

ПОДХОДЫ К ТРАНСФОРМАЦИИ ОБЪЕКТОВ В UNITY 3D

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Жерко А. Ю., Макаревич В. А.

Тонкович И. Н. – канд. хим. наук, доцент

Предложены подходы к трансформации объектов в Unity 3D: посредством вызова API-функций (на примере Rotate) и с помощью воспроизведения анимаций. Показано, что применение этих подходов в большей мере зависит от типа воспроизводимой динамической модели.

При разработке мобильных приложений возникает необходимость создания объектов, посредством которых требуется визуализировать взаимодействие между пользователем и виртуальным пространством. Достигается это путем изменения цвета, формы, размера, вращения одного или более объектов.

При взаимодействии пользователя с объектом наполняющие пространство объекты могут изменять свои характеристики, взаимодействовать друг с другом, перемещаться в различных направлениях, поворачиваться и т. д. Эти изменения могут определяться с помощью заранее заданного сценария, а могут происходить вследствие действий пользователя. Как правило, наиболее часто изменение объекта происходит за счет его трансформации.

В среде Unity каждый объект (GameObject) может иметь неотключаемый компонент Transform, отвечающий за положение, размеры объекта, поворот; содержащий инструменты и соответствующие им API-функции Translate, Rotate, Scale. Реализация функции поворота с помощью Rotate показана на Листинге 1 (рисунок 1).

```
public Transform turret;
public int speed;
void Update() {
    if(Input.GetKey(KeyCode.LeftArrow)) {
        turret.Rotate(new Vector3(0, -5,
0),speed * Time.deltaTime, Space.Self);
    } else if
        (Input.GetKey(KeyCode.RightArrow)) {
        turret.Rotate(new Vector3(0, 5,
0),speed * Time.deltaTime, Space.Self);
    }
}
```

Рисунок 1 – Листинг 1 Функция поворота с помощью Rotate

Данный пример иллюстрирует поворот объекта при нажатии клавиши «Стрелка влево» со скоростью speed градусов в секунду. При нажатии клавиши «Стрелка вправо» происходит аналогичное симметричное действие объекта. Параметр скорости изменения объекта speed имеет доступ public. Если этот параметр указать в интерфейсе Unity, тогда нет необходимости редактировать код скрипта.

Для решения задачи создания динамических изменений можно предложить подход, связанный с использованием анимации, посредством которой мы можем моделировать и визуализировать целый ряд изменений объекта: вращение, перемещение, изменение размеров и т.д.

Анимации используют тогда, когда известно несколько начальных параметров объекта, связанных с ключевыми точками временной линии анимации. Анимация удобна тем, что достаточно задать две точки, одну в начале, до того, как использовать анимацию, а вторую – итоговую точку. Посредством панели Animator можно создавать состояния, между которыми задаются переходы, при прохождении которых будет запускаться анимация конечного состояния. К примеру, при переходе из состояния HatchOpen0 в HatchClose0 будет запускаться анимация закрытия объекта. Также можно задавать различные параметры, контролирующие переходы из одного состояния в другое.

Таким образом, рассмотрены два различных подхода к созданию динамических имитационных моделей: посредством вызова API-функций (на примере Rotate) и с помощью воспроизведения анимаций. Как было показано выше, возможность применения данных подходов в большей мере зависит от типа воспроизводимой динамической модели. Вызов API-функций в скриптах используют, когда параметры трансформации объекта зависят напрямую от ввода пользователя. Анимацию применяют как воспроизведение трансформаций объектов с заранее заданными параметрами.