

HOCHSCHULE LUZERN HSLU

STUDIENGANG DIGITAL IDEATION, BACHELOR

5. SEMESTER

EXPOSÉ

Game Level generieren mit machine learning

Verfasser Simon Hischier

Lehrkräfte Manuela Hummel, Florian Krautkrämer

October 15, 2018

1 Problemstellung

Machine learning wird bei der Bildverarbeitung rege genutzt und findet in immer mehr Gebieten Anwendung. Obwohl Games sonst meist eine Vorreiterrolle für neue Technologie und Ideen einnehmen, halten sie sich bisher beim Einsatz von machine learning zurück. Eine mögliche Erklärung ist etwa, dass machine learning für Spiele ein Problem darstellt: Grössere Neuronale Netze benötigen viel Rechenzeit und Leistung. Da Spiele die selben Ressourcen auch benötigen, würde ein Neuronales Netz einen Teil der Leistung für sich beanspruchen. Des weiteren kann bei Prozeduralen Algorithmen das Verhalten sehr genau definiert werden. Fehler

lassen sich reproduzieren und damit einfacher reparieren. Neuronale Netze benötigen viele Beispieldaten damit sie lernen können. Diese Daten müssen erst generiert werden, aber wie? Wenn ein Netz dann trainiert ist kann man sein Verhalten nicht wie bei Prozeduralen Algorithmen über Werte ändern. Es muss es mit neuen Beispieldaten umlernen. Noch komplexer wird es bei der Fehlersuche: Wenn ein Netz das Verhalten eines Benutzers berücksichtigt, können Auftretende Fehler nicht so einfach reproduziert werden. Das selbe mehrere male erneut generieren lassen geht nur, wenn sich das Netz nicht ändert.

2 Zielsetzung und Erkenntnisinteresse

Ich möchte herausfinden, ob ein kleines Neuronales Netz Sinnvolle Levels für Spiele in Echtzeit generieren kann. Unter Sinnvoll gelten Level die passierbar sind und die nicht nach komplettem chaos aussehen.

Die Level-Generierung wird traditionell von Hand oder mit Prozeduralen Algorithmen gelöst, wobei prozedurale Methoden den Wiederspielwert erhöhen sollen. Bei Prozeduralen Algorithmen muss der exakte Levelaufbau wie es die Schöpferin oder der Schöpfer will weichen, weil man als Entwickler/in nur noch die Generierungsregeln aufstellt und nicht jedes Element selbst platziert. Selbst die Einflussmöglichkeit per Regeln wird bei Neuronalen Netzen dem Com-

puter überlassen. Dies erschwert die Einflussnahme bei der Level-generierung deutlich. Wie kann einem Neuronalen Netz beigebracht werden, was interessante Level sind und wie diese generiert werden sollen.

Prozedurale Generierung kann sich nur bedingt der Spielenden Person anpassen, da die Algorithmen nur auf vorher von dem/der Entwickler/in festgelegte Parameter in vorgeschriebener weise reagieren kann. Neuronale Netze haben das Potenzial, das Verhalten der Spieler selbstständig zu lernen und individuell darauf zu reagieren. Es kann gestaltungsregeln für jeden Typ spieler individuell generieren und so einmalige Spielerlebnisse gestalten.

3 Ziel der Arbeit

- Ein Computerspiel erstellen, das mit hilfe eines Neuronalen Netzwerks Level generiert.
- Prüfen, ob ein kleines Neuronales Netz mit geringer Leistungsaufnahme für die Levelgestaltung ausreicht.
- Eine Methodik finden, wie einem Netzwerk das Levelgenerieren beigebracht werden kann.
- Aufzeigen von Möglichkeiten zur Einflussnahme der Levelgenerierung.

4 Forschungsstand und theoretische Grundlage

Smith[1] et al. versuchen messbare Attribute von Leveln zu finden und definieren eine Methode um Level miteinander zu vergleichen. Sie nutzen dies um zu messen wie unterschiedlich die Level ihres Prozeduralen Level-Generators sind. Dies könnte nützlich sein um Schwachstellen oder Lücken im ML Modell auf zu decken.

Smith[2] et al. haben ein Spiel namens "Rathenn" erstellt. Dies wird zur Laufzeit generiert und der Spieler kann auf den Generator direkt Einfluss nehmen und den Levelbau beeinflussen. Dafür haben sie Level-Elemente mit Gegensätzen definiert und der Spieler kann dann durch 2 Aktionen Einfluss nehmen. Der Spieler sammelt beim Durchlauf farbige Münzen, die jeweils für ein Levelattribut stehen. Am Ende des Levels stehen Leitern, diese

werden dann anhand der meist gesammelten Münzen eingefärbt. Der Spieler kann sich nun für eine Leiter entscheiden und in den nächsten Leveln werden dann die Attribute mehr verwendet, für die die Farbe steht.

Jennings-Teats[3] et al. beschreiben ein System zur Einschätzung und Generierung von Levels in diversen Schwierigkeitsstufen. Ihr Spiel "Polymorph" passt zur Laufzeit die Generierung dem/der Spieler/in an damit das Spiel fordernd bleibt und nicht zu frustrierend wird. Sie beschreiben dabei, wie verschiedene Level-Elemente erst in Kombination auf ihre Schwierigkeit bewertet werden können. Ein Neuronales Netz soll dabei helfen, die Level-Komposition anzupassen und nicht etwa wie bei bisherigen Prozeduralen Algorithmen nur die Attribute ändern.

5 Methodik

- Unity Projekt aufsetzen und geeignete Datenstruktur definieren.
- Messbare Level-Elemente definieren.
- Geeignete Struktur für ein Neuronales Netz evaluieren.
- Prototyp erstellen zum Trainieren eines Neuronalen Netzes.
- Level generieren lassen und diese analysieren.

6 Gliederung

- Einleitung
- Definitionen
 - Neuronales Netz
 - Machine Learning
- Kriterien und Vergleichsmetriken
 - Einflussfaktoren
 - Machine Learning
- Aufbau des Netzes
- Diskussion & Zukünftige Arbeit
- Schlussfolgerung

References

- [1] Gillian Smith and Jim Whitehead. Analyzing the expressive range of a level generator. In *Proceedings of the 2010 Workshop on Procedural Content Generation in Games - PCGames '10*, pages 1–7, New York, New York, USA, 2010. ACM Press.
- [2] Gillian Smith, Elaine Gan, Alexei Othenin-Girard, and Jim Whitehead. PCG-based game design. In *Proceedings of the 2nd International Workshop on Procedural Content Generation in Games - PCGames '11*, pages 1–4, New York, New York, USA, 2011. ACM Press.
- [3] Martin Jennings-Teats, Gillian Smith, and Noah Wardrip-Fruin. Polymorph: A Model for Dynamic Level Generation. *Sixth Artificial Intelligence and Interactive Digital Entertainment Conference*, oct 2010.