**ВСТУП**

Актуальність: в наші часи особливо широке розповсюдження отримали різноманітні комп’ютерні ігри - явище інформаційного соціуму, що набуло виключну популярність за останні десятиліття. Вони привертают велику аудиторію, до якої входять представники усіх вікових груп та верств населення. Також спостерігається стрімкий розвиток ігрового програмного забезпечення, постійне удосконалення методів реалізації ігрових процесів, графічних представлень, розробка нових методів взаємодії користувача з додатком.

Значимість: C++ – це одна з найпопулярніших мов програмування, що володіє великою гнучкістю та широкими можливостями для розробки. Структурний підхід включає в себе всі стадії розробки проекту, а саме: специфікація, проектування і тестування. C++ використовується в прикладних програмах, операційних системах, драйверах різних пристроїв, високопродуктивних серверах і навіть у розважальних додатках.

SFML є портативним і легким у використанні API для мультимедіа-програмування. Написано мовою C++, але доступні прив’язки і для C, D, Python, Ruby, OCaml, .Net. Це об'єктно-орієнотована альтернатива для SDL. SFML забезпечує використання 2D-графіки з апаратним прискорення OpenGL. SFML також може бути використаний для OpenGL-проектів. SFML також надає різні модулі для полегшення програмування ігор і мультимедійних додатків. Вихідний код поширюється згідно з умовами zlib/png ліцензії.

Мета: дослідити доречність використання бібліотеки SFML та її модулів для розробки ігор мовою програмування C++; продемонструвати корисність роботи деяких з модулів бібліотеки, а саме Графічного та Аудіо модулів, на прикладі гри «Маріо».

Завдання: написати програму, в якій здійснюється управління користувачем ігрового персонажу, з метою набрати найбільшу кількість очок.

Методи дослідження:

* експеримент – активне цілеспрямоване вивчення об’єкта у визначених умовах, які необхідні для виявлення певних властивостей;
* аналіз та синтез – розчленування цілого на складові та вивчення предмета в цілісності, єдності й взаємозв’язку його частин;
* формалізація – вивчення об’єкта шляхом визначення його структури, що сприяє формуванню оптимальних моделей, однозначності.

**РОЗДІЛ 1. АНАЛІЗ ЗАВДАННЯ**

Важливим елементом для створення даного проекту є детальний аналіз його теоретичної складової, головних модулів обраної SFML бібліотеки, а також аналіз вхідних та вихідних даних програми.

**1.1. Теоретична складова**

SFML – мультиплатформна, проста і швидка мультимедіа-бібліотека. SFML є простим інтерфейсом для різноманітних компонентів комп’ютера, що необхідний спрощення процесу розвитку ігор та мультимедійних додатків. Він складається з п’яти модулів:

• «Системний модуль (System module SFML)» – забезпечує управління системними параметрами;

• «Віконний модуль (Window module SFML)» – забезпечує управління вікнами та взаємодію програми з користувачем;

• «Графічний модуль (Graphics module SFML)» – забезпечує відображення графічних елементів та управління ними;

• «Аудіо модуль (Audio module SFML)» – надає інтерфейс для обробки звуків і музики;

• «Мережевий модуль (Network module SFML)» – забезпечує управління мережевими функціями за допомогою сокетів.

Переваги використання мови програмування C++ у даній роботі:

• швидкодія – швидкість роботи програм на С++ практично не поступається програмам на С, хоча програмісти отримали в свої руки нові можливості і нові засоби;

• масштабованість – на мові C++ розробляють програми для найрізноманітніших платформ і систем;

• можливість роботи на низькому рівні з пам'яттю, адресами, портами;

• можливість створення узагальнених алгоритмів для різних типів даних, їхня спеціалізація, і обчислення на етапі компіляції, з використанням шаблонів;

• підтримуються різні стилі та технології програмування, включаючи традиційне директивне програмування, ООП, узагальнене програмування, метапрограмування (шаблони, макроси).

**1.2. Аналіз вхідних/вихідних даних**

Вхідними даними у програмі виступають клавіші «вверх», «вниз», «вправо», «вліво», що призначені для керування персонажем гри.

Вихідними даними програми є відповідні дії Маріо, що в тій чи іншій мірі впливають на процес проходження гри.

**РОЗДІЛ 2. ……..**

Проект створювався методом спроб та помилок, з постійними змінами та удосконаленнями, як у плані продуктивності, так і у плані зручності для користувача.

**2.1. Загальний опис продукту**

На початку виконання програми в основному файлі main.cpp відбувається підключення основних файлів та бібліотек, таких як:

СПИСОК

ТЕКСТ

**2.2. Опис головних класів і змінних програми**

1. void Menu(RenderWindow& window, int& num) – основна функція, що відповідає за меню. Вона приймає два параметри:

• RenderWindow& window – при запуску гри створюється новий об’єкт типу RenderWindow, який відповідає за ігрове вікно;

• int& num – змінна, що відповідає за кількість життів персонажа, якщо ж її значення рівне 0, то головне меню знову вмикається;

1. bool isGameStarted(int& num, int& level, bool& isEnd) – головна функція, що відповідає за ігровий процес, а саме:

• створення вікна;

• вирізання текстур з графічного файлу;

1. void GameRunning(int& num, int& level, bool& isEnd) - функція, що відповідає за продовження ігрового процесу після смерті персонажа;
2. class Mushroom : public Entity - похідний клас для грибів, що випадають зі секретних блоків.

**ТЕКСТ**

**Приклад написання підпункту:**

**«**променями; число різних визначальних входжень у кожному промені більш важливо, ніж загальне число використовуваних входжень змінних у кожному промені.

**2.2.4. Метрики складності потоку керування і даних програми**

Четвертим класом метрик є метрики, близькі як до класу кількісних метрик, класу метрик складності потоку…**»**

**2.3. Інтерфейс програми**

На рисунку 2.1 зображено головне меню гри.

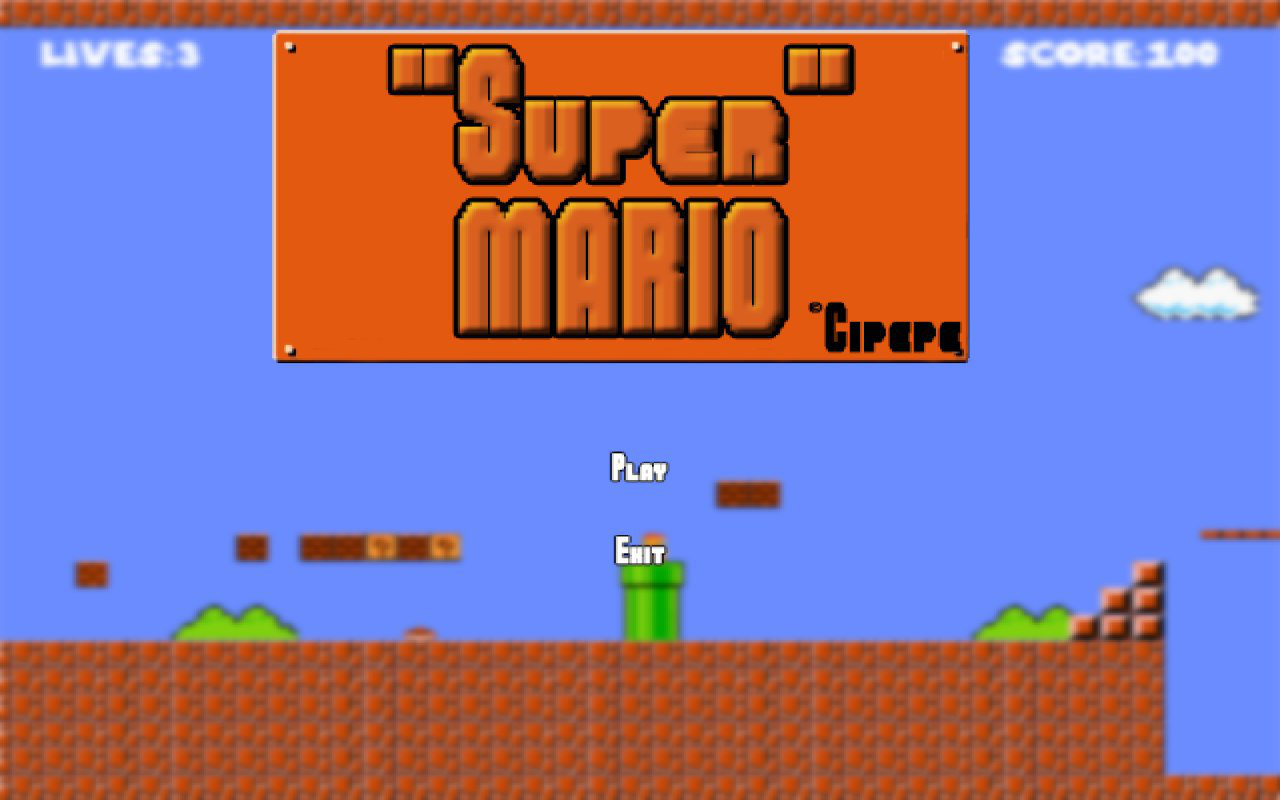


Рисунок 2.1. Головне меню

ТЕКСТ АБОЩО

**РОЗДІЛ 3. ……..**

**ТЕКСТ**

**Приклад оформлення формул:**

**«**Міра складності М обчислюється за формулою 3.1:

(3.1)

де fi – частота появи i-го символу;

pi – ймовірність його появи.**»**

**Приклад оформлення таблиць**

**«**Це можна побачити в таблиці 2.6, де показано який критерій моделі якості описує той чи інший набір метрик.

Таблиця 2.6. Відповідність метрик критеріям моделі якості

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | LOC-метрики | Метрики Холстеда | Метрики Джилба | Метрика МакКейба |
| Функціональна придатність | + | + |  |  |
| Сумісність | + |  | + |  |
| Надійність |  | + |  | + |

Продовження таблиці 2.6. Відповідність метрик

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | LOC-метрики | Метрики Холстеда | Метрики Джилба | Метрика МакКейба |
| Ефективність | + | + |  | + |
| Безпека |  |  | + | + |

**»**

**ВИСНОВКИ**

У ході виконання курсової роботи були освоєні основні навички роботи з класами та їх властивостями, об’єктами, одновимірними та двовимірними масивами, функціями, рядками, циклами.

Відпрацьовані власні алгоритми та методи виконання роботи. Створена робоча програма ігрового формату, що має багато можливостей покращення та оптимізації.

Метою розробки програмного продукту було …..

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. ПРИКЛАДИ ДЖЕРЕЛ
2. ДНАОП 0.00-1.31-99. Правила охорони праці під час експлуатації електронно-обчислювальних машин. [Електронний ресурс] – Режим доступу: http://document.ua/pravila-ohoroni-praci-pid-chas-ekspluataciyi-elektronno-obch-nor7609.html
3. Маляренко В.А. Енергетика, довкілля, енергозбережен­ня: Монографія / Л.В. Лисак, Під ред. проф. В.А. Маляренка. – Харків: «Рубікон», 2004. – 368 с.
4. Роберт Лафоре. Об’єктно-орієнтоване програмування в С++: навч. посіб. Сан-Франциско: 2013, 928 с.

Приклад розставлення посилань в тексті:

«Ще одна міра, що стосується стабільності модуля – міра Колофелло [34], вона може бути визначена ….. до даного класу відноситься метрика МакКлура [34].»