Introduction

Generative Adversarial Networks (GANs) are a way of training a computer to generate fake samples based on real examples. GANs are used to generate Mega Man levels that are similar to but distinct from levels in the original game. Due to the snaking pattern of levels, several GANs were used in order to handle different types of level segments, resulting in levels whose design is more human-like and better organized.

Mega Man

Mega Man is a side-scrolling game that requires the player to fight enemies and jump carefully to reach the end of each level.

Generative Adversarial Networks

- Used for generating novel results with the same style as the training data.
- Previous work [1] supports the idea that GANs can generate interesting video game level.
- GANs were used to generate new levels in Mega Man.

Evolutionary Algorithms

- Data was taken from the VGLC [2] and used to train the GAN(s).
- Evolving numeric latent vectors and convert into levels to input into the GAN.
- 10 segments per level.
- Segments placed vertically or horizontally adjacent to form snaking pattern.
- Multiple fitness functions for evolution:
  - Solution path length (determined using A*)
  - Connectivity (percentage of reachable locations in level)

Methods for Level Generation

- One GAN
  - All of the data in one training set
- Seven GANs
  - Each GAN is assigned one segment type (up, down, horizontal, and four corners)

Results

- A* path lengths are significantly longer with SevenGAN than OneGAN.
- Longer A* paths utilize more space in the levels, making them longer and harder to beat.
- SevenGAN levels look more natural.
  - Each GAN has an individual set of data and does not communicate with the other GANs to make perfectly smooth transitions from one segment to another.
- OneGAN levels look more chaotic.
  - This randomness is due to having all data in one GAN.

References