

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»

Кафедра інженерії програмного забезпечення

**КУРСОВИЙ ПРОЕКТ**  
(ПОЯСНЮВАЛЬНА ЗАПІСКА)

з дисципліни: «Об'єктно-орієнтоване програмування»

на тему:

**«Розробка гри “Doodle Jump”»**

студента І курсу групи ІПЗ-20-3  
спеціальності 121 «Інженерія програмного  
забезпечення»

Башманівський Максим

Олексійович

(прізвище, ім'я та по-батькові)

Керівник доктор педагогічних наук,  
професор, професор каф. ІПЗ Вакалюк Т.А.

Дата захисту: " \_10\_ " \_червня\_ 2021 р.

Національна шкала \_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

Члени комісії

_____	<u>В.Л. Левківський</u>
(підпис)	(прізвище та ініціали)
_____	<u>В.О.Власенко</u>
(підпис)	(прізвище та ініціали)
_____	<u>О.В. Чижмотря</u>
(підпис)	(прізвище та ініціали)

Житомир – 2021

Спеціальність 121 «Інженерія програмного забезпечення»

\_\_\_\_\_ А.В.Морозов  
“ ” 20\_\_ г.

1. Тема роботи: Розробка гри “Doodle jump”,  
керівник роботи: доктор педагогічних наук, професор, професор каф. ПІЗ  
Вакалюк Тетяна Анатоліївна.
2. Строк подання студентом: “\_” червня 2021р.
3. Вихідні дані до роботи: Розробити гру “Doodle jump”.
4. 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
5. Перелік графічного матеріалу(з точним зазначенням обов’язкових креслень)
  1. Презентація до КП

---

  - 2.Посилання на репозиторій:  
[https://gitlab.com/ipz203\\_bmo/kurs\\_project](https://gitlab.com/ipz203_bmo/kurs_project)
6. Консультанти розділів проекту (роботи)
7. Дата видачі завдання “ 02” березня 2021 р.

[illegible]

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів курсового проекту	Строк виконання етапів проекту	Примітки
1	Постановка задачі	02.03-10.03	виконано
2	Пошук, огляд та аналіз аналогічних розробок	10.03-14.03	виконано
3	Формулювання технічного завдання	14.03-20.03	виконано
4	Опрацювання літературних джерел	20.03-31.03	виконано
5	Проектування структури	01.04-13.04	виконано
6	Написання програмного коду	13.04-20.04	виконано
7	Відлагодження	20.04-23.04	виконано
8	Написання пояснювальної записки	23.04-06.05	виконано
9	Захист	10.06	

Студент \_\_\_\_\_  
(підпис)

Башманівський М.О.  
(прізвище та ініціали)

Керівник проекту \_\_\_\_\_  
(підпис)

Вакалюк Т.А.  
(прізвище та ініціали)

## РЕФЕРАТ

Моїм завданням на курсову роботу було створення гри “Doodle Jump”. Пояснювальна записка до курсового проекту на тему «Розробка гри “Doodle Jump”» складається з: Вступу; Основної частини (трьох розділів); Висновків; Списку використаної літератури; Додатків.

Текстову частину трактовано на 32 сторінках друкованого тексту. Пояснювальна записка записка має 13 сторінок додатків. Список використаних джерел містить 8 найменувань і займає 1 сторінку. В роботі наведено 15 рисунків. Загальний обсяг роботи – 46 сторінок.

Ключові слова: C#; ООП; “Doodle Jump”; ГРА ; ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

					ДУ «Житомирська політехніка».21.121.3.000 – ІПЗ						
Змн.	Арк.	№ докум.	Підпис	Дата							
Розроб.	Башманівський М.				Розробка гри “Doodle jump”.			Літ.	Арк.	Аркуші	
Перевір.	Вакалюк Т.А.									4	45
Керівник								ФІКТ Гр. ІПЗ-20-3[1]			
Н. контр.											
Зав. каф.											

## ЗМІСТ

ВСТУП .....	6
РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ .....	7
1.1 Аналіз задачі, засобів та методів її вирішення .....	7
1.2 Аналіз існуючого програмного забезпечення за тематикою курсowego проекту.....	9
РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....	15
2.1 Проектування загального алгоритму роботи програми .....	15
2.2 Розробка функціональних алгоритмів роботи програми.....	16
2.3 Розробка програмного забезпечення .....	19
РОЗДІЛ 3 ОПИС РОБОТИ З ПРОГРАМНИМ ДОДАТКОМ ТА ЙОГО ТЕСТУВАННЯ .....	25
3.1 Опис роботи з програмним додатком (Опис інтерфейсу).....	25
3.2 Тестування роботи програмного забезпечення .....	29
ВИСНОВОК.....	30
СПИСОК ЛІТЕРАТУРИ.....	31
ДОДАТКИ.....	32

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				5
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВСТУП

Комп'ютерні відеоігри в сучасному інтернет суспільстві є дуже популярними.

Кожен мав змогу пограти в гру, обрати серед великої кількості найулюбленішу. Деяким подобаються ігри з глибоким сюжетом, повноцінно сформованим внутрішнім світом та цікавою інтерактивною взаємодією гравця з грою. Однак більшість, цікавлять ігри, які пропонують простий та зрозумілий геймплей, звичні в управлінні та мають візуально приємний дизайн. Одною з таких є гра – “Doodle Jump”, яка вийшла для платформи IOS 6 квітня 2009 року. З моменту виходу, гра набула величезної популярності. Вже через рік було продано понад 5 мільйонів копій гри. На даний час кожного дня гру завантажують близько 25000 людей. Головна мета гри – допомогти головному герою дістатися якомога вище по платформах і набрати якомога більшу кількість очок.

Об'єктом дослідження даного курсового проекту є створення гри – аналога “Doodle Jump”, використовуючи мову програмування C#, яка дає змогу створити віконний додаток з допомогою інтерфейсу програмування додатків Windows Forms.

Предметом дослідження є застосування об'єктно – орієнтованого підходу до програмування на мові C# для створення гри: “Doodle Jump”

Мета дослідження: створити гру – аналог “Doodle Jump”, яка буде мати приємний дизайн, зручне управління та буде доступна на такій операційній системі як Windows.

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				6
Змн.	Арк.	№ докум.	Підпис	Дата		

# РОЗДІЛ 1 АНАЛІЗ ПРОБЛЕМАТИКИ, МЕТОДІВ ТА ЗАСОБІВ ВИРІШЕННЯ ЗАДАЧІ

## 1.1 Аналіз задачі, засобів та методів її вирішення

Створення гри “Doodle Jump” передбачає розробку додатку(гри) з такими складовими:

- Графічний інтерфейс
- Застосування звукових ефектів
- Різноманітність ігрового процесу
- Наявність списку рекордів(10 найвищих результатів)
- Створення елементів керування грою(пауза , переміщення героя)

Гра повинна бути легкою в управлінні, тому гравець буде використовувати не більше 3 елементів керування головним героєм. Наприклад цю роль будуть виконувати такі елементи управління як стрілки на клавіатурі. Обов’язковою умовою буде створення інтуїтивно зрозумілого інтерфейсу(меню), використання кнопок із зрозумілими написами, використання підказок в управлінні. Також гра має працювати стабільно на різних по потужності ПК на базі операційної системи Windows.

Для реалізації поставлених цілей потрібно обрати певний інтерфейс програмування додатків на мові програмування C#. На даний час на вибір доступні два таких інтерфейси – Windows Forms та Windows Presentation Foundation. При виборі я відштовхувався від знання певної графічної підсистеми та наявності певних елементів, які допоможуть мені в реалізації.

Я обрав Windows Forms так як маю досвід в розробці додатків саме на цій графічній підсистемі, також я звернув увагу на те що Windows Forms краще працюють на ПК зі слабкою потужністю. Обов’язковим елементом створення гри буде використання об’єктно – орієнтованого методу програмування на мові C#. Це означає що програма буде реалізована як множина “об’єктів”, що взаємодіють між собою, з допомогою класів – спеціальних конструкцій,

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				7
Змн.	Арк.	№ докум.	Підпис	Дата		

призначених для групування пов'язаних змінних та функцій(методів). Також будуть використані такі парадигми ООП як:

- Інкапсуляція
- Поліморфізм
- Абстрагування

Використання такого методу програмування полегшить написання гри, додаток буде оптимізованим та буде підпорядковуватися майбутнім змінам.

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				8
Змн.	Арк.	№ докум.	Підпис	Дата		



## 1.2 Аналіз існуючого програмного забезпечення за тематикою курсового проекту

Переглядаючи вже існуючі ігри типу “Doodle Jump” можливо зазначити що оригінальний розробник гри є – компанія “Lima Sky”. В даному розділі я розгляну різні варіації цієї гри саме цього розробника. Також буде розглянута гра при розробці якої оригінальний розробник лише частково приймав участь у реалізації.

Головним недоліком розглянутих ігор є відсутність кросплатформеності – властивість працювати програмного забезпечення працювати більш ніж на одній програмній чи апаратній платформах.

1) “Doodle Jump” – американської компанії – розробника ігор для мобільних платформ - “Lima Sky”.

Головна ціль гри – допомогти “Дудлеру” дістатися якомога вище по платформах і набрати як можна більше очок. Героєм можна керувати за допомогою всього трьох кнопок на мобільному пристрої (вліво, вправо, центральна кнопка). У версії для пристроїв з акселерометром управління відбувається за допомогою нахилу пристрою в потрібному напрямку.

Під час гри рахунок відображається у верхньому лівому кутку екрана, а потім зберігається в списку рекордів (у списку зберігається не більше 10 найвищих результатів). При наявності акселерометра, ви не можете змінити управління на стрілки.

Геймплей:

У “Doodle Jump”, мета полягає в керівництві чотириногим героєм, схожим на пилесос, на шляху з нескінченної серії платформ без падінь. Гравці можуть отримати короткий поштовх від різних об'єктів, таких, як пропелер-капельюхи, реактивні ранці, ракети, пружини або батуту. Є також монстри і НЛО, в яких герой повинен стріляти або стрибати на них, ліквідуючи. Стрільба

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				9
Змн.	Арк.	№ докум.	Підпис	Дата		

здійснюється натисканням на різні частини екрана. Не існує закінчення гри, але кінець кожної гри відбувається, коли гравець падає (після досягнення нижньої частини екрану), потрапляє на монстра, виявляється втягнутим в чорну діру, або викраденим НЛО.

До переваг можна віднести:

- 1) Класичний стиль, який надав грі популярність.
- 2) Захоплюючий геймплей.

Що до недоліків:

- 1) Дизайн з часом набридає
- 2) Управління з допомогою акселерометру

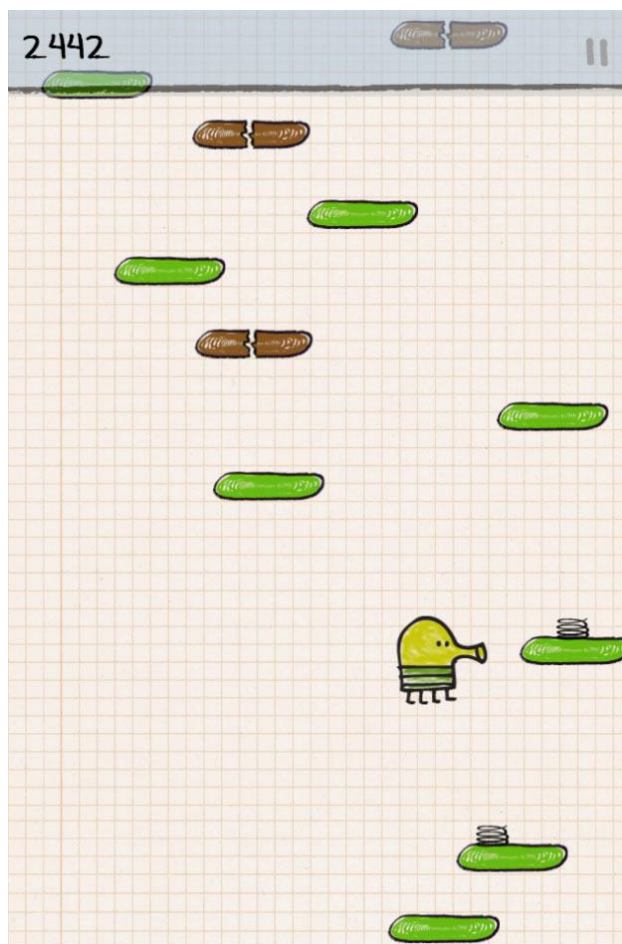


Рис.1.1 Фрагмент гри “Doodle Jump”

2) “Doodle Jump Kinect” – від розробників ігор Kinect Adventures и Smoking Gun.

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				10
Змн.	Арк.	№ докум.	Підпис	Дата		

Гра платформи Xbox 360. Гра використовує засоби управління за рухом Kinect. Основний сенс технології – управління головним героєм використовуючи власне тіло. Наприклад для переміщення головного героя вліво чи вправо потрібно відповідно перемістити тіло. Для пострілу потрібно використовувати помах руки в потрібну сторону. Візуальне проектування гри абсолютно нове – жоден з елементів оригінальної гри не був задіяний. Також були додані нові механіки гри, які допомагають по-новому боротися з ворогами та переміщуватись по платформах.



Рис.1.2 Фрагмент гри “Doodle Jump Kinect”

До переваг можна віднести:

- 1) Красива графіка, реалізовані візуальні ефекти, які сподобаються користувачу.
- 2) Захоплюючий геймплей, не схожий на оригінальну гру.

Що до недоліків:

- 1) Специфічність управління головним героєм.
- 2) Можливість грати лише за наявності додаткових апаратних платформ

		Башманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				11
Змн.	Арк.	№ докум.	Підпис	Дата		

3) “Doodle Jump 2” - американської компанії – розробника ігор для мобільних платформ - “Lima Sky”.

Продовження популярної гри “Doodle Jump” компанії – розробника ігор - “Lima Sky”. В новій частині гри зберіглися майже всі елементи першої частини. Зміни зазнала тільки механіка поведінки головного героя, а саме швидкість зміни положення головного героя. Також в другій частині додали нові локації(світи) , додали можливість зміни зовнішнього вигляду головного героя. На відміну від оригінальної частини, “Doodle Jump 2” поширюється безкоштовно, але модель монетизації змінилася - тепер в грі є реклама. Щоб її відключити, доведеться заплатити.

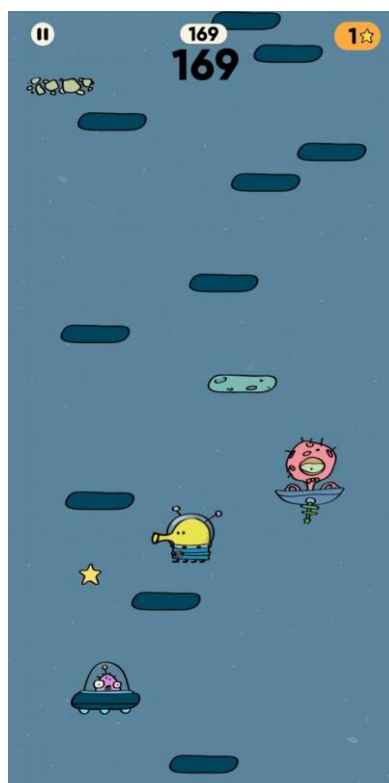


Рис.1.3 Фрагмент гри “Doodle Jump 2”

До переваг можна віднести:

- 1) Нові локації, різноманітність геймплею.
- 2) Зміна вигляду головного героя.

Що до недоліків:

- 1) Реклама, яка заважає.

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				12
Змн.	Арк.	№ докум.	Підпис	Дата		

4) “Doodle Jump: Easter Special” - американської компанії – розробника ігор для мобільних платформ - “Lima Sky”.

Спеціальна версія гри “Doodle Jump” виконана в пасхальній тематиці. Основна відмінність “Doodle Jump: Easter Special” від оригінальної це – система рівнів гри, тобто закінчивши всі рівні гравець проходить гру. Нова версія має 25 рівнів, на кожному наступному рівні збільшується складність гри. Гравець зустрічає нові перешкоди та ворогів, відмінні від оригінальної версії. Гра має особливу пасхальну тематику, яка проявляється в нових візуальних ефектах, вигляду головного героя та елементів гри.

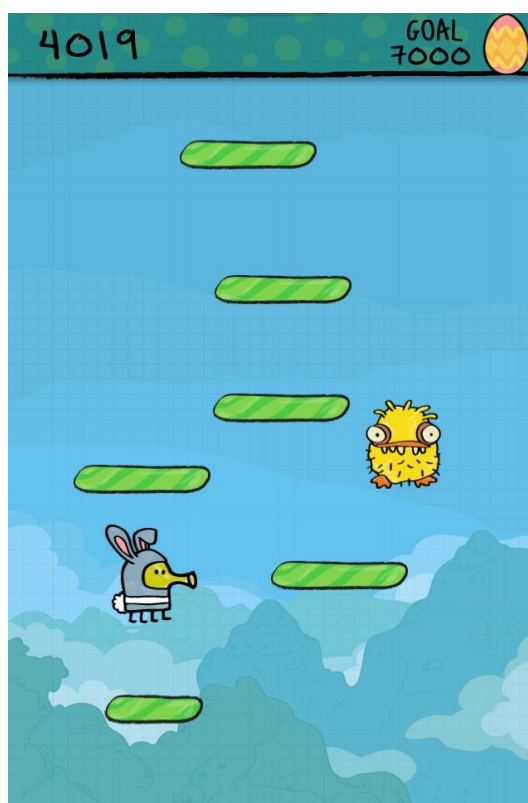


Рис.1.4 Фрагмент гри “Doodle Jump: Easter Special”

До переваг можна віднести:

- 1) Візуальний стиль, який створює атмосферу свята
- 2) Система рівнів

Що до недоліків:

- 1) Гра має кінець, при повному проходженні гри, гравець лишається контенту, який заохочує його грати.

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Розглянувши існуючі додатки гри “Doodle Jump” та їх аналоги можливо виділити те що гравцям подобається класичний стиль гри – “Doodle Jump”, різноманітність геймплею - “Doodle Jump 2”. Простота в управлінні, яка є в іграх - “Doodle Jump”, “Doodle Jump 2”, “Doodle Jump: Easter Special”.

Зважаючи на дані переваги додатків найбільш актуальнішим розвитком розробки є: класичний візуальний стиль, збереження головної ідеї гри – безкінечний рух вгору головного героя по платформах. Висновок по даному розділу можна винести використовуючи таблицю:

	<b>Doodle Jump</b>	<b>Doodle Jump 2</b>	<b>Doodle Jump: Easter Special</b>	<b>Doodle Jump Kinect</b>
Зручність інтерфейсу	+	+	+	-
Різноманітність геймплею	-	+	-	+
Зручність в управлінні	+	+	+	-
Доступність (Pay to play)	+	-	+	-

### **Висновки:**

В першому розділі, проаналізувавши задачу та методи її вирішення можливо зробити короткі висновки: додаток буде розроблятися на мові програмування C#, з використанням Windows Forms методом ООП. При аналізі існуючого програмного забезпечення був визначений шлях подальшого розвитку процесу розробки гри.

## РОЗДІЛ 2 ПРОЕКТУВАННЯ ТА РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 2.1 Проектування загального алгоритму роботи програми

Після отримання достатнього об'єму теоретичної інформації для виконання завдання, можна перейти до етапу проектування програмного забезпечення.

Загальний принцип роботи мого додатку можна зобразити використовуючи схему прецедентів.

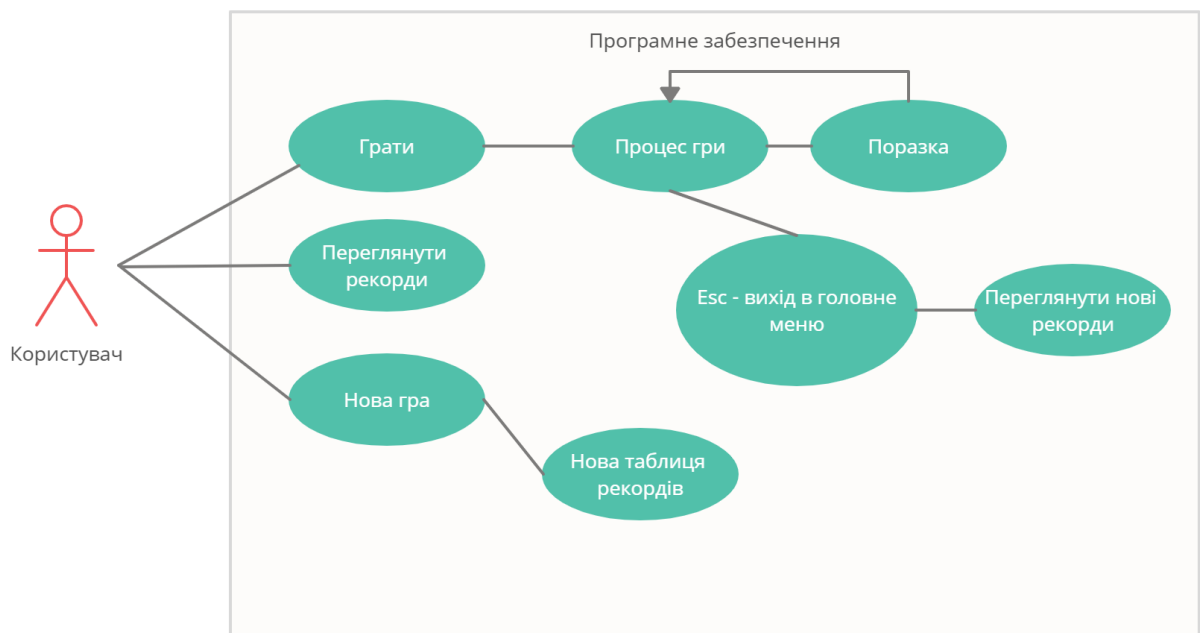


Рис.2.1 Схема прецедентів

На схемі прецедентів видно що гра буде мати звичне для ігор меню, таблицю рекордів, в яку будуть записані рекорди – найвищі результати гри. В таблиці будуть наявні лише 10 рекордів, записані вони будуть в спадаючому порядку.

Поразка враховується при падінні головного героя нижче чим 200 пікселів платформи на якій він був. При поразці враховується рахунок, записується в таблицю, якщо в ній є результати менші за набрані гравцем.



## 2.2 Розробка функціональних алгоритмів роботи програми

Розробивши певну схему роботи програми можливо створити діаграму класів, на якій будуть знаходитись класи та зв'язки між ними.

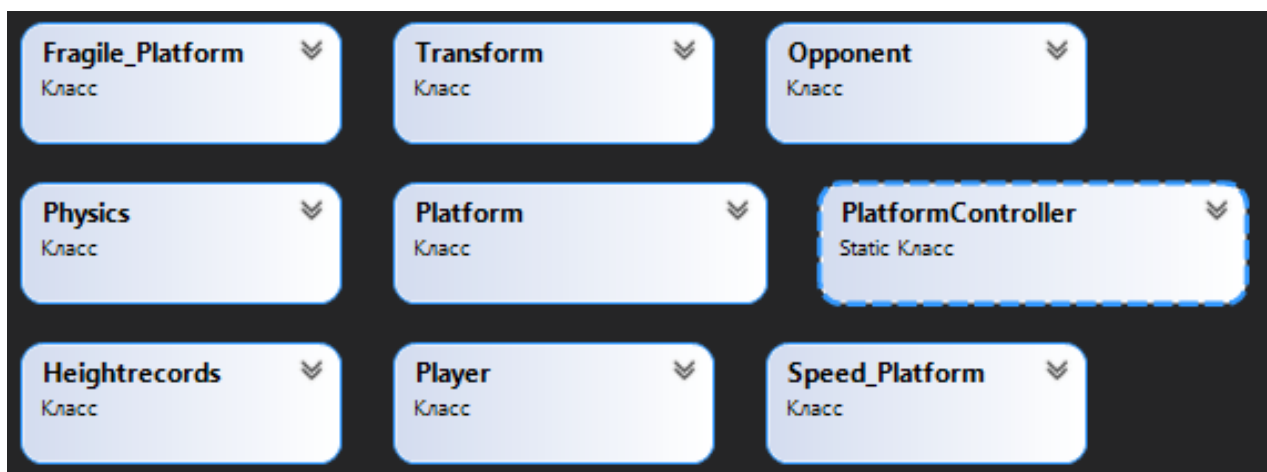


Рис.2.2 Діаграма класів(без методів)

На діаграмі класів видно що для роботи програми задіяно 8 класів, описуючи ці класи можливо зазначити деякі особливості:

- **Transform** – клас, який відповідає за розміщення та розміри будь-якого елемента гри, наявний конструктор з допомогою якого можливо задати розміри та позицію елемента

```
public class Transform
{
    public PointF position;
    public Size size;
    public Transform(PointF position, Size size)
    {
        this.position = position;
        this.size = size;
    }
}
```

- **Platform** – клас, який відповідає за звичайні платформи по яким буде переміщуватись головний герой. У класі наявний конструктор з параметрами в який передається позиція платформи, яка буде визначатись в класі PlatformController.

```
public class Platform
{
    Image sprite;
    public Transform transform;
```



```

public int sizeX;
public int sizeY;
public bool isTouchedByPlayer;

public Platform(PointF pos)
{
    sprite = Properties.Resources.platform;
    sizeX = 60;
    sizeY = 12;
    transform = new Transform(pos, new Size(sizeX, sizeY));
    isTouchedByPlayer = false;
}
public Platform()
{
    sizeX = 60;
    sizeY = 12;
}

```

- **Fragile\_Platform** – клас, який відповідає за ще один вид платформ, які при взаємодії зникають, реалізований схожий із звичайними платформами конструктор.
- **PlatformController** – клас, який відповідає за розміщення платформ на формі за певним алгоритмом.
- **Heightrecords** – клас, який відповідає за запис рекорду гравця.
- **Player** – клас, який відповідає за головного героя, його розміри, текстуру та поведінку.
- **Opponent** – клас, який відповідає за ворогів, які з'являються на ігровому полотні.
- **Speed\_Platform** – клас, який відповідає за ще один вид платформ, які переміщують головного героя набагато вище ніж звичайні.
- **Physics** – клас, який відповідає за поведінку об'єктів, їх фізику.

У процесі розробки були використані деякі текстури, які були розроблені власноруч з допомогою графічного редактору Adobe Photoshop 2021.

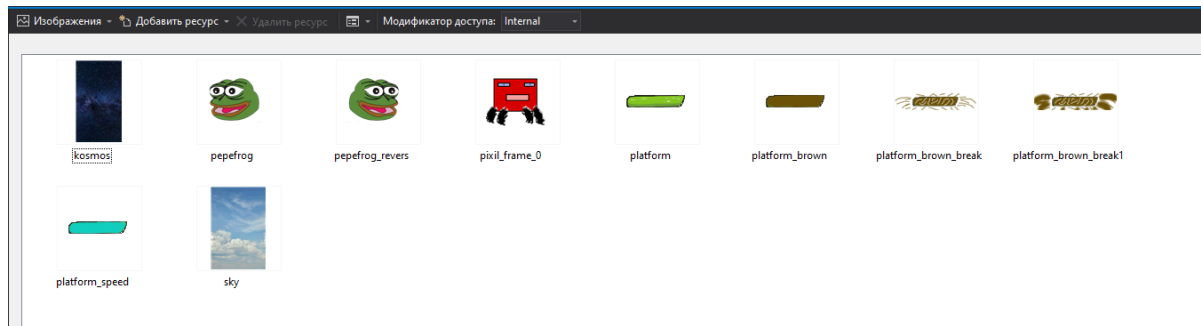


Рис.2.3 Графічні ресурси, які були задіяні в грі

Також у реалізації гри були використані такі бібліотеки класів як:

- **using System.Collections.Generic;** - використання простих колекцій, цей клас був застосований для зберігання таких елементів гри як платформи та вороги.
- **using System.Drawing;** - клас за допомогою якого були промальовані всі візуальні ефекти.
- **using System.IO;** - з допомогою цього класу відбувалось збереження власного рекорду у блокнот та використання даних з цього блокноту для виведення в подальшому на екран гравця.
- **using System.Media;** - бібліотека для програвання звукових ефектів.

## 2.3 Розробка програмного забезпечення

Структура рішення:

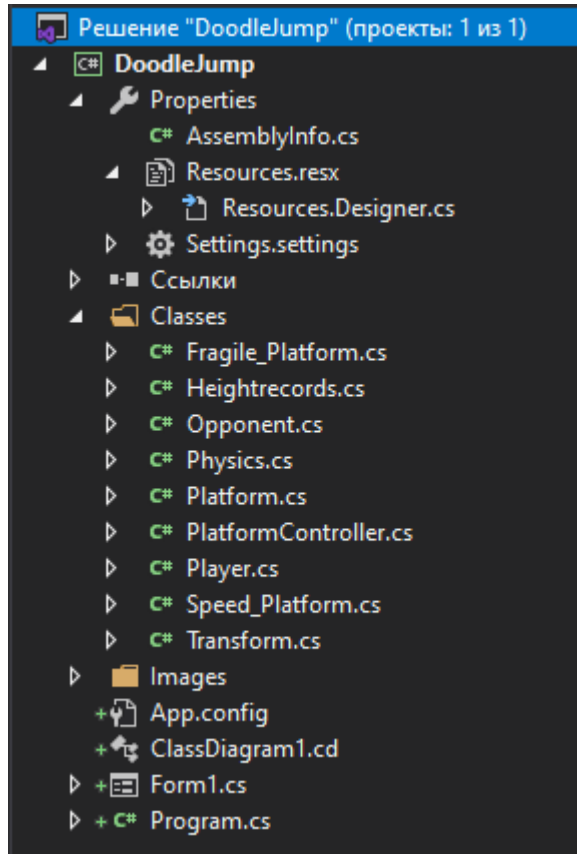


Рис.2.4 Структура рішення

Для реалізації гри використовуємо такі елементи Windows Forms як:

- **LABEL** – призначений для підказок гравцю, у моїй програмі також використовуються як кнопки.
- **LISTBOX** – розміщується на моїй формі для представлення списку рекордів.

Після повноцінної реалізації програмного коду можна представити діаграму класів:

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				19
Змн.	Арк.	№ докум.	Підпис	Дата		

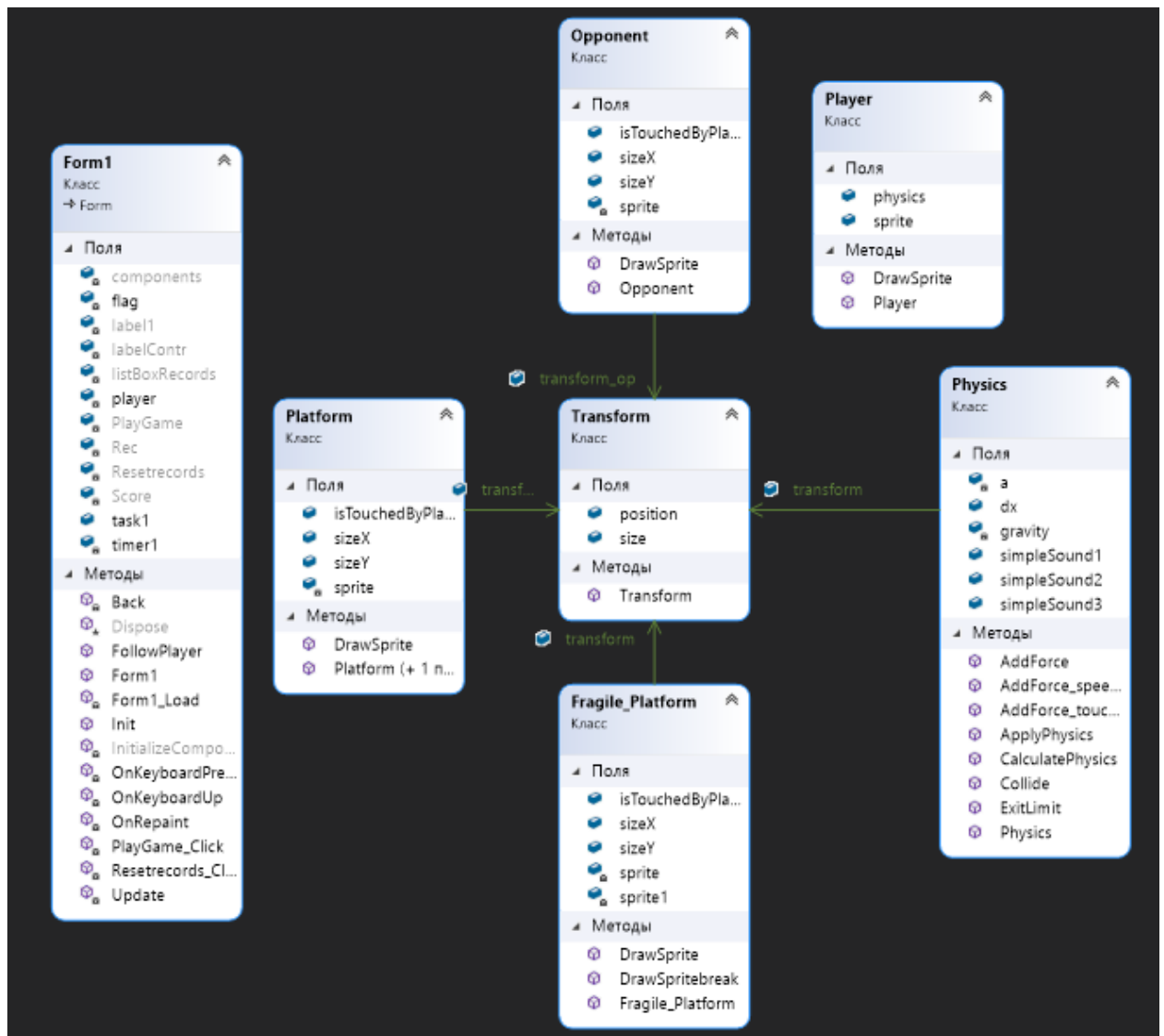


Рис.2.5 Діаграма класів

До основних реалізованих методів можна віднести:

- DrawSprite(graphics g) – метод, який відповідає за промальовування графічних елементів, у самому методі реалізований ще один метод, який малює заданий елемент в певній позиції певним способом та з визначеними розмірами.

```
public void DrawSprite(Graphics g)
{
    g.DrawImage(sprite, transform.position.X, transform.position.Y,
transform.size.Width, transform.size.Height);
}
```

- AddnewPlayer() – метод, який відповідає за таблицю рекордів нового гравця, а саме за видалення результатів.

```

public void AddnewPlayer()
{
    string path = @"C:\Users\madma\Desktop\DoodleJump-c-sharp-
master_master\DoodleJump\Records\note.txt";
    if (File.Exists(path))
    {
        File.Delete(path);
    }
    using (FileStream fs = File.Create(path)) { }
}

```

- ReadKatalog() – метод, який зчитує інформацію з блокнота, в якому знаходяться результати гри та повертає масив найбільших значень.

Основне значення цього алгоритму в читанні з блокноту значень та пробілів між ними, та запис їх.

```

StreamReader fstream = new StreamReader($"C:\Users\madma\Desktop\DoodleJump-c-sharp-
master_master\DoodleJump\Records\note.txt");
int counter = 0;
string line;
int[] task = new int[1000];

while ((line = fstream.ReadLine()) != null && counter <= 1000)
{
    string[] numbers = new string[1000];
    int numCount = 0;
    int i = 0;

    while (i < line.Length)
    {
        if (line[i] != ' ')
        {
            numbers[numCount] = "";
            while (line[i] != ' ')
            {
                numbers[numCount] += line[i];
                i++;
                if (i >= line.Length) break;
            }
            numCount++;
        }
        i++;
    }
    for (int ui = 0; ui < 1000; ui++)
    {
        task[ui] = Convert.ToInt32(numbers[ui]);
    }
    counter++;
}
fstream.Close();
Array.Sort(task);
Array.Reverse(task);
return task;

```

## Блок- схема алгоритму:

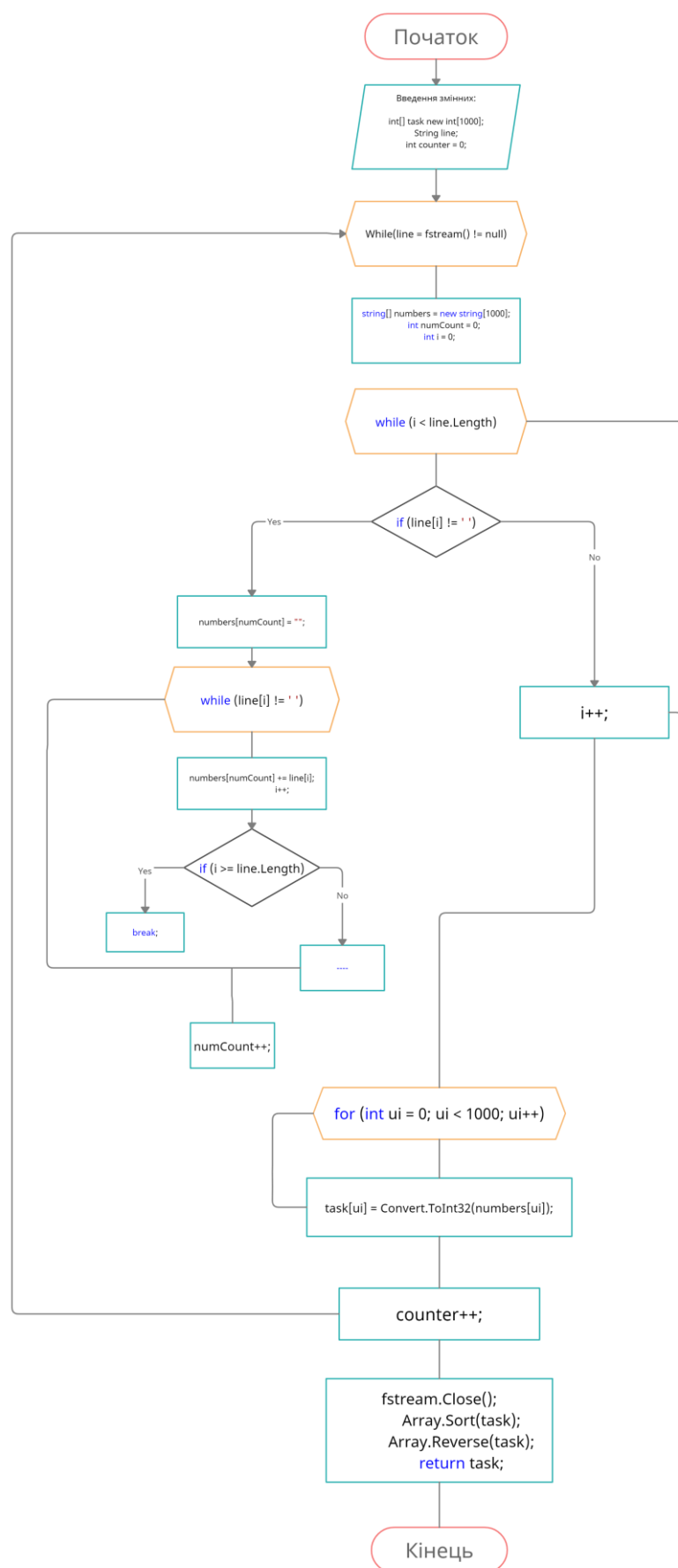


Рис.2.6 Блок – схема алгоритму ReadCatalog()

		Бащманівський М		
		Вакалюк Т.А		
Змн.	Арк.	№ докум.	Підпис	Дата

- Collide() – метод, який відповідає за взаємодію героя з іншими елементами форми. У методі ми порівнюємо координати героя та платформи по X і Y. Якщо взаємодія відбулася, рахунок збільшується. Створюється нова платформа.

```
public void Collide()
{
    for (int i = 0; i < PlatformController.platforms.Count; i++)
    {
        var platform = PlatformController.platforms[i];
        if (transform.position.X + transform.size.Width / 2 >=
platform.transform.position.X && transform.position.X + transform.size.Width / 2 <=
platform.transform.position.X + platform.transform.size.Width)
        {
            if (transform.position.Y + transform.size.Height >=
platform.transform.position.Y && transform.position.Y + transform.size.Height <=
platform.transform.position.Y + platform.transform.size.Height)
            {
                if (gravity > 0)
                {
                    AddForce();

                    simpleSound1.Play();
                    if (!platform.isTouchedByPlayer)
                    {
                        PlatformController.score += 20;
                        PlatformController.GenerateRandomPlatform();
                        platform.isTouchedByPlayer = true;
                    }
                }
            }
        }
    }
}
```

- GenerateRandomPlatform() – метод, який відповідає за визначення координат нових платформ та ворогів. Створюються випадкові координати в межах форми. Значення Y визначається динамічно, за для можливості проходження головним героєм вгору.

```
public static void GenerateRandomPlatform()
{
    Platform p1 = new Platform();
    Form1 Q = new Form1();
    ClearPlatforms();
    Random f = new Random();
    int x = f.Next(0, Q.Size.Width - p1.sizeX);

    int xB = f.Next(0, Q.Size.Width - p1.sizeX);
    int yB = f.Next(startPlatformPosY - 80, startPlatformPosY + 80);

    PointF position = new PointF(x, startPlatformPosY);

    PointF position2 = new PointF(xB, yB);
    PointF position3 = new PointF(f.Next(0, Q.Size.Width - p1.sizeX),
startPlatformPosY - 50);
}
```

```

Platform platform = new Platform(position);

Fragile_Platform platform2 = new Fragile_Platform(position2);
Speed_Platform platform3 = new Speed_Platform(position3);

platforms.Add(platform);

platforms1.Add(platform2);
platforms2.Add(platform3);
}

```

### Висновок:

У розділі було спроектовано загальний алгоритм роботи програми. Сформовані основні класи та методи програми. Реалізований графічний інтерфейс меню, також наявні звукові ефекти. Програма готова до подальшого тестування.

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				24
Змн.	Арк.	№ докум.	Підпис	Дата		



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

### 3.1 Опис роботи з програмним додатком (Опис інтерфейсу)

Після завершення програмування програми можна перейти до детального огляду.

Після запуску програми ми бачимо головне меню програми.



Рис.3.1 Головне меню гри

Користувач бачить головне меню, на якому є кнопки керування(Play , New game) програмою. Для початку гри користувач має обрати кнопку Play, розпочнеться гра, гравець знає елементи керування програмою, які він дізнався в головному меню, отже за допомогою стрілок він переміщується вгору по платформах.

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				25
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис.3.2 Геймплей гри на початку

Переміщуючись головний герой зустрічає різноманітні платформи:

- Сині – прискорюють рух героя, він летить набагато вище
- Коричневі – можливо відштовхнутись, але після цього платформа зникає, гравець не може взаємодіяти з цією формою.
- Зелені – їх можна використовувати неодноразово, переміщуючись вгору

При натисненні на стрілку вниз на клавіатурі, гра встановлює паузу, для продовження користувач має натиснути стрілку вгору.

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				26
Змн.	Арк.	№ докум.	Підпис	Дата		

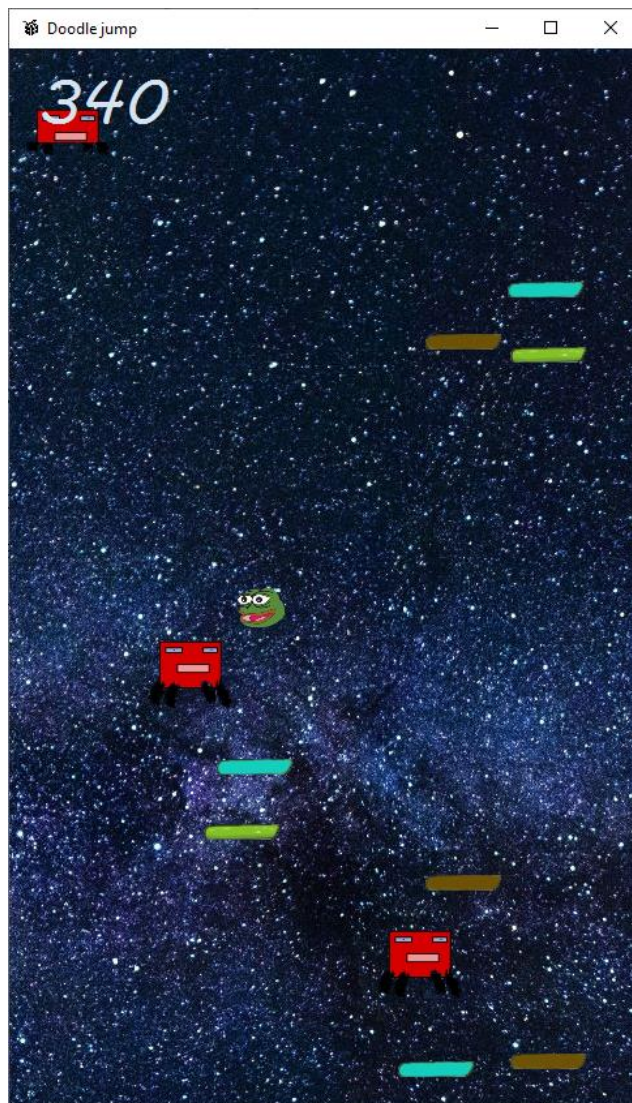


Рис.3.3 Геймплей гри коли користувач набрав більше 250 очок

При тривалій грі з'являються вороги, які заважають подальшому переміщенню героя, вони відштовхують його вниз.

Після набраних 250 очок змінюється фон гри, головний герой постає в космосі. Сам рахунок можливо спостерігати у верхньому лівому вікні.

Також реалізований метод проходження, який наявний в оригінальній грі, а саме переміщення в лівий чи правий кут спричиняє появу героя в протилежному боці екрану.

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				27
Змн.	Арк.	№ докум.	Підпис	Дата		



Рис.3.4 Головне меню після гри

Після того як гравець пограв, він може вийти в головне меню натиснувши кнопку ESC. В головному меню користувач побачить результати своєї гри у списку, результати зберігаються навіть при закритті гри.

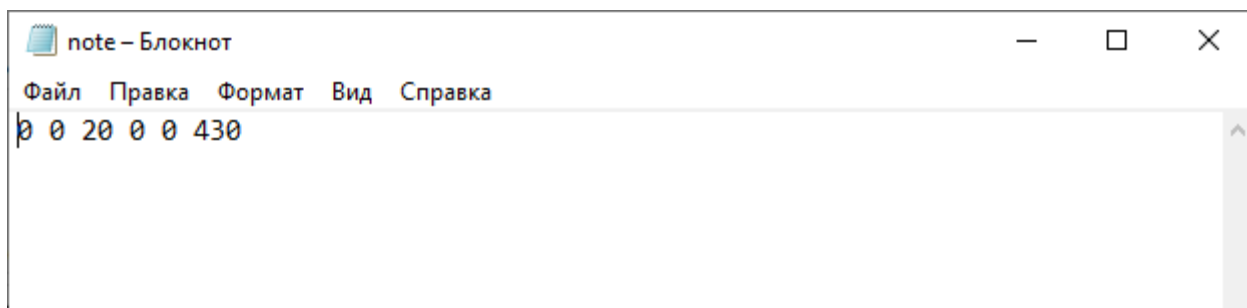


Рис.3.5 Блокнот результатів

Результати зберігаються в блокноті, при натисненні New game – всі значення видаляються.

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				28
Змн.	Арк.	№ докум.	Підпис	Дата		

### 3.2 Тестування роботи програмного забезпечення

При тестуванні програми не виявлено ні одного збою, додаток працює коректно. Користувачу надано лише деякі кнопки управління на клавіатурі, використовуючи які він не може спричинити якийсь збій. В головному меню, в процесі користування додатком, багів не виявлено.

#### **Висновки:**

В третьому розділі було проведено детальний опис роботи з програмним додатком, також створено детальний опис можливих дій користувача. Було проведено повне тестування додатку. Усі помилки, недоліки, які були знайдені в процесі тестування програмного додатку, були прибрані.

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				29
Змн.	Арк.	№ докум.	Підпис	Дата		

## ВИСНОВОК

Під час виконання даної курсової роботи я закріпив отримані в результаті навчання протягом семестру знання що до об'єктно – орієнтованого програмування. Також у процесі реалізації програмного продукту поглибив знання інтерфейсу програмування додатків – Windows Forms.

У першому розділі було здійснено аналіз задачі, визначення методів та засобів їх вирішення. Проведено аналіз існуючих аналогів програмного забезпечення за темою курсового проекту.

У другому розділі були розроблені функціональні алгоритми роботи програми, зображений загальний принцип роботи. Додаток програмно реалізований.

У третьому розділі був проведений опис програмного продукту та його тестування.

Програмний продукт повністю відповідає вимогам, заданим на початку курсового проекту.

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				30
Змн.	Арк.	№ докум.	Підпис	Дата		

## СПИСОК ЛІТЕРАТУРИ

1. 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>
2. Шилдт Герберт С# 4.0: полное руководство.: Пер. с англ. — М.: ООО "И.Д. Вильямс", 2011. — 1056 с.: ил.
3. Creately [Електронний ресурс] – Режим доступу до ресурсу: <https://app.creately.com/diagram/tFRAbdzSXy5/edit>
4. Клас List [Електронний ресурс] – Режим доступу до ресурсу: <https://metanit.com/sharp/tutorial/4.5.php>.
5. FANDOM [Електронний ресурс] - Режим доступу до ресурсу:  
[https://doodlejumpsky.fandom.com/wiki/Doodle\\_Jump\\_\(game\)](https://doodlejumpsky.fandom.com/wiki/Doodle_Jump_(game))
6. Клас File [Електронний ресурс] - Режим доступу до ресурсу: [https://professorweb.ru/my/csharp/thread\\_and\\_files/level3/3\\_1.php](https://professorweb.ru/my/csharp/thread_and_files/level3/3_1.php)
7. Клас Graphics [Електронний ресурс] - Режим доступу до ресурсу:  
<https://docs.microsoft.com/ru-ru/dotnet/api/system.drawing.graphics?view=net-5.0>
8. Клас SoundPlayer [Електронний ресурс] - Режим доступу до ресурсу:  
<https://docs.microsoft.com/ru-ru/dotnet/api/system.media.soundplayer?view=net-5.0>

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				31
Змн.	Арк.	№ докум.	Підпис	Дата		

# ДОДАТКИ

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				32
Змн.	Арк.	№ докум.	Підпис	Дата		



## Вихідний код додатку

Form.cs :

```
using DoodleJump.Classes;
using System;
using System.Drawing;
using System.Windows.Forms;
using System.Collections.Generic;

namespace DoodleJump
{
    public partial class Form1 : Form
    {
        Player player;
        Timer timer1;
        public Form1()
        {
            InitializeComponent();
            Heightrecords mass = new Heightrecords();
            Array.Copy(mass.ReadKatalog(), task1 , 10);
            for (int i = 0; i < 10; i++) listBoxRecords.Items.Add(task1[i]);
            BackgroundImage = Properties.Resources.sky;
            PlayGame.BackColor = Color.Transparent;
            label1.BackColor = Color.Transparent;
            Height = 850;
            Width = 500;
            KeyPreview = true;
        }
        public int[] task1 = new int[10];
        public void Init()
        {
            Form1 Q = new Form1();
            PlatformController.platforms = new List<Platform>();
            PlatformController.platforms1 = new List<Fragile_Platform>();
            PlatformController.platforms2 = new List<Speed_Platform>();
            PlatformController.enemy = new List<Opponent>();
            PlatformController.AddPlatform(new PointF(Q.Width/2, 400));
            PlatformController.startPlatformPosY = 400;
            PlatformController.startEnemyPosY = 300;
            Heightrecords hscore = new Heightrecords(PlatformController.score);
            PlatformController.score = 0;
            PlatformController.GenerateStartSequence();
            player = new Player();
            flag = false;
        }

        private void OnKeyboardUp(object sender, KeyEventArgs e)
        {
            player.physics.dx = 0;
        }
        private void Back(object sender, KeyEventArgs e)
        {
            if(e.KeyCode == Keys.Escape) Application.Restart();
        }
        private void OnKeyboardPressed(object sender, KeyEventArgs e)
        {
            switch (e.KeyCode.ToString())
            {
                case "Right":
                    player.physics.dx = 6;
                    break;
                case "Left":
                    player.physics.dx = -6;
            }
        }
    }
}
```

		Башманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				33
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        break;
    case "Down":
        timer1.Stop();
        break;
    case "Up":
        timer1.Start();
        break;
    }
}
private void Update(object sender, EventArgs e)
{
    Score.Text = $"{PlatformController.score}";

    if (player.physics.transform.position.Y >=
PlatformController.platforms[0].transform.position.Y + 200)
    {
        BackgroundImage = Properties.Resources.sky;
        Init();
    }
    player.physics.ApplyPhysics();
    player.physics.ExitLimit();
    FollowPlayer();
    Invalidate();
}
bool flag = false;
public void FollowPlayer()
{
    if(!flag && PlatformController.score >= 250)
    {
        BackgroundImage = Properties.Resources.kosmos;
        flag = true;
    }
    int offset = 400 - (int)player.physics.transform.position.Y;
    player.physics.transform.position.Y += offset;

    for(int i = 0; i < PlatformController.platforms.Count; i++)
    {
        var platform = PlatformController.platforms[i];
        platform.transform.position.Y += offset;
    }
    for (int i = 0; i < PlatformController.platforms1.Count; i++)
    {
        var platform = PlatformController.platforms1[i];
        platform.transform.position.Y += offset;
    }
    for (int i = 0; i < PlatformController.platforms2.Count; i++)
    {
        var platform = PlatformController.platforms2[i];
        platform.transform.position.Y += offset;
    }
    for (int i = 0; i < PlatformController.enemy.Count; i++)
    {
        var opponent = PlatformController.enemy[i];
        opponent.transform_op.position.Y += offset;
    }
}

private void OnRepaint(object sender, PaintEventArgs e)
{
    Graphics g = e.Graphics;
    if (PlatformController.platforms.Count > 0)
    {

```

		Башманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				34
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        for (int i = 0; i < PlatformController.platforms.Count; i++)
PlatformController.platforms[i].DrawSprite(g);
    }

    if (PlatformController.platforms2.Count > 0)
    {
        for (int i = 0; i < PlatformController.platforms2.Count; i++)
PlatformController.platforms2[i].DrawSprite(g);
    }

    if (PlatformController.platforms1.Count > 0)
    {
        for (int i = 0; i < PlatformController.platforms1.Count; i++)
        {
            PlatformController.platforms1[i].DrawSprite(g);
            if(PlatformController.platforms1[i].isTouchedByPlayer == true)
            {
                PlatformController.platforms1[i].DrawSpritebreak(g);
                PlatformController.platforms1.RemoveAt(i);
            }
        }
    }

    if (PlatformController.enemy.Count > 0)
    {
        for (int i = 0; i < PlatformController.enemy.Count; i++)
PlatformController.enemy[i].DrawSprite(g);
    }
    player.DrawSprite(g);
}

private void Form1_Load(object sender, EventArgs e)
{
}

private void PlayGame_Click(object sender, EventArgs e)
{
    Init();
    timer1 = new Timer();
    timer1.Interval = 15;
    timer1.Tick += new EventHandler(Update);
    timer1.Start();
    KeyDown += new KeyEventHandler(OnKeyPressed);
    KeyUp += new KeyEventHandler(OnKeyboardUp);
    KeyDown += new KeyEventHandler(Back);
    BackgroundImage = Properties.Resources.sky;
    Height = 850;
    Width = 500;
    Paint += new PaintEventHandler(OnRepaint);
    PlayGame.Visible = false;
    label1.Visible = false;
    Resetrecords.Visible = false;
    Score.Visible = true;
    listBoxRecords.Visible = false;
    Rec.Visible = false;
    labelContr.Visible = false;
    Score.BackColor = Color.Transparent;
}

private void Resetrecords_Click(object sender, EventArgs e)
{
    Heightrecords k = new Heightrecords();
    k.AddnewPlayer();
    Application.Restart();
}
}
}

```

		Башманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				35
Змн.	Арк.	№ докум.	Підпис	Дата		

## Fragile\_Platform.cs:

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DoodleJump.Classes
{
    public class Fragile_Platform
    {
        Image sprite;
        Image sprite1 = Properties.Resources.platform_brown_break1;
        public Transform transform;
        public int sizeX;
        public int sizeY;
        public bool isTouchedByPlayer;

        public Fragile_Platform(PointF pos)
        {
            sprite = Properties.Resources.platform_brown;
            sizeX = 60;
            sizeY = 12;
            transform = new Transform(pos, new Size(sizeX, sizeY));
            isTouchedByPlayer = false;
        }

        public void DrawSprite(Graphics g)
        {
            g.DrawImage(sprite, transform.position.X, transform.position.Y,
transform.size.Width, transform.size.Height);
        }
        public void DrawSpritebreak(Graphics g)
        {
            g.DrawImage(sprite1, transform.position.X, transform.position.Y, 72, 20);
        }
    }
}
```

## Heightrecords.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.IO;
using System.Security.Cryptography;
using System.IO.Compression;

namespace DoodleJump.Classes
{
    public class Heightrecords
    {
        public Heightrecords(int score)
        {
            path = @"C:\Users\madma\Desktop\DoodleJump-c-sharp-
master_master\DoodleJump\Records";
            AddtoKatalog(path , score);
        }
        public Heightrecords()
        {
        }
    }
}
```

		Башманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				36
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        path = @"C:\Users\madma\Desktop\DoodleJump-c-sharp-
master_master\DoodleJump\Records";

    }
    string path = @"C:\Users\madma\Desktop\DoodleJump-c-sharp-
master_master\DoodleJump\Records";

    public void AddtoKatalog(string path , int score)
    {
        using (StreamWriter writer = new StreamWriter($"{path}\note.txt" ,
true))
        {
            writer.Write(score + " ");
            writer.Close();
        }
    }
    public void AddnewPlayer()
    {
        string path = @"C:\Users\madma\Desktop\DoodleJump-c-sharp-
master_master\DoodleJump\Records\note.txt";
        if (File.Exists(path))
        {
            File.Delete(path);
        }
        using (FileStream fs = File.Create(path)) { }
    }

    public int[] ReadKatalog()
    {
        StreamReader fstream = new StreamReader(@"C:\Users\madma\Desktop\DoodleJump-
c-sharp-master_master\DoodleJump\Records\note.txt");
        int counter = 0;
        string line;
        int[] task = new int[1000];

        while ((line = fstream.ReadLine()) != null && counter <= 1000)
        {

            string[] numbers = new string[1000];
            int numCount = 0;
            int i = 0;

            while (i < line.Length)
            {

                if (line[i] != ' ')
                {
                    numbers[numCount] = "";
                    while (line[i] != ' ')
                    {

                        numbers[numCount] += line[i];
                        i++;
                        if (i >= line.Length) break;
                    }
                    numCount++;
                }
                i++;
            }
            for (int ui = 0; ui < 1000; ui++)
            {
                task[ui] = Convert.ToInt32(numbers[ui]);
            }
        }
    }

```

		Башманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				37
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        counter++;
    }
    fstream.Close();
    Array.Sort(task);
    Array.Reverse(task);
    return task;
}
}
}

```

## Opponent.cs:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Drawing;

namespace DoodleJump.Classes
{
    public class Opponent
    {
        Image sprite;
        public Transform transform_op;
        public int sizeX;
        public int sizeY;
        public bool isTouchedByPlayer;

        public Opponent(PointF pos)
        {
            sprite = Properties.Resources.pixil_frame_0;
            sizeX = 90;
            sizeY = 90;
            transform_op = new Transform(pos, new Size(sizeX, sizeY));
            isTouchedByPlayer = false;
        }
        public void DrawSprite(Graphics g)
        {
            g.DrawImage(sprite, transform_op.position.X, transform_op.position.Y,
transform_op.size.Width, transform_op.size.Height);
        }
    }
}

```

## Physics.cs:

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Media;

namespace DoodleJump.Classes
{
    public class Physics
    {
        public SoundPlayer simpleSound3 = new
SoundPlayer(@"C:\Users\madma\Desktop\DoodleJump-c-sharp-
master_master\DoodleJump\Sounds\blue.wav");
    }
}

```

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				38
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        public SoundPlayer simpleSound1 = new
SoundPlayer(@"C:\Users\madma\Desktop\DoodleJump-c-sharp-
master_master\DoodleJump\Sounds\green.wav");
        public SoundPlayer simpleSound2 = new
SoundPlayer(@"C:\Users\madma\Desktop\DoodleJump-c-sharp-
master_master\DoodleJump\Sounds\brown.wav");
        public Transform transform;
        float gravity;
        float a;

        public float dx;

        public Physics(PointF position, Size size)
        {
            transform = new Transform(position, size);
            gravity = 0;
            a = 0.4f;
            dx = 0;
        }

        public void ApplyPhysics()
        {
            CalculatePhysics();
        }

        public void CalculatePhysics()
        {
            if (dx != 0)
            {
                transform.position.X += dx;
            }
            if(transform.position.Y < 700)
            {
                transform.position.Y += gravity;
                gravity += a;

                Collide();
            }
        }

        public void Collide()
        {
            for (int i = 0; i < PlatformController.platforms.Count; i++)
            {
                var platform = PlatformController.platforms[i];
                if (transform.position.X + transform.size.Width / 2 >=
platform.transform.position.X && transform.position.X + transform.size.Width / 2 <=
platform.transform.position.X + platform.transform.size.Width)
                {
                    if (transform.position.Y + transform.size.Height >=
platform.transform.position.Y && transform.position.Y + transform.size.Height <=
platform.transform.position.Y + platform.transform.size.Height)
                    {
                        if (gravity > 0)
                        {
                            AddForce();

                            simpleSound1.Play();
                            if (!platform.isTouchedByPlayer)
                            {
                                PlatformController.score += 20;
                                PlatformController.GenerateRandomPlatform();
                                platform.isTouchedByPlayer = true;
                            }
                        }
                    }
                }
            }
        }
    }

```

		Башманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				39
Змн.	Арк.	№ докум.	Підпис	Дата		

```

    }
}
for (int i = 0; i < PlatformController.platforms1.Count; i++)
{
    var platform_fragile = PlatformController.platforms1[i];
    if (transform.position.X + transform.size.Width / 2 >=
platform_fragile.transform.position.X && transform.position.X + transform.size.Width / 2
<= platform_fragile.transform.position.X + platform_fragile.transform.size.Width)
    {
        if (transform.position.Y + transform.size.Height >=
platform_fragile.transform.position.Y && transform.position.Y + transform.size.Height <=
platform_fragile.transform.position.Y + platform_fragile.transform.size.Height)
        {
            if (gravity > 0)
            {
                simpleSound2.Play();
                AddForce();
                if (!platform_fragile.isTouchedByPlayer)
                {
                    PlatformController.score += 10;
                    PlatformController.GenerateRandomPlatform();
                    platform_fragile.isTouchedByPlayer = true;
                }
            }
        }
    }
}
for (int i = 0; i < PlatformController.platforms2.Count; i++)
{
    var platform_speed = PlatformController.platforms2[i];
    if (transform.position.X + transform.size.Width / 2 >=
platform_speed.transform.position.X && transform.position.X + transform.size.Width / 2 <=
platform_speed.transform.position.X + platform_speed.transform.size.Width)
    {
        if (transform.position.Y + transform.size.Height >=
platform_speed.transform.position.Y && transform.position.Y + transform.size.Height <=
platform_speed.transform.position.Y + platform_speed.transform.size.Height)
        {
            if (gravity > 0)
            {
                AddForce_speed_up();
                simpleSound3.Play();
                if (!platform_speed.isTouchedByPlayer)
                {
                    PlatformController.score += 30;
                    PlatformController.GenerateRandomPlatform();
                    platform_speed.isTouchedByPlayer = true;
                    PlatformController.GenerateRandomEnemy();
                }
            }
        }
    }
}
for (int i = 0; i < PlatformController.enemy.Count; i++)
{
    var enemy = PlatformController.enemy[i];
    if (transform.position.X + transform.size.Width/2 >=
enemy.transform_op.position.X && transform.position.X + transform.size.Width / 2 <=
enemy.transform_op.position.X + enemy.transform_op.size.Width)
    {
        if (transform.position.Y + transform.size.Height >=
enemy.transform_op.position.Y && transform.position.Y + transform.size.Height <=
enemy.transform_op.position.Y + enemy.transform_op.size.Height)

```

		Башманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				40
Змн.	Арк.	№ докум.	Підпис	Дата		



```

        {
            AddForce_touch_op();
        }
    }
}
public void ExitLimit()
{
    Form1 Q = new Form1();
    if (transform.position.X > Q.Width-30)
    {
        transform.position.X = 0;
    }
    if (transform.position.X < 0)
    {
        transform.position.X = Q.Width-30;
    }
}
public void AddForce()
{
    gravity = -10;
}

public void AddForce_speed_up()
{
    gravity = -18;
}
public void AddForce_touch_op()
{
    gravity = 10;
}
}
}

```

### Platform.cs:

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DoodleJump.Classes
{
    public class Platform
    {
        Image sprite;
        public Transform transform;
        public int sizeX;
        public int sizeY;
        public bool isTouchedByPlayer;

        public Platform(PointF pos)
        {
            sprite = Properties.Resources.platform;
            sizeX = 60;
            sizeY = 12;
            transform = new Transform(pos, new Size(sizeX, sizeY));
            isTouchedByPlayer = false;
        }
        public Platform()
        {
            sizeX = 60;

```

		Башманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				41
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        sizeY = 12;
    }
    public void DrawSprite(Graphics g)
    {
        g.DrawImage(sprite, transform.position.X, transform.position.Y,
transform.size.Width, transform.size.Height);
    }
}
}

```

## PlatformController.cs:

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DoodleJump.Classes
{
    public static class PlatformController
    {
        public static List<Platform> platforms;
        public static List<Fragile_Platform> platforms1;
        public static List<Speed_Platform> platforms2;
        public static List<Opponent> enemy;
        public static int startPlatformPosY = 400;
        public static int startEnemyPosY = 300;
        public static int score = 0;
        public static void AddPlatform(PointF position)
        {
            Platform platform = new Platform(position);
            platforms.Add(platform);
        }
        public static void GenerateRandomEnemy()
        {
            Form1 Q = new Form1();
            Random f = new Random();
            int x = f.Next(0, Q.Size.Width - 40);
            PointF position = new PointF(x, startEnemyPosY);
            Opponent opo = new Opponent(position);
            enemy.Add(opo);
        }
        public static void GenerateStartSequence()
        {
            Platform pl = new Platform();
            Form1 Q = new Form1();
            Random r = new Random();
            for(int i = 0; i < 15; i++)
            {
                int x = r.Next(0, Q.Size.Width - pl.sizeX);
                int y = r.Next(30, 40);
                int y_en = r.Next(80, 90);
                startPlatformPosY -= y;
                startEnemyPosY -= y_en;
                PointF position = new PointF(x, startPlatformPosY);
                Platform platform = new Platform(position);
                platforms.Add(platform);
            }
        }
    }
}

```

		Башманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				42
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public static void GenerateRandomPlatform()
{
    Platform pl = new Platform();
    Form1 Q = new Form1();
    ClearPlatforms();
    Random f = new Random();
    int x = f.Next(0 , Q.Size.Width - pl.sizeX);

    int xB = f.Next(0, Q.Size.Width - pl.sizeX);
    int yB = f.Next(startPlatformPosY - 80, startPlatformPosY + 80);

    PointF position = new PointF(x, startPlatformPosY);

    PointF position2 = new PointF(xB, yB);
    PointF position3 = new PointF(f.Next(0 , Q.Size.Width- pl.sizeX),
startPlatformPosY-50);

    Platform platform = new Platform(position);

    Fragile_Platform platform2 = new Fragile_Platform(position2);
    Speed_Platform platform3 = new Speed_Platform(position3);

    platforms.Add(platform);

    platforms1.Add(platform2);
    platforms2.Add(platform3);
}
public static void ClearPlatforms()
{
    for(int i = 0; i < platforms.Count; i++)
    {
        if (platforms[i].transform.position.Y >= 700) platforms.RemoveAt(i);
    }
    for (int i = 0; i < platforms1.Count; i++)
    {
        if (platforms1[i].transform.position.Y >= 700) platforms1.RemoveAt(i);
    }
    for (int i = 0; i < platforms2.Count; i++)
    {
        if (platforms2[i].transform.position.Y >= 700) platforms2.RemoveAt(i);
    }
    for (int i = 0; i < enemy.Count; i++)
    {
        if (enemy[i].transform_op.position.Y >= 700) enemy.RemoveAt(i);
    }
}
}
}
}

```

### Player.cs:

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DoodleJump.Classes
{
    public class Player
    {
        public Physics physics;
    }
}

```

		Башманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				43
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        public Image sprite;

        public Player()
        {
            Form1 Q = new Form1();
            sprite = Properties.Resources.pepefrog;
            physics = new Physics(new PointF(Q.Width/2, 350), new Size(60, 60));
        }

        public void DrawSprite(Graphics g)
        {
            g.DrawImage(sprite, physics.transform.position.X,
            physics.transform.position.Y, physics.transform.size.Width,
            physics.transform.size.Height);
        }

    }
}

```

### Speed\_Platform.cs:

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DoodleJump.Classes
{
    public class Speed_Platform
    {
        Image sprite;

        public Transform transform;
        public int sizeX;
        public int sizeY;
        public bool isTouchedByPlayer;

        public Speed_Platform(PointF pos)
        {
            sprite = Properties.Resources.platform_speed;
            sizeX = 60;
            sizeY = 12;
            transform = new Transform(pos, new Size(sizeX, sizeY));
            isTouchedByPlayer = false;
        }

        public void DrawSprite(Graphics g)
        {
            g.DrawImage(sprite, transform.position.X, transform.position.Y,
            transform.size.Width, transform.size.Height);
        }

    }
}

```

		Башманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				44
Змн.	Арк.	№ докум.	Підпис	Дата		

## Transform.cs:

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace DoodleJump.Classes
{
    public class Transform
    {
        public PointF position;
        public Size size;
        public Transform(PointF position, Size size)
        {
            this.position = position;
            this.size = size;
        }
    }
}
```

		Бащманівський М			ДУ «Житомирська політехніка».21.121.3.000 - ІПЗ	Арк.
		Вакалюк Т.А				45
Змн.	Арк.	№ докум.	Підпис	Дата		