МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

**КУРСОВИЙ ПРОЕКТ**

(ПОЯСНЮВАЛЬНА ЗАПИСКА)

з дисципліни: «Об’єктно-орієнтоване програмування»

на тему:

**«Розробка гри типу "Книга гра"»**

студента І курсу групи ІПЗ-20-4

спеціальності 121 «Інженерія програмного забезпечення»

Савченко Данило Павлович

(прізвище, ім’я та по-батькові)

Керівник: Вакалюк Т.А.

,

Дата захисту: " \_\_\_ " \_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_ р.

Національна шкала \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ .

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_

(підпис) (прізвище та ініціали)

Житомир – 2022

ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Факультет інформаційно-комп’ютерних технологій

Кафедра інженерії програмного забезпечення

Освітній рівень: бакалавр

Спеціальність 121 «Інженерія програмного забезпечення»

«ЗАТВЕРДЖУЮ»

В.о.зав. кафедри

\_\_\_\_\_\_\_\_\_\_\_\_ А.В.Морозов

“\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_р.

ЗАВДАННЯ

НА КУРСОВИЙ ПРОЕКТ СТУДЕНТУ

Савченку Данилу Павловичу

1. Тема роботи: Розробка гри типу "Книга-гра",

керівник роботи: Вакалюк Тетяна Анатоліївна.

1. Строк подання студентом: “ ” 2020р.
2. Вихідні дані до роботи: Розробити ігрову програму "Книга-гра".
3. Зміст розрахунково-пояснювальної записки(перелік питань. Які підлягають розробці)
   * + 1. Постановка завдання
       2. Аналіз аналогічних розробок
       3. Алгоритми роботи програми

4. Опис роботи програми

5. Програмне дослідження

1. Перелік графічного матеріалу(з точним зазначенням обов’язкових креслень)

1. Презентація до КП

2. Посилання на репозиторій:

Консультанти розділів проекту (роботи)

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посади консультанта | Підпис, дата | |
| завдання  видав | завдання прийняв |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

1. Дата видачі завдання “\_17\_”\_\_лютого\_\_2021 р.

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів курсової роботи | Строк виконання етапів роботи | Примітки |
| 1 | Постановка задачі | 13.05.21-  14.05.21 | Виконано |
| 2 | Пошук, огляд та аналіз аналогічних розробок | 15.05.21-  16.05.21 | Виконано |
| 3 | Формулювання технічного завдання | 17.05.21-  19.05.21 | Виконано |
| 4 | Опрацювання літературних джерел | 20.05.21-  21.05.21 | Виконано |
| 5 | Проектування структури | 22.05.21-  23.05.21 | Виконано |
| 6 | Написання програмного коду | 24.05.21-  25.05.21 | Виконано |
| 7 | Відлагодження | 26.05.21-  27.05.21 | Виконано |
| 8 | Написання пояснювальної записки | 28.05.21-  30.05.21 | Виконано |
| 9 | Захист | 08.06.21 |  |

**КАЛЕНДАРНИЙ ПЛАН**

**Студент** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Савченко Д.П.

(підпис) (прізвище та ініціали)

**Керівник роботи** \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Вакалюк Т.А.

(підпис) (прізвище та ініціали)

**РЕФЕРАТ**

Завдання курсового проекту було створення розважальної гри для усіх поколінь.

Пояснювальна записка до курсового проекту на тему «Розробка гри типу "Книга-гра"» складається з вступу, трьох розділів, висновків, списку використаної літератури та додатку.

Текстова частина викладена на 29 сторінках друкованого тексту.

Пояснювальна записка має 28 сторінки додатків. Список використаних джерел містить 5 найменувань і займає 1 сторінку. В роботі наведено 19 рисунків. Загальний обсяг роботи – 57 сторінок.

У першому розділі обґрунтовано створення гри типу книга-гра.

У другому розділі проведено проектування та розробка програмного продукту.

У третьому розділі проведений опис та тестування додатку.

Висновок містить у собі результати виконання роботи створення програми.

У додатку представлений лістинг розробленого програмного продукту.

Ключові слова: КНИГА-ГРА, РОЗВАЖАЛЬНА ГРА, ООП, МАЛЮВАННЯ.

ЗМІСТ

[ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 6](#_Toc106709174)

[**ВСТУП** 7](#_Toc106709175)

[**РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ** 9](#_Toc106709176)

[1.1 Аналіз задачі, засобів та методів її вирішення 9](#_Toc106709177)

[1.2 Аналіз існуючого програмного забезпечення за тематикою курсової роботи. 11](#_Toc106709178)

[Висновки до першого розділу 13](#_Toc106709179)

[**РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ** 14](#_Toc106709180)

[2.1 Проектування загального алгоритму роботи програми 14](#_Toc106709181)

[2.2 Розробка функціональних алгоритмів роботи програми 16](#_Toc106709182)

[2.3 Розробка програмного забезпечення 18](#_Toc106709183)

[Висновки до другого розділу 20](#_Toc106709184)

[**РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ** 21](#_Toc106709185)

[3.1 Опис роботи з програмним додатком 21](#_Toc106709186)

[3.2 Тестування роботи програмного забезпечення 25](#_Toc106709187)

[Висновки до третього розділу 26](#_Toc106709188)

[ВИСНОВКИ 26](#_Toc106709189)

[**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ** 28](#_Toc106709190)

[ДОДАТКИ 29](#_Toc106709191)

# ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

VS – Visual Studio

ПЗ – Програмне забезпечення

ООП – Об’єктно-орієнтоване програмування

ОО – Об’єктно-орієнтовний

WF – Windows Forms

Gif – анімовані картинки

# **ВСТУП**

У курсовій роботі буде наведений процес створення гри типу " Книга-гра ".

Книга-гра - це інтерактивна гра, cутність гри полягає у тому, що гравець повинен натискати на екран, або ж виконувати прості дії. Натискання виконуються щоб продвигатися далі по сюжету.

Ігри такого типу набули популярність у 2013 році, після успіху гри Нескінченне літо.

Такі ігри не дадуть сумувати користувачу, тому що мають простий та інтерактивний інтерфейс, який примушує користувача відпочивати від буденності.

Ця гра існує на усіх платформах та виглядає приблизно однаково, міняється тільки дизайн, сутність гри залишається такою ж.

Чудовий приклад гри клікер є Нескінченне літо, яка розроблена американською студією Huyase, ArseniXC, SoraSora, OrikaNekoi. Спочатку гра була випущена для браузерів у 2014 році, потім для мобільних пристроїв у 2015 році. Через декілька років її випустили для консолей ( Xbox One та PlayStation 4 ). Зараз ця гра є одною з найпопулярніших книг-гри світу.

Головна ідея гри – це читання книги та вибір подальшого сюжету.

Актуальність роботи полягає у застосуванні новітніх технологій об’єктно-орієнтованого підходу у програмуванні. За допомогою ООП можна реалізувати багато нового та цікавого для розробки такої гри.

Мета роботи полягає у розробці програмного продукту – розробка додатку з використанням ООП. Для досягнення мети потрібно провести ОО аналіз, об’єктно-орієнтоване програмування та об’єктно-орієнтоване проектування.

Об’єкт дослідження курсової роботи – процес керування за допомогою комп’ютерної миші, обробки та промальовування об’єктів.

Предмет дослідження – програма для відпочинку та розваги.

# **РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ**

## Аналіз задачі, засобів та методів її вирішення

Задача полягає в тому, що потрібно:

1. Розробка головного меню гри, де гравець може вибирати почати гру, продовжити, вийти, інструкцію.
2. Реалізація основних механік сюжету.
3. Створення унікального графічного дизайну, стилістики сюжетних ліній.

В процесі створення книги-гри брався для прикладу звичайний аналог цієї гри, спочатку розроблялася найпростіша версія, потім відбувалося вдосконалення.

Було вирішено розробляти програму у Visual Studio за допомогою додатку WF, оскільки він легкий та доступний у використанні і більшість елементів будуть зроблені за допомогою PictureBox та Label.

Спочатку було створено головне меню для вибору дії, за допомогою Picture-Box. Це була проста, але водночас чудова ідея, тому що я міг змінювати колір, розмір елементів меню. У кожен елемент Picture-Box я помістив змінну, за допомогою якої у подальшій роботі програми я міг пересуватися по головному меню.

Сюжетні рівні були розроблені за допомогою PictureBox. При переході на новий рівень сюжету, PictureBox змінював властивість Image, забезпечуючи появу нової картинки з відповідним ворогом.

Для обробки інформації про натискання PictureBox була використана подія Click, за допомогою неї я зчитував кожен вибір користувача на рівні та додаткову інформацію, щоб потім використовувати цю інформацію для показу наступної вітки сюжету, підрахунку загальної кількості здоров’я головго героя.

Відображення кожного елементу на форму здійснювалося за допомогою властивості Visible.

Також була створена бібліотека класів, яка дозволяла промальовувати так звану "кров" на PictureBox. Усе це я зробив за допомогою класів Graphics, Drawing. Ці класи мали у собі все те, що мені потрібно. Також я використав клас Drawing2D для надання фігурам "крові" градієнтного забарвлення.

Для створення унікального інтерфейсу було використано Adobe PhotoShop та Paint. Звісно я використав картинки та gif, які я знайшов у відкритому доступі в інтернеті. Відповідно кнопки та усі елементи на формі мали потрібні властивості Image та BackgroundImage.

Також я використав метод static extern IntPtr ( метод із зовнішньою реалізацією ) і атрибути [DllImport(“gdi32.dll”)] та [DllImport(“user32.dll”)] для створення структури, яка відповідає за розміри заокруглення моїх PictureBox, які я використав у головному меню.

## 1.2 Аналіз існуючого програмного забезпечення за тематикою курсової роботи.

При аналізі вже існуючого ПЗ за тематикою курсової роботи було виявлено декілька проектів. Всі вони схожі один на одного і мають майже однакову реалізацію.

Приклад 1:назва – «Everlasting Summer», розробник: Huyase, посилання на джерело - [https://store.steampowered.com/app/331470/Everlasting\_Summer/](https://store.steampowered.com/app/331470/Everlasting_Summer/%20)



Рисунок 1.1 – Приклад гри «Everlasting Summer»

На рисунку 1.1 ми можемо побачити типову книгу-гру. У ній немає головного меню, але інтерфейс на вищому рівні. Звукове супроводження наявне.

Приклад 2: назва – «The Walking Dead: The Telltale Definitive Series», розробник: HyperHG,

посилання на джерело- <https://store.steampowered.com/app/1449690/The_Walking_Dead_The_Telltale_Definitive_Series/>



Рисунок 1.2 – Приклад гри «The Walking Dead: The Telltale Definitive Series»

На рисунку 1.2 зображено мобільну версію гри The Walking Dead: The Telltale Definitive Series. Гра має чудовий інтерфейс, музичний супровід та добре продумані ігрові механіки.

Приклад 3: назва – «King Arthur: Knight's Tale», розробник: NeocoreGames,

Посилання на джерело <https://store.steampowered.com/app/1157390/King_Arthur_Knights_Tale/>

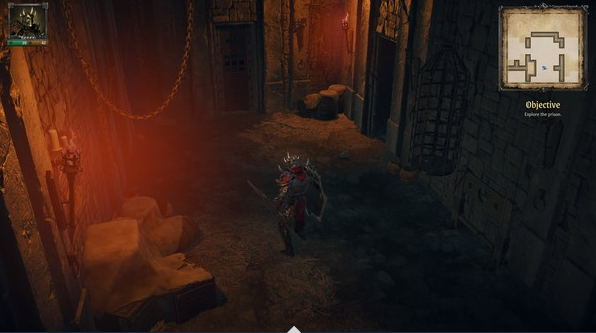


Рисунок 1.3 – Приклад гри «King Arthur: Knight's Tale»

На рисунку 1.3 зображена гра King Arthur: Knight's Tale. Гра має чудовий інтерфейс, незвичайний дизайн, музичний супровід, та унікальне меню для покращення головного героя.

Загальна статистика зображена на таблиці 1.1. Як бачимо різні аналоги мають свої плюси та мінуси.

Таблиця 1.1

**Загальна характеристика аналогів**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Стиль | Музика | Меню | 2D |
| Приклад 1 | **-** | **+** | **-** | **+** |
| Приклад 2 | **+** | **+** | **+** | **+** |
| Приклад 3 | **+** | **+** | **+** | **+** |

## Висновки до першого розділу

У ході виконання першого розділу було визначено необхідні інструменти для реалізації моєї гри та вирішено питання щодо промальовування сюжетних віток та стилізації програми.

Було проаналізовано аналоги Книги-гри та вирішено питання щодо функціоналу додатку.

# **РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

## 2.1 Проектування загального алгоритму роботи програми

На даному етапі розробки розглянемо загальний алгоритм роботи гри «Книги-гри».

Загальна схема роботи гри приведена на рисунку 2.1.

При запуску гри користувач бачить головне меню у якому він має можливість вибрати почати гру, продовжити гру, вийти або інструкцію. У центрі форми знаходиться 4 кнопоки. Знизу по центрі форми знаходиться кнопка з коротким описом сутності гри.

Користувачу доступний тільки почати гру, де починається сюжетна вітка. Продовжити гру – недоступно, також можна вийти з гри, або переглянути додаткову інформацію про гру.

Під час читання книги користувач може вийти у головне меню та потім повернутися назад, за допомогою кнопки продовжити гру. Або почати гру спочатку за допомогою кнопки почати нову гру.

Під час гри користувач повинен рухатися по сюжету, читаючи книгу, вибараючи певну дію або відповідь.

Гра має 8 віток сюжету, та 4 кінцівки, з них 2 позитивних та 2 негативних. Кінцівки гри залежать від виборів користувача.

При негативних виборах гравець втрачає одиниці здоров’я, при повній втраті здоров’я гра завершується і потрібно буде почати спочатку.

Загальна схема роботи програми

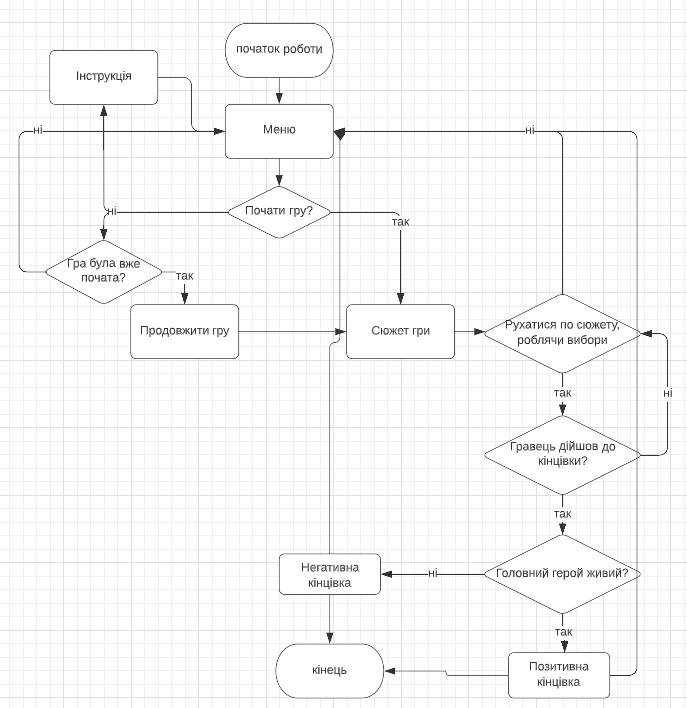


Рисунок 2.1 – Загальна схема роботи програми

## 2.2 Розробка функціональних алгоритмів роботи програми

Тепер можна детально розглянути алгоритми роботи основних методів програми.

Для малювання на PictureBox було задіяно елемент класу Graphics.Поле для малювання – це наш головний герой. Ініціалізується масив кольорів за допомогою абстрактного класу ColorOfBlood потім у кожен елемент нашого масиву заноситься фігура, яка була промальована за допомогою відповідного класу ( Blood, BuidlBlood, KnifeDamage). Далі за допомогою об’єкта Draw бібліотеки Drawing відбувається промальовування відповідної крові на нашому PictureBox( Ворогу ). За допомогою циклу for я можу відтворювати промальовування крові безліч разів.

Також я використовую метод Rand для псевдорандомного вибору кількості, та виду крові. Всередині методу розміщується цикл, який виконується відповідну кількість разів, далі за допомогою switch випадково вибирається 1 із декількох видів крові.

Також я використав метод із зовнішньою реалізацією для задання PictureBox круглої форми. Використав бібліотеку System.Runtime. InteropServices та функцію SetWindowRgn для встановлення області вікна. Використав структуру IntPtr circleEnemy для встановлення заокругленої форми PictureBox, які використовуються для кнопок у головному меню.

Діаграма класів та компонентів :

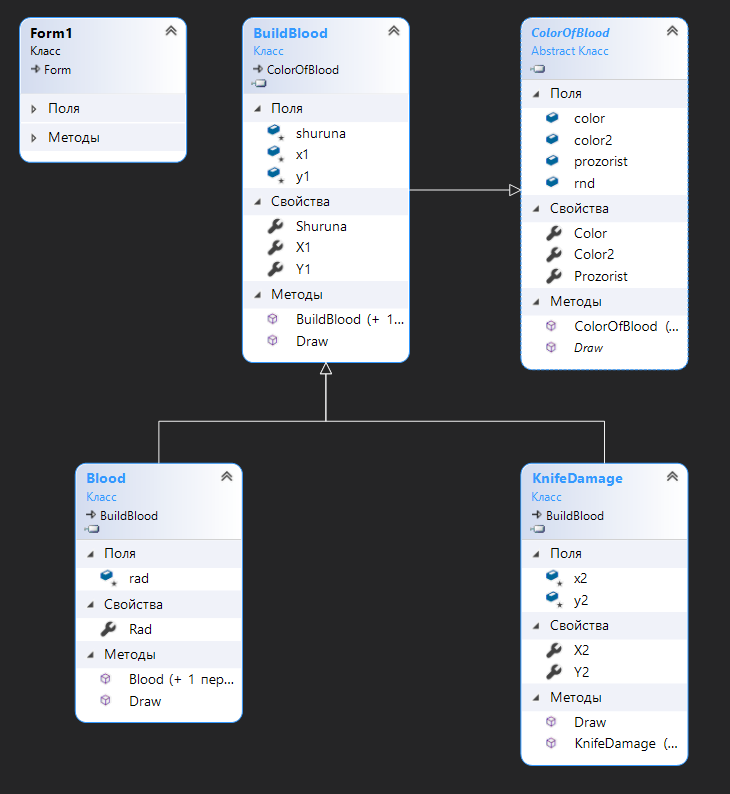


Рисунок 2.2 – Діаграма класів та компонентів

## 2.3 Розробка програмного забезпечення

Для розробки інтерфейсу додатку було використано Visual Studio 2022 року та створено додаток WF, адже він дозволяє реалізувати новітні візуальні можливості, які будуть зрозумілі користувачу програми.

Структура проекту показана на рисунку 2.3.

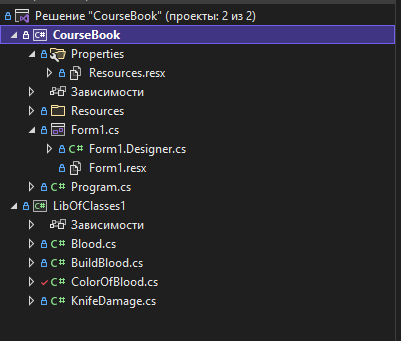


Рисунок 2.3 – Структура рішення проекту

LibOfClasses1 – це бібліотека класів, яка містить у собі класи програми. У ній містяться чотири класи: ColorOfBlood, Blood, BuildBlood, KnifeDamage.

Клас ColorOfBlood – це абстрактний базовий клас, який відповідає за замальовування фігур, які будуть будуватися у похідних класах.

Клас BuildBlood – це клас, який відповідає за малювання зелених крапок на PictureBox.

Клас Blood – це похідний клас від BuildBlood у якому відбувається промальовування червоного кругу, який найчастіше буде використовуватися при малюванні крові на ворогу.

Клас KnifeDamage – це похідний клас від BuildBlood, який відповідає за малювання відрізків, які будуть виглядати як червоні подряпини на PictureBox.

Кнопки зроблені за допомогою PictureBox, Button. Це дуже зручно, тому що можна задати будь який фон та картинку.

Для прикладу розглянемо код для розробки вітки сюжету:

public void Story2()

{

menu.Visible = true;

this.BackgroundImage = Properties.Resources.story1;

infoLabelStory.Text = "На горизонті виднілось декілька машин з нелюдами...";

infoLabelStory2.Text = "Перед Максом постав важкий вибір...";

radioButton1.Text = "Вступити в нерівний бій.";

radioButton2.Text = "Розвернутись та втікти.";

heroView.Visible = true;

helpInfoNext.Visible = true;

buttonNextStory.Visible = true;

infoLabelStory.Visible = true;

infoLabelStory2.Visible = true;

groupBox1.Visible = true;

radioButton3.Visible = false;

radioButton4.Visible = false;

if (radioButton1.Checked == true)

{

hpHero--;

storyOfGame = 2;

MessageBox.Show("Ви отримали поранення", "Попередження!!!", MessageBoxButtons.OK, MessageBoxIcon.Error);

EnemyBloodCreate();

}

else

if (radioButton2.Checked == true)

{

endSceneGame++;

storyOfGame = 3;

}

}

У цьому методі ми встановлюєм відображення для відповідних елементів форми. Далі встановлюємо фон для відповідної частини сюжету. Після цього у лейбли заносим відповідний текст для певної вітки сюжету. У radioButtons ми заносимо вибори та дії, які користувачу потрібно зроби на певному етапу читання книги, від цих виборів буде залежати подальний сюжет. Потім ми перевіряємо за допомогою if який radioButton вибраний і далі виконуємо певні дії, такі як: зменьшення здоров’я головного героя, встановлення нової вітки сюжету в storyOfGame, вивід повідомлення про поранення і виклик методу для промальовувавння крові EnemyBloodCreate();

Розглянемо один з методі малювання крові на PictureBox:

private void EnemyBloodCreate()

{

g = heroView.CreateGraphics();

ColorOfBlood[] arr = new ColorOfBlood[2000];

for (int i = 0; i < arr.Length; i++)

{

switch (r.Next(3))

{

case 1:

arr[0] = new LibOfClasses1.Blood();

arr[0].Draw(g);

break;

}

}

}

При викликанні даної функції ми ініціалізуємо змінну g класу Graphics для малювання на PictureBox( Головного героя ). Потім ініціалізується масив кольорів за допомогою абстрактного класу ColorOfBlood. Далі у циклі в кожен елемент нашого масиву заноситься фігура, яка буде промальована за допомогою відповідного класу Blood. Далі за допомогою об’єкта Draw бібліотеки Drawing відбувається промальовування крові на нашому PictureBox( Головного героя). За допомогою циклу for я можу відтворювати промальовування крові безліч разів.

## Висновки до другого розділу

В ході другого розділу було спроектовано загальну схему, діаграму класів. Визначено та детально описано основні можливості гри. Пояснено алгоритми найважливіших методів та приведено код їх основних частин. Показано структуру проекта.

# **РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ**

## 3.1 Опис роботи з програмним додатком

Після запуску програми ми бачимо головне меню рисунок 3.1, на якому розміщені головні елементи управління. Посередині знаходиться чотири кнопоки для вибору дії. Знизу по центру форми знаходиться кнопка з інструкцією, яка коротко пояснює сутність гри.



Рисунок 3.1 – Головне меню програми

Коли користувач натисне на кнопку почати гру він перейде на початок сюжету.

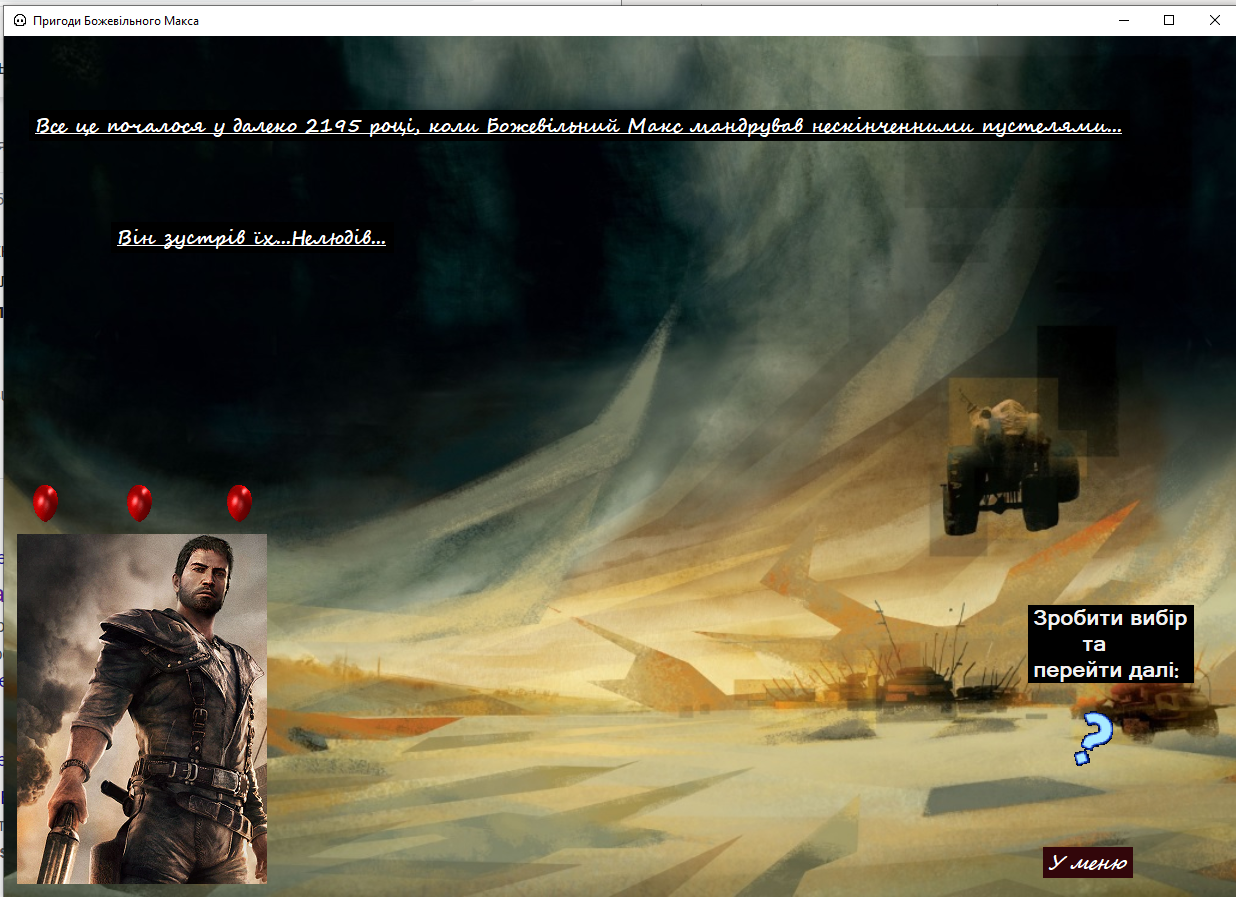


Рисунок 3.2 – Початок сюжету

Коли сюжет почався, користувач для переходу на наступну вітку сюжету може натиснути на GIF картинку.

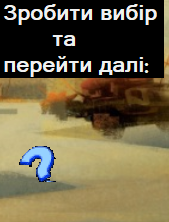


Рисунок 3.3 – Перехід на наступну вітку сюжету

В нижньому лівому кутку знаходиться головний герой.



Рисунок 3.4 – Головний герой

У верхньому лівому кутку знаходяться Labels з текстом для поточного рівня.

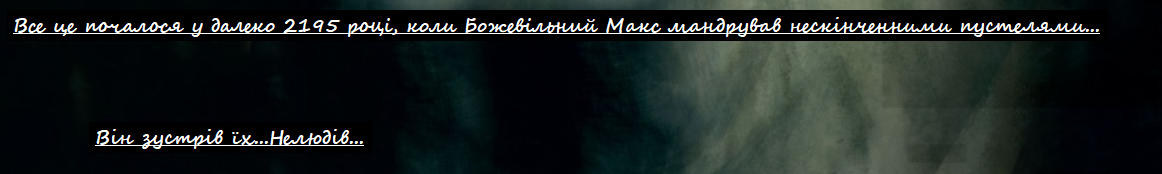


Рисунок 3.5 – Текст для поточного рівня сюжету

В нижньому лівому кутку знаходиться кнопка для повернення у головне меню.



Рисунок 3.6 – Кнопка для повернення у меню

При переході далі по сюжету з’являється поле з radioButtons, за допомогою яких вибирається вітка сюжету.

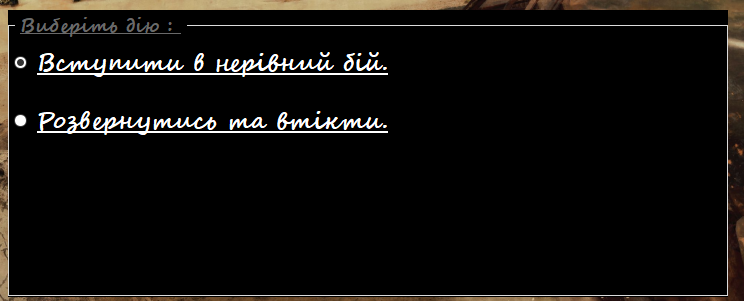


Рисунок 3.7 – Вибір вітки сюжету

При негативному вибору сюжету головний герой отримає поранення, яке відобразиться на його картинці.



Рисунок 3.8 – Поранення

Також відбувається зміна анімованих сердець, які відповідають за поточний рівень здоров’я користувача.



Рисунок 3.9 – Здоров`я користувача

При закінченні вітки сюжету з’явиться кнопка для повернення у головне меню, де можна почати гру спочатку



Рисунок 3.10 – Напис для переходу у головне меню

При проходженні гри користувач може вийти у головне меню, де він може продовжити поточну гру, або почати спочатку.



Рисунок 3.11 – Продовження гри

## 3.2 Тестування роботи програмного забезпечення

Додаток під час роботи завжди передбачає виведення попереджень або помилок при будь яких діях користувача. Наприклад, при виборі продовжити гру, коли користувач ще її не починав виведиться помилка.

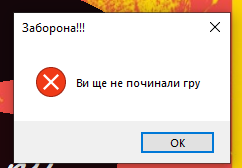


Рисунок 3.12 – Попередження про доступ до продовження

Коли користувач отримує поранення виводиться попередження.

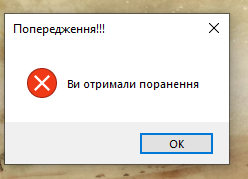


Рисунок 3.13– Попередження про поранення

## Висновки до третього розділу

Було описано роботу додатку. Роглянуто інтерфейс меню та вікна із сюжетом гри. Також було описано алгортми, які не давали змоги з'являтися помилкам. За допомогою цих методів користувачу комфортно користуватися грою.

# ВИСНОВКИ

У першому розділі курсового проекту, були поставлені завдання так медоти їх вирішення. Було проаналізовано знайдені аналоги зі схожим інтерфейсом та функціоналом. Після цього було визначено актуальні шляхи по яким потрібно рухатися у розробці програми.

У другому розділі курсового проекту, було проведено розробку основних алгоритмів гри. Наведено загальний алгоритм роботи програми у вигляді блок-схеми. Розроблено функціональний код програми, котрий виконував потрібні задачі. Після цього було проказано діаграму класів та компенентів гри. Було пояснено головні методи гри.

У третьому розділі курсового проекту, був описаний та зображений процес роботи з готовою грою, були пояснені та продемонстровані усі елементи форми. Було проведено тестування додатку, також були зазначені помилки та шляхи їх вирішення.

В результаті виконання курсового проекту отримано гру «Книга-гра», яка повністю відповідає запланованому функціоналу. Гра є достатньо зрозумілою та цікавою для користувачів будь-якого віку.

# **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Шилдт Герберт C# 4.0: полное руководство. : Пер. с англ. — М. : ООО "И.Д. Вильямс", 2011. — 1056 с.: ил.
2. Рихтер Д. CLR via C#. Программирование на платформе Microsoft.NET Framework 4.5 на языке C# / Джеффри Рихтер. – Питер, 2013. – 896 с. – (4-е изд).
3. Graphics Класс(System.Drawing) [Електронний ресурс] / admin. – 2021 – Режим доступу до ресурсу: <https://c-sharp.pro/?p=47>
4. Об'єктно-орієнтоване програмування [Електронний ресурс] / Освітній портал ДУ «Житомирська політехніка» – 2020. – Режим доступу до ресурсу: <https://learn.ztu.edu.ua/course/view.php?id=1628>
5. Створення графічних програм на C# і .NET [Електронний ресурс]/ admin. – 2015 – Режим доступу до ресурсу: <https://metanit.com/sharp/forms.php>

# ДОДАТКИ

Додаток А

Програмний код

Form1.cs:

using System;

using System.Drawing;

using System.Windows.Forms;

using System.Runtime.InteropServices;

using System.Collections.Generic;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using LibOfClasses1;

namespace CourseBook

{

public partial class Form1 : Form

{

public Graphics g;

private Random r = new Random();

public int storyOfGame = 0;

public int hpHero=3;

public string sceneGame = "";

public int endSceneGame = 0;

private void EnemyBloodCreate()

{

g = heroView.CreateGraphics();

ColorOfBlood[] arr = new ColorOfBlood[2000];

for (int i = 0; i < arr.Length; i++)

{

switch (r.Next(3))

{

case 1:

arr[0] = new LibOfClasses1.Blood();

arr[0].Draw(g);

break;

}

}

}

private void EnemyBloodCreate2()

{

g = heroView.CreateGraphics();

ColorOfBlood[] arr = new ColorOfBlood[2000];

for (int i = 0; i < arr.Length; i++)

{

switch (r.Next(4))

{

case 1:

arr[0] = new LibOfClasses1.BuildBlood();

arr[0].Draw(g);

break;

case 2:

arr[0] = new LibOfClasses1.Blood();

arr[0].Draw(g);

break;

case 3:

arr[0] = new LibOfClasses1.KnifeDamage();

arr[0].Draw(g);

break;

}

}

}

private void EnemyBloodCreate3()

{

g = heroView.CreateGraphics();

ColorOfBlood[] arr = new ColorOfBlood[500];

for (int i = 0; i < arr.Length; i++)

{

arr[0] = new LibOfClasses1.KnifeDamage();

arr[0].Draw(g);

}

}

public void Story1()

{

menu.Visible = true;

this.BackgroundImage = Properties.Resources.storyFirst;

infoLabelStory.Text = "Все це почалося у далеко 2195 році," +

" коли Божевільний Макс мандрував нескінченними пустелями...";

infoLabelStory2.Text = "Він зустрів їх..." +

"Нелюдів...";

heroView.Visible = true;

helpInfoNext.Visible = true;

buttonNextStory.Visible = true;

infoLabelStory.Visible = true;

infoLabelStory2.Visible = true;

buttonContinue.Visible = false;

buttonHelp.Visible = false;

buttonExit.Visible = false;

buttonStart.Visible = false;

startGame.Visible = false;

continueGame.Visible = false;

help.Visible = false;

exit.Visible = false;

storyOfGame=1;

}

public void Story2()

{

menu.Visible = true;

this.BackgroundImage = Properties.Resources.story1;

infoLabelStory.Text = "На горизонті виднілось декілька машин з нелюдами...";

infoLabelStory2.Text = "Перед Максом постав важкий вибір...";

radioButton1.Text = "Вступити в нерівний бій.";

radioButton2.Text = "Розвернутись та втікти.";

heroView.Visible = true;

helpInfoNext.Visible = true;

buttonNextStory.Visible = true;

infoLabelStory.Visible = true;

infoLabelStory2.Visible = true;

groupBox1.Visible = true;

radioButton3.Visible = false;

radioButton4.Visible = false;

if (radioButton1.Checked == true)

{

hpHero--;

storyOfGame = 2;

MessageBox.Show("Ви отримали поранення", "Попередження!!!", MessageBoxButtons.OK, MessageBoxIcon.Error);

EnemyBloodCreate();

}

else

if (radioButton2.Checked == true)

{

endSceneGame++;

storyOfGame = 3;

}

}

public void Story2\_1()

{

menu.Visible = true;

this.BackgroundImage = Properties.Resources.story2\_1\_1;

infoLabelStory.Text = "Починається сутичка....Що ж це буде!!!";

infoLabelStory2.Text = "БРрррр... КІяяяяя!!! Дзинь...";

heroView.Visible = true;

helpInfoNext.Visible = true;

buttonNextStory.Visible = true;

infoLabelStory.Visible = true;

infoLabelStory2.Visible = true;

groupBox1.Visible = false;

radioButton3.Visible = false;

radioButton4.Visible = false;

storyOfGame = 10;

radioButton1.Checked = false;

radioButton2.Checked = false;

backToMenu.Visible = false;

hpHero--;

MessageBox.Show("Ви отримали поранення", "Попередження!!!", MessageBoxButtons.OK, MessageBoxIcon.Error);

EnemyBloodCreate2();

}

public void Story2\_1\_1()

{

menu.Visible = true;

this.BackgroundImage = Properties.Resources.story2\_11;

infoLabelStory.Text = "Макс вступив в нерівний бій...БААХХХХХ.." +

"СІННННЬ...";

infoLabelStory2.Text = "Макс: 'отримано пораннення'. ";

radioButton1.Text = "Продовжувати бій.";

radioButton2.Text = "Спробувати уникнути бій, зважаючи поранненя.";

heroView.Visible = true;

helpInfoNext.Visible = true;

buttonNextStory.Visible = true;

infoLabelStory.Visible = true;

infoLabelStory2.Visible = true;

groupBox1.Visible = true;

backToMenu.Visible = false;

radioButton3.Visible = false;

radioButton4.Visible = false;

if (radioButton1.Checked == true)

{

storyOfGame = 11;

endSceneGame++;

}

else

if (radioButton2.Checked == true)

{

storyOfGame = 12;

hpHero--;

MessageBox.Show("Ви отримали поранення", "Попередження!!!", MessageBoxButtons.OK, MessageBoxIcon.Error);

EnemyBloodCreate3();

}

}

public void badEnd()

{

menu.Visible = true;

this.BackgroundImage = Properties.Resources.story\_badEnd;

infoLabelStory.Text = "Нажаль, Макс загинув...";

infoLabelStory2.Text = "Коли він спробував втекти, його машина була підрізана ворогами(((";

heroView.Visible = true;

helpInfoNext.Visible = false;

buttonNextStory.Visible = false;

infoLabelStory.Visible = true;

infoLabelStory2.Visible = true;

groupBox1.Visible = false;

radioButton3.Visible = false;

radioButton4.Visible = false;

menu.Visible = false;

backToMenu.Visible = true;

storyOfGame = 12;

}

public void happyEnd()

{

menu.Visible = true;

this.BackgroundImage = Properties.Resources.story\_happyEnd;

infoLabelStory.Text = "Вдача сьогодні на стороні Макса!!!";

infoLabelStory2.Text = "Він переміг усіх ворогів та безпечно вирушає додому..";

heroView.Visible = true;

helpInfoNext.Visible = false;

buttonNextStory.Visible = false;

infoLabelStory.Visible = true;

infoLabelStory2.Visible = true;

groupBox1.Visible = false;

radioButton3.Visible = false;

radioButton4.Visible = false;

menu.Visible = false;

backToMenu.Visible = true;

storyOfGame = 11;

}

public void Story2\_2()

{

menu.Visible = true;

this.BackgroundImage = Properties.Resources.story2\_2;

infoLabelStory.Text = "За головним геройом почалось прислідування";

infoLabelStory2.Text = "БРРрррр... БРрррррр.....";

heroView.Visible = true;

helpInfoNext.Visible = true;

buttonNextStory.Visible = true;

infoLabelStory.Visible = true;

infoLabelStory2.Visible = true;

groupBox1.Visible = false;

radioButton3.Visible = false;

radioButton4.Visible = false;

storyOfGame = 4;

radioButton1.Checked = false;

radioButton2.Checked = false;

}

public void Story2\_2\_2()

{

menu.Visible = true;

this.BackgroundImage = Properties.Resources.story2\_2\_2;

infoLabelStory.Text = "Незважаючи нінащо...Максу вдалося втіки...";

infoLabelStory2.Text = "Після вдалої втечі, він зустрів давню подругу...";

radioButton1.Text = "Поговорити з подругою.";

radioButton2.Text = "Поїхати по своїм справам.";

heroView.Visible = true;

helpInfoNext.Visible = true;

buttonNextStory.Visible = true;

infoLabelStory.Visible = true;

infoLabelStory2.Visible = true;

groupBox1.Visible = true;

radioButton3.Visible = false;

radioButton4.Visible = false;

if (radioButton1.Checked == true)

{

storyOfGame = 6;

hpHero--;

MessageBox.Show("Ви отримали поранення", "Попередження!!!", MessageBoxButtons.OK, MessageBoxIcon.Error);

EnemyBloodCreate2();

}

else

if (radioButton2.Checked == true)

{

storyOfGame = 7;

endSceneGame++;

}

}

public void Story7()

{

menu.Visible = true;

this.BackgroundImage = Properties.Resources.goodfinal1;

infoLabelStory.Text = "Вдача!!! Сьогодні, Максу вдалося вижити..";

infoLabelStory2.Text = "Юхуууууу.... Хуррааааааааа...!!!!!";

heroView.Visible = true;

helpInfoNext.Visible = false;

buttonNextStory.Visible = false;

infoLabelStory.Visible = true;

infoLabelStory2.Visible = true;

groupBox1.Visible = false;

radioButton3.Visible = false;

radioButton4.Visible = false;

menu.Visible = false;

backToMenu.Visible = true;

storyOfGame = 7;

}

public void Story5()

{

menu.Visible = true;

this.BackgroundImage = Properties.Resources.story5;

infoLabelStory.Text = "Всюди з`явились вороги.";

infoLabelStory2.Text = "Почалась кривава бійня.... Хто зна, чи виживе Макс...";

heroView.Visible = true;

helpInfoNext.Visible = true;

buttonNextStory.Visible = true;

infoLabelStory.Visible = true;

infoLabelStory2.Visible = true;

groupBox1.Visible = false;

radioButton3.Visible = false;

radioButton4.Visible = false;

storyOfGame = 8;

EnemyBloodCreate3();

}

public void Story8()

{

menu.Visible = true;

this.BackgroundImage = Properties.Resources.deadMax1;

infoLabelStory.Text = "Очікувано...Макс потрапив в полон.";

infoLabelStory2.Text = "Згодом, Макс помер від тяжких тортур.";

heroView.Visible = true;

helpInfoNext.Visible = false;

buttonNextStory.Visible = false;

infoLabelStory.Visible = true;

infoLabelStory2.Visible = true;

groupBox1.Visible = false;

radioButton3.Visible = false;

radioButton4.Visible = false;

hpHero = 0;

storyOfGame = 8;

EnemyBloodCreate3();

}

public void functionHpHero(int hpHero)

{

if (hpHero == 1)

{

hp1.Visible = true;

hp2.Visible = false;

hp3.Visible = false;

}

else

if (hpHero == 2)

{

hp1.Visible = true;

hp2.Visible = true;

hp3.Visible = false;

}

else

if (hpHero == 3)

{

hp1.Visible = true;

hp2.Visible = true;

hp3.Visible = true;

}

else

if(hpHero == 0)

{

hp1.Visible = false;

hp2.Visible = false;

hp3.Visible = false;

}

}

public Form1()

{

InitializeComponent();

this.DoubleBuffered = true;

}

private void buttonMusic\_Click(object sender, EventArgs e)

{

}

private void Form1\_Load(object sender, EventArgs e)

{

this.BackgroundImage = Properties.Resources.menuBack1;

heroView.Visible = false;

hp1.Visible = false;

hp2.Visible = false;

hp3.Visible = false;

groupBox1.Visible = false;

helpInfoNext.Visible = false;

buttonNextStory.Visible = false;

infoLabelStory.Visible = false;

infoLabelStory2.Visible = false;

backToMenu.Visible = false;

menu.Visible = false;

IntPtr startgame = CreateEllipticRgn(0, 0, buttonStart.Width-5, buttonStart.Height-5 );

SetWindowRgn(buttonStart.Handle, startgame, true);

IntPtr continuegame = CreateEllipticRgn(0, 0, buttonContinue.Width - 5, buttonContinue.Height - 5);

SetWindowRgn(buttonContinue.Handle, continuegame, true);

IntPtr endgame = CreateEllipticRgn(0, 0, buttonExit.Width - 5, buttonExit.Height - 5);

SetWindowRgn(buttonExit.Handle, endgame, true);

IntPtr helpmenu = CreateEllipticRgn(0, 0, buttonHelp.Width - 5, buttonHelp.Height - 5);

SetWindowRgn(buttonHelp.Handle, helpmenu, true);

}

[DllImport("gdi32.dll")]

public static extern IntPtr CreateEllipticRgn(int Left, int Top, int Right, int Bot);

[DllImport("user32.dll")]

public static extern int SetWindowRgn(IntPtr Vusota, IntPtr VusKr, bool bRed);

private void label4\_Click(object sender, EventArgs e)

{

MessageBox.Show("Суть гри полягає у тому, що Ви повинні читати книгу та обирати сюжет в залежності від вибору діаглогу та дій. За кожен негативний вибір або дію Ви будете втрачати одиницю здоров'я ( їх всього 3 ) та рухатися далі по сюжету. За кожну позитивну дію ви не будете втрачати здоров'я, а будете рухатись по сюжету. По завершенню гри буде показана позитивна або негативна кінцівка книги. Якщо здоров'я буде дорівнювати 0 - гра завершиться і потрібно буде почати все спочатку. Вдалої гри!!!", "Пояснення.", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

private void startGame\_Click(object sender, EventArgs e)

{

storyOfGame = 0;

hpHero = 3;

functionHpHero(hpHero);

Story1();

menu.Visible = true;

}

private void continueGame\_Click(object sender, EventArgs e)

{ if (storyOfGame > 0)

{

buttonContinue.Visible = false;

buttonHelp.Visible = false;

buttonExit.Visible = false;

buttonStart.Visible = false;

startGame.Visible = false;

continueGame.Visible = false;

exit.Visible = false;

help.Visible = false;

if (storyOfGame == 8)

{

backToMenu.Visible = true;

menu.Visible = false;

}

else

if (storyOfGame == 7)

{

backToMenu.Visible = true;

menu.Visible = false;

}

else

if (storyOfGame == 11)

{

backToMenu.Visible = true;

menu.Visible = false;

}

else

if (storyOfGame == 12)

{

backToMenu.Visible = true;

menu.Visible = false;

}

switch (storyOfGame)

{

case 1:

{

if (storyOfGame == 1)

Story2();

functionHpHero(hpHero);

break;

}

case 2:

{

if (storyOfGame == 2)

Story2\_1();

functionHpHero(hpHero);

break;

}

case 3:

{

if (storyOfGame == 3)

Story2\_2();

functionHpHero(hpHero);

break;

}

case 4:

{

if (storyOfGame == 4)

Story2\_2\_2();

functionHpHero(hpHero);

break;

}

case 6:

{

if (storyOfGame == 6)

Story5();

functionHpHero(hpHero);

break;

}

case 7:

{

if (storyOfGame == 7)

Story7();

functionHpHero(hpHero);

break;

}

case 8:

{

if (storyOfGame == 8)

Story8();

functionHpHero(hpHero);

break;

}

case 10:

{

if (storyOfGame == 10)

Story2\_1\_1();

functionHpHero(hpHero);

break;

}

case 11:

{

if (storyOfGame == 11)

happyEnd();

functionHpHero(hpHero);

break;

}

case 12:

{

if (storyOfGame == 12)

badEnd();

functionHpHero(hpHero);

break;

}

}

}

else

{

MessageBox.Show("Ви ще не починали гру", "Заборона!!!", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void exit\_Click(object sender, EventArgs e)

{

this.Close();

}

private void buttonNextStory\_Click(object sender, EventArgs e)

{

if (storyOfGame == 8)

{

backToMenu.Visible = true;

menu.Visible = false;

}

else

if (storyOfGame == 7)

{

backToMenu.Visible = true;

menu.Visible = false;

}

else

if (storyOfGame == 11)

{

backToMenu.Visible = true;

menu.Visible = false;

}

else

if (storyOfGame == 12)

{

backToMenu.Visible = true;

menu.Visible = false;

}

switch (storyOfGame)

{

case 1:

{

if(storyOfGame == 1)

Story2();

functionHpHero(hpHero);

break;

}

case 2:

{

if (storyOfGame == 2)

Story2\_1();

functionHpHero(hpHero);

break;

}

case 3:

{

if(storyOfGame ==3)

Story2\_2();

functionHpHero(hpHero);

break;

}

case 4:

{

if (storyOfGame == 4)

Story2\_2\_2();

functionHpHero(hpHero);

break;

}

case 6:

{

if (storyOfGame == 6)

Story5();

functionHpHero(hpHero);

break;

}

case 7:

{

if (storyOfGame == 7)

Story7();

functionHpHero(hpHero);

break;

}

case 8:

{

if (storyOfGame == 8)

Story8();

functionHpHero(hpHero);

break;

}

case 10:

{

if (storyOfGame == 10)

Story2\_1\_1();

functionHpHero(hpHero);

break;

}

case 11:

{

if (storyOfGame == 11)

happyEnd();

functionHpHero(hpHero);

break;

}

case 12:

{

if (storyOfGame == 12)

badEnd();

functionHpHero(hpHero);

break;

}

}

}

private void backToMenu\_Click(object sender, EventArgs e)

{

menu.Visible = false;

this.BackgroundImage = Properties.Resources.menuBack1;

heroView.Visible = false;

hp1.Visible = false;

hp2.Visible = false;

hp3.Visible = false;

groupBox1.Visible = false;

helpInfoNext.Visible = false;

buttonNextStory.Visible = false;

infoLabelStory.Visible = false;

infoLabelStory2.Visible = false;

backToMenu.Visible = false;

buttonContinue.Visible = true;

buttonHelp.Visible = true;

buttonExit.Visible = true;

buttonStart.Visible = true;

startGame.Visible = true;

continueGame.Visible = true;

help.Visible = true;

exit.Visible = true;

storyOfGame = 0;

hpHero = 3;

radioButton1.Checked = false;

radioButton2.Checked = false;

}

private void menu\_Click(object sender, EventArgs e)

{

this.BackgroundImage = Properties.Resources.menuBack1;

heroView.Visible = false;

hp1.Visible = false;

hp2.Visible = false;

hp3.Visible = false;

groupBox1.Visible = false;

helpInfoNext.Visible = false;

buttonNextStory.Visible = false;

infoLabelStory.Visible = false;

infoLabelStory2.Visible = false;

backToMenu.Visible = false;

buttonContinue.Visible = true;

buttonHelp.Visible = true;

buttonExit.Visible = true;

buttonStart.Visible = true;

startGame.Visible = true;

continueGame.Visible = true;

help.Visible = true;

exit.Visible = true;

radioButton1.Checked = false;

radioButton2.Checked = false;

menu.Visible = false;

}

}

}

ColorOfBlood.cs

using System;

using System.Drawing;

namespace LibOfClasses1 {

public abstract class ColorOfBlood // абстрактний клас( буде використовуватися тільки для в якості базового класу для інших класів) для створення кольору крові

{

public static Random rnd = new Random();

public static Color color;

public static Color color2;

public static int prozorist;

public Color Color

{

set;

get;

}

public Color Color2

{

set;

get;

}

public int Prozorist

{

set;

get;

}

public ColorOfBlood(Color color, Color color2, int prozorist)

{

ColorOfBlood.color2 = color2;

ColorOfBlood.color = color;

ColorOfBlood.prozorist = prozorist;

}

public ColorOfBlood(Color color, int opacity)

{

ColorOfBlood.color = color;

ColorOfBlood.prozorist = opacity;

}

public ColorOfBlood()

{

prozorist = rnd.Next(256);

color = Color.FromArgb(prozorist, 255, 36, 0);

prozorist = rnd.Next(256);

color2 = Color.FromArgb(prozorist, rnd.Next(255), rnd.Next(36), 0);

}

public ColorOfBlood(ColorOfBlood obj)

{

Color = obj.Color;

Color2 = obj.Color2;

Prozorist = obj.Prozorist;

}

public abstract void Draw(Graphics g);

}

}

Blood.cs

using System.Drawing;

using System.Drawing.Drawing2D;

namespace LibOfClasses1 {

public class Blood : BuildBlood

{

protected int rad;

public int Rad { get; set; }

public Blood() : base()

{

rad = rnd.Next(40);

}

public Blood(Color color, Color color2, int prozorist, int x1, int y1, int rad) : base(color, color2, prozorist, x1, y1)

{

this.rad = rad;

}

public override void Draw(Graphics g) // віртуальний модифікатор

{

Brush brush = new LinearGradientBrush(new System.Drawing.Rectangle(0, 0, 150,150), Color.FromArgb(prozorist, color), Color.FromArgb(prozorist, color2), LinearGradientMode.ForwardDiagonal);

g.FillEllipse(brush, x1, y1, rad, rad);

}

}

}

KnifeDamage.cs

using System.Drawing;

using System.Drawing.Drawing2D;

namespace LibOfClasses1

{

public class KnifeDamage : BuildBlood

{

protected int x2, y2;

public int X2 { get; set; }

public int Y2 { get; set; }

public KnifeDamage() : base()

{

x2 = rnd.Next(400);

y2 = rnd.Next(400);

x1 = rnd.Next(400);

y2 = rnd.Next(400);

}

public KnifeDamage(Color color, Color color2, int prozorist, int x1, int y1, int x2, int y2) : base(color, color2, prozorist, x1, y1)

{

this.x2 = x2;

this.y2 = y2;

}

public override void Draw(Graphics g) // віртуальний метод

{

Brush brush = new LinearGradientBrush(new System.Drawing.Rectangle(0, 0, 800, 800), Color.FromArgb(prozorist, color), Color.FromArgb(prozorist, color2), LinearGradientMode.ForwardDiagonal);

Pen pen = new Pen(brush, 2);

g.DrawLine(pen, x1, y1, x2, y2);

}

}

}

BuildBlood.cs

using System.Drawing;

using System.Drawing.Drawing2D;

namespace LibOfClasses1

{

public class BuildBlood: ColorOfBlood

{

protected int y1, x1, shuruna;

public int X1 { get; set; }

public int Y1 { get; set; }

public int Shuruna { get; set; }

public BuildBlood(Color color, Color color2, int prozorist, int x1, int y1) : base(color, color2, prozorist)

{

this.x1 = x1;

this.y1 = y1;

}

public BuildBlood() : base()

{

x1 = rnd.Next(400);

y1 = rnd.Next(600);

shuruna = rnd.Next(6);

}

public override void Draw(Graphics g) // модифікатор override розширяє віртуальний метод базового класу

{

Brush brush = new LinearGradientBrush(new System.Drawing.Rectangle(0, 0, 375, 375), Color.FromArgb(0,255,0), Color.FromArgb(prozorist, 255,35), LinearGradientMode.ForwardDiagonal);

g.FillEllipse(brush, x1, y1, shuruna, shuruna);

}

}

}