|  |  |
| --- | --- |
| Министерство науки и высшего образования Российской Федерации  ОРСКИЙ ГУМАНИТАРНО-ТЕХНОЛОГИЧЕСКИЙ ИНСТИТУТ (ФИЛИАЛ)  ФЕДЕРАЛЬНОГО ГОСУДАРСТВЕННОГО БЮДЖЕТНОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ ВЫСШЕГО ОБРАЗОВАНИЯ  «ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»  Факультет среднего профессионального образования | |
| **Курсовая работа**  по междисциплинарному курсу «Технология разработки программного обеспечения»  профессионального модуля «Осуществление интеграции программных модулей»    **Разработка компьютерной игры «Переверни фишки»**  Пояснительная записка  ОГУ 09.02.07. 3024. 313 ПЗ | |
|  | Руководитель работы  преподаватель высшей категории  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ж. В. Михайличенко  «\_\_\_»\_\_\_\_\_­\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.  Студент группы 22ИСП-1  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Н. Е. Павлов  «\_\_\_»\_\_\_\_\_­\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. |
| Орск 2024 | |

|  |  |
| --- | --- |
| Утверждаю  председатель ПЦК дисциплин профессионального цикла | |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  подпись | Ж.В. Михайличенко |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г. | |

**ЗАДАНИЕ**

**на выполнение курсовой работы**

студенту \_\_\_\_\_Павлову Николаю Евгеньевичу\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

фамилия, имя, отчество

по специальности \_\_\_09.02.07 Информационные системы и программирование\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

по междисциплинарному курсу \_\_Технология разработки программного обеспечения\_\_\_\_\_\_\_\_\_

1. Тема работы: \_\_\_Разработка компьютерной игры «Переверни фишки» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Срок сдачи студентом работы «10» \_\_июня\_\_\_\_ 2024 г.
3. Цель и задачи работы \_\_Программно смоделировать компьютерную игру для проверки внимательности и логики пользователя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Исходные данные к работе: \_\_Учебники и интернет-источники по технологии разработки программного обеспечения и программирования на языке С#\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. Перечень вопросов, подлежащих разработке: \_ а) изучить предметную область, выполнить анализ требований к программному обеспечению, составить техническое задание на разработку; б) выполнить проектирование системы с помощью CASE-средств; в) для решения поставленной задачи реализовать оконное приложение на языке C# и протестировать его; г) сформулировать предложения по внедрению, эксплуатации и сопровождению разработанного программного обеспечения. Сделать выводы по результатам проделанной работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
6. Перечень графического (иллюстративного) материала: таблицы, графики, рисунки, схемы, отражающие теоретический материал и программную реализацию поставленной задачи\_\_\_\_

Дата выдачи и получения задания

Руководитель «19» \_февраля\_\_\_\_\_ 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Ж.В. Михайличенко\_\_\_

подпись инициалы, фамилия

Студент «19» \_февраля\_\_\_\_\_ 2024 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_Н.Е. Павлов\_\_\_\_\_\_\_\_\_

подпись инициалы, фамилия

**Аннотация**

Изм.

Лист

№ докум.

Подпись

Дата

Лист

3

ОГУ 09.02.07. 3024. 313 ПЗ

Разраб.

Павлов Н.Е.

Провер.

Михайличенко Ж

Реценз.

Н. Контр.

Утверд.

Разработка компьютерной игры «Переверни фишки»

Лит.

Листов

\*

22ИСП-1

В курсовой работе по междисциплинарному курсу «Технология разработки программного обеспечения» профессионального модуля «Осуществление интеграции программных модулей» проведена разработка компьютерной логической игры «Переверни фишки».

В первой главе курсовой работы проведён анализ требований и проектирование программного продукта.

Во второй главе курсовой работе описывается процесс реализации и тестирования программного продукта.

В третьей главе курсовой работы находятся рекомендации по внедрению, эксплуатации и сопровождению программного обеспечения.

Пояснительная записка содержит 23 страницы, в том числе 6 рисунков, 1 таблицу, 10 источников, 1 приложение.

Разработка приложения выполнена \*\*\*.

**Содержание**

[Введение 5](#_Toc168498473)

[1 Анализ требований и проектирования программного продукта 6](#_Toc168498474)

[1.1 Анализ предметной области 6](#_Toc168498475)

[1.2 Техническое задание на разработку 7](#_Toc168498476)

[1.3 Построение модели 10](#_Toc168498477)

[2 Реализация и тестирование программного продукта 13](#_Toc168498478)

[2.1 Обоснование программных средств реализации 13](#_Toc168498479)

[2.2 Разработка пользовательского интерфейса 14](#_Toc168498480)

[2.3 Алгоритмизация и программирование 15](#_Toc168498481)

[2.4 Тестирование 17](#_Toc168498482)

[3 Рекомендации по внедрению, эксплуатации и сопровождению программного обеспечения 18](#_Toc168498483)

[3.1 Руководство пользователя 18](#_Toc168498484)

[3.2 План внедрения и сопровождения 20](#_Toc168498485)

[Заключение 21](#_Toc168498486)

[Список использованных источников 22](#_Toc168498487)

[Приложение А 23](#_Toc168498488)

# Введение

Актуальность темы курсовой работы по разработке компьютерной логической игры «Переверни фишки» заключается в том, что компьютерные игры остаются популярным развлечением для людей всех возрастов. Разработка логической игры может способствовать развитию креативного мышления, логики и аналитических способностей у пользователей. Кроме того, такая работа может быть актуальна с точки зрения изучения современных методов программирования и разработки игровых приложений.

Целью курсовой работы по разработке компьютерной логической игры «Переверни фишки» является изучение и применение принципов программирования, разработки игровых алгоритмов и создание интерактивного пользовательского интерфейса. Основной задачей является не только создание игры, но и изучение процесса её разработки с использованием современных технологий и методов программирования.

Для достижения поставленной цели необходимо решить следующие задачи:

* Изучение методов разработки компьютерных игр. Это включает в себя изучение основных концепций разработки игровой логики, алгоритмов и структур данных, которые необходимы для создания игрового процесса.
* Анализ аналогичных игр. Проведение исследования существующих игр со схожим замыслом создания, анализ их особенностей и механики для понимания того, что можно заимствовать и улучшить, а также чего следует избегать.
* Выбор программных приложений. Выбрать средство документирования, средство моделирования бизнес-процессов, среду программирования для разработки компьютерной логической игры «Переверни фишки», учитывая требования к созданию игровых алгоритмов, пользовательского интерфейса
* Проектирование игры. Создание документации, включающей в себя план игры, описание механики игры, игровых уровней, интерфейса и других компонентов.
* Разработка программы «Переверни фишки». Непосредственное создание программного продукта на основе разработанных ранее концепций и документации.
* Тестирование и отладка. Проведение тестирования игры на наличие ошибок, корректировка и оптимизация кода, улучшение игрового процесса и интерфейса на основе обратной связи.
* Анализ результатов. Оценка полученного опыта, выявление проблем и способы их решения, а также формулирование рекомендаций для дальнейшего развития игры.

Таким образом, цель курсовой работы по разработке компьютерной логической игры «Переверни фишки» заключается не только в создании самой игры, но и в освоении процесса разработки игр, погружении в мир игровой индустрии и приобретении практического опыта в данной области.

# Анализ требований и проектирования программного продукта

## Анализ предметной области

Благодаря развитию рынка инди-игр, многие разработчики компьютерных игр получили возможность работать над своими игровыми проектами без финансовых и юридических обязательств перед компаниями-издателями. Инди-игры — это компьютерные игры, созданные отдельными разработчиками или небольшими коллективами без финансовой поддержки издателя компьютерных игр. Распространение осуществляется посредством каналов цифровой дистрибуции.

Данная курсовая работа относится к инди-разработке. Благодаря развитию программного обеспечения разработки компьютерных игр, команде разработчиков не требуется тратить несколько лет на разработку игрового движка. Это позволяет сразу же приступать к непосредственной работе над игровым проектом и значительно сокращает время его разработки.

Игра «Переверни фишки» представляет собой головоломку, в которой игроку предлагается игровое поле размером четыре на четыре клетки, заполненное фишками двух разных цветов, которые были выбраны пользователем. При щелчке мыши на фишку, его цвет и цвет соседних по вертикали и горизонтали фишек меняется на противоположный. После щелчка в центре игрового поля цвет изменят пять кружков, на краю поля — четыре, а в углу — только три кружка Игра завершается, если цвета всех фишек на игровом поле одинаковы.

Пример игрового поля показан на рисунке 1.

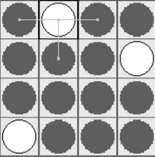


Рисунок 1 – Пример игрового поля игры «Переверни фишки»

Для успешной реализации игры необходимо провести анализ существующих игр с подобной механикой, определить основные элементы игрового процесса, разработать удобный и интуитивно понятный интерфейс для пользователя, а также написать программный код, который будет обрабатывать действия игрока.

Программный продукт будет представлять собой приложение, реализующее игру «Переверни фишки». Проект будет конкурировать не только с другими реализациями похожих головоломок, но и с другими простыми логическими играми, такими как, например, «Судоку» или «2048».

Целевую аудиторию составляют люди, которые хотят играть в головоломки и развивать свои логические способности. Игроки будут иметь возможность играть в «Переверни фишки», стремясь достичь цели минимальным числом ходов.

Разрабатываемый программный продукт будет отличаться от конкурентов простотой и минималистичным дизайном, что делает игру легко воспринимаемой и не отвлекает пользователя от игрового процесса.

Программа будет полезна для тех, кто хочет развивать своё мышление, логику, и просто приятно провести время, сыграв пару партий в «Переверни фишки». Для игры не требуются какие-то специфические знания, поэтому в неё сможет играть абсолютно кто угодно.

Основной целью данного программного продукта является его распространение с целью предоставления пользователям интересной и полезной игры для развития логического мышления.

Таким образом, создание игры «Переверни фишки» требует комплексного подхода, включающего в себя анализ, проектирование, программирование и тестирование игрового приложения.

## Техническое задание на разработку

Техническое задание на разработку программного обеспечения – это документ, который содержит подробное описание требований к программному продукту. Техническое задание на разработку компьютерной игры «Переверни фишки» составлена согласно ГОСТ 34.602-2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы» и представлена ниже.

1. Общие сведения.
   1. Наименовании автоматизированной системы (АС): Компьютерная логическая игра «Переверни фишки»;
   2. Наименование заказчика: Факультет среднего профессионального образования Орского гуманитарно-технологического института (филиала) ОГУ в лице преподавателя высшей категории Михайличенко Ж.В.;
   3. Наименование разработчика: студент второго курса группы 22ИСП-1 Павлов Н.Е.;
   4. Документ, на основании которого создается АС: протокол закрепления тем курсовой работы по дисциплине «Технология разработки программного обеспечения» от 19.02.2024 года.;
   5. Дата начала работ: 19.02.2024;
   6. Дата окончания работ: 10.06.2024.
2. Цели и назначение создания АС.
   1. Цели создания АС: интересное и увлекательное времяпровождение, стимулирующее их критическое мышление, стратегическое планирование и навыки решения задач. Эта игра также помогает развивать внимание и ментальную гибкость, поскольку игрокам требуется постоянно адаптироваться к изменяющемуся расположению фишек на доске.
   2. Назначение АС: игровая деятельность.
3. Характеристика объекта автоматизации.

Объектом автоматизации является логическая игра «Переверни фишки». Игра предназначена для одного человека. На столе разложены 16 фишек разного цвета в четыре ряда, и игрок, переворачивая фишки по определенным правилам, должен сделать так, чтобы все фишки были одного цвета.

1. Требования к АС.
   1. Требования к функциям, выполняемыми АС:
      * функция начала новой игры, создающая игровую область из 16 фишек различного цвета;
      * функция для переворота фишек, переворачивающая соседние фишки при нажатии на фишку;
      * функция проверки цветов фишек, выдающая сообщение о победе, если все фишки одного цвета;
      * функция запоминания лучших результатов, которая запоминает минимальное количество времени и ходов для достижения цели.
   2. Требования к видам обеспечения АС:
      * требования к математическому обеспечению: программный продукт не требует специального математического аппарата. Достаточно вести подсчёт количества нажатий на игровые фишки и проверка расположения фишки на игровой области;
      * требования к информационному обеспечению: исходные данные должны вводится пользователем, вывод лучшего и текущего результатов на экран монитора в виде сообщений;
      * требования к лингвистическому обеспечению: интерфейс системы будет на русском языке. Для организации диалога с пользователем должны использоваться диалоговые окна;
      * требования к программному обеспечению: операционная система: Windows. Язык программирования: C#. Среда программирования: Microsoft Visual Studio 2019. Средство документирования: Microsoft Word. Средство моделирования бизнес-процессов: Ramus Educational;
      * требования к техническому обеспечению: система должна быть совместима с большинством современных персональных компьютеров (ПК) и не требовать высоких аппаратных ресурсов;
      * требования к организационному обеспечению: при взаимодействии с АС пользователь может: создавать новую игру, сохранять прогресс, переворачивать фишки. Так же он может просматривать рекорды, которые хранят игру, в которой было произведено минимальное количество ходов при достижении цели;
   3. Общие технические требования к АС
      * требования к численности и квалификации пользователей: игра предназначена для одного пользователя. Система должна быть понятна и легка для использования для пользователей любой возрастной категории;
      * требования к эргономике и технической эстетике: интерфейс игрового поля должен быть не ярким (например, серый). Цвет фишек не должен сливаться со цветом игрового поля (например, синий и красный цвета);
      * требования к защите информации от несанкционированного доступа: защита от ввода неверных данных;
2. Состав и содержание работ по созданию АС.

В таблице 1 показаны этапы разработки компьютерной игры «Переверни фишки».

Таблица 1 – Этапы разработки АС

|  |  |  |
| --- | --- | --- |
| Номер и название этапа разработки | Сроки выполнения | Содержание |
| 1 Анализ требований | 19.02.2024 – 05.03.2024 | Анализ предметной области, изучение программных аналогов, разработка технического задания на создание АС |
| 2 Проектирование | 06.03.2024– 20.03.2024 | Разработка диаграмм IDEF различного уровня, диаграмм вариантов использования. Выбор и обоснование технологий и инструментов разработки. Проектирование пользовательского интерфейса. |
| 3 Программирование | 21.03.2024 – 14.05.2024 | Разработка и программная реализация алгоритмов и методов ввода, хранения, обработки и вывода данных. Разработка пользовательского интерфейса, организация диалога с пользователем. Отладка работы системы. Документирование программного кода. |
| 4 Тестирование | 15.05.2024 –  31.05.2024 | Проведение функционального, модульного, интеграционного и системного тестирования. Исправление ошибок и несоответствий. |
| 5 Внедрение | 01.06.2024 –  10.06.2024 | Установка и настройка системы. Подготовка сопровождающей документации (пояснительной записки к курсовой работе, руководства пользователя и других).  Защита курсовой работы. |

Продолжение таблицы 1

|  |  |  |
| --- | --- | --- |
| Номер и название этапа разработки | Сроки выполнения | Содержание |
| 6 Эксплуатация и сопровождение | с 10.06.2024 | Разработка рекомендаций по эксплуатации и сопровождению системы. |

1. Порядок разработки АС.

Этапы разработки будут выполняться в соответствии с приведенной в пункте пять таблицей этапов разработки АС с обязательным контролем и консультацией с заказчиком.

1. Порядок контроля и приемки АС.

Контроль за разработкой осуществляется заказчиком и принимается в указанные сроки – 10.06.2024.

1. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу АС в действие: для использования игрового приложения достаточно скопировать папку с программным продуктом и запустить из неё файл с расширением .exe. Вся справочная информация с правилами игры должна хранится в интерфейсе игры.
2. Требования к документированию.
   * + Документирование программного кода (комментарии);
     + Пояснительная записка к курсовой работе;
     + Встроенная в программу инструкция с правилами игры;
     + Руководство пользователя.
3. Источники разработки.
   * + Протокол закрепления тем курсовой работы по дисциплине «Технология разработки программного обеспечения» от 19.02.2024 года;
     + ГОСТ 34.602-2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы»;
     + Работы студенческие. Общие требования и правила оформления. СТО 02069024.101 – 2015. – Оренбург: Изд-во ОГУ, 2015. – 89 с;
     + ГОСТ 19.701-90 «Единая система программной документации. Схемы алгоритмов, программ, данных и систем.»;
     + ГОСТ 34.601-90 «Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания».

## Построение модели

Построение функциональной модели поможет определить ключевые компоненты игры и их взаимосвязи. Это облегчит процесс разработки программного кода и позволит создать более эффективное и стабильное приложение. Понимание всех функциональных элементов игры поможет создать более удобный и привлекательный пользовательский интерфейс. Это повысит удовлетворение пользователя от процесса игры.

Диаграммы IDEF0 обеспечивают структурированный подход к описанию бизнес-процессов, что обеспечивает лучшее понимание происходящих процессов. Они позволяют наглядно представить взаимосвязи между различными элементами системы и их влияние на результат работы.

Для построения диаграммы в формате IDEF0 выбрана программный продукт Ramus Educational. Ramus Educational, при помощи которого можно создавать визуальные диаграммы в форматах IDEF0 и DFD, используемые для наглядного отображения различных бизнес-процессов.

Для создания функциональной модели IDEF0 разрабатываемого программного обеспечения необходимо построить контекстную диаграмму, состоящую из единственного функционального блока А0 «Функционирование компьютерной логической игры «Переверни фишки». Входные данные этого блока: переворачиваемая фишка и палитра цветов фишек. Управление осуществляется по правилам игры, а механизмами являются пользователь и компьютер. Выходными данными является сообщение о завершении игры, количество сделанных пользователем ходов и длительность игры.

Контекстная диаграмма IDEF0 показана на рисунке 2.

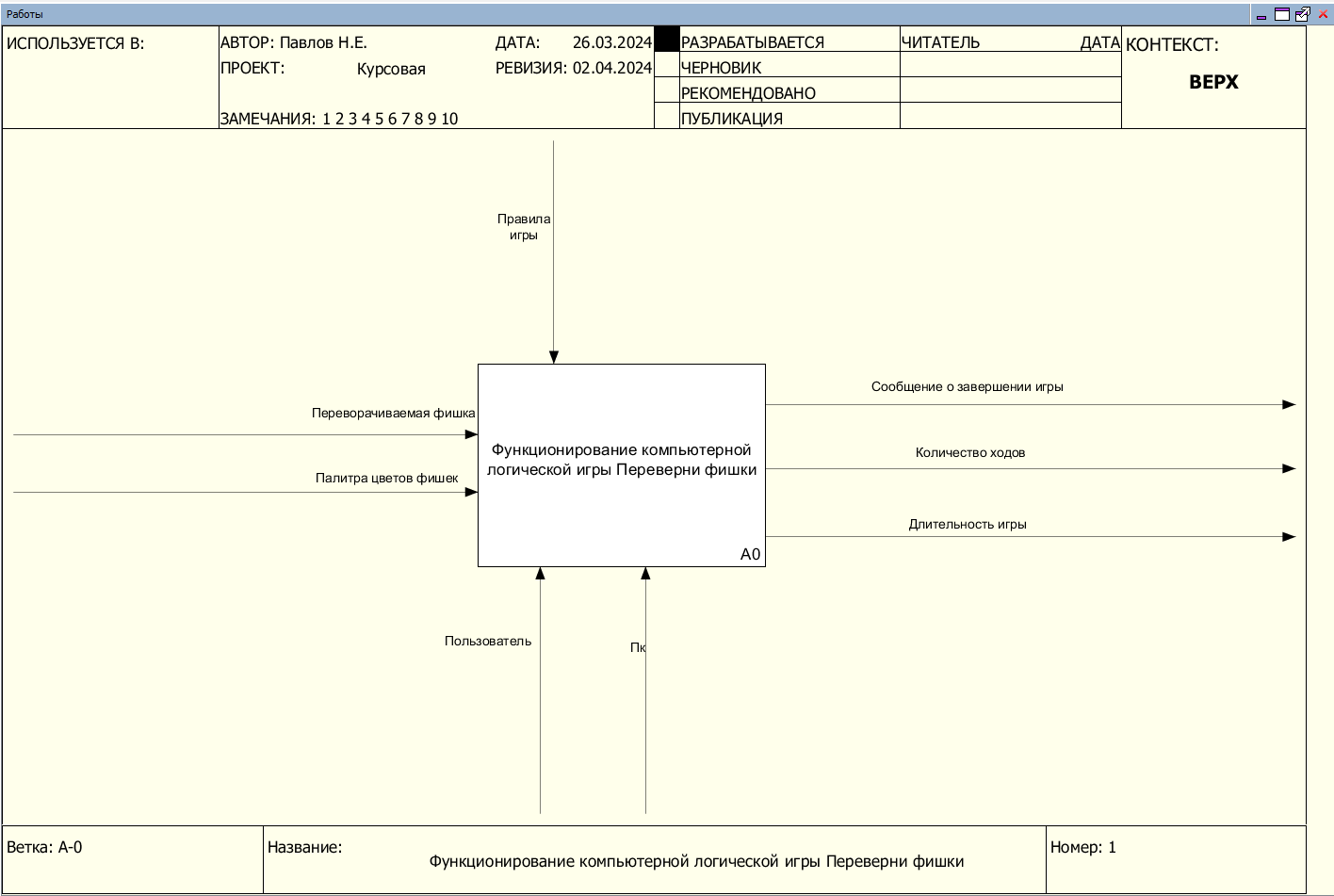


Рисунок 2 – Контекстная диаграмма IDEF0

Для лучшего понимания функций системы необходимо провести декомпозицию контекстной диаграммы, определив следующие функции:

A1 – «Выбор цвета фишек», где на входе указана палитра цветов фишек. Механизмы – пользователь и ПК. На выходе функции A1 мы получаем выбранный цвет фишек.

A2 – «Игровой процесс», на вход подаются переворачиваемая фишка и выбранный цвет фишек. Механизмы – пользователь и ПК, а управлением служат правила игры, на выходе мы получаем количество фишек первого и второго цвета.

A3 – «Проверка окончания игры» на вход блока подаются: количество фишек первого и второго цвета. В качестве механизма используется ПК, управлением служат правила игры. В случае выполнения требований для завершения игры, на выходе мы получаем сообщение о завершении игры, количество ходов и длительность игры, иначе возвращаемся в функцию A2. В случае завершения игры, пользователь может начать новую игру, что перенесёт действие в функцию А1.

Диаграмма декомпозиции первого уровня показана на рисунке 2.

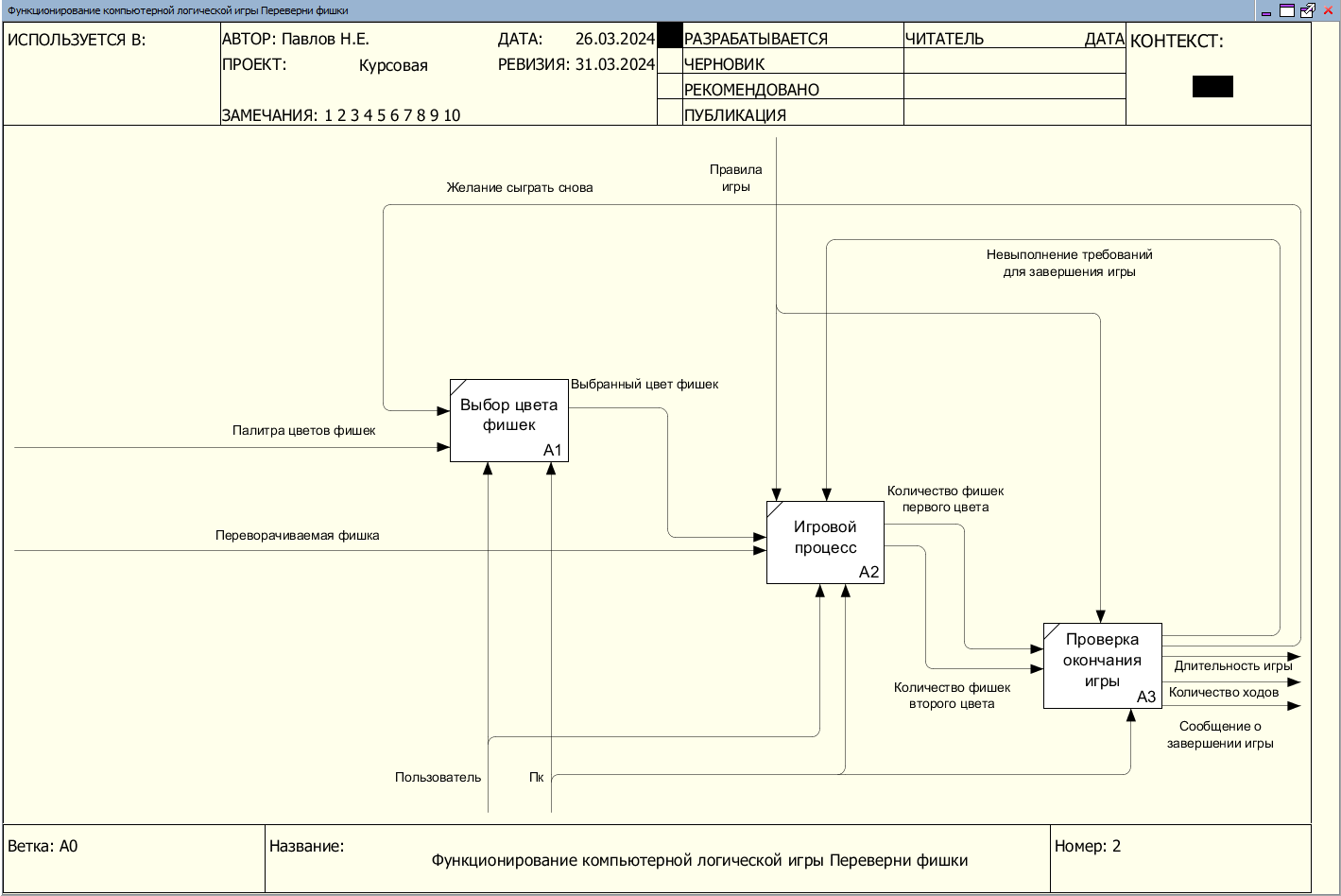


Рисунок 3 – Диаграмма декомпозиции первого уровня

Все построенные функциональные диаграммы IDEF0 будут использованы при разработке программного обеспечения «Компьютерная логическая игра «Переверни фишки».

# Реализация и тестирование программного продукта

## Обоснование программных средств реализации

Для разработки компьютерной логической игры «Переверни фишки» решено использовать язык программирования C# и интегрированную среду разработки Visual Studio 2019, тип проекта – Windows Forms на платформе .NET Framework.

Язык C# является мощным и эффективным объектно-ориентированным языком программирования, разработанным компанией Microsoft. Он широко используется для создания разнообразных приложений, включая десктопные приложения, веб-приложения, игры, мобильные приложения и многое другое.

Основными преимуществами языка C# являются:

* простота и понятность синтаксиса, что делает его отличным выбором для начинающих программистов;
* обширная документация и активное сообщество разработчиков, которые всегда готовы помочь с возникающими вопросами. Также стоит отметить поддержку платформы .NET, которая позволяет легко создавать кроссплатформенные приложения с использованием C#;
* мощная система типов и проверка типов во время компиляции, что помогает предотвращать ошибки и облегчает разработку;
* возможность использования средств параллельного программирования для эффективной работы с многопоточностью.

Visual Studio – это интегрированная среда разработки, которая предлагает широкий спектр инструментов для разработки на различных языках, включая C#. Она предлагает функции, такие как автодополнение кода, поддержка отладки, профилирование, автоматическое форматирование кода и многое другое.

Visual Studio поддерживает разработку на многих платформах, включая Windows, Android и iOS, благодаря интеграции с Xamarin и поддержке C++. Это делает её универсальным инструментом для разработки приложений.

Одним из ключевых преимуществ Visual Studio является ее тесная интеграция с C#. Среда обладает следующими возможностями и преимуществами:

* интуитивно понятный интерфейс, который облегчает создание, отладку и тестирование приложений;
* богатый набор инструментов для управления проектами, контроля версий, анализа кода и автоматизации процессов разработки;
* встроенная поддержка отладки, что помогает быстро находить и исправлять ошибки в коде;
* возможность интеграции с другими сервисами и инструментами разработки.

Все вышеперечисленные преимущества повлияли на выбор данного программного обеспечения для разработки компьютерной логической игры «Переверни фишки».

## Разработка пользовательского интерфейса

Программный интерфейс приложения, также известный как Application Programming Interface (API) – это способ взаимодействия двух или более компьютерных программ или компонентов. Это тип программного интерфейса, предлагающий услугу другим частям программного обеспечения. Документ или стандарт, который описывает, как создать или использовать такое соединение.

Основные особенности пользовательского интерфейса: интуитивность, удобство использования, совместимость с различными устройствами и операционными системами, адаптивность под разные экраны, надежность, поддержку многоязычности, соответствие стандартам и обратную связь.

Требования к программному интерфейсу:

* надёжность и стабильность, чтобы обеспечить бесперебойную работу приложений;
* совместимость с различными операционными системами и платформами;
* обеспечение безопасности данных и функций, которые он предоставляет;
* лёгкость в использовании для разработчиков, чтобы упростить процесс разработки;
* наличие чёткой и полной документации, чтобы разработчики могли легко понять, как использовать его функции и процедуры.

Программный интерфейс компьютерной логической игры «Переверни фишки» должен соответствовать всем вышеперечисленным требованиям. Структура интерфейса и его компоненты показаны на рисунке 4.

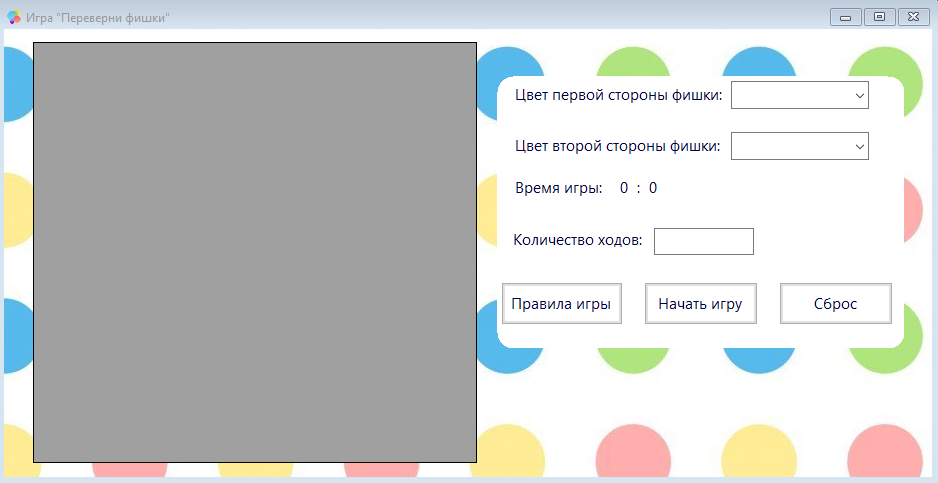


Рисунок 4 – Программный интерфейс игры «Переверни фишки»

Интерфейс приложения является однооконным. На форме расположены следующие компоненты WindowsForm:

* comboBox1 – для выбора цвета одной стороны фишки;
* comboBox2 – для выбора цвета обратной стороны фишки;
* label1 – хранит длительность игры;
* rulesButton – для просмотра правил игры;
* start\_game\_button – для старта игры;
* reset\_Button– для преждевременного окончания игры;
* dataGridView1 – элемент для генерации игрового поля, на котором нужно переворачивать фишки.

## Алгоритмизация и программирование

Алгоритм – чёткое описание последовательности действий, которые необходимо выполнить для решения поставленной задачи.

Алгоритм обладает основными свойствами:

* Конечность (результативность): определяет возможность получения результата за конечное число шагов.
* Дискретность: любой процесс алгоритма можно разбить на определенные этапы или шаги.
* Определённость: каждое правило алгоритма должно быть чётким и однозначным, а также выполниться в заданной последовательности.
* Массовость: получение результата при любых исходных данных.

В процессе разработки алгоритма могут использоваться различные способы описания алгоритмов, например, словесная запись, псевдокод, схемы и так далее.

Алгоритмы удобно представлять в виде блок-схем согласно ГОСТ 19.701-90 «Единая система программной документации. Схемы алгоритмов, программ, данных и систем.». Для компьютерной логической игры «Переверни фишки» блок-схема алгоритма представлена на рисунке 5.

Старт

Выбор цветов сторон фишек

1

Рисунок 5 – Блок-схема алгоритма компьютерной логической игры «Переверни фишки», часть 1

Продолжение блок-схемы алгоритма логической игры «Переверни фишки» показано на рисунке 6.

1

Переворачивание выбранной и соседних ей фишек

Цвета всех фишек совпадают

Время игры и количество ходов

Конец

+

-

Рисунок 6 – Блок-схема алгоритма компьютерной логической игры «Переверни фишки», часть 2

Для реализации функций компьютерной логической игры «Переверни фишки» разработаны следующие методы:

1. private void timer1\_Tick(object sender, EventArgs e) – метод для обработки события «Tick» таймера;
2. private void generationGameTable() – метод заполнения игрового поля фишками выбранных цветов;
3. private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e) – метод для изменения цвета выбранной и соседним ей фишек и прибавления единицы к количеству ходов при одном нажатии на ячейку;
4. private void ClearDataGridView() – метод очистки игрового поля;
5. private void reset\_Button\_Click(object sender, EventArgs e) – метод сброса состояния игры при нажатии на кнопку;
6. private void resetAll() – метод сброса состояния игры;
7. private void start\_game\_button\_Click(object sender, EventArgs e) – метод для старта игры при нажатии на кнопку;
8. private void rulesButton\_Click(object sender, EventArgs e) – метод вызова окна с правилами игры при нажатии на кнопку;
9. private void FlipCellColor(int row, int col) – метод проверки положения выбранной фишки и изменения её цвета;
10. private bool CheckWin() – метод проверки цветов фишек на игровом поле;
11. private void dataGridView1\_CellDoubleClick(object sender, DataGridViewCellEventArgs e) – метод для изменения цвета выбранной и соседним ей фишек и прибавления единицы к количеству ходов при двойном нажатии на ячейку;
12. private Color GetColor1() – метод получения первой стороны фишки;
13. private Color GetColor2() – метод получения обратной стороны фишки.

## Тестирование

Текст

# Рекомендации по внедрению, эксплуатации и сопровождению программного обеспечения

## Руководство пользователя

1. Системные требования

Для стабильной и эффективной работы игры «Переверни фишки» рекомендуется использовать следующую конфигурацию:

Частота процессора (CPU): 1,6 GHz

Количество ядер процессора (CPU): 3

Объем оперативной памяти (RAM): 2 GB

Объем свободного места на диске (HDD): 2 GB

Операционная система (OS): Windows

Браузер: Google Chrome, Mozilla Firefox, Microsoft Edge

1. Начало работы

Данный раздел поможет вам быстро установить, настроить и начать работать с игрой «Переверни фишки».

Перед началом работы, пожалуйста, ознакомьтесь с системными требованиями.

1. Основные понятия и термины

Перед началом работы с игрой рекомендуем ознакомиться с основными понятиями и терминами:

Компьютерная программа – это набор инструкций, предназначенных для выполнения определенных задач на компьютере или другом электронном устройстве. Она состоит из последовательности команд и данных, которые позволяют компьютеру выполнять определенные операции. Программы могут быть написаны на различных языках программирования и могут иметь различные цели, такие как управление устройствами, обработка данных, предоставление информации и так далее.

Компьютерная логическая игра — это вид компьютерных игр, в которых игроку предстоит решать различные логические задачи и головоломки. В таких играх игроку часто нужно использовать свою способность к анализу, рассуждению и принятию решений для решения задач и достижения целей. Компьютерные логические игры могут быть представлены в разных форматах, включая головоломки, стратегии, головоломки и многое другое. Эти игры обычно развивают интеллектуальные способности игрока и требуют логического мышления.

Переверни фишки это название самой игры. Она представляет собой настольную стратегическую игру для одного игрока, в которой переворачиваются фишки, до того момента, как цвет фишек не будет одинаковым.

В игре «Переверни фишки» фишка представляет собой игровой элемент, который может быть размещен на игровом поле. Фишки обычно имеют две стороны разного цвета.

Игровое поле – это прямоугольное поле, на котором размещаются фишки. Игровое поле обычно имеет размер 4x4 и состоит из клеток, на которых можно размещать фишки.

Ход в игре «Переверни фишки» представляет собой действие игрока, заключающееся в размещении фишки на игровом поле. Ходы выполняются игроком до тех пор, пока все клетки на игровом поле не будут заполнены одним цветом.

Победа в игре «Переверни фишки» достигается, когда игрок сделает все фишки одинакового цвета, путём переворачивания фишек.

C# — это язык программирования, разработанный компанией Microsoft. Он является объектно-ориентированным языком, который широко используется для создания приложений на платформе .NET. C# сочетает в себе элементы языков C и C++, и предоставляет разработчикам мощные инструменты для создания разнообразных программных решений.

Visual Studio — это интегрированная среда разработки (IDE) от компании Microsoft. Она используется для создания компьютерных программ, веб-сайтов, веб-приложений, веб-сервисов и мобильных приложений. Visual Studio предоставляет разработчикам множество инструментов и функций, которые помогают им эффективно писать, отлаживать и собирать код.

Windows Forms в Visual Studio — это фреймворк пользовательского интерфейса, который используется для создания приложений для Windows. Он предоставляет удобный способ создания приложений с помощью визуального дизайнера, доступного в Visual Studio.

1. Установка

Для установки игры «Переверни фишки», пожалуйста, загрузите дистрибутива последней версии игры «Переверни фишки», доступный по адресу ссылка.

Перед установкой ознакомьтесь с системными требованиями.

В процессе установки, пожалуйста, не закрывайте окно установщика.

1. Запуск

Для запуска игры нажмите на ярлык игры в меню Пуск либо наберите в командной строке Flip the chips.

1. Пользовательский интерфейс

Этот раздел описывает основные элементы пользовательского интерфейса игры «Переверни фишки»: основных режимов работы, предназначение окон и экранов, доступные операции.

1. Главное окно программы

Главное окно игры «Переверни фишки» позволяет выполнять следующие операции:

* + Выбор цветов сторон фишек.
  + Вызов текстового сообщения с правилами игры.
  + Начало игры.
  + Сброс состояния игры.

Главное окно игры «Переверни фишки» показано на рисунке 8.

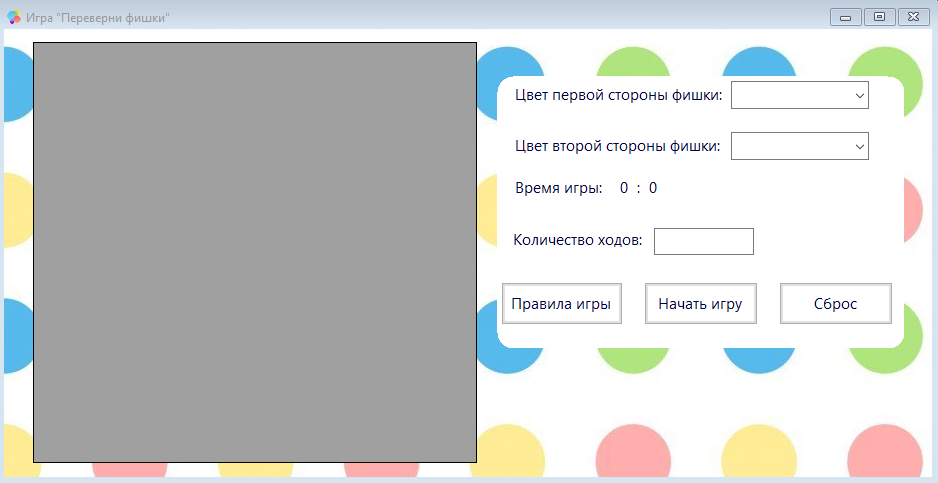


Рисунок 8 – Главное окно игры

1. Режимы работы

Пользовательский интерфейс игры «Переверни фишки» обеспечивает работу только в режиме пользователя, которому доступны все права.

1. Горячие клавиши

Следующий раздел содержит все сочетания клавиш и способы управления при помощи мыши, поддерживаемые игрой.

Общие

ЛКМ – управление игровыми элементами игры.

ALT + F4 – закрытие программы.

1. Примеры использования

В данном разделе собраны примеры реального использования игры «Переверни фишки», демонстрирующие применения продукта в различных отраслях.

1. Устранение типовых проблем

Проблема: не переворачиваются фишки при нажатии

Решение: перезапуск игры

Проблема: не начинается игра

Решение: перезапуск игры

Если вы не нашли ответа на свой вопрос, пожалуйста, свяжитесь с нами.

1. Частые вопросы (FAQ)

Как переворачивать фишки?

Ответ: фишки переворачиваются нажатием на неё правой кнопкой мыши, так же переворачиваются соседние фишки по вертикали и горизонтали.

Какая польза от игры?

Ответ: компьютерные логические игры могут быть полезны для развития логического мышления, улучшения концентрации, стимуляции мозговой активности, развлечения и улучшения навыков решения проблем.

Как запустить игру?

Ответ: для запуска игры нажмите на ярлык игры в меню Пуск либо наберите в командной строке Flip the chips.

Если вы не нашли ответа на свой вопрос, пожалуйста, свяжитесь с нами.

1. Контактная информация

Компьютерная логическая игры "Переверни фишки" разрабатывается одиночным разработчиком, который является правообладателем.

Сайт продукта - <http://github.com/fantazer887/FlipTheChips/>

Сайта у игры «Переверни фишки» нет.

Техническая поддержка

Вы можете направить вопросы по функциональности игры «Переверни фишки» следующими способами:

* + Email: n.pav2019@yandex.ru.
  + Телефон: +79068354650.

Продажи и сотрудничество

По вопросам лицензирования и сотрудничества, пожалуйста используйте следующие каналы:

* + Email: n.pav2019@yandex.ru.
  + Телефон: +79068354650.

## План внедрения и сопровождения

Текст

# Заключение

В ходе выполнения курсовой работы была проведена разработка компьютерной логической игры «Переверни фишки», которая представляет собой интересную задачу в контексте современных тенденций в развитии компьютерных игр. Разработка логической игры является актуальной и значимой задачей, учитывая растущую популярность компьютерных игр среди пользователей всех возрастов. Такие игры не только способствуют развлечению, но и могут быть использованы в образовательных целях, способствуя развитию креативного мышления, логики и аналитических способностей.

Выполнены следующие цели работы: изучены и применены принципы программирования, разработки игровых алгоритмов и создания интерактивного пользовательского интерфейса.

Для достижения поставленной цели были решены следующие основные задачи:

* изучение методов разработки компьютерных игр;
* анализ аналогичных игр на рынке;
* выбор необходимых программных приложений для реализации проекта;
* проектирование логической игры «Переверни фишки», разработка программы с использованием выбранных технологий, тестирование и отладка игры;
* анализ результатов и эффективности выполненной работы.

В процессе выполнения курсовой работы были получены ценные практические навыки в области разработки компьютерных игр. Были освоены современные методы программирования и технологии разработки игровых приложений. Разработка игры «Переверни фишки» позволила применить полученные знания на практике, что способствовало углубленному пониманию принципов разработки игровых приложений.

В результате выполнения работы была создана полноценная компьютерная логическая игра «Переверни фишки». Это программное приложение может быть использовано как инструмент для развития когнитивных способностей пользователей всех возрастов. Создание данной игры подтвердило актуальность и значимость изучения современных методов программирования и разработки игровых приложений в контексте современной компьютерной индустрии.

Таким образом, выполнение курсовой работы не только позволило успешно создать игру, но и обогатило знаниями и опытом в области разработки компьютерных игр. Полученный опыт и результаты работы могут быть использованы в дальнейших проектах, а также служить основой для дальнейшего изучения и совершенствования навыков в области программирования и разработки игровых приложений.

# Список использованных источников

1. ГОСТ 34.601-90 «Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания».
2. ГОСТ 19.701-90 «Единая система программной документации. Схемы алгоритмов, программ, данных и систем.».
3. ГОСТ 34.602-2020 «Информационные технологии. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.».
4. Работы студенческие. Общие требования и правила оформления. СТО 02069024.101 – 2015. – Оренбург : Изд-во ОГУ, 2015. – 89 с.
5. Требования к пользовательскому интерфейсу [Электронный ресурс]. – Режим доступа: <http://studfile.net/preview/16491868/page:2/> (дата обращения: 16.05.2024).
6. Преимущества использования диаграмм IDEF0 [Электронный ресурс]. – Режим доступа: <http://studfile.net/preview/7307109/page:11/> (дата обращения: 02.04.2024).
7. Описание программного обеспечения «Ramus Educational» [Электронный ресурс]. – Режим доступа: [http://studbooks.net/2270933/informatika/business\_studio](http://studbooks.net/2270933/informatika/business_studio/)

# Приложение А

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Переверни\_фишки

{

public partial class Form1 : Form

{

int moves; //переменная, хранящая количество ходов

int timeSecond = 0; //переменная, хранящая количество секунд

int timeMinute = 0; //переменная, хранящая количество минут

public Form1()

{

InitializeComponent();

}

private void rulesButton\_Click(object sender, EventArgs e) //метод вызова окна с правилами игры при нажатии на кнопку

{

MessageBox.Show("Игровое поле представляет собой таблицу размером 4х4 клеток. В клетках размещены фишки двух выбранных цветов. При щелчке мышкой по какой - нибудь фишки, её цвет и цвет соседних по горизонтали и соседних по вертикали фишек изменяется на противоположный. После щелчка в центре игрового поля цвет изменят пять фишек, на краю поля — четыре, а в углу — только три фишки. Цель игры состоит в том, чтобы сделать все фишки одного и того же цвета.", "Правила игры", MessageBoxButtons.OK);

}

private void start\_game\_button\_Click(object sender, EventArgs e) // метод для старта игры при нажатии на кнопку

{

if (comboBox1.Text == "" || comboBox2.Text == "") { MessageBox.Show("Выберите цвета сторон фишек!", "Ошибка"); timer1.Enabled = false; }

else if (comboBox1.Text == comboBox2.Text && comboBox1.Text != "") { MessageBox.Show("Цвета сторон фишек не должны совпадать!", "Ошибка"); timer1.Enabled = false; }

else

{

generationGameTable();

if (CheckWin() && moves == 0) // проверка на совпадение всех цветов при генерации игрового поля

{

ClearDataGridView();

generationGameTable();

}

else

{

timer1.Enabled = true;

dataGridView1.ClearSelection();

comboBox1.Enabled = false;

comboBox2.Enabled = false;

start\_game\_button.Enabled = false;

}

}

}

private void reset\_Button\_Click(object sender, EventArgs e) //метод сброса состояния игры при нажатии на кнопку

{

DialogResult result = MessageBox.Show("Сбросить состояние игры?", "Сброс", MessageBoxButtons.YesNo, MessageBoxIcon.Question);

if (result == DialogResult.Yes) resetAll();

}

private void dataGridView1\_CellClick(object sender, DataGridViewCellEventArgs e) //метод для изменения цвета выбранной и соседним ей фишек и прибавления единицы к количеству ходов;

{

dataGridView1.ClearSelection();

FlipCellColor(e.RowIndex, e.ColumnIndex);

dataGridView1.ClearSelection();

moves++;

textBox1.Text = moves.ToString();

if (CheckWin())

{

timer1.Enabled = false;

MessageBox.Show("Вы выиграли!\nКоличество ходов: " + moves + "\nВремя прохождения игры: " + label1.Text, "Победа!");

resetAll();

}

}

private void generationGameTable() //метод заполнения игрового поля фишками выбранных цветов

{

Color color1, color2;

color1 = GetColor1();

color2 = GetColor2();

dataGridView1.RowCount = 4; //количество строк

dataGridView1.ColumnCount = 4; //количество столбцов

dataGridView1.ReadOnly = true;

int[,] a = new int[4, 4]; //инициализация матрицы

int i, j;

//заполнение игрового поля фишками и вывод

Random rand = new Random();

for (i = 0; i < dataGridView1.RowCount; i++)

for (j = 0; j < dataGridView1.ColumnCount; j++)

{

dataGridView1.Columns[i].Width = 100;

dataGridView1.Rows[i].Height = 100;

a[i, j] = rand.Next(0, 2);

if (a[i, j] == 0)

dataGridView1.Rows[i].Cells[j].Style.BackColor = color1;

else dataGridView1.Rows[i].Cells[j].Style.BackColor = color2;

dataGridView1.Rows[i].Cells[j].Value = "";

}

}

private void FlipCellColor(int row, int col) //метод проверки положения выбранной фишки и изменения её цвета;

{

Color color1, color2;

color1 = GetColor1();

color2 = GetColor2();

if (dataGridView1.Rows[row].Cells[col].Style.BackColor == color1)

dataGridView1.Rows[row].Cells[col].Style.BackColor = color2;

else dataGridView1.Rows[row].Cells[col].Style.BackColor = color1;

// Проверка и переключение цветов соседних ячеек

if (row > 0) // Проверка верхней ячейки

if (dataGridView1.Rows[row - 1].Cells[col].Style.BackColor == color1)

dataGridView1.Rows[row - 1].Cells[col].Style.BackColor = color2;

else dataGridView1.Rows[row - 1].Cells[col].Style.BackColor = color1;

if (row < 3) // Проверка нижней ячейки

if (dataGridView1.Rows[row + 1].Cells[col].Style.BackColor == color1)

dataGridView1.Rows[row + 1].Cells[col].Style.BackColor = color2;

else dataGridView1.Rows[row + 1].Cells[col].Style.BackColor = color1;

if (col > 0) // Проверка левой ячейки

if (dataGridView1.Rows[row].Cells[col - 1].Style.BackColor == color1)

dataGridView1.Rows[row].Cells[col - 1].Style.BackColor = color2;

else dataGridView1.Rows[row].Cells[col - 1].Style.BackColor = color1;

if (col < 3) // Проверка правой ячейки

if (dataGridView1.Rows[row].Cells[col + 1].Style.BackColor == color1)

dataGridView1.Rows[row].Cells[col + 1].Style.BackColor = color2;

else dataGridView1.Rows[row].Cells[col + 1].Style.BackColor = color1;

}

private bool CheckWin() //метод проверки цветов фишек на игровом поле

{

bool allColor1 = true;

bool allColor2 = true;

Color color1, color2;

color1 = GetColor1();

color2 = GetColor2();

for (int i = 0; i < dataGridView1.RowCount; i++)

for (int j = 0; j < dataGridView1.ColumnCount; j++)

{

if (dataGridView1.Rows[i].Cells[j].Style.BackColor != color1) //проверка на несовпадение первого выбранного цвета и цветов всех фишек

allColor1 = false;

if (dataGridView1.Rows[i].Cells[j].Style.BackColor != color2) //проверка на несовпадение первого выбранного цвета и цветов всех фишек

allColor2 = false;

}

if (allColor1 || allColor2)

return true;

else return false;

}

private void ClearDataGridView() //метод очистки игрового поля;

{

dataGridView1.Rows.Clear(); // Очищаем строки

dataGridView1.Columns.Clear(); // Очищаем столбцы

}

private void resetAll() //метод сброса состояния игры

{

comboBox1.Text = "";

comboBox2.Text = "";

comboBox1.Enabled = true;

comboBox2.Enabled = true;

timer1.Enabled = false;

label1.Text = "0 : 0";

textBox1.Text = "";

timeSecond = 0;

timeMinute = 0;

ClearDataGridView();

moves = 0;

start\_game\_button.Enabled = true;

}

private void dataGridView1\_CellDoubleClick(object sender, DataGridViewCellEventArgs e)

{

dataGridView1\_CellClick(sender, e);

}

private void timer1\_Tick(object sender, EventArgs e) //метод для обработки события «Tick» таймера

{

timeSecond++;

if (timeSecond == 60) { timeSecond = 0; timeMinute++; }

label1.Text = timeMinute.ToString() + " : " + timeSecond.ToString();

}

private Color GetColor1() //метод получения цвета обратной стороны фишки

{

switch (comboBox1.SelectedItem.ToString())

{

case "Красный":

return Color.Red;

case "Оранджевый":

return Color.Orange;

case "Жёлтый":

return Color.Yellow;

case "Зелёный":

return Color.Green;

case "Синий":

return Color.Blue;

case "Фиолетовый":

return Color.Purple;

case "Чёрный":

return Color.Black;

default:

return Color.White;

}

}

private Color GetColor2() //метод получения цвета обратной стороны фишки

{

switch (comboBox2.SelectedItem.ToString())

{

case "Красный":

return Color.Red;

case "Оранджевый":

return Color.Orange;

case "Жёлтый":

return Color.Yellow;

case "Зелёный":

return Color.Green;

case "Синий":

return Color.Blue;

case "Фиолетовый":

return Color.Purple;

case "Чёрный":

return Color.Black;

default:

return Color.White;

}

}

}

}

