

## Усовершенствование простого AI для игры на Unreal Engine 4

*Беликов Андрей Геннадьевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### **Аннотация**

В данной статье был рассмотрен процесс добавления новых умений AI для игры на Unreal Engine 4. В исследовании применялась программа Unreal Engine 4. В результате работы был усовершенствован простой AI.

**Ключевые слова:** Unreal Engine 4, 3D, геймдев

## **Enhancing Simple AI for Unreal Engine 4 Play**

*Belikov Andrey Gennadievich*

*Sholom-Aleichem Priamursky State University*

*Student*

### **Abstract**

This article looked at the process of adding new AI skills to play on Unreal Engine 4. The study used the Unreal Engine 4 program. As a result, simple AI was improved.

**Keywords:** Unreal Engine 4, 3D, gamedev

В данной статье продолжена работа по созданию игры в жанре Side Scroller.

Цель данной статьи усовершенствование простого AI.

Для создания проекта была рассмотрена статья Н. А. Морошкин в которой автор пытается продемонстрировать и исследовать возможности программного обеспечения [1], изучена статья А. М. Бакунов, И. Л. Калитеня, А. С. Дворецкий, И. О. Мартыненко, В. О. которые рассмотрели инструмент создания игр Unreal engine [2], П. Д. Гришков, В. В. Кугуракова рассмотрели подходы для реализации методов по синхронизации действий человека и виртуального аватара [3], С. В. Дьяченко, Р. А. Ходикян изучили игровой движок unreal engine [4].

Для начала создадим препятствия для AI (Рисунок 1-2).



Рисунок 1. Стена



Рисунок 2. Куча мусора

После переходим в папку AI и запускаем MyAiController (Рисунок 3-5).

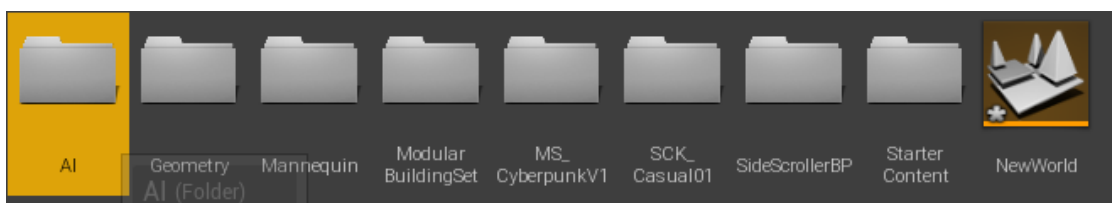


Рисунок 3. Папка AI

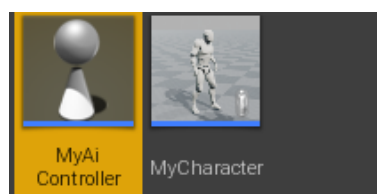


Рисунок 4. MyAiController

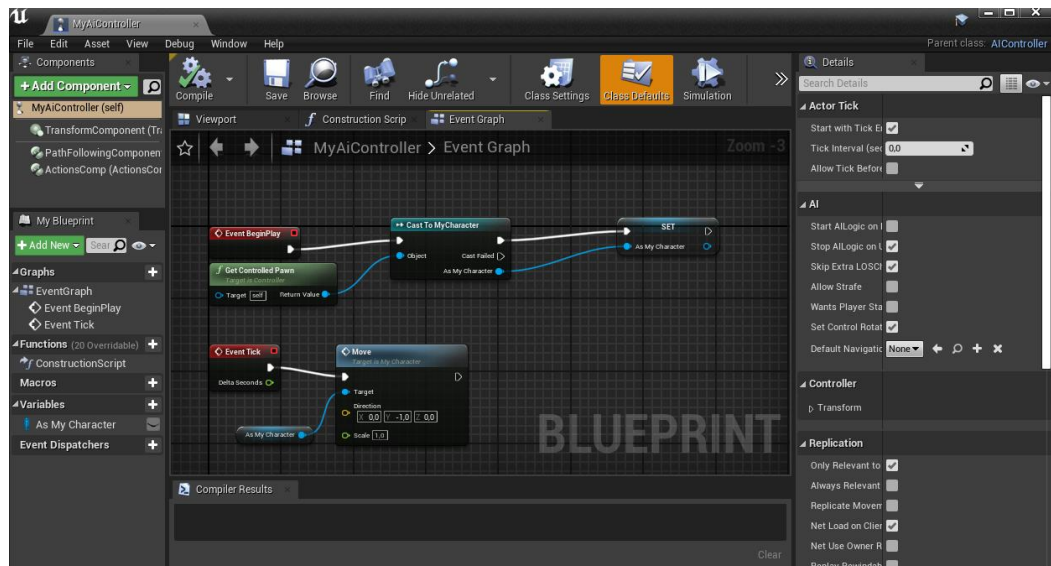


Рисунок 5. Настройки MyAiController

Меняем значение Scale на 0,35 в блоке Move, для умеренной ходьбы манекена (Рисунок 6-7).

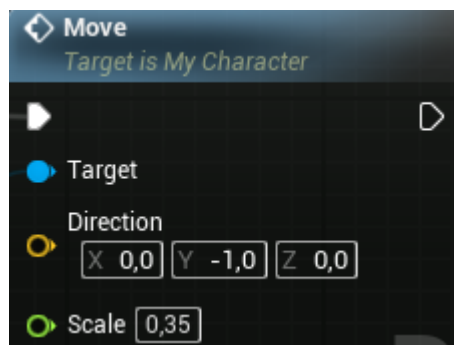


Рисунок 6. Блок Move



Рисунок 7. Ходьба манекена

С помощью нажатия ЛКМ по соединению Event Tick и Move используем Break All Pin Link(s) разрываем соединение (Рисунок 8).

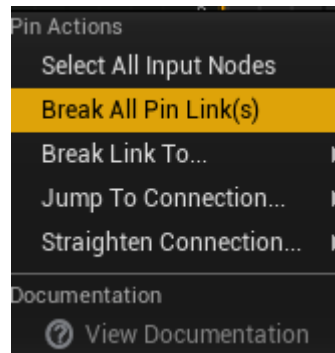


Рисунок 8. Break All Pin Link(s)

Далее перетягиваем As My Character и соединяем с Move (Рисунок 9).

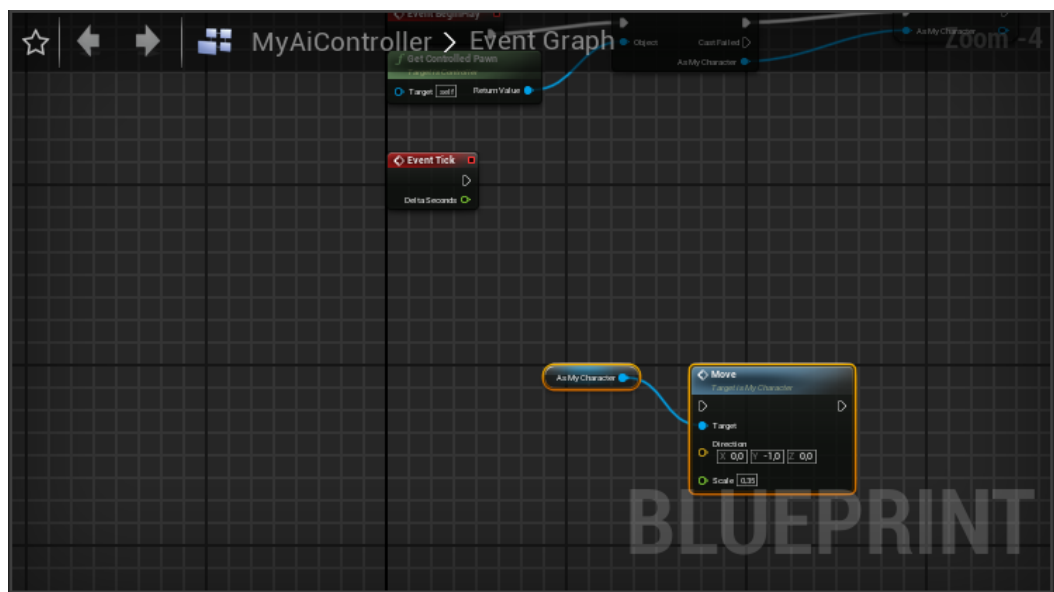


Рисунок 9. As My Character и Move

Далее тянем Event Tick и добавляем Sequence (Рисунок 10).

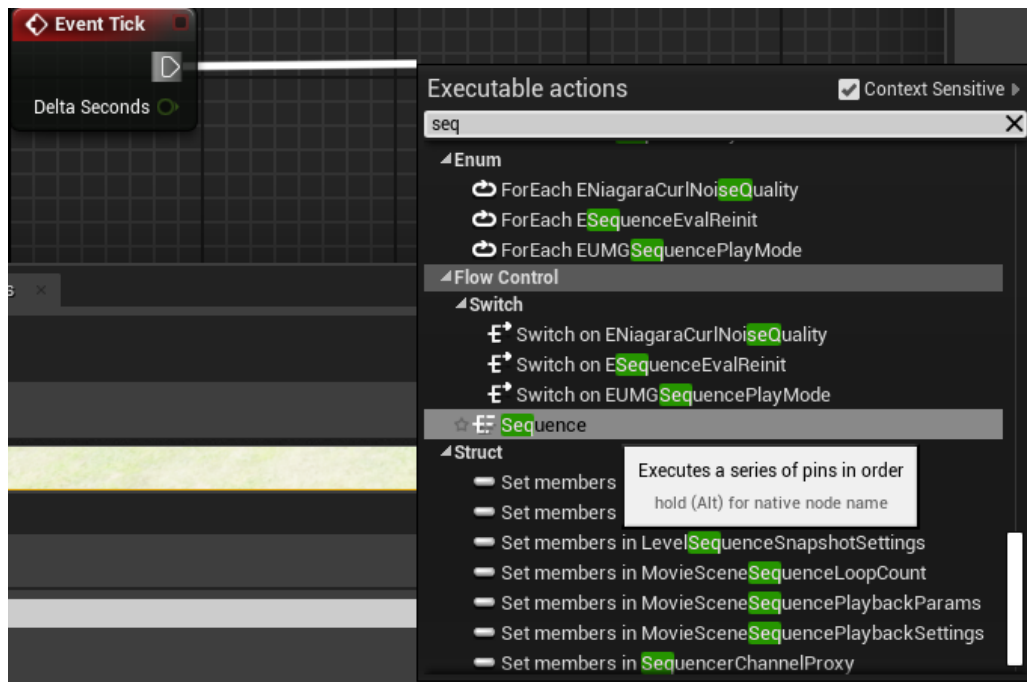


Рисунок 10. Sequence

Далее тянем Then 0 и добавляем MultiLineTraceByChannel (Рисунок 11).

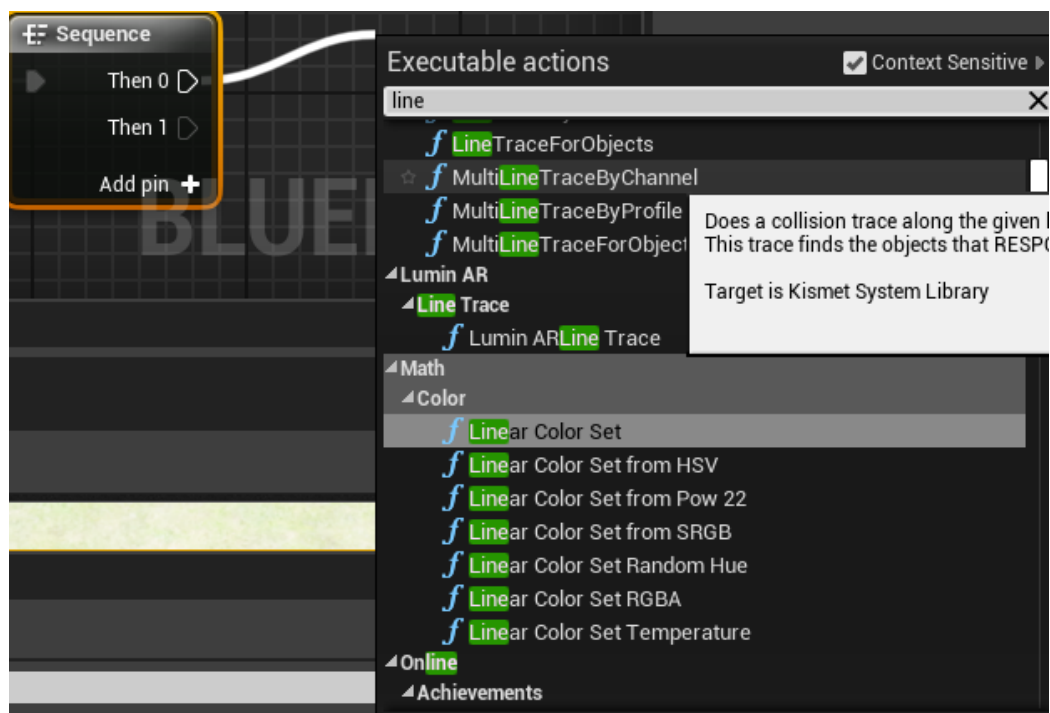


Рисунок 11. MultiLineTraceByChannel

Добавляем As My Character (Рисунок 12).

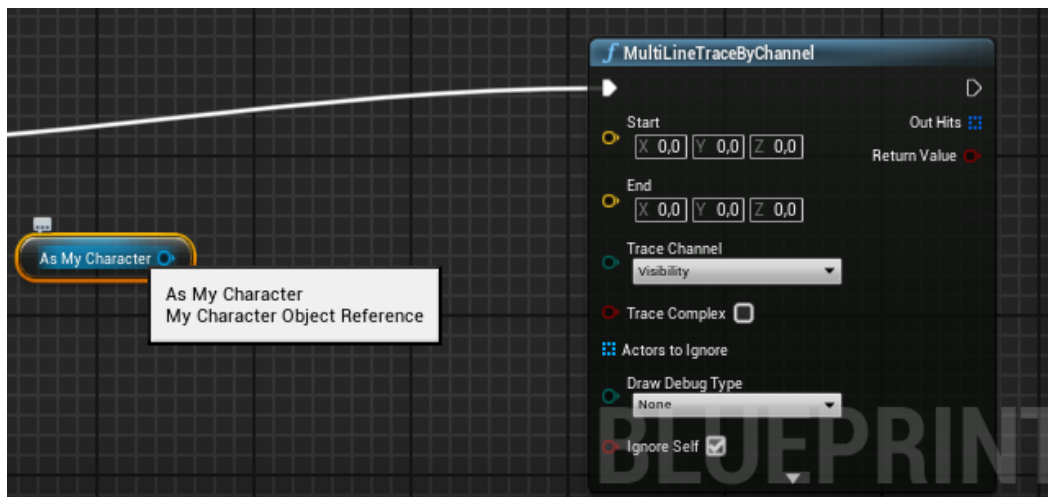


Рисунок 12. Добавление Direction

Добавляем GetActorLocation (Рисунок 13).

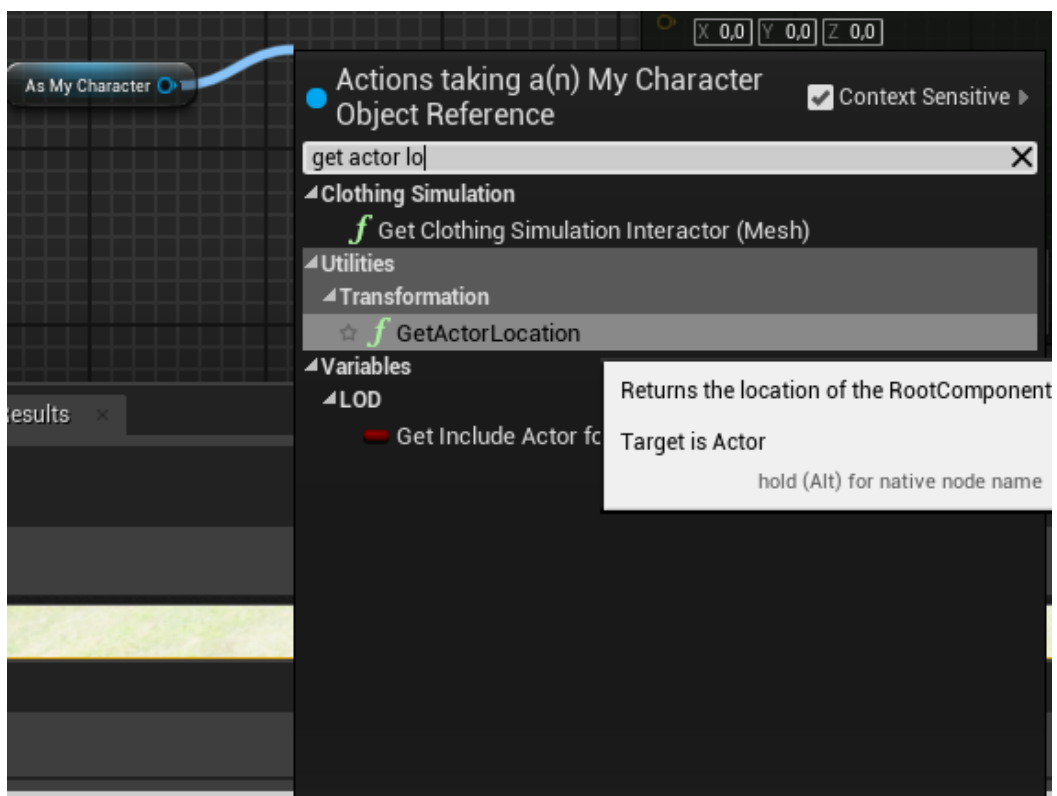


Рисунок 13. GetActorLocation

Соединяем GetActorLocation с MultiLineTraceByChannel (Рисунок 14).



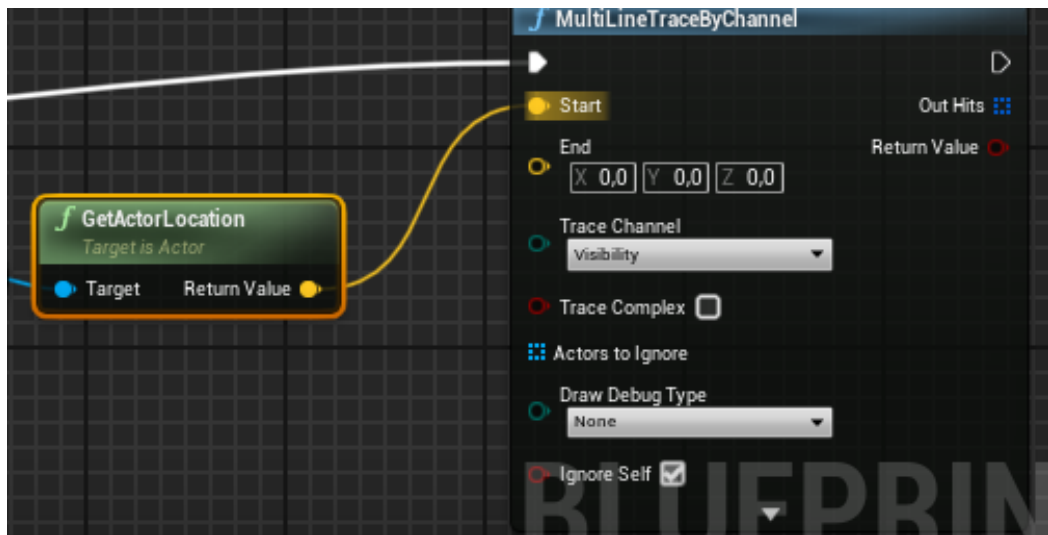


Рисунок 14. Добавление Add Movement Input

Добавляем Get Actor Forward Vector (Рисунок 15).



Рисунок 15. Get Actor Forward Vector

Добавляем Vector \* float (Рисунок 16).

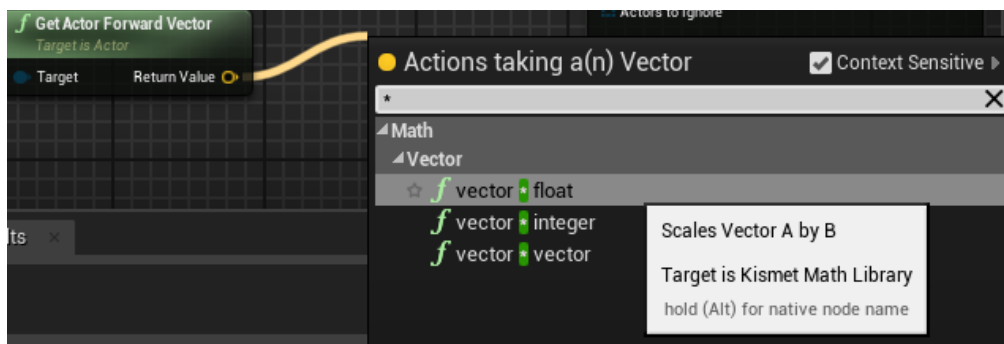


Рисунок 16. Добавление Scale

Выставляем значение 300, что значит расстояние до того как начнёт поворачиваться манекен (Рисунок 17).

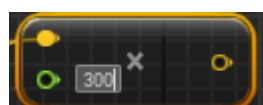


Рисунок 17. Настроенный Vector \* float

Добавляем Vector + vector (Рисунок 18).

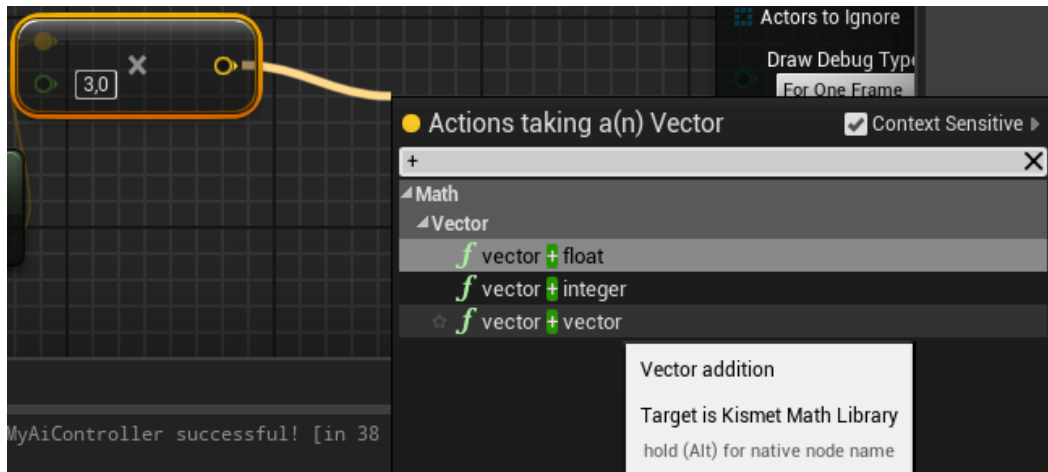


Рисунок 18. Добавление Get Controlled Pawn

Соединяем как на рисунке (Рисунок 19).

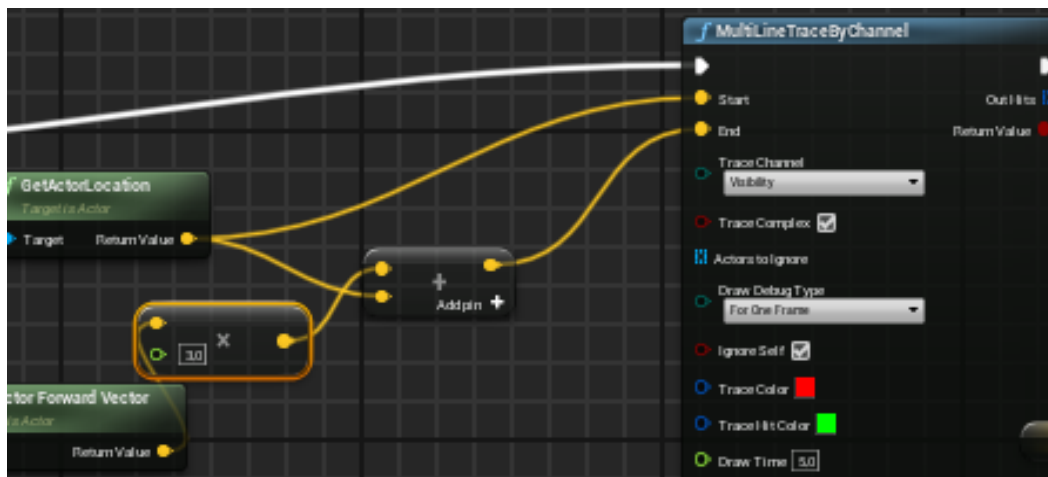


Рисунок 19. Готовая схема

Соединяем MultiLineTraceByChannel с Branch (Рисунок 20).

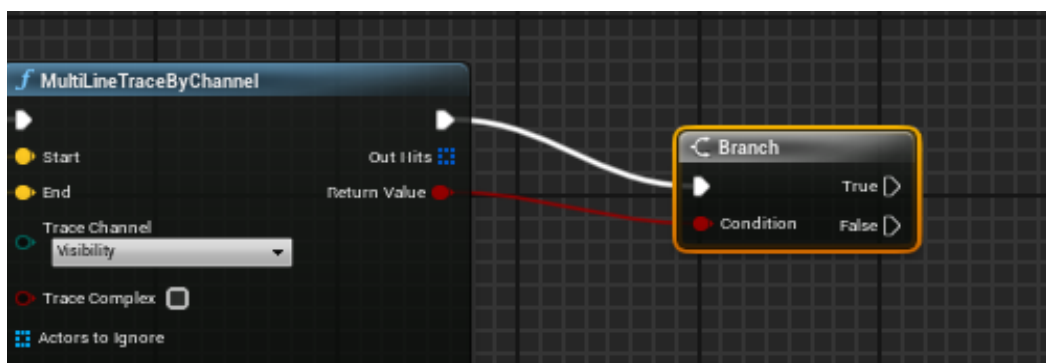


Рисунок 20. Соединённая схема

В меню Variables добавляем Direction и настраиваем (Рисунок 21-22).



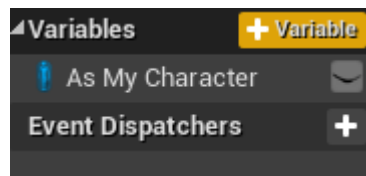


Рисунок 21. Добавление Direction

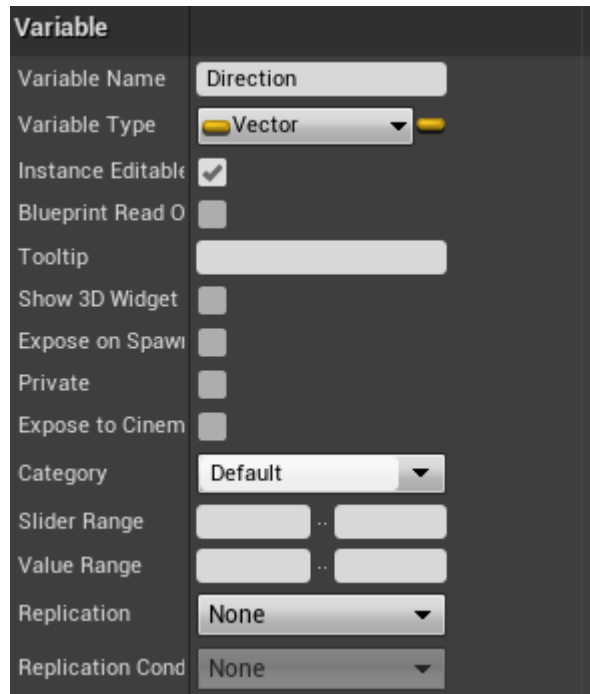


Рисунок 22. Добавленный As My Character

Добавляем Direction перетаскивая на свободную область и выбираем Set Direction (Рисунок 24).



Рисунок 23. Set Direction

Соединяем Branch и Set (Рисунок 24).



Рисунок 24. Соединённая схема

Добавляем Get Direction и соединяем с RotateVector и настраиваем угол на который будет поворачиваться манекен (Рисунок 25-26).

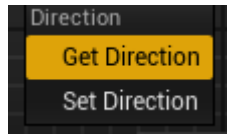


Рисунок 25. Get Direction

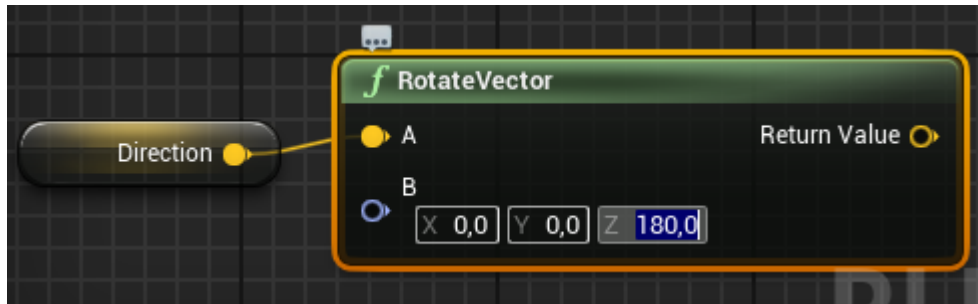


Рисунок 26. Настройка RotateVector

Соединяем RotateVector и Set (Рисунок 27).

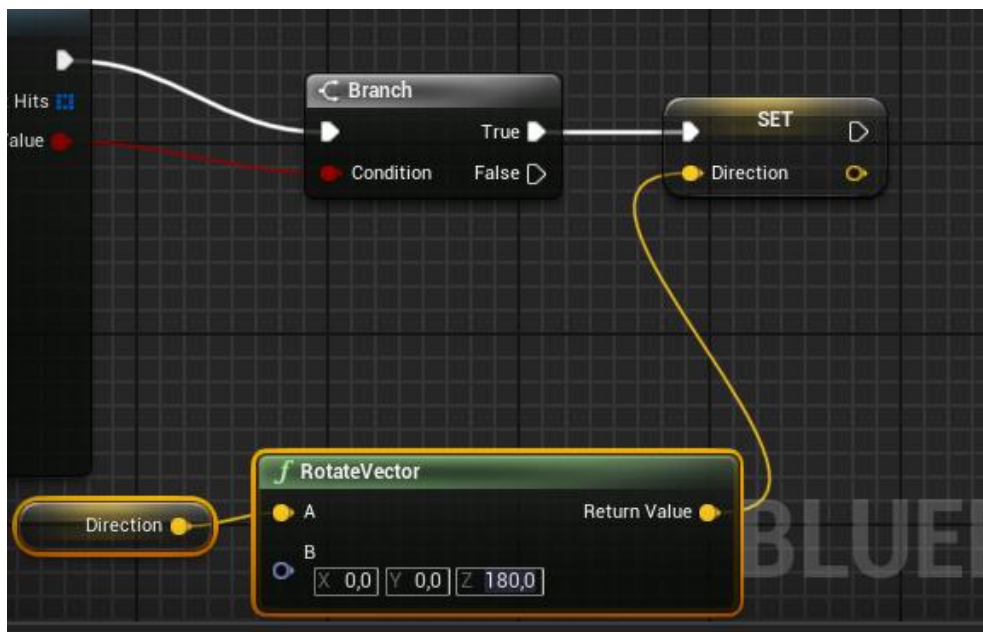


Рисунок 27. Соединённая схема

Добавляем Get Direction и соединяем с Move (Рисунок 28).

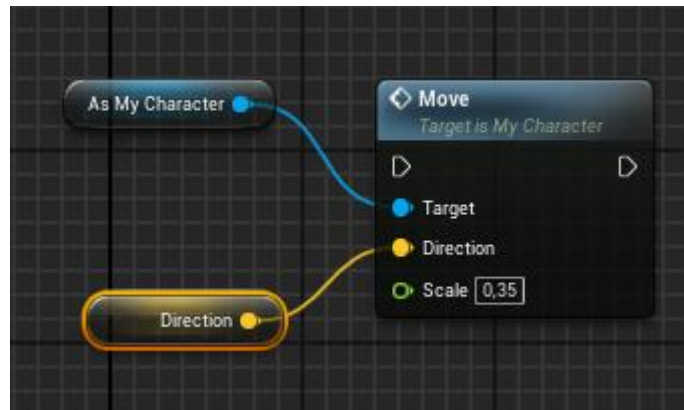


Рисунок 28. Соединённая схема

Соединяем Move и Then 1 (Рисунок 29).



Рисунок 29. Соединённая схема

В настройках Direction настраиваем начальное направление манекена (Рисунок 30).

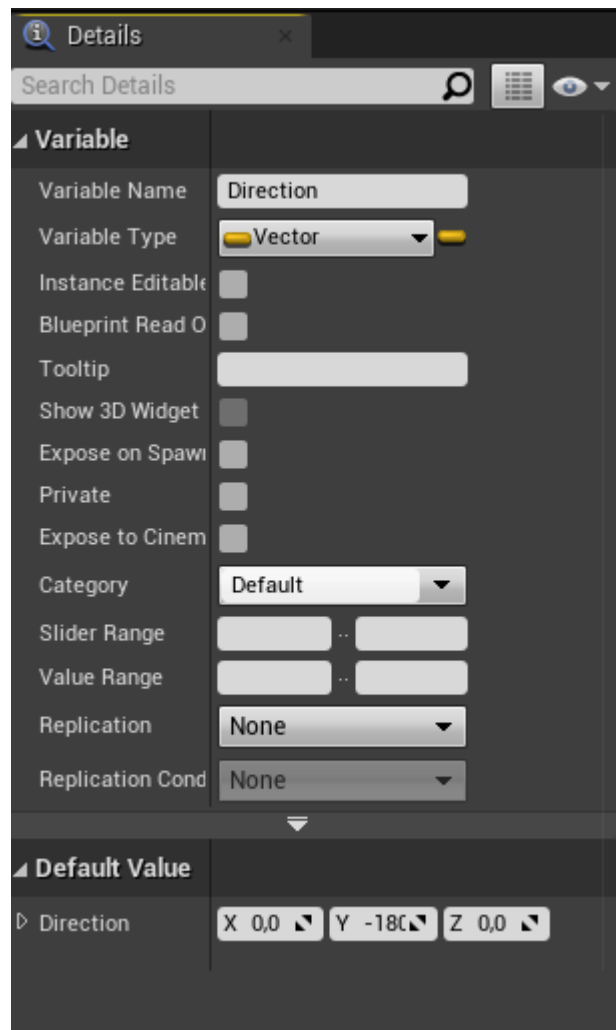


Рисунок 30. Настройка Direction

После компиляции видим, что манекен начинает ходить из стороны в сторону (Рисунок 31-32).



Рисунок 31. Движение AI



Рисунок 32. Движение AI

В данной статье был показан процесс усовершенствования простого AI для игры в жанре Side Scroller в среде Unreal Engine 4.

### Библиографический список

1. Морошкин Н. А. Исследование применения игровых движков в сфере кинематографа на примере unreal engine 4 // Моя профессиональная карьера. 2020. С. 90-98.
2. Бакунов А. М., Калитеня И. Л., Дворецкий А. С., Мартыненко И. О., Гимик В. О. Применение unreal engine в разработке игр // Web of Scholar. 2018. С. 26-28.
3. Гришков П. Д., Кугуракова В. В. Синхронизация движений игрока и виртуального аватара // Электронные библиотеки. 2018. С. 323-337.
4. Дьяченко С. В., Ходикян Р. А. Игровой движок unreal engine 4 // В сборнике: Научные меридианы. 2016. С. 210-213.