

Requirement Analysis Document

Maine Coon

1. Introduction:

- 1.1 Purpose: The purpose of this software is to meet the requirements for the overarching project in CMSC 447. The software is meant to entertain and present improving software development skills through the development of a web based game.
- 1.2 Scope: The scope of the software will include the goals, requirements, and targets to meet. The goal of this application is to improve our team software development abilities, improve knowledge of web development, and simulate the software development process. Requirements include the creation of a simple web based game which utilizes a database to keep track of scores, make api calls to both public and local, and various other documents that organize the creation of the software. The targets to meet are achieved through the use of three sprints throughout the semester. The tentative dates for sprints one through three are March 7th 2024, April 4th 2024, and May 2nd 2024. In addition to this, there is a peer evaluation and a final presentation that must be completed by the end of this development process.
- 1.3 Objectives and Criteria: Objectives have been mentioned in the section above. To extrapolate, development of this software should improve the abilities of the team in back end - front end development by actually putting a face on the software that has been developed throughout years of student work. There will be improvement in development using databases, web based programming, scripting, virtual environments, and more. Some criteria include the use of databases, api calls, graphical interface development, interactive application development, and proper teamwork.
- 1.4 References:
 - Florida State University: Provided a broad template that can be utilized to understand how a requirement analysis should be carried out
 - Add other references here:

2. Proposed Application:

- 2.1 Broad Description: The game being developed by Maine Coon is a modified version of Gravity Guy. Gravity Guy is a simple “platformer” style game where the character in the game can utilize reversing gravity to traverse an obstacle course and reach the end of a stage to proceed to the next stage.

- 2.2 Features: The game will feature a UMBC themed version of Gravity Guy, which utilizes a dog (your choice on the breed) to traverse stages of increasing difficulty. Technical features include: at least three stages, saves of the state of a user, a high score database, simple collectables in the game, and being fully web based. *may not be hosted online.

3. Requirements:

- 3.1 Functional Requirements:
 - Functionality:
 - The System shall record user information for account creation
 - The system shall employ databases to record high scores for recorded users
 - The system shall provide at least three different stages for the user to complete
 - The system shall save where the user leaves off in the game
 - The system shall make API calls to allow the web based game to work appropriately.
 - Business Requirements: Usually this would include objectives from the sponsor, market, and customer. In this case, the requirements are from the project documents. Maine Coon is responsible for achieving the objectives of working as a team to produce a web based game with at least three stages, a leaderboard using a database, and a save system that allows the user to continue off from a point they may have left off.
 - User Requirements: The user requirements can be found in the Use Case Document: https://github.com/SouthTem/cmssc447-sp2024-main_coon
 - System Requirements: System requirements include the subsystems included in the game. This includes systems to store user information, allow the user to play the game, give the user a GUI, and store high scores. Hardware that may be required is a space bar, making the game a little difficult to play on a mobile device.
- 3.2 Nonfunctional Requirements:
 - Constraints:
 - The application must be developed for web use. This means there can be no use of game development tools such as unity.
 - The application must make use of databases to store user information and high scores

- Github and Jira must be utilized to organize work
 - API calls must be made, publicly when applicable.
 - The game must have two additional features to what is required in the document of the team's choosing
- External Interfaces:
 - Although errors will be avoided, if there is an error, there will be an error signal and display an error message.
 - A graphical user interface (GUI) will be utilized to allow the user to interact and create an account, store high scores, or play the game.
- Quality:
 - Developers:
 - The game software files must be easy to interpret for all members of the group (comment your code!)
 - The software must be modifiable to easily fix errors and bugs
 - The software must be able to run on devices that can play the game (devices with a space bar)
 - The software must be able to be tested through gameplay by the developers
 - Users:
 - Users must be able to easily play the game
 - The user must be able to create an account and save their information
 - The user must be able to play the game without any workarounds or loopholes
 - The user must be able to play the game multiple times without running into bugs or errors
- Business Rules:
 - Rules that may affect the practice of the business or institute
 - All content must be original and without plagiarism
 - The use of AI is prohibited
 - All graphics must be made by the team
- 3.3 Diagrams and Models:

- Use Case Diagram:
https://github.com/SouthTem/cmsc447-sp2024-main_conon
- Activity Diagram:
https://github.com/SouthTem/cmsc447-sp2024-main_conon
- Class Diagram: https://github.com/SouthTem/cmsc447-sp2024-main_conon
- ER Diagram: https://github.com/SouthTem/cmsc447-sp2024-main_conon
- Use Case Document:
https://github.com/SouthTem/cmsc447-sp2024-main_conon