Министерство образования и науки Украины

Харьковский национальный университет имени В.Н. Каразина

Факультет компьютерных наук

Курсовая работа на тему:

«Flappy Bird»

Выполнил:

студенты группы КИ-21

Анжуров Валентин

Проверил:

старший преподаватель

Литвинов Д. Н.

Харьков - 2016

**Оглавление**

Введение…………………………………………………………………….…......…...3 КОНЦЕПЦИЯ………………………………………………………………….........….....4

* 1. Общая характеристика продукта…………………………......…….…..…4
  2. Основные возможности и ограничения……………..……………..…..….4

АНАЛИЗ………………………………………………………………………...…....……5

2.1. Состав системы…………………………..…………….………….…...…..5

2.2. Описание Use Case’ов……………………………………..………......…….5

ПРОЕКТИРОВАНИЕ………………………………………………...…………..…...…..8

3.1. Общее описание архитектуры………………….…………………......….....8

3.2. Классы предметной области……………………………………....…..…….8

3.4. Классы UI……………………………………………………....….……..…..8

ТЕСТИРОВАНИЕ………………………………....……………………...…....…...…...10

Выводы………………………………………..………………...…………....……......11

ПриложениЯ……………………………………..……………………………..…….12

Введение

Когда передо мной стал вопрос выбора темы курсовой роботы, я не думая выбрал тему компьютерной игры. Во-первых я посчитал это хорошей возможностью отточить свои навыки ООП и закрепить пройденный в этом семестре материал. Ну а во-вторых, я просто посчитал это более интересным, чем написание каких бы то ни было утилит для решения тривиальных задач. Из компьютерных игр я выбрал Flappy Bird, так как версий под платформы IOS и Android несчетное множество, а вот под Windows аналогов в интернете я не встречал. После выбора темы я приступил к написанию кода. Во время разработки были использованы: среда программной разработки IDE IntellijIdea, сервис для хостинга IT-проектов GitHub, Microsoft Word, онлайн-редактор диаграмм Creately и простой графический редактор Paint для создания внутренне-игровой графики. Данные средства я счел наиболее удобными и простыми для разработки конкретно этого проекта. Всего от создания проекта до его финальной версии было 20 итераций, где каждая итерация представляла собой введение новой и полностью рабочей функции + поправлялись некоторые баги прошлых итераций(каждая занимала от пары часов до пары дней). При работе постоянно возникали сложности, которые решались по мере написания кода. После завершения проекта были сделаны следующие выводы:

* Объектно-ориентированное программированиеявляется отличным способом разработки как крупных, так и малых проектов. Имеет множество преимуществ над процедурно-ориентированными.
* Представление архитектуры проекта и взаимодействий компонента является не менее важной частью разработки, чем написание кода.

Концепция

**1.1 Общая характеристика**

Цель создания – разработка приложения для повседневного пользования, в котором можно выполнять различные задания. Отличительной чертой моего проекта стало то, что в него было добавлены такие возможности, которых не имеют аналоги в Интернете. А в целом это игра с той же логикой и игровым процессом

**1.2 Возможности и ограничения**

Возможности:

1. Игра и прохождение встроенных в игру заданий
2. Выбор оформления, более близкого к вкусу пользователя
3. Ведение таблицы рекордов

Ограничения:

1. Только одиночная игра
2. Отсутствие редактирования таблицы рекордов

Анализ

* 1. **Состав системы**

1. Основным предметом взаимодействия является окно размеров 540х720, размещенное в центре монитора.
2. Классы Bird, Tube, Strip, Score, Record являются базовыми классами, на которые ориентируются остальные классы графики и логики. Классы MenuPanel, GamePanel, RecordsPanel, LosePanel, SettingsPanel, NewRecordPanel, ThemePanel являются классами, которые реализуют графику и содержат кнопки, которые в свою очередь переключают окна с их содержимым.
   1. **. Описание Use Case’ов**

1.Меню – Рис.1

* 1. Игра – Рис.2

С помощью мыши юзер управляет птичкой и проходит движущиеся на нее трубы.

1.1.1 Пауза – Рис.3

Игровой процесс останавливается и высвечивается окно, которое сообщает, что игра приостановлена, а также показывает условия выполнения текущего задания

* + 1. Проигрыш – Рис.4

При проигрыше высвечивается окно, отображающее набранные очки, рекорд и предоставляется возможность повторить игру либо вернуться в главное меню

* + 1. Установка нового рекорда – Рис.5

Если установлен новый рекорд, то соответствующее окно сообщает, что рекорд побит и нужно ввести свой никнейм для дальнейшего отображение рекорда в таблице, а затем переводит в вышеуказанное поле

* 1. Настройки – Рис.6

1.2.1 Включение/отключение звуков

1.2.2 Выбор сложности игры

3 уровня сложности, которые меняют размер промежутка в трубах

1.2.3 Выбор игрового оформления – Рис.7

Доступно 3 оформления, 1 из которых становится доступным после выполнения всех заданий, предусмотренных разработчиком

* 1. Просмотр таблицы рекордов – Рис.8

Можно увидеть максимальные набранные очки, а также имена игроков, которые эти очки набрали

* 1. Закрытие приложения

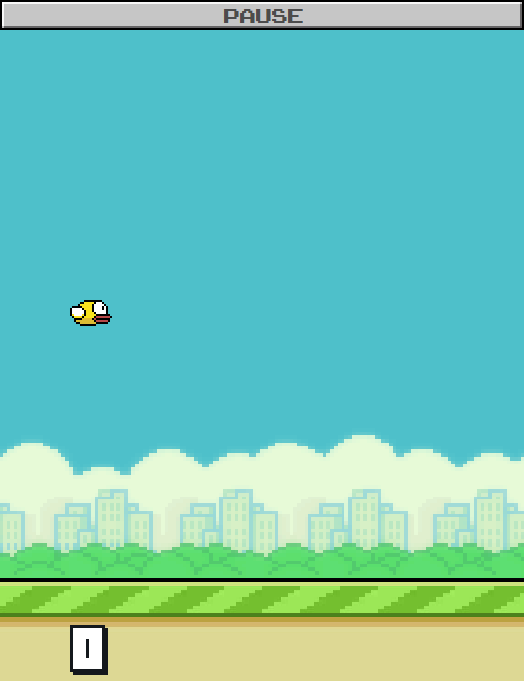
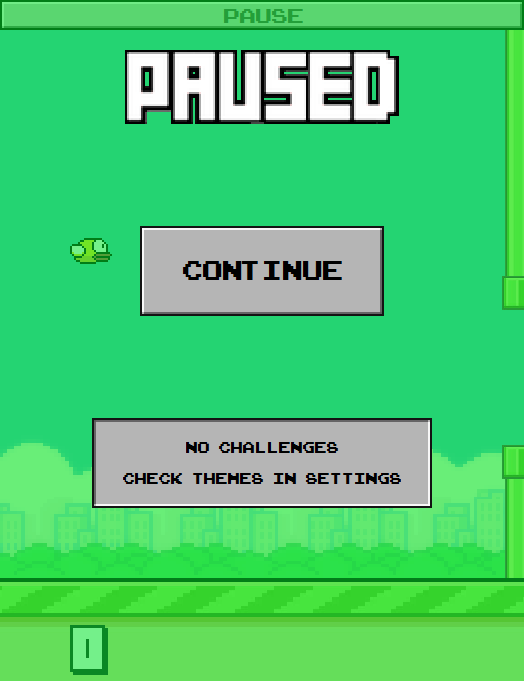
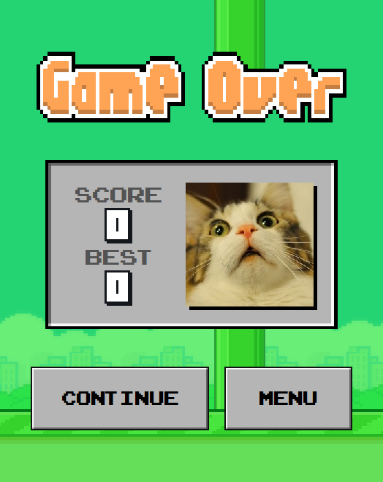
   

Рис.1 Рис.2 Рис.3 Рис.4

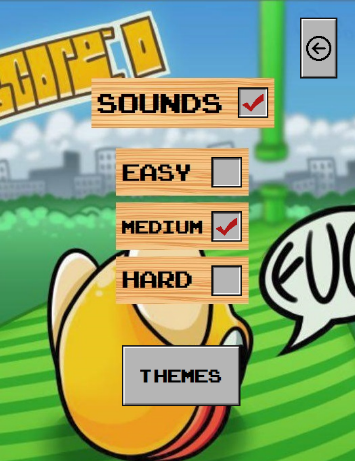
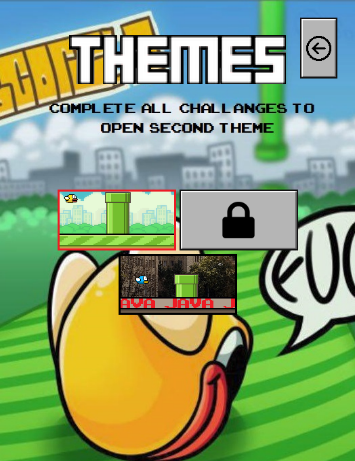
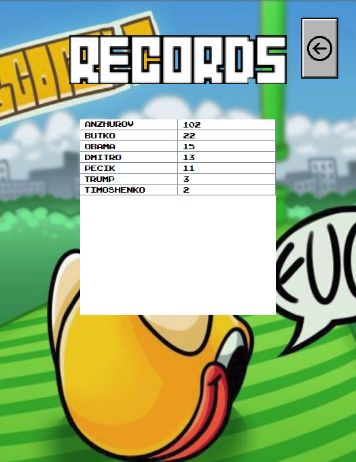
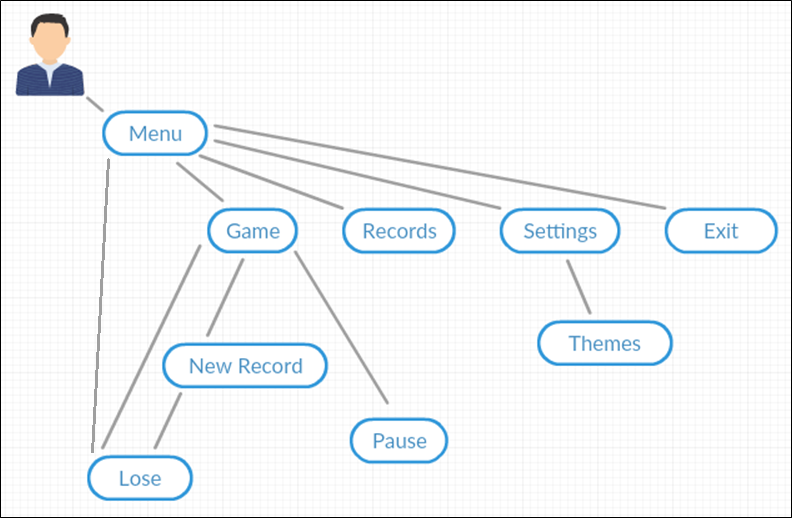
   

Рис.5 Рис.6 Рис.7 Рис.8

Для большего представления ниже приведена use case диаграмма.



ПРОЕКТИРОВАНИЕ

**3.1 Общее описание архитектуры**

Цели проектирования

1. Отделить бизнес-логику от пользовательского интерфейса
2. Window – единственное окно программы, которое показывает все остальные части приложения
3. Массивы RecordsArray и ChallengesArray, которые хранят рекорды/задания и работают с файлами для сохранения результатов
   1. **Классы предметной области**

Bird, Tube, Strip: объекты этих классов являются потоками, которые изменяет положение соответствующих элементов на экране

Score: хранит состояние игрового счета

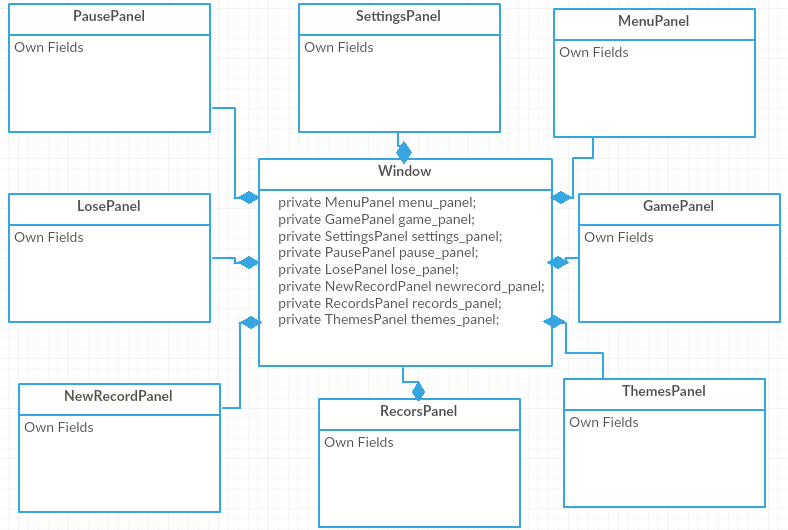
Challenge, Record: хранит 1 задание/рекорд, из которых потом создается соответствующий массив

* 1. **Классы UI**

MenuPanel, GamePanel, RecordsPanel, LosePanel, SettingsPanel, NewRecordPanel, ThemePanel:

агрегируют классы предметной области и отображают их.

Для большего представления ниже представлена диаграмма классов UI



Тестирование

Так как программа состоит из нескольких классов, то и тестировались сперва классы по отдельности, затем - некоторые классы в связке, после этого проверялась работоспособность всей системы. И уже после всех успешно проведенных тестов настал черед для тестирования GUI вкупе с другими компонентами системы.

1. Проверка работы методы crush(Tube tube) из класса bird, который превряет столкновения.
2. Проверка правильности заполнения массивов данными из фалйла, а также перезаписи

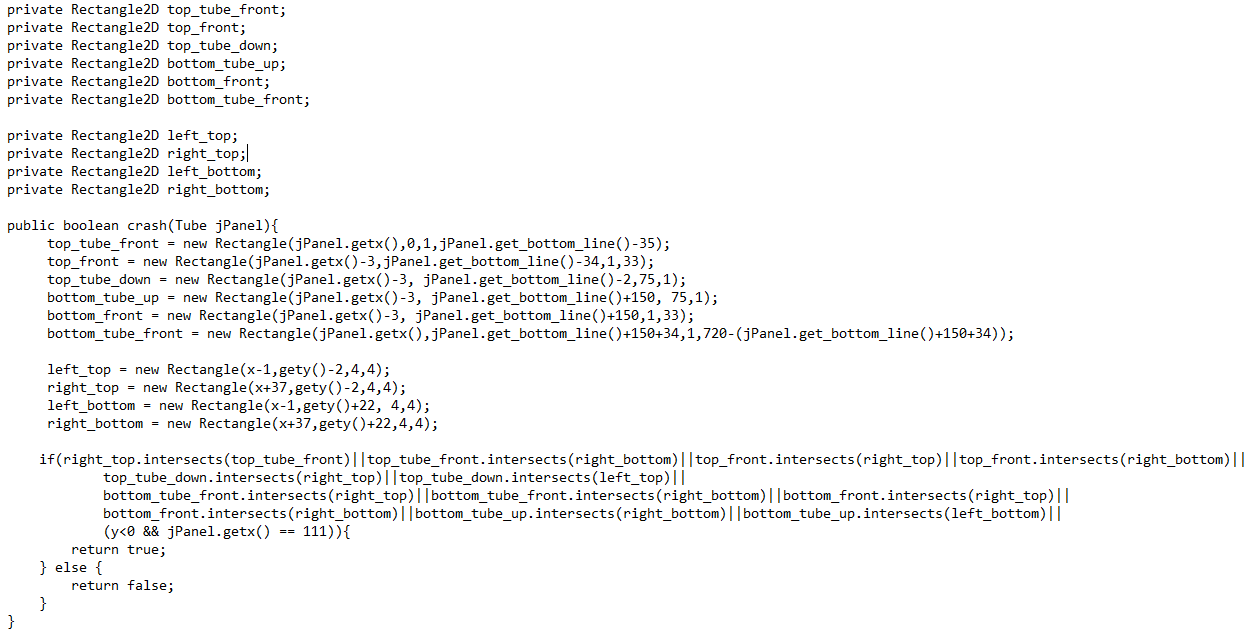
Выводы

После выполнения данного проекта я получил, по моему мнению, неплохой опыт работы с ООП и его компонентами. Удалось почувствовать себя неким работником, перед которым есть чекто поставленные требования и сроки. В результате получил готовый продукт, который реализует всё, чтобы было задумано еще перед началось разработки. А задумывалось всего-то создать приложение для развлечения с фичами, которых не имеют аналоги. Разумеется программу еще можно и нужно развивать. Имея логику и структуру можно переписать ее под другие OS на любых доступных движках, изменить вид, работая в паре с 3d-дизайнером, добавить новые возможности и многое много другое.

Огромным плюсом считаю то, что теперь я намного лучше оперирую средствами весьма популярной среди разработчкиков любых направлений Intellij IDEA и хорошо ориентируюсь в системе GIT, что несомненно пригодится в будущей карьере. Также получил опыт тестирования программы и отладки кода

Приложения

Фрагменты кода. Тут я хотел бы вкратце описать работу метода crush(Tube tube), который отлавливает столкновения.



Создаются прозрачные прямоугольники вокруг птички, а также в конкретных частях трубы. В потоке игры этот метод возвращает true, если прямоугольники птицы накладываются на прямоуголники труб и false в противном.

