Bayes2Unity: Experience-driven procedural content generation using Bayesian Networks

Danial Smith
The Department of Scientific Computing
Florida State University

Abstract

Bayes2Unity is a package that allows for easy creation and use of Bayesian networks in Unity. The package relies on three main components: Unity, Infer.NET, and Netica. Each of these products has a unique role in the package. Netica allows for easy creation of a network. The graphical user interface allows users to simply layout and connect nodes. Initial training will be accomplished in Netica as well. Infer.NET is an inference engine that completes all the assessment, updating, and prediction. Most of the computation is completed by Infer.NET. Unity acts as the game engine and development environment. It provides a front end for displaying results and more importantly, a game world to propagate the results into. Two projects currently use the Bayes2Unity package: eRebuild and Preempting Path.

eRebuild is a first-person building game focusing on math education. A player collects materials and builds structures to home displaced families. These families then request players perform a variety of other tasks to help restore and improve the community. These include trading, painting, fencing, moving, and organizing. Each of these tasks links to a math competency with the results of the level being feed to a Bayesian network powered by Bayes2Unity. This model helps select a suitable next level based on the performance on different mathematical tasks. Middle school students and teachers currently test the game with static level paths, and the project has seen great success in increasing the mathematical ability of the students who played regularly compared to those who did not play the game.

Preempting Path is a first-person exploration game that presents a series of procedural generated mazes with increasing difficulty to the player. As the player searches for the end, they may also pickup different power-ups to make it easier. The game starts with the simplest maze, a 2x1 maze, and generates a larger maze after each success. As the player completes levels, a Bayes2Unity updates a player model that represent their current play style. Preempting Path presents a simple experience that anyone can jump in and play without any explanation. We designed Preempting Path as a testbed for the experience driven procedural content generation. Initial results show a modest increase in engagement compared to players who experience a generic procedurally generated game world.

While initial results look promising using this method and package, there is always work to be done. Bayes2Unity is in early stage, and in the next iteration the reliance on Netica, a proprietary and costly software package, may be removed entirely, replaced by a more robust user interface in Unity itself. This streamline allows the use of a single environment. A long-term study should be completed that compare users of the default version of eRebuild to one where levels and learning support are offered as the model sees fit.

Usage

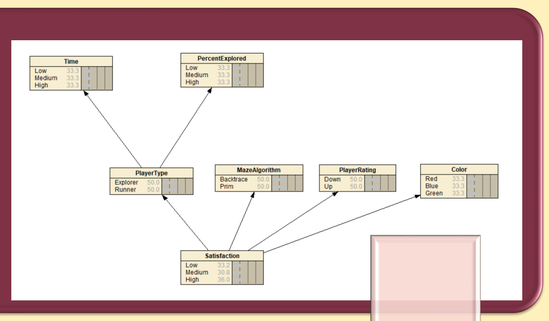
The Bayes2Unity package combines two tools into a workflow that allows for simple, graphical creation and use of Bayesian networks into the Unity game engine. Networks are created and trained using the full featured Netica environment.

Once created the network is exported and read into Unity. At this point it is transformed into a network that works with the Microsoft Research inference engine, Infer.Net.

Once imported, the network nodes can be used like any other variable. The probabilities of each state are returned after prediction, and observations of gameplay or user data can be used to further train the network.

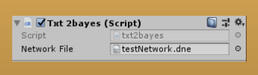
Create and train network

- Design nodes and connections using Netica
- Initialize node weight
  - Data driven
  - Expert opinion
  - Random



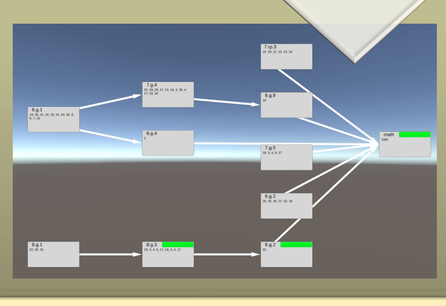
Import network

- Append .txt to file making it readable in unity
- Select file as target in Unity editor
- Bayes2Net translates from Netica's DNE to Infer.NET nodes and connections



Update and use network

- Use game data as observation for node
- Predict any unobserved nodes
- Update node weights based
- Repeat as necessary to increase accuracy of model



In-game Implementations

Two gameshave been designed using Bayes2Unity to manage the models used to adjust procedural generation parameters, eRebuild and Preempting Path.

Preempting Path was designed to test the effectiveness of using experience-driven procedural content generation to increase engagement. The model tracks gameplay habits and the generation parameters are tweaked to create a new maze for the player based on their play style.

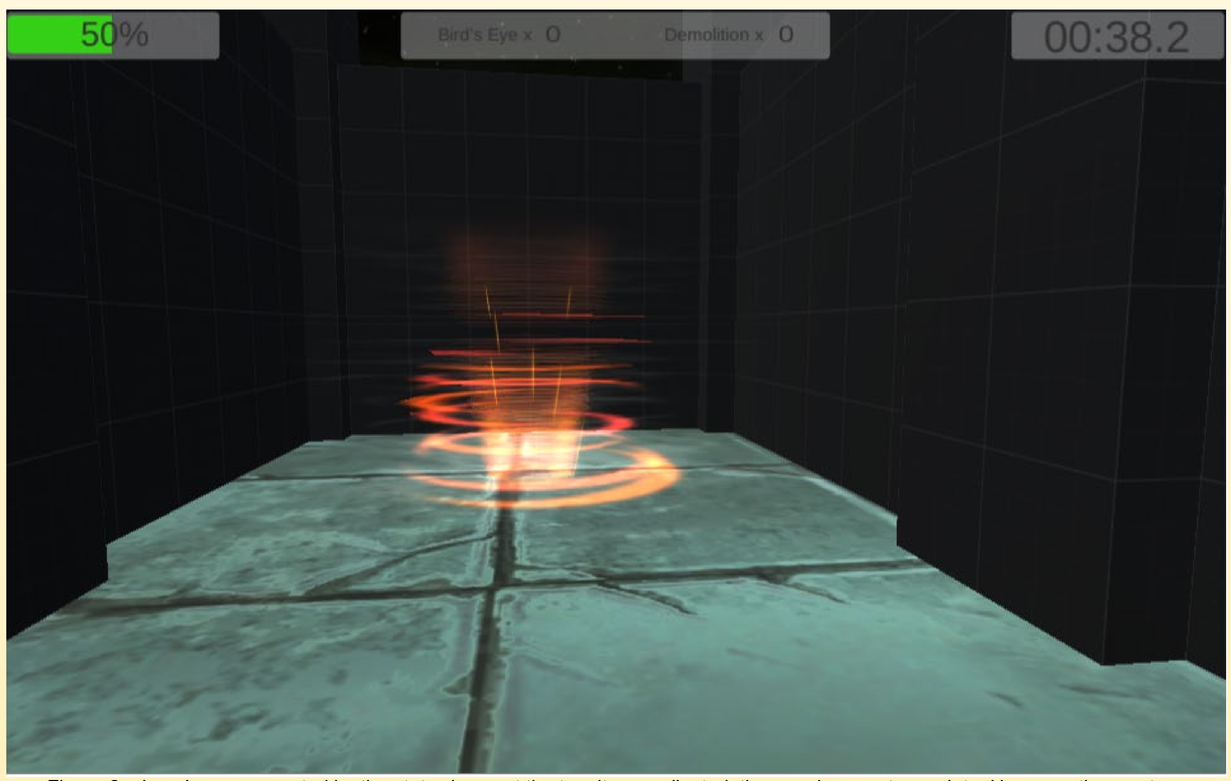


Figure 2: Levels are generated by the stats shown at the top: items collected, time, and percent complete. However, the most important factor is the rating they give each level

eRebuild is a more ambitious project. More than a game, eRebuild was designed as a math learning platform for teachers, students, and parents. All design decisions have math learning at the forefront while maintaining the fun of the game. Here, the Bayesian network models a player's mathematical abilities. This information is used to select levels of appropriate difficulty if available. If not, a level is generated to target deficiencies in the student's math skills.

In addition to procedurally generated levels, a level editor is included to further increase the amount of suitable levels. The mixed-initiative environment allows a creator to create as much or little as they desire.

Figure 3: In eRebuild, students can login and continue the game from school or home using the same player model. Teachers and parents can check math skills using this login info.

Future Work

Bayes2Unity allows for a quick implantation of Bayesian models in Unity but requires several steps and software environments. Netica was chosen due to its use in several projects the eRebuild team is associated with. making it a good choice for us. However, the added step will be unnecessary for most other use cases. By creating a GUI for Bayes2Unity in the Unity editor, the workflow is simplified, and the network can be more easily visualized and debugged as it evolves.

Acknowledgements

This work would not be possible without the eRebuild team. A special thank you to Dr. Gordon Erlebacher and Dr. Fengfeng Ke for their guidance and patience over the years.