Procedural Generation Using Markov Models

Alexandru Loghin

July 10, 2019

ABSTRACT

The main selling point of this approach is the possibility of dynamically changing the requirements when generating a given set of objects. The idea of this method is to infer the probabilities for a given model using an observed sequence of labels then use the trained model to generate new randomly generated arrangement of set labels, which in this case are objects, in order to construct the environment. I used a game engine called *Unity*, in order to develop and prove that this method is valid. This thesis revolves around the library I built inside the game engine that models the behavior of a HMM [a] alongside with the algorithm for the automated learning in [1]. As with other methods this model suffers from local optima since it is an iterative algorithm that relies on convergence in order to terminate [2]. It was also necessary to make a spawn system since the model is used just for the generation of the object sequence. The final result is a small game with a system that generates the environment on the spot, mimicking real world scenarios.

[a] Hidden Markov Model

References

[1] Madhusudana Shashanka, *A Fast Algorithm For Discrete HMM Training Using Observed Transitions*, 2011

[2] Fanny Yang, Sivaraman Balakrishnan, Martin J. Wainwright, *Statistical and Computational Guarantees for the Baum-Welch Algorithm*, 2015

[3] Gleidson Mendes Costa, Tiago Bonini Borchartt, *Procedural terrain generator for platform games using Markov chain* , 2018