Creating Zelda Dungeons with a Graph Grammar and a Generative Adversarial Network

Jake Gutierrez

Introduction

A Graph Grammar defines rules for random graph generation. A Generative Adversarial Network (GAN) imitates a certain style based on the input, but produces new output. Combing both can generate dungeons similar to those in The Legend of Zelda, by defining the high-level structure with a graph, and the low-level rooms with the GAN.

Features of Zelda Games

- Progress through dungeons
- Fight enemies
- Pick-up keys
- Solve a puzzle
- Get trapped
- Defeat the boss
- Get the Triforce

Graph Grammar

Using a Graph Grammar [1], we can create a high-level blueprint for a dungeon mission, but allow the grammar to fill in a variety of different details to create distinct dungeons by finding rules and replacing them with smaller graphs to make a larger graph.

Generating Rooms

We use a Generative Adversarial Network [2] to generate the rooms. GANs allow us to mimic the training data -- rooms from the original game -- but still create new rooms distinct from the originals.

Conclusion

Generating dungeons with a graph grammar is effective in creating Zelda-like dungeons. Generating rooms with a GAN creates rooms like the original rooms. In the future, we hope to conduct a human subject study to see whether or not players prefer the original or generated dungeons. AI could also be used to create new graph grammars in the future to get more interesting and challenging dungeons.

References