**Project Name: *Altair***

**Repo:** <https://github.com/BUUPE/Altair>

**Members:**

* Dominic Magione
  + Computer Science, 2024
* Carlos Lopez
  + Computer Science & Physics, 2023
* Nina Athma
  + Computer Science, 2024
* Matt Henriksen
  + Computer Science, 2024

**Language To Be Written In (Tech Stack):**

* React Native
  + Open-source mobile application framework that enables web developers to create mobile applications using their existing JavaScript knowledge. Offers faster mobile development, and more efficient code sharing across mobile operating systems and the web.
* Node.js
  + Primarily used for event-driven, non-blocking servers. Used for traditional web sites and back-end API services. However, it was designed with real-time, push-based architectures in mind.

**Brief Overview:**

A digital take at Boston University on the popular senior game Assassin. A live-action game in which players try to eliminate one another using mock weapons, in an effort to become the last surviving player.

In this digital variant players mobile devices will utilize opt in GPS tracking to enable target acquisition across the span of Boston University’s one-mile campus. The technical details of the mobile application and client-side website are further elaborated upon throughout the document.

**General Gameplay:**

Since the game will require users to opt-in GPS tracking there will be a series of five *Assassination Times,* each being approximately five hours, over the course of a week. These *Assassination Times,* for example 7:00 AM - 3:00 PM, will be randomly determined at the beginning of each new week. During these *Assassination Times* players will be granted the ability to track their targets and assassinate them.

**Developers Note:** *Students may choose to opt out of a Assassination Week or the game itself. Assassin is meant to be an enjoyable experience and no student is by any means forced to participate if they do not wish to do so. The Assassination Time is necessary as this is designed to be a tense yet enjoyable experience for students. With that in mind having these designated times for gameplay allows students to truly enjoy the game at their own pace and not feel forced as if they were always being tracked.*

Players will each be assigned targets. Once the player's assigned target has been successfully assassinated the player will be assigned a new target. The game will rest on a weekly basis totaling a player's earned points for the week and updating the players statistics, campus leaderboard, etc. The player will receive points using two different criteria:

* The number of targets assassinated within a given week.
* The method of assassination used by the player.
* Number of events completed by the player.

**Methods of Assassination:**

* Melee
  + If the player is within *5 feet* of their target the *melee* assassination may be used. This will result in a successful assassination *100%* of the time.
* Ranged
  + If the player is within *15 feet* of their target the *ranged* assassination may be used. This will result in a successful assassination *50%* of the time.
    - **Note:** Once a ranged assassination is attempted (regardless of result) the player can no longer assassinate for the give *Assassination Time.*

**Events:**

* Weekly
  + There will be weekly goals or objectives player’s can elect to pursue which will offer bonus points.
    - Example: Complete 2 successful ranged assassinations.
      * Reward: 20+ bonus points.
* Monthly
  + Higher stakes events which offer bonus points and boosters or temporary advantages over other players.
    - Example: Complete 20 successful assassinations.
      * Reward: 60+ bonus points, +20% increase on ranged probability.
* Seasonal/Holiday:
  + Vanity events which will reward badges or profile specific flair.
    - *Longer term, will be designed as game advances through development phases.*
* Guild:
  + Randomly timed events which present an opportunity for players to score individual and guild points.
    - *Longer term, will be designed as game advances through development phases.*

**Features:**

* Home UI
  + Current target (displayed via GPS)
  + Timer
    - Time until next *Assassination Time* (takes precedent over weekly reset)
    - Time until weekly reset
  + Name and photo (if user chose to add a photo) of target
* Log-in screen:
  + Create Account option
    - Must @bu.edu email
  + Standard login
    - Email, Password
* Assassin Guilds:
  + Players can band up into groups, gain notoriety, and participate in group events
* Semester Leaderboard:
  + Leaderboard will reset by semester to ensure active players are known
* Assasination Screen:
  + Will appear as the player closes the gap between themself the target.
  + Simple one click button.
  + Two options:
    - Melee, which is the default assassination method
    - Ranged, which can be selected by a prompt if the player is within the designated range of their target

**Development Timeline:**

**Developer’s Note:** *This is a rough estimate of the project's timeline factoring in the current ability of the development team alongside the potential assistance of current UPE members.*

**Planning: (~2 weeks)**

* Including the week spent contributing to the development of this document.
* Should provide adequate time to scrape the former code of *Project Ezio* and attribute it to the features covered in this document.

**Design: (~4 weeks)**

* Determining what the end goal should look like to some degree.
* Primarily will focus on the Home UI and Profile.

**Content and Assembly: (~6 weeks)**

* Porting over content from existing sites or sources.
* Determining what data will be necessary to implement in the following development and programming process.

**Development and Programming: (~12 weeks)**

* Will use the design and specified content to begin creating the website.
  + Mobile version will be worked on after a successful web variant.
* Development team is still fairly new to this degree of web development. Therefore, this part of the process will be the most unpredictable.

**Beta Testing and Debugging: (~10 weeks)**

* Since the game is played on a weekly/monthly rotation-like basis.
* Will be tested amongst UPE members which will lead to further testing on a larger scale.
  + Example:
    - Other CS clubs which could provide solid feedback.

**Launch: (~4 weeks)**

* Once the game runs successfully for a month among a closed beta.
* Will be a rolling launch that will slowly accommodate more students interested in the game.

**Maintenance and Enhancements: (Ongoing)**

* Events will need to be continuously updated.
* Team will watch over application during the first month of its rolling launch to ensure game is running as planned.