Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

Дисциплина: Программирование на языках высокого уровня

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе на тему

Компьютерная игра «Пазлы»

Студент Кузьмин Д. С.

Руководитель Богдан Е. В.

МИНСК 2023

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

*––––––––––––––––––––––––*

(подпись)

––––––––––––––––––2023 г.

ЗАДАНИЕ

по курсовому проектированию

Студенту *Кузьмин Дмитрий Сергеевич*

1. Тема проекта *Компьютерная игра «Пазлы»* –– ––––
2. Срок сдачи студентом законченного проекта *15 декабря 2023 г.–––*
3. Исходные данные к проекту: картинки png для кусочков пазлов в папке paint
4. Содержание расчетно-пояснительной записки (перечень вопросов, которые подлежат разработке)

1. Введение.

1. Задание.
2. Обзор литературы.
   1. Обзор методов и алгоритмов решения поставленной задачи.
3. Функциональное проектирование.
   1. Структура входных и выходных данных.
   2. Разработка диаграммы классов.
   3. Описание классов.
4. Разработка программных модулей.
   1. Разработка схем алгоритмов (два наиболее важных метода).
   2. Разработка алгоритмов (описание алгоритмов по шагам для двух методов).
5. Результаты работы.
6. Заключение
7. Литература
8. Приложения

5. Перечень графического материала (с точным обозначением обязательных чертежей и графиков)

*1. Диаграмма классов. ––––––––––––––––––––––––––––––*

*2. Схема алгоритма RandTabels().*

*3. Схема алгоритма animation().––––––––––––––––––––––––––––––*

6. Консультант по проекту (с обозначением разделов проекта) Е.В.Богдан

1. Дата выдачи задания 15 *сентября 2023 г.–*

*––––––––––––––––––––– –*

1. Календарный график работы над проектом на весь период проектирования (с обозначением сроков выполнения и трудоемкости отдельных этапов):

*1. Выбор задания. Разработка содержания пояснительной записки. Перечень графического материала – 15 %; ––––––––––––––––––––––––––––\_\_\_\_\_\_\_\_\_* *разделы 2, 3 – 10 %; ––––––––––––––––––––––––––––––––––––– \_\_\_\_\_\_\_\_\_\_*  *разделы 4 к –20 %; –––––––––––––––––––––––––––––––––––––––\_\_\_\_\_\_\_\_\_* *разделы 5 к – 35 %; –––––––––––––––––––––––––––––––––––––\_\_\_\_\_\_\_\_\_\_\_–* *раздел 6,7,8 – 5 %; ––––––––––––––––––––––––––––––––––––––\_\_\_\_\_\_\_\_\_\_–* *раздел 9 к – 5%; ––––––––––––––––––––––––––––––––––––––––– \_\_\_\_\_\_\_\_*  *оформление пояснительной записки и графического материала к 15.12.23 –*

*10 %*

*Защита курсового проекта с 21.12 по 28.12.23г.–––––––––––––––––––––––––* РУКОВОДИТЕЛЬ Е.В.Богдан

#### *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

Задание принял к исполнению \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.С.Кузьмин

(дата и подпись студента)

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ…………………………………………………………………………5

[1 ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ](#_Toc152195038) 8

[2. ОБЗОР ЛИТЕРАТУРЫ 10](#_Toc152195039)

[2.1 ОБЗОР МЕТОДОВ И АЛГОРИТМОВ РЕШЕНИЯ 12](#_Toc152195040)

[ПОСТАВЛЕННОЙ ЗАДАЧИ 12](#_Toc152195041)

[3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВНИЕ 15](#_Toc152195042)

[4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ 19](#_Toc152195043)

[4.1 Разработка схем алгоритмов 19](#_Toc152195044)

[4.2 Разработка алгоритмов 20](#_Toc152195045)

[5 РЕЗУЛЬТАТ РАБОТЫ 22](#_Toc152195046)

[ЗАКЛЮЧЕНИЕ 26](#_Toc152195047)

[СПИСОК ЛИТЕРАТУРЫ 29](#_Toc152195048)

[ПРИЛОЖЕНИЕ А 30](#_Toc152195049)

[ПРИЛОЖЕНИЕ Б 31](#_Toc152195050)

[ПРИЛОЖЕНИЕ В 32](#_Toc152195051)

[ПРИЛОЖЕНИЕ Г 33](#_Toc152195052)

[ПРИЛОЖЕНИЕ Д 34](#_Toc152195053)

**ВВЕДЕНИЕ**

С++ - это высокоуровневый язык программирования, который благодаря своей мощности и гибкости часто используется для разработки различных видов программного обеспечения и имеет несколько важных аспектов:

* Поддержка объектно-ориентированного программирования(ООП), которое позволяет создавать программы, основанные на объектах, а не на процедурах, что делает код модульным и легко поддерживаемым.

Это позволяет создавать более гибкие и масштабируемые приложения;

* Высокая производительность. С++ обеспечивает эффективное управление ресурсами, что позволяет создавать приложения, которые критически важны для оптимизации работы;
* Стандартные библиотеки. С++ предлагает стандартные библиотеки (Libpng, STL и др.), которые содержат широкий спектр функций и контейнеров, что значительно облегчает разработку кода;
* Управление памятью. Ручное управление памятью даёт разработчику полный контроль над ресурсами компьютера, что полезно для работы с системами, которые количеством этих ресурсов ограничены;
* Кросс-платформенность. С++ позволяет создавать приложения, которые могут быть совместимы с несколькими операционными системами;

Итак, изучение объектно-ориентированного программирования (ООП) показало мне возможности создания серьёзных приложений и увеличило моё понимание принципов ООП как таковых. Погружение в мир ООП расширило мои знания и, обогатив мой опыт в сфере программирования.

Одним из ключевых выводов, которые я сделал в процессе написания курсовой работы, является то, что ООП предоставляет как эффективные средства организации кода, так и обеспечение более высокого уровня абстракции, из-за чего разработка программ становится интуитивной и гибкой. Использование классов и объектов позволяет создавать модульные, легко читаемые и поддерживаемые на долгий срок системы.

В наше время, по причине роста требований к программному обеспечению, ООП становится обязательной частью обучения разработчика. Модульность кода, возможность расширения функционала проектов и повторного использования компонентов становятся ключевыми аспектами успешной разработки. ООП предоставляет огромный спектр возможностей для разработки сложных, но гибких систем, способных адаптироваться к изменяющимся требованиям.

Принципы инкапсуляции, наследования и полиморфизма, вытекающие из ООП, являются своего рода основой для создания устойчивого, расширяемого и гибкого кода. Понимание этих принципов не только повышает качество программного обеспечения, но и упрощает внесение изменений и обеспечивает долгосрочную поддержку проектов.

По итогу, изучение ООП открывает перед разработчиком множество возможностей для создания инновационных и эффективных приложений. ООП становится базовым инструментом для современного разработчика, обеспечивая не только технические навыки, но и фундаментальное понимание принципов, необходимых для успешной разработки программного обеспечения.

Пазлы – игра-головоломка, в которой требуется составить фигуру из множества её фрагментов, в классическом варианте – плоского рисунка, но есть пазлы и с различными объёмными формами. По своей сути пазл представляет своего рода мозаичное полотно, то есть цельный рисунок состоящий из фрагментов, но не является мозайкой в полном его смысле. В начале игры показывается полная картина, которая позже разбивается на фрагменты. Цель игрока собрать фрагменты воедино как можно быстрее.

# 1 ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Овладеть практическими навыками проектирования и разработки законченного, отлаженного и протестированного программного продукта с использованием языка высокого уровня С++ ,овладеть практическими навыками проектирования и разработки законченного, отлаженного и протестированного программного продукта с использованием языка высокого уровня С++. Разработать игру “Пазлы” с использование среды разработки SFML. Вот несколько общих целей, которые могут быть установлены для освоения ООП:

Понимание Основных Принципов ООП:

Цель: Освоить базовые концепции ООП, такие как инкапсуляция, наследование и полиморфизм для обеспечения более глубокого понимания организации кода и его взаимодействия.

Навыки Проектирования Классов и Объектов:

Цель: уметь создавать классы и объекты, определять их атрибуты и методы, так как это является основой ООП и позволяет структурировать код для более легкого понимания и поддержки.

Применение Инкапсуляции:

Цель: использовать инкапсуляцию для скрытия внутренних деталей реализации классов и предоставления публичного интерфейса. Почему это важно: это способствует безопасности кода и облегчает его сопровождение.

Мастерство В Обработке Наследования:

Цель: понимать, как использовать наследование для создания иерархии классов и расширения функциональности для эффективного использования кода.

Овладение Исключениями и Обработкой Ошибок:

Цель: Знание, как обрабатывать исключения и ошибки в объектно-ориентированных программах, так как это повышает устойчивость программы к ошибкам и улучшает ее отказоустойчивость.

Эти цели могут служить отправной точкой для разработки программистом плана обучения и практического применения концепций ООП в реальных проектах.

Преимущество SFML перед Qt:

Ориентированность на мультимедиа: SFML предоставляет более прямой и легковесный подход для обработки мультимедийных данных, таких как графика, звук и сетевое взаимодействие, а также к ресурсам и операциям, когда требуется тонкое управление процессами.

Простота: Низкий порог вхождения для разработчиков полезен для создания независимых проектов, из-за чего SFML популярен в игровой разработке и других графических приложениях.

SFML — это библиотека для представления мультимедийных данных. SFML обеспечивает простой интерфейс для разработки игр и прочих мультимедийных приложений. Состоит она из пяти модулей: system, window, graphics, audio и network. Ниже перечислены некоторые преимущества SFML:

Портативность: SFML является кроссплатформенной библиотекой, а это значит, что приложения будут работать на большинстве операционных систем (Windows, Linux, Mac OS X, Android и iOS). Это упрощает разработку и позволяет достичь большей аудитории.

Поддержка различных языков программирования: SFML поддерживает большое число языков программирования (официально Си и .Net подобные языки, а благодаря различным сообществам ещё и такие как Java, Ruby, Python, Go, и другие), что повышает популярность библиотеки, а также способствует её развитию.

Простота: SFML предоставляет простой и интуитивно понятный API, что делает её отличным выбором для начинающих разработчиков, а также позволяет более опытным программистам быстро и эффективно создавать мультимедийные приложения. Библиотека имеет обширную документацию, которая поможет разработчикам понять основные концепции и функции SFML и использовать их в своих проектах.

Инструменты: SFML предоставляет разработчикам мощные инструменты для создания 2D и 3D графики, включая возможность работы с текстурами, спрайтами, шейдерами, окнами, камерами и многое другое. Библиотека также поддерживает воспроизведение аудиофайлов различных форматов и позволяет осуществлять сетевое взаимодействие для создания многопользовательских игр или приложение.

Огромное сообщество: SFML имеет обширное сообщество разработчиков, которые обсуждают новые возможности библиотеки SFML и всегда готовы помочь с решением проблемы. Это позволяет сообществу быстро находить решения для своих проблем и получать поддержку от других разработчиков.

**2. ОБЗОР ЛИТЕРАТУРЫ**

Перед началом разработки нужно было ознакомится с принципом игры “Пазлы”. При ознакомлении с принципами был использован сайт [11].

Пазл (или мозаика) – это игра-головоломка, в которой необходимо собрать одну картинку из множества фрагментов различной формы. Слово "пазл" переводится с английского как "загадка" или "головоломка". Вариаций у пазлов множество. К ним относятся классические пазлы из картона и фетра, объемные пазлы из дерева, мягкие и 3D-пазлы, а также рамки-вкладыши и разрезные картинки.

Целью игры в пазл является получение единой картины из отдельных ее элементов. Пазл – это интересный и оригинальный подарок как для ребёнка, так и для взрослого. Игра с пазлами развивает аккуратность, образное и логическое мышление, способность принимать сложные решения, внимание и фантазию, терпение и усидчивость, а также укрепляет память и стимулирует творческий потенциал. Пазлы также учат воспринимать связи между целой картинкой и ее фрагментами, помогают развить мелкую моторику, координацию движений и сенсорику. Управляясь с маленькими фрагментами, движения рук у детей становятся более осмысленными и четкими. В будущем это позволяет избежать проблем с речью и почерком.

Помимо классических двухмерных (плоских) пазлов, в наши дни можно найти и трёхмерные объёмные (3D-пазлы). Суть - составить объект из множества фрагментов рисунка в трёхмерном пространстве, в итоге получается своего рода конструктор, только более сложный. Обычно 3D-пазлы представляют собой различные архитектурные здания, фрукты, машинки, корабли и прочее. Такие поделки можно делать не только из пластиковых деталей. Интересны объёмные головоломки, в которых из картонных фрагментов нужно сделать фигурную конструкцию в виде шара или вазы.

Так же при разработке была использована официальная документация SFML documentation. SFML — это библиотека для разработки графических приложений на C++. Она предоставляет широкий функционал для создания графических интерфейсов, обработки событий, работы с сетью, базами данных и другими задачами. Вот краткое описание документации SFML:

Официальная документация: Сайт SFML: Официальный сайт SFML содержит обширную документацию, включая руководства, API-справочники, примеры кода и другие материалы.

Руководства и Введение в SFML:

Getting Started: Раздел "Getting Started" в документации обычно предоставляет информацию о установке SFML, настройки среды разработки и создании простого приложение. Overview: Введение в SFML, её основные концепции и принципы.

Создание графического интерфейса: SFML Graphics: Информация о базовых элементах управления, таких как кнопки, поля ввода и другие.

Работа с сетью и базами данных: SFML Network: Инструменты для работы с сетью, включая HTTP-запросы, сокеты и другие, а также информацию о работе с базами данных, включая поддержку различных СУБД.

Многозадачность и Параллелизм: SFML Concurrency: Раздел, посвященный поддержке многозадачности и параллелизма в SFML.

Межплатформенная разработка: Platform Notes: Рекомендации и особенности для кроссплатформенной разработки на разных операционных системах.

Примеры кода и Учебные проекты: SFML Examples: Обширный набор примеров кода для различных компонентов SFML. SFML Tutorials: Учебные проекты и туториалы, позволяющие освоить функционал библиотеки.

Форумы и Сообщества: SFML Forum: Онлайн-форумы, где разработчики могут задавать вопросы, делиться опытом и получать поддержку от сообщества SFML.

Обновления и Дополнительные ресурсы: Блог SFML: Официальный блог с новостями, статьями и обновлениями от команды разработчиков. Дополнительные ресурсы: Дополнительные материалы, такие как видеоуроки, вебинары и другие образовательные ресурсы. SFML предоставляет обширные функционал для создания высококачественных приложений, и его документация является важным ресурсом для разработчиков, стремящихся освоить эту библиотеку.

## 2.1 обзор методов и алгоритмов решения поставленной задачи

Существует множество аналогов игры “Пазлы”. В общем случае они подразделяются на плоские(2D-пазлы) и объёмные(3D-пазлы).

Целью игры в плоский пазл является получение единой картины из отдельных ее элементов. Игра с пазлами развивает аккуратность, образное и логическое мышление, способность принимать сложные решения, внимание и фантазию, терпение и усидчивость, а также укрепляет память и стимулирует творческий потенциал. Пазлы также учат воспринимать связи между целой картинкой и ее фрагментами, помогают развить мелкую моторику, координацию движений и сенсорику. Управляясь с маленькими фрагментами, движения рук у детей становятся более осмысленными и четкими.

В отличие от плоских пазлов, в объёмных нужно не только разрезать картинку на разные кусочки, но и необходимо спроектировать будущий макет, точно рассчитав все размеры и пазы соединения. Не стоит бояться трудностей в сборке пазла, обычно каждый набор содержит подробную иллюстрированную инструкцию и детальное фото готового объекта. Для сборки не требуются дополнительные материалы и инструменты. Детали соединяются особой системой крепления без использования клея. Такая конструкция позволяет многократно ее использовать.

Объемные пазлы намного сложнее собирать, чем обычные. От этого интерес к ним повышается. В процессе сборки развивается логическое и пространственное мышление, внимание, усидчивость, мелкая моторика. В дополнение к этому, например, в процессе сборки архитектурных пазлов ребенок невольно познакомится с историей создания того или иного здания, расширит свой кругозор. Взрослый человек отметит сложность и солидность постройки, увлечение подобным хобби идеально подходит для коллекционирования, такие архитектурные сооружения станут гордостью любой коллекции.

В процессе разработки программы были использованы различные возможности языка С++, которые будут описаны ниже. Для решения задачи был выбран язык программирования С++ и методология объектно-ориентированного программирования[4].

Для хранения объектов было использовано массивы. Также реализован класс Puzzle хранящий фрагменты и картину для использования в интерфейсе программы.

Для удобства работы с классами HardPuzzle, MediumPuzzle и EazyPuzzle было использовано наследования, которое позволило использовать методы и поля классов, используемые в трёх классах. Так же были использованы такой принцип как инкапсуляция. На практике это означает, что класс должен состоять из двух частей: интерфейса и реализации[6].

В программе осуществляется обработка исключительных ситуаций – механизм, предназначенный для описания реакции программы на ошибки, которые могут возникнуть при выполнении программы и приводят к невозможности (бессмысленности) дальнейшей обработки программой её базового алгоритма [7].

# 

# 3 ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВНИЕ

В данном разделе описываются входные и выходные данные программы, диаграмма классов, а также приводится описание используемых классов и их методов.

##### 3.1 Входные данные

Для работы программы и отображение фрагментов картины были сделанные специальные png картинки, помещенные в папку paint.

**3.2 Разработка диаграммы классов**

Диаграмма классов данной работы показана в приложении А.

##### 3.3 Описание классов

###### 3.3.1 Класс setSound

setSound – класс в котором реализуется озвучка действий игрока.

Описание полей класса:

SoundBuffer w, t, p, ta, s – необходимы для хранения звука под определённую ситуацию.

Sound win, time, put, take, start – вызов звука под определённую ситуацию.

bool stopwin – нужен для остановки воспроизведения звука.

Описание методов:

setSound()- загрузка файлов со звуками и запись в буфер для последующего вызова.

###### 3.3.2 Класс RandTabels

RandTabels - рандомизация положения фрагментов пазла.

Описание полей класса:

int k = 0 – для определения координады положения фрагмента.

int a[12] – массив для хранения фрагментов пазла.

int Px[12], Py[12] – координаты x и y для определения нахождения каждого фрагмента пазла.

Описание методов:

RandTabels()– рассчитывает координаты нахождения каждого фрагмента пазла в рандомном порядке.

###### 3.3.3 Класс EazyPuzzle

###### 

EazyPuzzle– класс реализующий пазл лёгкого уровня(12 фрагментов).

Описание полей класса:

Texture pu[12] - устанавливаем текстурку фрагмента.

Sprite sprite[12] – форма фрагмента.

Описание методов:

EazyPuzzle() – хранит фрагменты, и координаты, в которых должны стоять фрагменты для получения полноценной картины.

**3.3.4 Класс Stopwatch**

Stopwatch – класс реализующий отановку и счёт времени.

Описание полей класса:

Sprite min[2], sec[2], toc – хранение текстурки чисел минут и секунд, а также их положение на экране.

int m[2], s[2], ms – значение времени.

Описание методов:

Stopwatch(Texture& image) – устанавливает время на экране.

void update() - для корректного значения времени.

# 4 РАЗРАБОТКА ПРОГРАММНЫХ МОДУЛЕЙ

## 4.1 Разработка схем алгоритмов

Метод RandTabels()рандомно устанавливает фрагменты пазла на экране. Схема метода RandTabels() показана в приложении Б.

Метод animation() анимирует разбор картинки на фрагменты в начале игры. Схема метода animation() показана в приложении В.

## 4.2 Разработка алгоритмов

Метод RandTabels() класса RandTabels.

Первый цикл for (цикл по переменной i):

Шаг 1. Инициализируем a[i],Px[i],Py[i] с 0, проходимся по этому циклу n раз;

Второй цикл for (цикл по переменной i):

Шаг 2. Заполняем массив a значениями от 1 до 12 в случайном порядке. Используем временную переменную tmp для генерации случайных индексов до тез пор, пока a[temp] не станет равным 0;

Шаг 3. Отображаем значения из массива a в координаты массивов Px и Py, используя формулы для преобразования линейного индекса k в 2D-координаты (x,y) на основе определённых расчётов. Полученные координаты сохраняются в массивах Px и Py;

Шаг 4. Сбрасываем переменную k в 0;

Метод animation() класса EazyPuzzle.

Первый цикл for (цикл по переменной i):

Шаг 1. Создаём массив векторов ps и заполняем его позициями из массива Rparts;

Шаг 2. Проверяем, если флаг got равен true, выполняем следующий блок кода;

Шаг 3. Уменьшаем переменную t и, если она достигает 0, сбрасывает флаг got, устанавливает t в 500 и воспроизводит звук;

Второй цикл for (цикл по переменной i):

Шаг 4. Если флаг got равен false, уменьшает переменную t. Если t>0, обновляет позиции в массивах Px и Py в соответствии с позициями из ps. Иначе, устанавливает флаг play в true и воспроизводит звук;

Третий цикл for (цикл по переменной i):

Шаг 5. Обновляет позиции спрайтов в массиве sprite на основе координат из массивов Px и Py в объекте puzzles;

# 5 РЕЗУЛЬТАТ РАБОТЫ

На рисунке 5.1 изображена начало работы программы. При старте программы показывается начальная картинка. Позже картинка разделяется на фрагменты и запускается таймер, после чего пользователь может начинать собирать пазл.

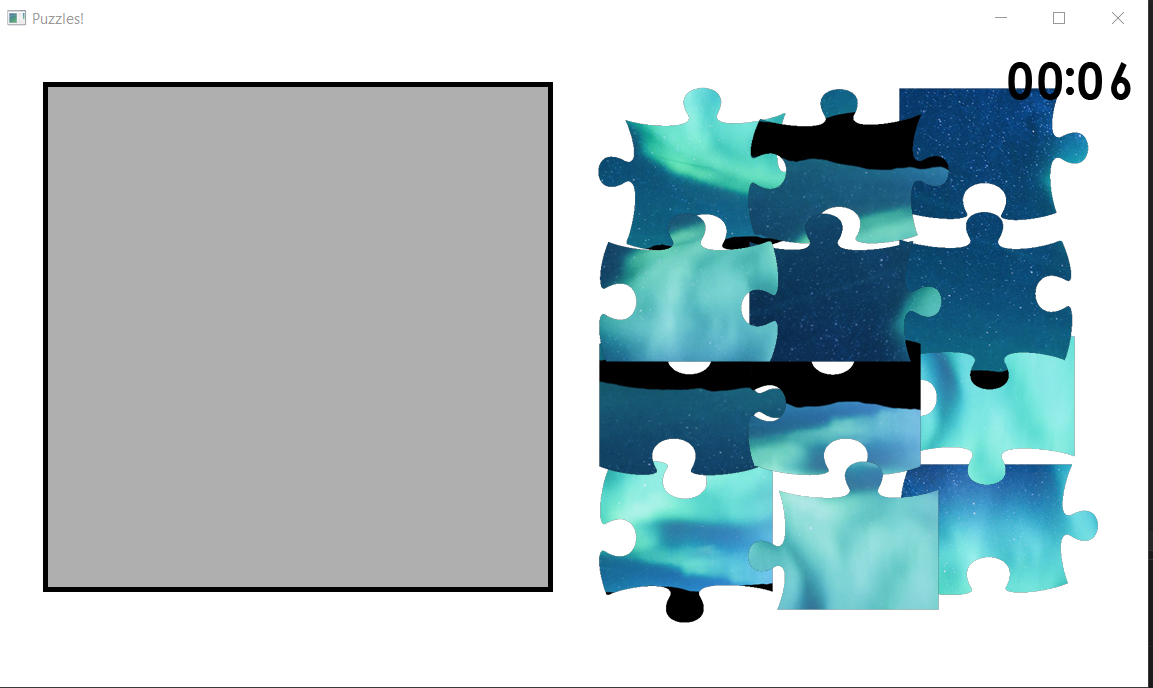


Рисунок 5.1 – Начало работы программы

На рисунке 5.2 показа работа кнопки остановки таймера. При нажатии кнопки программа отслеживает сигнал нажатия и всплывает текст на главный экран. При нажатии паузы игрок не может передвигать фрагменты. После повторного нажатия клавиши, пауза будет отменена и таймер продолжит отсчёт с того времени когда остановился, а игроку будет снова дана возможность передвигать фрагменты.

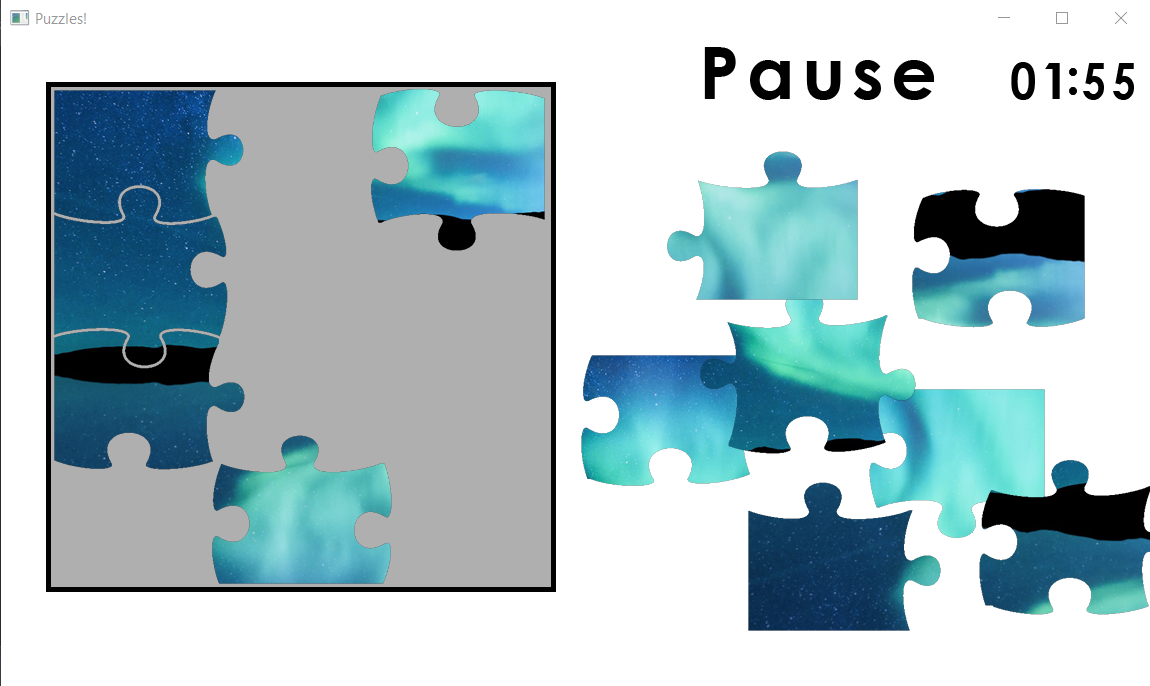


Рисунок 5.2 – Прожатие паузы

На рисунке 5.3 отражена работа подсказки. Благодаря подсказкам пользователь может понять какой фрагмент пазла подходит для данной позиции, а какой нет. На этом экране отображается контур подходящего для этого места фрагмента картины.

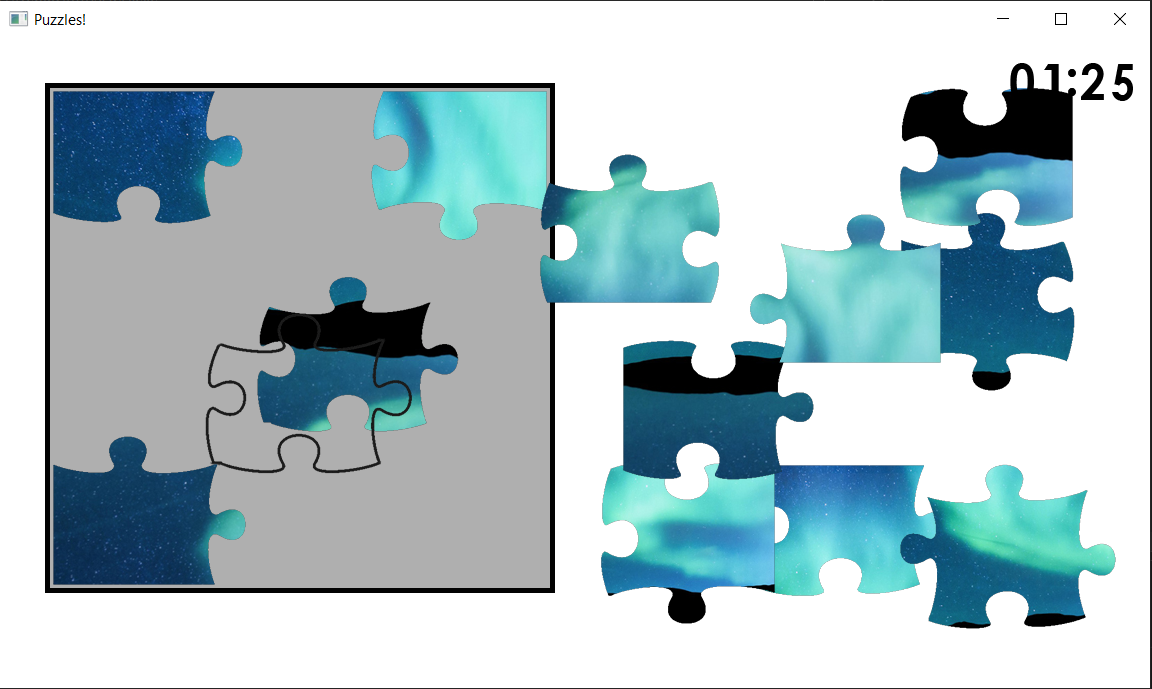


Рисунок 5.3 – Подсказки

На рисунке 5.4 показан экран главного меню. С помощью него можно начать игру с помощью кнопки “Start”, изменить цвет доски и сложность с помощью кнопки “Options” и выйти из программы с помощью кнопки “Exit”.

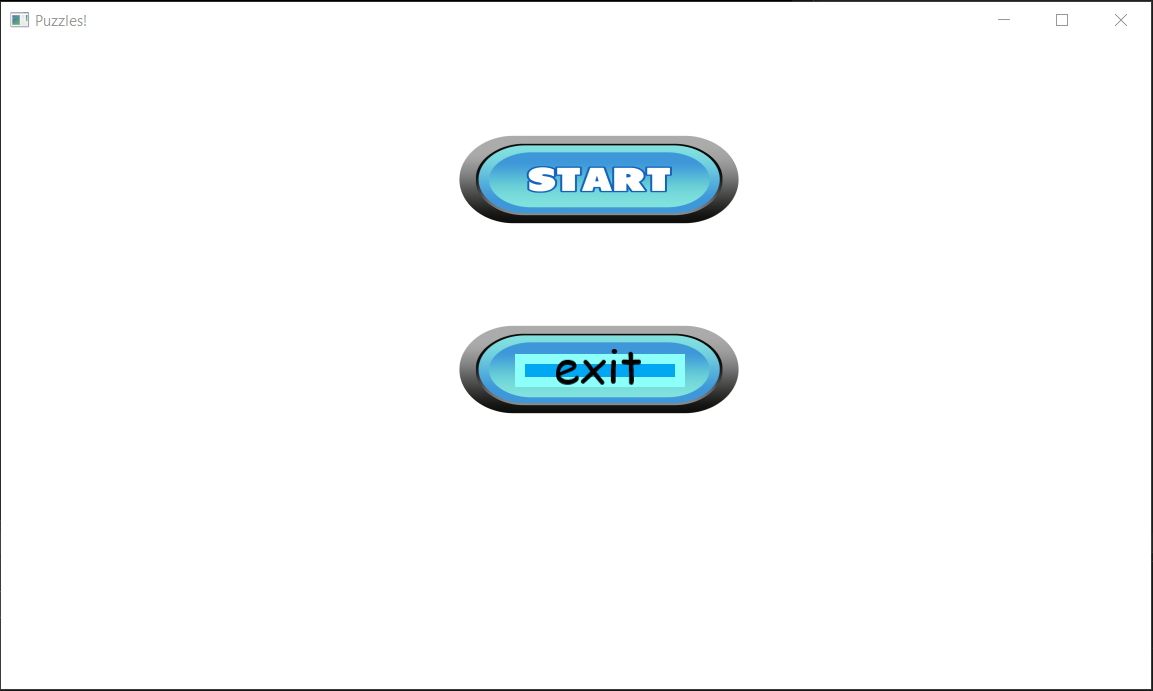


Рисунок 5.4 – Главное меню

На следующем рисунке можно увидеть работу меню и кнопки “Options” с помощью которой мы изменили цвет доски с серого на розовый. Также можно поменять сложность пазла, которая зависит от количества фрагментов.

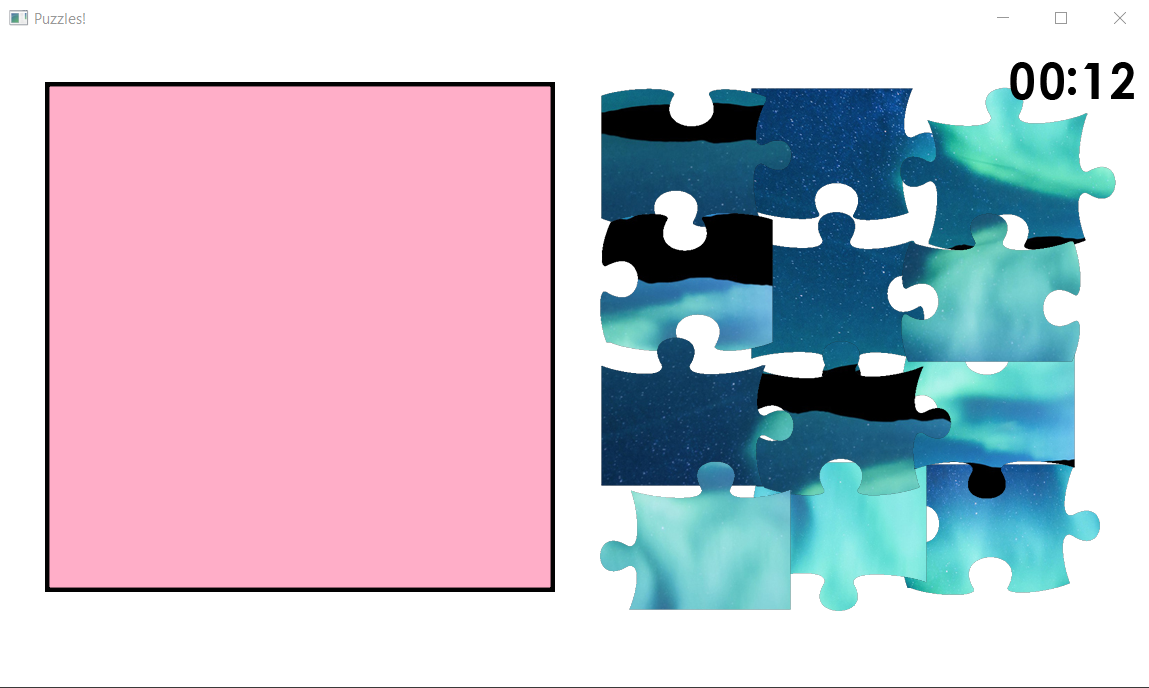


Рисунок 5.5 – Изменение цвета доски

Рисунок 5.6 показывает конец программы. При правильном заполнении пользователем доски фрагментами, выскакивает текст “You win” на экран, при этом таймер останавливается и пользователь разобрать картину обратно не может.

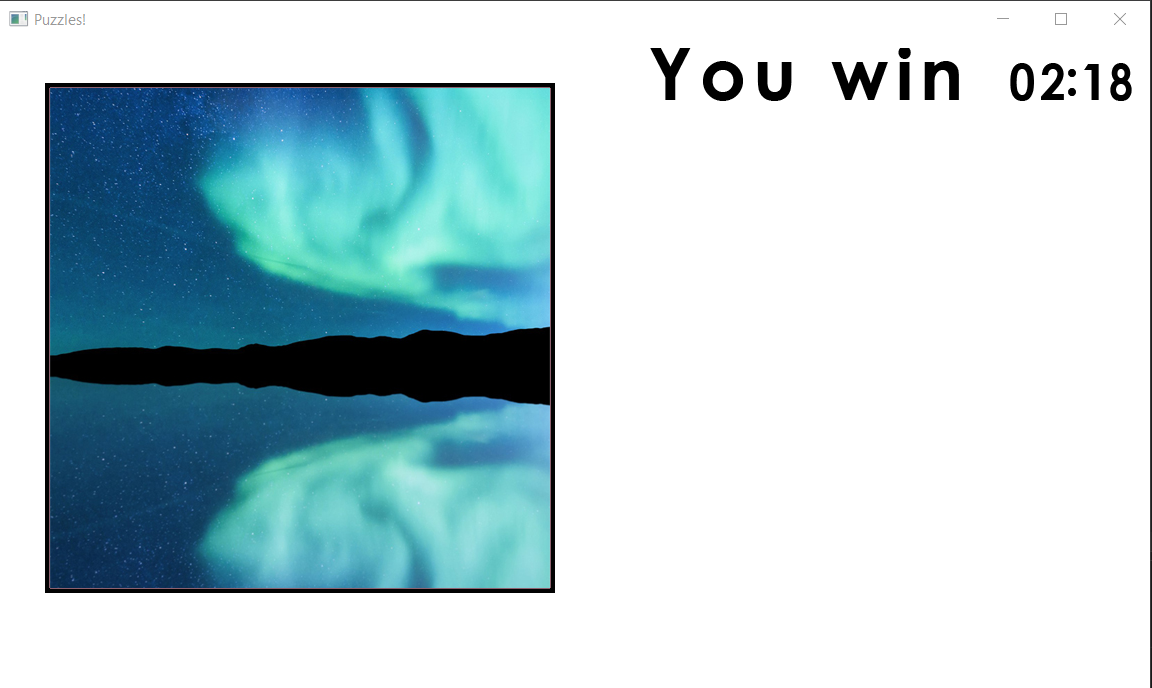


Рисунок 5.6 – Конец программы

# ЗАКЛЮЧЕНИЕ

В процессе разработки игры "Пазлы" были успешно достигнуты первоначально поставленные цели. Реализована полноценная игра с удобным пользовательским интерфейсом, позволяющим игрокам участвовать в игре и принимать решения. Также в проекте были реализованы следующие ключевые элементы:

Логика игры: В игре "Пазлы" правильно реализована игровая механика, включая разборка картины, ход таймера и определение победы.

Взаимодействие с пользователем: Игра предоставляет удобный интерфейс для игроков, позволяющий им кастомизировать, выбирать сложность и многое другое.

Опции игры: Разработаны дополнительные функции, такие как возможность ставить паузу, подсказки, сложность и т. д., что делает игру более интересной и разнообразной.

Графика: для улучшения визуального восприятия игры были созданы графические элементы, включая изображения кнопок, фон доски.

Алгоритмы: Реализованы алгоритмы определения выигрыша, паузи и так далее.

В результате разработки игры “Пазлы” я понял, что данная игра сохраняет свою актуальность в современном мире развлечений и разработки игрового контента. “Пазлы” представляет собой не только классическую головоломку, но и важный элемент интеллектуальной культуры, который может быть успешно адаптирован и в современных виртуальных пространствах. Вот несколько аспектов, делающих игру “Пазлы” актуальной:

Универсальность:

Пазлы подходят для всех возрастов. Они предоставляют вызов как детям, так и взрослым, причём для каждого возраста существует своя сложность.

Развивающий аспект:

Сборка пазлов способствует развитию когнитивных навыков, таких как память, внимание к деталям, концентрация внимания, логическое мышление, улучшение моторики рук и многое другое.

Отдых:

Пазлы — это еще и отличный способ снять стресс. Они помогают расслабиться психологически, отвлечься от повседневных проблем. Решение головоломок и поиск решений погружает мозг в медитативное состояние.

Вариативность:

Огромное разнообразие изображений. Каждый найдёт пазлы с изображением на свой вкус.

Социальность:

Пазлы помогают наладить сотрудничество между коллегами. Исследователи Йельского университета обнаружили, что, когда работники могут вместе решать задачи на рабочем месте, это помогает им улучшить их отношения и способность к сотрудничеству и командной работе.

Игра "Пазлы" — это хороший пример того, как разработка игрового приложения может достичь успешных результатов, предоставляя пользователю увлекательный и интерактивный опыт. Приложение было тестировано на операционной системе Windows 10 и разработано в Microsoft Visual Studio 2022 и SFML.

В процессе написания данной курсовой работы я глубоко погрузился в мир объектно-ориентированного программирования (ООП) и рассмотрел разнообразные его аспекты. Изучение ООП выдало для меня не только новые знания, но и значимый опыт, который открыл широкие перспективы в области разработки программного обеспечения.

Полученные знания об ООП стали ценным ресурсом для моего дальнейшей карьеры в сфере IT. Я уверен, что применение концепций ООП в реальных проектах будет способствовать созданию более качественного и поддерживаемого программного обеспечения. Изучение объектно-ориентированного программирования стало не только частью моего обучения, но и важным этапом в развитии моих навыков и понимания важности правильной организации кода для достижения поставленных целей в мире современной разработки ПО.

# 

# СПИСОК ЛИТЕРАТУРЫ

1. "Объектно-ориентированное программирование на С++" Бьярн Страуструп
2. "Язык программирования С++" Герберт Шилдт
3. "C++ Primer" Липман, Лажойе, Му, Хопкинс
4. "Алгоритмы. Построение и анализ" Кормен, Лейзерсон, Ривест, Штайн
5. "Введение в алгоритмы" Кормен, Лейзерсон, Ривест, Штайн
6. "Алгоритмы на C++" Роберт Седжвик, Кевин Уэйн
7. C++: эффективное программирование. 55 способов улучшения структуры программ и стиля кода" Scott Meyers
8. "Алгоритмы. Построение и анализ" Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein
9. "Структуры данных и алгоритмы в C++" Robert Lafore
10. "C++ for Game Programmers" Noel Llopis
11. Описание правил игры “Пазлы”[Электронный ресурс].-Электронные данные.-Режим доступа: <https://grandgames.net/puzzle/pravila/->

[Дата](https://www.gipsyteam.ru/poker/pravila-blekdzheka%20-%20Дата) доступа:30.11.2023

# ПРИЛОЖЕНИЕ А

*(обязательное)*

Диаграмма классов

# ПРИЛОЖЕНИЕ Б

##### *(обязательное)*

Схема метода RandTabels()

# ПРИЛОЖЕНИЕ В

##### *(обязательное)*

Схема метода animation()

# 

# ПРИЛОЖЕНИЕ Г

*(обязательное)*

Код программы

# ПРИЛОЖЕНИЕ Д

*(обязательное)*

Ведомость документов