

### Prozedurale Generierung von Template- basierten Graphen

Procedural Content Generation in Spielen bezieht sich auf die automatisierte Erstellung von Spieleinhalten mittels Algorithmen, normalerweise mithilfe von (Pseudo-) Zufallszahlengeneratoren (RNG). Zufall ist ein Kernfaktor von PCG, um sicherzustellen, dass aus einigen ausgewählten Variablen viele diverse Inhalte generiert werden können. Prozedurale Generierung von Inhalten ist jedoch nicht nur für die Spielentwicklung von Nutzen, sondern kann auch für andere Arten von Inhalten verwendet werden, von Texturen über Meshes zu Planungsbäumen für künstliche Intelligenz.

Dieses Projekt hat zum Ziel, eine C# Softwarebibliothek zu erstellen, welche prozedural jede Art von Inhalten erstellen kann, basierend auf Template Definitionen. Die Template Definitionen beinhalten die Struktur der zu generierenden Daten und die Generierungsmethoden.

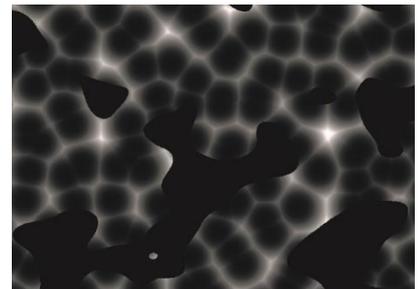
Um das Ziel zu erreichen, musste als erstes ein tiefes Verständnis für prozedurale Generierungstechniken entwickelt werden. Hierfür mussten zuerst Fraktale, L-Systeme und Algorithmen für die Generierung von Karten und Planungsalgorithmen studiert werden. Als nächstes wurde eine Software entworfen und gebaut, welche prozedurale Generierung von Inhalten implementiert und es einem Benutzer erlaubt, jede Variable des Generationsprozesses und jede Generierungstechnik konfigurieren zu können. Die Konfiguration wurde in XML-Files vorgenommen und die Software so gebaut, dass sie während der Laufzeit mit verschiedenen File-Loader und Generierungstechniken erweiterbar ist. Nachdem die Implementation der Softwarebibliothek beendet wurde, sind verschiedene Tests mit leicht unterschiedlichen Definitionen durchgeführt worden, um die Leistungsfähigkeit der Software zu testen.

Die Tests haben gezeigt, dass die Software ihren Zweck erfüllt. Zukünftige Anpassungen erlauben eine Steigerung der Geschwindigkeit und der Speichereffizienz, zu diesem Zweck kann zum Beispiel Multithreading implementiert werden. Die Software kann auch mit weiteren Techniken oder einem neuen genetischen Generierungs-Layer erweitert werden.

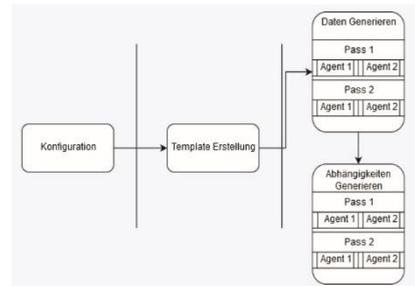


Diplomand  
Daniel Fabian

Dozent  
Gernot Kurt Boiger



Querschnitt eines  
Generierungsergebnisses



Generierungsprozess im Überblick