



Software Requirements Specification

for

Ayai-AI

Prepared by Marjorie Bartell,
Christian Benincasa,
Andrew Gotow,
Benjamin Kos,
Dan Muller,
Rory O'Kane and
Kyle Sheehan

Advisor: Dr. Santiago Ontañón

Drexel University, Senior Design 2015

Contents

Contents

- 1 Introduction
 - 1.1 Purpose
 - 1.2 Scope
 - 1.3 Audience
 - 1.4 Overview
- 2 Overall Description
 - 2.1 Product Perspective
 - 2.1.1 System Interfaces
 - 2.2 User Characteristics
- 3 Requirements
 - 3.1 Priorities Legend
 - 3.2 Functional Requirements
 - Installation
 - Ayai World Editor
 - The Ayai Web Application
 - Ayai Game Framework
 - 3.3 Non-Functional Requirements
- 4. Glossary

1 Introduction

1.1 Purpose

This document details the software requirements for the Ayai Massively Multiplayer Online (MMO) Game Project, with specific focus placed on the artificial intelligence and higher level content generation components of the system. The document describes the functional and non-functional requirements, constraints, and interfaces of the application.

1.2 Scope

This document describes the requirements for the second iteration of components for the Ayai Massively Multiplayer Online (MMO) Game Project. The components detailed in this document are:

- Ayai World Editor / Administration Page
- Front-end implementation of the Ayai project
- Back-end framework
- Game Definition / Configuration File
- Basic Agent AI Components
- Procedural Map and Quest Generation

1.3 Audience

The intended audience for this document includes the developers, testers, and end-users of the Ayai application.

1.4 Overview

Ayai is an Open-Source MMORPG project. It includes a framework that allows for the open source community to quickly create their own MMORPGs using the project's source code, and provides a way to seamlessly integrate research level AI into the MMORPG. The server-side framework exposes source code and API calls into the Game World management systems, including Entity and AI systems. Also included is a front-end implementation of the Ayai framework, and HTML5 game which employs Canvas and WebGL. Built on this framework is a series of components designed to explore the potential of Experience Management systems in the context of an online multiplayer game, as well as some preliminary components necessary for the implementation of significantly more extensive AI systems, such as detailed agent perception, and player behavior modeling.

2 Overall Description

2.1 Product Perspective

Currently no open source MMORPGs allow for a quick and seamless way to implement AI code into playable games. This stems from a poorly defined game state seen in most open source MMORPGs, which does not permit changing the attributes of in-game entities. Ayai intends to create a platform that allows for quick changes and additions to its AI code, along with an editor and system to easily add new components to the game (such as items and skills) In addition, few (if any) MMORPGs have explored the prospects of Experience Management AI to direct the flow of multiplayer games. This project also includes a simple system to model player behavior, and adaptively alter the game world to match.

2.1.1 System Interfaces

The Ayai system includes several internal systems to provide the desired functionality.

- Authentication: the Ayai system has an internal user login system that allows for verification of a user.
- Client Web Application: The Ayai software interfaces with the user's web browser and expects that it is capable of HTML 5.0
- Database: The Ayai database is used to store player account information, player game information (location, items, etc) and all items that can be stored in the game.
- Server: The server is used to run the game and send all updates to each user. It can be run on any operating system that uses the Scala Build Tool (sbt) and has Java 7 installed.

2.2 User Characteristics

End users of the Ayai game component do not need to have any experience with MMORPGs or video games in general. Developers of the open source repository should have some familiarity with the Scala programming language and familiarity with entity component systems.

3 Requirements

3.1 Priorities Legend

1. Required for release - A requirement that must be met for the project to be considered complete
2. Desired - A feature that could be implemented in the patches shortly after release
3. Luxury - A feature that would take at least two iterations of major patches to implement.

All priorities are marked next to each functional requirement as (P:X)

3.2 Functional Requirements

1. Installation

- 1.1. Installation Script - User executes an installation script. (P:1)
 - 1.1.1. Administrator - User creates superuser. (P:1)
 - 1.1.1.1. Email Address - User defines an email address for the superuser. (P:1)
 - 1.1.1.2. Confirm Email Address - User retypes email address for validation. (P:1)
 - 1.1.1.3. Password - User defines password for superuser. (P:1)
 - 1.1.1.4. Confirm Password - User confirms password for validation. (P:1)
 - 1.1.1.5. Password entry is hidden. (P:1)
 - 1.2. Run Script - User executes a script that launches necessary services.
 - 1.2.1. Debug Mode - User executes the script in debug mode for local testing. (P:1)
 - 1.2.2. Web Server - User launches the web server. (P:1)
 - 1.2.3. Game Server - User launches the game server. (P:1)

The screenshot shows the ayai game development platform's search interface. At the top right are links for 'admin', 'game', 'settings', and 'logout'. A search bar contains the word 'frost'. To the left is a sidebar with navigation links: 'game' (selected), 'npcs', 'items', 'props', 'classes', 'spells', 'effects', 'factions', 'quests', 'rooms', 'assets' (selected), 'tilesheets', 'spritesheets', 'community' (selected), 'users', 'groups', and 'analytics'. The main content area displays search results for 'frost':

- Frostbolt (spell)**
An ice spell that deals...
- Frost Wolf (NPC)**
A dangerous beast that will...
- Helm of the Frost King (item)**
Once owned by the Frost...

Figure 1: Searching for game assets

The screenshot shows the ayai game development platform's asset overview interface. At the top right are links for 'admin', 'game', 'settings', and 'logout'. A search bar contains the word 'search'. To the left is a sidebar with navigation links: 'game' (selected), 'npcs', 'items', 'props', 'classes', 'spells', 'effects', 'factions', 'quests', 'rooms', 'assets' (selected), 'tilesheets', 'spritesheets', 'community' (selected), 'users', 'groups', and 'analytics'. The main content area displays a list of spells under the 'spells' category:

name	last modified
frostbolt	January 3rd, 2014 (5:35 pm)
fireball	January 3rd, 2014 (5:30 pm)
heal	January 3rd, 2014 (5:15 pm)
attack	January 3rd, 2014 (4:35 pm)
dash	January 2nd, 2014 (9:35 am)

A 'new' button is visible in the top right corner of the main content area.

Figure 2: A sample overview

2. Ayai World Editor

2.1. Search (Fig. 1)

- 2.1.1. Search text field is located at the top of page. (P:1)
- 2.1.2. Scope - Finds all game assets that match the search keyword. (P:2)
- 2.1.3. Name - Results have a name. (P:1)
- 2.1.4. Description - Results have a short description. (P:2)
- 2.1.5. Action - User clicks on a search result to go to the item's page. (P:1)

2.2. Menu

- 2.2.1. NPCs - Users click a button to go to the NPC overview. (P:1)
- 2.2.2. Items - Users click a button to go to the items overview. (P:1)
- 2.2.3. Classes - Users click a button to go to the class overview. (P:1)
- 2.2.4. Spells - Users click a button to go to the spell overview. (P:1)
- 2.2.5. Effects - Users click a button to go to the effects overview. (P:1)
- 2.2.6. Quests - Users click a button to go to the quests overview. (P:1)
- 2.2.7. Spritesheets - Users click a button to go to the spritesheets overview. (P:1)
- 2.2.8. Tilesheets - Users click a button to go to the tilesheets overview. (P:1)

2.3. Overviews (This section applies to each entry in 2.2) (Fig. 2)

- 2.3.1. Items - Users click on an item to edit that entry. (P:1)
- 2.3.2. Name - Items list their name. (P:1)
- 2.3.3. Last Modified - Items list the date when they were last modified. (P:1)
- 2.3.4. Creation - Users click on a button to create a new entry. (P:1)
- 2.3.5. Users are directed to the item page upon creation. (P:1)
- 2.3.6. Users delete an item by clicking a button. (P:1)

2.4. Entries

- 2.4.1. Breadcrumb - Every entry has a breadcrumb visible at the top. (P:2)
- 2.4.2. Breadcrumb - User clicks part of the breadcrumb to redirect to that section. (P:2)
- 2.4.3. Save - Users click a button to save an entry. (P:1)
- 2.4.4. Cancel - Users click a button to cancel editing an entry. (P:1)
- 2.4.5. Cancel Confirmation - Users are presented with a dialog box when cancelling. (P:2)
 - 2.4.5.1. Yes - User clicks "Yes" to confirm. (P:2)
 - 2.4.5.2. No - User clicks "No" to confirm. (P:2)
 - 2.4.5.3. Validation - Users are prompted of any errors before saving. (P:2)

2.5. Game

2.5.1. NPC Entry

- 2.5.1.1. ID - The system assign entities an ID automatically. (P:1)
- 2.5.1.2. Name - Users assign a name with a text field. (P:1)
- 2.5.1.3. Visual - Users select available visual assets for the in-game appearance by typing an asset ID. (P:1)
- 2.5.1.4. Visual Search - Users search for available assets by typing search keywords into a search field. (P:2)
- 2.5.1.5. Health - Users type a default starting health for the players into a field. The maximum value is the largest 64 bit integer. (P:1)

2.5.2. Class Entry

- 2.5.2.1. Name - Users assign a name with a text field. (P:1)
- 2.5.2.2. Skills

- 2.5.2.2.1. Users add skills by typing their name into a filterable dropdown box. (P:1)
- 2.5.2.2.2. When added, skills appear underneath with their name. (P:1)
- 2.5.2.2.3. Users remove skills by clicking a button next to the skill. (P:1)
- 2.5.2.2.4. Criteria - Users can assign criteria to skills for when they are unlocked. (P:1)
- 2.5.2.3. Visual - Users select an available visual asset for the in-game appearance. (P:1)
- 2.5.2.4. Visual Search - Users search for available assets by typing search keywords into a search field. (P:2)
- 2.5.3. Item Entry
 - 2.5.3.1. Name - Users assign a name with a text field. (P:1)
 - 2.5.3.2. Criteria - Users assign stat or level requirements for use to the item. (P:1)
 - 2.5.3.3. Icons - Users assign an icon to items. (P:1)
 - 2.5.3.4. Actions
 - 2.5.3.4.1. Consumables - Users Items can assign an action which may consume the item. (P:1)
 - 2.5.3.4.2. Cooldowns - Users may be assign an action to an item which may trigger a cooldown. (P:1)
 - 2.5.3.4.3. Stat Bonuses - Users may assign multiple stat bonuses to an item which are applied to the character when equipped. (P:1)
 - 2.5.3.4.4. Equipable - Users may designate items as equipable or not equipable. (P:1)
- 2.5.4. Quest Entry
 - 2.5.4.1. Name - Users assign a name to quests. (P:1)
 - 2.5.4.2. Description - Users assign a description to quests. (P:1)
 - 2.5.4.3. Objectives - Users assign multiple objectives to quests. (P:1)
 - 2.5.4.4. Rewards - Users assign multiple rewards to quests which is given to the player upon completing all the objectives. (P:1)
 - 2.5.4.4.1. Items - Users assign items to be a reward. (P:1)
 - 2.5.4.4.2. Currency - Users assign currency to be a reward. (P:1)
 - 2.5.4.5. NPCs - Users assign multiple quests to NPCs for them to hand out. (P:1)
 - 2.5.4.6. Items - Users assign a quest to an item which is started when the item is picked up. (P:1)
- 2.5.5. Effect Entry
 - 2.5.5.1. Name - Users assign a name with a text field. (P:1)
 - 2.5.5.2. Description - Users assign a description with a text field. (P:1)
 - 2.5.5.3. Icons - Users can assign an icon with an upload button. (P:2)
 - 2.5.5.4. Icons - Users can assign an icon by searching previously uploaded icons with a filterable dropdown menu. (P:1)
 - 2.5.5.5. Stat Modification - Users assign stat modifications if applicable by typing it into a text field. (P:1)
 - 2.5.5.6. Duration - Users assign a duration by typing into a text field. (P:1)
- 2.5.6. Spell Entry (Fig. 3)
 - 2.5.6.1. Name - Users assign a name with a text field. (P:1)
 - 2.5.6.2. Description - Users assign a description with a text field. (P:1)
 - 2.5.6.3. Icons - Users can assign an icon with an upload button. (P:2)

- 2.5.6.4. Icons - Users can assign an icon by searching previously loaded icons with a filterable dropdown menu. (P:1)
 - 2.5.6.5. Damage - Users can assign a damage amount to a spell. (P:1)
 - 2.5.6.5.1. Damage Type - Users must assign a damage type with the damage. (P:1)
 - 2.5.6.6. Targeting - Users select how a spell can be targeted with a dropdown menu. (P:1)
 - 2.5.6.6.1. Allow for targeting a single entity via a click command. (P:1)
 - 2.5.6.6.2. Allow for targeting a group of entities via an AOE on click. (P:1)
 - 2.5.6.6.3. Allow for targeting a group of entities within a certain range of the caster. (P:1)
 - 2.5.6.6.4. Allow for targeting the caster and only the caster. (P:1)
 - 2.5.6.7. Effects - Users assign multiple effects to a spell that is applied to anyone targeted by the spell. (P:1)
 - 2.5.6.8. Range - Spells have a certain range in which they have an effect. (P:1)
- 2.6. Assets
- 2.6.1. Sprites
 - 2.6.1.1. Name- Users assign a name with a text field. (P:1)
 - 2.6.1.2. Upload - Users upload a spritesheet with an upload button. (P:1)
 - 2.6.2. Tilesheets
 - 2.6.2.1. Name - Users assign a name with a text field. (P:1)
 - 2.6.2.2. Upload - Users upload a tilesheet with an upload button. (P:1)

3. The Ayai Web Application

- 3.1. Splash
- 3.1.1. Email Field - User enters email address. (P:1)
 - 3.1.2. Password Field - User enters password. (P:1)
 - 3.1.3. User has the option to toggle password obfuscation. (P:1)
 - 3.1.4. Register Button - User clicks a button to register. (P:1)
 - 3.1.5. Login Button - User clicks a button to login. (P:1)
 - 3.1.6. User redirects to account page on registration or login. (P:1)
 - 3.1.7. Password Recovery Modal (P:2)
 - 3.1.7.1. User clicks a button to bring up a modal for password recovery. (P:2)
 - 3.1.7.2. User enters email address into a text field. (P:2)
 - 3.1.7.3. User submits information to initialize password recovery. (P:2)

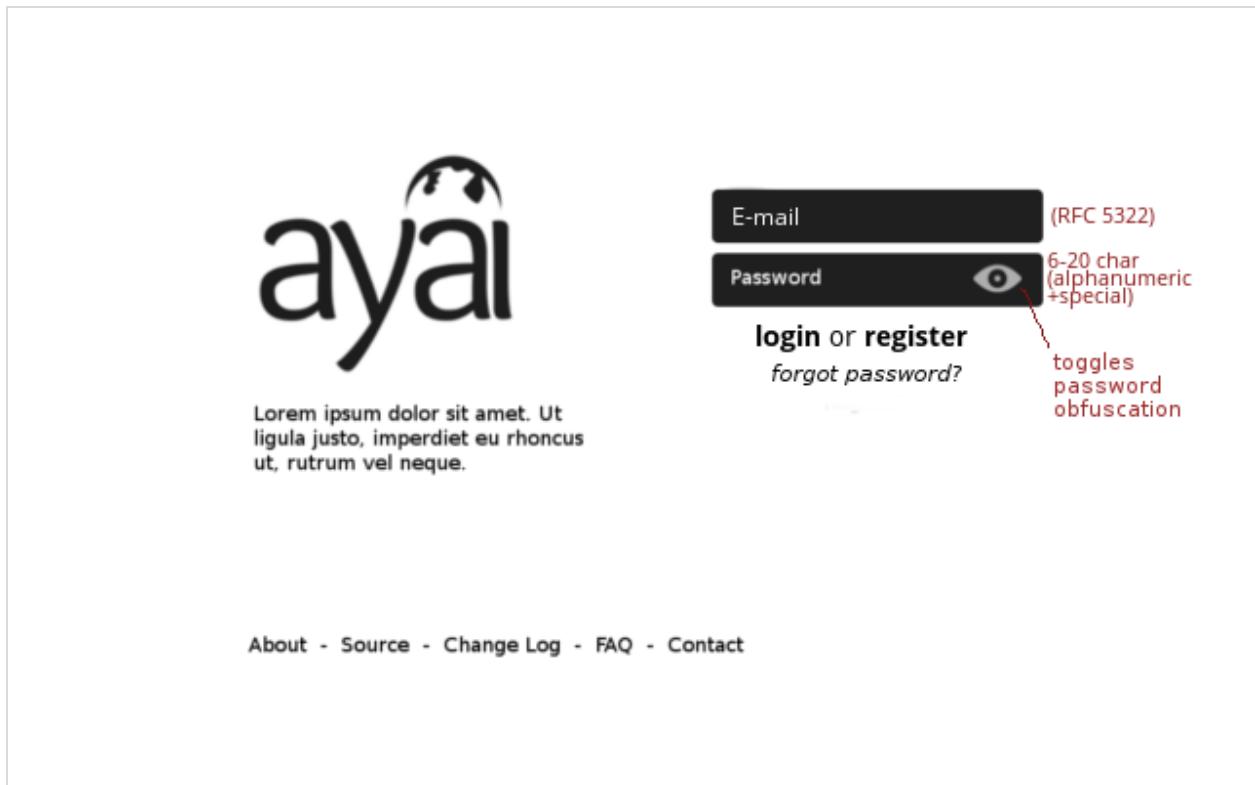


Figure 4: The login screen

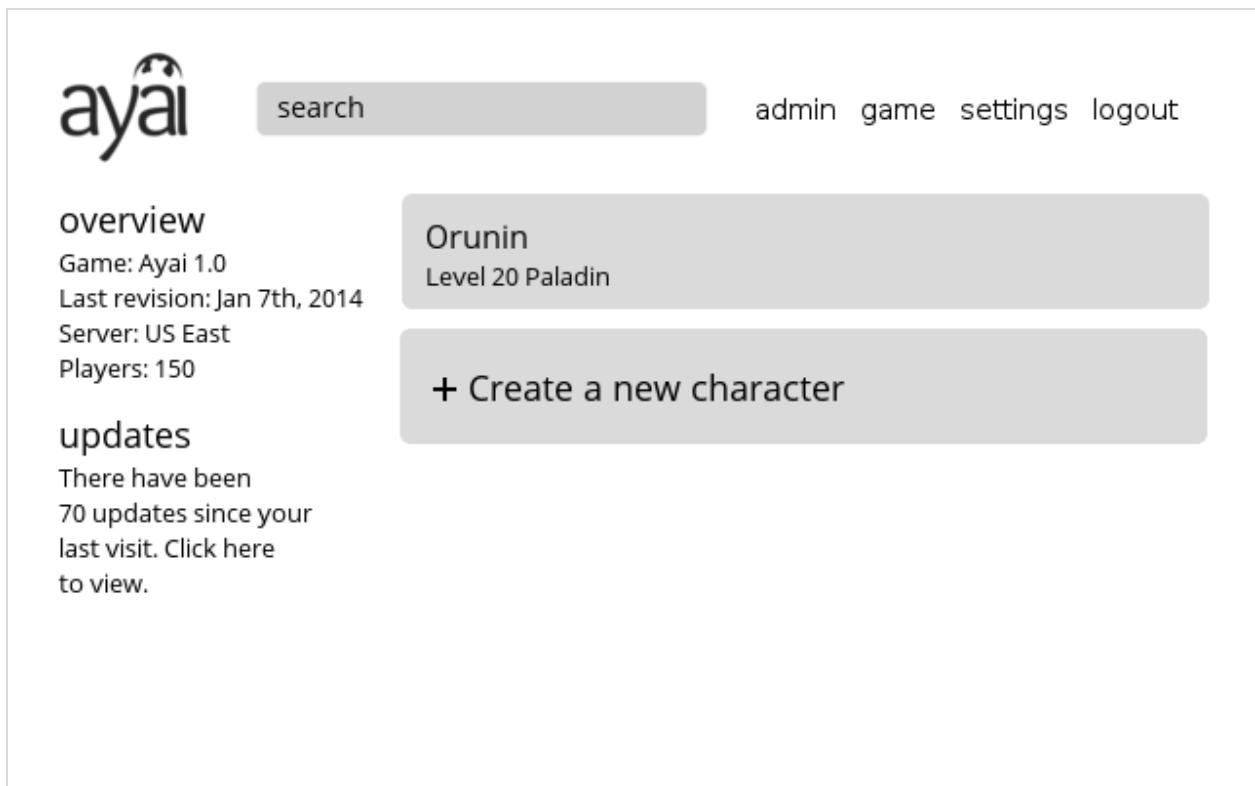


Figure 5: The character select screen

3.2. Account

3.2.1. Character Selection (Fig. 5)

- 3.2.1.1. System displays a list of previously created characters. (P:1)
- 3.2.1.2. Name - Characters display their name. (P:1)
- 3.2.1.3. Level - Characters display their level. (P:1)
- 3.2.1.4. Class - Characters display their class. (P:1)
- 3.2.1.5. Character Selection - User clicks a character to log into game. (P:1)
- 3.2.1.6. Character Creation - Users select a button to create a new character. (P:1)
- 3.2.1.7. Server Selection - Users have a menu to select a server. (P:3)

3.2.2. Character Creation (Fig. 6)

- 3.2.2.1. Cancellation - User clicks a cancel button to go back to the previous screen. (P:1)
- 3.2.2.2. Class Options - System displays a list of all character classes. (P:1)
- 3.2.2.3. Name - System lists classes by their name. (P:1)
- 3.2.2.4. Class Selection - User clicks a class's name to select it. (P:1)
- 3.2.2.5. Avatar - The system shows the class avatar when that class is selected. (P:2)
- 3.2.2.6. Description - The system shows the class description when that class is selected. (P:1)
- 3.2.2.7. Stats - The system shows the class starting stats when that class is selected. (P:1)
- 3.2.2.8. Skills - The system shows the class starting skills when that class is selected. (P:1)
- 3.2.2.9. Name Field - User enters a name for their character. (P:1)
- 3.2.2.10. Submit - User clicks a button to create a character with the supplied information. (P:1)
- 3.2.2.11. Validation - System prompts user with an error, if a name is already taken. (P:1)

3.2.3. Settings (Fig. 7)

- 3.2.3.1. Settings Link - User clicks a link to open the settings page. (P:1)
- 3.2.3.2. Email Field - User enters a new email address to change it. (P:1)
- 3.2.3.3. Password - User enters a new password to change it. (P:1)
- 3.2.3.4. User has the option to toggle password obfuscation. (P:1)
- 3.2.3.5. Save Button - User clicks a button to save entered information. (P:1)

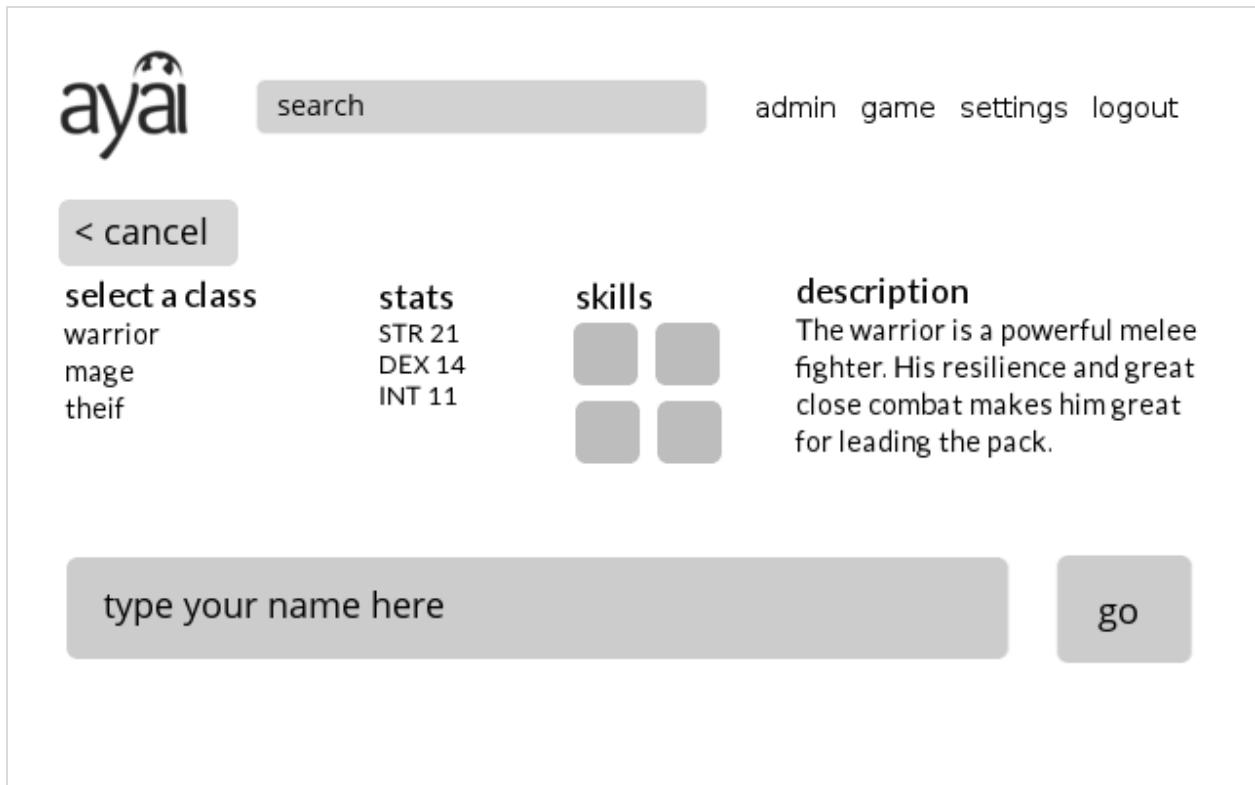


Figure 6: The character creation screen



Figure 7: The settings screen

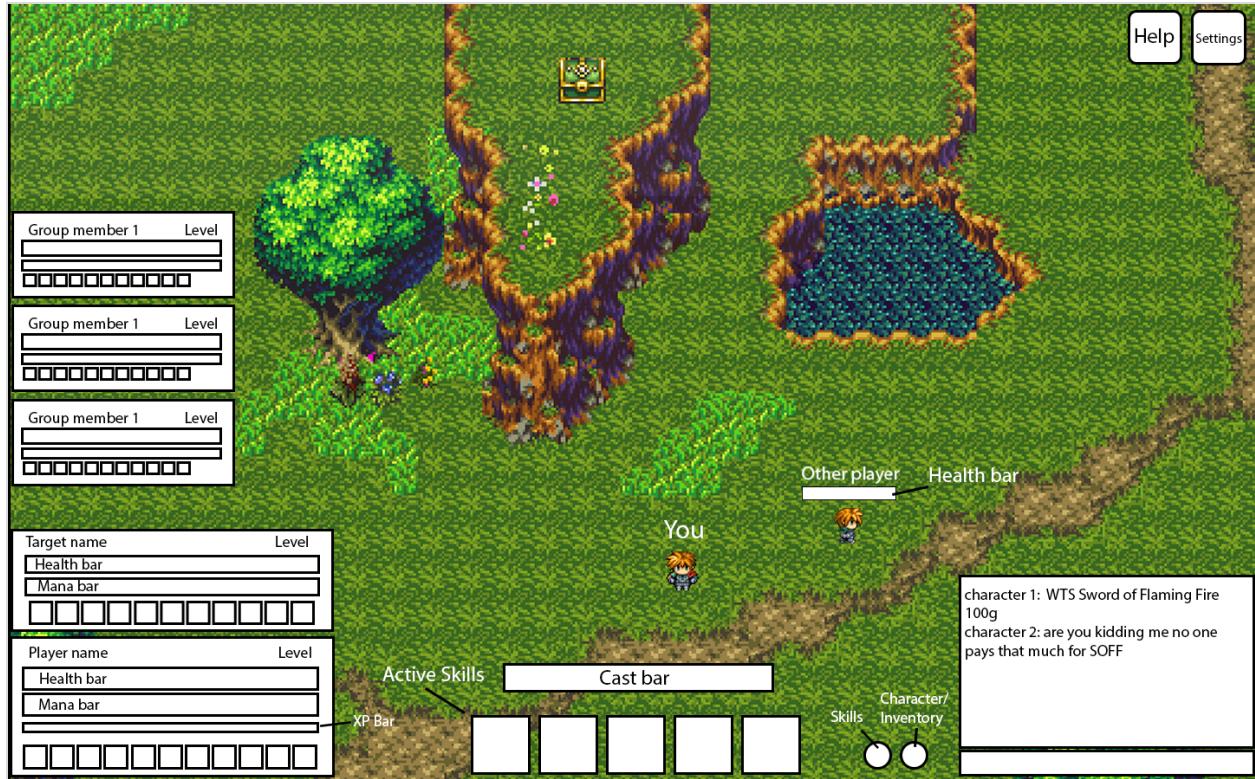


Figure 8: The game screen

3.3. Game Screen (Fig. 8)

3.3.1. Player Unitframe

- 3.3.1.1. Screen displays Player Unitframe in the bottom left corner of the screen
- 3.3.1.2. Screen displays Player name in the Player Unitframe
- 3.3.1.3. Screen displays Player level in the Player Unitframe
- 3.3.1.4. Screen displays Player's Health bar in the Player Unitframe as the ratio of a player's current health value (ex: 80/100, is only .8 of full size) (P:1)
- 3.3.1.5. Screen displays Player's mana bar in the Player Unitframe as the ratio of a player's current mana value (ex: 80/100, is only .8 of full size) (P:1)
- 3.3.1.6. Screen displays Player's Experience bar in the Player Unitframe as the ratio of a player's current experience value (ex: 80/100, is only .8 of full size) (P:1)
- 3.3.1.7. Status Effect boxes (Fig. 8), underneath XP bar - Screen displays icons for all of the Player's current status effects underneath XP bar (P:1)
 - 3.3.1.7.1. Screen displays remaining time of status effect on bottom of status effect box (P:1)

3.3.2. Target Unitframe

- 3.3.2.1. Screen displays Target Unitframe in the bottom left corner of the screen, above Player Unitframe
- 3.3.2.2. Screen displays Target name in the Target Unitframe
- 3.3.2.3. Screen displays Target level in the Target Unitframe
- 3.3.2.4. Screen displays Target's Health bar in the Target Unitframe as the ratio of a Target's current health value (ex: 80/100, is only .8 of full size) (P:1)

- 3.3.2.5. Screen displays Target's mana bar in the Target Unitframe as the ratio of a Target's current mana value (ex: 80/100, is only .8 of full size) (P:1)
- 3.3.2.6. Status Effect boxes (Fig. 8), underneath Mana bar - Screen displays icons for all of the Target's current status effects underneath Mana bar (P:1)
 - 3.3.2.6.1. Screen displays remaining time of status effect on bottom of status effect box (P:1)
- 3.3.3. Group Unitframes
 - 3.3.3.1. Screen displays a Group Unitframe for each member currently in the Player's group on the left side of the screen, above the Target Unitframe (P:2)
 - 3.3.3.2. Screen displays a group member's name in each Group Unitframe (P:2)
 - 3.3.3.3. Screen displays a group member's level in the Group Unitframe (P:2)
 - 3.3.3.4. Screen displays a Health bar in each Group Unitframe as the ratio of a group member's current health value (ex: 80/100, is only .8 of full size) (P:2)
 - 3.3.3.5. Screen displays a Mana bar in each Group Unitframe in the Target Unitframe as the ratio of a group member's current mana value (ex: 80/100, is only .8 of full size) (P:2)
 - 3.3.3.6. Status Effect boxes (Fig. 8), underneath Mana bar - Screen displays icons for all of the group member's current status effects underneath Mana bar (P:2)
 - 3.3.3.6.1. Screen displays remaining time of status effect on bot- tom of status effect box (P:2)
- 3.3.4. Player avatars
 - 3.3.4.1. Screen shows all avatars in the map location of the player. (P:1)
 - 3.3.4.2. System shows a portrait of the player's avatar next to the status bars on the edge of screen (P:1)
 - 3.3.4.3. Other Players' Health Bars -
 - 3.3.4.3.1. System displays a bar above other players avatars showing the ratio of that players health (example: if player has 80/100 health, then bar is .8 of size) (P:1)
 - 3.3.4.4. Other Players Mana Bar -
 - 3.3.4.4.1. System displays a bar above others players avatars on the screen showing only the ratio of the other players mana ratio (P:1)
- 3.3.5. Action boxes
 - 3.3.5.1. Skill Boxes (Fig. 8), large boxes in middle of screen
 - 3.3.5.1.1. System displays icon of skill at bottom of screen (P:1)
 - 3.3.5.1.2. System displays keyboard representation of skill to action (P:1)
 - 3.3.5.1.3. Player clicks on box to use skill (P:1)
 - 3.3.5.1.4. Player presses the accompanying keyboard representation to use skill (P:1)
- 3.3.6. Game Interface Buttons
 - 3.3.6.1. Inventory Button (Fig. 9)
 - 3.3.6.1.1. System displays a button representing inventory next to Action Boxes (P:1)
 - 3.3.6.1.2. Player clicks button to display inventory view (P:1)
 - 3.3.6.1.3. Player pressing the key corresponding to inventory displays inventory view. (P:1)

3.3.6.2. Journal Button

- 3.3.6.2.1. System displays a button representing Journal next to action boxes. (P:1)
- 3.3.6.2.2. Player clicks button to display Journal view. (P:1)
- 3.3.6.2.3. Player pressing the key corresponding to Journal displays Journal view. (P:1)

3.3.6.3. "Other Players" Button

- 3.3.6.3.1. The system displays a button representing "Other Players" next to Action Boxes (P:1)
- 3.3.6.3.2. Player clicks a button to display "Other Players" view (P:1)
- 3.3.6.3.3. Pressing key action of "Other Players" button displays view (P:1)

3.3.7. Help Button (Fig. 13)

- 3.3.7.1. System displays button representing help in the upper right corner (P:1)
- 3.3.7.2. Player clicks a button to display help overlays (P:1)
- 3.3.7.3. Player presses the key corresponding to help to display help overlays (P:1)

3.3.8. Settings button (Fig.10)

- 3.3.8.1. System displays button representing settings in upper right corner (P:1)
- 3.3.8.2. Player clicks the settings button to display options in settings view (P:1)
- 3.3.8.3. Player presses the key corresponding to settings to display options in the settings menu (P:1)

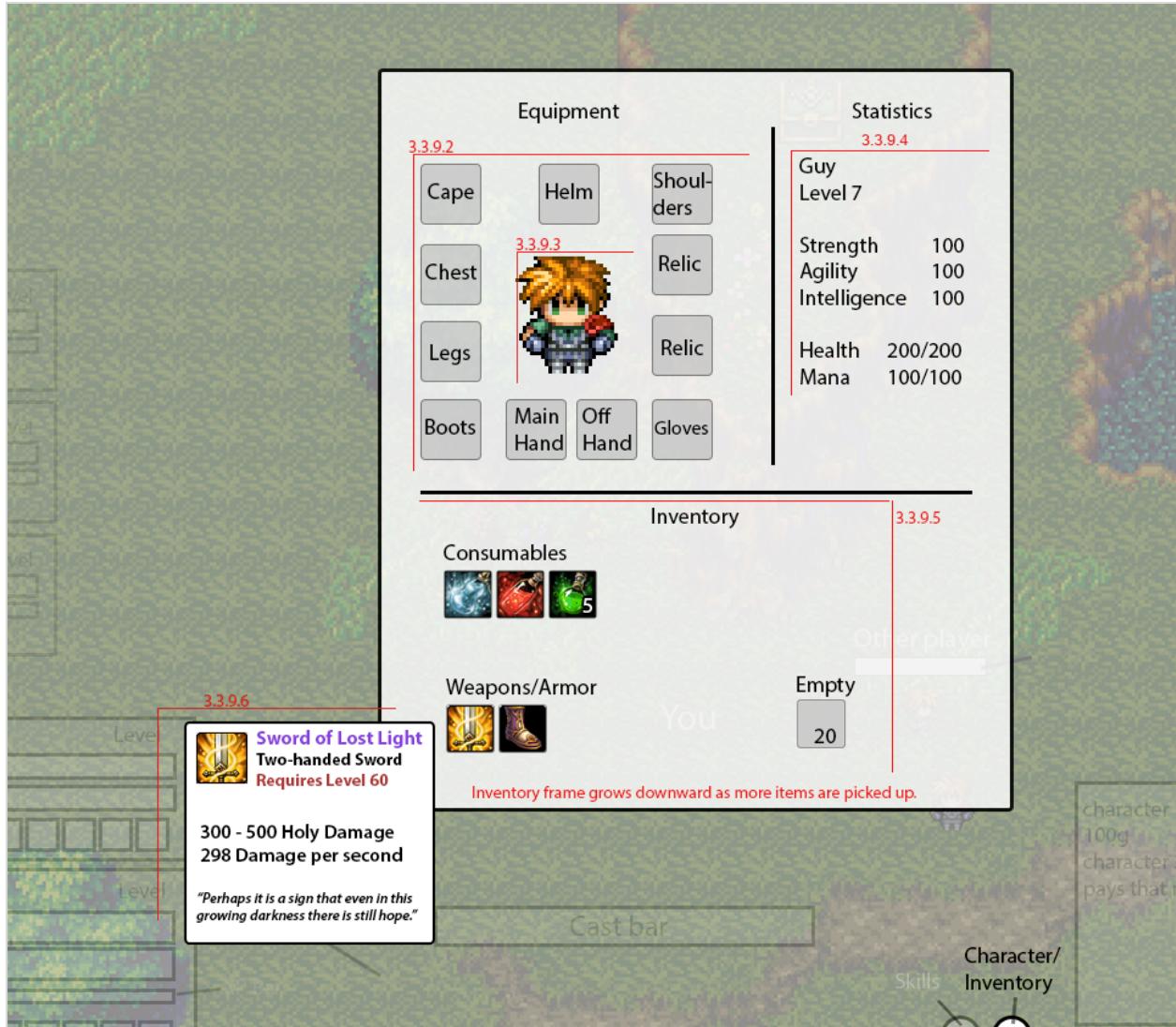


Figure 9: Player Inventory

3.3.9. Inventory (Fig. 9)

- 3.3.9.1. View - The inventory view is a modal window which displays the character's currently equipped items (Equipment Slots), the character's attributes (Statistics), and the items that the character is currently carrying (Inventory Slots) (P:1)
- 3.3.9.2. Equipment Slots - Displays items that the player's character currently has equipped (P:1)
 - 3.3.9.2.1. Equipping Item - If the player double clicks an equippable item in the inventory, or drags an equippable item from the inventory onto the character model or an equipment slot, the item is equipped into the correct equipment slot (P:1)
 - 3.3.9.2.2. Unequipping Item - If the player drags an equipped item from an equipment slot into an inventory slot, the item is unequipped (P:1)

- 3.3.9.3. Character Model - Displays a preview of the player's character with his or her equipped items (P:1)
- 3.3.9.4. Statistics - Displays information relating to the status of the player's current character (P:1)
 - 3.3.9.4.1. Player Name - Displays player character's name (P:1)
 - 3.3.9.4.2. Player Level - Displays player character's level (P:1)
 - 3.3.9.4.3. Attributes - Displays values relating to the overall effectiveness of the player's character (P:1)
- 3.3.9.5. Inventory Slots - Displays items currently in the player character's inventory (P:1)
- 3.3.9.6. Item Tooltip - If a player mouses over an item in the inventory view, a tooltip appears containing additional information about the item. (P:1)
 - 3.3.9.6.1. Item Name - Displays the item name (P:1)
 - 3.3.9.6.2. Item Quality - Deones the quality / rarity of the item by the color of the Item Name (P:1)
 - 3.3.9.6.3. Equipment Type - If equippable, displays the equipment type of the item. (P:1)
 - 3.3.9.6.4. Required Level - Displays the character level required to use/equip the item (P:1)
 - 3.3.9.6.5. Item Details - Displays details about the item including the amount of damage / protection / healing that the item provides (P:1)
 - 3.3.9.6.6. Flavor Text - Displays plot-related quote or description about the item (P:1)
- 3.3.9.7. Using an Item - Players double click an item in the inventory view to use it. (P:1)
 - 3.3.9.7.1. Item not usable - If item is not usable, prompt on screen tells player that it cannot be used at this time (P:1)
- 3.3.9.8. Opening the inventory - System displays a button which when clicked opens the inventory view (P:1)
- 3.3.9.9. Closing Inventory Screen
 - 3.3.9.9.1. Key Press - Player presses the escape key to close the inventory screen (P:1)
 - 3.3.9.9.2. Close button - Player presses the button (X button) on the upper right of the inventory to close the screen (P:1)
- 3.3.9.10. Dragging an item
 - 3.3.9.10.1. Dragging and dropping an item off of the grid brings up an OK/Cancel confirmation dialogue (P:1)
 - 3.3.9.10.1.1. If the player clicks OK, the system deletes the item forever (P:1)
 - 3.3.9.10.1.2. If the player clicks Cancel, the system does not delete the item, and the item keeps its place in the inventory. (P:1)
 - 3.3.9.10.2. Dragging and dropping an item with another swaps the positions of the two items. (P:1)
 - 3.3.9.10.3. Dragging and dropping an item with an empty slot moves the item to that slot (P:1)
- 3.3.10. Chat Box and Text Field
 - 3.3.10.1. Chat Box

- 3.3.10.1.1. Screen displays the Chat Box in the lower right corner of the screen (P:2)
- 3.3.10.1.2. Chat box displays messages received by the server (P:2)
- 3.3.10.2. Text Field
 - 3.3.10.2.1. Screen displays a Text Field underneath the chat box and allows the user to type a message in the Text Field when the player presses Enter (P:2)
 - 3.3.10.2.1.1. Message is sent to server and is displayed in Player's Chat box when the player presses Enter again. Screen no longer displays Text Field (P:2)
- 3.3.11. Minimap
 - 3.3.11.1. Show other players
 - 3.3.11.1.1. Screen shows a small map that conveys the player's location and the immediate area outside of the player's viewable screen (P:2)
 - 3.3.11.2. Show Quests
 - 3.3.11.2.1. Screen shows directions to any active quests a player has (P:2)
 - 3.3.11.3. Interact - If a player right clicks on an NPC, the NPC performs an action. (P:1)
 - 3.3.11.4. Dialogue - NPCs have a predefined message that they say when activated. (P:1)
 - 3.3.11.5. Quests - If an NPC has a quest, they give it to the player when activated (P:1)

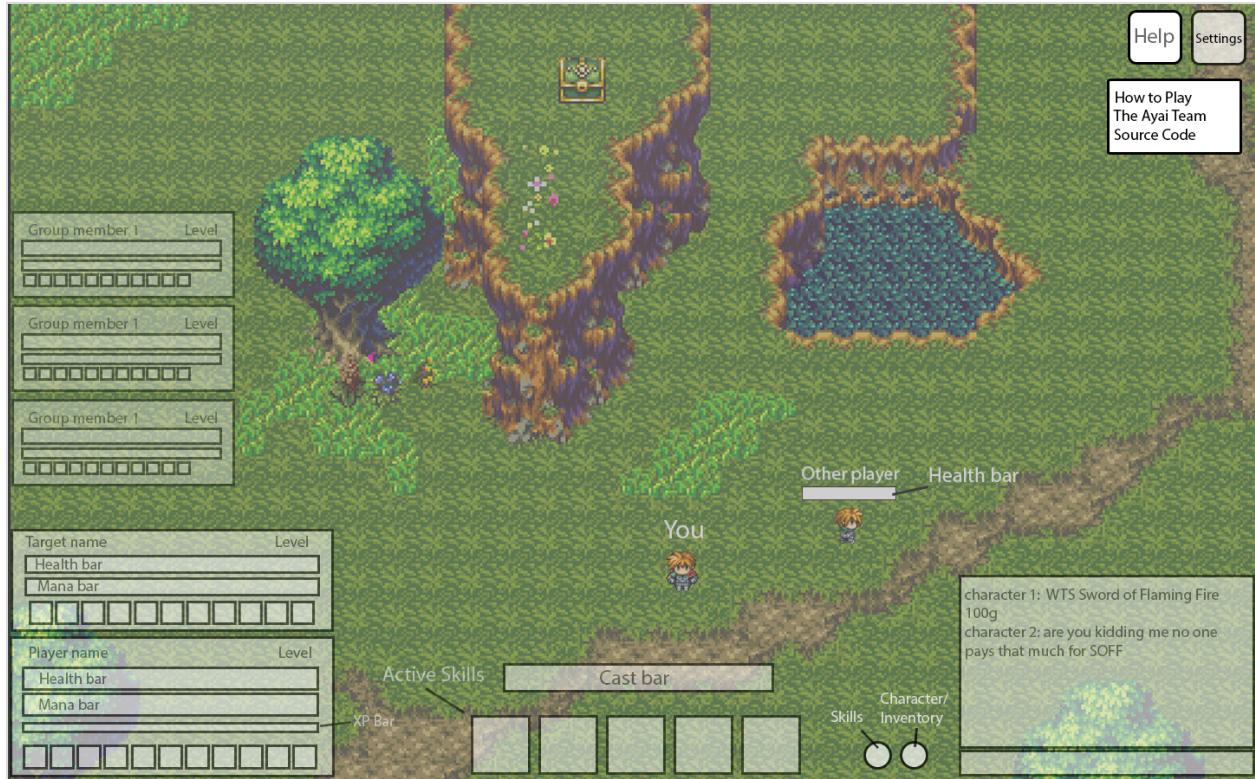


Figure 10: Settings Menu

3.4. Settings Menu (Fig. 10)

3.4.1. 3.4.1 Controls Screen Button

3.4.1.1. Player clicks button to bring up the “Control Screen” which shows game actions and the corresponding keys that are set to activate them (P:1)

3.4.2. Help Screen Button

3.4.2.1. Player clicks button to bring up “Help Screen” overlay and conveys information about the game world and initial objectives of the game (P:1)

3.4.3. Sound Screen Button

3.4.3.1. Player clicks button to bring up “Sound Screen” overlay and the sound controls which the player can change to their preference (P:1)

3.4.4. Switch Character Button

3.4.4.1. Player clicks button to bring up an OK/Cancel confirmation dialog.

3.4.4.1.1. If the player clicks OK, the system logs the player out, tells the server to save the player’s current data and takes the player back to the character select screen (P:1)

3.4.4.1.2. If the player clicks Cancel, the confirmation dialog closes and no further action is taken (P:1)



Figure 11: Logout Confirmation

3.4.5. Quit Game Button (Fig. 11)

3.4.5.1. Player clicks button to bring up an OK/Cancel confirmation dialog.

3.4.5.1.1. If the player clicks OK, the system logs the player out, tells the server to save the player's current data, and takes the player back to the login screen (P:1)

3.4.5.1.2. If the player clicks Cancel the confirmation dialog closes and no further action is taken (P:1)

3.5. Control Screen (Fig. 12)

3.5.1. Change key binding of action

3.5.1.1. When a player clicks on an action (example: Key "A" is for "Move Left Action"), the system prompts the player to enter another key and changes the value to that key. (Fig. 12) (P:1)

3.5.2. Show all available actions

3.5.2.1. Control screen shows all actions and key bindings that are available in game (P:1)

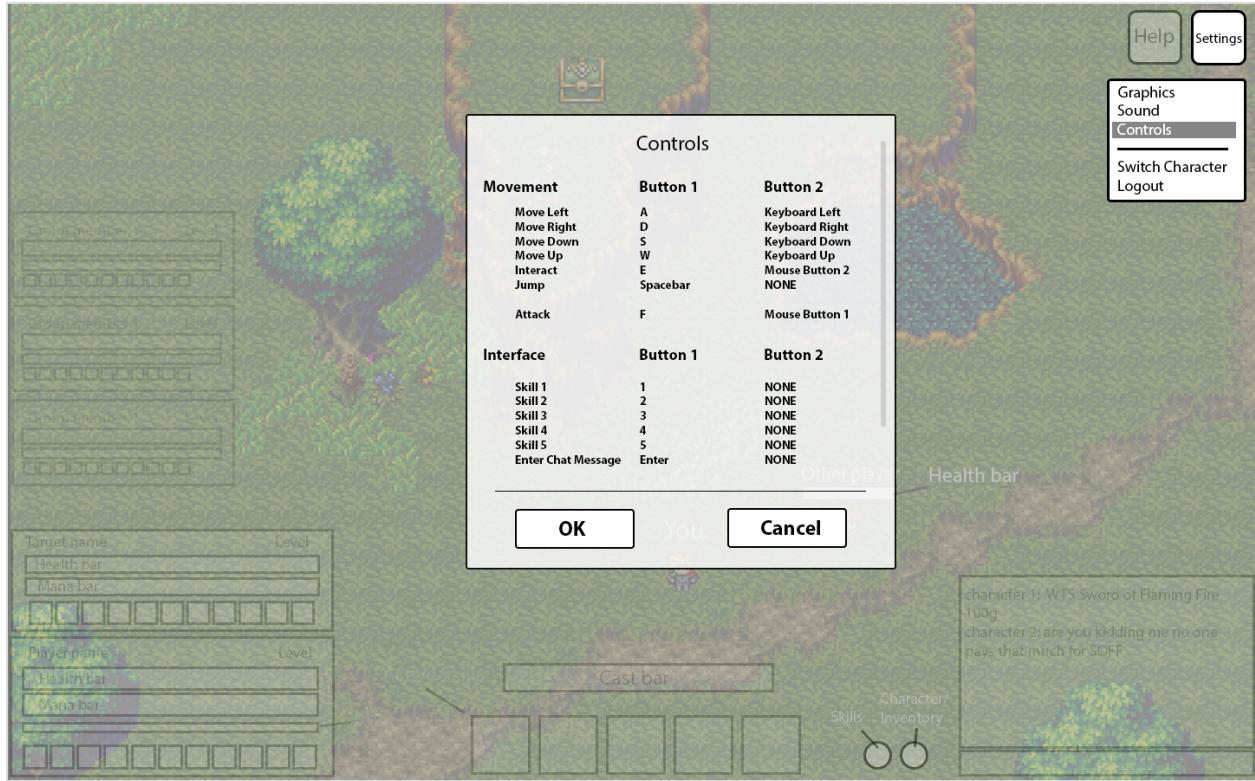


Figure 12: Control Screen

3.6. Help Screen

3.6.1. Help Screen Overlay

3.6.1.1. Arrows between each page

3.6.1.1.1. The system displays back and forward arrows to read through help tutorials (P:1)

3.6.1.2. Page Specifics

3.6.1.2.1. Ayai Tutorial - The system displays a screen which explains what Ayai is and the goal of the game (P:1)

3.6.1.2.2. Control Example - The system displays a screen which explains how to move your character and small examples of boundaries (P:1) (Fig. 13)

3.6.1.2.3. Attacking - The system displays a screen which explains different types of attacks (P:1) (Fig. 14)

3.6.1.2.4. Quests - The system displays a screen which explains how to accept and track quests (P:1)



Figure 13: Help Screen Overlay - Movement



Figure 14: Help Screen Overlay - Attacks/Skills



Figure 15: Sound Screen

3.7. Sound Screen (Fig. 15)

3.7.1. Master Volume Value

3.7.1.1. Screen has a text field or sliding bar that allows players to change the volume of sound effects in the game. (P:1) (Fig 15)

3.7.2. Cancel button

3.7.2.1. Player presses cancel button to ignore changes made and return to main screen.

3.7.3. Ok/Submit Button

3.7.3.1. Player presses OK button to accept the new values in the bars and save the values.

3.8. Other players Screen

3.8.1. Who's playing in your region overlay

3.8.1.1. Screen displays the names, classes, and levels of other players in the room that the current player is in (P:1)

3.8.2. Friends Overlay

3.8.2.1. Screen displays friends and their locations (What region they are in) and stats. (P:2)

3.8.3. Messaging a Friend

3.8.3.1. When a player double clicks on a friend's name, the system enables the chat box and automatically sets it to private message the friend (P:2)

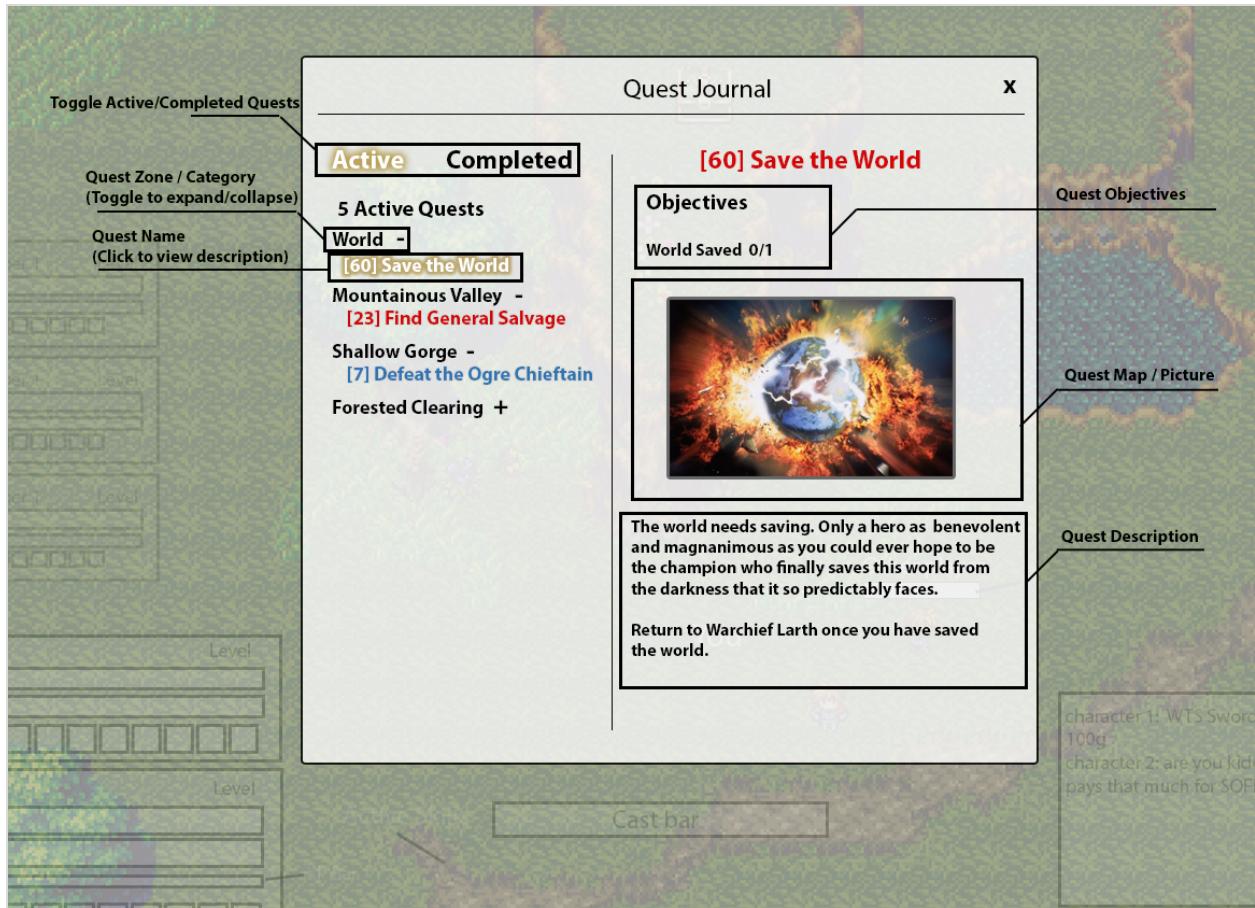


Figure 16: Journal Screen

3.9. Journal Screen (Fig. 16)

3.9.1. List Quests

3.9.1.1. Screen displays the name of the quest (P:1)

3.9.1.1.1. Quest names are unique (P:1)

3.9.1.2. Screen shows destination of quest (P:1)

3.9.1.3. Screen shows objectives / key items (P:1)

3.9.1.4. Screen shows backstory of quest (P:1)

3.9.1.5. Screen shows rewards for completing a quest (P:1)

3.9.2. Screen shows all accepted quests the player has (P:1)

3.9.3. Show All Quests toggle

3.9.3.1. When clicked, "Show All Quests" toggle shows both completed and active quests the player is on (P:1)

3.9.4. Player Sounds

3.9.4.1. Attacking Sounds

3.9.4.1.1. Attacking with a physical attack has a unique sound effect (P:1)

3.9.4.1.2. Attacking with a projectile attack has a unique sound effect (P:1)

3.9.4.1.3. Casting a spell has a unique sound effect (P:1)

3.9.4.2. Walking Sound - The system will play footstep sounds when the character walks. These depend on the type of surface the character is walking across (P:1)

- 3.9.5. UI Sounds (Menu Sounds) - The system plays a click sound when a player clicks on a button (P:1)
- 3.9.6. Music - Music changes depending on the room the player is currently in (P:1)

4. Ayai Game Framework

4.1. AI

4.1.1. Entity AI

- 4.1.1.1. AI Entity Detection - The framework exposes a hook which allows a developer to add any number of detection algorithms to an entity which allow the entity to determine proximity to other entities. Ayai implements the two examples below. (P:1)
 - 4.1.1.1.1. Entity Sight
 - 4.1.1.1.1.1. Entities determine that objects and other entities are perceived by sight via Line-of-Sight (LoS) algorithms (P:1)
 - 4.1.1.1.1.2. If an entity is aware of another entity, it knows when it is in the LoS of the other entity (P:1)
 - 4.1.1.1.2. Entity Hearing
 - 4.1.1.1.2.1. Sound Entities are created when certain entities and/or objects interact in predetermined ways (P:1)
 - 4.1.1.1.2.2. Entities perceive objects and other entities by sound when in range of a sound entity (P:1)
 - 4.1.1.1.2.3. Entities who are in range of a sound entity become aware of the source of the sound entity (P:1)
 - 4.1.1.1.2.4. Sounds may be occluded by solid tiles in the world (P:3)
- 4.1.1.2. AI Entity Interaction
 - 4.1.1.2.1. Entity Memory
 - 4.1.1.2.1.1. Entities keep a record of other previously perceived entities (P:3)
 - 4.1.1.2.1.2. After a predetermined amount of time, entities forget that they have perceived another entity or sound entity (P:3)
 - 4.1.1.2.2. Entity Communication
 - 4.1.1.2.2.1. Entities propagate known information to other entities (P:1)
- 4.1.1.3. AI Entity Pathfinding
 - 4.1.1.3.1. AI Pathfinding components conform to a common interface
 - 4.1.1.3.2. A* - The A* algorithm implementation (P:1)
 - 4.1.1.3.2.1. Heuristic functions conform to a common interface, and may be swapped out easily (P:1)
 - 4.1.1.3.2.2. Algorithm takes into account various attributes of tiles in the map (height, roughness, terrain type) (P:1)
 - 4.1.1.3.3. D* Lite - The D* Lite algorithm implementation (P:2)
- 4.1.1.4. AI Melee Confrontation - The framework exposes a hook which allows a developer to add AI to an entity that allows the entity to execute a melee attack on a target entity (P:1)

- 4.1.1.5. AI Ranged Confrontation - The framework exposes a hook which allows a developer to add AI to an entity that allows the entity to execute a ranged attack on a target entity (P:1)
- 4.1.1.6. AI Spell Casting - The framework exposes a hook which allows a developer to add AI to an entity that allows the entity to cast a spell on a target entity (P:1)
- 4.1.2. Director AI
 - 4.1.2.1. AI Quest Generation - The framework exposes a hook which allows a developer to add AI to generate quests for users (P:1)
 - 4.1.2.1.1. World State - The Director AI generates and updates a persistent world model to manage global state of the world. The global state contains all information needed to completely reconstruct the game state at any point, including player behavior data. (P:1)
 - 4.1.2.1.1.1. All actions taken by any player in the world update the World State using a model of player behavior. (P:2)
 - 4.1.2.1.1.2. The World State is used to generate Quests for NPCs and the Director AI assigns these quests to the player via Quest Selection Algorithms (P:1)
 - 4.1.2.1.1.2.1. Types of Quests
 - 4.1.2.1.1.2.1.1. “Fetch Quests” require players to retrieve items and deliver them to a location. Items are any objects chosen from the “Items” collection in the Quest Editor
 - 4.1.2.1.1.2.1.2. “Kill Quests” require players to eliminate a given NPC enemy or character.
 - 4.1.2.1.1.2.1.3. “Escort Quests” require players to protect an NPC as they approach a target. The target is a coordinate on the World Map.
 - 4.1.2.1.1.2.2. Quest Selection Algorithms
 - 4.1.2.1.1.2.2.1. Quests are selected due to parameters defined before generation.
 - 4.1.2.2. AI Enemy Spawning - The framework exposes a hook which allows a developer to add AI to spawn enemies in an area during a quest. (P:1)
 - 4.1.2.2.1. The Director AI generates NPCs as required by quests assigned to the player (P:1)
 - 4.1.2.2.1.1. Types of NPCs
 - 4.1.2.2.1.1.1. “Friendly” NPCs have limited interaction with the player and issue quests
 - 4.1.2.2.1.1.2. “Neutral” NPCs ignore the player until threatened, and may become “Hostile” NPCs. While unthreatened, Neutral NPCs wander randomly within a fixed distance of their spawn point.
 - 4.1.2.2.1.1.3. “Hostile” NPCs attack the player when detected (see 4.1.1 AI Entity Detection). “Hostile” NPCs

- may move any distance from their spawn point to follow a player
- 4.1.2.2.1.1.4. “Companion” NPCs follow the player, and aid them in combat by fighting a player’s enemies. “Companion” NPCs stay within a fixed distance from the player, but may be instructed to wait at a specific location by the player
- 4.1.2.3. AI Map Generation - The framework exposes a hook which allows a developer to add AI to generate new areas based on the generation of a new quest (P:1)
- 4.1.2.3.1. Map Classifications
- 4.1.2.3.1.1. World Map
- 4.1.2.3.1.1.1. A World map is composed of tiled Local Maps
 - 4.1.2.3.1.1.2. A World map has no size limit
 - 4.1.2.3.1.1.3. A World map is initially composed of MxM Local Maps before generation takes place.
 - 4.1.2.3.1.1.4. A World map is shared amongst all players on the server.
- 4.1.2.3.1.2. Local Maps
- 4.1.2.3.1.2.1. A local map is an NxN map and acts as a sub-section of the World Map.
 - 4.1.2.3.1.2.2. A local map is classified into one of four types.
 - 4.1.2.3.1.2.2.1. Open Air - Composed entirely of open air (i.e. exposed to the open sky) tiles
(Snow, mountain, etc.)
 - 4.1.2.3.1.2.2.1.1. Generated when any player crosses a generation threshold on the world map
 - 4.1.2.3.1.2.2.1.2. Generation Thresholds are determined by proximity to ungenerated Local Maps
 - 4.1.2.3.1.2.2.2. Dungeon - Composed entirely of dungeon tiles. Dungeon maps are of any size, and are not contained within the world map.
 - 4.1.2.3.1.2.2.2.1. One dungeon map is generated before entry for each player on the server.
 - 4.1.2.3.1.2.2.2.2. All dungeon maps have at least one entrance, and one exit, and may have more
 - 4.1.2.3.1.2.2.3. Indoor - Composed entirely of indoor tiles. Indoor maps are of any size, and are not contained within the world map
 - 4.1.2.3.1.2.2.3.1. Indoor maps are not dynamically generated and are predefined in the editor.

4.1.2.3.1.2.2.4. Transition - Composed of transitioning open air tiles, and exist to blend together types of local maps.

4.2. Front End Messages

4.2.1. Entity Actions

- 4.2.1.1. Accepts attack messages (P:1)
- 4.2.1.2. Accepts movement messages (P:1)
- 4.2.1.3. Accepts item action messages (P:1)

3.3 Non-Functional Requirements

1. Users
 - 1.1. Performance Requirements
 - 1.1.1. The system must be able to support a minimum of 100 players per server
 - 1.2. User Roles
 - 1.2.1. Players use the web app to play the game
 - 1.2.2. Administrators use the web app to moderate the game
 - 1.2.3. Game creators use the world editor to create and modify games
 - 1.2.4. Developers use the open source code base to extend and modify the game as they please
 - 1.2.5. Training
 - 1.2.5.1. Player users are able to figure out how to play the game through in-game text tutorials
 - 1.2.5.2. Game creators are able to figure out how to create and modify games through help features
 - 1.2.5.3. No support is given for developers to learn source code (besides comments)
 - 1.2.5.4. Tutorials are created to help learn the default game controls and simple game mechanics in 10 minutes
 - 1.2.6. Errors
 - 1.2.6.1. Players are protected from accidentally deleting items and characters by a confirmation dialog box.
 - 1.2.6.2. All game files created by the World Editor are checked for errors before being used to run the game
 - 1.2.7. Interfaces
 - 1.2.7.1. Players are able to play the web app using a keyboard and mouse combination
 - 1.3. System Modifications
 - 1.3.1. Distributed System
 - 1.4. Security Issues
 - 1.5. Hardware
 - 1.5.1. Hardware must have at least 1 GB of system memory, X CPU speed, and X hard disk space to run the game.
 - 1.5.2. A maximum of 200 ms latency between the server and client.

4. Glossary

A* a pathfinding algorithm that finds the most efficient path between 2 points.

Action A spell or ability a character or an item can perform.

Administrator User with ability to ban users or give access to certain players.

Algorithm a step by step procedure for calculations and data processing.

Animation rapid display of static pictures based on certain player movement and commands.

Authentication verify the users credentials on the server to give access to game and characters.

Back End any processing that takes place remotely from the players location.

Breadcrumb A navigation aid which allows users to keep track of their locations within the program.

Character A single entity in an MMORPG game world which can interact with the game world.

Character Level Measures the overall effectiveness of a character. As the character's level increases, so does the value of their statistics.

Character Statistics (Stats) Measure how effective a given character is at certain tasks.
Example: Strength, Agility, Intellect.

Class a method of differentiating game characters that have different sets of abilities and statistics.

Component A structure of data which is held inside of an entity.

Cooldown After an attack has been down, there is a time based countdown before the player can do that same attack again.

D* Lite A variant on the A* algorithm, which accounts for unknown environments.

Damage A reduction in a character's health.

Damage Type The type of damage that is being dealt to a character. Examples: fire, physical, etc.

Database organized collection of data and supports processing of information

Effect A magical component which applies a status to its target.

Entity A list of components.

Experience A value that measures a character's progress to the next Character Level.

Faction An organization within the game which NPCs may belong to.

FrontEnd any processing that takes place on the player's computer / application.

Game State The complete knowledge of everything contained within the game at a current point in time.

Game World The collection of all rooms, or zones and the characters they contain which are managed by the server(s).

Health A statistic which measures how much damage a character can sustain before the character dies.

Heuristic A function used to estimate the “quality” of a situation, used in AI programming.

Hyper Text Markup Language (HTML) The latest revision of a markup language used to organize content for the web.

Java Virtual Machine (JVM) An abstract computing machine used to execute Java bytecode.

Latency time delay experienced by a system.

Line of Sight an unbroken line from one entity to another, representing simple visibility between the two.

Mana a resource that a character can expend to use different abilities.

Massively Multiplayer Online Game (MMO) An online video game in which there is a central game world managed by one or more servers to which many players, or clients, can connect in order to interact with one another.

Melee a short range attack that is only limited to the area immediately around a character.

Non-Player Character (NPC) Non-playable characters whose actions are processed by server(s) of an MMORPG.

Player A player is a person who controls an avatar.

Player Character (PC) The representation of a player in the MMORPG game world.

Prop A purely aesthetic visual element which has no impact on game play. (Example: a bush.).

Quest A mission with one or more objectives, usually resulting in a reward and/or story advancement when all objectives are complete.

Role Playing Game (RPG) A game in which players control characters intended to represent themselves.

Room One piece of the game world. Rooms will be connected by portals which will be the only way to enter or leave a room.

Scala A JVM programming language incorporating object oriented and functional programming paradigms.

Scala Build Tool (SBT) A tool used to compile and run Scala projects.

Sprite A small image which is used to represent a game entity.

Spritesheet A file which has multiple images representing different stages of animations for a game entity.

Status Effect An effect on a character/player/enemy that increases or decreases a statistic from the normal amount.

Super User a user with access to all abilities and moderation functions of an application.

Tilesheet A list of sprites for use in building a map.

Unit Frame A UI element that displays vitals about characters and NPCs.

Web Graphics Library (WebGL) is a Javascript API for rendering interactive 3D graphics and 2D graphics within a compatible browser.