

Федеральное государственное образовательное учреждение  
Высшего профессионального образования  
«Омский государственный технический университет»

Кафедра «Информатика и вычислительная техника»

## ОТЧЕТ ПО РАСЧЕТНО-ГРАФИЧЕСКОЙ РАБОТЕ

на тему:

«Разработка игры мозаика»

по дисциплине:

«Проектирование и тестирование программного обеспечения»

Принял: руководитель работы к.т.н.,  
доцент кафедры ИВТ

Калекин Д.В.

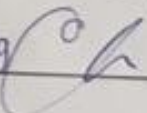
29.05.19.



Выполнил: студент группы МО-171

Ермакова Т.А.

29.05.19.



Омск 2019

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Анализ задания и формулировка задач на проектирование.....	4
2 Разработка программы.....	5
3 Результаты тестирования.....	9
ЗАКЛЮЧЕНИЕ.....	13
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	14

## ВВЕДЕНИЕ

В связи с доступностью компьютерных технологий в современном мире игры стали популярным досугом для людей любого возраста. Они могут быть установлены на любом доступном девайсе, например, на компьютере, смартфоне или планшете. Кроме того, игры могут помогать осваивать различные знания, но чаще всего они используются именно как развлечение.

Современная мозаика гармонично сочетает в себе секреты древних мастеров и современные промышленные технологии. Мозаика красива, практична и может использоваться в помещениях разного назначения, а в нашем случае - мозаика будет отличным развлечением и поспособствует развитию мышления.

Данный проект актуален, так как полученный результат приложения можно будет в дальнейшем использовать в повседневной жизни.

Задачи проекта: изучить возможности языка C#, среды разработки компьютерных игр и реализовать алгоритм игры, а также провести тестирование и сделать выводы о результате проделанной работы.

## **1 Анализ задания и формулировка задач на проектирование**

Целью расчетно-графической работы является закрепление теоретических знаний и отработка практических навыков программирования. Для реализации поставленной цели, необходимо решить следующие задачи: разработать игру мозаика, а также меню для взаимодействия с пользователем. Данная работа должна включать в себя следующие элементы:

- Меню;
- Игру мозаика.

Выполнение каждого элемента осуществляется выбором соответствующего пункта в меню.

В меню содержатся кнопки «Картинка», «Перемешать», «Восстановить», «Помощь», «Настройки».

С помощью кнопки «Картинка» пользователь должен самостоятельно выбрать на компьютере изображение, которое он хотел бы собрать в игре мозаика.

## 2 Разработка программы

Необходимо разработать меню, с помощью которого пользователь сможет взаимодействовать с приложением. Для решения поставленной задачи была разработана схема алгоритма работы главного меню, которое объединяет все поставленные задачи в один проект.

Меню реализовано в виде панели- бесконечного цикла выбора одного из пунктов посредством нажатия левой кнопкой мыши на один из пунктов программы. Для разработки приложения была использована среда разработки VisualStudio.

Требованием к аппаратному и программному обеспечению является наличие любого IBM-совместимого компьютера с накопителем гибких и жестких магнитных дисках и достаточным количеством оперативной памяти. Предпочтительной операционной системой является система Windows.

В процессе разработки своего приложения я пользовалась Windows Forms.

Windows Forms позволяет разрабатывать интеллектуальные клиенты. Интеллектуальный клиент — это приложение с полнофункциональным графическим интерфейсом, простое в развертывании и обновлении, способное работать при наличии или отсутствии подключения к Интернету и использующее более безопасный доступ к ресурсам на локальном компьютере по сравнению с традиционными приложениями Windows.

Windows Forms — это технология интеллектуальных клиентов для платформы .NET Framework, набор управляемых библиотек, упрощающих выполнение распространенных задач, таких как чтение и запись в файловой системе. При использовании среды разработки, как Visual Studio, можно создавать приложения интеллектуальных клиентов Windows Forms, которые отображают информацию, запрашивают ввод от пользователей и взаимодействовать с удаленными компьютерами по сети.

В Windows Forms форма — это визуальная поверхность, на которой выводится информация для пользователя. Обычно приложение Windows Forms

строится путем помещения элементов управления на форму и написания кода для реагирования на действия пользователя, такие как щелчки мыши или нажатия клавиш. Элемент управления — это отдельный элемент пользовательского интерфейса, предназначенный для отображения или ввода данных.

Для наиболее продуктивного процесса разработки программного средства необходимо определить основные функции системы:

- загрузка изображения
- генерация расположений квадратов
- создание квадратов
- изменение количества разбиений на квадраты
- помощь пользователю

Объектно-ориентированное проектирование не является восходящим или нисходящим по своему типу, но в системе можно выделить начальное и конечное состояния. Анализ системы следует начать с жизненного цикла.

Начальным этапом является создание (открытие) изображения, полученное путем сохранения в любое доступное для пользователя место и открытия (выбора) в появившейся форме. Когда изображение загружено, оно сжимается или расширяется согласно размерам заданной формы, поэтому изображение, не соответствующее размерам формы, может иметь искаженный вид и/или ухудшенное качество. В связи с этим следует внимательно отнестись к выбору загружаемого изображения, чтобы игра-пазл доставляла лишь положительные эмоции.

Когда изображение уже загружено, с ним можно делать несколько операций:

- разбить на квадраты
- сгенерировать (перемешать) квадраты
- изменить количество разбиений
- восстановить в исходный вид

Дополнительно для пользователя есть возможность помощи, заключающейся в дополнительном окне с фотографией собираемого изображения, чтобы наглядно видеть, какой из квадратов должен быть на каком месте.

Проект программы содержит следующие модули:

- FormMain.cs
- FormMain.Designer.cs
- HelpDlg.cs
- HelpDlg.Designer.cs
- Program.cs (содержит точку входа в программу)
- SetDlg.cs
- SetDlg.Designer.cs

При выполнении пользователем какого-либо действия с формой или одним из ее элементов управления создается событие. Приложение реагирует на эти события с помощью кода и обрабатывает события при их возникновении.

Основное окно программы, изображенное на рисунке 1, представляет собой пустую область с панельным меню сверху.

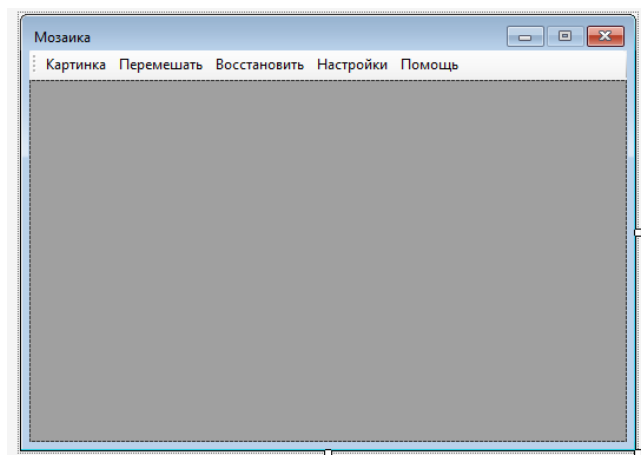


Рисунок 1 – Основное окно программы

Конструктор окна помощи представлен на рисунке 2. Пока картинка не загружена, форма пустая и залита зеленым фоном.



Рисунок 2 – Окно помощи пользователю

Конструктор настроек, в которых указывается число разбиений, представлен на рисунке 3. Начальное значение (по умолчанию) установлено и равно трем (то есть  $3 \times 3 = 9$  прямоугольников, один из которых пустой).

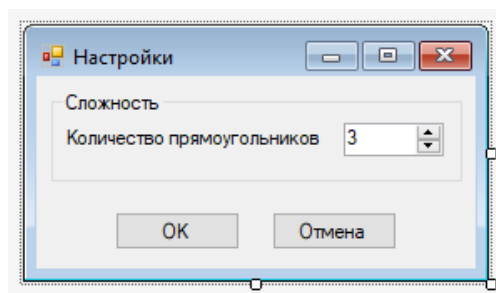


Рисунок 3 – Окно настроек



### 3 Результаты тестирования

Тестирование программы проходило при ее отладке, при тестировании проверялись функциональные возможности подсистем и отдельных инструментов. С целью тестирования в данной игре- мозаике создавалось графическое приложение и осуществлялась загрузка изображения в него при помощи многократного использования различных инструментов. Проверялась работа приложения при выборе каждого пункта панельного меню, а также, в случае выбора меню настроек, проверялось изменение программы при изменении параметров.

Тестирование показало, что программа должным образом реагирует на события от мыши, обрабатывая их определенным способом и производя заданные действия.

Результат тестирования работы приложения игра- мозаика представлен на следующих рисунках.

На рисунке 4 представлен результат загрузки изображения с компьютера в приложение.



Рисунок 4 – Загрузка изображения с компьютера в приложение

Передвигаясь по панели меню далее и нажимая на кнопку «перемешать», получим старт нашей игры, представленный на рисунке 5.

При нажатии данного пункта меню, происходит главный процесс и запускается основной механизм программы. Прямоугольники располагаются случайным образом, но при этом исключается вариант совпадения изображения с начальным. Все прямоугольники разделены тонкой линией в цвет серого фона. Один прямоугольник, чей порядковый номер также определяется случайным образом, остается пустым.

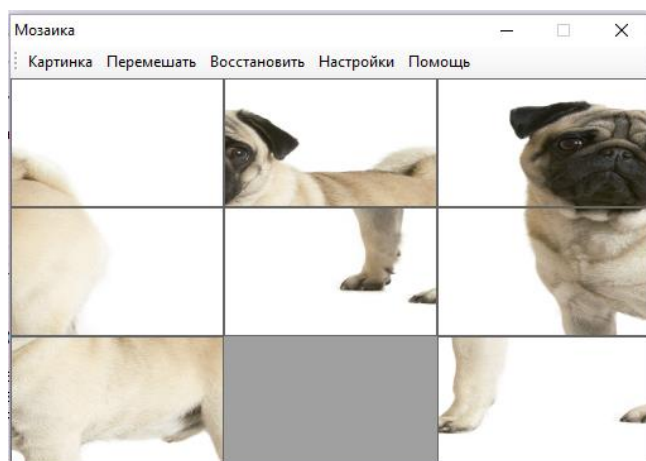


Рисунок 5 – Старт игры

Нажимая кнопку «восстановить», получаем исходное изображение, также разделенное тонкими линиями. Результат выбора данного пункта меню представлен на рисунке 6.

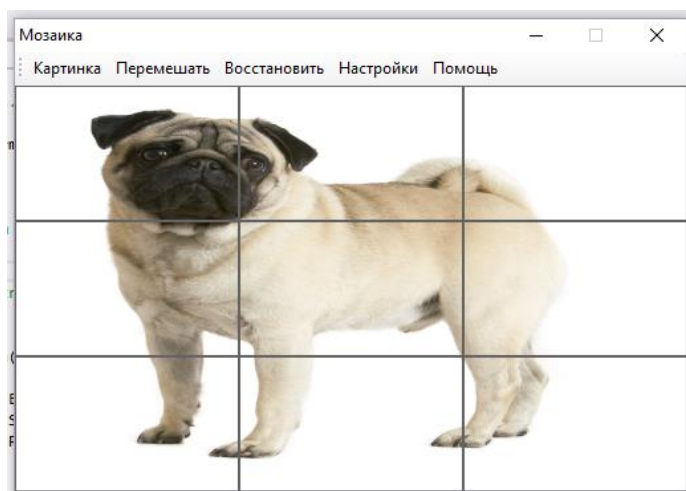


Рисунок 6 – Результат восстановления изображения в первоначальное состояние

Двигаясь далее и выбирая пункт меню настройки, можно регулировать количество разбиений нашего изображения. На рисунке 7 представлен результат выбора пункта меню настройки, а на рисунке 8- результат изменения и применения изменений на значение 4. В ходе написания программы не имело смысла указывать общее количество разбиений (например, 9, как в первоначальном условии), так как пользователь может выбрать не подходящее для игры значение и вызовет ошибку при сборке программы или некорректную работу приложения.

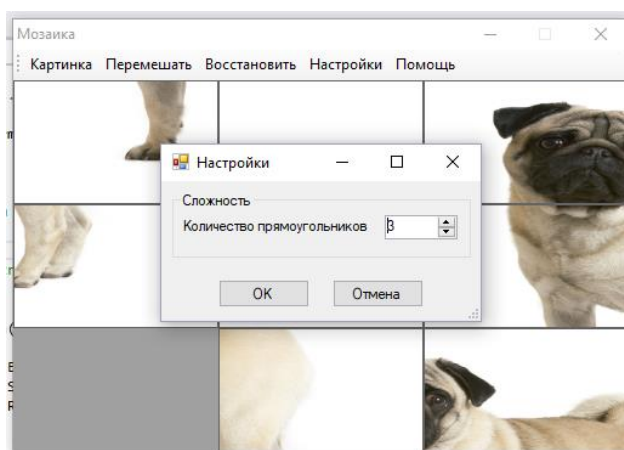


Рисунок 7 – Результат выбора пункта меню настройки



Рисунок 8 – Результат изменения и применения изменений на значение

4

При выборе пункта меню помощь появляется новое окно с исходным цельным изображением, при этом можно продолжать собирать моза

ику не закрывая это окно. Результат представлен на рисунке 9.

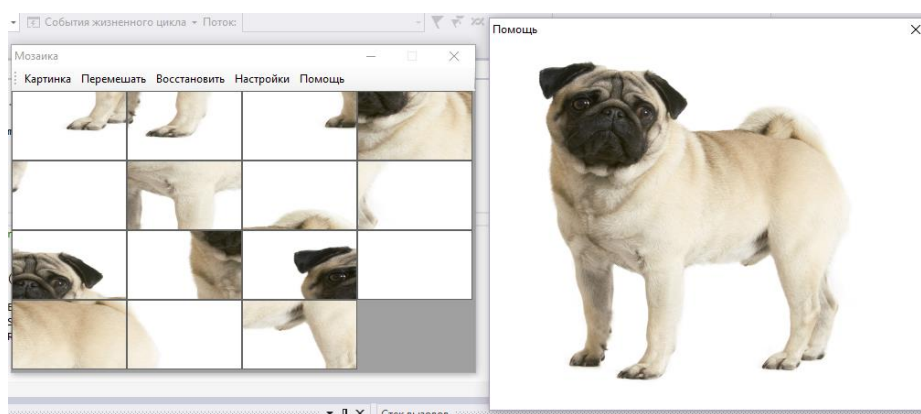


Рисунок 9 – Результат выбора пункта меню помощь

## ЗАКЛЮЧЕНИЕ

Программа, описанная в расчетно-графической работе, разработана в соответствии с поставленной задачей по теме «Разработка игры мозаика» по дисциплине «Проектирование и тестирование программного обеспечения».

Интерфейс созданной программы удобен, прост, наглядно отображает ее возможности. Главное меню представлено в виде верхней панели, содержит команды работы с файлами, команды занесения изображения в приложение.

Тестирование подтвердило, что программа корректно выполняет обработку данных и демонстрацию результатов.

Все это свидетельствует о работоспособности программы и позволяет сделать вывод о ее пригодности для собирания мозаики из загруженных изображений.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Албахари, Джозеф С# 3.0. Справочник / Джозеф Албахари, Бен Албахари. – М.: БХВ-Петербург, 2012. – 944 с.
2. Вагнер, Билл С# Эффективное программирование / Билл Вагнер. – М.: ЛОРИ, 2013. – 320с.
3. Зиборов, В.В. Visual С# 2012 на примерах / В.В. Зиборов – М.: БХВ-Петербург, 2013. – 480 с.
4. Ишкова, Э. А. Самоучитель С#. Начала программирования / Э. А. Ишкова. – М.: Наука и техника, 2013. – 496 с.
5. Подбельский, В.В. Язык С#. Базовый курс / В.В. Подбельский – М.: Финансы и статистика, Инфа-М, 2011. – 384 с.