

СОЗДАНИЕ 3D-ГОНОК НА UNITY

В работе приводится описание основ создания 3D-игры в жанре «гонки» на базе движка Unity.

ВВЕДЕНИЕ

Unity — межплатформенная среда разработки компьютерных игр. Unity позволяет создавать приложения, работающие под более чем 20 различными операционными системами. Редактор Unity имеет простой DragAndDrop интерфейс, который легко настраивать, благодаря чему можно производить отладку игры прямо в редакторе. Движок поддерживает два скриптовых языка: C, JavaScript (модификация).

I. ПРОЦЕСС СОЗДАНИЯ ИГРЫ

Наша игра представляет собой 3D-гонки на различных геометрических объектах. Игроку предстоит быстрее соперников преодолеть чек-поинты и добраться до финиша. Простой и увлекательный геймплей, а также понятный и удобный интерфейс идеален для таких казуальных игр. Для реализации нашей задумки мы прописывали специальные скрипты. С самого начала мы реализовали код гравитации планеты и код гравитации нашего игрока, которые мы крепили на нашу «планету» и игрока соответственно. Добавив игроку код реализации его движения мы получили полноценное перемещение по поверхности планеты. Для реализации этих трех кодов мы воспользовались кватернионами. Кватернион представляет собой вращение на определенный угол вокруг произвольной оси. Основной идеей игры было создание полноценных гонок, которые могли бы захватывать игрока с первых секунд. Для создания нужной атмосферы соперничества перед нами встала задача реализации искусственного интеллекта для нашего врага, который будет соревноваться с игроком за первое место. При создании искусственного интеллекта

мы смогли добиться движения врага, которое во многом похоже на реальное движение транспорта: ускорение и замедление, плавность на поворотах при объезде препятствий и т.п. Чтобы добиться такого эффекта мы прибегли к помощи корутин. Корутины в Unity — простой и удобный способ запускать функции, которые должны работать параллельно в течение некоторого времени. Корутины представляют собой простые C итераторы, возвращающие IEnumerator и использующие ключевое слово yield. Далее Unity опрашивает зарегистрированные корутины после каждого вызова Update, определяя по возвращаемому в yield значению, когда нужно переходить к следующему блоку кода.

II. ИНТЕРФЕЙС

В нашей игре мы задействовали простой и удобный интерфейс. В нашей игре мы задействовали простой и удобный интерфейс. Игроки видят похожий во многих других играх, что позволяет не затрачивать время игрока для понимания как и что работает. Ему не нужна дополнительная информация, чтобы разобраться в назначении каждой кнопки. Для его проектирования мы создавали дополнительные сцены, в которых использовали скрипты со специальной библиотекой using UnityEngine.SceneManagement; для последующего использования специальных функций этой библиотеки, например для загрузки других сцен проекта.

Список литературы

1. www.docs.unity3d.com/Manual/PhysicsSection
2. www.habr.com/post/216185/
3. www.habrahabr.ru/post/30347

Азаревич Евгения Сергеевна, студентка 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, azarevich.evgeniya@gmail.com

Орел Анастасия Николаевна, студентка 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, orelanastasia@mail.ru

Рашкевич Полина Александровна, студентка 2 курса факультета информационных технологий и управления Белорусского государственного университета информатики и радиоэлектроники, polinarash75@gmail.ru

Научный руководитель: Рак Татьяна Александровна, ассистент кафедры вычислительных методов и программирования Белорусского государственного университета информатики и радиоэлектроники, tatianarak@bsuir.by.