**Федеральное государственное образовательное бюджетное учреждение  
высшего образования**

**«Финансовый университет при Правительстве Российской Федерации»**

**КОЛЛЕДЖ ИНФОРМАТИКИ И ПРОГРАММИРОВАНИЯ**

**ПМ.03 Участие в интеграции программных модулей**

**Группа: 4ПКС-114**

**УТВЕРЖДАЮ**

**Председатель цикловой комиссии**

**программирования и баз данных**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ /Пестов А.И./**

**\_\_\_\_.\_\_\_\_. 2017**

**ПРОЕКТ КУРСОВОЙ**

**На тему:** Компьютерная игра «Funny Boxes» c графическим интерфейсом на языке С#

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**2017**

**Руководитель курсового проекта**

**\_\_\_\_\_\_\_\_\_\_\_\_\_ /Морозова М.В./**

**Исполнитель курсового проекта**

**\_\_\_\_\_\_\_\_\_\_\_\_\_ /\_Кустов В.А.\_\_/**

**Оценка за проект: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_.\_\_\_\_.2017**

Содержание

[Введение 4](#_Toc499674508)

[1 Общий раздел 5](#_Toc499674509)

[1.1 Системные требования 5](#_Toc499674510)

[1.2 Характеристика системы программирования 5](#_Toc499674511)

[2 Технологический раздел 6](#_Toc499674512)

[2.1 Предпроектное исследование предметной области 6](#_Toc499674513)

[2.2 Анализ требований и определение спецификаций программного обеспечения 6](#_Toc499674514)

[2.2.1 Диаграмма вариантов использования 6](#_Toc499674515)

[2.2.2 Функциональная диаграмма 8](#_Toc499674516)

[2.2.3 Диаграмма потоков данных 9](#_Toc499674517)

[2.3 Разработка пользовательских интерфейсов программного обеспечения 9](#_Toc499674518)

[3 Руководство по использованию программы 11](#_Toc499674519)

[3.1 Руководство системного программиста 11](#_Toc499674520)

[3.1.1 Назначение и условия применения программы 11](#_Toc499674521)

[3.1.2 Структура программы 11](#_Toc499674522)

[3.1.3 Обращение к программе 13](#_Toc499674523)

[3.1.4 Входные и выходные данные 13](#_Toc499674524)

[3.2 Руководство программиста 14](#_Toc499674525)

[3.2.1 Общие сведения о программе 14](#_Toc499674526)

[3.2.2 Системные требования 14](#_Toc499674527)

[3.2.3 Программные требования 15](#_Toc499674528)

[3.2.4 Файлы программы 15](#_Toc499674529)

[3.2.5 Настройка программы 15](#_Toc499674530)

[3.2.6 Проверка программы 15](#_Toc499674531)

[3.3 Руководство пользователя 16](#_Toc499674532)

[3.3.1 Основная терминология проекта 18](#_Toc499674533)

[Заключение 20](#_Toc499674534)

[Приложение А 21](#_Toc499674535)

[Приложение Б 22](#_Toc499674536)

# Введение

Курсовой проект представляет собой разработанное ядро игры «Funny Boxes», написанное на языке программирования C# с использованием «Xamarin», «Visual Studio» и библиотеки «Apple Sprite Kit».

С каждый днем с высокой скоростью растет популярность в аркадных играх, в которые можно поиграть, пока едешь в метро, автобусе и т.д. Такие игры в простонародье называют «Убивалка времени».

Основной целью игры Funny Boxes является занять игрока на некоторое время, когда у него есть свободное время, а его хочется скоротать, развитие реакции игрока. Так же преподнести что-то новое в подобных играх игроку, такое, как удержание ящиков на платформе, при этом уворачиваясь от множества посторонних объектов (будет непросто набрать высокий результат).

1. Общий раздел
   1. Системные требования

Система: macOS

Процессор: 2.0 GHz

Оперативная память: Минимум 1 ГБ

Видеокарта: Nvidia, Intel iris Graphics

* 1. Характеристика системы программирования

В целях разработки кроссплатформенного приложения, был выбран язык программирования C#.

С 2017 года, после того, как Microsoft разработали платформу Xamarin, C# можно считать кроссплатформенным языком, однако Xamarin не предусматривает создание нативного интерфейса, ввиду чего необходимы дополнительные языки программирования. Само же портирование с Xamarin на, например, macOS происходит за счет конвертирования C# кода в Objective-C.

Для реализации игры(графика, физика) в целом была использована библиотека Apple SpriteKit. Ее приемущества перед другими подобными библиотеки в том, что она намного быстрее и оптимизированней.

# Технологический раздел

# Предпроектное исследование предметной области

В игре присутствуют платформа и объекты взаимодействия.

Объекты взаимодействия: Ящики, бомбы.

Игровое пространство – сцена игры с задним фоном, на которой расположена платформа (основной объект в игре), которой нужно управлять и падающие объекты, которые нужно ловить или пропускать.

Платформа – главный объект игры, платформой нужно ловить ящики.

Ящик – объект который нужно собирать.

Фейк-ящик – ящик, который очень похож на настоящий, но при соприкосновении с оригинальным ящиком – исчезают оба объекта.

Бомба – объект который раскидывает ящики при соприкосновении с ними, так же есть шанс прибавить к таймеру время и играть дольше.

На стадии планирования, системы на которых должна была работать игра: macOS/iOS. Такое решение было принято ввиду условий, которые Apple предоставляет своим разработчикам, а именно: узкопрофильные библиотеки и широкие возможности для проведения тестирования.

Игра содержит базу данных, состоящую из рекордов разных пользователей.

На сегодняшний день игрокам нужна игра, называемая в простонародье «убивалка времени», часто кому нужно чем-нибудь заняться, чтобы «убить» время, пока есть пустое окно между дел, для того и создаются подобные игры.

Смысл игры – собрать как можно больше ящиков и удержать их, уворачиваясь от бомб и фейк-ящиков за отведенное время.

# Анализ требований и определение спецификаций программного обеспечения

# Диаграмма вариантов использования

Диаграмма вариантов использования представлена на рисунке 3, и состоит из действий, совершаемых пользователем в программе.

После запуска программы пользователь может выбрать одно из следующих действий:

* Начать игру.
* Выйти из игры
* Рекорды

При выборе действия «начать игру» запустится процесс игры.

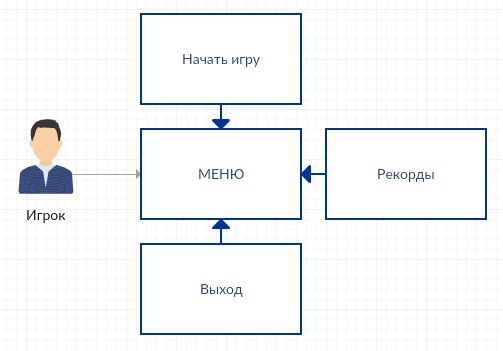


Рисунок 1 - Диаграмма использования

# Функциональная диаграмма

Функциональная диаграмма отображает основные взаимосвязи.

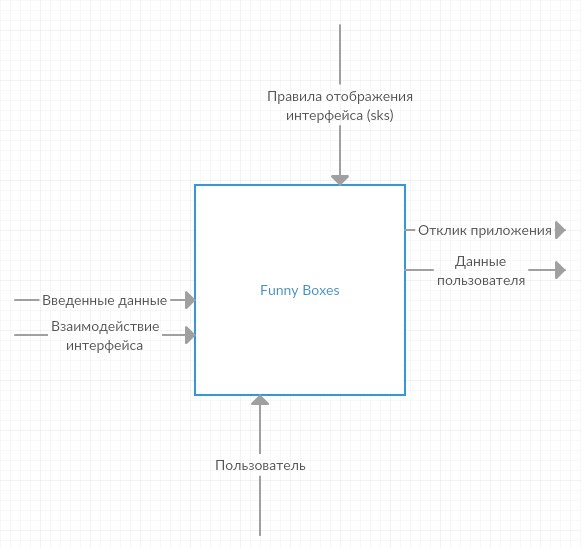


Рисунок 2 – Контекстная функциональная диаграмма

Из контекстной функциональной диаграммы можно заметить, что приложение использует набор файлов .sks, которые содержат правила отображения интерфейса.

* + 1. Диаграмма потоков данных

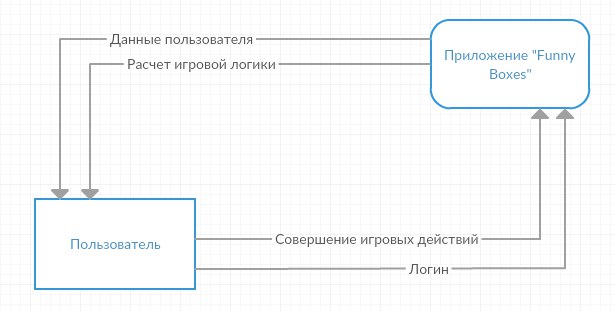


Рисунок 3 – Диаграмма потоков данных

# Разработка пользовательских интерфейсов программного обеспечения

Игровой процесс начинается с кнопки «Старт» находящейся в меню игры, где мы видим кнопки «старт» и «выход» над которыми логотип с названием игры, слева сверху Ваш логин и место в топе, ниже рекорд.



Рисунок 4 - Экран главного меню

Сам же игровой процесс начинает где мы видим задний фон, платформу в виде куска земли с травой. Далее видим таймер, количество очков. Потом начинают падать игровые объекты: Ящики, бомбы.

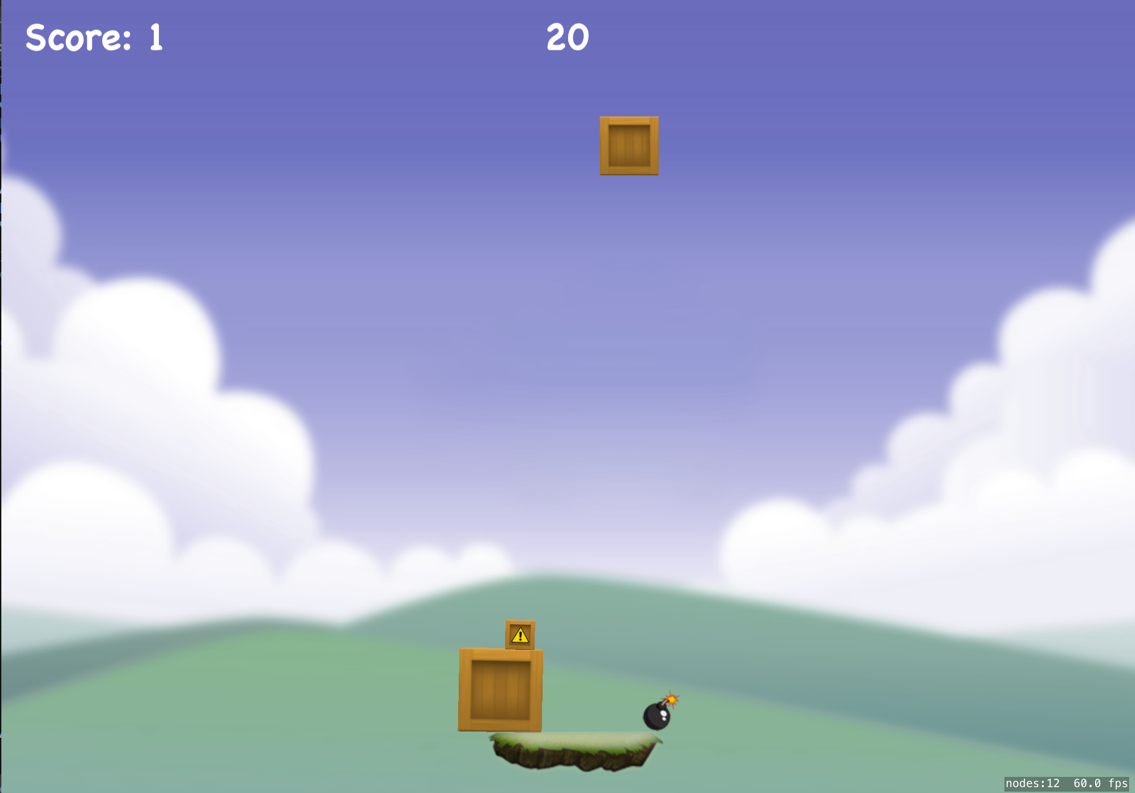


Рисунок 5 – Игровой процесс



Рисунок 6 – Окно окончания игры

1. Руководство по использованию программы
   1. Руководство системного программиста
      1. Назначение и условия применения программы

Программный продукт является аркадной игрой «Funny Boxes», развивающей реакцию.

Возможности игры:

* Система никнеймов;
* Возможность мериться рекордами с другими пользователями;
  + 1. Структура программы

Программа состоит из двух основных компонентов: логика приложения и интерфейс взаимодействия.

Логику приложения составляют огромное количество классов, перечислений и структур, пройденных тестирование «белого ящика» и детальную отладку.

Интерфейс взаимодействия это не только набор текстур и спрайтов, но и элементы кода. Основные программные элементы интерфейса представлены в таблице 1.

Таблица 1. Программные элементы интерфейса

|  |  |
| --- | --- |
| Элемент | Описание |
| SKS файл | Файл разметки интерфейса, который содержит XML конфигурацию. |
| SKNode | Нулевой элемент – родитель любого элемента интерфейса. Каждый объект на экране является производным объектом от SKNode. |
| SKSpriteNode | Графический объект, содержащий текстуру. |
| UINode | Объект интерфейса, который реализует взаимодействие с объектами (нажатие, наведение, перемещение). |
| UIGroup | Группа элементов для взаимодействия. |
| UIScene | Сцена, позволяющая размещать объекты и взаимодействовать с ними. |
| UINavigator | Объект, осуществляющий переключение между сценами UIScene. |

* + 1. Обращение к программе

Для запуска программы необходимо запустить приложение “FunnyBoxes” на любом из устройств, на котором оно поддерживается (см. Системные требования).

* + 1. Входные и выходные данные

Входными данными являются действия, совершенные с помощью мыши или клавиатуры. Они обрабатываются внутри программы и отправляют выходные данные – реакцию на совершенный пользователем процесс.

Реакцией может быть любое изменение в пользовательском интерфейсе или активация фоновой задачи. Например, при касании клавиш в меню, происходит смена сцены.

* 1. Руководство программиста
     1. Общие сведения о программе

Программный продукт включает в себя программную логику, реализованный с использованием Xamarin и ресурсы. Ресурсы включают в себя:

* Спрайты ящиков, бомбы, платформы
* Графические ящиков, бомбы, платформы
* SKS файлы игровых сцен

Программа собрана с использованием системного Apple App Bundler, что приводит к созданию единого пакета со своей системой внутри него. Приложение может использовать исключительно директории временного хранилища данных или место внутри приложения.

* + 1. Системные требования

Система: macOS

Процессор: 2.0 GHz и выше

Оперативная память: Минимум 1 ГБ

Видеокарта: Intel Iris Graphics, Nvidia

Для получения максимальной производительности, рекомендуется процессор Intel Core i5.

* + 1. Программные требования

Для запуска игры необходим предустановленный NET Framework 4.5 и выше, а также от 100МБ свободного места на диске.

NET Framework необходим для непосредственного запуска приложения, так как оно написано на языке C#.

* + 1. Файлы программы

Программа состоит из одного файла – Apple App Bundle и не требует установки. Данная система разработана для macOS, так как, обычно, на «яблочных системах», программы не устанавливаются, а просто загружаются в директорию Applications.

Apple App Bundle содержит в себе необходимые ресурсы приложения – графические элементы игры: основные текстуры и спрайты для объектов игры.

* + 1. Настройка программы

Так как приложение скомпилировано через систему Apple App Bundler, оно не нуждается в какой-либо предустановке или настройке. Всё необходимое уже запаковано внутри файла “FunnyBoxes.app”.

На текущем этапе разработки проекта, у пользователя нет возможностей какой-либо модернизации или настройки приложения внутренними утилитами.

* + 1. Проверка программы

Результатом отлаженных алгоритмов и оптимизованного кода является высокий показатель FPS (Frames Per Second) счетчика и плавность анимаций. В любом случае, когда, во время выполнения кода программы, возникает ошибка, то «фреймрейт» стримится к нулю, что приводит к замедленной анимации и отрисовки в целом.

* 1. Руководство пользователя

В начале игры нас встречает главное меню, выраженное в оптимистичных гаммах.

В меню есть 2 главные кнопки: Играть (Рисунок 7) и Выход (Рисунок 10)



Рисунок 7 – Кнопка играть



Рисунок 8 – Кнопка выход

Для начала игры нажмите кнопку «Играть» игра перейдет на игровую сцену (Рисунок 9) где и начнется процесс игры. В левой стороне в верху экрана написаны ваши очки. Ваша задача поймать и удержать наибольшее количество ящиков на платформе за отведенное время.

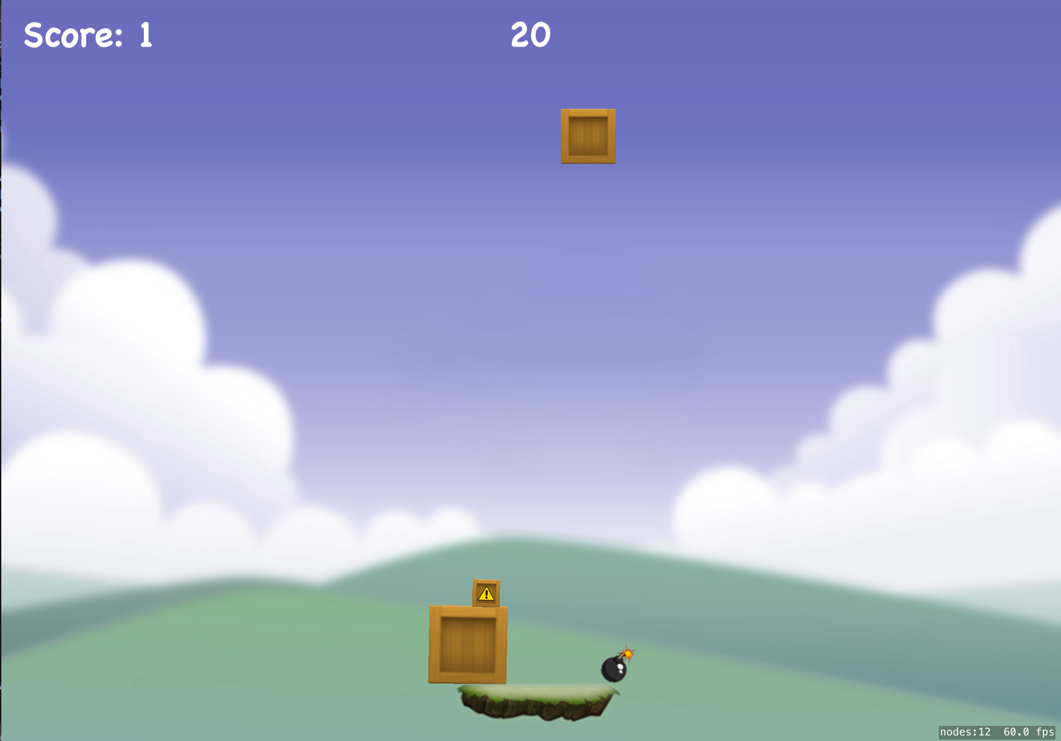


Рисунок 9 – Игровой экран

Нас встретят 4 основных объекта. Первый это платформа.

Платформа – это ваш объект управления, передвигайте мышь влево/вправо чтобы ее двигать.

FunnyBoxes/Mac/Resources/TexturePacks/Default/platform.png

Рисунок 10 - Платформа

С неба повалятся ящики и бомбы. Вы должны поймать и удержать как можно больше ящиков (Рисунок 11).

FunnyBoxes/Mac/Resources/TexturePacks/Default/box.png

Рисунок 11 – Ящик

Старайтесь не собирать фейк-ящики(Рисунок 12). Фейк-ящик представляет собой тот же ящик, но с отличием, в данном случае отличием является восклецательный знак на нем. Фейк-ящик при сопрекосновении с ящиком (Рисунок 11) уничтожает ящик и уничтожается сам.

FunnyBoxes/Mac/Resources/TexturePacks/Default/box-fake.png

Рисунок 12 – Фейк-ящик

Так же нежелательно ловить бомбы (Рисунок 13). Бомбы разбрасывают ваши ящики на платформе в стороны, так же есть вероятность увеличить таймер в плюс.

FunnyBoxes/Mac/Resources/TexturePacks/Default/bomb.png

Рисунок 13 – Бомба

После того, как таймер вышел, Ваш результат будет показан на экране конца игры (Рисунок 14) и запишет рекорд на Ваш никнейм.



Рисунок 14 – Окно конца игры

* + 1. Основная терминология проекта
* Игровое пространство - плоскость, на которой происходит игровой процесс.
* Платформа – объект управления, которым нужно ловить объекты.
* Ящик - объект, который нужно ловить и удерживать на платформе.
* Фейк-ящик – объект который при сопрекосновении с ящиков уничтожает его и уничтожается сам.
* Бомба – объект, который при сопрекосновении с ящиками разбрасывает их, а так же есть шанс получить прибавку к таймеру.
* Очки – количество собраных ящиков на платформе в данную секунду таймера, очки изменяются в процессе игры.
* Таймер – время, за которое нужно собрать и удержать как можно больше ящиков.

# Заключение

Игра Funny Boxes – это не только игра «убивалка времени», но и опыт разработки, непосредственно объединенный с практикой дизайна. Некоторые решения принесли игре новизну, т.к. есть полно подобных игр, но именно с такой идеей еще не встречал. Главная идея игры, удержать как можно большое количество ящиков уворачиваясь от посторонних объектов.

Во время разработки дизайна, были рассмотрены такие инструменты, как Adobe Photoshop.

Работа над кодом – не менее сложная часть разработки игры. Использованы инструменты: Xcode, Visual Studio, Xamarin[1], SpriteKit[2]. Построена простая игровая логика. Разработана и оптимизирована физика в игре.

Во время разработки, также возникли сложные вопросы связанные с физикой игры и расходованием ресурсов процессора. Была применена огромная работа по оптимизации целой системы игры – прочитано много статей [1][2][3].

Данная стадия игры не является завершенным вариантом, игра находится на стадии альфа-тестирования. На данном этапе реализована основная маторика игрового процесса и системы экранов меню, написан итоговый вариант логики, который в будущем будет дорабатываться.

Список используемых источников

Интернет ресурсы:

1. Develop Center Xamarin – Документация для работы с инструментами NET Framework Xamarin.
2. Develop Center Apple – Документация для работы с нативными инструментами.
3. Stack Overflow - популярная [система вопросов и ответов](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D0%B2%D0%BE%D0%BF%D1%80%D0%BE%D1%81%D0%BE%D0%B2_%D0%B8_%D0%BE%D1%82%D0%B2%D0%B5%D1%82%D0%BE%D0%B2_(%D0%B2%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B8%D1%81)) о [программировании](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5), разработанная [Джоэлем Спольски](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%BE%D0%BB%D1%8C%D1%81%D0%BA%D0%B8,_%D0%94%D0%B6%D0%BE%D1%8D%D0%BB) и [Джеффом Этвудом](https://ru.wikipedia.org/w/index.php?title=%D0%AD%D1%82%D0%B2%D1%83%D0%B4,_%D0%94%D0%B6%D0%B5%D1%84%D1%84&action=edit&redlink=1) ([англ.](https://en.wikipedia.org/wiki/Jeff_Atwood)) в 2008 году

# Приложение А

# Приложение Б

![](data:application/pdf;base64,)