



Dungeon Battle

A rogue-like game



4 people

Project Description

Dungeon crawls, or more specifically rogue-likes are turn-based computer games in which a single player evolves through a procedurally generated dungeon, fighting creatures, finding various objects, acquiring experience. The final objective may be to reach the top (or bottom, if the dungeon is a cave) of the dungeon and come back, or to find a special object. Death is typically permanent, and plays can be very short, especially for beginners.

Skills

Scenario



Real Time programming



GUI



(*)[The skill scale is from 0 (Fundamental Awareness) to 6 (Expert).]

Level 1

You may now pursue to the level 1 of the project.

- * Board
The game is played on a series of dungeon levels which are simple square grids. The grid cells may be empty, walls, or floor. Empty cells must never be next to floor cells. Maps can be loaded from and save to a file.
- *** Graphics
The graphics can either be 2D or text based. There is a window where the board is displayed.
- * Controls
The player is controlled with zqsd and can move through floor cells, but should be blocked by walls.
- * Monsters
Monsters appear on the map, and move along some predefined path.



Dungeon Battle

A rogue-like game



4 people

Project Description

Dungeon crawls, or more specifically rogue-likes are turn-based computer games in which a single player evolves through a procedurally generated dungeon, fighting creatures, finding various objects, acquiring experience. The final objective may be to reach the top (or bottom, if the dungeon is a cave) of the dungeon and come back, or to find a special object. Death is typically permanent, and plays can be very short, especially for beginners.

Skills

Scenario

Real Time programming

GUI

(*)[The skill scale is from 0 (Fundamental Awareness) to 6 (Expert).]

Level 2

Level 1 must be unlocked to read this section

- ★ Developer Documentation Required for lvl 2 validation
Document your project (not necessarily only in the source code) so that a newcoming developer could understand and contribute to the code.
- ★ Release Required for lvl 2 validation
Produce a release as a source archive or git tag. The release files should have up-to-date README and INSTALL files and more generally allow anyone to deploy the application.
- ★★ Fighting
The hero can attack monsters, kill them. He can take hits and may eventually die.
- ★★ Intelligent Monsters
Some monsters will go towards the hero without getting stuck. In simple situations without other moving entities, an optimal path is used.
- ★ Test Pathfinding
Test pathfinding on basic examples.
- ★★ Story
A simple story unfolds based on player actions (e.g. picking up a special object, interacting with some character)
- ★★ Menu
The game should launch on some special level which acts as the menu, allowing the player to select options, and start a game.
- ★★ Save and Load Games [optional]
It must be possible to save a game, and load it later after restarting the application.