

Behaviour Machine Getting Started

Anderson Cardoso

July 23, 2014



1 Introduction

Behaviour Machine is a visual scripting plugin to design of Hierarchical Finite State Machines and Behaviour Trees in the Unity 3D game engine.

With Behaviour Machine, designers and artists can create games without coding, and programmers can better organize their scripts and easily build complex interactions.

This guide will serve as a basic introduction to the plugin, for more information visit the [user manual](#).

2 Installing

Download and import the BehaviourMachine.unitypackage from the Asset Store. After the installation you should have a folder called 'Plugins' in the Project View. All assets belonging to the Behaviour Machine plugin are in the 'Plugins / BehaviourMachine' and 'Plugins / Editor / BehaviourMachineEditor' folders.

3 Key Concepts

This section has an overview about the key concepts of Behaviour Machine. The plugin is component based, so all elements presented below (except the Node) are Unity components.

3.1 StateMachine

The StateMachine is a Unity component that controls other components by enabling or disabling them. The components controlled by the StateMachine are called states (Section 3.2).

At any given time, only one state can be active. There must always be an active state if the StateMachine is enabled. When a state is enabled it'll perform some actions, or wait some time.

3.2 State

States are also components. You can code your own state using the MonoBehaviour API (Awake, Start, Update...).

Don't know how to program? No problem, the Behaviour Machine has two visual scripting components/states: the ActionState (Section 3.3) and the BehaviourTree (Section 3.5).

States are decoupled from the StateMachine, this means you can use them as regular components without a StateMachine controlling them.

3.3 Action State

The ActionState is one of the two visual scripting components. It has a simple list of nodes (Section 3.4) that runs when the component is enabled.

3.4 Node

A node is not a component, it's a simple script that has a very specific goal. Nodes are used by the visual scripting components to create different types of interactive objects, A.I. behaviors, controllers, gameplay, and much more.

The Behaviour Machine already comes with hundreds of nodes, you can also script your own custom game node.

3.5 Behaviour Tree

Despite of being a relatively new way of develop decision making, the Behaviour Trees are widely used in commercial games (e.g. Halo 2, Spore, GTA, Prototype).

The BehaviourTree is a component that has a hierarchical set of nodes, you can combine these nodes in many different ways to create unique behaviours.

The node's parent-child relationship gives you the power of a programming language in a drag-and-drop interface.

3.6 Blackboard

The Blackboard is a component that holds values, some objects write values on the Blackboard and others read these values. This allows objects to communicate without need for them to know each other.

The Blackboard can be used by nodes, states or any other component/script.

4 Where do I Start?

If you are not familiar with the concepts presented above or you want to know how they work in Behaviour Machine, then take a look at the sample scenes on the folder "Plugins/BehaviourMachine/SampleScenes/".

If you are a programmer that wants to script your own custom states or nodes, then checkout the [scripting section](#) on the online manuals.

Have fun :)