Time of the Ages

Group 7 - Adolfo Moreno, John Toledo, Jose Hernandez, Pranjal Desai

Game Description and Overview

Time of the Ages is geo location based game that is available on all major mobile operations systems. The objective of this game while playing is to solve and complete story quests, some of which will require the help of other real world users to complete in order to level up your character ranking. This game utilizes a 3D graphics engine for rendering, although an option to downshift to 2D graphics is also available for lower end hardware on mobile platforms.

The main layout during gameplay is similar to PokemonGo, in the sense that a 3D map is displayed, a long with your character at the center of it. Character movement is based on current GPS location. To move your character to a specific location on the map you must walk towards that direction (n,s,w,e) until you've reached your destination physically, as well as in game. While playing on your instance of the game, you may potentially see other real in-game players, and have the option to message each other inside the app.

During gameplay players will encounter story quests which are optional missions that are placed in various locations ranging from historic locations, all the way to ordinary houses or parks. These quests are divided into non-fiction and fiction categories which are awarded variable number of token rewards, and character wearable items(ie. hats, capes, attire, pets etc.) upon completion which can utilized for purchasing in game-content. Fictional story quests will be depicted around historically accurate events that once occurred, with the exception of essentially reliving that scenario with you as the main character. This allows users to potentially learn about history without the boring traditional classroom setting.

The purpose of a quest is to make you as the player become the role for the main character as the story quests plot and objectives begin unraveling. Multiple quests may also be started simultaneously. During quests you may be asked to talk to NPC's for clues, travel to relatively close locations in search for quest related items, as well as riddles, and input from the user to advance in the story. These are the fundamentals for quests in the game.

The game will contain an in-game store that will get updated at least once a month, with content such as new in-game character attire, as well as additional quest stories some with substantially more rewards created by a special team of story developers. A portion of the monthly released content for the store will be available for purchase, and the remaining will be free to download.

Features

Key Functional Requirements

- Game UI should allow for the player to view stats, customize character appearance, adjust settings, and access content store for purchases**.**
- Game UI should allow for users to choose options in the menu such as-start a new game, resume, change settings, and logout.
- Game character should simulate walking, and update character location, as your geolocation changes.
- Game should allow for interaction through the user clicks at a particular location on the screen, which would engage an activity if there is any significance at a clicked location.
- Game character should be able to interact with other players, and in-game NPC's as well as start quests, that are within a certain radius(represented by an occasional light ring)
- Additional actions and options will be displayed if player clicks on the screen a significant area of interest (IE. read sign, enter building, attempt to search bush, cancel)

Key Non-functional Requirements

- **Data**- Will be supported by Amazon Cloud Services to provide a strong foundation for rapid scalability using elasticity during load peak times for client and server interactions.
- **Usability**-Should be easily understood by audiences 8+ with minor learning curve.
- **Dependability** – The game should not fail to be available more than once every 6 months for as most 30 mins of downtime, with the exception of crucial patch updates.
- **Performance** –Less than 1000ms client/server responses to ensure good user experience.

Design Goals

- **Efficiency:** This game application is only allowed to run in full screen window to ensure sufficient system resources are available for performing graphics rendering during gameplay.
- **Reliability:** Time of the ages has been extensively tested using unit and integration tests that reinforce core game functionality, to ensure a stable build for release of the game.
- **Availability:** This game has an uptime availability of 99.9%, 24/7. This is supported by a SAAS infrastructure on the server side that allow for rapid elasticity of resources during peak loads to accommodate rapid flux of user load and capacity.
- **Portability:** Cross platform availability on Android and IOS phones, and tablets.
- **Efficiency:** The game application design should maintain fast performance by offering an option for graphics that allow for graphics rendering modification that can be adjusted automatically or manually to optimize your systems hardware capabilities.
- **Security:** During gameplay, any connections and communication to the server are encrypted using SSL 3.0 certificates to avoid eavesdropping, and man in the middle attacks.
- **Extensibility:** The core game system is influenced by an object-oriented architecture as well as defined interfaces along with documentation to support future development. This allows for rapid modifications to the core system without causing the system to fail.

Testing Plans

Testing will be done by creating unit and integrations tests with a testing framework such as JUnit, or NUnit. We will also be issuing invitations for a beta run of our game. The public beta will begin 6 months prior to projected launch date. This Beta period will contain most core functionality, and will last approx. 3 months. This gives us 3 months to fix any serious bugs found during the beta phase, to increase a more successful stable launch on the first public release of the game.

Major features we will test for:

- 1) Ensure that the game on the client side is always transmitting data to the server using SSL.
- 2) Interaction with other real in-game players and NPC's (non-playable characters)
- 3) Ability to correctly recognize returning players state information
- 4) Whenever user input is required, validate and check for valid characters.

Test Cases:

- 1) During any gameplay attempt to eavesdrop using a proxy to observe network traffic – sensitive data should not be in readable format if secure transmission is in place.
- 2) Perform client side API calls to server- Assert expected result with actual result
- 3) Save state information in an external log prior to logging out. When logging back in, compare the current state information with the external log for any state differences(should be the same)
- 4) XSS(cross site scripting) and SQLI(SQL Injection) malicious strings in any user input prompt.