

Jimmy's Labyrinth

Developed by The Screenshots:

Rebecca Rivera, Trevor Green, Mason Ferrell, Wesley Lang, Cole Philipp, Mykayla Stout

Project Feature List

1. Player accounts, ranks, and scoreboard
2. Maze Levels
3. Full player control: Player can walk, run, jump, and suicide
4. Custom difficulty
5. Friends list
6. Player achievements
7. Unique website

Requirements

1. Player accounts, ranks and scoreboard Functional and Nonfunctional requirements
 - a. Functional Requirements-
 - i. Database that stores your player data (usernames, passwords, ranks, etc.)
 - b. Nonfunctional requirements
 - i. Using PHP to connect to a MySQL database
 - ii. Player table with username, password etc
 - iii. Player History table with past scores
2. Maze Levels
 - a. Functional Requirements:
 - i. Finish one level unlocks the next, and each level gets more difficult
 - b. Non-functional Requirements:
 - i. Tracks which levels the player has completed and with which times
 - ii. PHP scripts compare players data with what levels should be displayed and playable.
3. Full player Control:
 - a. Functional Requirements:
 - i. Players use keyboard to explore the maze
 - b. Non-Functional Requirement:
 - i. Player has x,y, z coordinates and a health/alive boolean

- ii. Key presses affect player velocities
 - iii. Jumping will enlarge the character to simulate them jumping toward the normal of the screen
- 4. Custom difficulty
 - a. Functional requirements
 - i. Harder maps will have more turns, be bigger, and possibly have a time limit
 - b. Non-functional requirements
- 5. Friends list
 - a. Functional requirements
 - i. We will use SQL to connect the database to our in-game search engine so that users wishing to look at their friends' progress can access that? I don't really know how to verbalize this
 - b. Non-functional requirements
 - i. The game will contain a feature by which players can see the achievements and progress of other players by searching for the other players' usernames
- 6. Player achievements
 - a. Functional Requirements:
 - i. As the player progresses through games levels and rank status in the database they earn specific achievements
 - b. Non-functional Requirements:
 - i. We'll have a set of achievement objects defined by requirements such as levels reached, speed of levels, etc.
 - ii. When player data updates we check for what unearned achievements they are applicable for.

Project Plan

<https://github.com/beccaarivera/Group111-2-ProjectComponents>

Using Github's "issues" feature