**3. Behavior Tree. Blackboard**

1. Как можно программировать ИИ в UE? Как мы это делаем? Что из себя представляет дерево поведения?

2. Как его создаем? Что создаем дополнительно (с чем он работает в связке)?

3. Из чего состоит дерево поведения? Что имеем сразу в его редакторе? Какие есть виды нод? Какие есть виды управляющих нод?

4. Рассматриваем ноду для последовательного выполнения нескольких тасков – какую?

5. Какую реализуем логику в данной ноде? Через что мы определяем локацию, куда надо бежать персонажу?

6. Какую логику реализуем в ивент графе у контроллера?

1. Программировать ИИ в UE можно несколькими способами – можно описать всю логику поведения персонажа в контроллере, но есть более удобный инструмент – использовать деревьев поведения. Глобально – это математическая модель, представляющая ориентированный ациклический граф, узлами которого являются возможные варианты поведения. В UE имеется собственный редактор Behavior Tree, который в визуальной форме предоставляет данную математическую модель.

2. Чтобы создать BT, кликаем ПКМ-AritificialIntelligence->BehaviorTree. Назовем BP\_STUCharacter.

Второй ассет, который нам потребуется, это Blackboard. Находится там же, назовем BB\_STUCharacter.

Данные два ассета работают в связке.

3. Открываем редактор BT. Дерево поведения состоит из узлов (управляющей ноды) и из листьев, которые являются тасками (заданиями), в которых описываются непосредственные действия персонажа.

В начале у нас только одна нода – Root, это точка входа нашего дерева. Вытаскиваем стрелочку из нашего Root и нам предлагают выбрать одну из трех управляющих нод – Selector, Sequence и Simple Parallel.

4. Рассмотрим ноду Sequence – она обеспечивает последовательное выполнение нескольких тасков. Аналогичным образом вытягиваем стрелку из данной ноды и получаем уже выбор не только из управляющих нод, но и из тасков. Сейчас тут представлены те, что есть в UE по умолчанию. Но их можно создавать и самим.

5. Начнем с простейшего таска – MoveTo. Данный таск отвечает за поиск пути в определенную локацию – по сути, это другое представление функции MoveAITo. В панели Details у нас имеется собственное проперти, которое называется Blackboard Key. Оно имеет тип Blackboard Key Selector и через него определяется точка в мире или актор, в который мы хотим, чтобы наш персонаж начал перемещение.

При клике в пустом месте на Behavior Tree откроется его вкладка Details, где есть поле Blackboard Asset, тут надо выбрать наш ранее созданный BP-класс.

Behavior Tree – это мозг нашего персонажа, набор правил, переходов и логических состояний NPC. Blackboard представляет собой базу данных, набор переменных, значения которых мы можем менять из различных участков нашего кода. Добавлять новые переменные можно с помощью New Key. Создадим два вектора, описывающих две точки, между которыми будет перемещаться наш NPC – Location1 и Location2.

Изображение выглядит как текст

Автоматически созданное описание

Теперь в нашем таске MoveTo можем выбрать в Blackboard Key наши переменные.

Следующим таском будет задание Wait – данный таск ничего не делает, наш персонаж остановится и будет ждать указанное количество времени. Добавляем еще две ноды. После окончания последней ноды Wait наш процесс запустится снова.

У каждой ноды имеется цифра, означающая приоритет выполнения. Еще одна важная особенность деревьев поведения – таски имеют статус выполнения – успешно или нет – в случае успеха запускается следующий таск, в случае провала – вся цепочка прерывается.

Изображение выглядит как текст, внутренний

Автоматически созданное описание

6. Переходим в BP\_STUAIController. Вызываем функцию Run Behavior Tree. Дерево поведения передаем в качестве параметрва данной функции.

Изображение выглядит как текст, монитор, внутренний, снимок экрана

Автоматически созданное описание

Далее надо получить указатель на Blackboard – через Get Blackboard.

Семейство функций, позволяющих устанавливать различные значения, называются Set Value As. Для наших Location1 и 2 нужны Set Value As Vector.

Так же в данные функции нужно передать имя ключа. Для этого используется функция Make Literal Name.

Создали второго Empty Actor и передали в Location2 его координаты.

При нажатии на апостроф появляется дебаг-информация, относящаяся к ИИ.