УДК 004

Реализация механики передвижения по клику в Unity

Ульянов Егор Андреевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье рассматривается и описывается реализация механики передвижения персонажа по клику мышки при помощи стандартных средств Unity. Реализация механики осуществляется посредством карты навигации. Практическим результатом является реализованная механика.
Ключевые слова: Unity, управление, стратегия

Implementation of the mechanics of movement by click in Unity

Ulianov Egor Andreevich
Sholom-Aleichem Priamursky State University
Student

Abstract
This article discusses and describes the implementation of the mechanics of character movement by mouse click using standard Unity tools. The mechanics are implemented by means of a navigation map. The practical result is implemented mechanics.
Keywords: Unity, movement, strategy

Игроки склонны ожидать определенных типов движений в определенных типах игр. Если запускаете игру жанра платформер, ожидается, что сможете бегать и прыгать благодаря “стрелочкам”. Если запускаете стратегию, ожидается управление при помощи мыши. Так что, при разработке игры, неплохо бы знать эти правила управления.

Цель данной статьи рассмотреть возможности игрового движка Unity 3D в создании реалистичного освещения в 2D играх.

А.А. Кабанов в своей статье описал исследование процесса создания игровой графики. Близость игровой графики и архитектуры дизайна [1]. С.А.Суродин в своей статье представил сценарий углубленного изучения одного из лучших движков, существующих на данный момент, для создания красивых 2D и 3D игр [2]. В своей работе Р. Ф. Гайнуллин, В. А. Захаров, Е.А. Аксенова изучили инструмент для разработки двух- и трёхмерных игр – Unity 3D [3]. К. В. Богданов, П. Р. Михеев, И. Н. Суворов в своей работе описали развитие игровых движков, а именно провели обзор от примитивной графики до высокоуровневых инструментариев [4].

Начинаем реализацию механики с создания проекта и добавления куба в роли игрока см. рисунок 1.

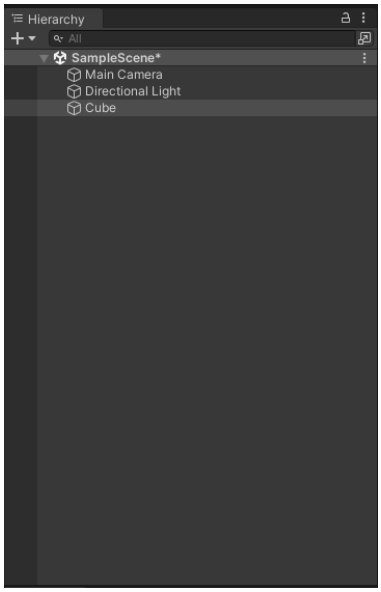

Рис. 1. Добавление игрока

Сразу добавим к кубу компонент «Nav Mesh Agent», который позволит указывать игроку пункт назначения см. рисунок 2.

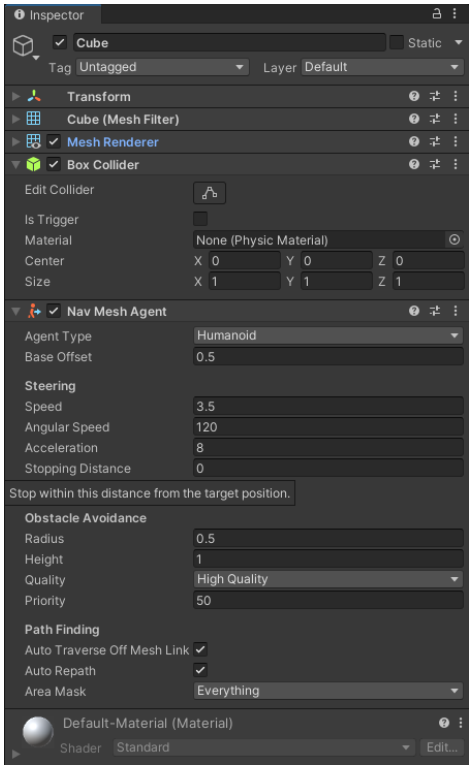

Рис. 2. Добавление компонента «Nav Mesh Agent»

Далее добавляем на сцену плоскость, по которой в дальнейшем будет двигаться куб см. рисунок 3.

Рис. 3. Добавление плоскости

Под настроем сцену: растянем плоскость, поднимем куб и расположим камеру повыше см. рисунок 4.

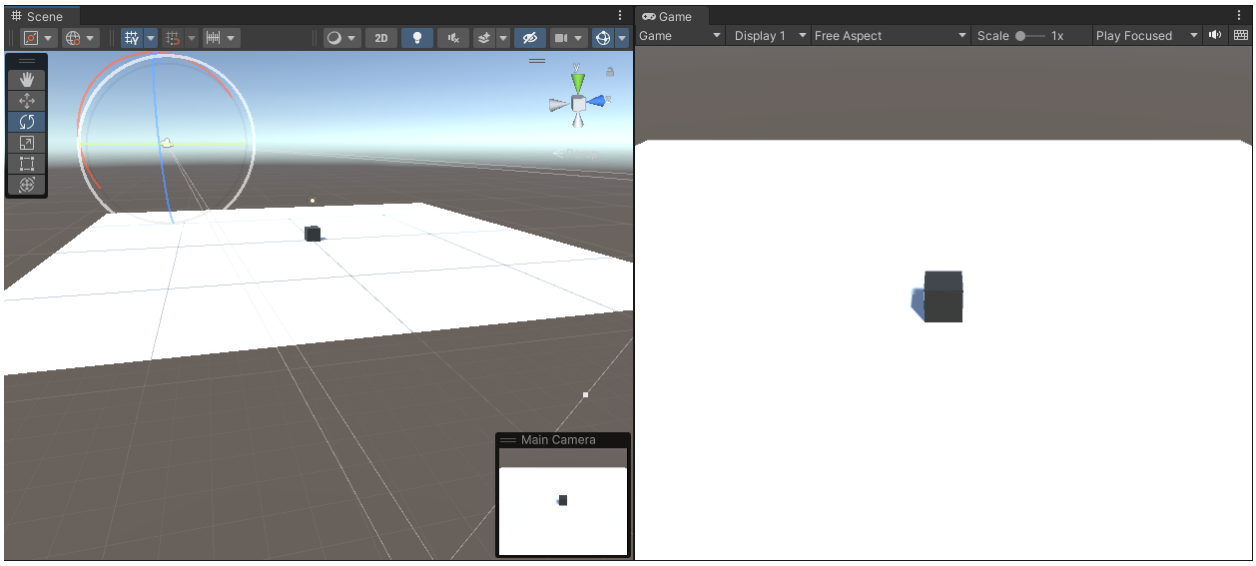

Рис. 4. Настройка объектов сцены

Для задания границ карты и зон перемещения, в разделе «Window» откроем пункт «AI» и выберем «Navigation». В панели «Navigation» выбираем “запечь” и нажимаем соответствующую клавишу см. рисунок 5-7.

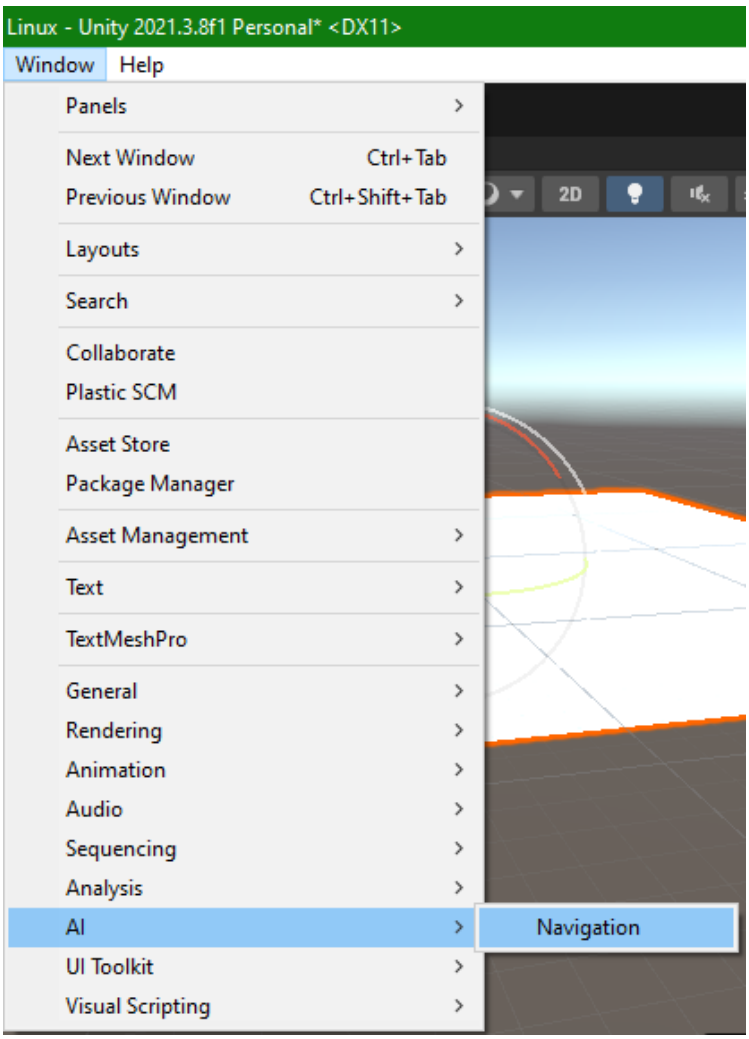

Рис. 5. Использование инструмента «Navigation»

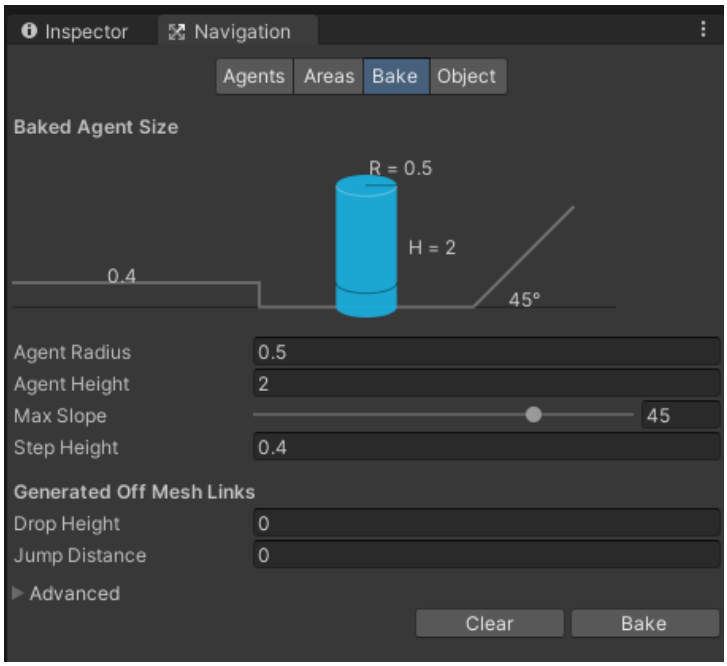

Рис. 6. Раздел “запечки”

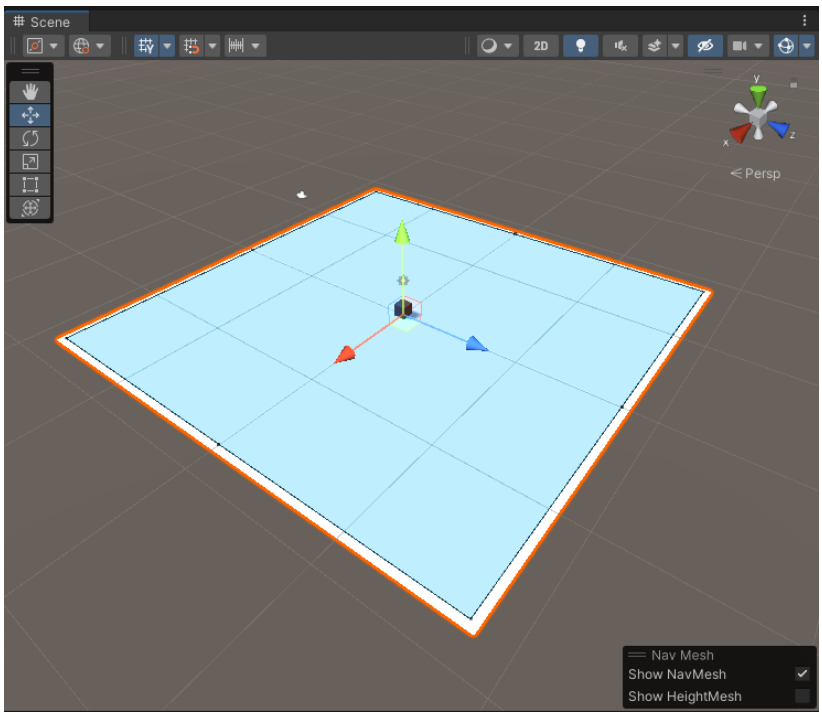

Рис. 7. Результат запекания

Теперь переходим к написанию кода, для этого создаем скрипт и называем «Movement». Начинаем с подключения необходимых библиотек и создания необходимых переменных. см. рисунок 8-12.

Рис. 8. Создание скрипта

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.AI;
```

Рис. 9. Подключение библиотек

```
private NavMeshAgent agent;

private RaycastHit hit;

public Camera camera;

private string groundTag = "Ground";
```

Рис. 10. Создание переменных

```
void Start()
{
    agent = GetComponent<NavMeshAgent>();
}
```

Рис. 11. Получение навигации плоскости

```
Сообщение Unity | Ссылок: 0
void Update()
{
    if (Input.GetMouseButtonDown(0))
    {
        Ray ray = camera.ScreenPointToRay(Input.mousePosition);

        if (Physics.Raycast(ray, out hit, Mathf.Infinity))
        {
            if (hit.collider.CompareTag(groundTag))
            {
                agent.SetDestination(hit.point);
            }
        }
    }

}
```

Рис. 12. Проверка клика мыши и получение координат

Далее необходимо прикрепить скрипт к игроку (кубу) и в созданную публичную переменную поместить основную камеру сцены см. рисунок 13.

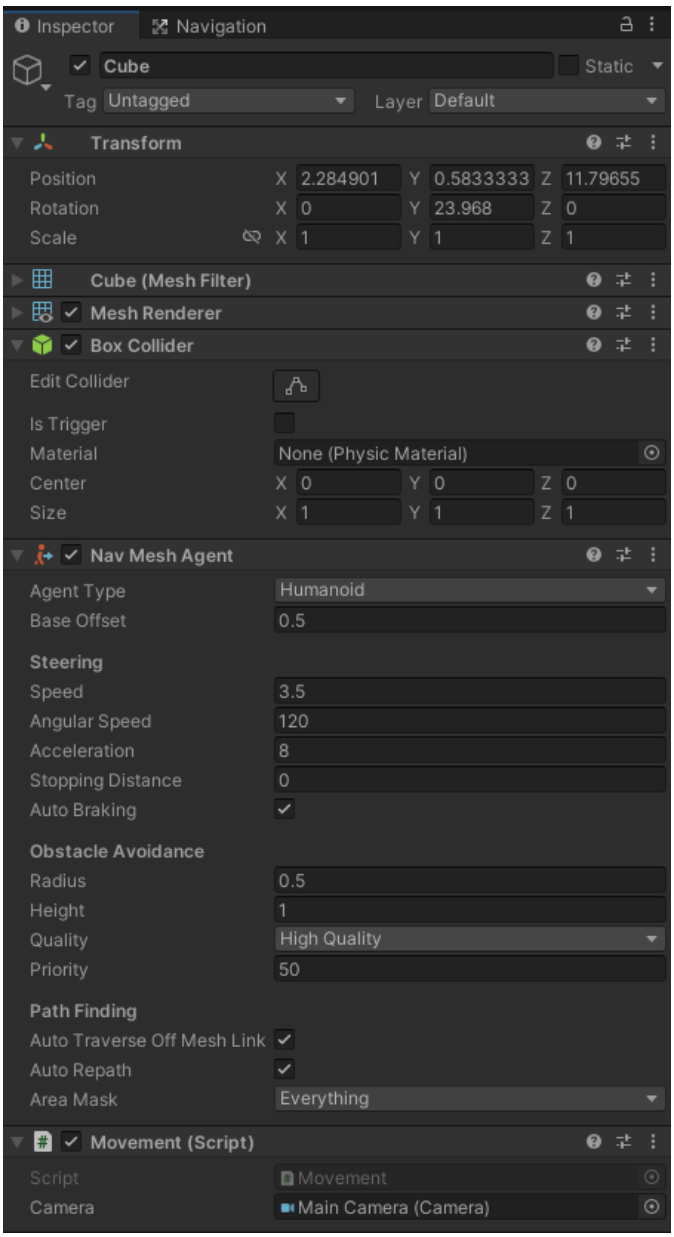

Рис. 13. Свойства куба

Плоскости присваиваем тег «Ground» см. рисунок 14.

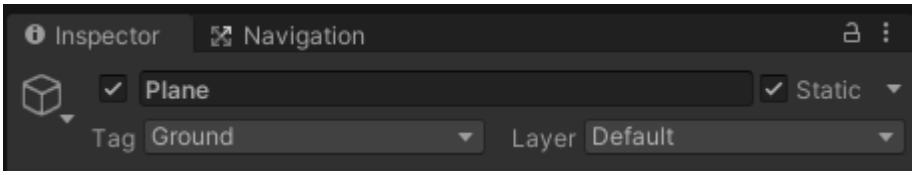

Рис. 14. Свойства плоскости

Проверяем работу скрипта см. рисунок 15-17.

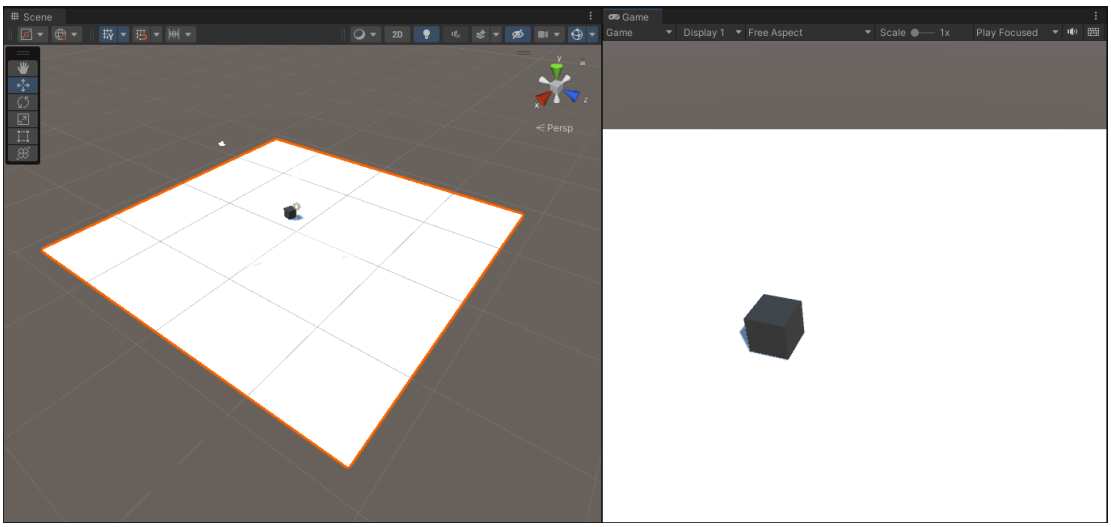

Рис. 15. Проверка работы механики

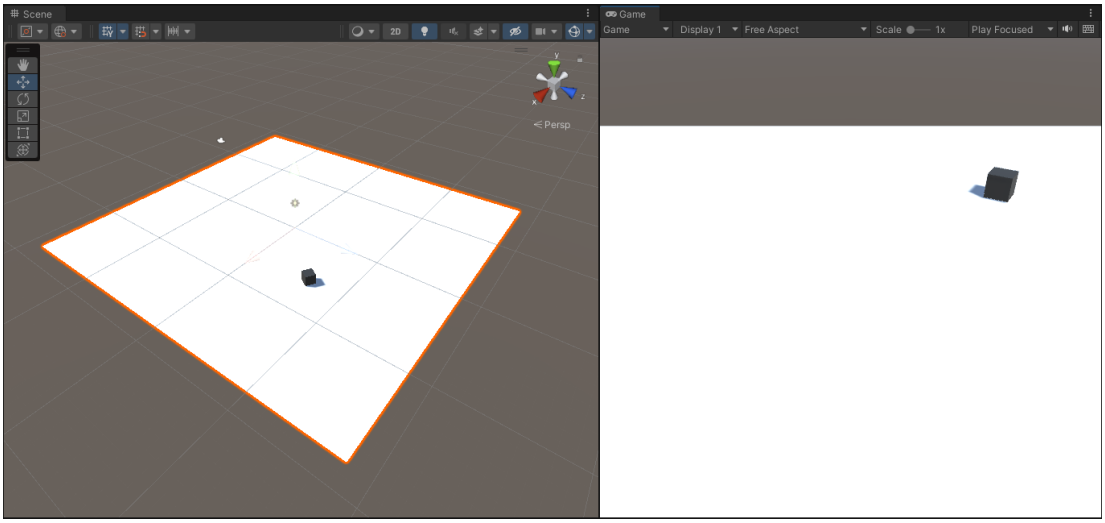

Рис. 16. Проверка работы механики

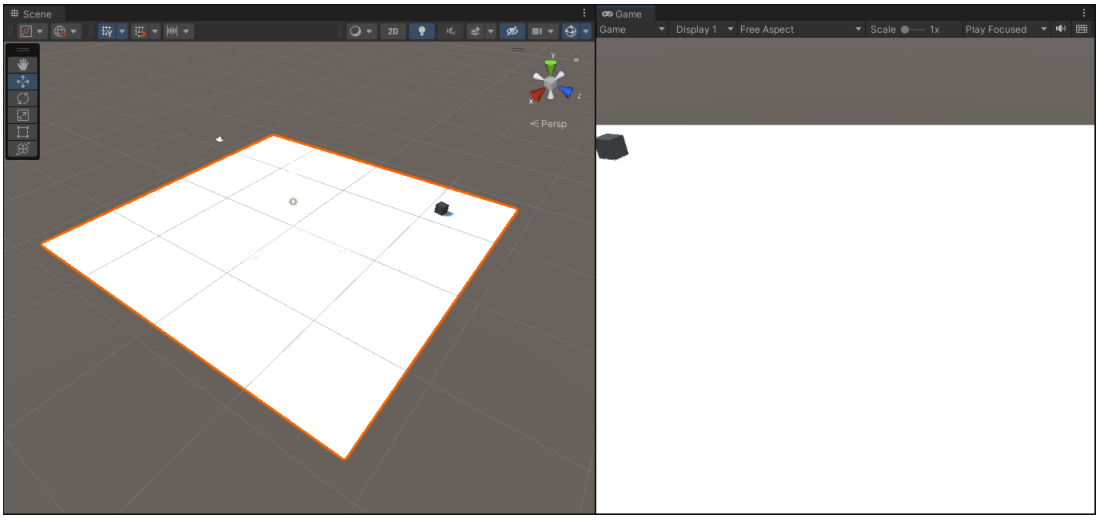

Рис. 17. Проверка работы механики

Как показано на скриншотах механика передвижения работает и зависит только от размера плоскости. В данной статье была реализована механика передвижения по клику мыши.

Библиографический список

1. Кабанов А.А. Графика видеоигр// Россия молодая: передовые технологии – в промышленность. 2013. №2. С. 039-040.
2. Суродин С. А. Unity 3D. разработка сценария проектирования в среде Unity 3D// Информатика и вычислительная техника. 2015. №3. С. 504-511.
3. Гайнуллин Р. Ф., Захаров В. А., Аксенова Е. А. Создание 2d игры на Unity 3D 5.4 // Вестник современных исследований. 2018. №4. С. 78-82.
4. Богданов К. В., Михеев П. Р., Суворов И. Н. Развитие игровых движков// Актуальные научные исследования в современном мире. 2021. №4. С. 24-29.