

HOCHSCHULE LUZERN HSLU

STUDIENGANG DIGITAL IDEATION, BACHELOR

5. SEMESTER

EXPOSÉ 2

Der Zufall und die Artistische Vision

Verfasser Simon Hischier

Lehrkräfte Manuela Hummel, Florian Krautkrämer

November 19, 2018

1 Problemstellung

Der Zufall in Spielen began schon bei den Arcade-Spielen, damals aus dem Grund, dass die Systeme noch stark beschränkte Speicherkapazität hatten. Die Spiele mussten einfach verständlich sein, spass machen und eine stets höher werdenden Schwierigkeitsgrad aufweisen, was schnell in Frust enden konnte. Als Konsolen Einzug ins Wohnzimmer fanden, wurde nach besseren Geschichten, komplexeren Spielen und neuen Spielgenres verlangt. Der Zufall und [Procedural Content Generation, Prozedurale Inhaltsgeneration \(PCG\)](#) wurden dabei zunehmend unbeliebt. Stetig neue Spiele, billigerer Speicher und neue Konsolen dominierten die Verkäufe. Zufall und [PCG](#) wird seither noch immer mit simplen Geschichten und Spielen die stark auf Reaktionsfähigkeit und können setzen in Verbindung gebracht. Teilweise hat [PCG](#) bis heute noch einen

schlechten Ruf. Mittlerweile gibt es Spiele wie *Binding of Isaac* (Edmund McMillen, Florian Himsl, 2011), *Darkest Dungeon* (Red Hook Studios, 2016), *Borderlands* (2K Games, 2009), *Rogue Legacy* (Cellar Door Games, 2013), *Spelunky* (Derek Yu, Mossmouth, LLC, 2008), *Path of Exile* (Grinding Gear Games, 2013), *Reigns* (Devolver Digital, 2016) usw. die den Zufall geschickt teils prominenter teils weniger prominent ins Spiel einbauen und den Zufall wieder Salonfähig machen. Spielern ist nicht immer bewusst, wie viel [PCG](#) im Spiel steckt. Gleichzeitig wird [PCG](#) aber bei Spieleentwickler immer noch als Einschränkung für Designer betrachtet. Zufall gilt immer noch als Beschneidung der artistische Vision. Spiele die Zufall enthalten werden oft mit schwacher Story assoziiert und mit Fokus des Spiels auf das Gameplay verbunden.

2 Zielsetzung und Erkenntnisinteresse

Die Arbeit soll im wesentlichen Möglichkeiten aufzeigen, wie Spiele mit dem Zufall umgehen und wie ein Spiel trotz Zufall gut designed sein kann. Es soll zeigen, dass der Zufall nicht als Ersatz für Designer gedacht ist, sondern eine Erweiterung des Designer-repertoires darstellt. Gezeigt wird, dass es auch mit Zufall möglich ist Geschichten zu erzählen und bleibende Erinnerungen gestaltet werden können. Die Arbeit widmet sich dem erkennen, wo Schwierigkeiten beim be-

nutzen des Zufalls auftauchen und wo der Zufall geschickt in Spielen eingesetzt werden kann. Konkret “Wie kann beim designen eines Spiels der Zufall und die artistische Vision koexistieren?”

Es soll die Möglichkeiten zeigen, wie man Zufall als Werkzeug benutzen kann. Aufzeigen, wo ein Einsatz von Zufall einen Mehrwert generiert und was damit bewirkt werden kann.

3 Ziel der Arbeit

- Analyse, welche Reize der Zufall in Spielen bieten kann.
- Aufzeigen von Schwierigkeiten bei der Nutzung von Zufall.

4 Forschungsstand und theoretische Grundlage

Es wurden bereits Unterteilungen definiert zum kategorisieren der Spiele anhand der Ausprägung der Zufallsgeneration[1].

- **Keine Generation** Ein mögliches Beispiel wäre *Super Mario Bros.* (Nintendo, 1985) bei dem alles von Hand platziert, gezeichnet und animiert wurde. Herr Miyamoto erläutert dabei Details, auf die beim erstellen des Level 1-1 geachtet wurden.[2].
- **Inhaltsgenerierung während der Spielerstellung** *The Elder Scrolls Oblivion* (Bethesda Softworks, 2006) benutzte PCG um die Welt zu erstellen und die Designer kurierten und änderten erst nach der generierung die Welt[3]. Ein Beispiel für eine weitverbreitete PCG Algorithmus Middleware für Spielestudios ist SpeedTree[4].
- **Gameplay (teilweise) definiert oder beeinflusst** durch PCG wie zum Beispiel die Nebenaufträge in *The Elder Scrolls Skyrim* (Bethesda Softworks, 2011) welche immer wieder neu generiert wurden[5]. Auch die Schlösser in *Rogue Legacy* (Cellar Door Games, 2013) werden jeweils neu generiert, dabei bleibt gibt es jedoch eine Art kontinuierität die beim erneuten Spielen vorhanden bleibt[6].
- **(Fast) vollständig generierte Spiele.** *Dwarf Fortress* (Tarn Adams, 2006) hört nicht beim Levelgenerieren auf. Es startet beim erstellen einer kompletten Hintergrundgeschichte der Welt und allem was seit begin der Welt passiert ist[7].

Komplexe Charaktere, ausgeklügelte Level und komplexe Stories werden mit starker “artistic vision” verbunden. Um diese Eigenschaften steuern zu können braucht der Artist detaillierte Kontrolle über alle Teile der Zufallsgenerierung. Van der Linden vermutet, dass intuitive und zugängliche Werkzeuge für Artisten dabei helfen könnte, die Zufallsgeneration wieder vermehrt in Spielen ein zu bauen[1].

Algorithmen, die dem Designer mehr kontrolle überlassen und trotzdem intuitiv bleiben gibt es schon und erste Spiele sind auf dem Markt erschienen. Als Beispiel kann hier “Bad North” genannt werden das den Algorithmus “Wave Function Collapse”[8][9] geschickt ins Spiel integriert und der Author Oskar Stålberg kann dabei teile der Level genau vorgeben und den rest nach seinen Regeln zufällig ausfüllen lassen[10].

5 Methodik

- Literaturrecherche
- Analyse diverser Spiele
- Zusammenfassung von Entwicklerberichten

6 Gliederung

- Einleitung
- Definitionen
 - Zufall / PCG
 - artistische vision
- Vorteile und Gefahren beim Einsatz von PCG
- Beispiel Zufall als Werkzeug

- Diskussion & Zukünftige Arbeit
- Schlussfolgerung

References

- [1] Roland Van Der Linden, Ricardo Lopes, and Rafael Bidarra. Procedural generation of dungeons. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(1):78–89, mar 2014.
- [2] Eurogamer. Miyamoto on World 1-1, 2015.
- [3] Bethesda Game Studios. The Elder Scrolls 5: Skyrim, 2011.
- [4] SpeedTree. SpeedTree Vegetation Modeling.
- [5] Matt Bertz. The Technology Behind The Elder Scrolls V: Skyrim, 2011.
- [6] Rich Stanton. The making of Rogue Legacy • Eurogamer.net, 2013.
- [7] Alex J. Champandard. Beyond Terrain Generation: Bringing Worlds in DWARF FORTRESS to Life — AiGameDev.com, 2012.
- [8] A.A. Efros and T.K. Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 1033–1038 vol.2. IEEE, 1999.
- [9] Isaac Karth and Adam M. Smith. WaveFunctionCollapse is constraint solving in the wild. In *Proceedings of the International Conference on the Foundations of Digital Games - FDG '17*, pages 1–10, New York, New York, USA, 2017. ACM Press.
- [10] Oskar Stalberg. (3) EPC2018 - Oskar Stalberg - Wave Function Collapse in Bad North - YouTube, 2018.