

Entwicklung eines dreidimensionalen zyklischen Dungeon-Generators mit Unity3D

Bachelorarbeit

Nils Gawlik
Matrikel-Nummer 553449

Betreuer Prof. Lenz
Erstprüfer Prof. Lenz
Zweitprüfer Prof. Strippgen

Inhaltsverzeichnis

Abbildungsverzeichnis	1
1 Einführung in das Thema	2
1.1 Dungeons	2
1.1.1 Definition von Dungeon	2
1.1.2 Wichtige Beispiele von Dungeons	2
1.1.3 Prozedurale Dungeon-Generierung in Spielen, Geschichte	3
1.2 Genaue Beschreibung der Fragestellung	3
1.2.1 Definition von zyklisch	4
1.2.2 Definition von Dungeon	4
1.2.3 Dreidimensionalität	4
1.3 Warum Unity3D?	4
1.3.1 Entity/Component	4
1.3.2 Editor Scripting	4
2 Erklärung des Ersetzungs-Algorithmus	5
2.1 Replacement Grammars	5
2.2 Allgemeiner Algorithmus	5
2.3 Spezieller Algorithmus	5
3 Implementierung	6
3.1 Tool-Charakter der Implementierung	6
3.2 Wichtige Klassen	6
3.3 Die Pattern-Datenstruktur	6
3.4 Lazy updating, suchen	6

Inhaltsverzeichnis

4	Auswertung der generierten Dungeons	7
4.1	Einschränkungen	7
4.2	qualitative Auswertung	7
4.3	quantitative Auswertung	7
4.4	Mögliche Anwendungsszenarien	7
4.5	Roguelike, Rogue-lite	7
4.5.1	Minigolf	7
4.5.2	Text adventure	7
5	Zusammenfassung	8
	Literaturverzeichnis	10
	Bildquellenverzeichnis	11

Abbildungsverzeichnis

1.1	Dungeon-Beispiel	3
3.1	Test-Bild	6

1 Einführung in das Thema

TODO: Einleitung

Im Rahmen dieser Arbeit soll ein dreidimensionaler zyklischer Dungeon-Generator konzipiert und implementiert werden.

1.1 Dungeons

1.1.1 Definition von Dungeon

Der englische Begriff „Dungeon“ wurde für diese Arbeit bewusst gewählt, er hat im Bereich Spiele eine spezielle Bedeutung, die sich von der wörtlichen Übersetzung „Verlies, Kerker“ unterscheidet. TODO: Wörterbuch, Quelle für Satz Der Begriff geht auf frühe

1.1.2 Wichtige Beispiele von Dungeons

TODO: Zelda

1 Einführung in das Thema

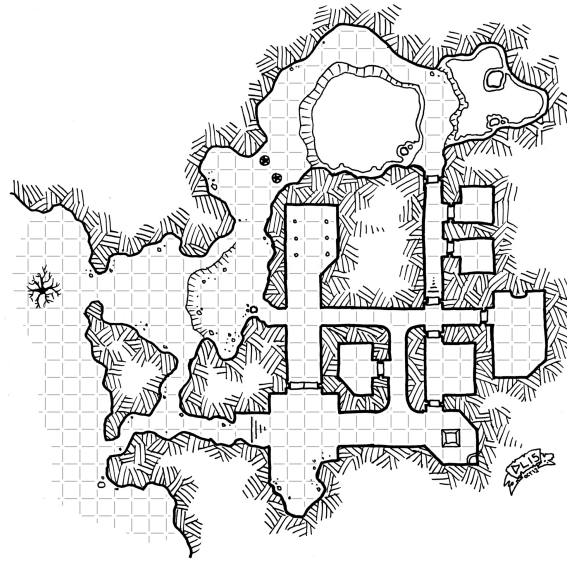


Abbildung 1.1: Beispiel eines Dungeons [Log15]

1.1.3 Prozedurale Dungeon-Generierung in Spielen, Geschichte

1.2 Genaue Beschreibung der Fragestellung

Der Inhalt der Arbeit ist die Entwicklung eines dreidimensionalen zyklischen Dungeon-Generators mit Unity3D.

Im Folgenden soll erläutert werden, wie die Einschränkungen „dreidimensional“ und „zyklisch“ im Kontext dieser Arbeit zu verstehen sind.

1 Einführung in das Thema

1.2.1 Definition von zyklisch

1.2.2 Definition von Dungeon

1.2.3 Dreidimensionalität

1.3 Warum Unity3D?

1.3.1 Entity/Component

1.3.2 Editor Scripting

2 Erklärung des Ersetzungs-Algorithmus

2.1 Replacement Grammars

2.2 Allgemeiner Algorithmus

2.3 Spezieller Algorithmus

3 Implementierung

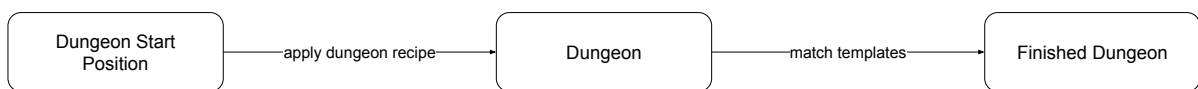


Abbildung 3.1: Test-Bild mit langer Bildunterschrift

3.1 Tool-Charakter der Implementierung

3.2 Wichtige Klassen

TODO: Tile, Pattern, ReplacementEngine, ...

3.3 Die Pattern-Datenstruktur

3.4 Lazy updating, suchen

4 Auswertung der generierten Dungeons

4.1 Einschränkungen

TODO: Subjektivität, Spiele-spezifisch

4.2 qualitative Auswertung

4.3 quantitative Auswertung

4.4 Mögliche Anwendungsszenarien

4.5 Roguelike, Rogue-lite

4.5.1 Minigolf

4.5.2 Text adventure

5 Zusammenfassung

Verwendete Materialien

L^AT_EX-Vorlage von Martin Bretschneider [[Bre06](#)]

Literaturverzeichnis

- [Bre06] Bretschneider, Martin: *Bachelorarbeit, Masterarbeit und Dissertation mit einer LaTeX-Vorlage schreiben*. <http://www.bretschneider.net.de/tips/thesislatex.html>, 2006. –
letzter Zugriff: 15.10.2018

Bildquellenverzeichnis

[Log15] Logos, Dyson: *Map of the fictional Warrek's Nest dungeon*. https://commons.wikimedia.org/wiki/File:Warrek%E2%80%99s_Nest.jpg, 2015. – letzter Zugriff: 25.12.2018

Erklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegt wurde.

Ort, Datum

Unterschrift