

www.timepath.io

Simple Behavior Networks

Simple Behavior Networks is a Unity plugin from Timepath designed to facilitate the creation of smart interactive characters. Simple Behavior Networks implements Extended Behavior Networks as defined in Dorer (2004). It also includes an intuitive user interface and a carefully thought modular structure to simplify as much as possible the design of behavioral artificial intelligence.

Introduction

In videogames, artificial intelligence is often created with Behavior Trees, a simple yet robust method to create characters with simple rules, such as the ones needed in first person shooters. Several implementations are available in Unity.

Behavior Networks are a more powerful alternative to Behavior Trees, particularly for dynamic and fast-changing environments, which require more reactive and sophisticated behavior.

Behavior Networks are also more intuitive to design. Simple Behavior Networks work by defining Skills and Goals in the interactive character. A goal has a syntax like:

When **x** I want **y**

A Skill is defined in a syntax like:

If **x** and **y** doing **a** has effect **z**

Both for goals and skills, **x**, **y** and **z** are Perceptions that can be quantified by a function returning a value between 0 and 1. Typical examples of Perceptions can be "I am near a fridge", "I have the thermometer", "I do not have ammo", "She is smiling", etc. In addition, in Skills, **a** is an action, which can be performed with a certain intensity, also between 0 and 1.

Decisions In Behavior Networks are taken through an energy spreading mechanism: a certain skill will perform a certain action if its effect is expected to contribute to a goal. Skills also self-organize according to preconditions (in the previous example, **x** and **y**) and its effects in the previous example, **z**). For example, if I want to kill a character, but I need a gun to do so, a skill to go pick the gun will automatically receive a lot of energy. Later on, once the gun is obtained, then the energy will move from the search to the shooting skill, now more likely to contribute to achieve the goal.

User Interface

Simple Behavior Networks come with a dedicated user interface, designed to allow non-technical users to create their own skills and goals, and allowing to visualize during game playout the evolution of the character and the behavior network.

By dragging and dropping, the AI designer can define the relations between Perceptions and Goals. During playout, the AI designer can see how the changes in the perception values affect the relevance of a given goal. For example, in figure 1, a simple diagram corresponding to “When: not Perception0 and Perception1 I want: Perception0” is shown. The black and gray rectangles show the extent at which a certain perception is true: in this case, Perception0 is true at 83%, and Perception1 is true at 54%. The green and red lines show the relation between the Goal and the Perceptions. The Goal, shown as a blue box, has a relevance corresponding to the smallest condition satisfied (in this case, the negation of Perception0, which is true at 17%). The evolution of the Perception values will in turn affect the Relevance of that particular Goal.

The user interface also shows the relevance of a given Goal, which evolves in real time depending on the different perceptions. In their turn, Perceptions can be defined through dropdown menus, from a set of simple functions that return a value between 0 and 1, and which can also be customized by the end-user (see figure 2). Skills and actions are defined in a similar way to goals and perceptions.

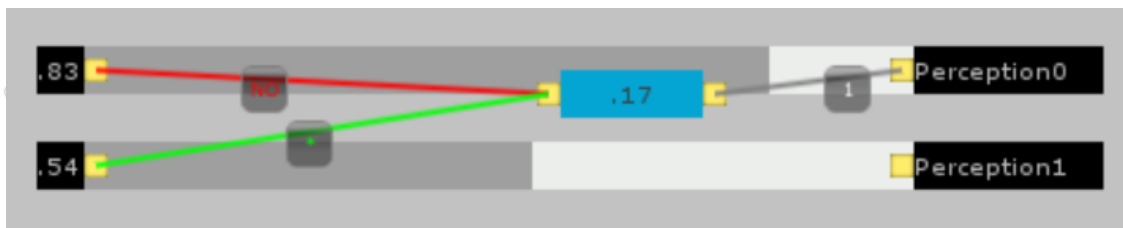


Figure 1: a simple user interface also allows visualizing the real time evolution of perceptions and goals during playout

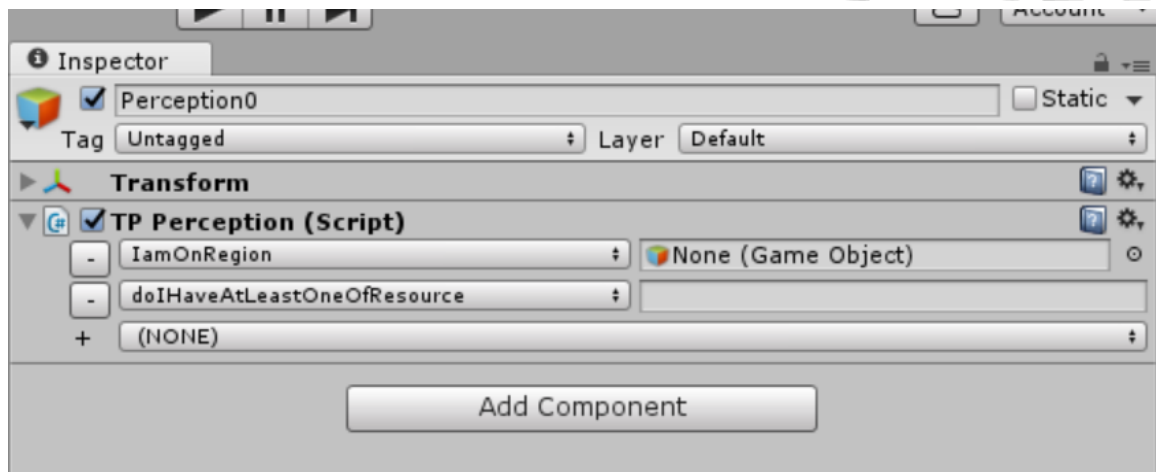


Figure 2: through the Unity Inspector, perceptions can be defined easily from a list of pre-defined and customizable functions.

Animation Synthesis

Simple Behavior Networks provide a modular way to design smart interactive behavior, based on goals and skills. Consistently, it also includes an innovative method to blend different animations together (Shoulson 2014). This library integrates several character animation techniques to easily method, and it is easily extendable and available open source.

Customization and extension

Simple Behavior Networks can be defined as prefabs, stored, shared and modified like any other character model or animation fil3. In addition, the list of Perceptions and Actions already available can be customized by selecting combinations of built-in perception and action functions, or expanded through small chunks of code leveraging the power of the Unity3D API.

For further information, please go through the tutorial included with the Unity Package.



References

- Dorer, K. (2004). Extended behavior networks for behavior selection in dynamic and continuous domains. In *Proceedings of the ECAI workshop Agents in dynamic domains, Valencia, Spain*.
- A. Shoulson, N. Marshak, M. Kapadia, and N. I. Badler (2014) ADAPT: The Agent Development and Prototyping Testbed. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*,