|  |
| --- |
| МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  “ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”  Кафедра “Автоматизовані системи управління”  КУРСОВА РОБОТА  «Проектування та реалізація програмного забезпечення для представлення та взаємодії елементів гри П'ятнадцять з використанням бібліотеки OpenGL».  Керівник роботи:  асистент каф. ПІІТУ О.С. КІЗІЛОВ  Виконавець:  студент групи КН-222г О.м сАВЧЕНКО  Харків – 2023 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ | | | | | | | | | | | | |
| НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  «ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ» | | | | | | | | | | | | |
| Кафедра програмної інженерії та інтелектуальних технологій управління | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
|  | | | | | | | | | | | | |
| **КУРСОВИЙ ПРОЕКТ**  **(РОБОТА)** | | | | | | | | | | | | |
| з | Основи програмування (частина 2) | | | | | | | | | | | |
| (назва дисципліни) | | | | | | | | | | | | |
| на тему: | | Проектування та реалізація програмного забезпечення для представлення та взаємодії елементів гри П’ятнадцять з використанням бібліотеки OpenGL | | | | | | | | | | |
|  | | | | | | | | | | | | |
| Студента | 1 | | курсу | | | КН-222г | | групи |
| спеціальності | | 121 Програмна інженерія | | | | | | | | |
|  | | | | | | | | |
| Савченко О.М. | | | | | | | | |
| (прізвище та ініціали) | | | | | | | | |
| Керівник | асистент каф. ПІІТУ, Кізілов О.С | | | | | | | |
|  | | | | | | | | |
| (посада, вчене звання, науковий ступінь, прізвище та ініціали) | | | | | | | | |
|  | | | | | | | | | | |
| Національна шкала | |  | | | | | | | |
| Кількість балів |  | | | Оцінка ECTS | | |  | | | |
|  | | | | | | | | | | | | | |
| Члени комісії |  | | |  | О.С. КІЗІЛОВ | | | | | |
|  | (підпис) | | |  | (прізвище та ініціали) | | | | | |
|  |  | | |  |  | | | | | |
|  | (підпис) | | |  | (прізвище та ініціали) | | | | | |
|  |  | | |  |  | | | | | |
|  | (підпис) | | |  | (прізвище та ініціали) | | | | | |
|  | | | | | | | | | | | | |
| м. Харків – 2023 рік | | | | | | | | | | | | |

Міністерство освіти і науки України

Національний технічний університет "Харківський політехнічний інститут"

Кафедра програмної інженерії та інформаційних технологій управління

Студент Савченко О. М. Група КН-222г Курс 1

З А В Д А Н Н Я

на науково-дослідну курсову роботу  
з курсу "Основи програмування (частина 2)"

Тема:

Проектування та реалізація програмного забезпечення для представлення та взаємодії елементів гри ... з використанням бібліотеки OpenGL

Постановка задачі:

Під час виконання роботи необхідно спроектувати та реалізувати програмне забезпечення для представлення та взаємодії елементів гри ... . Необхідно реалізувати графічні об’єкти, описати їх взаємодію.

Програма повинна бути створена з використанням бібліотеки OpenGL. Необхідно реалізувати наступні можливості програми:

* створення нового поля гри (очищення вікна та відтворення початкової ситуації;
* вибір та додавання елементу гри за допомогою функцій клавіатури;
* маніпуляції з елементами гри відповідно до завдання;
* обертання поля гри;
* зберігання результатів (рекорди, рівень) та відтворення їх за бажанням користувача.

Для представлення елементів та опису їхньої взаємодії слід застосувати об’єктно-орієнтований підхід. Слід представити необхідну інформацію про програму та способи її застосування (презентацію).

Короткий зміст роботи:

а) реферативна частина

Дослідження поставленої задачі. Дослідження та аналіз необхідних програмних методів розв’язання задачі. Постановка задачі.

б) теоретична частина

Визначення вимог щодо програмного забезпечення.. Розробка необхідних UML-діаграм.

в) програмна частина

Реалізація та тестування програмного забезпечення. Опис інтерфейсу користувача

Термін демонстрації програмного забезпечення 10.05.2023

Термін захисту курсової роботи 20.05.2023

Керівник курсової роботи (Олексій КІЗІЛОВ)

ВІДГУК

на курсову роботу студента групи КН-222Г Савченко О. М.

Курсова робота присвячена проектуванню та реалізації програмного забезпечення для представлення та взаємодії елементів гри П'ятнадцять з використанням бібліотеки OpenGL .

Під час виконання курсової роботи здійснено аналіз гри "П'ятнацять", проектування її механіки та реалізацію програмного забезпечення для її відтворення.

Робота виконана та може бути захищена.

Керівник курсової роботи,

асистент каф. ПІІТУ. Олексій КІЗІЛОВ

РЕФЕРАТ

КР: 28 с., 8 рис., 7 джерел

*Ключові слова*: КОМП'ЮТЕРНА ГРА, OPENGL, ІНІЦІАЛІЗАЦІЯ, 3D-ГРАФІКА, FREEGLUT, ГРАФІЧНИЙ ІНТЕРФЕЙС, ГЕЙМПЛЕЙ.

Об'єкт дослідження є гра "П'ятнадцять", написана на мові програмування C++ з використанням бібліотеки FreeGlut на платформі OpenGL.

Метою даної курсової роботи є розробка програми, яка реалізує комп'ютерну 3D гру під назвою "П'ятнадцять", що включає створення функціональності для гри, графічний інтерфейс, обробку подій, перемішування та розв'язування гри.

Результати розробки є функціонуюча гра "П'ятнадцять", що включає можливість настройки головного поля гри, ініціалізації нової гри, перегляд результатів та розв'язування гри "П'ятнадцять". Гра має графічний інтерфейс та дозволяє користувачеві взаємодіяти за допомогою миші або клавіатури.

Геймплей даной гри є збиранням чисел в правильному порядку, з метою впорядкування кубів відповідно до номерів, розташованих на них.

Розроблене програмне забезпечення дозволяє користувачу насолоджуватися грою "П'ятнадцять" на комп'ютері з операційною системою, підтримуваною бібліотекою FreeGlut та платформою OpenGL. Це демонструє можливості мови програмування C++ у створенні ігрових програм з використанням графічних бібліотек.

Подальшим розвитком цієї гри може бути вдосконалення графіки, додавання текстур, додавання рівнів складності та видів кубів і створення нового функціоналу.

ABSTRACT

SW: 28 p., 8 fig., 7 sources

Keywords: COMPUTER GAME, OPENGL, INITIALIZATION, 3D GRAPHICS, FREEGLUT, GRAPHIC INTERFACE, GAMEPLAY.

The object of research is the game "Fifteen", written in the C++ programming language using the FreeGlut library on the OpenGL platform.

The purpose of this course work is to develop a program that implements a 3D computer game called "Fifteen", which includes creating functionality for the game, graphical interface, event handling, mixing and solving the game.

The result of the development is a functioning game "Fifteen", which includes the ability to set the main game field, initialize a new game, view the results and solve the game "Fifteen". The game has a graphical interface and allows the user to interact using a mouse or keyboard.

The gameplay of this game is to collect the numbers in the correct order, in order to arrange the cubes according to the numbers located on them.

The developed software allows the user to enjoy the game "Fifteen" on a computer with an operating system supported by the FreeGlut library and the OpenGL platform. This demonstrates the capabilities of the C++ programming language in creating game programs using graphics libraries.

Further development of this game can be improving graphics, adding textures, adding levels of difficulty and types of cubes and creating new functionality.

ЗМІСТ

[ВСТУП 8](#_Toc137203758)

[1. ТЕОРЕТИЧНІ ТА ІСТОРИЧНІ АСПЕКТИ КОМП’ЮТЕРНОЇ ГРИ «П'ятнадцять» 9](#_Toc137203759)

[1.1 Історія та походження гри 9](#_Toc137203760)

[1.2 Правила гри "П’ятнадцять" 10](#_Toc137203761)

[1.3 Постановка задачі 11](#_Toc137203762)

[1.4 Основні функції програмного забезпечення 12](#_Toc137203763)

[1.5 Графічні засоби OpenGL для реалізації комп'ютерної гри 12](#_Toc137203764)

[2 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 14](#_Toc137203765)

[2.1 Діаграма варіантів використання 14](#_Toc137203766)

[2.2 Визначення структури класів 17](#_Toc137203767)

[2.3 Діаграма діяльності 19](#_Toc137203768)

[3 ТЕСТУВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ 21](#_Toc137203769)

[3.1 Опис розробленого програмного забезпечення 21](#_Toc137203770)

[3.2 Взаємодія з користувачем 23](#_Toc137203771)

[4 ВИСНОВКИ 27](#_Toc137203772)

[5 СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ 29](#_Toc137203773)

# ВСТУП

У сучасному світі комп'ютерні ігри є популярним видом розваги, що пропонують користувачам різноманітні сценарії та виклики. Одним з найцікавіших жанрів є головоломки, які сприяють розвитку мислення, логіки та уяви, а комп'ютерна графіка надає унікальні можливості для створення цікавих та креативних ігрових програм.

У рамках курсової роботи буде створено комп'ютерну версію гри-головоломки "П'ятнадцять", яка знайомить гравців із класичною графікою, правилами та принципами цієї гри, що дозволить гравцям насолоджуватись цією захоплюючою грою на своєму персональному комп'ютері.

Курсова робота є описом розробки та реалізації цієї головоломки з використанням мови програмування С++ та бібліотеки FreeGLUT, що передбачає використання сучасних методів програмування та графічних технологій для створення естетичного та функціонального ігрового середовища.

# ТЕОРЕТИЧНІ ТА ІСТОРИЧНІ АСПЕКТИ КОМП’ЮТЕРНОЇ ГРИ «П'ятнадцять»

## Історія та походження гри

Гра "П'ятнашки" виникла в кінці 19 століття і стала популярною логічною грою по всьому світу. Її винайшли у 1874 році в США. Перша відома комерційна версія гри була представлена в 1880 році.

Основна мета гри полягає у тому, щоб перемістити числа від 1 до 15 (або від 1 до 16, в залежності від розміру гри) на дошці розміром 4x4 (або 4x4, 5x5 і т.д.) шляхом переміщення плиток по дошці. У центрі дошки залишається порожнє місце, що дозволяє переміщати плитки поруч з ним. Мета полягає в тому, щоб впорядкувати числа у порядку зростання, зліва направо і від верхнього до нижнього рядка.

Гра "П'ятнашки" завойовала популярність завдяки своїй простоті і одночасно складному завданню. Це гра, яка сприяє розвитку логічного мислення, пам'яті та стратегічного планування. Існує багато варіацій гри з різними розмірами дошок та складностями.

З часом гра "П'ятнашки" також стала доступною у віртуальному середовищі, і сьогодні її можна знайти у вигляді мобільних додатків, онлайн-ігор та комп'ютерних програм. Вона залишається популярною грою, яка привертає увагу людей різного віку і залишається відмінним способом провести час і розумово попрацювати.

## Правила гри "П’ятнадцять"

Гра є набором 15 однакових кубів (4 x 4), з нанесеними числами від 1 до 15, укладеними на квадратну дощечку (Рис. 1.2.1). Довжина сторони дошки в чотири рази більша за довжину сторони кубів, відповідно залишається незаповненим одне квадратне поле.

Изображение выглядит как прямоугольный, снимок экрана, Прямоугольник, число

Автоматически созданное описание

Рисунок 1.2.1- Гра "П’ятнадцять"

Гра складається з послідовних кроків. На кожному кроці гравець може перемістити один з кубів, поряд якого знаходиться квадратне поле, з місця на дане поле в кожному не діагональному напрямку.

Мета гри полягає в розміщенні кубів відповідно цифр в порядку зростання та в тому, щоб знайти найкоротший шлях до вирішення гри.

Гравець вважається переможцем, якщо він досягає кінцевої конфігурації, де числа від 1 до 15 розташовані вірно, а останнє поле порожнє.

Під час гри здійснюється підрахунок часу (Рис. 1.2.2).

Изображение выглядит как Шрифт, снимок экрана, Графика, текст

Автоматически созданное описание

Рисунок 1.2.2 – Таймер

## Постановка задачі

Необхідно розробити програму, яка реалізує уявну комп'ютерну гру під назвою " П'ятнадцять ".

Програма повинна реалізовувати такі основні функції:

* ініціалізація поля гри (встановлення початкових значень);
* вибір елементів гри за допомогою клавіатури та маніпулятора "миша";
* маніпуляція елементами відповідно до правил гри;
* обертання поля гри;
* масштабування;
* підрахунок і зберігання статистики ігор.

Реалізація програми повинна здійснюватися з використанням об'єктно-орієнтованих технологій. Зокрема, необхідно створити класи для представлення основних елементів гри.

## Основні функції програмного забезпечення

Програма повинна реалізовувати такі основні функції:

1 підготовка нового поля гри;

2 реалізація гри з можливістю маніпуляції за допомогою маніпулятора "миша";

3 підрахунок часу та виведення результатів;

1. можливість налаштування поля гри.

## 1.5 Графічні засоби OpenGL для реалізації комп'ютерної гри

Програма повинна містити засоби 3D реалізації, тому при створенні гри використовується бібліотека віконного інструментарію OpenGL FreeGLUT.

FreeGLUT (відкрита альтернатива застарілому OpenGL Utility Toolkit (GLUT)) - це бібліотека програмування, призначена для створення графічного інтерфейсу користувача (GUI) та обробки подій у віконному середовищі.

FreeGlut дозволяє створювати та відображати 3D графіку з використанням OpenGL. Вона надає функції для створення та управління графічним контекстом, налаштування параметрів відображення, рендерингу тривимірних об'єктів, управління камерою та освітленням.

За допомогою FreeGlUT потрібно створити тривимірну сцену, моделювати та відобразити об'єкти з можливим використанням полігональної геометрії, текстур, матеріалів та шейдерів. Також потрібно використовувати керування положенням та орієнтацією об'єктів, а також застосовувати трансформацію для їх масштабування, повороту та переміщення.

FreeGlut надає можливість обробляти введення з клавіатури та миші, що дозволяє створювати інтерактивні 3D програми. Тому програма мусить реагувати на події введені користувачем, такі як натискання клавіш, переміщення миші або клацання кнопок миші, і змінювати поведінку сцени та об'єктів відповідно.

FreeGlut підтримує освітлення у 3D сценах. Програма повинна використовувати джерела світла, такі як спрямовані світлові джерела, точкові джерела світла та прожектори, та налаштовувати їх параметри, такі як позиція, інтенсивність, колір та типи освітлення (наприклад, дифузне, дзеркальне та ін.).

Ця бібліотека дає розробникам 3D ігор і додатків свободу у створенні складних сцен, створенні інтерфейсів і обробці подій. Вона є потужним інструментом для створення реалістичних та інтерактивних візуалізацій у 3D просторі.

Glut (OpenGL Utility Toolkit) є невеликою бібліотекою, розробленою для спрощення створення та роботи з віконними системами на основі OpenGL. Однак початкова версія Glut була створена для використання в мові програмування C. Надалі з'явилася вільна версія Glut під назвою freeGlut, яка підтримує як C, а й інші мови програмування, включаючи C++.

# ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 2.1 Діаграма варіантів використання

Функціональні вимоги до програми наведені на діаграмі варіантів використання (Рис. 2.1.1). Ця діаграма включає в себе одну дійову особу (Користувач), яка може взаємодіяти з нашою програмою декількома способами:

Оновлення поля гри, у програмі має бути реалізація функції підготовки поля гри 4х4, яке передбачає генерацію ввипадкового рпозташування кубів та скидання таймера гри;

Реалізація гри з можливістю маніпуляції за допомогою маніпулятора "миша". Повинен бути забезпечений інтерактивний інтерфейс, який дає змогу гравцеві переміщати елементи гри за допомогою курсору миші. Програма повинна коректно зчитувати, який елемент був обраний користувачем;

Налаштування поля гри з можливістю взаємодії за допомогою маніпулятора "клавіатура" та “миша”, яке передбачає наближення чи віддалення поля та його обертання;

Здійснення гри з підрахунком часу. Повинен бути реалізований функціонал для відстеження часу гри і підрахунку часу, що минув з моменту початку гри;

Перегляд результатів;

Вихід з програми при натисненні клавіші “Esc” та відновлення гри з початкового стану за допомогою клавіші F2.

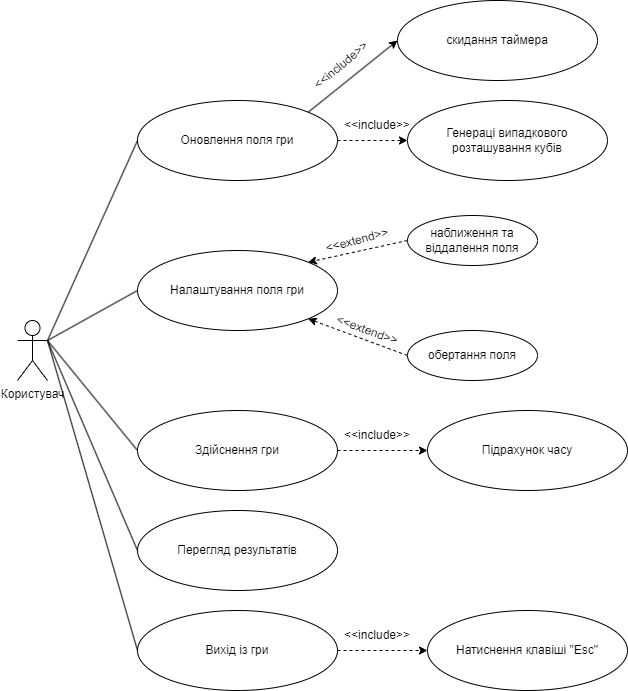
****

Рисунок 2.1.1 – Діаграма варіантів використання

## Визначення структури класів

Для організації проекту рекомендовано розділити сирцевий код на окремі незалежні частини, використовуючи простори імен або пакети, а також фізично розміщувати класи в окремих файлів.

У нашому випадку, виділено простір імен CubeGame для класів та функцій, специфічних для гри, та простір імен GraphUtils для графічних функцій і констант, які можуть бути корисними в інших подібних проектах. Фізично кожен значний клас слід розташувати в окремому сирцевому файлі. У відповідності до цього, класи розміщено в окремих файлах реалізації (Shape.h, Shape.cpp, Cube.h, Cube.cpp…).

Основним класом, що відповідає за логіку гри, є клас Scene (сцена). Він містить посилання на об'єкти, які є похідними від абстрактного класу Shape. Крім того, на діаграмі можна відобразити функцію main() для уявного зображення контексту використання класів.

Основний клас проекту – Scene. Цей клас представляє геометрію сцени, контролює правила гри, розташування елементів, а також реалізує обробку подій GLUT.

Клас Shape є базовим класом для інших фігур. До опису класу слід додати поля, які репрезентують атрибути, наведені в діаграмі класів. Відповідно слід створити функції доступу. Клас Shape буде абстрактним, оскільки він надаватиме суто віртуальну функцію draw(), яка буде перекрита в похідних класах. Також необхідно додати конструктор з параметрами.

Крім цього потрібно створити заголовний файлу utils.h і файл реалізації utils.cpp, так як для відображення фігур, їх кольорів та розташування допоміжних елементів гри можуть придатися масиви, змінні та функції, які не входять у жоден клас.

Незважаючи на це, потрібно створити класи, похідні від Shape. Кожний клас представлено заголовним файлом і файлом реалізації. Таким чином було створено клас Board та клас Cube.

Для наглядного демонстрування класів було зроблено діаграму класів (Рис. 2.2.1).

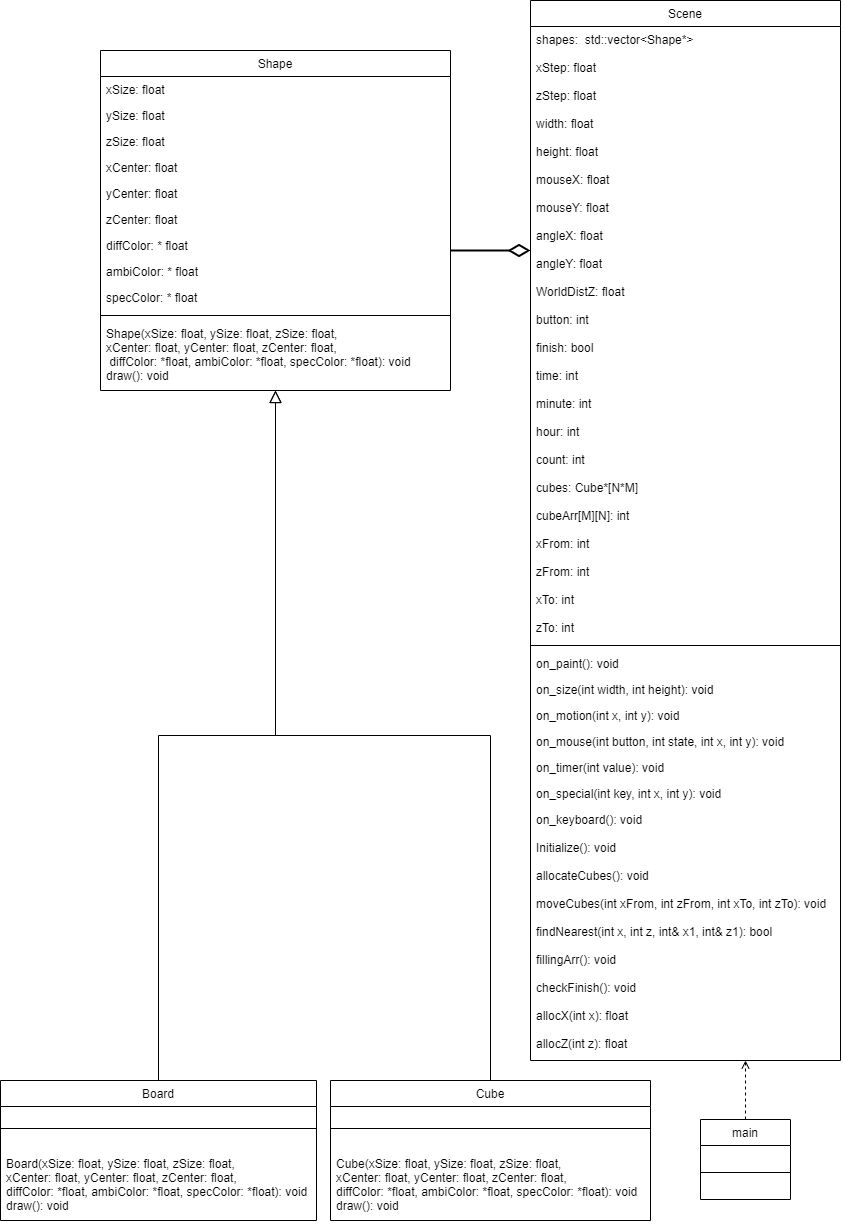


Рисунок 2.2.1 – Діаграма класів

## Діаграма діяльності

На рис. 2.3.1 наведено діаграму діяльності ініціалізації сцени. Діаграма починається з того, що відбувається ініціалізація масиву, що зберігає дані про інформацію про поля або позиції на сцені. Після цього нам потрібно виконати "перетасування" елементів масиву за допомогою функції shuffle(). Потім має викликатися функція для розміщення кубів відповідно до вмісту масиву, ініціалізованого раніше.

Після цього ініціалізуються такі змінні:

* 2 змінні (по Х і Y) для визначення кутів повороту або нахилу в тривимірному просторі.
* 1 змінна для визначення відстані по осі Z у тривимірному просторі.
* 3 змінні для відстеження часу або кроків у процесі виконання програми.
* 1 змінна для відстеження завершення гри.



Рисунок 2.3.1 – Діаграми діяльності для ініціалізації сцени

# ТЕСТУВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

## 3.1 Опис розробленого програмного забезпечення

Основним файлом у нашому проекті є Scene.cpp. Цей файл відіграє важливу роль, оскільки в ньому реалізовані основні функції гри. Він обробляє взаємодію з користувачем, відповідає за ініціалізацію гри, управління графікою та обробку подій. Scene.cpp є центральною точкою проекту, де знаходяться всі компоненти гри та взаємодіють між собою.

У файлі Scene.cpp відбуваються такі важливі процеси:

Додавання елементів гри до кінця масиву за допомогою методу push\_back. Кожен новий елемент є об'єктом класі Cube та Board, який створюється за допомогою оператора new. Конструктор класу Cube використовується для ініціалізації нового об'єкта з параметрами, такими як розмір, колір, текстура та інші характеристики об'єкта.

Ініціалізація виконується на початку гри та під час оновлення гри. Спочатку заповнюється масив "cubeArr", який відповідає за стан ігрового поля. Потім масив тасується за допомогою функції “shuffle()” з бібліотеки <algorithm>. Оскільки двовимірний масив у C++ зберігається як одновимірний, його приводять до типу одновимірного масиву та передають у функцію тасування. Після цього розташовуються куби за допомогою функції "allocateCubes".

Ініціалізація значень елементів масиву, який відображає розміщення кубів. У функції для ініціалізації масиву проходимося по всьому масиву кубів " cubeArr [i][j]" і прирівнюємо кожен елемент значеннями від 1 до 15. Кожен куб (елемент масиву cubeArr [i][j]) отримує унікальне значення, що відповідає його розташуванню на ігровому полі.

Функція on\_paint() відповідає за відображення гри на екрані. Вона викликається кожен кадр і відображає куби (елементи масиву cubeArr[i][j]) на відповідних позиціях у графічному вікні. Також ця функція відповідає за оновлення візуального відображення на екрані вікна/сцени, включаючи текстове повідомлення, обробку матеріалів, джерел світла, а також малювання фігур та вимикання відповідних режимів.

Функція on\_timer() відповідає за нарахування секунд, хвилин, та годин від початку роботи програми або його оновлення. Нарахування кожної секунди здійснюється за допомогою лічильника “tick”, де кожні 25 мс., здійснюється нарахування 1 секунди та перемалювання поля (on\_paint()).

Функція on\_motion() перевіряє події, як рух миші з натисненою кнопкою, і виконує відповідні дії. Наприклад, якщо користувач натискає клавішу для переміщення куба, виконується відповідна функція для переміщення куба на ігровому полі, якщо клавішу для обертання, здійснються обератання поля. За вид дії, відповідно натисненої кнопки, відповідає функція on\_button().

Функція on\_special() відповідає за події, як натискання спеціальних клавіш (стрілка вгору, стрілка вниз, “F2”). Наприклад, якщо користувач натискає клавішу стрілка вгору або стрілка вниз, виконується маштабування ігрового поля, при натискані на клавішу “F2” гра оновлюється.

Функції, такі як findNearest() та moveCubes() відповідають за головнгу логіку гри. findNearest() – функція за допомогою якої здійснюється пошук найближчого об’єкта (куба) для переміщення. moveCubes() – функці за дпомогою якого, здійснюється збереженн координат до та після переміщення, а також перевіряються умови для пересування кубів.

Також наявна функція перевірки стану гри – checkFinish(), яка відповідає за перевірку правильності розташування кубів на полі. Якщо куби відсортовано від 1 до 15 і останнє поле в правому нижньому кутку є пустим, то гра завершується та відображає результати.

Код Scene.cpp також може містити інші допоміжні функції та класи, які використовуються для реалізації функціональності гри. Наприклад, можуть бути класи для обробки колізій, розрахунку фізики або анімації об'єктів.

Загалом, файл Scene.cpp містить основну логіку гри, яка включає створення гри, оновлення її стану, відображення на екрані та обробку подій користувача. Цей файл зазвичай використовується разом з іншими файлами та бібліотеками для створення повноцінної гри.

## Взаємодія з користувачем

Гра "П'ятнадцять" — це головоломка, в якій гравець повинен впорядкувати числа від 1 до 15 на дошці розміром 4x4. На початку гри користувача чекає повністю оновлене поле, де початковий стан гри виглядає так, що числа від 1 до 15 випадковим чином розташовані на дошці, а останнє поле залишається порожнім (Рис. 3.2.1). Крім цього, при кожному початку гри таймер починає свій відлік від початкового значення, яке дорівнює нулю.

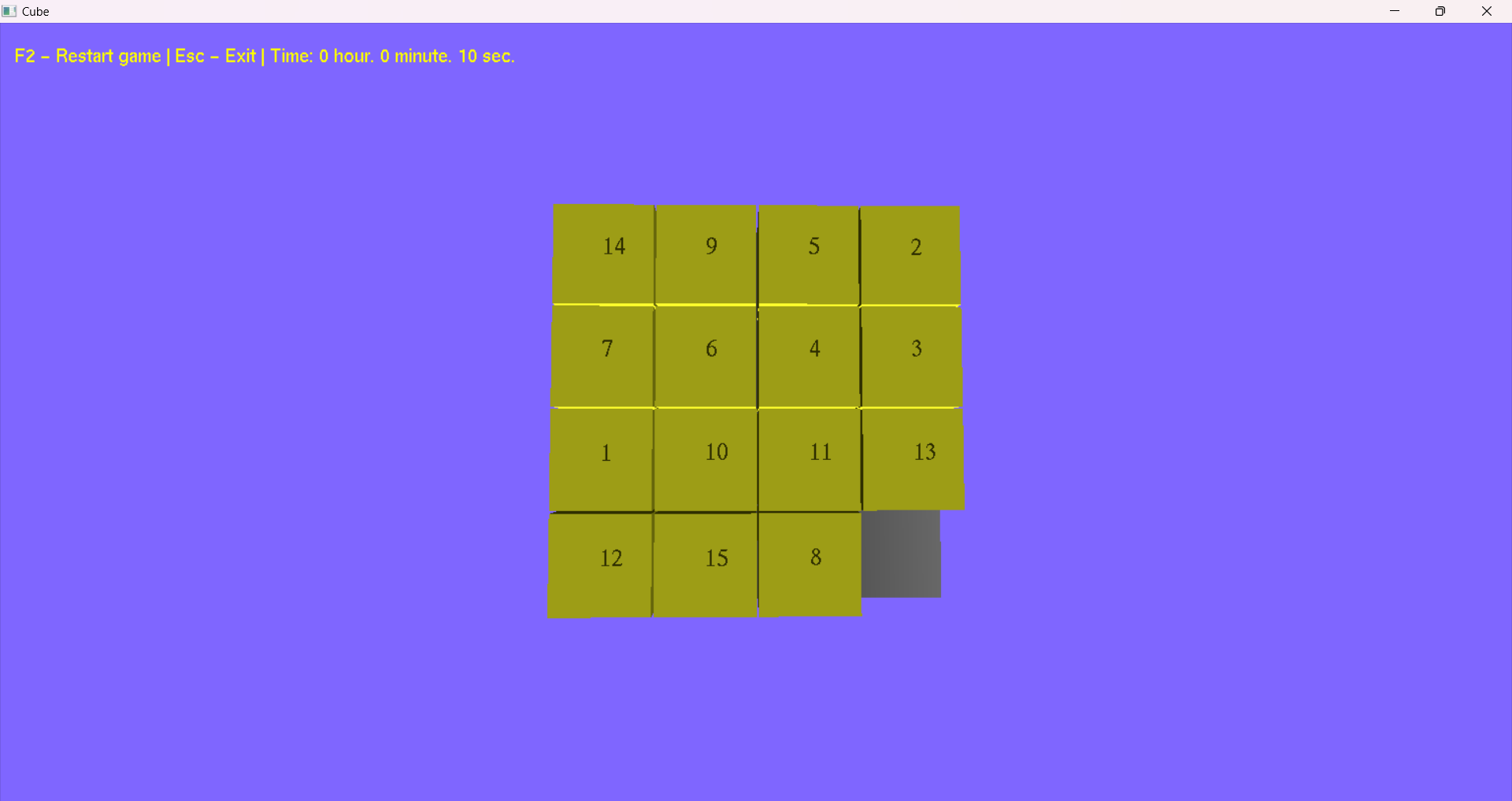


Рисунок 3.2.1 – Початковий стан гри

При кожному запуску, програма автоматично ініціалізує нове поле, проте при натисканні на клавішу **“F2”**, поле гри можна оновити в будь-який момент під час якого, відбувається нова ініціалізації гри та нове випадкове розташування кубів.

Поле гри можна налаштувати двома способами:

1 Натиснувши праву кнопку миші **“ПКМ”**, ми зможемо регулювати поле, обертаючи його навколо будь-якої осі.

2 Натиснувши на клавіатурі стрілку вгору “”, поле починає віддалятися від екрану, натиснувши на клавіатурі стрілку вниз “”, поле починає наближатися до екрану відповідно.



Рисунок 3.2.2 – Маштабування та обертання поля

Для того, щоб взаємодіяти (перетягувати) куби по дошці потрібно обрати потрібний об’єкт біля якого знаходиться пусте квадратне поле, натиснувши на нього лівой кнопкой миші **“ЛКМ”**, перетягнути на дане пусте поле (Рис. 3.2.3).

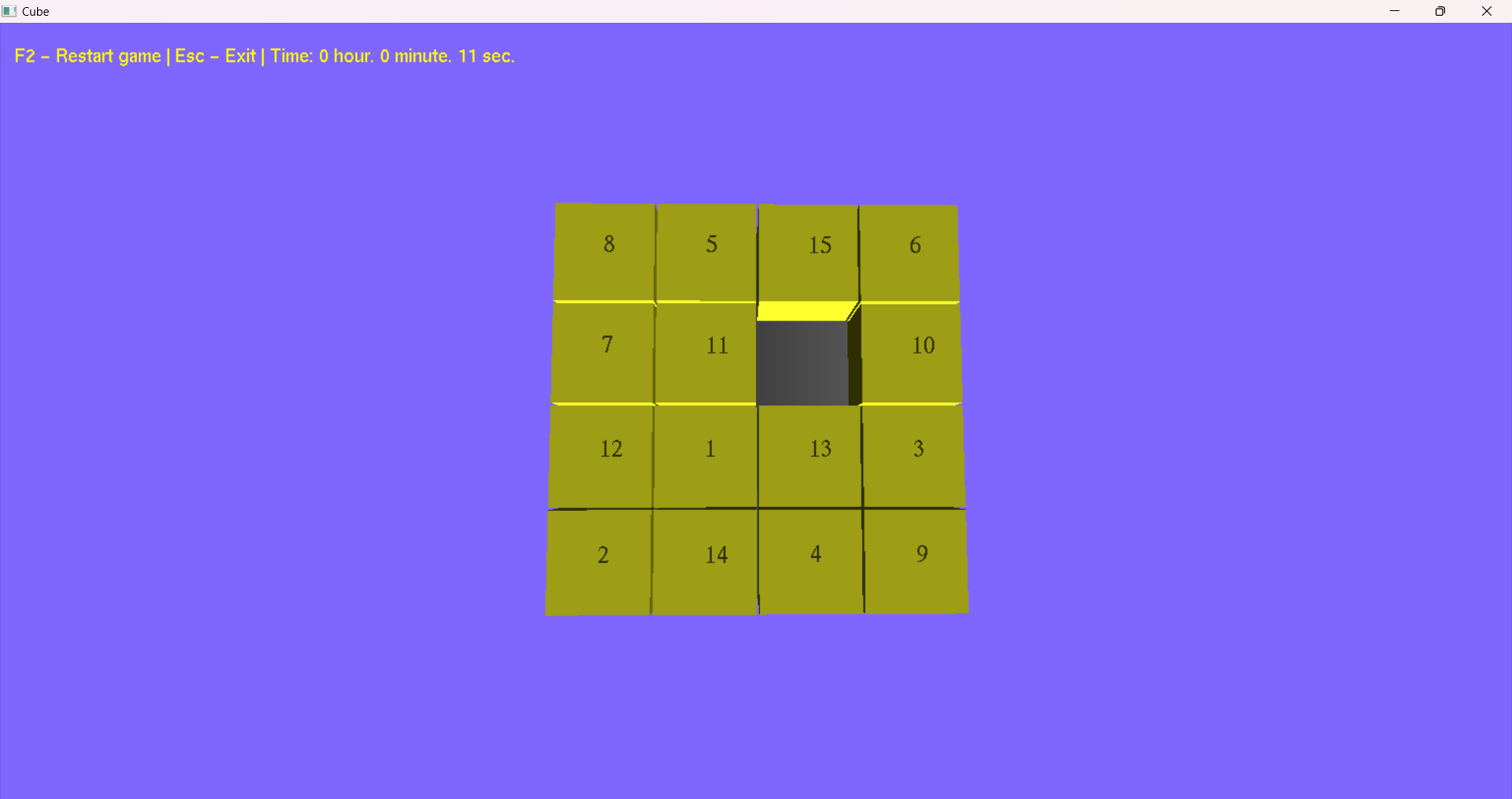


Рисунок 3.2.3 – Взаємодія з об’єктами

Для того, щоб вийти з програми достатньо натиснути на клавішу **“Esc”** або в правому кутку вікна програми натиснути на позначку “закрити”.

# ВИСНОВКИ

В даній роботі було розглянуто проектування та розробку гри "П'ятнашки" з використанням мови програмування C++ та бібліотек OpenGL і FreeGLUT. Процес розробки гри пройшов через кілька етапів, включаючи проектування та реалізацію головного функціоналу.

На першому етапі було проведено постановку основних задач та проектування гри, де було створено детальний опис гри, визначено правила та механіку, а також вимоги до інтерфейсу та дизайну. Макети головного меню, ігрового поля та елементів гри були розроблені для подальшої реалізації.

Другий етап включав реалізацію головного функціоналу гри. З використанням мови програмування C++ та бібліотеки FreeGLUT на платформі OpenGL був створений графічний інтерфейс, який відображав ігрове поле та дозволяв переміщати куби. Також була реалізована перевірка правильного розташування кубів, забезпечуючи логіку гри. При реалізації було враховано принципи об'єктно-орієнтованого програмування, що сприяло структурованості та модульності коду.

Загальним результатом даної курсової роботи є розроблена гра "П'ятнашки" з використанням мови програмування C++ та бібліотек OpenGL і FreeGLUT, що має графічний інтерфейс, функціонал переміщення кубів та перевірку правильного розташування.

Гра представляє першу версію програми, яка повністю функціонує і готова до використання на платформі Windows. Однак, існує потенціал для подальшого розширення та вдосконалення функціоналу гри, що дозволить користувачам отримати ще більше задоволення від гри.

Гра "П'ятнашки" може бути прекрасним засобом розваги та відпочинку після напруженої роботи. Вона дозволяє користувачам розслабитись і розважитись, використовуючи свої логічні та стратегічні навички.

# СПИСОК ДЖЕРЕЛ ІНФОРМАЦІЇ

1. 3D Graphics with OpenGL // <https://www3.ntu.edu.sg/home/ehchua/programming/opengl/CG_Examples.html> (Дата звернення 25.03.2023).
2. OpenGL tutorials // <http://www.opengl-tutorial.org/beginners-tutorials/tutorial-1-opening-a-window/> (Дата звернення 20.03.2023).
3. NeHe gamedev // <http://nehe.gamedev.net/tutorial/simpleglxwindow_class/35003/> (Дата звернення 01.04.2023).
4. FreeGLUT // <https://ru.wikipedia.org/wiki/Freeglut> (Дата звернення 05.06.2023).
5. The Free OpenGL Utility Toolkit // <http://freeglut.sourceforge.net/> (дата звернення 28.05.2023)
6. Грехема Селлса, Ніколаса Хоу, Річарда С. Райт. "OpenGL SuperBible" – 2015р.
7. Роберта Ніструпа. "Game Programming Patterns" – 2014р.