**Основные определения.**

Дерево поведения (Behavior Tree, BT) – ориентированный ациклический граф, узлами которого являются возможные варианты поведения ИИ.

«Ширина» дерева указывает на кол-во доступных действий, «длина» - на сложность.

Узлы BT – задачи, или поведения, имеют одно из четырёх состояний:

1. Успех – задача выполнена успешно

2. Неудача – условие не выполнено или задача невыполнима

3. В работе – задача запущена и ожидает завершения

4. Ошибка – в программе возникает неизвестная ошибка

Типы узлов:

1. Действие (Action) – запись переменных или какое-то движение

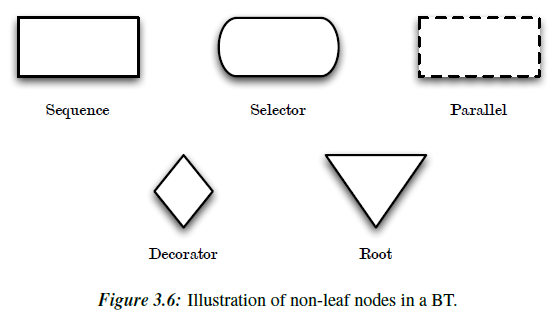
2. Исполнение последовательности (Sequence) – поочерёдно выполняют поведение каждого дочернего узла, пока один из них не вернёт одно из состояний.

3. Параллельные (Parallel) – выполняют поведения дочерних узлов, пока заданное кол-во из них не вернёт «Неудачу» или «Успех».

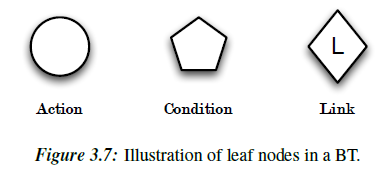
4. Селекторы (Selector) - поочередно исполняют поведения каждого дочернего узла до тех пор, пока один из них не выдаст значение «Успех», «В работе» или «Ошибка». Если этого не произошло, возвращает значение «Неудача».

5. Условия (Condition) - содержат критерий, по которому определяется исход, и переменную. Например, условие «Есть ли в этой комнате человек?» перебирает все объекты в комнате и сравнивает их с переменной «Человек».

6. Инверсии (Inverter) - выполняют функцию оператора NOT.



Неконечные узлы должны иметь любое количество дочерних узлов (кроме корневого и декоратора, у которых должен быть только один дочерний узел). Неконечные узлы можно рассматривать как задачи, которые нужно выполнить. Каждая задача определяется поддеревом, состоящим из различных элементов. Существует два основных неконечных узла, которые дополняют друг друга, — это селекторы и последовательности. Также существуют декораторы, обеспечивающие большую функциональность ВТ, и параллельные узлы, добавляющие параллелизм.



Конечные узлы в ВТ определяют наблюдение и взаимодействие с игровой средой. Их называют действиями, условиями и ссылками. Конечные узлы часто рассматриваются как элементарные действия, которые должны быть как можно более краткими.

**Принцип работы.**

Выполнение ВТ происходит вглубь, начиная с корневого узла. Обычно все неконечные узлы будут выполнять их дочерние элементы слева направо, но есть исключения. Когда узел завершает выполнение, он возвращает статус, который может быть успешным, неудачным или исключением. Обстоятельства, при которых возвращается статус исключения, зависят от типа узла. Исключения возвращаются в том случае, если узел не смог вернуть ни успешного, ни неудачного статуса.

1. Селекторы (Selector) – последовательно выполняет свои дочерние узлы слева направо, пока не получит успешный ответ, после чего сам селектор возвращает успешный ответ. Если же один из дочерних узлов вернёт неуспешный статус, будет выбран следующий дочерний элемент. Если все дочерние элементы вернут неуспешный статус, и узел селектора вернёт неуспешный статус.

2. Вероятностный селектор (Probability selector) – узел селектора с распределением вероятности по его дочерним элементам. Вероятность указывает, насколько возможен выбор дочернего элемента. Если выбранный дочерний узел вернёт неуспешный статус, распределение вероятности будет нормализовано по оставшимся дочерним узлам, после чего будет выбран новый дочерний узел. Возвращаемое значение вероятностного селектора выбирается по тем же правилам, что и для обычного.

3. Последовательность (Sequence) – узел последовательности будет выполнять каждый из своих дочерних узлов слева направо, пока все узлы не вернут успешный статус, либо пока один из них не вернёт неуспешный статус. В первом случае узел последовательности вернёт успешный, во втором случае – неудачный статус.

4. Декоратор (Decorator) – добавляется в ВТ для обеспечения большей гибкости. Декоратор позволяет добавлять новое поведения без изменения существующего кода. Чаще всего используются в качестве фильтров, чтобы действия выполнялись с некоторым условием. Например, чтобы действие выполнялось определённое количество раз, с некоторой вероятностью и т. д. Помимо фильтрации декоратор можно использовать для практически любой модификации поведения. Декоратор может иметь только один дочерний узел.

Декоратор возвращает успешный статус, если заданные условия выполнены и дочерний узел был выполнен успешно.

5. Параллельный узел (Parallel node) - параллельно выполняет все дочерние элементы. Обстоятельства, при которых узел возвращает успех или неудачу, задаются разработчиком. Возможно, он будет аналогичен логике селектора или узла последовательности.

6. Условие (Condition) – наблюдают за состоянием игровой среды и возвращают успех или неудачу на основе наблюдения. Это может быть сравнение значений, проверка состояния и т. д.

7. Действие (Action) – используется для взаимодействия с игровой средой посредством контролируемых действий, таких как движение персонажей и взаимодействие с объектами. Возвращает успех при удачном выполнении действия, в противном случае – неудачу.

Действие должно быть как можно более простым, то есть представлять собой отдельные действия в игровом мире.

8. Ссылка (Link node) – содержит ссылку на корень другого ВТ. Выполняет узел связанного ВТ и ожидает ответа. Вводит модульность и возможность повторного использования поведений.

**Доска объявлений.**

Доска объявлений используется для хранения информации, полученной от узлов действия или условий во время выполнения. Хранение такой информации позволяет получить данные в некоторой части дерева и сохранить эти данные для дальнейшего использования.

Доска объявлений может быть общедоступной, чтобы любой узел мог считывать и записывать данные.