

Funktionsweise des von mir verwendeten Algorithmus zur Generierung zufälliger Dungeons

Tobias Nöthlich

30.08.2017

Zur Aufgabenstellung dieser Code-Competition gehörte es, Dungeons, die die Spielfläche bilden, zufällig zu generieren. Um dieses große Ziel zu erreichen, müssen zuerst einmal die kleineren Ziele des Generators formuliert werden, um die genaue Funktionsweise Schritt für Schritt aufzubauen. Die Ziele lassen sich also wie folgt beschreiben:

- Eine beliebig große Gruppe von miteinander verbundenen Räumen und Korridoren.
- **Alle Felder müssen erreichbar sein.**
- Es gibt sowohl einen Start- als auch einen Endpunkt, welche zufällig im Dungeon platziert werden.

Bei meinen Recherchen zu diesem Thema bin ich auf das OpenSource Rogue-like “Tyrant” von Mike Anderson gestoßen, welcher seinen Algorithmus zum Entwerfen von Dungeons näher schildert:

1. Fülle die ganze Map
2. Erschaffe einen einzelnen Raum in der Mitte der Map
3. Wähle eine Wand eines Raumes aus
4. Wähle was dort gebaut werden soll (in meinem Fall Korridor oder Raum)
5. Schau ob genug Platz vorhanden ist um das gewählte Feature zu platzieren
6. Wenn ja, fahre fort, wenn nicht gehe zurück zu Schritt 3
7. Füge das Feature durch die Wand hinzu
8. Zurück zu Schritt 3, bis der Dungeon fertig ist
9. Füge Ein- und Ausgang hinzu

10. Füge Items und Monster hinzu

Der Schritt, der dabei dafür sorgt, dass alle Felder erreichbar sind ist Schritt 3, da dort die Regel aufgestellt wird nur am schon existierenden Dungeon anzubauen. Dies führt dazu, dass es von jedem Raum mindestens einen Weg in jeden anderen Raum gibt, da nichts von Dungeon abgetrennt generiert wird.

In dem von mir verwendeten Algorithmus werden diese Schritte im Großen und Ganzen so umgesetzt, ich habe ihn jedoch so modifiziert, dass

- Feinde nicht mehr bei der Generierung platziert werden, sondern erst danach
- Dekorationen zufällig über den ganzen Dungeon verteilt werden
- es zwischen doppelten Wänden alle 3 Tiles zu Durchbrüchen kommt, um weitere Wege zu schaffen
- der Dungeon die Anzahl der verschiedenen Zellarten (Wand, Dekoration, Boden,...) zurückgeben kann

In meinem Projekt findet man den Code zur Generierung in der “dungeon_generator”-Klasse.