

Студ. Н.А. Чуркин; студ. С.Н. Кирьянов
Науч. рук. доц. Н. А. Жилияк
(кафедра информационных технологий, БГТУ)

СОЗДАНИЕ КОМПЬЮТЕРНЫХ ИГР

1. Актуальность темы

В 2016 году ушли в прошлое те времена, когда разработка компьютерных игр была доступна лишь избранным. Каждый день создаются и усовершенствуются новые мощные и удобные инструменты для разработки. Теперь каждый человек может опробовать себя в этой интересной сфере.

Однако, несмотря на большое количество многофункциональных инструментов, от разработчика всё равно требуется очень большое количество труда, чтобы создать действительно интересный продукт. Однозначным плюсом разработки компьютерной игры на уже готовом инструменте (движке), является скорость разработки программной части. Если раньше программисту нужно было писать множество строк кода, чтобы использовать простую возможность проверки столкновения между двумя объектами, то теперь все эти вычисления делаются всего одной командой. Таким образом, игровые движки сильно упростили жизнь программистам. Однако, программирование это только одна ипостась разработки компьютерной игры. Создание качественного игрового продукта будет так же включать в себя создание большого объёма графического материала. И ко всему прочему, потребуется сделать качественное озвучивание игры. С последним проблем обычно не возникает, так как в интернете полно бесплатных звуков и музыки.

2. Выбор метода реализации задачи

Игровой проект можно разделить на отдельные крупные области и использовать в каждой из них приемлемые средства, выстраивая некий стек:

– графический движок. Тут работа с нижележащими API (OpenGL, DirectX), работа с буферами памяти, шейдерами, процедурная генерация, жесткие оптимизации. Здесь C++ вне конкуренции;

– игровая логика. Как правило в конечном счете это просчет взаимодействий объектов с объектами. Причем объекты вполне соответствуют объектам в традиционном ООП понимании. Действия одних объектов, могут вызывать реакции в других; объекты могут образовывать сложные иерархии. Удобно воспользоваться объектно-ориентированным языком с автоматической сборкой мусора, чтобы

сосредоточиться на поведении игровой среды. Например, C#, Java, Python, Ruby;

– алгоритмическая база. Различные варианты AI, работа с графами и сложными структурами данных, поиски и сортировки. Задачи, типичные для функциональных языков F#, Scala, Lisp, Haskell, OCaml, Clojure. Разумеется, не стоит разводить зоопарк трудносовместимых сред в одном проекте. Но некоторые комбинации могут быть вполне эффективными: C++/Java/Scala, C++/C#/F#, C++/Python.

В рамках данного проекта был разработан программный продукт, представляющий из себя игру. В нашем случае был использован движок Unity. Он является современным движком для создания игр и приложений, разработанный *Unity Technologies*. Также широко известен в индустрии, как многоплатформенный игровой движок, с помощью которого вы легко сможете расширить свою аудиторию. Один щелчок мышью — и ваша игра развернута на всех мобильных, VR-, настольных, игровых и телевизионных платформах.

В качестве основного языка был выбран C#. Так как подавляющее большинство разработчиков используют его, все классы, плагины написаны на нем. На официальном сайте присутствует документация по API и всем возможностям этого движка.

3. Алгоритм игрового процесса

Название разработанной игры: Defence. Игра представляет собой “таймкиллер”, такой жанр игр, в которых нет элементов прогресса. Как правило присутствуют таблицы рекордов и внутренние покупки в виде моделей персонажей и прочего. Цель игры: заработать победные очки и побить предыдущий рекорд.

Общее описание процесса. При запуске игры выводится всплывающее окно с возможностью выбора качества графики и разрешения экрана. После чего открывается главное игровое меню с возможностью выбора между тремя пунктами: “Играть” и “Выход”. При нажатии “Играть” мы попадаем в игровую сцену. Принцип игры заключается в том, что каждый раз вы начинаете с нулевым счетом. В процессе игры образуются волны противников, за которые вы можете получить игровые очки (рисунок 1).



Рисунок 1 – Главное меню игры

У игрока присутствует 100 основных единиц здоровья и возможность вернуться в игру после смерти три раза. После поражения на экране выводится набранное кол-во очков и рекорд, а также быстрая возможность начать заново или выйти.

4. Функциональные части

4.1 Техническая часть

Для реализации GUI (GraphicsUserInterface) используется UnityAPI, который содержит в себе возможность добавления таких элементов как: Button (Кнопка), Canvas (Полотно), Panel (Панель), Textbox (Поле для ввода текста), Label / Text (Текст), Checkbox и прочее. Каждый из этих методов может принимать параметры для настройки размеров, цвета, расположения в пространстве сцены.

Для перемещения игрока использовалась функция `GetButtonDown()`, которая принимает код нажатой клавиши и передает в присвоенную переменную. Соответственно был написан скрипт для обработки нажатых клавиш в действие, соответствующее этой клавише. Т.е. если мы нажмем клавишу “А”, то сработает условие, в котором мы задаем игровому объекту (игрок) импульс влево. Также в функции расчета движения учитывается скорость передвижения игрока и `DeltaTime` для плавного перемещения.

Задний фон состоит только из одной текстуры. При первоначальном запуске программы, для уменьшения времени загрузки, на камеру был наложен скрипт. Он определяет: если край камеры достиг края текстуры фона, создается “продолжение” (копия) текстуры фона и игрок не видит обрывов изображения.

4.2 Аудиовизуальное восприятие

Визуальная часть в игре реализована с помощью игровых объектов (`GameObject`), которые имеют свойство `Render`. В нем мы можем задать объекту текстуру отображения с возможностью изменения размера и положения в пространстве по X, Y, Z.

Т.к. это 2D игра, мы использовали такой эффект как параллакс, для создания иллюзии трехмерного пространства. Такой эффект достигается за счет расположения игровых слоев на разных расстояниях друг от друга и движущихся с разной скоростью относительно игрока. Каждое действие игрока сопровождается звуковыми эффектами.

Заключение.

На конечном этапе было получено функционирующее программное средство, которое может являться одним из примеров по разработке игр.

Создание игр – развлечение? К сожалению, это далеко не так. Даже если не касаться профессиональной работы в индустрии производства игр, то и любительский проект — это не развлечение в прямом смысле этого слова. Конечно, процесс работы над воплощением своих идей приносит определенное удовлетворение и удобен как средство самореализации, но самым приятным является момент рождения идеи и её проработка.

УДК 681.9

Студ. О.М. Карпач, В.В. Окулович

Науч. рук. доц. Н. А. Жилияк

(кафедра информационных технологий, БГТУ)

ТЕКСТОВЫЕ ТРЕНАЖЕРЫ КАК СРЕДСТВА ПОВЫШЕНИЯ ПРОДУКТИВНОСТИ ТРУДА

В современном мире все чаще возникает потребность в использовании компьютерных средств обработки информации, но для того, чтобы эффективно расходовать время, необходимо обладать специальными навыками, которые, впоследствии, помогут выполнять большее количество работы с меньшими затратами сил и энергии, именно для этого были разработаны текстовые тренажеры.

Текстовые тренажеры должны удовлетворять следующим требованиям:

- простота использования;
- качество текстовой информации;
- приемлемая стоимость.

Было проведено изучение тестовых тренажеров, представленных в Интернете, учтены все их недостатки, разработано веб-приложение позволяющее облегчить и без того нелегкое обучение слепому 10-пальцевому методу печати.

В основе веб-приложения лежит механизм многократного повторения специального набора слов, который подобран таким обра-