

release V1.3.1

行为树 AkiBT Verisoin 1.4.0 简介 Intro

爱姬kurisu优化GraphView视图并拓展内置行为和编辑器功能的行为树。
行为树衍生自UniBT,原作者Yoshida.

AkiBT is a visual node editor derived from UniBT created by Yoshida for making behavior tree or other tree-based function. AkiKurusu Extends it with more features so that you can enjoy it.

安装 Setup

1. Download [Release Package](#)
2. Using git URL to download package by Unity PackageManager
<https://github.com/AkiKurusu/AkiBT.git>

支持的版本 Supported version

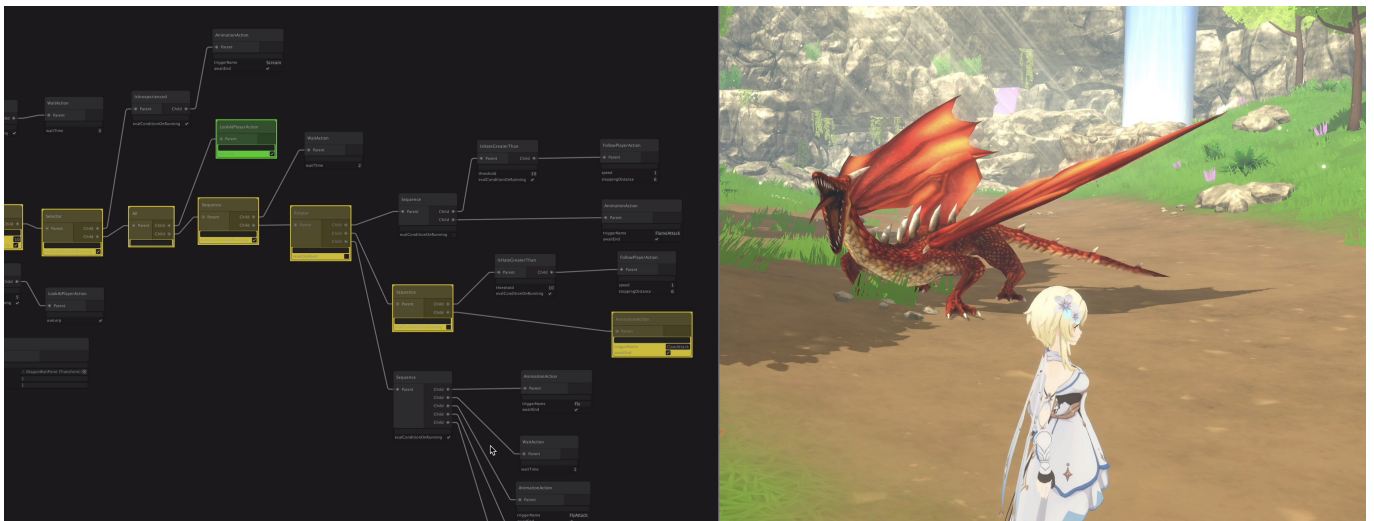
- Unity 2021.3 or later.

特点 Features

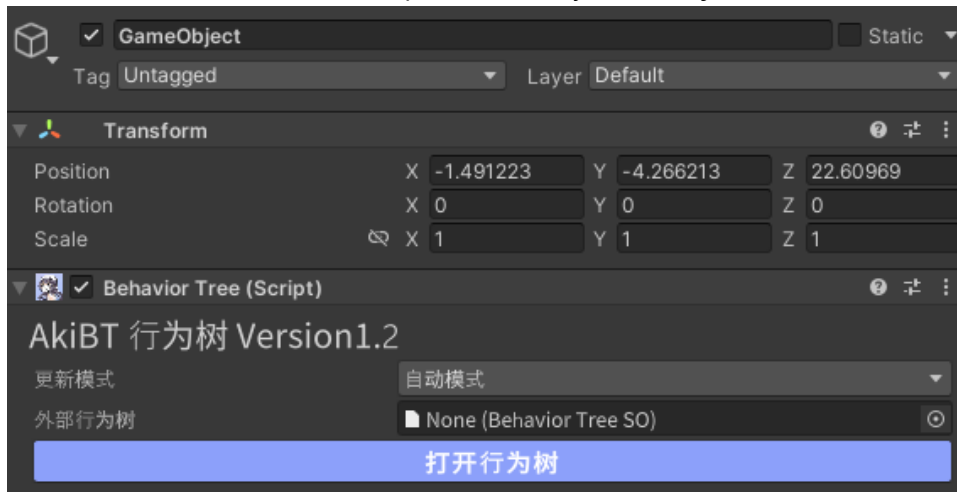
- 支持使用可视化节点编辑器构造行为树
- Supports constructing behavior tree by visual node editor.
- 支持运行时可视化结点状态
- Supports visualizing active node in runtime.
- 非常便于拓展和自定义新的行为
- Easily add original behaviors(Action,Conditional,Composite,Decorator).

使用方式 How To Use

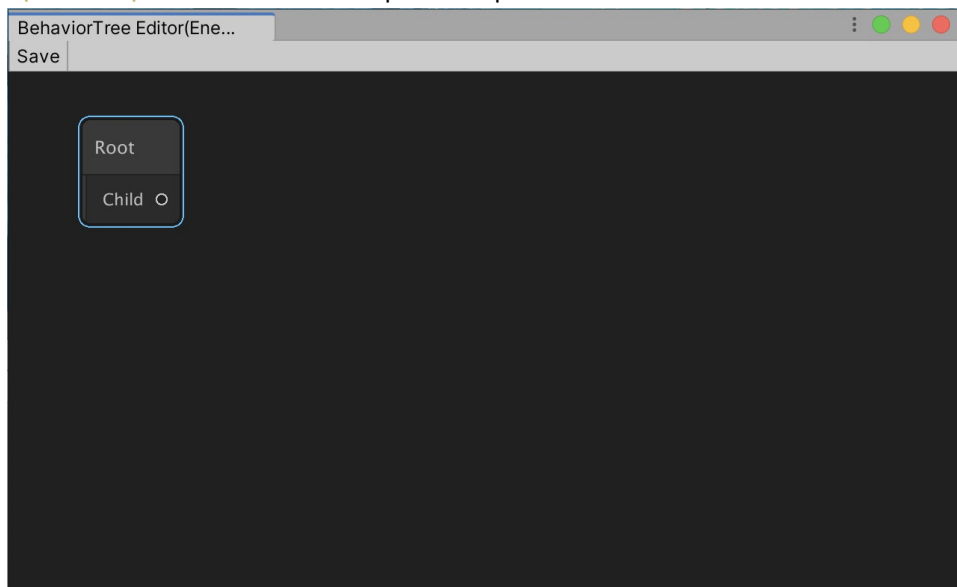
Modified from [UniBT documentation](#)



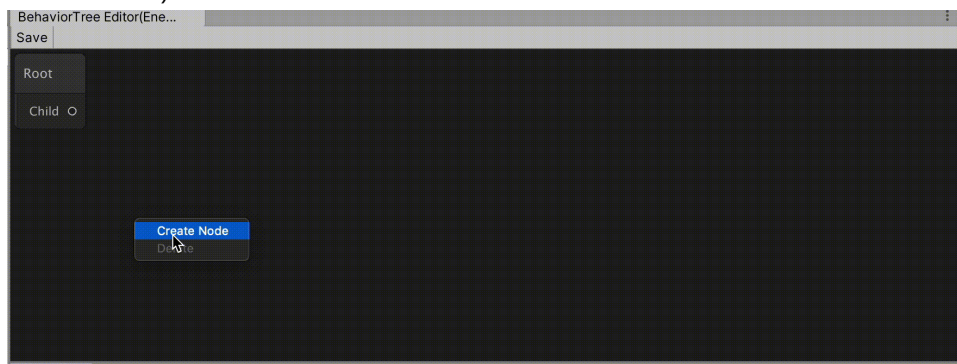
1. Add **AkiBT.BehaviorTree** component for any GameObject.



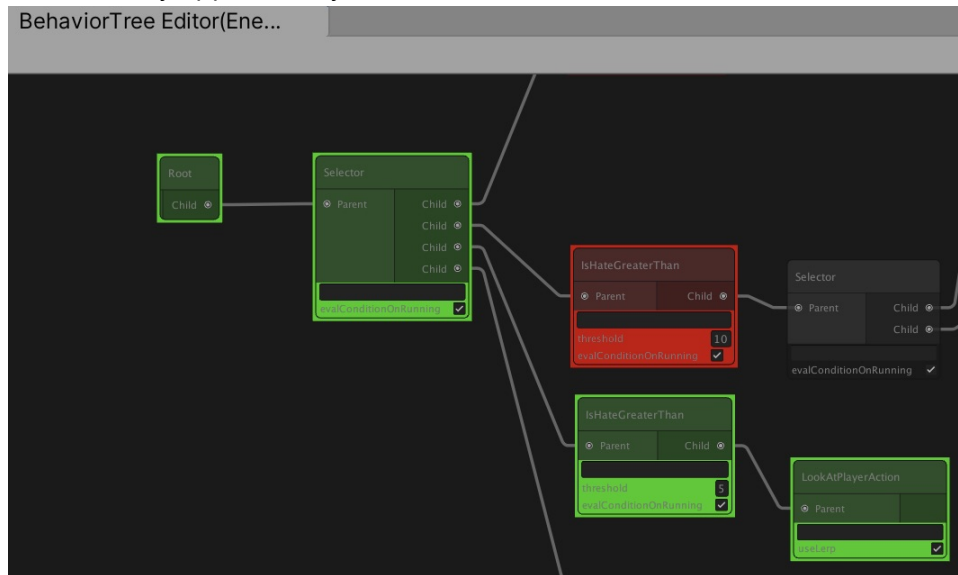
2. **Open Graph Editor** button opens GraphView for Behavior Tree.



3. Add behaviors and set parameters.
4. Finally press save button on tool bar of the editor window. (If invalid node found the color of the node become red.)



- Run the unity application. you can see node status in the editor window.



- The red node means that last **Update** returned **Status.Failure**.
 - The green node means that last **Update** returned **Status.Success**.
 - The yellow node means that last **Update** returned **Status.Running**.
- you can save the GameObject with **AkiBT.BehaviorTree** as prefab or save to ScriptableObject or save to json file.
 - Tutorial Video On BiliBili

[开源行为树AkiBT使用教程](#)

工作原理 How It Works

- AkiBT.BehaviorTree** updates child nodes in **Update** timing when the UpdateType is **UpdateType.Auto**.
- If you want to update at any time, change UpdateType to **UpdateType.Manual** and call **BehaviorTree.Tick()**;
- Only **AkiBT.BehaviorTree** is the **MonoBehavior**. Each node is just a C# Serializable class.

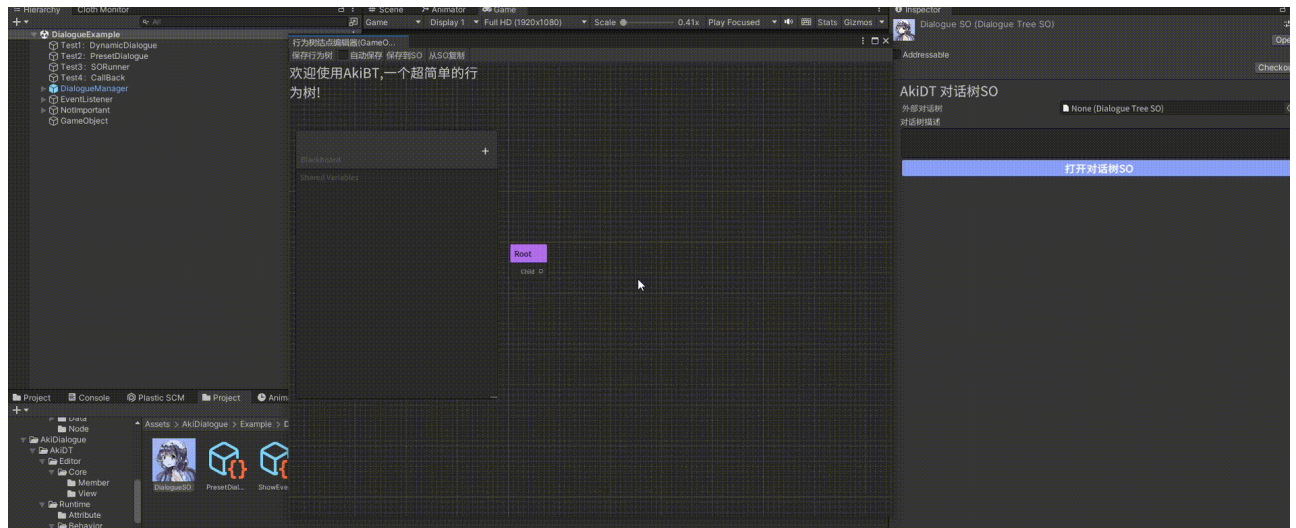
拓展结点 Extend Node

See [API Document](#)

编辑器功能说明 Editor Function Info

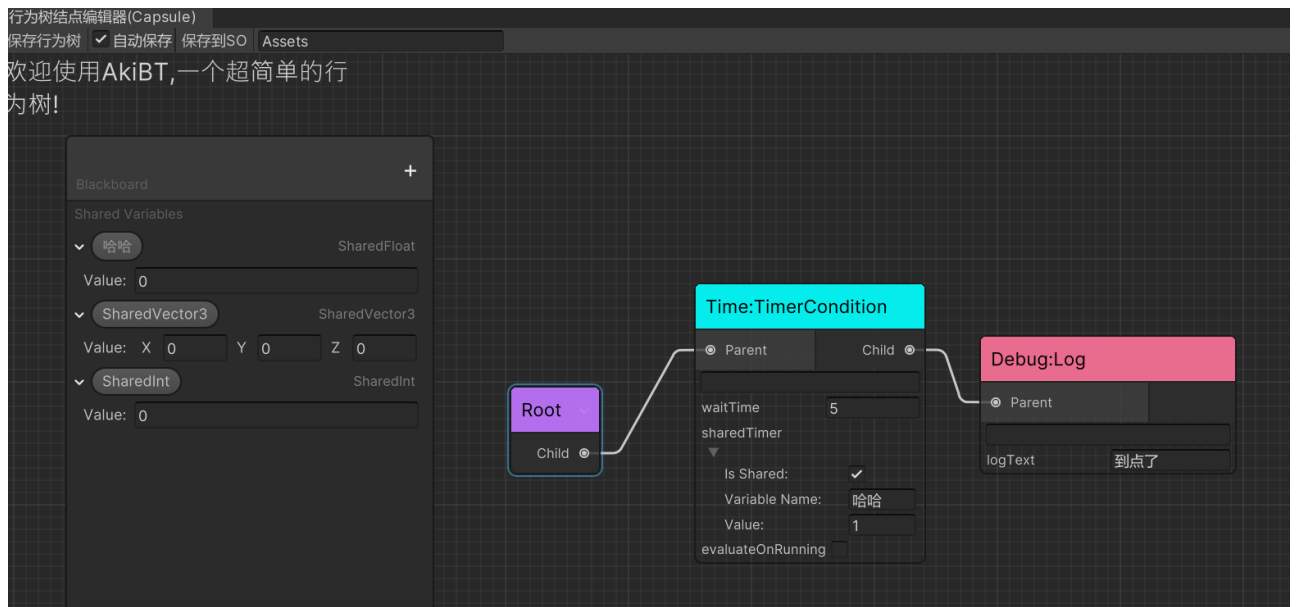
- 你可以从上侧工具栏选择 **Copy From SO**, 也可以将BehaviorTreeSO、BehaviorTree组件或挂载BehaviorTree的GameObject、Json文件拖拽进编辑器中进行复制粘贴。

You can select **Copy From SO** from the upper toolbar, or you can drag BehaviorTreeSO, BehaviorTree components, or GameObject and Json files that mount BehaviorTree into the editor to copy and paste.



2. 共享变量SharedVariable可以在黑板中添加,目前支持Float、Int、Vector3、Bool、String、UnityEngine.Object类型变量

SharedVariable let you have access to add it in a blackboard and share value between different node.Now it supports Float,Int,Vector3,Bool,String,UnityEngine.Object,

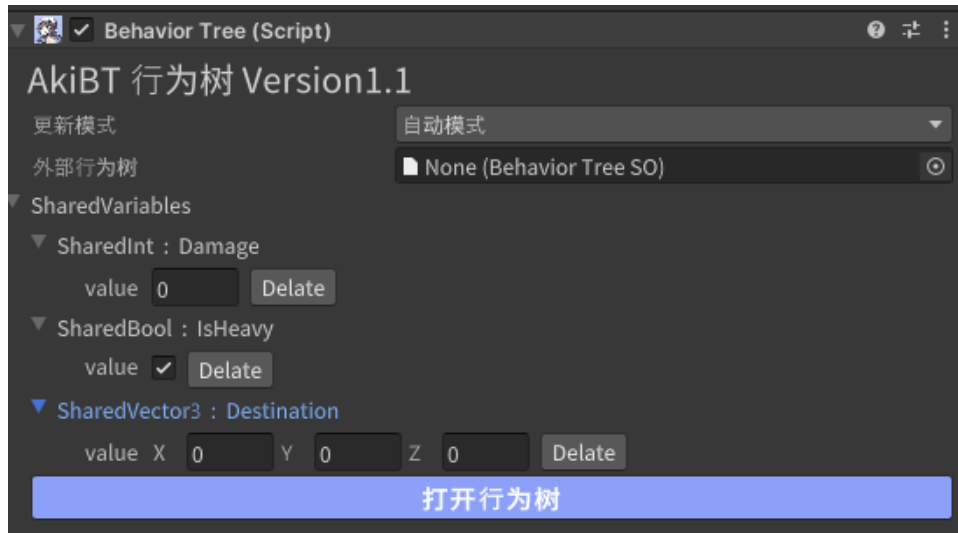


- 注意：修改共享变量名称的方式为双击变量,为空时自动删除

You can edit variable's name by double-click it and the variable will auto delete when it's name becomes empty.

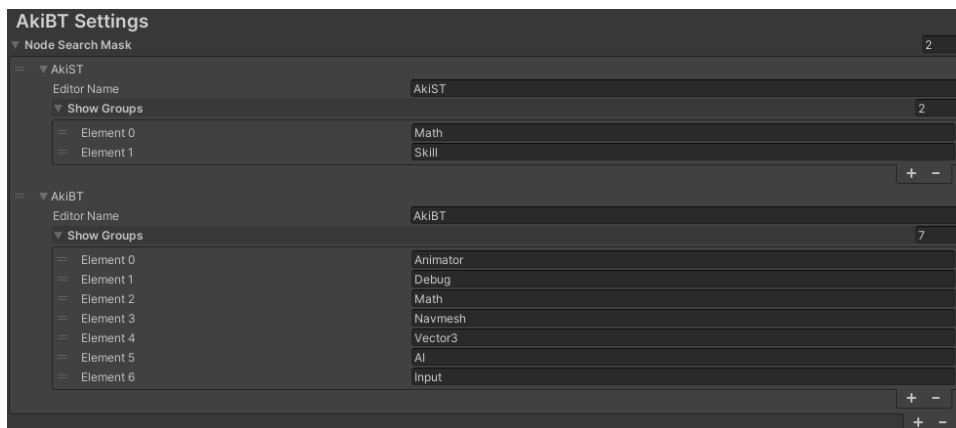
- Inspector中也可以进行共享变量的修改和删除功能

You can edit the value of SharedVariable in the inspector.



3. 你可以在ProjectSetting中设置AkiBT编辑器或者其余继承自AkiBT的编辑器的搜索遮罩。你可以设置工作中需要的Group类型（Group特性相关见上文）,没有添加Group特性的结点不会被过滤。

You can edit the setting in ProjectSetting where you can add mask for the node group you want to see in the SearchWindow.To be mentioned,the mask is relating to the editor you used.As default,the AkiBT editor is named with 'AkiBT',so you should edit the 'EditorName' with it.



拓展功能 Extra Module

1. 运行时更新 Runtime Update Support

你可以使用[AkiBTVM](#)实现运行时编辑行为树

You can have access to runtime-updating by using [AkiBTVM](#)

2. 开发便捷服务 User Service

插件目前内置了新的User Service(Tools/AkiKurusu/AkiBT User Service), 提供了两个功能Serialize Service和Search Service

- Serialize Service:由于AkiBT使用ScriptableObject进行数据存储,在修改结点的字段名称时会导致数据的丢失（该问题可以通过在修改字段上添加[FormerlySerializedAsAttribute](#)进行避免）。而对于结点的名称、命名空间进行修改后也会导致整个结点无法被反序列化,从而丢失该结点以及之后结点的所有数据。序列化为Json后, 你可以使用文本编辑器批量对结点进行修改, 再重新反序列化为ScriptableObject。

需要注意的是, 并非ScriptableObject所有的字段都被序列化, Serialize Service只对行为树的结点和共享变量进行序列化,反序列化同理。

- Search Service:选择结点类型快速找到使用该结点的所有行为树, 结合Serialize Service可以同时找到对应的Json文件。

The plug-in currently has a new User Service (Tools/AkiKurusu/AkiBT User Service) built in, which provides two functions Serialize Service and Search Service

- Serialize Service: Since AkiBT uses ScriptableObject for data storage, data loss will occur when modifying the field name of the node (this problem can be avoided by adding `FormerlySerializedAsAttribute` to the modified field). However, after modifying the name and namespace of the node, the entire node cannot be deserialized, thus losing all data of the node and subsequent nodes. After serializing to Json, you can use a text editor to modify the nodes in batches, and then re-deserialize to ScriptableObject.

It should be noted that not all fields of ScriptableObject are serialized. Serialize Service only serializes the nodes and shared variables of the behavior tree, and the deserialization is the same.

- Search Service: Select a node type to quickly find all behavior trees using the node, and combine with Serialize Service to find the corresponding Json file at the same time.