НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«Київський політехнічний інститут імені Ігоря Сікорського»

кафедра Технічної кібернетики

**Тема: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**UA.ІК.01310-01 01**

*Курсова робота*

дисципліна

*“Структури даних та алгоритми”*

Керівник: Виконав:

доц. Богданова Н.В. \_\_\_\_\_\_\_

“Захист дозволено” гр. ІК-9х

“\_\_\_” \_\_\_\_\_\_\_\_\_\_ 2021 р.

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

/підпис керівника/ /підпис студента/

Захищено з оцінкою

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

2021

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра Технічної кібернетики

Дисципліна Структури даних та алгоритми

Курс **I**  Група Семестр **II**

ЗАВДАННЯ

**на курсову роботу студента**

(прізвище, ім’я, по батькові)

1. Тема роботи Створення комп’ютерної гри «Battle City»

2. Строк здачі студентом закінченої роботи

3. Вихідні дані до роботи мова програмування

4. Зміст розрахунково – пояснювальної записки (перелік питань, які підлягають розробці)

Розробка алгоритму програми

Архітектура програми

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

*Блок-схема алгоритму роботи програми (Формат А1)*

6. Дата видачі завдання

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  п\п | **Назва етапів курсової роботи** | **Строк виконання**  **етапів роботи** | **Примітки** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Студент

(прізвище, ім’я, по батькові студента)

Керівник Богданова Н.В.

( прізвище, ім’я, по батькові викладача)

« » 2021 р.

ЗМІСТ

[1. ВСТУП 4](#_Toc71489450)

[1. ІГРОВИЙ ПРОЦЕС 5](#_Toc71489451)

[2. ПОСТАНОВКА ЗАДАЧІ 6](#_Toc71489452)

[2. 3. АЛГОРИТМІЧНА МОДЕЛЬ ТА СТРУКТУРА ПРОГРАММИ 7](#_Toc71489453)

[3.1 UML-СХЕМА КЛАСІВ 8](#_Toc71489454)

# ВСТУП

У ході роботи над проектом є:

1. Набути навиків використання основних принципів ООП при розробці програми мовою програмування C++

# ІГРОВИЙ ПРОЦЕС

Гравець обирає кількість гравців (1-2) які будуть грати та початковий рівень гри. Танк контролюється за допомогою клавіш WASD та стріляє за допомогою пробілу (для другого гравця це клавіші стрілочки та правий ctrl). Якщо граветь знищив 20 ворожих танків то його переносить на наступний рівень. Мета набрати якнайбільше балів поки не закінчяться життя танка або не буде зруйнована його база. Після чого гра закінчується, підраховуються бали за цей рівень та виводиться таблиця рекордів. Якщо гравець набрав більше балів за 10 місце в таблиці йому пропонується ввести його ім’я.

# ПОСТАНОВКА ЗАДАЧІ

Цільовим напрямком курсової роботи є розробка комп’ютерної гри «Battle City». Гра має забезпечувати виконання таких умов і містити:

1. Головне меню з логотипом, з можливостями почати гру, переглянути
2. таблицю рекордів або вийти із гри.
3. Таблицю рекордів, де відображаються 10 найкращих результатів гравців.

Дивлячись на це, потрібно вирішити такі проблеми:

1. Намалювати потрібні для гри графічні об’єкти (спрайти);
2. Розробити алгоритм гри:
3. Рух танка по мапі з уникненням колізій як з самою мапою так і іншими танками, як фізичними об’єктоми;
4. Рух ворожих танків за алгоритмом схожим на алгоритм з оригінальної гри
5. Реалізувати зв’язок між класами;
6. Анімувати рухомі об’єкти на мапі;
7. Виконати зв’язок між сценами за допомогою графічного інтерфейсу;
8. Реалізувати збереження та завантаження таблиці рекордів

# 3. АЛГОРИТМІЧНА МОДЕЛЬ ТА СТРУКТУРА ПРОГРАММИ

Реалізація програмного коду для гри «Battle City» була здійснена за допомогою об’єктно-орієнтованої мови програмування C++ в середовищі розробки Microsoft Visual Studio.

Основним керуючим класом є Game, який ініціалізує всі ігрові об’єкти, виконує їх оновлення та вивід на екран.

Графічний інтерфейс був реалізований за допомогою графічної бібліотеки SFML, яка дає можливість керувати вікнами та виводи малювати на них. Кожен важливий клас має процедуру Draw, яка виводить об’єкти цього класу на екран.

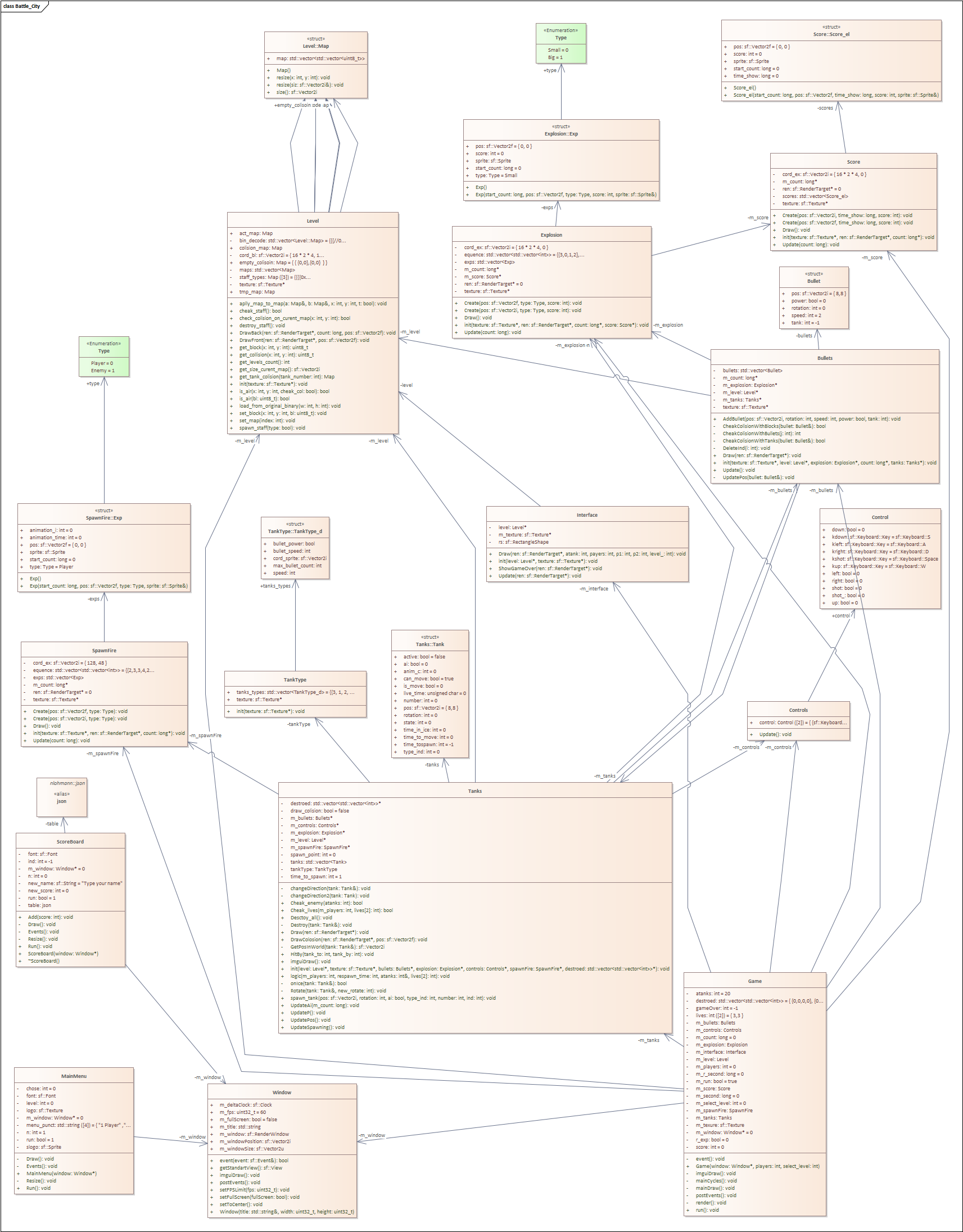
Пересування танків гравця відбувається за допомогою зміні координат в двовимірній системі. В класі Tanks в методі UpdatePos описана логіка пересування та перевірка на зіткнення з іншими об’єктами.

Пересування ворожих танків відбувається за допомогою алгоритму подібному до оригінального, він розбитий на 3 фази довжина яких залежить від рівня складності та кількості гравців. Фаза обирається за часом життя танка. На першій фазі рух танків випадковий, на другій вони починають переслідувати гравців, а на третій беруть напрям на базу гравця.

Мапа, як фізичний об’єкт являється квадратною матрицею, кожний елемент якої є типом блоку в цих координатах. Перевірка колізій між об’єктами виконується за допомогою перевірки типу блоку в координатах.

Таблиця рекордів реалізована за допомогою збереження файлу в форматі JSON.

## 3.1 UML-СХЕМА КЛАСІВ



# 4. ПЕРЕЛІК І ПРИЗНАЧЕННЯ РЕЖИМІВ ТА СТРУКТУРА ДІАЛОГУ

Дана програма має два режими роботи:

1. Без рівнів доступу користувача

2. В незалежності від рівня доступу користувача

При запуску програми перед користувачем з’являється віконна оболонка гри «Battle City» з логотипом та головним меню. Меню надає можливість обрати початковий рівень, розпочати гру, переглянути таблицю рекордів та вийти із гри. При натисненні кнопки 1 Player перед користувачем з’являється сама гра. Після завершення гри користувачу вісвітлюється таблиця рекордів, і якщо він набрав більше балів за останє місце, він може ввести своє ім’я. Далі він може повернутись до головного меню.

Діалог між системою та користувачем відбувається завдяки користувацькому, інтуїтивного інтерфейсу. Кожне меню має опис керування.

# 5. СТРУКТУРА ДАНИХ ТА РЕСУРСІВ ПРОГРАМИ

## 5.1 КЛАСИ ПРОГРАМИ

Для реалізації програмного коду використовуються такі класи, як: Bullets, Controls, Explosion, Game, Interface, Level, MainMenu, Score, ScoreBoard, SpawnFire, Tanks, TankType, Window.

|  |  |
| --- | --- |
| Клас | Використання |
| Bullets | Відповідає за обробку пуль: створення, оновлення, вивід на екран. Перевіряє та оброблює їх зіткнення з між собою, з танками, та з елементами миру. |
| Controls | Відповідає за керування танками за допомогою клавіатури |
| Explosion | Відповідає за створення та відображення вибухів |
| Game | Головний клас, що вміщує реалізацію гри і зв’язує всі головні класи. Також відповідальний за обробку виграшу та програшу |
| Interface | Відповідає за вивід інтерфейсу гри на екран |
| Level | Відповідальний за завантаження, обробку та зміну ігрового поля. Також оброблює події з головною базою гравця. |
| MainMenu | Відповідальний за керування головним меню та вибором початкового рівня |
| Score | Відповідальний за відображення зароблених балів за знищення ворожого танку |
| ScoreBoard | Відповідальний за вивід таблиці рекордів та додавання нового рекорду |
| SpawnFire | Відповідає за анімацію створення танків |
| Tanks | Відповідає за створення, пересування, зітхання з іншими танками та ігровим полем, видалення танків. Відповідає за штучний інтелект танків. |
| TankType | Зберігає інформацію про існуючі типи танків |
| Window | Відповідає за керування вікном гри |