the iBeacon from these signals and GPS coordinates. Additional information is available below in the combat subsection of the gameplay guide.

**GPS**

Through the use of the iPhones built in GPS tracking, we will be able to understand not only where our players are, but how far they traveled, how long it took them, and which other players are nearby. The Core Location framework also allows the application to determine the current altitude of the iPhone, which could be useful to determine how high the user walked. This allows us to tailor the player’s experience to his or her own walking habits. Additional information is available below in the leveling and traits subsection of the gameplay guide.

**Client-Server Model and Database**

Responsible for tracking all players locations and their information, as well as all sessions of iBeacon based enemies, treasure, and multiplayer combat, the database and server are essential to our games functionality. We will be using PHP, MySQL, and possibly other technologies such as Django and Python to handle the multiplayer combat system.

**Languages and Development Environments**

We are using the following languages and development environments for the production of our game. As needed, we will add additional technologies to this list.

**Swift** **and Xcode 6 (as of now, beta-4)**

Our entire iPhone application is being coded in Apple's new language, Swift. While still a new language, it has all of the potential that Objective-C could have offered us. As it is designed to run side-by-side with Objective-C, if we ever hit a point where the code does not work in Swift for some reason, we can switch to Objective-C and have a mixture of both. Swift is also said to run faster than Objective-C so that is a plus for iPhone applications especially for games.

**iOS Frameworks**

Some frameworks we’ve been using include Core Motion, Core Location, Accounts, Sprite Kit, Map Kit, and UI Kit.

**Ubuntu Server 14.04.1 LTS with Lubuntu Desktop**

For our server, we’ve installed Ubuntu Server to one of the lab computers. Also to make maintenance easier on us, we installed Lubuntu Desktop rather than doing everything through a CUI. Lubuntu Desktop is a lightweight operating system based off of Ubuntu Desktop, which uses the minimal Desktop LXDE. Since we did not need anything fancy, we just went with a lightweight Linux distribution.

**MySQL and phpMyAdmin**

The database is using MySQL in conjunction with phpMyAdmin to keep out database organized and accessible.

**PHP**

Our website, which is used to output our MySQL tables, as well as to show our databases contents is being coded in PHP.

**JSON**

In order to send database information to the iPhone efficiently, we are saving it as a JSON file, something easily parsed from the iPhone so that the phone could simply retrieve the single file.

**GitHub**

For source control between each other we are using GitHub to push and pull each other’s programs. This way, we will always have backup as well as a copy of each other’s code on the web.

**Inspirations and Related Works**

**Ingress:** <https://www.ingress.com/>

Ingress is a location-based game where players, divided into two factions, fight to gain control of real world spaces. By setting up three portals, a player can capture a plane of space for their faction. Players try to increase the power of their own portals and defenses near those portals, while at the same time waging war on the opposing faction's portals and defenses.

**The Good:**

* Ingress has an excellent mapping of game world object to real world locations.

**The Bad:**

* Ingress encourages bad behavior (such as using ones phone wile driving) while at the same time failing to encourage a positive thing that gameplay generally involves, health.
* To achieve a high level of gameplay that is rewarding is challenging in that teamwork is required for interesting play, but little to no encouragement of this behavior is provided by the game itself.
* Ingress does not encourage its players to travel to new locations and obtain new points; rather, player find locations along their daily commute or routine and continue to capture those points day after day until the game no longer captures their interest and they give the game up.
* Ingress provides insignificant positive feedback for completing in-game objectives. The game takes place on a global scale and the impact of a single person is very difficult to determine during a normal gameplay session.

****

**Dungeons and Dragons:** <http://www.wizards.com/dnd/>

Dungeons and Dragons is a tabletop game typically played with groups ranging between 4 and 8 players. Played mostly in the players imagination, players take on a new identity; consisting of a class (job or set of abilities such as a wizard, warrior, archer, etc.), a race, a set of abilities and values for traits (such as strength, health, speed etc.). Guided by a dungeon master (a real life person who is not actually playing the game but rather telling the and guiding the story for the players), players band together to travel through dungeons, fight of monsters, discover treasure, and engage in conversation with non-player characters.

**The Good:**

* Dungeons and Dragons, to those who find it interesting, is extremely engaging. Players spend weeks upon weeks working to complete just a single quest.
* Dungeons and Dragons encourages cooperative play to a very high degree. Most quests are virtually incomplete able without the support of the other players at the table.
* Dungeons and Dragons provides positive feedback, which yields further gameplay through evolving characters and a dynamic difficulty (set forth by the dungeon master).

**The Bad:**

* Typically, during sessions of dungeons and dragons, players are sedentary, sitting down for long periods of time and traditionally eating a large amount of unhealthy foods.
* Complexity. Dungeons and Dragons may be one of the most complicated games in the world, due to its desire to emulate the real world exactly as it is, while pinning an entire fantasy landscape onto it.
* Though it provides positive feedback through characters evolving, a large amount of the time the growth of characters is so slow that the positive feedback is too infrequent.



**Find Mii:** http://nintendo.wikia.com/wiki/Find\_Mii\_(3DS)

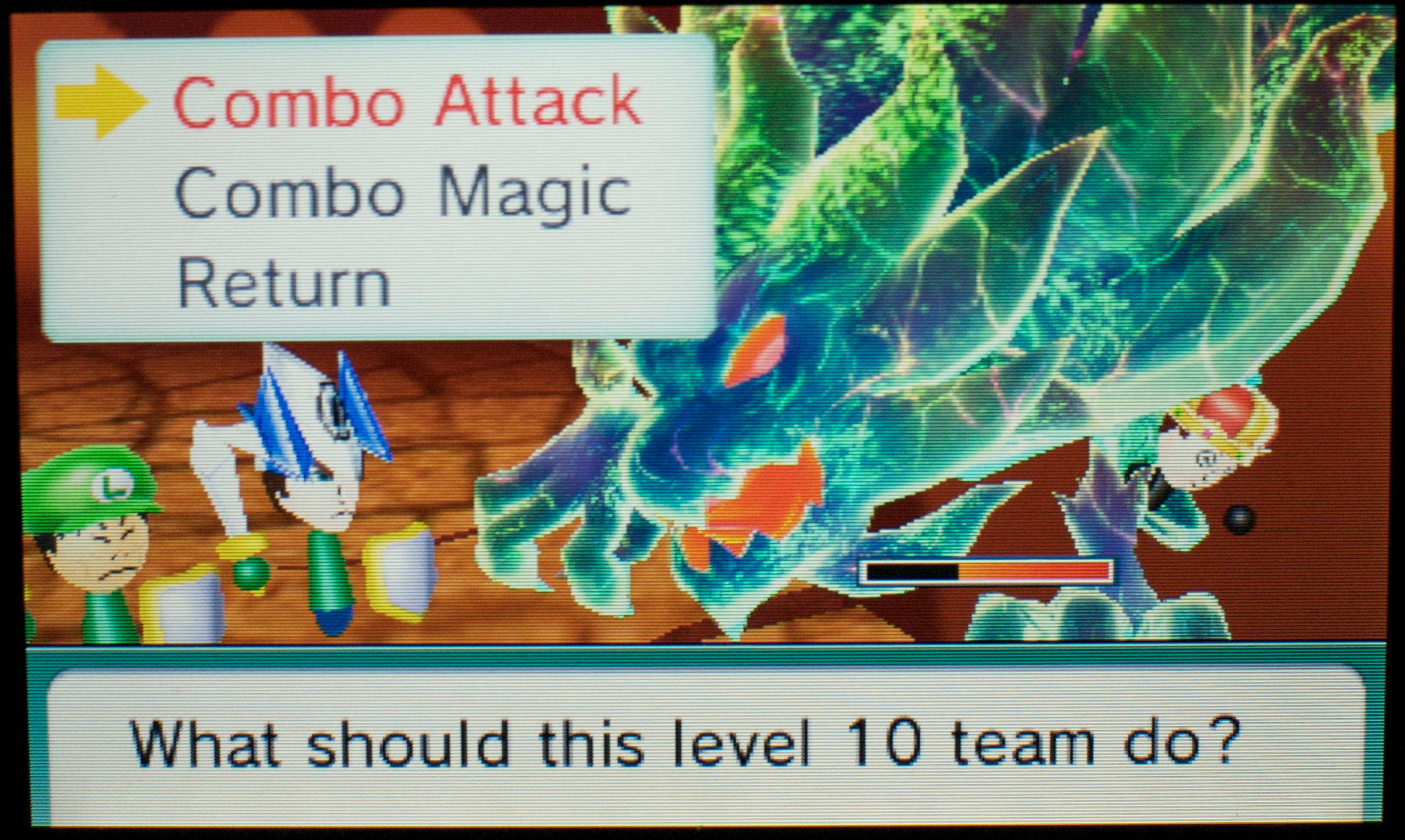
Find Mii is a game built into the Nintendo 3DS handheld system. By leaving their systems networking switch on and their system in sleep mode while going about their daily life, players are able to collect other players and use them to fight off monsters. The more frequently a player finds another player, the higher the player’s levels become in the other player’s game. By collecting people, players use their army of friends to fight off monsters and collect treasure that their character can wear, to be seen by other players.

**The Good:**

* Find Mii allows for constant gameplay. The player only need carry their 3DS system around with them, and the game does the rest.
* Find Mii encourages players go out into the world. The longer a player is out and about, the more people they have the possibility of collecting, and the more enjoyable gameplay becomes.

**The Bad:**

* Find Mii does not actually encourage game playing in the real world to a high enough degree. Typically, a player will carry the game around during the day, for example a college student walking to class, and will then play the game the next time they are at home.



**Gameplay Guide**

**Gameplay Systems:**

The following are all of the systems and mechanics involved in our game. The design of these sections is constantly evolving; this section is subject to drastic change.

**Players:**

Each player is broken down into a set of traits and a set of abilities they are capable of using. The use of classes (such as mage, warrior, etc.) was explored, but proved to be overly complex and unnecessary for our experiential goals. Both the equipment and ranking system were removed due to resource constraints (our development team has no artist). The list of traits and their impact on gameplay is explained below in the Leveling and Traits section.

**Enemies**

Each enemy (monster that the players encounter) is also described by a set of traits (explained in the Leveling and Traits section below). Enemies are broken down into two types: generic enemies that are found while walking around alone and special enemies found at iBeacon locations. The enemies at the iBeacon locations require multiple players to be present near the iBeacon for the encounter to begin.

The basic enemies will follow a basic finite state machine in order to determine their strategy for fighting. Each subtype of enemies will have its own state machine which determines it strategy; for example alternating between actions that lower the opponents strength and then attacking them). The larger enemies found at iBeacon locations will have slightly more complex state machines than the basic enemies.

**Treasure:**

As a way to convince players to walk more, they are able find treasures (which provide benefits to their character). The recommended number of steps a person should take each day is approximately 10,000. In light of this, each player is capable of finding one piece of treasure per day after they have taken 10,000 steps.

These treasures grant your character a benefit in some way based on your daily walking. For example Treasures can also be obtained by defeating special enemies found at iBeacon locations.

* If you burned a large amount of calories (based on the number of steps taken and temperature outside), your characters health trait would get a boost from the treasure found.
* If you walked up and down a mountain on a given day, your character would find a treasure that would benefit their strength trait.
* (Idea still developing)-If your character spent a lot of time walking in the vicinity of others; your character would find a treasure to boost their magic trait.
* If you had a particularly high walking speed for a given day, your character would find a treasure that would boost your characters speed trait.

**Leveling and Traits:**

As players play the game, their character will grow more powerful and the player is able to dictate exactly how that growth takes place. Players may assign up to 5 points per level to the health, strength, magic, and speed traits. These traits determine the abilities of each player and enemy as well as the power of those abilities.

Players and Enemies each have the following elements:

* Name: This is used to identify players and is the same as their login identifier.
* Level
* Health: The players walking constantly restores this trait’s value.
* Strength, Magic, and Speed

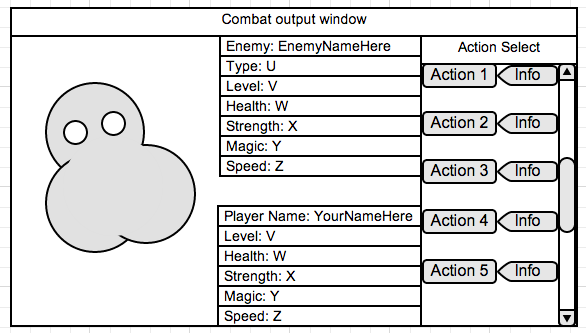
**Abilities:**

Abilities are the actions that players and enemies can take during an encounter. All abilities are available to all players, however a player’s ability to use abilities is dependent upon their trait’s values. For example: the punch ability is available to all characters as it requires only 1 strength to use, while an ability like Fireball is only usable to by a character who is level 5 or higher and has a magic ability of at least 45. These rules also apply to the randomly generated enemies and enemies found at beacon locations.

**Combat:**

The combat system for fighting enemies is turn based. Below, is a mock up of what we believe our combat system will look like.

* On the top of the screen will be an output window, showing the result of the most recent ability used. The player may click it in order to see past results.
* On the left side is the image of the enemy being fought.
* In the center of the screen, are windows showing the enemy’s trait values (if they are know) as well as your own trait values.
* On the right side is a scrolling list of all abilities your character is capable of using. Next to each ability, is an info button that, if pushed, will provide a description of the move were it to be used.

****

**Single Player:**

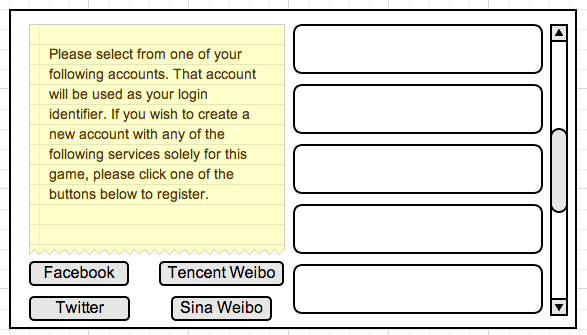
After each 1000 steps that the player takes, they will encounter a randomly generated enemy (the random generation is based on their characters level). The player will then be transferred into a view that looks like the above image. Here, they will engage in turn based combat against the randomly generated enemy. If they win the battle, they will receive some experience that will eventually help them to level up their character. If the lose the fight, they cannot engage in another fight until their health has been restored (by walking).

**Multiplayer:**

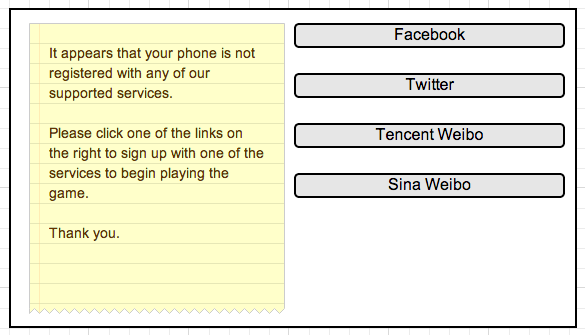
Upon finding a beacon, while accompanied by other players, players will engage the beacon’s enemy in combat. Combat here is very similar to the single player combat; however there will be points during gameplay where they must wait for their partners to input their attacks and receive the attacks done by the enemy from the server. Once the enemy has been defeated, all players involved in the fight will receive a treasure (explained above in the Treasure section).

**First Gameplay Experience:**

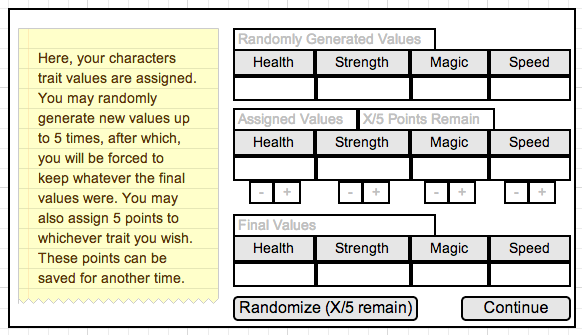
Content

****

Text goes here

****

Text goes here

****

**Recurring Gameplay Experiences:**

Content

**Timeline**

|  |  |
| --- | --- |
| **Milestones** | **Date** |
| Set up server with LAMP stack | July 7-11 |
| Make HTTP requests, server returns JSON, manage accounts, iBeacon, basic combat system (text) working | July 14-18 |
| Work on combat with SpriteKit framework, iOS map | July 21-25 |
| Core Motion API working for pedometer and etc. | July 28 |
| More SpriteKit, as well as MultiPeerConnectivity API for networked play. | July 29-August 1 |
| Finish Enemy AI, any other things we missed | August 4-22 |
| Alpha Testing | August 25 – 29 |
| Build beta based on feedback | September 1-12 |
| Beta testing | September 15-19 |
| Polish | September 22-26 |

As we do not know yet how long certain parts of our application will take to implement, we have planned it so that we finish earlier than presentation day.

**References**

**Images:**

<http://www.bandwagmag.com/wp-content/uploads/2014/02/ingress-phone.jpg>

<http://digitalfootprintblog.files.wordpress.com/2013/04/dsc_0735.jpg>

<http://www.wizards.com/dnd/files/dungeon/155/155_Sleeper.pdf>

[https://moqups.com](https://moqups.com/)

**Info:**

<http://www.livescience.com/43956-walking-10000-steps-healthy.html>