

Разработка игры грузового погрузчика и обзор различных способов движения трёхмерных объектов в Unity 3D

Козич Полина Александровна

*Приамурский государственный университет имени Шолом-Алейхема
студент*

Научный руководитель:

Лучанинов Дмитрий Васильевич

*Приамурский государственный университет имени Шолом-Алейхема
Старший преподаватель*

Аннотация

В данной работе рассмотрен способ создания манипулятора погрузчика в программе unity 3d. Для реализации данной задачи был использован встроенный модуль эмулятора физики.

Ключевые слова: unity3d, моделирование, манипулирование, программирование.

Development of a cargo loader game and an overview of different ways of moving three-dimensional objects in Unity3D

Kozich Polina Alexandrovna

*Sholom-Aleichem Priamursky State University
Student*

Scientific Supervisor:

Luchaninov Dmitry Vasilyevich

Sholom-Aleichem Priamursky State University

Senior lecturer of the Department of Information Systems, Mathematics and Law Informatics

Abstract

In this paper, we consider a method of creating a loader manipulator in the Unity 3D program. To implement this task, a built-in physics emulator module was used.

Keywords: unity3d, modeling, manipulation, programming.

Реализовать прототип любой идеи за короткое время является полезным навыком в современном мире при нехватке времени. Межплатформенная среда разработки компьютерных игр Unity 3d является отличным инструментом для реализации быстрых прототипов, которые позволяют продемонстрировать жизнеспособность любой идеи, что также позволит оценить необходимость дальнейшего развития идеи.

Целью данного исследования является изучение способов манипулирования объектами в Unity 3D и создание прототипа игры для управления грузовым погрузчиком.

В исследовании Троховой Татьяны [1] рассматривается использование среды unity3d для реализации имитационных компьютерных моделей динамических технических объектов. Результатами являются сложные модели физических компонентов и эмуляции физики реального мира, которые можно использовать при обучении дисциплинам трёхмерного моделирования. В работе «Моделирование технических систем в среде unity 3D» [2] предложена концепция трёхмерного моделирования технических систем и процессов с помощью программных средств разработки компьютерных игр, одним из которых является среда Unity 3D.

Ф.Р. Аметов и его коллеги, в своём исследовании [3] подчеркивают актуальность моделирования качественной игровой сцены. Описывают проблему моделирования игровой сцены. Поэтапно рассматривают процесс моделирования. В том числе описывают процесс добавления поверхности и его настройки и свойства.

Для манипуляции объекта в Unity существует три основных способа (ниже gm – Gameobject gm):

- (1) gm.GetComponent<Rigidbody>().velocity += new Vector3(x,y,z);
- (2) gm.transform.position += new Vector3(x,y,z);
- (3) gm.GetComponent<Rigidbody>().AddForce(Vector3.up * 10f);

Первый способ: установка velocity, вектора скорости твёрдого тела. В большинстве случаев нет необходимости изменять скорость объекта напрямую, т.к. это может быть причиной нереалистичного поведения или физики. Velocity используется в основном в тех местах, где нужно немедленно изменить скорость, например прыжок в игре. При вызове компонента Rigidbody, происходит вызов к обработчику физики, что делает движения реалистичными, но более затратными по требуемым ресурсам.

Второй способ: изменение позиции объекта в пространстве Unity относительно мировых координат. Объект движется без обращения к физическому компоненту Unity.

Третий способ: моментальное применение силы к объекту. Для обычного движения машины или персонажа, это действие бесполезно, но будет правильным, использовать данный метод движения для пуль и снарядов.

Можно также рассмотреть четвёртый способ, который является комбинацией второго способа, плюс использование особенностей физического симулятора программы. Мы можем двигать один объект с помощью transform, который будет влиять на другие объекты, у которых не установлено никаких скриптов кроме компонента Rigidbody, устанавливающий данный объект, как твёрдое физическое тело.

В нашем проекте был избран именно последний способ. Подъёмник изменяет своё положение с помощью transform, а ящик является примитивом с компонентом Rigidbody.

Для реализации данного проекта в Unity необходимо выполнить следующие шаги:

1. Скачать трёхмерную модель погрузчика [4] и грузовика.
2. Собрать набор текстур, бетонный пол, кирпичная стена
3. Создадим трёхмерный проект в unity 3d.
4. Установим несколько Plane примитивов для пола и стен.
5. Установим погрузчик на сцену. На данном этапе за нашим погрузчиком необходимо закрепить объект камеры, для того, чтобы камера следовала за погрузчиком при движении.
6. Создадим скрипт управления погрузчиков `move_loader.cs`.

В данном скрипте объявляем public переменные типа `GameObject`, `loader` и `planes`. В данные переменные мы укажем объекты погрузчика и грузоподъёмника соответственно.

Первым напишем скрипт для движения грузоподъёмника. Движение происходит с помощью обращения к свойству `transform`. Также необходимо указать ограничение максимальной высоты и максимального низкого положения грузоподъёмника, чтобы грузоподъёмник визуально не проходил сквозь пол или не отделялся от погрузчика выше его 3d модели. Данное ограничение накладывается по переменной `transform.position.y`, отвечающей за высоту. Назначим клавиши управления грузоподъёмником, на `space` для поднятия, на `shift` для опускания. Листинг исходного кода управления грузоподъёмником изображён на рисунке 1.

```
void Update()
{
    LiftUp();
    LiftDown();
}

void LiftUp()
{
    if (planes.transform.position.y <= 4.8f) 1
    {
        if (Input.GetKey(KeyCode.Space)) 2
        {
            planes.transform.position += new Vector3(0, speed, 0); 3
        }
        else
            planes.transform.position += new Vector3(0, 0, 0);
    }
    else
        planes.transform.position += new Vector3(0, 0, 0);
}

void LiftDown()
{
    if (planes.transform.position.y >= 0.1f)
    {
        if (Input.GetKey(KeyCode.LeftShift))
        {
            planes.transform.position += new Vector3(0, -speed, 0);
        }
    }
}
}
```

Рис. 1. Исходный код управления движением подъёмником

Создадим пустой объект `manager` и закрепим за ним наш скрипт, укажем скрипту где находится объект погрузчика, а где грузоподъемника. Для того, чтобы на коробку действовала сила тяжести, добавим компоненты `Rigidbody` и `boxCollider`. `Boxcollider` прикрепим и к грузоподъёмнику. На данном этапе мы уже можем поставить коробку на грузоподъёмник в окне редактора и задействовать управление с помощью скрипта.

Для движения машины напишем аналогичный скрипт. В нашем случае, движение машины можно также написать с помощью `transform`, но так как стоит задача, что игра будет иметь вид сбоку, то необходимо закрепить движения по координате `z`, для исключения случайного поворота объекта. Укажем функции управления влево и вправо, на клавиши `A` и `D`. Пример работы погрузчика можно увидеть на рисунке 2.



Рис. 2. Пример работы погрузчика в динамике

Цель по созданию прототипа погрузчика была достигнута и при некоторых доработках, данная игра может быть интересна в долгосрочной перспективе.

В данной работе мы рассмотрели основные способы задания движения объектов и выбрали наиболее подходящий способ для решения текущей задачи. Был создан рабочий вариант компьютерной игры грузового погрузчика за короткое время. Программа Unity 3d продемонстрировала себя как отличный инструмент для создания быстрых и качественных прототипов. Созданный проект можно использовать для демонстрации возможностей игровой среды для обучения студентов по дисциплинам связанным с трёхмерных моделированием.

Библиографический список

1. Трохова Т. А. Компьютерное моделирование динамических объектов с использованием среды разработки компьютерных игр Unity. 2018.
2. Ульянов Р. С., Прокопьев С. В., Делибалтов В. В. Моделирование технических систем в среде Unity 3D //Молодой ученый. 2015. №. 11. С.

452-455.

3. Аметов Ф. Р. и др. Моделирование игровой сцены при разработке компьютерной игры на платформе Unity 3D //Таврический научный обозреватель. 2016. №. 6 (11).
4. Forklift 3D модель // Free3d URL: <https://free3d.com/3d-model/forklift-977987.html> (дата обращения: 22.12.2019).