

Core

- ☒ Draw Map
- ☒ Input, Player Moves in Map
- ☒ Player Collides with Walls
- ☒ Entry & Exit
- ☒ Traverse to next level
- ☒ Map Rooms
- ☒ Data logs
- ☒ Player and Enemy Attributes
- ☒ Static Enemies, participate in combat
- ☒ Enemies move when ~~in range~~ Sees the player (Raycast)
- ☒ Treasure Chests
- ☒ Traps
- ☒ ~~40~~ 5 different maps

Evaluation

- ☒ Clean, organized code.
- ☒ Naming Conventions
- ☒ Good partitioning of classes and methods. Make sure your code resides where it should.
- ☒ Good use of a struct
- ☒ Use of suitable data structures

Advanced Features

Easy

- ☒ Prerequisites: Add requirements for leaving a level (such as you must kill all the enemies on the playing field)
- ☒ Console Colors: Use colors within the console to denote different entities
- ☒ SFX: Add at least 2 sound effects
- ☒ Progression: Add a progression system, so enemies will become stronger from level to level
- ☒ Healing Potions: add healing potions that can be used inside\outside of battle.
- ☒ Smooth Refresh: make the frame refresh smooth
- ☐ Random Enemy Locations: make enemies spawn in random (and valid!) locations on the map

Medium

- ☒ HUD: create a HUD to present game data, such as player's HP, gold count, etc. Hud can replace the Data Feature.
- Options Menu: create an options menu that can set various gameplay elements:
 - ☒ Choose player's avatar
 - ☒ Choose monster's avatar
 - ☒ Difficulty level (only if chosen Progression)
 - ☒ 2 more options of your choice
- ☒ Inventory: Create an inventory system: the player can collect at least two items and use them in the game world in some way (keys, weapons, armors, etc.)
- ☒ Doors: insert doors into your game. The player can open doors with keys\levers\simple interaction. Doors provide a way to enter rooms within the level.
- ☒ File: read the maps from a file\files. Cannot be done with Procedural Levels.

Hard

- ☒ Big Enemies: Create enemies bigger than one tile
- ☐ Save and Load system
- ☒ Asymmetric Maps: Make the maps asymmetrical
- Combat System: Make an elaborate combat system that uses:
 - ☐ Damage Reduction
 - ☒ Evasion
 - ☐ Critical hits
 - ☒ Hit Chance
 - ☐ You may add more parameters to your liking
- ☒ Shop: Create shops that sell items, boosts, potions either within the level or between levels. Add currency that can be dropped from enemies or found in treasure chests.
- ☐ Real-Time: Make the game real-time (game refreshes without waiting for player input)
- ☐ Procedural Levels: Create the levels procedurally. This includes the map layouts, enemies, treasure chests, traps, etc. The game is now a roguelike, where the player can play an indefinite amount of maps but when he dies the game is over. Must choose Progression to implement this.
- ☒ Menu System: there's a Main Menu that can be used to pick a new game, see credits, change options, and Load (if chosen). Must pick the Options feature to choose this.

Custom Features

Features not listed in the brief you decided to add cause its f cool.

- ☒ Implemented a smart pathfinding algorithm for the enemies (It is Dijkstra i think)
- ☒ Explosive potions that deals damage to enemies and destroy cracked walls

- ✓ Spikes - Moving obstacle that damages the player, if he touches it
- ✓ Potions with long effects, such as invisibility, Accuracy and more
- ✓ Fog of war. Cause it was only one line of code in right place and endless level-design options
- ✓ Raycast for better Fog of War and enemies trigger