

Студ. И. Г. Бондарев

Науч. рук. ассист. И. А. Миронов

(кафедра информатики и веб-дизайна, БГТУ)

ПРОГРАММИРОВАНИЕ ПРИЛОЖЕНИЙ ВИРТУАЛЬНОЙ РЕАЛЬНОСТИ В СРЕДЕ UNITY

Виртуальная реальность – это искусственно созданный мир, который заменяет нам реальный, посредством воздействия на различные органы чувств: зрение, слух, обоняние, осязание и другие.

Технологически можно сказать, что потомками современных систем виртуальной реальности, являлись стереоскопические проекторы, которые создавали стереоскопическое изображение и тем самым подменяли нам орган зрения. Дальнейшее развитие виртуальной реальности происходило бурно, создавались различные проекты, позволяющие погружаться в виртуальную реальность, однако, все они либо не могли применяться коммерчески, либо потерпели провал. Это продолжалось вплоть до создания шлема виртуальной реальности нового поколения под названием *OculusRift*. Этот шлем является относительно доступным, при этом давая очень качественную виртуальную реальность для пользователя. Но также появились проекты доступные почти всем, это тенденция началась с устройства *DurovisDive*. Это устройство, самодельный шлем виртуальной реальности, который можно распечатать на 3D принтере и затем, вставив в этот шлем свой смартфон, любой пользователь получит доступ к виртуальной реальности. Также два инженера из компании *Google* придумали устройство под названием *GoogleCardboard*. Это такой же шлем виртуальной реальности, однако он складывается из кусочков картона. Вот такие устройства сделали погружение в виртуальную реальность действительно доступной любому пользователю.

Для отображения виртуальной реальности используются различные типы устройств. Эти устройства в основном делают две вещи: формирование стереоскопического зрения и отслеживание положения головы.

Для формирования стереоскопического зрения экран разделяется на две части, в результате чего, наши глаза видят немного разные изображения, что создаёт эффект «стерео» в голове. Отслеживание головы нужно для того, чтобы, когда мы двигали головой, изображение менялось соответствующим образом. Поэтому основное, что делают шлемы виртуальной реальности, они выполняют эти две функции.

Для формирования стереоизображения необходимо два источника изображения, расположенных на некотором расстоянии друг от друга. В виртуальном 3D-мире за это отвечают две камеры, расположенные со сдвигом. На рисунке 1 показан пример формирования стереоскопического изображения в среде Unity. В данном примере показаны параметры этих камер симулирующих наши глаза. Для дальнейшей работы важны два параметра, X и W.

W – параметр, который показывает какая часть экрана будет выводить изображение камеры. В нашем случае и одна, и вторая камеры будут занимать только 50 % от стандартного размера экрана. При этом обе камеры будут показывать совершенно одинаковые изображения, что не даёт нам стереоизображения. В дальнейшем понадобится параметр камеры X.

X – параметр, отвечающий за сдвиг камеры по оси X от начального положения. Сдвинув камеру симулирующую правый глаз на длину, равную половине длине экрана, получаем два разных изображения. А именно, вторая камера будет немного сдвинута, что позволит симулировать глаза внутри виртуальной реальности и таким образом сформировать стереоизображение.

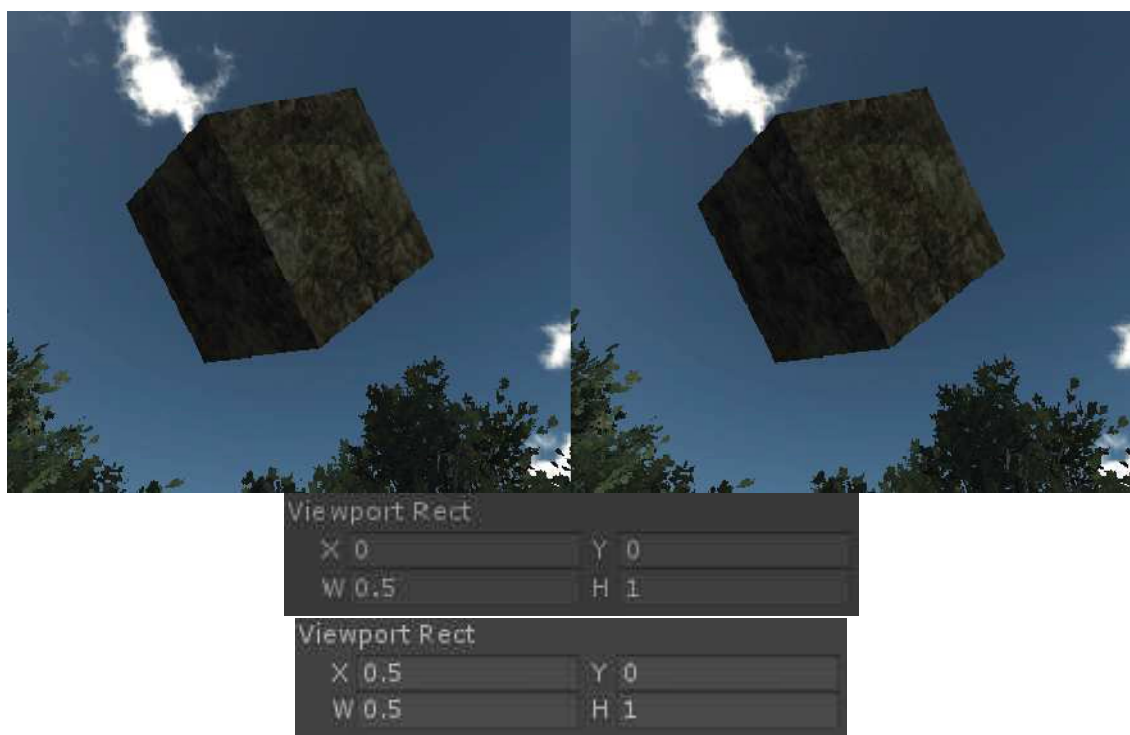


Рисунок 1 – Формирование стереоскопического изображения

За положение телефона в пространстве в *WindowsPhone* отвечает специальный класс под названием *Motion*, фрагмент которого представлен на рисунке 2. Этот класс, комбинирует в себе различные сен-

соры, такие как гироскоп и акселерометр для того, чтобы выдать на выходе один вектор вращения телефона в пространстве.

```

Motion m = new Motion();
m.CurrentValueChanged += m_CurrentValueChanged;
m.Start();
...
void m_CurrentValueChanged (object s,
SensorReadingEventArgs<MotionReading> e)
{
    DoSomething(e.SensorReading.Attitude.Roll,
                e.SensorReading.Attitude.Yaw,
                e.SensorReading.Attitude.Pitch);
}

```

Рисунок 2 – Листинг класса *Motion* необходимый для контроля положения головы

То есть вектор из трёх компонентов *Yaw*, *Pitch*, *Row*, который и определяет положение телефона в пространстве. Необходимо создать объект типа *Motion*, подписаться на изменение значения положения телефона и после этого определённая функция будет постоянно вызываться, при этом ей передаются новые показания датчиков телефона. И чтобы осуществить вращение камеры в соответствии с этими показаниями, необходимо в *Unity* предусмотреть некоторую функцию поворота камеры, и из *VisualStudio* проекта *WindowsPhone* создается объект класса *Motion*, чтобы при изменении координат изменять вращение камеры. Необходимо создать в *Unity C#* скрипт, который будет содержать функцию.

```

public void SetAttitude(float x, float y, float z)
{
    Quaternion tmpRotation = Quaternion.Euler ( x *
Mathf.Rad2Deg,
                                                y * Mathf.Rad2Deg,
                                                z * Mathf.Rad2Deg);
    this.transform.rotation = Quaternion.Euler(180, 0, 0) *
tmpRotation *
Quaternion.Euler(-90, 0, 0);
}
}

```

Рисунок 3 – Листинг функции реализующий поворот камеры в *Unity*
 Данная функция принимает *Yaw*, *Pitch* и *Row*, после чего, из-за того, что телефон передаёт углы поворота в радианах, а в *Unity* всё от-

считывается в градусах, необходимо преобразовать данные параметры из радиана в градусы. Также надо учитывать то, что оси координат смартфона и Unity не совпадают. Как выглядят системы координат смартфона и в среде Unity представлено на рисунке 4.



Рисунок 4 – Системы координат смартфона и в среде Unity

Поэтому, перед использованием углов поворота внутри приложения, необходимо преобразовать данные координаты. Для решения данной проблемы запрограммированы специализированные функции, базирующиеся на таком понятии как кватернион [1]. Кватернионы, очень полезны, когда необходимо описать движения объекта в пространстве.

После чего можно создавать различные интерактивные 3D- модели в среде Unity и связывать с проектами под *WindowsPhone8*. В дальнейшем проект, можно будет открыть через *VisualStudio*. Теперь необходимо прописать описанный выше класс *Motion*, далее устанавливать данной приложение на смартфон с *WindowsPhone8*, и теперь, запустив приложение и вставив смартфон в очки *GoogleCardboard*, сможем наслаждаться полноценной виртуальной реальностью.

ЛИТЕРАТУРА

1. Скепко Р.Ю., Дятко А.А. Применение кватернионов в компьютерной графике. Информационные технологии: тезисы 80-й науч.-техн. конф. профессорско-преподавательского состава, научных сотрудников и аспирантов [Электронный ресурс]. – Минск: БГТУ, 2016. – С. 27.