

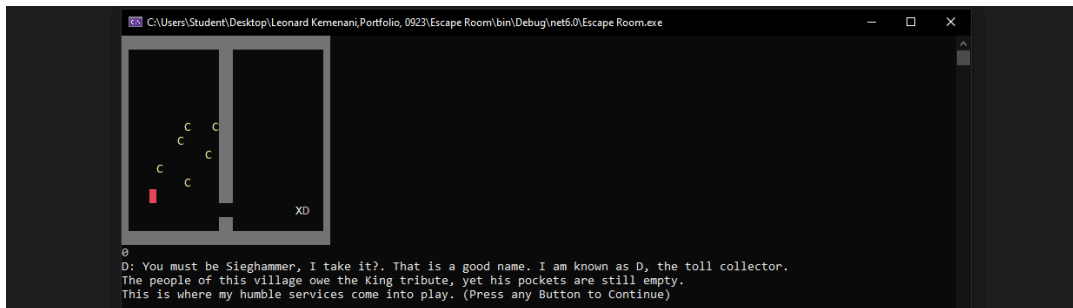
## C++ RPG GDD

### Kurzbeschreibung

Das C++ RPG ist ein Konsolenspiel in dem der Spieler die Rolle eines Ritters übernimmt und verschiedene Räume von Monstern säubern muss. Ist ein Raum gesäubert öffnet sich der nächste, bis der letzte Raum geöffnet wurde.

### Anforderungen an das Projekt und Features

Die Darstellung des Spiels soll erst einmal sehr rudimentär sein, wie mein Escape Room Projekt aus dem ersten Semester.



*Escape Room Projekt aus dem ersten Semester*

Die Kämpfe sollen erst einmal über Text dargestellt werden, wie beim Monsterkampf Simulator.

Die Kämpfe werden getriggert, wenn der Spieler Charakter auf ein Feld läuft, auf dem sich ein Monster befindet.

In dem Startgebiet befindet sich ein Bonfire, an dem der Spieler rasten kann, womit er auflevelt und seine Heiltränke auffüllt, was allerdings auch alle Gegner wieder spawnen lässt (freigeschaltete Räume bleiben freigeschaltet).

## Stats

Der Spieler, sowie die Monster verfügen RPG typisch über Werte, die über ihre Tauglichkeit in unterschiedlichen Aspekten entscheiden.

Diese sind wie folgt:

Level (LVL)

. Die Summe der verteilten Werte der Figur

Vitalität (VIT)

. Legt die maximale Lebenspunkte Anzahl der Figur fest

Stärke (STR)

. Legt fest wieviel Schaden ein Angriff der Figur verursacht

Verteidigung(DEF)

.verringert den eingehenden Schaden

Speed (Spd)

.Legt wer zu Kampfbeginn anfängt und wie viele Schritte die Figur sich außerhalb des Kampfes pro Zug bewegt.

Erfahrungspunkte (EXP)

.Der Spieler nutzt die gesammelten Erfahrungspunkte um am Bonfire aufzuleben.

.Der EXP wert der Gegner legt fest wieviele EXP der Spieler für das Besiegen des gegners erhält.

## Progression

Der Spieler startet in einer sicheren Zone in der sich nur ein Bonfire befindet (vielleicht später noch ein NPC...) der Gang zum ersten Dungeon Level ist bereits geöffnet.

Betritt der Spieler den ersten Dungeon Level findet er dort erst schwache Gegner, hat er all diese besiegt öffnet sich die Tür in den nächsten Level wo sich stärkere Gegner wiederfinden.

Geplant sind 3 Dungeon Level, hat der Spieler alle Räume gesäubert hat er das Spiel gewonnen.

## Bewegung ausserhalb von Kämpfen

Auch die Bewegung ausserhalb von Kämpfen findet Rundenbasiert statt.

Befindet sich der Spieler in einem Dungeon Level kann er sich X Schritte (definiert durch Speed) bewegen, ehe sich die Gegner X Schritte bewegen können.

Reihenfolge der Produktionsschritte:

1. Erstellen der Monstertypen
2. Erstellen des Spielers
3. Erstellen der Hub Area mittels 2D Array
4. Implementierung Player Movement
5. Implementierung Enemy Movement
6. Implementierung Combat Screen
7. Implementierung Rückkehr zum Play Screen
8. Generierung der Dungeons
9. Spawnen der Gegner
10. Respawn der Gegner