Издатель
ФГБОУ «Петрозаводский государственный университет»
Российская Федерация, г. Петрозаводск, пр. Ленина, 33

Студенческий научный электронный журнал

StudArctic Forum

http://saf.petrsu.ru

№1(21) / 2021

Главный редактор
И. М. Суворова

Заместитель главного редактора
М.И. Зайцева

Редакционный совет
В.А. Шлямин
В.С. Сюнёв
Г.Н. Колесников
С.В. Волкова

Редакционная коллегия
А.Ю. Борисов
П.С. Воронина (ответственный секретарь)
Р.В. Воронов
Т.А. Гаврилов
Е.О. Графова
Л.А. Девятникова
М.И. Зайцева
А.А. Ившин
А.Ф. Кривоноженко
А.А. Кузьменков
Е.Н. Лузгина
Ю.В. Никонова
М.И. Раковская
А.А. Скоропадская
Е.И. Соколова
И.М. Соломещ
А.А. Шлямина

Редакция
А. Г. Марахтанов
А. А. Малышев
Р. А. Мацуев

ISSN 2500-140X

© ФГБОУ «Петрозаводский государственный университет», 2012

Адрес редакции
185910, Республика Карелия, г. Петрозаводск, ул. Ленина, 33.
E-mail:saf@petrsu.ru
http://saf.petrsu.ru

© ФГБОУ «Петрозаводский государственный университет», 2012

Студенческий научный электронный журнал
StudArctic Forum

ПЕТРОЗАВОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

http://saf.petrsu.ru **http://petrsu.ru**

Информатика и вычислительная техника

Разработка прототипа игры для мобильных устройств в 3D представлении для одного или двух игроков

БЕТЕЛЕВ Владимир Кириллович

Петрозаводский государственный университет (Россия, Республика Карелия, г. Петрозаводск, Ленина проспект, 33),
vova.betelev@yandex.ru

Ключевые слова:
игра
кююккя
kyykkä
прототип
игрок
мобильныеустройства
Unity3D
Геймплей
оптимизация.

Аннотация: В данной статье автором описывается создание прототипа игры. Приведены основные особенности процесса создания прототипа, указаны методы разработки. Описаны варианты оптимизации прототипа игры для мобильных платформ.

Основной текст

В качестве реально существующей игры для создания её цифровой копии была взята игра kykke (на русском — кююккя). Эта народная командная игра популярна у финнов и карелов. Она напоминает игру в городки.

Для создания прототипа игры была выбрана среда разработки игр Unity 3D версии 2019.3.1f1 по следующим причинам:

Кроссплатформенность — возможность «в один клик» выпускать готовую игру под другую платформу (Android, IOS, PC);

Простота использования. Из-за огромного набора готовых функций и методов разработка игр на Unity позволяет разработчику не отвлекаться на изобретении «велосипеда», а сосредоточиться на создании игрового процесса и большого количества игровых механик;

Удобный скриптовый язык программирования C#, позволяющий быстро и качественно написать сложные программные конструкции.

Для создания текстур использовался растровый редактор GIMP, а для создания 3D моделей - 3D-редактор Blender.

Перед стартом игры персонаж разворачивается к игровому полю. Когда игрок нажимает на одну из кнопок интерфейса (Left или Right), первому или второму персонажу передаётся команда для смены позиции. Если игрок продолжает нажимать на кнопку, персонаж передвигается в сторону до тех пор, пока не достигнет крайней позиции (или игрок не перестанет нажимать на кнопку).

Когда пользователь нажимает на 3D-модель игрового персонажа, объект «бита» перестаёт быть дочерним персонажу, становится самостоятельным объектом, и ему придаётся ускорение в определённом направлении через метод AddForce с помощью компонента Rigidbody (стандартный компонент Unity), отвечающего за физику. Сила броска зависит от положения специального объекта SliderFrow, значение которого можно получить с помощью компонента Slider (стандартный компонент Unity) и его параметра value. Также бита закручивается с помощью метода AddTorque компонента Rigidbody (стандартный компонент Unity). В параметры броска как через AddForce, так и через AddTorque, вносятся каждый раз новые случайные значения для большей вариативности и увлекательности игрового процесса.

Камера игрока начинает двигаться по вектору движения биты, чтобы проследить за полётом биты. После того, как бита прекратила своё движение, камера игрока возвращается в стандартную позицию. С помощью компонента MeshCollider (стандартный компонент Unity) можно отследить, что какой-то объект коснулся городка. Тогда надо начать проверять угол наклона городка в глобальных координатах по двум осям к нулевым глобальным координатам. Если этот угол больше или равен 135, то городок считается упавшим (135 градусов — игровая условность).

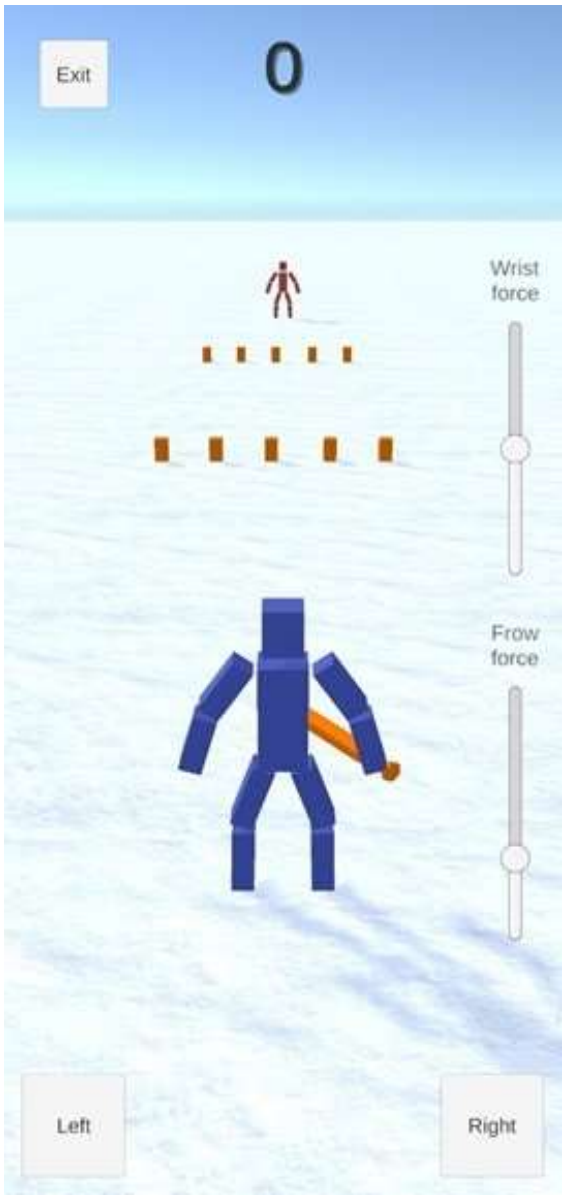

Рисунок 1. Экран после начала игры.

Если играют два игрока, то один балл начисляется тому, чей городок был сбит. Если сбит чужой городок, то у игрока снимается один балл. Когда городок упал, вызывается метод AddPoints из компонента Cube (создан автором статьи).

Вместе с созданием уровня — инстантиированием (от англ. instantiate — создать экземпляр) на сцену кубиков в определённом порядке передаётся и количество очков для победы в переменную countPointsToWin компонента GameController (создан автором статьи). После каждого обновления очков у обоих игроков вызывается метод CheckIfWin компонента GameController (создан автором статьи), проверяющий количество очков для победы.

Так как игра создавалась для мобильных платформ, то были подробно изучены и применены следующие варианты оптимизации:

- объединение текстур в текстурные атласы. Аппаратное видеоядро теперь загружает не много текстур по отдельности, непрерывно обращаясь к памяти устройства, а один большой элемент. Прирост в скорости загрузки был отмечен в 2,5 раза;
- оптимизация полигональной сетки моделей и использование подхода low-poly, чтобы сократить кол-во вызовов отрисовки полигонов видеоядром (drawcalls);

Использование Occlusion Culling — системы, с помощью которой приложение получает информацию о тех объектах, которые видит (или нет) камера игрока. Можно нарисовать объекты, попадающие в поле видимости игрока. Тогда не придется тратить ресурсы на отрисовку объектов, которых игрок не наблюдает;

- Также был оптимизирован и программный код игры. Стандартный метод Update, который Unity3D вызывает каждый кадр, удалось уменьшить до 5 строчек. Все физические вычисления были перенесены в специальный метод FixedUpdate, не зависящий от количества кадров в секунду (FPS — frame rate per seconds), а подчиняется внутреннему измерению времени: если у игрока на устройстве будет малый FPS, все физические вычисления (полёт биты, падение городка) будут просчитываться верно и вовремя.

Для усложнения игрового процесса был добавлен специальный слайдер, отвечающий за силу закручивания биты при броске. Однако этот слайдер никуда не отправляет данные и является лишь «заглушкой», чтобы создать у пользователя ощущение сложности игровой механики и заставить его прочувствовать геймплей, увлечься тактикой и расчётами броска.

Данный прототип полностью готов для дальнейшего развития. Например, можно заменить модели персонажей, создать много новых уровней с разной расстановкой городков и разным количеством победных баллов, заменить текстуры, добавить спецэффекты и многое другое.

© 2021 Автор Владимир Кириллович.

Computer Science and Engineering

Development of a prototype game for mobile devices in 3D performance for one or two players

BETELEV Vladimir Kirillovich

Petrozavodsk state university (Russia, Karelia republic, Lenina prospect 33), vova.betelev@yandex.ru

Ключевые слова:
game
kyuyukkya
kyykkä
prototype
player
mobile
devices
Unity3D
gameplay
optimization.

Аннотация: In this article, the author describes howto create a prototype of a mobile game. There are main features of the prototype creation process,development methods are indicated. The author describesthe options for optimizing the prototype of the game for mobile platforms.