

The Quest for the Lost City

Teil 1

1. Eigene Räumlichkeiten definieren. Definieren Sie eigene Räumlichkeiten eines Gebäudes oder einer Landschaft, ganz nach Ihrer Phantasie.

- The Lost City (Story)
- The Laboratory of the Mad Scientist (Equipment sammeln. e.g. Healing/Damage Potion, Schwert, ...)
- The Maze of Roots (Informationen über den Boss sammeln)
- The Mines of Despair (Üben gegen kleinere Gegner. Die Gegner geben münzen ab, mit denen man die Items aus dem Labor aufrüsten kann, wenn der Kampf gegen den Boss zu schwierig ist)
- The Dungeon of Ice (Boss level)

2. Beschreiben Sie in ihrer Dokumentation wann Sie gewonnen haben:

Wenn man den Bösewicht im Bosslevel besiegt, hat man gewonnen!

Wahrscheinlich wird unser Character ein Energieball haben, mit dem er angreifen und sich verteidigen kann. Wenn man verliert, sieht man der Begriff "You died" auf dem Bildschirm und man wird an den anfang gebracht (Es folgt den regeln eines Roug like game wie [Hades](#)). Wenn man gewonnen hat kommt noch eine Statue wo man eine Belohnung bekommt.

3. Beschreiben Sie in ihrer Dokumentation wie Sie es realisieren möchten:

Man geht in den vorherigen Raum zurück.

4. Im Raum kann ein Gegenstand sein.

- a) Beschreiben Sie in ihrer Dokumentation wie Sie vorgehen:
 - Mann kann Items in Truhen oder in Vasen finden. Wenn ein Item aufgehoben wird dieses in ein Array List hinzugefügt werden. Diese haben Attribute welche dann in der Mine geupgraded werden können. Ausserdem können Items wieder abgelegt werden.

- b) Wir müssen auch nach dem Gegenstand im Raum "fragen" können/Im Raum können mehrere Gegenstände sein.
 - Die Items eines Raumes werden in einer ArrayList abgespeichert. Falls diese aufgehoben werden, werden diese aus dem ArrayList des Raumes entfernt und dem des Spielers hinzugefügt

5. Gegenstand hat ein Gewicht:

Im Objekt des ArrayLists wird auch das Gewicht eines Items definiert. Wenn die Maximalgewicht erreicht wird kann kein weiteres Item aufgehoben werden.

6. Vererbung

Wir haben noch nicht wirklich Vererbung im Programm geplant werden dies jedoch falls dies sinn macht implementieren.

7. Techstack

- [Asprite](#) (Assets-Designen)
- [BFXR](#) (Sounds)
- [Itch.io](#) (Assets-Store)
- [Liteengine](#) (Engine)

Teil 2

1. Konstruktor der Klasse Game

```
Game zorkgame = new Game();
zorkgame.play();
```

2. Konstruktor der Klasse Raum

```
public Room(String description)
```

3. Konstruktor der Klasse Command

```
public Command (String commandWord)
public Command (String commandWord, String secondWord)
```

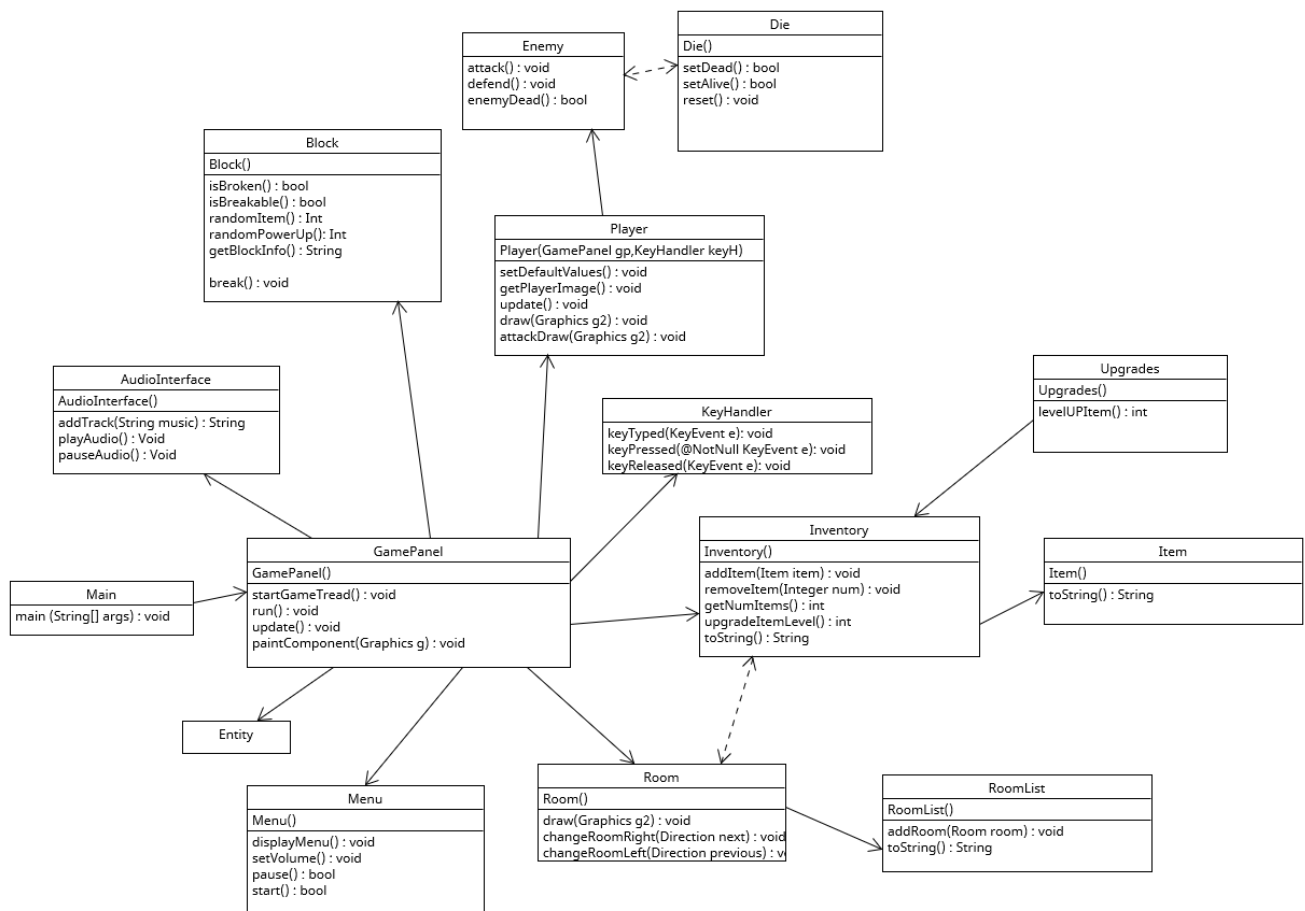
4. Ausgänge eines Raumes

```
private HashMap<String, Room> exits;
this.description = description;
this.exits = new HashMap<>();
```

5. Hier drin ist es aber Kalt!

```
office = new Room("the computing admin office, Das Fenster ist
offen brrrrr.....");
```

Klassendiagramm



Skizze der Räume

