



Project Report: BamBirds 2019

Creating an Intelligent Game Playing Agent for Angry Birds

Samet	Patrick
Akcabay	Haller

August 28, 2019

Contents

1	Aufgabenstellung des Smart Environment Projekts	3
2	GAN als möglicher Levelgenerator	3
3	GAN im Vergleich zu Alternativen	4
4	Vorgehensweise	4
4.1	Erkennen von Konturen der Zentroide	4
4.2	Parsen von und in XML bzw. JSON	4
4.3	Automatisierung der Abläufe mithilfe einer GUI und Powershell unter Windows	4
4.4	Konvertierung von Pythons Pillow Koordinaten in ein kartesisches XML-Koordinatensystem	4
4.5	Verdeutlichen der Konturen der erzeugten RAW-Bilder zur besseren Erkennung vor Training	4
4.6	Aufsetzen eines Re-Evaluierungssystems	4
5	Ergebnisse	4
6	Fazit	4
7	Ausblick	4

1 Aufgabenstellung des Smart Environment Projekts

AngryBirds, ein in 2009 von Rovio Entertainment entwickeltes Casual-Puzzle-Videospiel, ist ideal als Maßstab für künstliche Intelligenz geeignet. Der Weg zum Erfolg setzt voraus, dass ein Verständnis der physikalischen Zusammenhänge auf das Spiel übertragen werden können und strategische Entscheidungen getroffen werden müssen. Ziel des Projekts war es, die Schwächen des BamBirds-Agenten zu verbessern und um Funktionen zu erweitern, um mit diesem an der Angry Birds AI Competition teilnehmen zu können und diesen zu gewinnen.

Parallel zu diesem Wettkampf, lief die sogenannte AIBirds COG 2019 Level Generation Competition, in welchem das Ziel daraus bestand, mit einem selbsterstellten Programm der Wahl, automatisiert Level zu generieren, welche viel Spaß und Herausforderung anbieten sollen. Dieser Aufgabe haben wir uns gewidmet.

2 GAN als möglicher Levelgenerator

Generative Adversarial Networks, kurz GAN (zu deutsch "erzeugende gegnerische Netzwerke"), stellen in der Informatik eine Gruppe von Algorithmen zu unüberwachtem Lernen dar. Sie bestehen aus zwei Neuronalen Netzwerken, eines erstellt Kandidaten (**Generator**), das zweite bewertet diese (**Diskriminator**).

Der Generator lernt, Ergebnisse nach einer bestimmten Verteilung zu erzeugen. Der Diskriminator hingegen lernt, die Ergebnisse des Generators gegen die echte, vorgegebene Verteilung zu evaluieren (hier: konkret spielbare Level). Findet der Diskriminator keine Unterschiede mehr im direkten Vergleich der vorgegebenen Verteilung, so wird das Ziel erreicht.

Neuronale Netzwerke kommen häufig zur Visualisierung verschiedener Gegenstände, zur Erstellung von 2D- bzw. 3D-Modellen oder zur Bildbearbeitung (astronomischer Bilder) zum Einsatz.

Aus diesem Grund widmeten wir uns den GAN als Kandidat eines Level-Generators.

3 GAN im Vergleich zu Alternativen

4 Vorgehensweise

4.1 Erkennen von Konturen der Zentroide

4.2 Parsen von und in XML bzw. JSON

4.3 Automatisierung der Abläufe mithilfe einer GUI und Powershell unter Windows

4.4 Konvertierung von Pythons Pillow Koordinaten in ein kartesisches XML-Koordinatensystem

4.5 Verdeutlichen der Konturen der erzeugten RAW-Bilder zur besseren Erkennung vor Training

4.6 Aufsetzen eines Re-Evaluierungssystems

5 Ergebnisse

6 Fazit

7 Ausblick