

Введение

На курсовое проектирование была поставлена задача разработать игровое приложение «Кто из нас гроссмейстер?».

Цель курсового проекта заключается в разработке программного продукта, который позволит пользователю приятно провести время, познакомиться с основами шахматной партии поспособствует улучшению мышления, навыки запоминания, принятие решений.

Создаваемое приложение будет ориентировано на пользователей любого возраста, интересующихся шахматами и желающих развивать свои логические и аналитические навыки.

Далее приведем краткое описание разделов пояснительной записки.

Первый раздел носит название “Анализ задачи”. В нем вы сможете ознакомиться с постановкой задачи, которая включает в себя: исследование предметной области поставленной задачи, определение ее организационно-экономической сущности. Также в этом разделе вы сможете узнать о том, как данная задача решается в настоящее время. Все входные и выходные данные тоже будут описаны в первом разделе. В подразделе “Инструменты разработки” будет рассмотрена среда, в которой создается данный курсовой проект. Здесь также будут установлены минимальные и оптимальные требования к аппаратным характеристикам, обеспечивающим правильное функционирование поставленной задачей.

В разделе “Проектирование задачи” будут рассмотрены основные аспекты разработки программного продукта. Здесь можно будет узнать об организации данных в контексте среды разработки. В данном разделе будет четко описан пользовательский интерфейс, составлены алгоритмы процесса обработки информации, описана разработка системы справочной информации.

“Реализация задачи” – это третий раздел пояснительной записки, в котором описываются все элементы и объекты, которые будут использованы при реализации данного приложения. В этом разделе будут четко описаны функции пользователя и их структура. Здесь можно будет найти таблицу, в которой будет представлена полная аннотация файлов, используемых в данном проекте.

Четвертый раздел – “Тестирование”. В нем будет описано полное и функциональное тестирование данной программы, т.е. будет оттестирован каждый пункт меню, каждая операция, которая выполняется приложением. Будут смоделированы все возможные действия пользователя при работе с программой, начиная от запуска до выхода.

					КП 2-40 01 01.35.40.13.24	Лист
Изм.	Лист	№докум.	Подпись	Дата		4

В разделе “Применение” будет описано назначение, область применения, среда функционирования курсовой программы. Также в нем будет описано использование справочной системы.

“Заключение” будет содержать краткую формулировку задачи, результаты проделанной работы, описание использованных методов и средств, описание степени автоматизации процессов на различных этапах разработки.

В “Литературе” будет приведен список используемых при разработке источников.

В приложениях к пояснительной записке будет приведен листинг программы с необходимыми комментариями.

Схема работы системы будет представлена в графической части.

					КП 2-40 01 01.35.40.13.24	Лист
Изм.	Лист	№докум.	Подпись	Дата		5

1 Анализ задачи

1.1 Постановка задачи

Тема курсового проекта «Разработка игрового приложения «Кто из нас Гроссмейстер» в жанре детектив.

Цель разработки: создание игрового приложения, которое позволит пользователю приятно провести время, познакомиться с основами шахматной партии поспособствует улучшению мышления, навыки запоминания, принятие решений.

Игровое приложение ориентировано на пользователей любого возраста, интересующихся шахматами и желающих развивать свои логические и аналитические навыки.

Программный продукт — это игра, в которой два игрока соревнуются друг с другом, используя различные стратегии и тактики для того, чтобы поставить мат королю противника. Уникальность шахмат заключается в их глубокой стратегической и тактической составляющей. Каждая фигура на шахматной доске обладает своими особенными способами передвижения и возможностями влиять на ход партии. Это требует от игроков постоянного просчета вариантов, понимания слабых и сильных сторон позиции, а также выработки долгосрочных планов.

Аналоги данного программного продукта:

- stockfish, komodo, houdini - мощные шахматные движки, способные играть на уровне гроссмейстеров;
- chess.com, lichess.org - онлайн-платформы для игры в шахматы, проведения турниров и обучения;
- fritz, chessbase - программы для анализа и обработки шахматных партий.

Периодичность использования зависит от нужд потребителя, может использоваться ежедневно.

Источники и способы получения данных: Контент-анализ документов (а именно: шахматные учебники, книги по истории шахмат и биографии известных шахматистов, а так же шахматные журналы, газеты и интернет-издания).

Входной информацией является ход сделанный пользователем.

Постоянной информацией является - информация о правилах игры, содержащаяся справке. Так же картинки и медиа-файлы.

Разрабатываемый программный продукт позволит выполнить следующие действия:

- возможность просмотра справки;
- включение мелодии в игре;
- возможность самостоятельно влиять на игровой процесс;

					КП 2-40 01 01.35.40.13.24	Лист
Изм.	Лист	№докум.	Подпись	Дата		6

- анализировать позицию на доске, оценивая силы и слабости фигур;
- искать объекты, над которыми производятся различные действия, такие как атака или защита
- перемещение между сценами.

1.2 Инструменты разработки

Для разработки данного проекта выбрана среда Delphi (Delphi 11), так как это среда объектно-ориентированного программирования, относящаяся к классу RAD — (Rapid Application Development — «Средство быстрой разработки приложений»), реализованная на Object Pascal. Используется для разработки визуализированного представления программного обеспечения.

Также для разработки программы необходимы:

- Help & Manual — система создания и редактирования документов и справок Windows, как для одиночного, так и для многопользовательского редактирования;
- Google-браузер – нужен для нахождения информации;
- Microsoft Word 2021 — редактор текста для написания документации;
- Power Point 2021 — нужен для создания отчётной презентации;
- Adobe Photoshop CC 2019 — нужен для создания и редактирования изображений;
- Smart Install Maker — нужен для создания инсталлятора;
- Блокнот – для создания текстовых файлов.

При разработке данного программного продукта был использован компьютер со следующими характеристиками:

- процессор: Intel Core i5-12500H;
- видеокарта Nvidia GeForce RTX 3060 6Gb;
- ОЗУ: 16 Gb;
- память: SSD 512Gb;
- ОС — Windows 11.

1.3 Требования к приложению

На этапе исследования предметной области был установленный целый ряд требований, предъявляемые к разрабатываемой задаче.

При моделировании форм следует учесть такие моменты:

- интуитивно понятный интерфейс и управление;
- небольшое количество информации на игровом экране;

					КП 2-40 01 01.35.40.13.24	Лист
Изм.	Лист	№докум.	Подпись	Дата		7

- небольшие окна формы;
- визуально понятный и приятный глазу интерфейс;
- наличие информации о функционале игры;
- небольшие кнопки.

Минимальные системные требования:

- 1.8 GHz процессор;
- 1 GB RAM;
- от 616 MB доступного места на жестком диске.

Рекомендуемые системные требования:

- процессор Intel Core i5-12500H (или более производительный);
- 2 GB RAM;
- от 616 MB доступного места на жестком диске.

Требования к интерфейсу: при разработке интерфейса приложения использованы преимущественно синие и светлые оттенки. Основные разделы приложения доступны с первой страницы. Разработан понятный пользователю интерфейс, фрагменты текста располагаются на экране так, чтобы пользователь не испытывал какого-либо дискомфорта в плане восприятия игровой информации, отображённой на экране.

Ошибки программы вследствие некорректного взаимодействия пользователя при работе с программой через графический интерфейс не должны влиять на работоспособность.

Надёжность программы должна быть на высоком уровне, не допуская ситуаций программных ошибок и некорректной работы приложения.

					КП 2-40 01 01.35.40.13.24	Лист
Изм.	Лист	№докум.	Подпись	Дата		8

2 Проектирование задачи

2.1 Организация данных

Основным средством хранения информации в приложении являются: png-изображения для хранения фоновых изображений, bmp-изображения для хранения текстур для кнопок. Звуковое оформление игрового процесса осуществляется при помощи файлов формата mp3.

Система справочной информации представлена файлом справки, который содержит информацию о приложении и правилах его использования, а также информацию об авторе.

Основными функциями приложения являются:

- включение мелодии, сопровождающей игровой процесс;
- вхождение в игру;
- осуществление игры.

2.2 Процессы

Согласно всем перечисленным требованиям и указаниям, которые были рассмотрены в разделе «Анализ задачи», было определено, чем конкретно должна заниматься разрабатываемая программа. Главной её задачей является победа 1 из игроков.

Для реализации задач будут использоваться процедуры. С помощью процедур, например, будет осуществляться загрузка шахматного поля. Так же с помощью процедур будут осуществляться: проверка хода пользователя из клетки в клетку (реализовано с помощью компонента TImage); переход к следующей форме при нажатии на различные объекты, изображённые на картинке (реализовано с помощью события ImageClick); вывод побитых фигур в компонент TLabel.

С помощью компонента TMediaPlayer, будет воспроизводиться медиа файл (подгружается из файла mp3).

2.3 Описание внешнего пользовательского интерфейса

Важным при выполнении курсового проекта является организация диалога между пользователем и самой программой. Во многом это зависит от того, как программист разрабатывает данную программу, какие компоненты использованы и какие методы автоматизированы.

Особое внимание уделяется интерфейсу. Внешний вид программы организован так, чтобы пользователь понял, что от него требуется. Таким образом,

					КП 2-40 01 01.35.40.13.24	Лист
Изм.	Лист	№докум.	Подпись	Дата		9

ключевые кнопки графически выделены для визуализации интерактивных возможностей приложения как показано на рисунке 1.

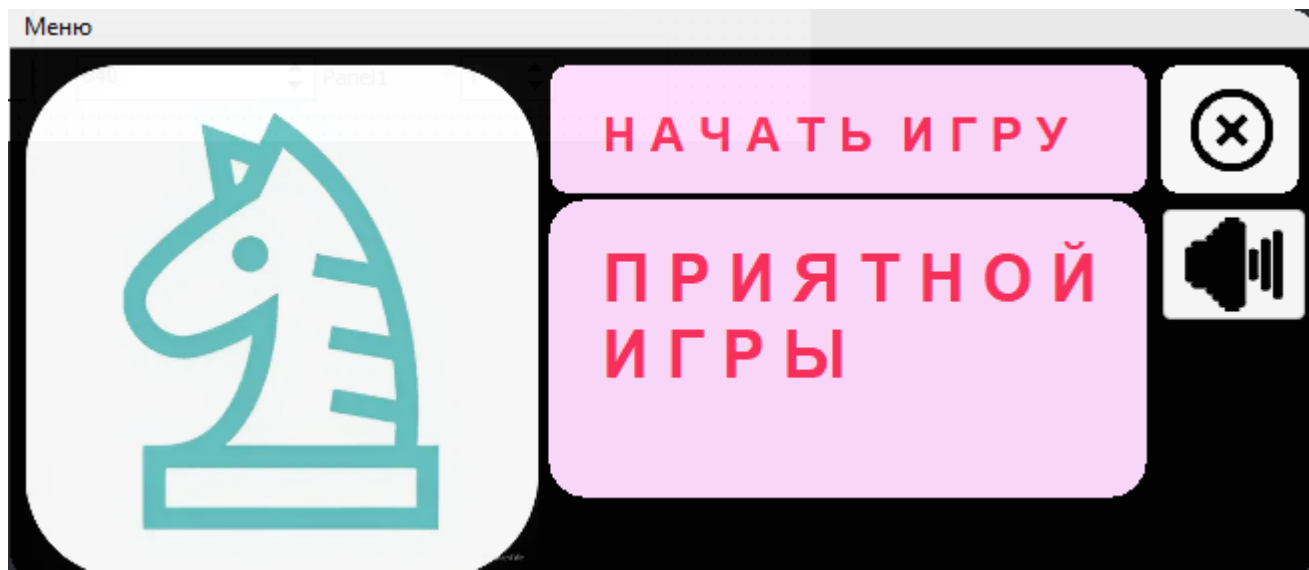


Рисунок 1 — Визуализация интерактивных возможностей приложения

Для организации эффективной работы пользователя создаётся целостное приложение данной предметной области, в котором все компоненты приложения сгруппированы по функциональному назначению, со всех второстепенных форм можно вернуться на главную, как показано на рисунке 2. При этом же главной целью является удобный и приятный графический интерфейс пользователя.

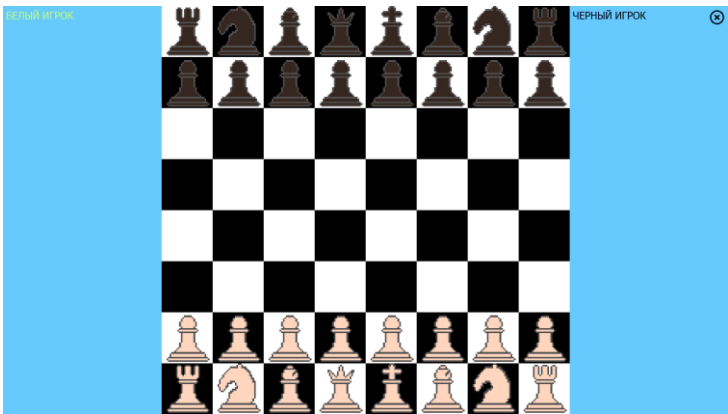


Рисунок 2 — Графический интерфейс игры

Таким образом, для успешной работы всего проекта реализован интуитивно понятный интерфейс и управление с приятной гаммой цветов и шрифтами.

Структура навигации по проекту представлена на рисунке 3.

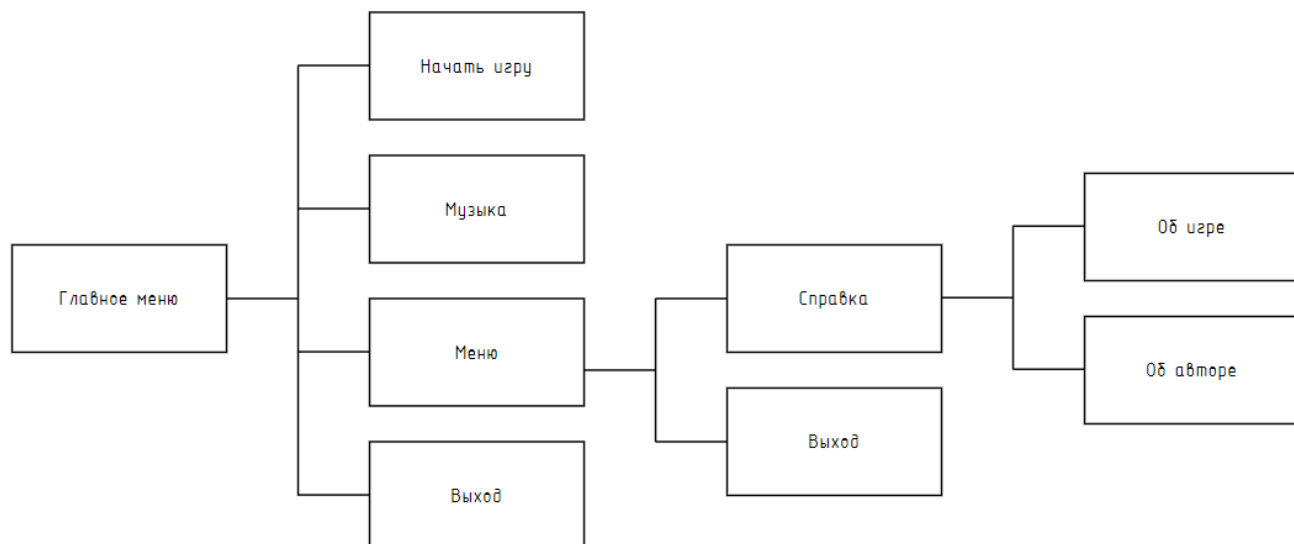


Рисунок 3 — Структура навигации по проекту

3 Реализация

3.1 Структура программы

Данный курсовой проект содержит 5 модулей, рассмотрим их назначение:

1 модуль LoadU — отвечает за загрузочный экран;

2 модуль Launcher — главное меню приложения;

3 модуль PlayerU — игровое поле;

4 модуль EngineUI — модуль;

5 модуль EngineClasses — модуль.

Таблица 1 – Процедуры и функции

Имя процедуры (функции)	В каком модуле находится	За каким компонентом закреплена	Назначение
1	2	3	4
Procedure TLoading_Screen.Timer1Timer (Sender: TObject);	LoadU.pas	Timer1	Заполнение ProgressBar1
procedure TLoading_Screen.FormCreate(Sender: TObject);	LoadU.pas	LoadU	Загрузка изображения
procedure LaunchGame;	Launcher.pas	TfrmLauncher	Запускает игру с заданными настройками
procedure FormCreate(Sender: TObject);	Launcher.pas	TfrmLauncher	Инициализация формы и настройка пользовательского интерфейса
procedure sedWidthChange(Sender: TObject);	Launcher.pas	TfrmLauncher	Поддерживает соотношение сторон игрового окна
procedure tmrTimer(Sender: TObject);	Launcher.pas	TfrmLauncher	Обработка событий наведения и щелчка мыши для воспроизведения, настроек отображения и кнопок закрытия
procedure N2Click(Sender: TObject);	Launcher.pas	N2	Открывает файл справки (Manual.chm)
procedure N3Click(Sender: TObject);	Launcher.pas	N3	Прекращение работы приложения
procedure FormShow(Sender: TObject);	Launcher.pas	TfrmLauncher	Воспроизведение музыки при отображении формы
procedure VolumeButtonClick(Sender: TObject);	Launcher.pas	Volume Button	Переключает воспроизведение музыки.

Продолжение таблицы 1

1	2	3	4
procedure FormDestroy(Sender: TObject);	PlayerU.pas	TChessForm	Освобождает ресурсы при удалении формы.
procedure FormKeyDown(Sender: TObject; var Key: Word; Shift: TShiftState);	PlayerU.pas	TChessForm	Обрабатывает события ввода с клавиатуры.
procedure PlayerRefreshTimer(Sender: TObject);	PlayerU.pas	PlayerRefresh (TTimer)	Периодически обновляет состояние игры и пользовательский интерфейс.
procedure imgCloseClick(Sender: TObject);	PlayerU.pas	imgClose (TImage)	Закрывает приложение при нажатии кнопки закрытия.
procedure imgCloseMouseEnter(Sender: TObject);	PlayerU.pas	imgClose (TImage)	Изменяет изображение кнопки закрытия, когда мышь входит в ее область.
procedure imgCloseMouseLeave(Sender: TObject);	PlayerU.pas	imgClose (TImage)	Изменяет изображение кнопки закрытия, когда мышь покидает ее область.
procedure FormCreate(Sender: TObject);	PlayerU.pas	TChessForm	Инициализирует форму и ее компоненты при ее создании.
procedure setWhiteColorExecute(Sender: TObject);	PlayerU.pas	setWhiteColor (TAction)	Задает цвет белых фигур.
procedure setBlackColorExecute(Sender: TObject);	PlayerU.pas	setBlackColor (TAction)	Задает цвет черных фигур.
procedure setOutlineColorExecute(Sender: TObject);	PlayerU.pas	setOutlineColor (TAction)	Задает цвет контуров фигур.
procedure setBackColorExecute(Sender: TObject);	PlayerU.pas	setBackColor (TAction)	Задает цвет фона формы.
procedure autoDeselectExecute(Sender: TObject);	PlayerU.pas	autoDeselect (TAction)	Переключает функцию автоматического отмены выбора менеджера шахматной доски.

Продолжение таблицы 1

1	2	3	4
procedure SetSettings;	PlayerU.pas		Загружает настройки игры из файла.
procedure saveSettingsExecute(Sender: TObject);	PlayerU.pas	saveSettings (TAction)	Сохраняет текущие настройки игры.
procedure showDebugExecute(Sender: TObject);	PlayerU.pas	showDebug (TAction)	Показывает режим отладки.
procedure resetSettingsExecute(Sender: TObject);	PlayerU.pas	resetSettings (TAction)	Сбрасывает настройки игры к значениям по умолчанию.
procedure reloadGame;	PlayerU.pas		Перезагружает состояние игры и пользовательский интерфейс.
procedure setAssetsPathExecute(Sender: TObject);	PlayerU.pas	setAssetsPath (TAction)	Задаёт путь к игровым ресурсам.
procedure roundEdges;	PlayerU.pas		Скругление краев формы
procedure ScaleComponents;	PlayerU.pas		Масштабирует компоненты пользовательского интерфейса на основе текущего коэффициента масштабирования.
procedure FormMouseDown(Sender: TObject; Button: TMouseButton;	PlayerU.pas	TChessForm	Обрабатывает события нажатия кнопки мыши.
procedure setUIScaleExecute(Sender: TObject);	PlayerU.pas	setUIScale (TAction)	Устанавливает коэффициент масштабирования пользовательского интерфейса.
procedure exitButtonClick(Sender: TObject);	PlayerU.pas	exitButton (TButton)	Закрывает приложение при нажатии кнопки выхода.
function GetData(FilePath : string; Tag : string) : string;	PlayerU.pas		Извлекает данные из файла на основе тега.
function PickPawnPromotion : byte;	EngineUI.pas		Отображает диалоговое окно для выбора пользователем фигуры для продвижения пешки.
procedure TSquare.SetCords(const Value: TPoint);	Engine-Classes.pas	TSquare	Устанавливает координаты объекта TSquare

Продолжение таблицы 1

1	2	3	4
procedure TSquare.Click(Sender:TObject);	Engine- Classes.pas	TSquare	Обрабатывает нажатие на объект TSquare
procedure TSquare.SetForwardClick(const Value: TForward);	Engine- Classes.pas	TSquare	Устанавливает обработчик события нажатия на объект TSquare
procedure TSquare.SetKind(const Value: integer);	Engine- Classes.pas	TSquare	Устанавливает тип фигуры на объекте TSquare
procedure TSquare.SetColor(const Value: integer);	Engine- Classes.pas	TSquare	Устанавливает цвет объекта TSquare
procedure TSquare.SetPreKind(const Value: integer);	Engine- Classes.pas	TSquare	Устанавливает предыдущий тип фигуры на объекте TSquare
procedure TBoardMannager. SetDebug(const Value: TDebug);	Engine- Classes.pas	TBoardMannager	Устанавливает значение компонента TDebug
procedure TBoardMannager. Setbishop(const Value: TBitmap);	Engine- Classes.pas	TBoardMannager	Устанавливает значение растрового изображения слона.
procedure TBoardMannager. Sethorse(const Value: TBitmap);	Engine- Classes.pas	TBoardMannager	Устанавливает значение растрового изображения лошади.
procedure TBoardMannager. Setking(const Value: TBitmap);	Engine- Classes.pas	TBoardMannager	Устанавливает значение растрового изображения короля.
procedure TBoardMannager. Setknight(const Value: TBitmap);	Engine- Classes.pas	TBoardMannager	Устанавливает значение растрового изображения ладьи.
procedure TBoardMannager. Setpawn(const Value: TBitmap);	Engine- Classes.pas	TBoardMannager	Устанавливает значение растрового изображения пешки.
procedure TBoardMannager. Setqueen(const Value: TBitmap);	Engine- Classes.pas	TBoardMannager	Устанавливает значение растрового изображения ферзя.
procedure TBoardMannager. SetSelected(const Value: boolean);	Engine- Classes.pas	TBoardMannager	Устанавливает значение выбранного свойства.
procedure TBoardMannager. SetSelectedSqr(const Value: TSquare);	Engine- Classes.pas	TBoardMannager	Устанавливает значение выбранного квадрата.

Продолжение таблицы 1

1	2	3	4
procedure TBoardMannager. SetTurn(const Value: integer);	Engine-Classes.pas	TBoardMannager	Устанавливает ценность хода.
function TBoardMannager. GetBlackPiecesTook(index:integer):integer;	Engine-Classes.pas	TBoardMannager	Возвращает взятые черные фигуры.
function TBoardMannager. GetWhitePiecesTook(index:integer):integer;	Engine-Classes.pas	TBoardMannager	Возвращает взятые белые фигуры.
procedure TBoardMannager. SetBlackPiecesTook(Index: Integer; Value: Integer);	Engine-Classes.pas	TBoardMannager	Устанавливает взятые черные фигуры.
procedure TBoardMannager. SetWhitePiecesTook(Index: Integer; Value: Integer);	Engine-Classes.pas	TBoardMannager	Устанавливает взятые белые фигуры.
function TBoardMannager. Move(ASquare: TSquare; Abm : TBitmap) : integer;	Engine-Classes.pas	TBoardMannager	Перемещает фигуру на доске и возвращает тип перемещенной фигуры.
procedure TBoardMannager. TakePiece(ASquare: TSquare; Abm : TBitmap);	Engine-Classes.pas	TBoardMannager	Берет фигуру с доски.
procedure TBoardMannager. SetOrientation(const Value: Integer);	Engine-Classes.pas	TBoardMannager	Устанавливает ориентацию доски.
procedure TBoardMannager. SetAutoDeselect(const Value: boolean);	Engine-Classes.pas	TBoardMannager	Устанавливает значение свойства автоматической отмены выбора.
procedure TBoardMannager. SetBlackColor(const Value: TColor);	Engine-Classes.pas	TBoardMannager	Устанавливает черный цвет.
procedure TBoardMannager. SetOutlineColor(const Value: TColor);	Engine-Classes.pas	TBoardMannager	Устанавливает цвет контура.
procedure TBoardMannager. SetWhiteColor(const Value: TColor);	Engine-Classes.pas	TBoardMannager	Устанавливает белый цвет.
procedure TBoardMannager. SetPlayerNameBlack(const Value: string);	Engine-Classes.pas	TBoardMannager	Устанавливает имя черного игрока.
procedure TBoardMannager. SetPlayerNameWhite(const Value: string);	Engine-Classes.pas	TBoardMannager	Устанавливает имя белого игрока.

Продолжение таблицы 1

1	2	3	4
function TBoardMannager. getLastSquareLeft : integer;	Engine- Classes.pas	TBoardMannager	Возвращает координату X самого левого квадрата на доске.
function TBoardMannager. getSquareHeightWidth : integer;	Engine- Classes.pas	TBoardMannager	Возвращает высоту и ширину квадрата на доске.
function TBoardMannager. getBlackTookLength : integer;	Engine- Classes.pas	TBoardMannager	Возвращает количество взятых черных фигур.
function TBoardMannager. getWhiteTookLength : integer;	Engine- Classes.pas	TBoardMannager	Возвращает количество взятых белых фигур.
procedure TBoardMannager. Clear;	Engine- Classes.pas	TBoardMannager	Очищает доску.
procedure TBoardMannager. InvalidMove;	Engine- Classes.pas	TBoardMannager	Указывает на неверный ход и возвращает доску в предыдущее состояние.
procedure TBoardMannager. SetSquareTo(Location : TPoint; kind : integer);	Engine- Classes.pas	TBoardMannager	Устанавливает вид и растровое изображение квадрата на доске.
function TBoardMannager. CheckDetect : byte; overload;	Engine- Classes.pas	TBoardMannager	Проверяет, находится ли король под шахом.
function TBoardMannager. CheckDetect(APoint: TPoint; multiplier : integer) : boolean; overload;	Engine- Classes.pas	TBoardMannager	Проверяет, находится ли точка под шахом.
procedure SetLinkedBoard(const Value: TBoardMannager);	Engine- Classes.pas	TSaveManager	Устанавливает значение связанной платы.
procedure SetrootDir(const Value: string);	Engine- Classes.pas	TSaveManager	Устанавливает корневой каталог для сохранения и загрузки игр.
procedure SaveToFile(filepath : string);	Engine- Classes.pas	TSaveManager	Сохраняет текущее состояние игры в файл.
procedure SaveToFileOverwrite(filepath : string);	Engine- Classes.pas	TSaveManager	Сохраняет текущее состояние игры в файл.
procedure LoadFromFile(filepath : string);	Engine- Classes.pas	TSaveManager	Загружает состояние игры из файла.

Описание использованных компонентов приводится в таблице 2.

Таблица 2 – Использованные компоненты

Компонент	На какой форме расположен	Назначение
1	2	3
TImage	Loading_Screen, ChessForm, frmLauncher	Используется для задания фона, в качестве кнопки, для отображения изображений
TLabel	Loading_Screen, ChessForm, frmLauncher	Отображение надписей на форме
TProgressBar	Loading_Screen	Используется для отображения прогресса загрузки приложения
TTimer	Loading_Screen	Используется для определения длительности загрузочного экрана
TImageList	ChessForm	Используется для хранения набора изображений
TColorDialog	ChessForm	Вспользуется для изменения цветов различных элементов интерфейса, таких как цвет белых и черных фигур, контура и фона.
TActionList	ChessForm	Вспользуется для определения действий, которые могут быть применены к элементам меню в PopUpMenu.
PlayerRefresh	ChessForm	Используется для обновления игрового процесса на форме.
TCheckBox	frmLauncher	Используется для переключения между оконным и полноэкранным режимами при запуске игры.
MainMenu1	frmLauncher	Содержит пункты меню для справки и выхода.
MediaPlayer1	frmLauncher	Он загружает MP3-файл и воспроизводит его при отображении формы.
Panel1	frmLauncher	Используется для группировки других компонентов.
TShape	frmLauncher	Используются для создания пользовательских кнопок для запуска в игру
Tmr	frmLauncher	Используется для анимации кнопок
VolumeButton	frmLauncher	Используется для включения и выключения музыки

Точное название проекта и его состав приводится в Таблице 3.

Таблица 3 — Спецификация программы

Имя файла	Назначение
1	2
ChessP.exe	Исполняемый файл проекта
ChessP.dproj	Файл проекта, связывает все файлы из которых состоит приложение.
ChessP.exe	Установочный пакет программы
TheHappyBride.mp3	Музыкальный файл
load.png	Фотография загрузочного экрана
close.png	Фотография кнопки «Заккрыть»
6.bmp	Фотография дамы
5.bmp	Фотография пешки
4.bmp	Фотография короля
3.bmp	Фотография коня
2.bmp	Фотография ладьи
1.bmp	Фотография слона
PlayerU.dfm	Файл описания формы PlayerU
PlayerU.pas	Файл программного модуля для формы PlayerU
LoadU.pas	Файл программного модуля для формы LoadU
LoadU.dfm	Файл описания формы LoadU
Справка.chm	Файл справки, содержит помощь по работе с программой
Launcher.dfm	Файл описания формы Launcher
Launcher.pas	Файл программного модуля для формы Launcher

4 Тестирование

При разработке данной программы многие возникающие ошибки и недоработки были исправлены на этапе реализации проекта. После завершения испытания реализации программы было проведено тщательное функциональное тестирование. Функциональное тестирование должно гарантировать работу всех элементов программы в автономном режиме.

Отчёт о результатах тестирования предоставлен в таблице 4.

Таблица 4 – Отчёт результатах тестирования

№ теста	Тест	Ожидаемый результат	Физический результат	Результат тестирования
1	2	3	4	5
T1	Проверка запуска игры	Отображение окна загрузки и открытие главного окна	Отображение окна загрузки и открытие главного окна	Выполнено
T2	Проверка кнопки «Справка»	Открытие справочного файла	Открытие справочного файла	Выполнено
T3	Проверка кнопки «Музыка»	Выключение музыки	Выключение музыки	Выполнено
T4	Проверка кнопки «Выход»	Заккрытие проекта	Заккрытие проекта	Выполнено
T5	Проверка кнопки «Начать игру»	Начало игры (открытие формы, содержащей шахматную доску)	Начало игры (открытие формы, содержащей шахматную доску)	Выполнено
T6	Проверка кнопки-изображения, пешка	Перестановка пешки на новую позицию	Перестановка пешки на новую позицию	Выполнено
T7	Проверка кнопки-изображения, ладья	Перестановка ладьи на новую позицию	Перестановка ладьи на новую позицию	Выполнено

Продолжение таблицы 4

1	2	3	4	5
T8	Проверка кнопки-изображения, конь	Перестановка коня на новую позицию	Перестановка коня на новую позицию	Выполнено
T9	Проверка кнопки-изображения, слон	Перестановка слона на новую позицию	Перестановка слона на новую позицию	Выполнено
T10	Проверка кнопки-изображения, ферзь	Перестановка ферзя на новую позицию	Перестановка ферзя на новую позицию	Выполнено
T11	Проверка кнопки-изображения, король	Перестановка короля на новую позицию	Перестановка короля на новую позицию	Выполнено

При разработке программного продукта было решено множество проблем, например, не правильно производилось осуществлялась проверка на ход фигуры, поэтому в будущем пользователь не столкнется с данными проблемами.

Элементы программы были проверены, и было установлено, что все они работают правильно и выполняют задачи, указанные в процедурах.

5 Руководство пользователя

5.1 Общие сведения о программном продукте

Цель курсового проекта заключается в разработке программного продукта, который позволит пользователю приятно провести время, получить эстетическое удовольствие, поспособствует улучшению таких умений, как логическое мышление, навыки запоминания, принятие решений.

Игровое приложение ориентировано на пользователей любого возраста, интересующихся шахматами и желающих развивать свои логические и аналитические навыки.

Быстродействие любой программы во многом зависит от характеристик выбранного персонального компьютера: рабочей частоты процессора, объема оперативной памяти и т.д. Несмотря на все реализованные в ней задачи, она легко запускается и функционирует на любых машинах.

Тестирование проводилось на разных классах ЭВМ и работать с данной программой было комфортно. Программа разработана на ПК со следующими характеристиками:

- модель процессора 12th Gen Intel(R) Core(TM) i5-12500H 2.50 GHz объем ОЗУ 16 Гб;
- графический адаптер NVIDIA GeForce RTX 3060 6 Гб;
- операционная система Windows 11.

5.2 Инсталляция

Для того, чтобы установить программу необходимо запустить файл Setup.exe.

Приложение встречает пользователя экраном заставки, после чего мы видим кнопку "Играть", и в верхнем левом углу есть кнопка "Закрыть" и кнопка "Справка", после чего необходимо щелкнуть по кнопке "Играть" для полноценного запуска приложения.

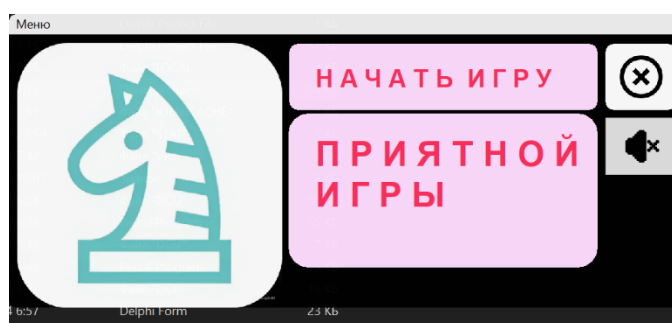


Рисунок 3 – Главное меню

На данной доске находится 32 фигуры разных цветов.

					КП 2-40 01 01.35.40.13.24	Лист
Изм.	Лист	№докум.	Подпись	Дата		22

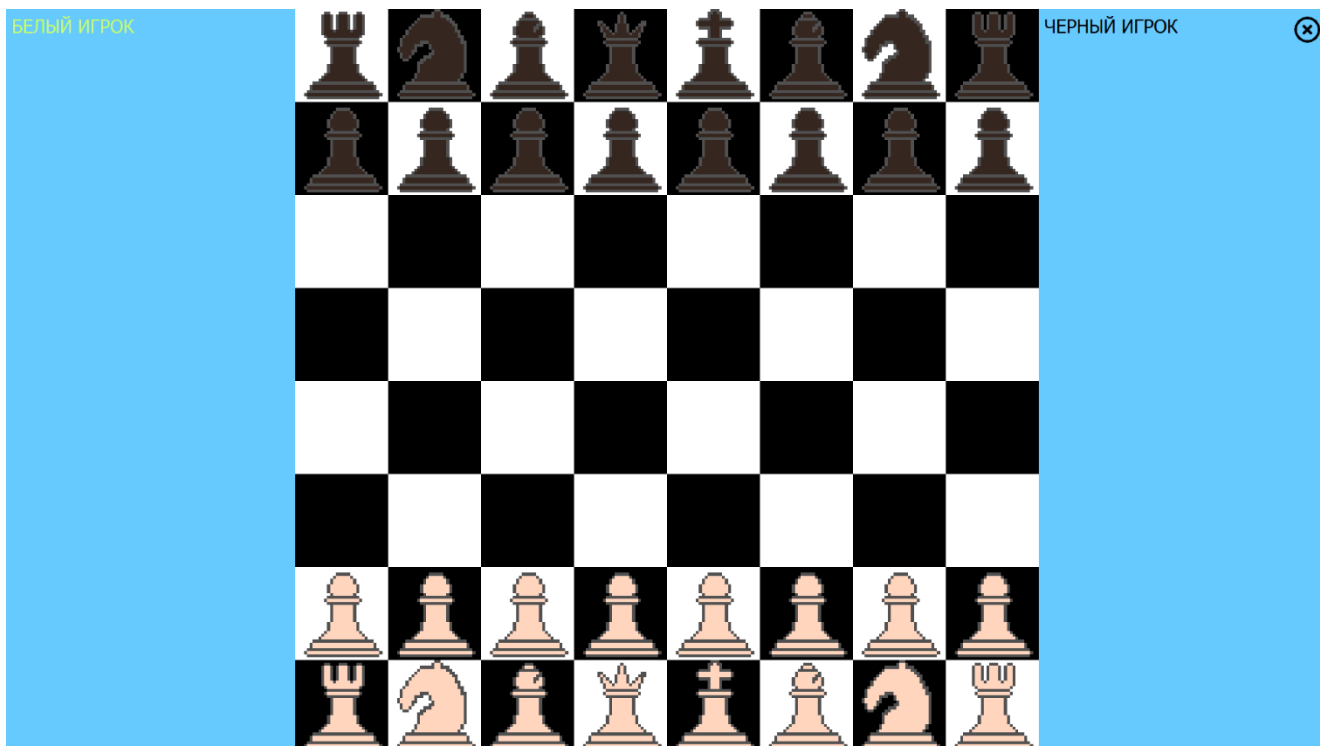


Рисунок 4 – Геймплей

Начальная расстановка фигур:

В начале игры фигуры располагаются следующим образом:

- пешки выстраиваются в ряды перед главными фигурами;
- ладьи занимают угловые клетки;
- кони находятся рядом с ладьями;
- слоны располагаются рядом с конями;
- королева ставится на клетку своего цвета (белая на белую, черная на черную);
- король занимает оставшуюся центральную клетку.

Правила движения фигур:

- король может ходить на одну клетку в любом направлении;
- ферзь (королева) может ходить на любое количество клеток в любом направлении;
- ладья ходит на любое количество клеток по вертикали или горизонтали;
- слон ходит на любое количество клеток по диагонали;
- конь ходит буквой "Г": две клетки в одном направлении и затем одна клетка перпендикулярно;
- пешка ходит на одну клетку вперед, но может сделать два шага вперед с начальной позиции. Пешка берет фигуры противника, перемещаясь на одну клетку по диагонали.

Особые ходы:

– рокировка: одновременный ход короля и ладьи, при котором король перемещается на две клетки в сторону ладьи, а ладья ставится рядом с королем с противоположной стороны;

– взятие на проходе: если пешка делает ход на два поля вперед и встает рядом с пешкой противника, та может взять ее, как если бы она сделала ход только на одно поле;

– превращение пешки: когда пешка достигает противоположного конца доски, она может превратиться в любую другую фигуру (кроме короля).

Уникальность шахмат

– стратегия и тактика: Шахматы сочетают стратегическое планирование с тактическими маневрами. Игроки должны продумывать свои ходы на несколько шагов вперед и предугадывать действия противника;

– разнообразие партий: Каждая партия шахмат уникальна благодаря множеству возможных комбинаций ходов и стратегий.;

– образовательная ценность: Шахматы развивают логическое мышление, концентрацию и умение решать проблемы.

Шахматные композиции и задачи: Существуют специальные шахматные задачи и этюды, которые тренируют решение конкретных позиций и улучшение навыков игры.

					КП 2-40 01 01.35.40.13.24	Лист
Изм.	Лист	№докум.	Подпись	Дата		24

Заключение

Цель курсового проекта заключается в разработке программного продукта, который позволит пользователю приятно провести время, получить эстетическое удовольствие, поспособствует улучшению таких умений, как логическое мышление, навыки запоминания, принятие решений.

Создаваемое приложение ориентировано на пользователей от 12 до 18 лет, но использовать его могут и просто люди, заинтересовавшиеся игрой.

В ходе выполнения данной курсовой работы было разработано и протестировано игровое приложение «Кто из нас Гроссмейстер». Приложение может быть использовано в развлекательных целях.

Данный программный продукт (игровое приложение) является законченной, полнофункциональной программой.

Использование среды разработки Delphi11 при программной реализации, дало возможность создать качественное современное программное обеспечение, обладающее высокой скоростью работы, удовлетворяющее всем уровням требований, предъявляемых к интерфейсу, обеспечивающее эффективность и гибкость работы.

Игра легко модифицируется и масштабируется путем добавления новых элементов таких как:

- игра между человеком и компьютером;
- возможность проходить уровни различной сложности;
- добавить отслеживание прямых эфиров по шахматам.

Список использованных источников

1. Валерий Рубанцев «Большой самоучитель по дэлфи» [Электронный ресурс] — Режим доступа: <https://drive.google.com/file/d/17yCnaW15Cg7qRuTQHTDtMdhGXDBgK8-W/view> — Дата доступа: 20.06.2024
2. Методические указания по построению, изложению и оформлению документов учебной и научной деятельности учащихся [Электронный ресурс] — Режим доступа: <https://drive.google.com/file/d/1gumRnoL0ytSghvwzCzK45ype5925sbyv/view> — Дата доступа: 25.06.2024
3. Статья на Дзене [Электронный ресурс] — Режим доступа: <https://dzen.ru/a/ZAYfD3nRki-TfWj5> Дата доступа: 18.06.2024
4. Программирование на языке Delphi [Электронный ресурс] — Режим доступа: https://www.bsuir.by/m/12_100229_1_90135.pdf Дата доступа: 27.05.2024