Misbehave

Aug 05, 2020

Contents:

Misbehave is a Python behavior tree implementation. A behavior tree is a design pattern that breaks complex state-based actions into their component parts to improve readability and re-usability for autonomous agents. A good starting place to understand them is here.

Misbehave focused on python language features not available in other languages, so some of the common features of AI design are missing by default. The rest of this document is lightly hedged in its language as behavior trees can be useful for video games or robots.

CHAPTER 1

Getting Started

Misbehave does little on its own, and is best used as a dependency for a game project that you're working on. To start, you'll want to add it to your dependency list in requirements.txt:

And then install it into your virtual environment

Once you've set that up, you can define your tree like so:

```
import misbehave

import my_actions

tree = misbehave.selector.Sequence(
    misbehave.action.IncreaseValue("behavior_tree_runs"),
    my_actions.MoveForward(10),
    my_actions.Rotate(90)
)

running = True
while running:
    tree(my_actor, None)
```

This simplified example demonstrates the basic usage:

You combine selectors with your own action and decorator nodes to produce a tree structure. Then, each tick (or some number of ticks) through your main loop you pass in the object the tree controls, and a context object (which is `None` here.)

CHAPTER 2

API Reference

CHAPTER 3

Indices and tables

- genindex
- modindex
- search