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

Jake Gutierrez

SOUTHWESTERN UNIVERSITY

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

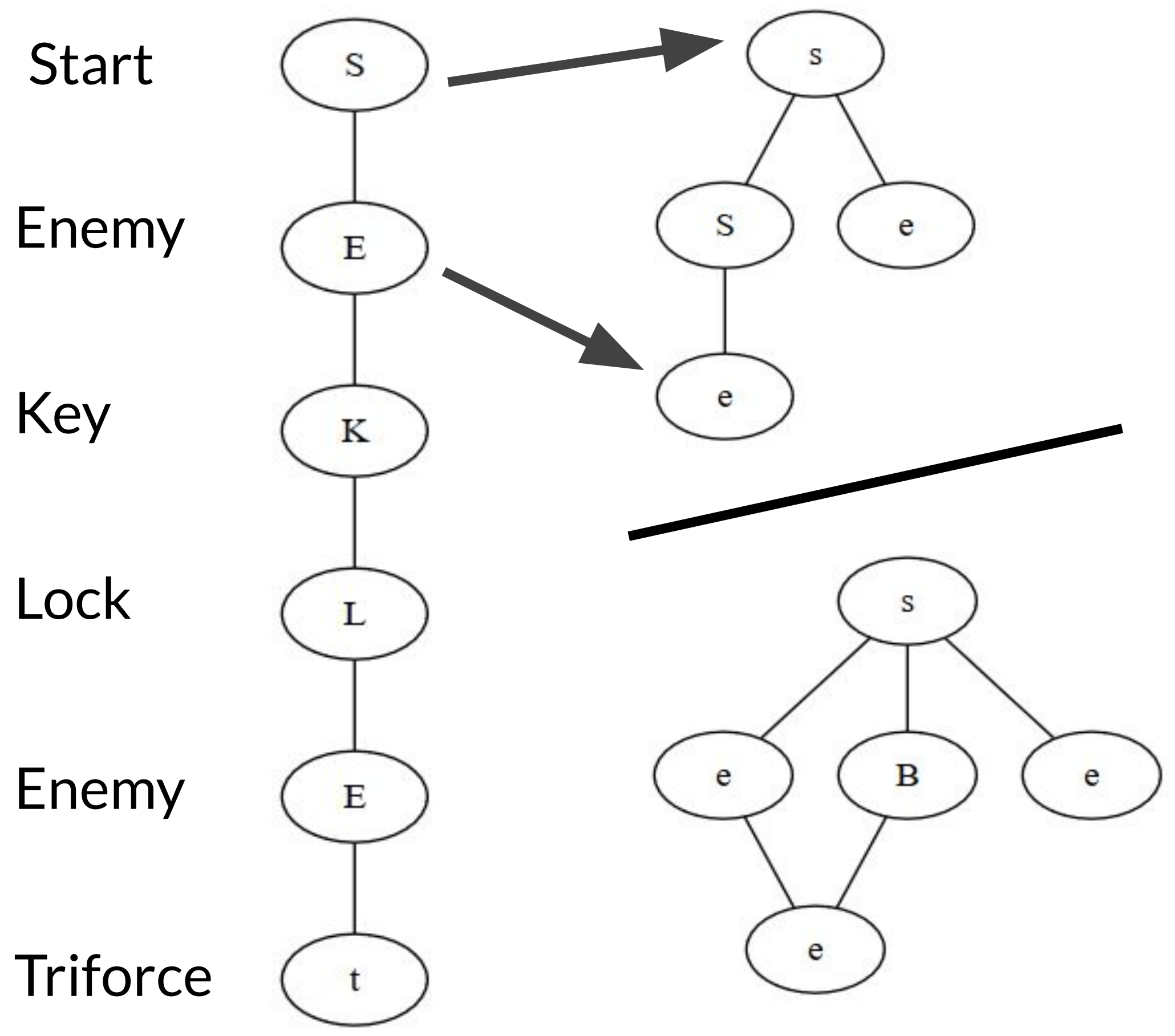

Figure 1. Find the graph rule Start -> Enemy from the starting graph mission. The two mini-graphs on the right show which one can replace S -> E

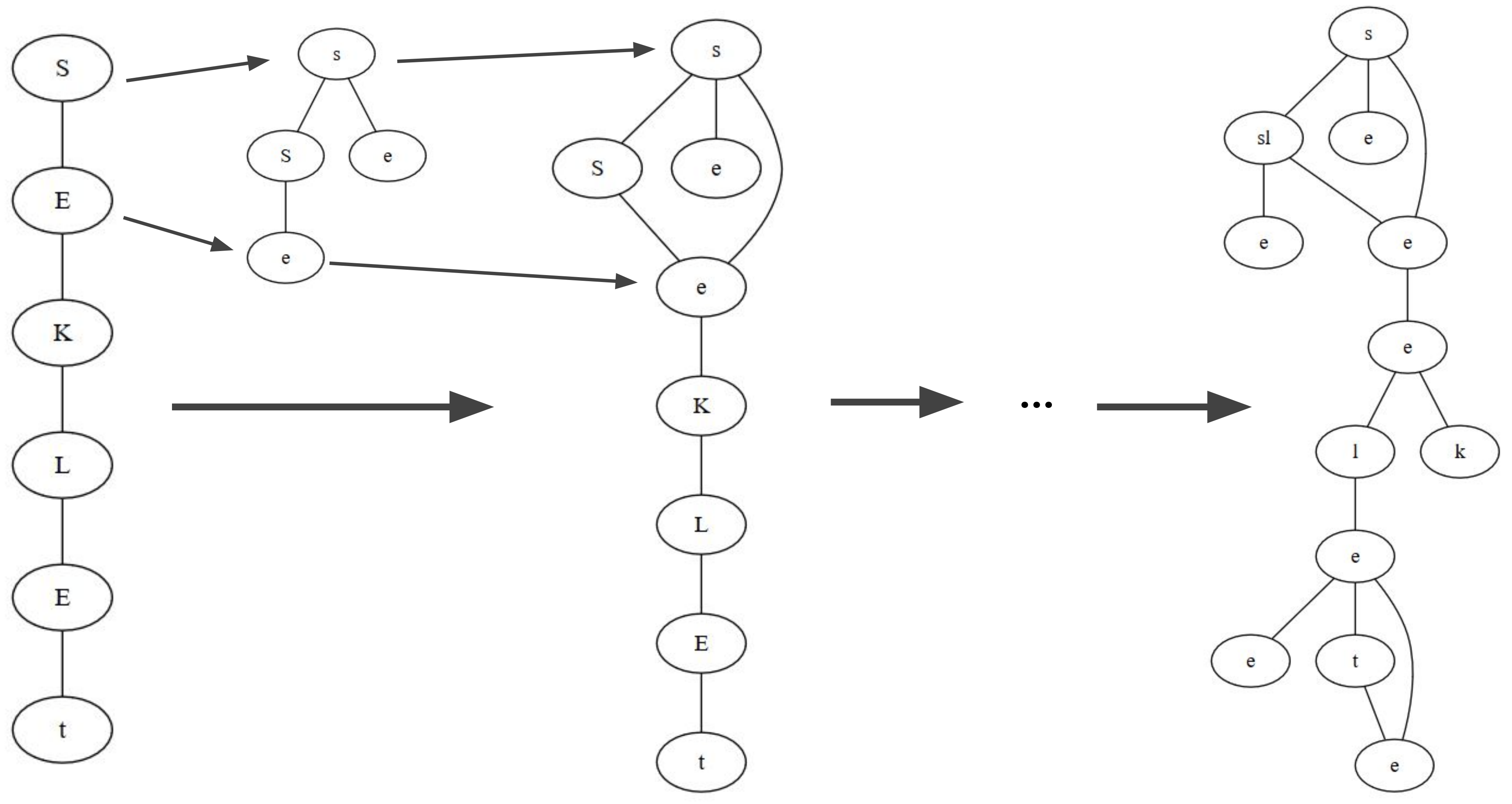

Figure 2. Apply the graph grammar to the specific rule until there are no non-terminal symbols.

Graph to Dungeon

From a finished graph, we can place the nodes of the graph into a 2D layout while placing elements based on the node.

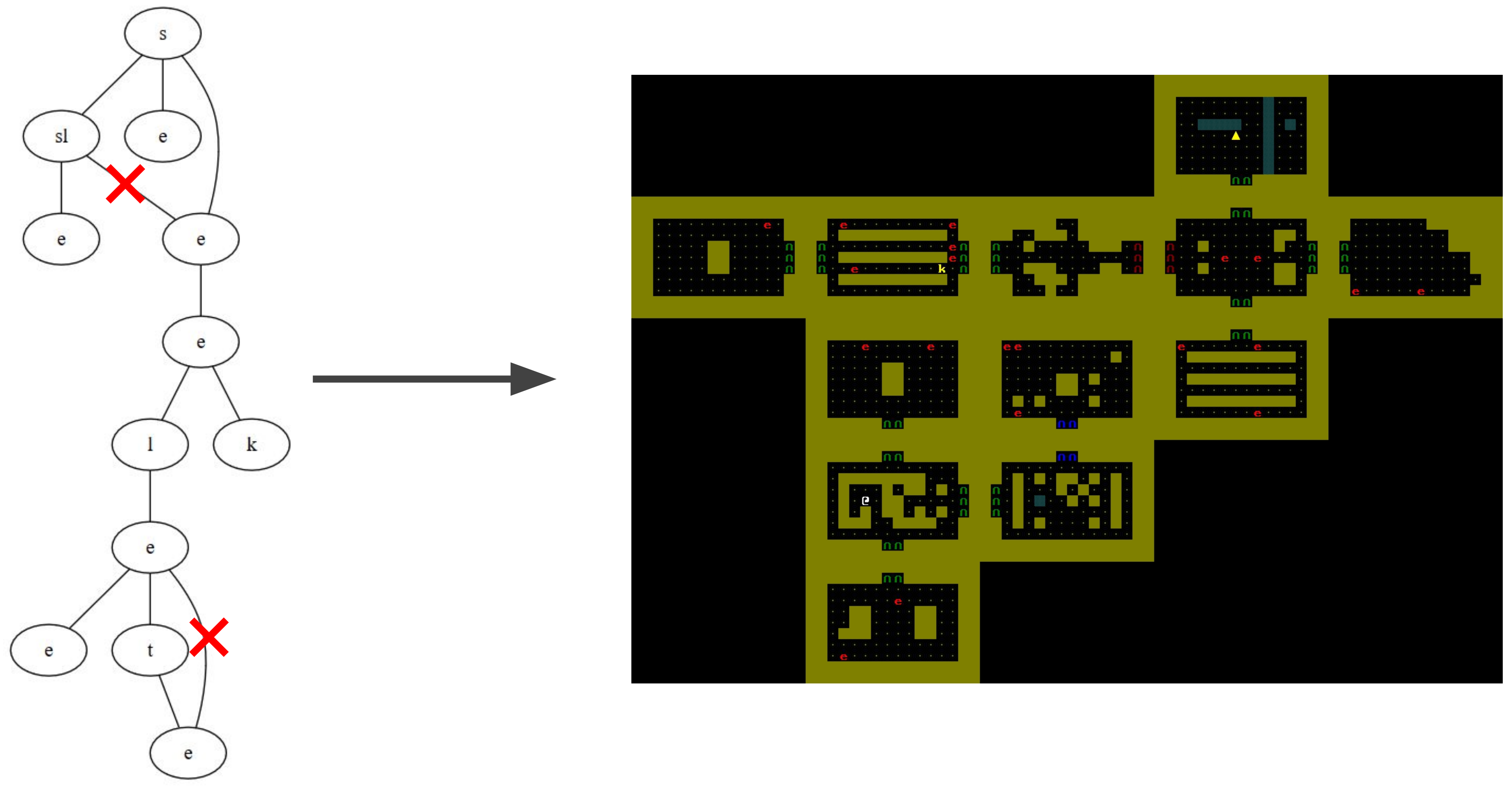

Figure 3. Going from a finished graph to a 2D dungeon with generated rooms and placement of items and enemies

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.

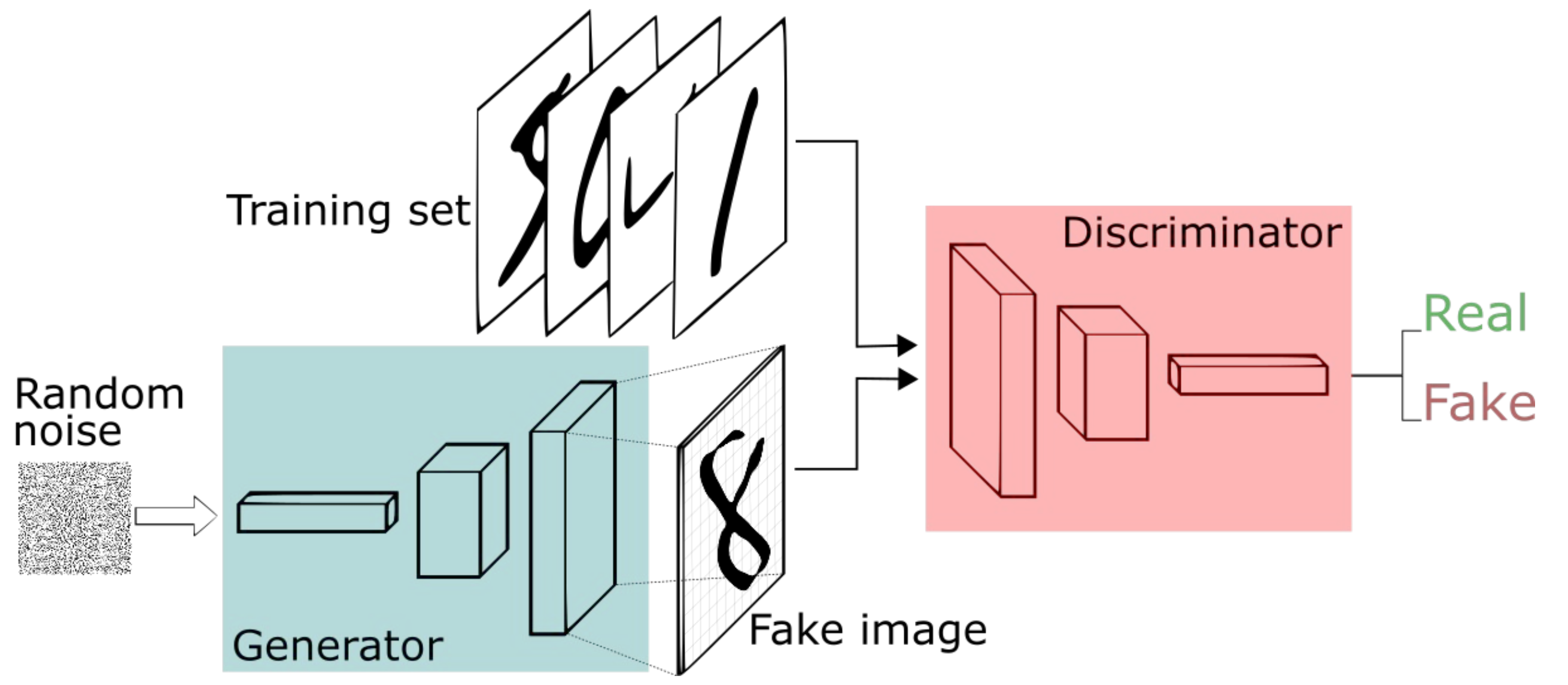

Figure 4. GAN Training. The generator maps random noise to rooms, whereas the discriminator determines whether the output from the generator is real or fake based on the training data. Through training, the generator learns to produce more convincing fake rooms. Image credit to Thalles Silva from freeCodeCamp

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

[1] J. Dormans. 2010. Adventures in Level Design: Generating Missions and Spaces for Action Adventure Games. *PCGames*

[2] V. Volz, J. Schrum, J. Liu, S. Lucas, A. Smith, S. Risi. 2018. Evolving Mario Levels in the Latent Space of a Deep Convolutional Generative Adversarial Network. *Genetic and Evolutionary Computation Conference,*