**Методы взаимодействия с LLM для управления игровым агентом в трехмерном движке Unreal Engine 5**

**[#2. Постановка задачи]**

* Цель работы: Разработать игровую систему, моделирующую взаимодействие LLM-модели и игрового агента.

Задачи:

* + Создать 3D-сцену с объектами, включающие себя различные параметры;
  + Разработать механизм приёма/передачи сообщений между LLM-моделью и игровым агентом/окружающим миром;
  + Создать интерфейс с возможность выбора игровых блоков ЛКМ;
  + Создать управляемое LLM дерево поведения агента.

**[#3. Актуальность]**

Актуальность работы заключается в следующем:

* + LLM является универсальным инструментом, позволяющим сократить написание кода;
  + На рынке представлено мало решений для интеграций LLM, управляющих какими-либо функциями внутри Unreal Engine 5;
  + Данная работа экспериментальная и позволяет исследовать оптимальность использования LLM в Unreal Engine 5.

**[#4. Принцип работы LLM]**

Принцип работы таких моделей достаточно прост:

* Модель получает на вход «промпт» (слово или набор слов) и далее подбирает наиболее подходящее следующее слово.
* После этого полученная строка вновь подается на вход модели и она подбирает еще одно слово.
* И так далее.

Так получается «разумное продолжение» на основе изначального запроса. Для пользователя это выглядит как ответ, который имеет смысл.

**[#5. Разработка игрового поля]**

Игровое поле представляет собой поле кубов 15х15. Его созданием занимается объект ASpawner. В него заранее вводится многострочная переменная WorldMap с символами, отвечающие за тип блока. Например, f – это лес, n – это гнездо. Если ввести некорректный символ, блок не появится – таким образом можно составлять различные миры произвольной формы.

**[#6. Игровой агент]**

|  |  |
| --- | --- |
| Характеристики   * Здоровье (0-100) * Сытость (0-150) * Усталость (0-100) * Температура | Функции  go\_near\_<area> - идти в ближайшее место определённого типа  go\_random – идти наугад в любое место  go\_maxforest – идти в лес с наибольшим числом пищи.  eat – охота.  rest – идти в гнездо |

**[#7. Правила игры]**

Здоровье уменьшается если усталость < 0, или температура > 40 или температура < 15 или сытость < 0

В гнезде восстанавливается здоровье и усталость. Восстанавливать сытость можно только находясь в лесу. Если здоровье меньше 0, агент уничтожается.

**[#8. Дерево поведения]**

Дерево поведения состоит из 4 групп состояний, обозначающие действия крокодила. Самое первое – idle, каждую секунду запрашивается действие у контроллера – если таково имеется. Go\_random включает в себя поиск точки, передвижение к ней и небольшую паузу. Go\_target – тоже самое только без поиска конечной цели. Eat – это 5 секундное действие с соответствующей анимацией. После каждого действия дерево возвращается в состояние idle.

**[#9 Взаимодействие с LLM]**

Взаимодействие начинается с вызова Get answer у BP\_Animal. Модели посылается запрос, в котором включены текущие параметры персонажа, его местоположение и подсказки. Затем запускается таймер на 20 сек., который запускается заново при получении нового токена. После получения ответа, данные фильтруются и контроллер персонажа получает действие, которое затем становится значением ключа Action у компонента BlackBoard. В зависимости от него выбирается группа действий из дерева поведения. Затем Action становится равным idle, и процесс начинается заново. Если агент не умер.

**[#10. Анимирование]**

В проекте используется один компонент Animation Blueprint – который переключает состояния бездействие, ходьба, плавание; и два объекта BlendSpace, отвечающие за скорость анимации в зависимости от скорости персонажа.

**[#11. Заключение]**

В результате работы были выполнены следующие задачи:

1. Создание игровой модели из ISMC кубов;

2. Создание персонажа с различными характеристиками;

3. Интеграция LLM модели llama-2-7b в UE 5;

4. Работа с анимацией и деревом поведения;