

УДК 004.42

Д. Ю. Бычковский, Ф. Н. Абу-Абед
**РАЗРАБОТКА ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ИМИТАЦИИ
ГОРОДСКОГО НАСЕЛЕНИЯ СРЕДСТВАМИ ГРАФИЧЕСКОГО
ДВИЖКА UNREAL ENGINE 4**

Тверской государственной технической университет, Тверь, Россия

В статье рассмотрен принцип реализации искусственного интеллекта имитирующего поведение городского населения, разработанного для применения в качестве пешеходов в симуляторе вождения средствами графического движка Unreal Engine 4. Рассмотрен принцип работы деревьев поведения в данном графическом движке и их основных составных частей, с помощью которых составляется логика поведения. Приведена внутренняя структура класса искусственного интеллекта пешехода и методы реализации определения объекта в поле зрения и слуха, используя встроенные средства разработки в графическом движке Unreal Engine 4. Показан принцип работы с системой запросов к окружению (Environment Query System), которая входит в состав рассматриваемого графического движка. Приведены визуализации результатов выполнения реализованных запросов, которые используются в процессе работы дерева поведения виртуальных пешеходов. Так как в разрабатываемом симуляторе вождения необходимо добиться высокой частоты смены кадров и малой требовательности к ресурсам вычислительной системы, в статье предложено оптимальное решение для имитации большого числа пешеходов в виртуальном городе. Показано дерево поведения и описан принцип работы основных частей реализованного искусственного интеллекта имитации городского населения в виде блок-схем с пояснениями.

Ключевые слова: Unreal Engine 4, Environment Query System, деревья поведения, искусственный интеллект, симулятор вождения.

Введение. В современных городах страны значительно повысилась интенсивность движения и аварийность на дорогах, ввиду чего к участникам дорожного движения предъявляются довольно высокие требования. Это усложняет процесс обучения в автошколах России и добавляет необходимость в значительном увеличении часов практических занятий.

Чтобы дать начальный уровень практики начинающим водителям и заложить представление о вождении реального автомобиля, в автошколах применяют симуляторы вождения. Таким образом, обучение вождению начинающих водителей с помощью тренажеров учит правильно пользоваться средствами управления автомобиля, исключая возможность повреждения транспортного средства или нанесения вреда другим участникам движения.

В данной статье рассматривается практический пример использования средств графического движка Unreal Engine 4 (далее UE4), в качестве

инструмента реализации искусственного интеллекта, имитирующего городское население в разрабатываемом симуляторе вождения.

ИИ городского населения являются частью проекта разработки симулятора вождения с применением технологий виртуальной реальности (далее VR) и необходим для воспроизведения случайных ситуаций при вождении автомобиля в городских условиях с участием пешеходов.

Целью разработки является реализация максимально правдоподобной модели поведения ИИ городского населения с помощью деревьев поведения, работающей самостоятельно с момента появления в виртуальном мире и автоматически реагирующей на изменения обстановки вокруг, а также затрачивающего минимум ресурсов вычислительной мощности компьютера.

Внутренняя структура класса пешехода. Перед тем, как начать говорить о реализации отдельных частей ИИ, следует пояснить внутреннее устройство и принцип работы системы ИИ в UE4. В выбранном графическом движке используется компонентный подход программирования, а также присутствует большое количество инструментальных средств, доступных «из коробки». Данный графический движок поставляется бесплатно и с полным исходным кодом, что дает возможность модифицировать его составные части как угодно, а также позволяет исправлять ошибки самолично, не дожидаясь обновлений от производителя.

В качестве языка разработки в UE4 используется C++. Также в UE4 присутствует собственный инструмент реализации алгоритмов с помощью языка визуального скриптинга «Blueprint» [1], который представляет собой полноценную систему сценариев, основанную на концепции использования интерфейса на основе узлов для создания элементов игрового процесса внутри Unreal Editor. Как и многие распространенные языки сценариев, он используется для определения объектно-ориентированных классов или объектов в графическом движке.

На Рисунке 1 показана схема основных компонентов, использованных при разработке класса пешехода.



Рисунок 1 - Структура класса ИИ пешехода

Компонент движения реализует перемещение в трехмерном пространстве по навигационным зонам. Корневая коллизия используется в нем для определения маршрутов обхода препятствий или других пешеходов. Компонент движения реализует перемещение ИИ по земле и корректирует маршрут в реальном времени. Корневая коллизия также требуется для определения пересечений или столкновений с объектами.

Контроллер применяет логику поведения к классу ИИ. Внутри находится компонент мозга, содержащий дерево поведения (Behavior Tree) [2], внутри которого описана логика работы ИИ в древовидной структуре. За память отвечает Black Board [3] - доска переменных, которые используются в процессе работы деревьев.

За компонент зрения и слуха отвечает AI Perception [4]. На основе него в ИИ построена логика определения опасности, чтобы пешеходы убегали от быстро приближающегося или сигнализирующего автомобиля.

В проекте также присутствуют классы пешеходных переходов и светофоров. Их внутренняя реализация проста и сводится к запрету или разрешению на переход дороги для ИИ, находящихся в зоне их действия.

Экспериментальная конструкция, материалы и методы.

Опуская процесс настройки скриптов анимаций, перейдем к процессу написания логики поведения. Необходимо реализовать обработчики событий обнаружения автомобиля в радиусе поля зрения и слуха, столкновения автомобиля с ИИ, пересечения зоны действия светофора.

Удар и пересечения с объектами определяет корневая коллизия в классе ИИ. UE4 генерирует события при определении пересечения или удара, в котором передает объект, с которым было произведено взаимодействие. Получив удар от автомобиля, ИИ перестает работать, а его трехмерная модель начинает процесс симуляции физического тела, что в конкретном случае сопровождается отбрасыванием тела от автомобиля на землю. Расчеты симуляции физики производятся встроенным в UE4 физическим движком.

Корневая коллизия обрабатывает события пересечения с объектами с целью определения нахождения ИИ на перекрестке. Когда ИИ по пути своего следования заходит в зону действия светофора перекрестка, он записывает его в доску значений дерева поведения. Если ИИ выходит из зоны действия светофора, значение стирается.

Одной из задач разработки является реализация определения опасности для ИИ. Пешеход должен стремиться избежать удара с автомобилем. Для этого используется компонент AI Perception, используемый для обнаружения объектов вокруг.

ИИ проверяет, слышен ли в радиусе звуковой сигнал автомобиля или виден ли быстродвигающийся автомобиль. При обнаружении такого автомобиля ИИ записывает в доску значений информацию об этом, а спустя 3 секунды стирает эти данные и возвращается к нормальному режиму работы.

Деревья поведения. В графическом движке UE4 ИИ реализуется при помощи деревьев поведения. Доска значений (Black Board) используется для хранения данных, необходимых для работы. Каждый узел отдает результат выполнения в вышестоящие узлы на ветви. Ветви выполняются слева на право. Чем правее ветвь, тем ниже у нее приоритет на выполнение.

Деревья поведения в UE4 имеют два основных составных узла (узла принятия решений):

- Selector (далее селектор)[5] - узел в котором переходы по дочерним элементам будут производиться до тех пор, пока один из них не вернет успешный результат выполнения.
- Sequence (далее секвенция)[5] - узел в котором переходы по дочерним элементам будут выполняться до тех пор, пока один из них не завершится с ошибкой.

Задачи в деревьях поведения UE4 имеют 3 состояния:

- Выполняется.
- Завершена с ошибкой.
- Завершена успешно.

Условия (Decorator) [6] прикрепляются к задачам или узлам принятия решений и определяют можно ли выполнить ветвь или узел в данный момент. Условия в UE4 могут прерывать выполнения ветви, на которой они закреплены или ветвей с более низким приоритетом при изменении результата выполнения.

Помимо описанного выше, в деревьях поведения UE4 имеются службы (Service) [7]. Они подключаются к составным узлам дерева и вызываются с заданной частотой, пока выполняется ветвь, на которой находится служба. Чаще всего они используются для проверки и обновления данных, записанных в доску значений, и заменяют традиционные параллельные узлы, используемые в других системах деревьев поведения.

В UE4 также имеются встроенная система запросов к окружению (Environment Query System - далее EQS) [8]. Ее часто используют для написания более корректного определения точек перемещения ИИ, подходящих под заданный набор условий. Это одна из функций системы

ИИ в UE4, используемая для сбора данных об окружении, которые запрашиваются с при помощи тестов.

При вызове генерируется набор элементов от точки, в которой находится вызывающий объект, одним из предложенных системой методом в заданном радиусе. Для выбора одной из этих точек, на генератор добавляют тесты. Они могут работать как фильтры данных или влиять на коэффициент соответствия условию теста. Также они могут работать одновременно в двух этих режимах. Результатом выполнения является один элемент, который наилучшим образом соответствует заданным условиям. На Рисунке 2 представлено дерево поведения ИИ пешехода.

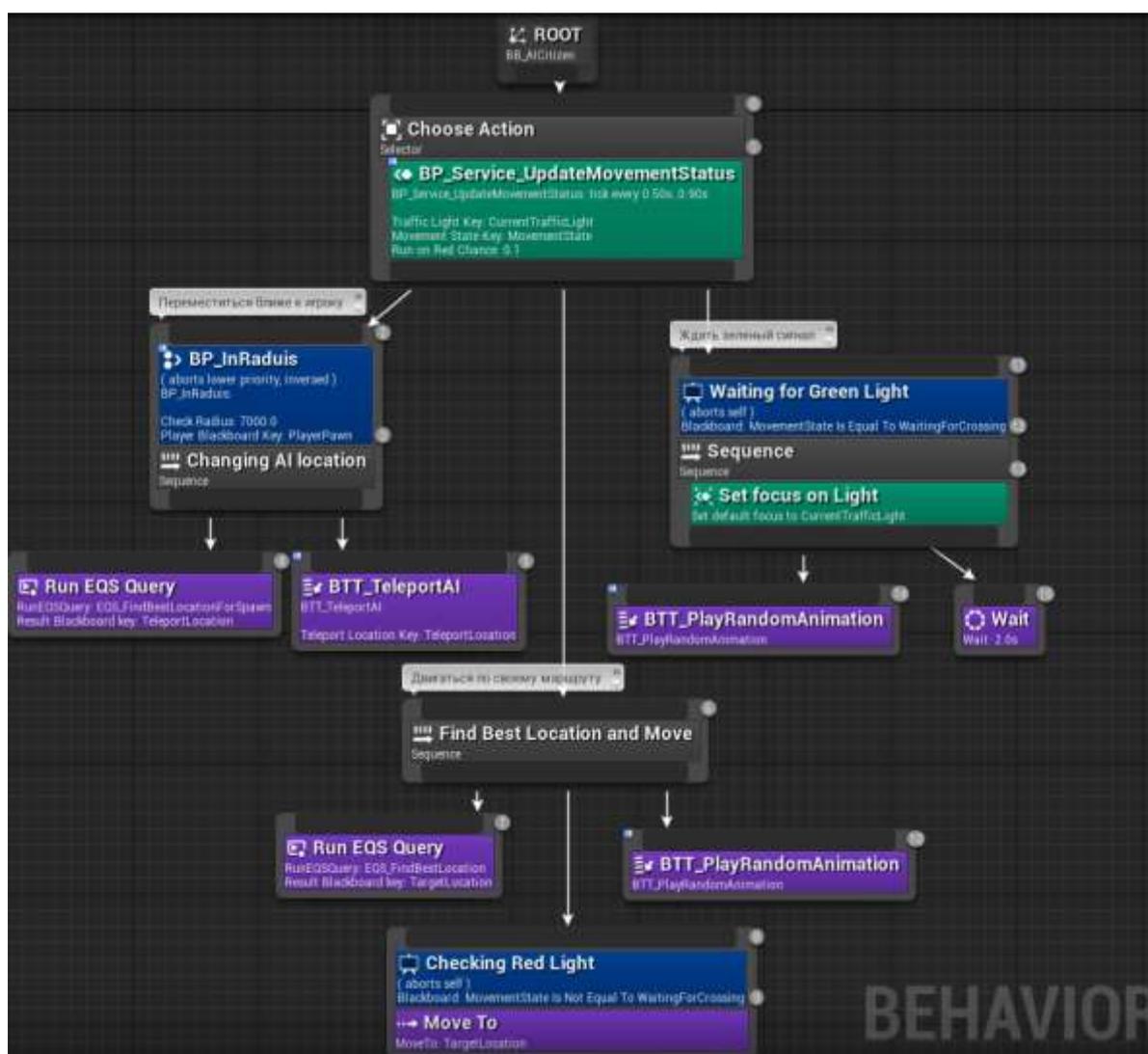


Рисунок 2 - Дерево поведения ИИ пешехода

Рассмотрим работу дерева последовательно по всем узлам. Первый узел после корня — это селектор, на котором находится служба. Задача

этого узла заключается в выборе одного из трех наборов поведения ИИ, с параллельным контролем за перемещением ИИ на перекрестках и проезжей части. Служба использует ключи из доски значений, в которых записаны текущий отслеживаемый светофор и режим перемещения ИИ. В процессе ее работы происходит обновление данных на доске значений, чтобы останавливать ИИ на красный сигнал светофора и избегать опасность. Служба также принимает параметр вероятности перехода дороги на красный свет, который отменяет логику ожидания зеленого сигнала для конкретной ИИ. На Рисунке 3 представлен алгоритм работы службы.

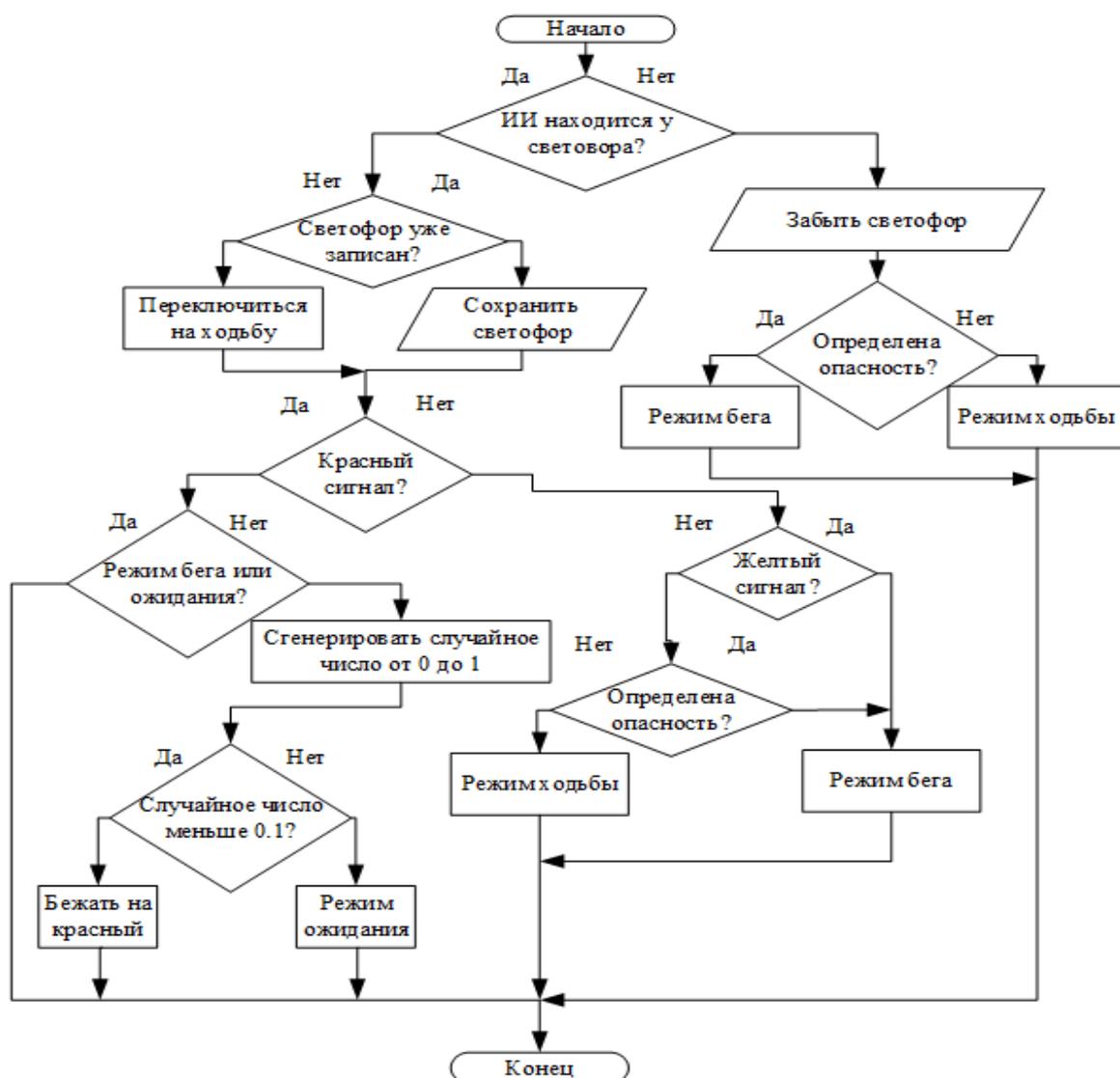


Рисунок 3 - Служба контроля перемещения ИИ

Данная служба выполняется на протяжении всего цикла работы ИИ. В случае нахождения ИИ в зоне действия светофора проверяется текущий

световой сигнал. Находясь на проезжей части в последние секунды зеленого сигнала, ИИ начнет перебежать дорогу. Зайдя в зону действия светофора во время активного красного светового сигнала, ИИ переключится в ожидание зеленого светового сигнала, но с вероятностью 10% может принять решение перебежать дорогу на красный сигнал. Подобное необходимо, чтобы обучаемый на тренажере начинающий водитель также тренировал свою внимательность и реакцию.

Если в доске значений указано, что ИИ определил опасность, то служба переключит режим перемещения на бег, чтобы как можно скорее убрать его с проезжей части и попытаться избежать удара от автомобиля.

Следующий узел в дереве поведения — это секвенция, введенная для имитации большого количества жителей в городе. В целях экономии ресурсов системы, в симуляторе используется малое количество горожан на всю площадь виртуального города. Задача этой ветви дерева заключается в перемещении ИИ ближе к автомобилю. Таким образом, используя около 20 горожан, имитируется оживленность города, так как пешеходы автоматически переносятся ближе к виртуальному автомобилю в процессе работы дерева поведения. Декоратор допускает выполнение последовательности задач только когда автомобиль отделился от ИИ на большое расстояние. Данное условие прерывает выполнение всех ветвей правее, когда автомобиль уезжает слишком далеко от ИИ.

В секвенции используется EQS-запрос, генерирующий по направлению движения игрока набор точек в конусе. С помощью тестов в запросе фильтром отсекаются все точки, до которых невозможно проложить маршрут, и которые находятся вне ограничения по расстоянию. Наилучший вариант среди оставшегося набора выбирается случайным образом. Результат работы данного запроса представлен на Рисунке 4.



Рисунок 4 - Визуализация работы EQS-запроса поиска точки перемещения ИИ

На рисунке видно, что точки, находящиеся на проезжей части вне пешеходных зон отсечены, и выборка будет производиться из дальних результатов. После выбора точки перемещения выполнится узел секвенции, расположенный правее, который реализует мгновенное перемещение ИИ.

Следующий узел в дереве поведения — это секвенция, выполняющая логику выбора конечной точки маршрута и перемещения ИИ к этой точке. Выбор точки назначения производится с помощью EQS-запроса, схожего с предыдущим.

Данный запрос генерирует из местоположения ИИ сетку на определенное расстояние. С помощью тестов отсекаются все точки, до которых невозможно проложить маршрут и которые находятся ближе заданного расстояния. Наилучшим результатом будет самый отдаленный. Результат работы данного запроса представлен на Рисунке 5.



Рисунок 5 - Визуализация работы EQS-запроса поиска точки назначения движения ИИ

Логика работы данного EQS-запроса схожа с рассмотренным ранее. В этом случае ИИ генерируют набор точек во все стороны вокруг себя и выбирают наиболее подходящую точку на расстоянии не менее заданного значения.

За этим запросом в последовательности следует задача «Move To», с помощью которой ИИ начинает движение в сторону выбранной в EQS-запросе точки. Пока ИИ двигается к заданной точке, службой проверяется не находится ли он в зоне действия светофора, а условие, установленное на задаче движения, прервет ее выполнение в случае, когда пешеход начнет ожидать зеленый сигнал светофора.

Крайняя правая ветвь реализует логику ожидания зеленого сигнала светофора. Виртуальный пешеход будет проигрывать случайную анимацию во время ожидания зеленого сигнала светофора или по достижению точки назначения движения.

Заключение. В результате разработки ИИ городского населения с помощью инструментальных средств, входящих в состав графического движка UE4, удалось реализовать малотребовательный к ресурсам компьютера ИИ, позволяющий создать эффект густонаселенного города, при малом фактическом количестве пешеходов в пространстве, что положительно сказывается на производительности системы.

ИИ реагирует на сигналы светофора, понимая, когда необходимо двигаться быстрее, и когда нужно ожидать. Также благодаря компоненту зрения и слуха, удалось реализовать определение опасности ИИ и методы избегания столкновения с автомобилем. Со случайной вероятностью ИИ может поменять направление следования и пойти в обратную сторону. С вероятностью 10% он может перебежать дорогу на красный свет, что заставит ученика быть всегда внимательным, управляя виртуальный автомобиль.

ЛИТЕРАТУРА

1. Blueprint визуальный скриптинг - Электронный ресурс [Режим доступа]. URL: <https://docs.unrealengine.com/latest/INT/Engine/Blueprints/GettingStarted/index.html>
2. Деревья поведения в Unreal Engine 4 - Электронный ресурс [Режим доступа]. URL: <https://docs.unrealengine.com/en-US/Engine/AI/BehaviorTrees/QuickStart/14>
3. Доска значений Black Board в Unreal Engine 4 - Электронный ресурс [Режим доступа]. URL: <https://docs.unrealengine.com/en-US/Engine/AI/BehaviorTrees/QuickStart/5>
4. Компонент зрения и слуха AI Perception в Unreal Engine 4 - Электронный ресурс [Режим доступа]. URL: <https://docs.unrealengine.com/en-us/Engine/Components/AI>
5. Селектор и секвенция в Unreal Engine 4 - Электронный ресурс [Режим доступа]. URL: <https://docs.unrealengine.com/en-US/Engine/AI/BehaviorTrees/QuickStart/10>
6. Условия в Unreal Engine 4 - Электронный ресурс [Режим доступа]. URL: <https://docs.unrealengine.com/en-US/Engine/AI/BehaviorTrees/QuickStart/13>
7. Службы в Unreal Engine 4 - Электронный ресурс [Режим доступа]. URL: <https://docs.unrealengine.com/en-US/Engine/AI/BehaviorTrees/QuickStart/11>

8. Система запросов к окружению EQS в Unreal Engine 4 - Электронный ресурс [Режим доступа]. URL: <https://docs.unrealengine.com/en-US/Engine/AI/EnvironmentQuerySystem/UserGuide>

D.Y. Bychkovskiy, F.N. Abu-Abed

DEVELOPMENT OF PEDESTRIAN ARTIFICIAL INTELLIGENCE UTILIZING UNREAL ENGINE 4 GRAPHIC ENGINE.

*Tver State Technical University,
Tver, Russia*

The article demonstrates the implementation of AI for pedestrian simulation in the driving simulator by means of Unreal Engine 4 graphic engine and VR-technologies. The authors review the operation principles of behavior trees and their components. The paper describes the internal structure of a pedestrian AI class and methods of implementing object detection in the field of view and audibility utilizing Unreal Engine 4. The authors give the example of using Environment Query System included in the engine AI system and display the result of executing its queries, which used in the process of the virtual pedestrian behavior tree simulation. Since in the developed driving simulator it is necessary to achieve a high frequency of frame changes and low demands on the resources of the computer system, the article suggests an optimal solution for simulating a large number of pedestrians in a virtual city. The article shows the behavior tree and represents operation algorithms of the pedestrian AI's basic components in block diagrams with annotations.

Keywords: Unreal Engine 4, Environment Query System, behavior tree, artificial intelligence, driving simulator.

REFERENCES

1. Blueprints. URL: <https://docs.unrealengine.com/latest/INT/Engine/Blueprints/GettingStarted/index.html>
2. Behavior tree. URL: <https://docs.unrealengine.com/en-US/Engine/AI/BehaviorTrees/QuickStart/14>
3. Black Board. URL: <https://docs.unrealengine.com/en-US/Engine/AI/BehaviorTrees/QuickStart/5>
4. AI Perception component. URL: <https://docs.unrealengine.com/en-us/Engine/Components/AI>
5. Selector and sequence. URL: <https://docs.unrealengine.com/en-US/Engine/AI/BehaviorTrees/QuickStart/10>
6. Decorators. URL: <https://docs.unrealengine.com/en-US/Engine/AI/BehaviorTrees/QuickStart/13>
7. Services. URL: <https://docs.unrealengine.com/en-US/Engine/AI/BehaviorTrees/QuickStart/11>
8. Environment Query System. URL: <https://docs.unrealengine.com/en-US/Engine/AI/EnvironmentQuerySystem/UserGuide>