

Team Members

Will McLaughlin
WillM5462
wimc5462@colorado.edu
7204419769

Joseph Taylor
gitRot
jota2125@colorado.edu

Kyle Leonesio
kyleJL314
kyle9639@colorado.edu
720 320 4037

Samuel Britten
ssaammbb
sabr0469@colorado.edu
720-496-8169

Application name:

Software Quest

Application description:

Software Quest is a casual, 3D physics-based “rogue-like” game that plays across brief episode-like levels with a self-aware software development theme that introduces the player to elements of the design and production process.

Gameplay will be a third person, top down perspective, physics-based action-adventure where the player controls an avatar that must navigate through an environment, solving puzzles to advance and fighting enemies in real-time using items and powerups collected. Adversaries, equipment, powerups etc, will follow a software-development theme such as collecting skills like “Agile” (the agile development method), finding companions like the “3D modeler”, and overcoming obstacles like the “Clueless producer.”

Vision Statement:

For gamers, who want to have fun in between classes or assignments while picking up industry knowledge, the game “Software Quest” is a 3D “Rogue-like,” that is entertaining but also informative and inspirational.

Unlike the leading rogue-likes such as “Risk of Rain 2” that challenge a player to merely overcome an evil adversary, our product challenges the player to assemble a team of talented

developers, collect valuable skills and methodologies used in the industry, and implement them in an effective way to successfully develop a project.

Version Control and Development Method:

<https://github.com/CSCI-3308-CU-Boulder/204-5>

We will be using the agile/iterative development method, as it is important for us to get a Minimum Viable Product (MVP) out and then be able to revise and improve the product throughout the development timeline until turn-in. We need the capability to make changes during our development cycle as we discover what is fun and what we have the resources to accomplish within the timeline. In addition, we may use Unity teams to help us organize our development.

We will be implementing task-assignment using a Trello board. Link:

<https://trello.com/b/Xm0X1gIM/204-5-project>

We will be also be implementing issue-tracking and user stories with JIRA. Link:

<https://csci-3308-fa20-204-5.atlassian.net/>

Communication Plan:

Our team will communicate with shared google-docs, zoom meetings, and GroupMe, a group chat app accessible on mobile and desktop. We will use this group chat to keep in touch with one another and for basic discussions on the project. We will use googledocs and github for code sharing and other development around the game as well as Unity Teams, a teamwork app built for use with Unity. The majority of our collaborative work and discussions will be done over Zoom meetings.

Meeting Plan:

Meet with TA every Friday - 3:00pm to 3:15pm (Zoom)

Meet with group every Friday - 3:15pm to 5:15pm (Zoom)

Check GroupMe chat at least daily.

Proposed Architecture Plan:

The majority of our product will be a 3D "Rogue-like" game developed in the Unity game-engine. Unity will be responsible for assembling our components in one place, including our 3D environments and models, game code, and server/database communication.

Art for our game will be made with a variety of softwares. The bulk of the work will be done in blender (modeling, animation), but other software will be used as well.

Sculpting: Zbrush, Blender

Modeling & Animation: Blender

Materials & Shaders: Quixel, Substance
Textures, Normals, etc: Krita

Scores and wiki/game information will be stored and displayed via an interactive real-time database and/or web-page. The database will be built with MySQL and phpMyAdmin (for sql), and the webpage will be built using HTML, CSS, Javascript, and potentially React.

Use Case Diagram:

Actors:

1. Player/Gamer
2. Database Manager (manages wiki / scores)
3. Website Viewer (lookup wiki, scores)

Use Cases:

1. exploring the map
2. Looking up stats of character
3. Fighting the enemies
4. looking up scores of other players
5. creating an account
6. Finding information on enemies
7. Looking through the inventory
8. Display current health
9. Picking up items in game
10. Ability to removal of scores of from leaderboard
11. Display of current score in game
12. Equipping items/using items
13. Way to contact admins about potential problems
14. Able to change setting of sound or other aspects of the game
15. Ability to change controls of game
16. Banning accounts

