

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр
на тему: **«РОЗРОБКА КОМП'ЮТЕРНОЇ ГРИ ЖАНРУ ФЛЕШ НА
УВАЖНІСТЬ МОВОЮ ПРОГРАМУВАННЯ C++»**

Виконав: студент 5 курсу, групи ППЗ–52
спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Макогонюк М. О.

(прізвище та ініціали)

Керівник Залива В. В.

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Київ – 2021

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Напрямок підготовки – 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О. В.

“ _____ ” _____ 2021 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Макогонюк Максим Олегович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка комп'ютерної гри жанру флеш на уважність мовою програмування С++»

Керівник роботи: асистент кафедри ІПЗ, аспірант Залива В. В.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 12 березня 2021 року, №65.

2. Строк подання студентом роботи 01 червня 2021 року.

3. Вихідні дані до роботи: мова програмування С++, мова програмування для створення комп'ютерної гри, флеш гра, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Постановка ігрової задачі

4.2 Огляд аналогів і програмних засобів розробки комп'ютерної гри

4.3 Проектування архітектури та розробка ігрового додатку

5. Перелік графічного матеріалу

5.1 Титульний слайд.

5.2 Об'єкт, предмет та мета дослідження.

5.3 Актуальність вибраної теми.

5.4 Автоматизація роботи фітнес-клубу.

5.5 Системи аналогії.

5.6 Постановка задачі.

5.7 Програмні засоби для вирішення поставленої задачі.

5.8 UML діаграма прецедентів.

5.9 Алгоритм роботи програми .

5.10 UML діаграма компонентів.

5.11 Схема бази даних.

5.12 Результати роботи ПЗ.

5.13 Висновки.

6. Дата видачі завдання _____ 19 квітня 2021 року _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04.2021	Виконано
2	Вимоги до встановленого додатку	23.04.2021	Виконано
3	Оцінка якості тестування до систем	29.04.2021	Виконано
4	Концепція та архітектура додатку	02.05.2021	Виконано
5	Вступ, висновки, реферат	05.05.2021	Виконано
6	Розробка презентаційних матеріалів	06.05.2021	Виконано
7	Попередній захист роботи	11.05.2021	
8	Здача роботи	01.06.2021	

Студент _____ Макогонюк М. О.

(підпис)

(прізвище та ініціали)

Керівник роботи _____ Залива В. В.

(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 56 с., 26 рис., 13 джерел.

Об'єкт дослідження – дослідження використання флеш ігор у повсякденному житті.

Предмет дослідження – комп'ютерна флеш гра «Маріо» жанру «Аркади» створена на мові програмування C++.

Метою роботи є розробка гри «Mario» з гнучкими налаштуваннями, в жанрі «Флеш» на платформі персонального комп'ютера під управлінням операційної системи Windows. Жанр «Флеш» був обраний з метою розвитку моторики і реакції користувача.

Актуальність роботи полягає у тому, що ігрова індустрія продовжує розвиватися і з кожним роком її прибуток на ринку збільшується, як в світі, так і, зокрема, в Україні. Таким чином, ідея створення цікавої гри є перспективним напрямком.

Для досягнення даної мети мають бути вирішені наступні **завдання**:

- 1) здійснити постановку ігрової задачі;
- 2) виконати огляд аналогів і програмних засобів розробки комп'ютерної гри;
- 3) спроектувати додаток;
- 4) реалізувати і скласти документацію (керівництво користувача) на додаток.

Стислий опис результатів дослідження: реалізовано веб-додаток та базу даних, які пов'язані у систему для ведення обліку наукового складу кафедри.

Галузь використання – користувачі комп'ютерів, віком від 6 років.

Ключові слова: ФЛЕШ ІГРИ, ІГРИ НА УВАЖНІСТЬ, АРКАДИ, ІГРОВА ІНДУСТРІЯ, МАРІО, C++, БІБЛІОТЕКИ.

ЗМІСТ

ВСТУП	9
РОЗРОБКА МУЛЬТИМЕДІЙНОГО ІГРОВОГО ДОДАТКУ	11
РОЗДІЛ 1. ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЇ ТА ОПЕРАЦІЙНОГО СЕРЕДОВИЩА	11
1.1. Ігрова індустрія	11
1.2. Вибір платформи	12
1.3. Вибір жанру гри	14
1.4. Вибір програмної складової додатку	16
Висновки до розділу 1	22
РОЗДІЛ 2. ОГЛЯД АНАЛОГІВ	23
2.1. Super Mario Bros.	23
2.2. The Tomb	24
2.3. Fairy	25
Висновки до розділу 2	26
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ	27
3.1. Розробка концептуальної схеми	27
3.1.1. Формування цілей и завдання проекту	27
3.1.2. Розробка UML- діаграм	27
3.1.3. Розробка бізнес сутностей і логіки	29
3.2. Розробка гри з використанням стандартних об'єктів	29
3.2.1. Створення Scene и View	29
3.2.2. Графічне представлення в бібліотеках.	29
3.2.3. Обробка колізій	30
3.2.4. Створення таймерів	31
3.3. Завантаження текстур, музики	31
3.4. Розробка типів пересування	31
Висновки до розділу III.....	32
РОЗДІЛ 4. КЕРІВНИЦТВО КОРИСТУВАЧА	33

4.1. Загальний опис	33
ВИСНОВКИ	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	42
ДОДАТОК А. ВИХІДНИЙ КОД	44
ДОДАТОК Б. ПРЕЗЕНТАЦІЯ	54

ВСТУП

Актуальність теми. IT-спеціаліст – широке поняття, що об'єднує в собі представників багатьох професій, які працюють в області інформаційних технологій. Це програмісти, розробники, адміністратори мереж і баз даних, модератори, фахівці з робототехніки, з інформаційної безпеки, web-дизайнери і 3D-аніматори. При цьому, з проникненням інформаційних технологій в усі нові сфери діяльності, з'являються нові професії для IT-фахівців. За даними на сьогоднішній день і думку багатьох аналітиків фахівці даної області є затребуваними і будуть затребувані в найближчому майбутньому. Зокрема, сьогодні в світі з'являються багато перспективних ігрових розробок і проєктів.

GameDev (вимовляється «геймдев») — скорочення від Game Development (розробка ігор). У процес розробки ігрового додатку входить: розробка дизайну ігрового процесу (геймплея), програмування ігрового движка або використання готових рішень, розробка візуального концепту і його складових, створення графіки і моделювання об'єктів, створення музичного та звукового супроводу.

Ігрова індустрія продовжує розвиватися і з кожним роком її прибуток на ринку збільшується, як в світі, так і, зокрема, в Україні. Таким чином, ідея створення цікавої та нової гри є перспективним напрямком.

Метою роботи є розробка гри «Магію» з гнучкими налаштуваннями, в жанрі «Флеш» на платформі персонального комп'ютера під управлінням операційної системи Windows. Жанр «Флеш» був обраний з метою розвитку моторики і реакції користувача.

Для досягнення даної мети мають бути вирішені наступні **завдання**:

- 5) здійснити постановку ігрової задачі;
- 6) виконати огляд аналогів і програмних засобів розробки комп'ютерної гри;
- 7) спроектувати додаток;
- 8) реалізувати і скласти документацію (керівництво користувача) на додаток.

Структура і обсяг роботи. Робота складається зі вступу, чотирьох розділів, висновків і списку використаних джерел. Обсяг роботи становить 56 сторінок, обсяг бібліографії – 13 джерел.

Перший розділ містить аналіз і обґрунтування вибору індустрії, платформи, операційної системи та інших технічних складових, а також огляд програмних засобів розробки комп'ютерних ігор.

У другому розділі проводиться огляд ігрових аналогів. Проводиться їх аналіз і складається таблиця особливостей кожного з програмних продуктів. Виявляються їх недоліки та формується висновок.

У третьому розділі представлено опис ігрової задачі. Визначення функціональних вимог до розроблюваної відеогри. Представляються діаграми варіантів використання, короткий опис реалізації основних технічних складових додатку, а також файлова структура додатку.

Четвертий розділ присвячений демонстрації розробленого ігрового додатку. Містить в собі короткий посібник користувача.

Додаток А містить в собі код програми головного меню (вихідний, заголовочний і файл форми).

Додаток Б містить в собі презентацію.

РОЗРОБКА МУЛЬТИМЕДІЙНОГО ІГРОВОГО ДОДАТКУ

В результаті випускної кваліфікаційної роботи потрібно створити гру в жанрі «Аркада». Гра повинна бути зроблена в 2D-стилі. При запуску ми побачимо знайомий багатьом, дизайн відомої гри Маріо з деякими відмінностями. Як видно з назви головною дійовою особою гри є сантехнік Маріо.

Мета гри — пройти через всі локації та вціліти під час зустрічі з ворогами — слугами Боузера. Ігровий процес складається з переміщення в право та вліво, та з уникнення перешкод, таких як вороги чи частини оточення, наприклад блоки. На перший погляд такий ігровий процес або як його ще називають

геймплей може здаватися нудним та не цікавим але через можливість збільшувати складність та швидкість гри вона стає дуже цікавою та хардкорною.

У першій грі, де з'явився цей персонаж, його звали Джампмен (англ. Jumpman). Проте через схожість орендодавця головного офісу представництва компанії Nintendo в США на ім'я Маріо Сегал, персонаж отримав нове ім'я — Маріо.

1 ОБҐРУНТУВАННЯ ВИБОРУ ТЕХНОЛОГІЇ ТА ОПЕРАЦІЙНОГО СЕРЕДОВИЩА

1.1 Ігрова індустрія

На сьогоднішній день ігрова індустрія – це один з найзнаменитіших сегментів цифрового контенту в світі. Дана індустрія постійно розвивається і росте (рис. 1.1.1). Так, за підсумками 2020 року – світовий ігровий ринок склав близько 144 мільярдів доларів.

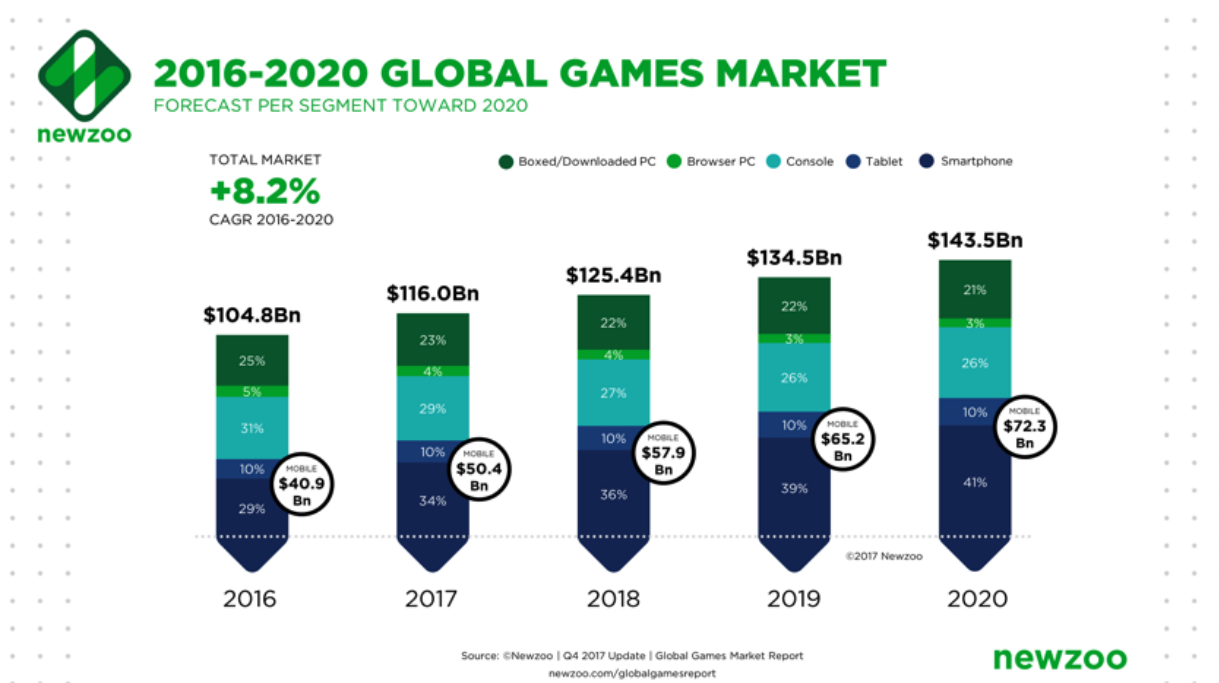


Рис. 1.1.1 – Світовий ринок ігор у 2016–2020 рр.

За результатами останнього десятиріччя, ігровий ринок в Україні є значущим і розвиваються напрямком. При цьому, в Україні обсяг ігрового ринку продовжує збільшуватися з 2010 року (рис. 1.1.2). Ринок ігор в Україні досі є інвестиційно привабливим.

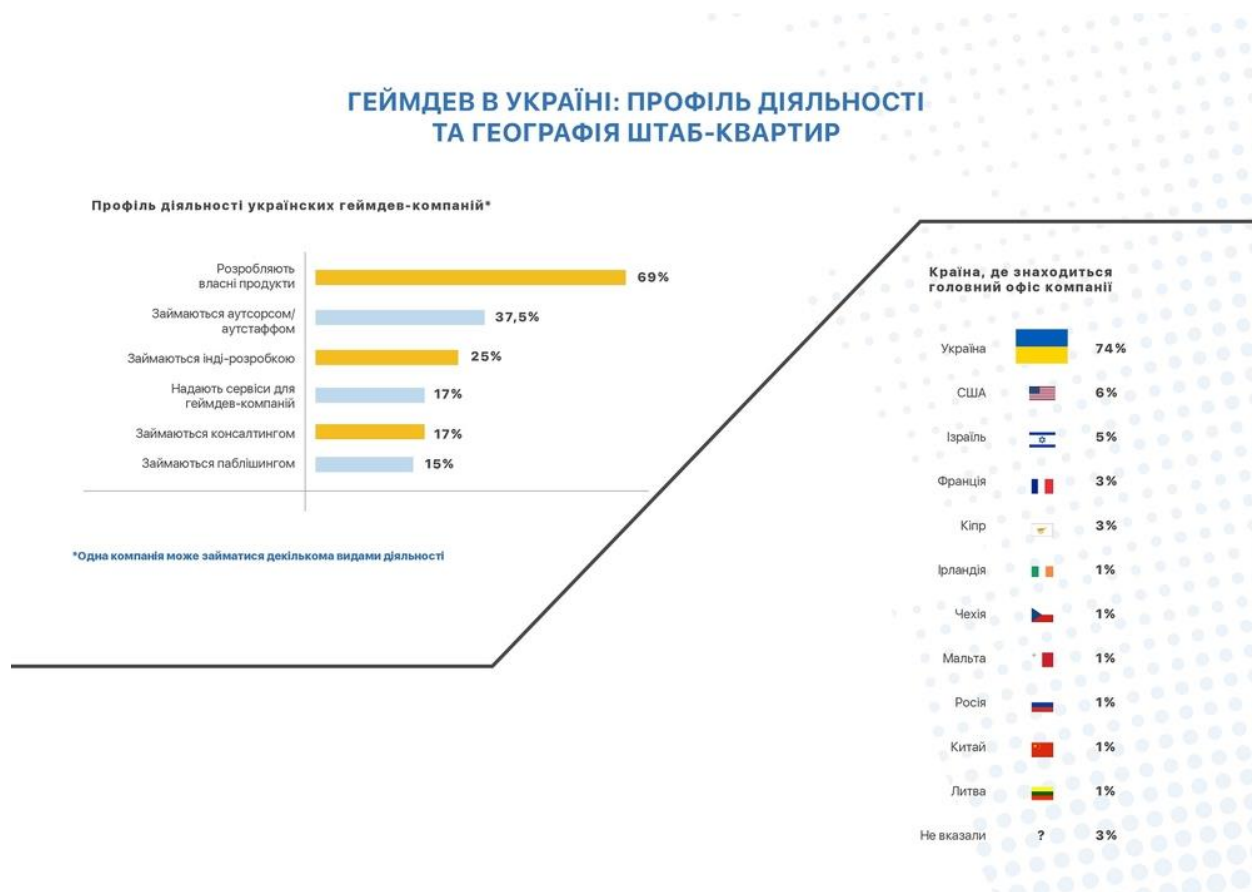


Рис. 1.1.2 – Україна на світовому ринку ігор

Український ринок лише недавно став розвиватися в даному напрямку. Проте вже сьогодні тут знаходиться 74% головних офісів компаній усього світового ринку ігор.

1.2 Вибір платформи

На сьогоднішній день можна виділити три ігрові платформи:

- персональні комп'ютери;
- мобільні пристрої;
- консолі.

Зародження ігор почалося з появи комп'ютера, тому користувачі ПК є основною ігровою аудиторією. Так, за даними 2019 року, найбільш популярними є ігри розроблені для Android, IOS, PC/Mac. (рис 1.2.1.).

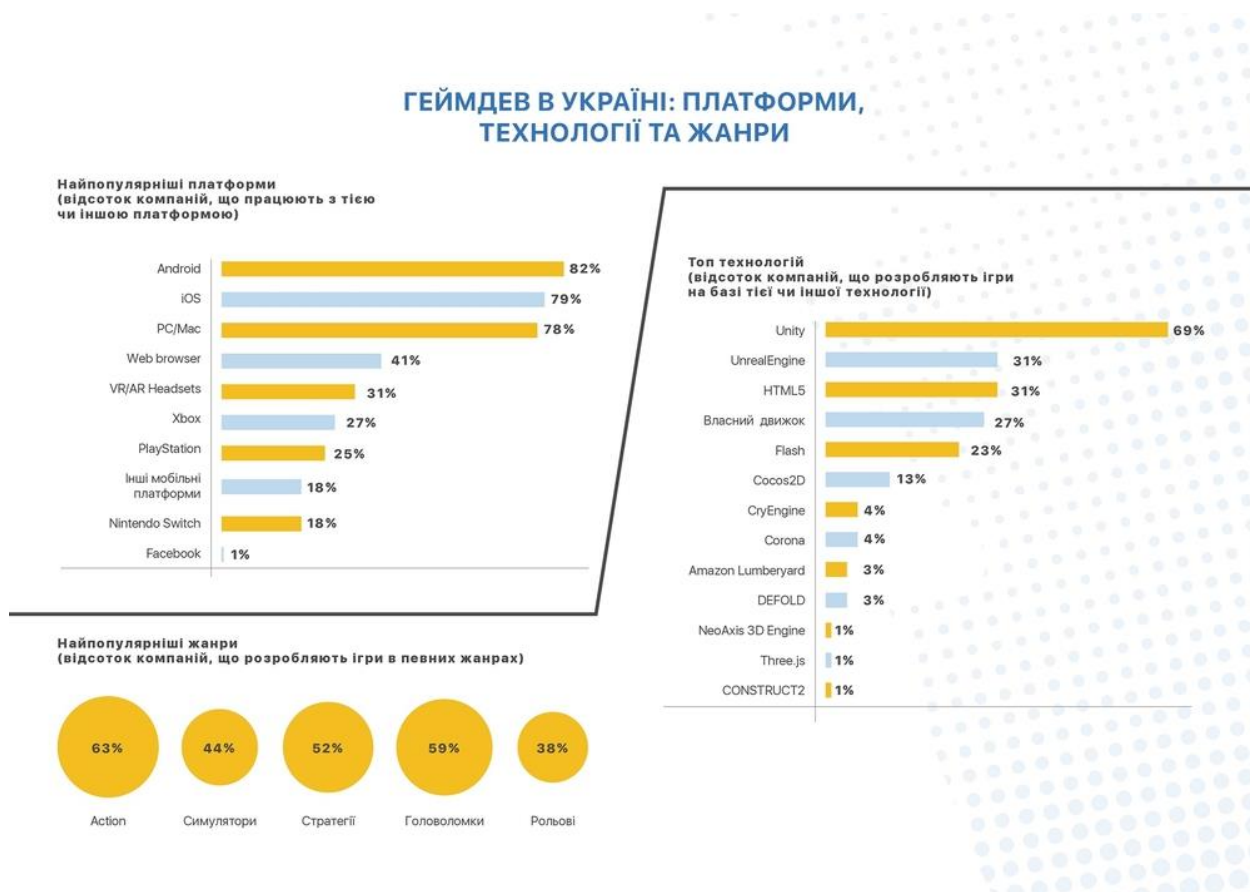


Рис. 1.2.1 – Ігровий ринок на різних платформах, технологіях, жанрах у 2019 році

Під час аналізу ринку персональних комп'ютерів був проведений приблизний підрахунок величини операційних систем, що використовуються на них. Результати аналізу представлені на рисунку 1.2.2.

Із рисунка 1.2.2 видно, що більшість користувачів персональних комп'ютерів надають перевагу операційній системі сімейства Windows.



Рис. 1.2.2 – Операційні системи, що використовуються

1.3 Вибір жанру гри

Сьогодні в світі переважно виділяють наступні жанри ігор:

- азартні
- спортивні
- головоломки
- стратегії
- бойовики
- аркади

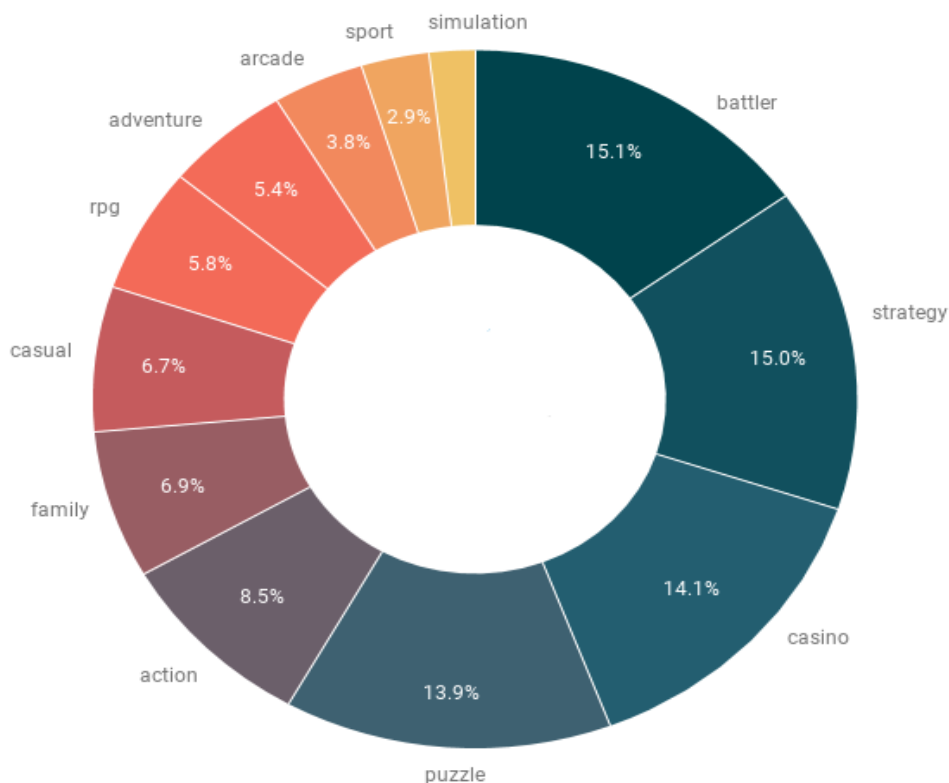


Рис. 1.3 – Популярність ігрових жанрів

За даними компанії «Google» багато користувачів надають перевагу жанру «Аркада». Аркади є досить популярними відносно інших жанрів з наступних причин (рис 1.3):

1. Аркади не вимагають потужних технічних характеристик

Не всі користувачі мають можливість купувати сучасні і потужні пристрої з відповідними характеристиками. Багато користувачів відзначають, що чим потужніша гра, тим більше вона гальмує. Таким чином, ігровий процес є досить складним і неприємним для користувача, що відбиває бажання грати і, тим самим, зменшує рейтинг додатку, що, в свою чергу, зменшує зацікавленість нових користувачів (гравців) встановлювати цю програму. Однак аркади не вимагають серйозних технічних характеристик, що дозволяє більшості користувачів насолоджуватися ігровим процесом без затримок, зависань і інших проблем, пов'язаних з продуктивністю.

2. Аркади не займають багато часу

Всі користувачі ігрових додатків (гравці) займаються певною діяльністю (працюють, навчаються і т.д.). Таким чином, у гравців не завжди є вільний час, щоб проводити його за іграми. Аркади створені для людей, у яких немає можливості проводити за грою достатню кількість часу. Зазвичай даний жанр дозволяє гравцеві зайняти його час на якийсь період. Тим самим, більшість користувачів називають аркади «ТаймКіллерами», так як вони є цікавими і незамінними помічниками весело провести період часу.

3. Аркади цікаві

Багато користувачів відзначають, що жанр аркада різнобічний. Він може включати в себе логічні загадки, красиву графіку або цікавий ігровий процес (геймплей), тобто не дозволяють гравцеві занудьгувати.

1.4 Вибір програмної складової додатку

В даний час існує безліч програмного забезпечення, що дозволяє розробляти власні проекти різного характеру, в тому числі і комп'ютерних ігрових додатків. Виділяють дві основні групи з них:

- використання програмного компоненту (комплексу), що дозволяє створювати ігровий додаток (ігровий движок);
- створення власного «движку» за допомогою мови програмування і спеціалізованих бібліотек.

Розглянемо кожен з груп детальніше:

1) ігрові «движки»;

Unity 3D — це потужний інструмент для створення додатків та ігор під різні платформи. Є можливість створювати як 2D, так і 3D проекти. Є підтримка DirectX і OpenGL, підтримує безліч різних форматів даних. Unity 3D підтримує такі мови програмування як: C #, JavaScript. Основне робоче вікно Unity 3D представлено на рисунку 1.4.1.

Переваги:

- не найбільші витрати ресурсів (у порівнянні з іншими «движками»);

- кросплатформність;
- регулярні оновлення.

Недоліки:

- володіє малим набором інструментів;
- вимагає пристойні технічні комплектуючі;
- платна комерційна ліцензія;
- незручна робота при створенні 2D – додатку.

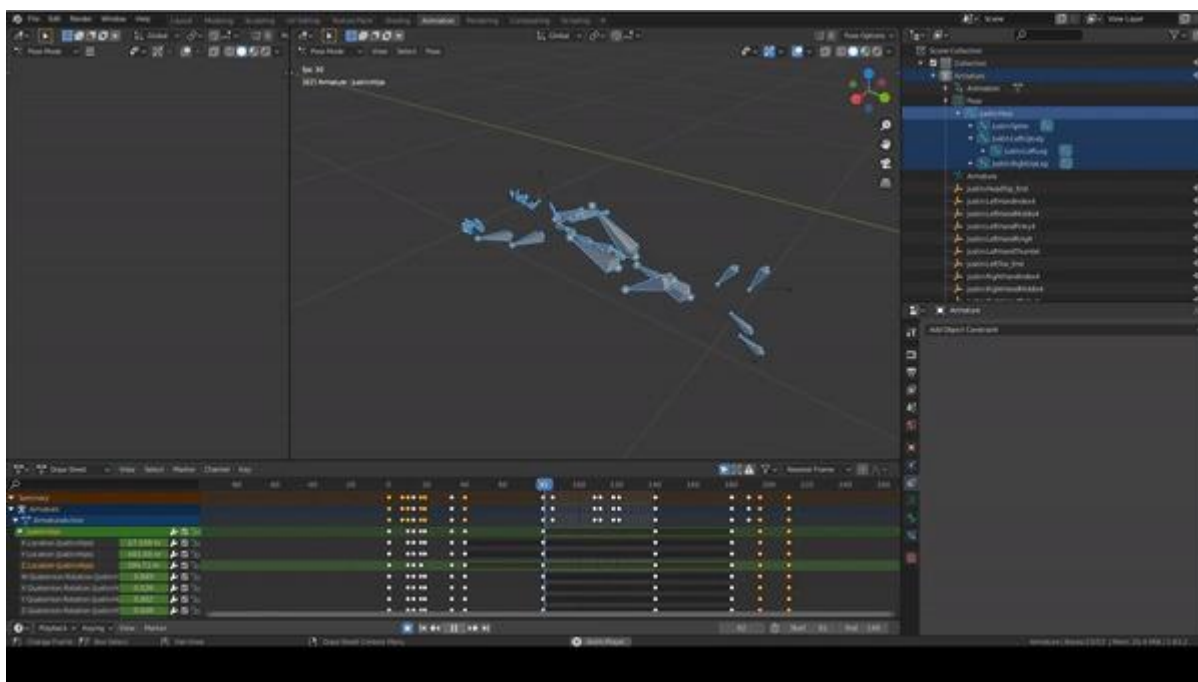


Рис. 1.4.1 – Робоче вікно ігрового движка Unity 3D

CryEngine — найпотужніший із сучасних ігрових движків, що забезпечує фотореалістичну графіку з підтримкою DirectX 12 і шейдерів (тіней). Третя версія «движка» створена в 2009 році, поширюється платно. П'ята версія «движка» вийшла в кінці 2016 року і має умовно безкоштовну ліцензію. Основним напрямком даного засобу є створення ігор жанру «Шутер». Робоче вікно CryEngine представлено на рисунку 1.4.2.

Переваги:

- високі показники графіки;
- кросплатформність.

Недоліки:

- володіє малим набором інструментів;
- спрямований на створення певного жанру ігор;
- самий ресурсномісткий «движок» з усіх представлених.

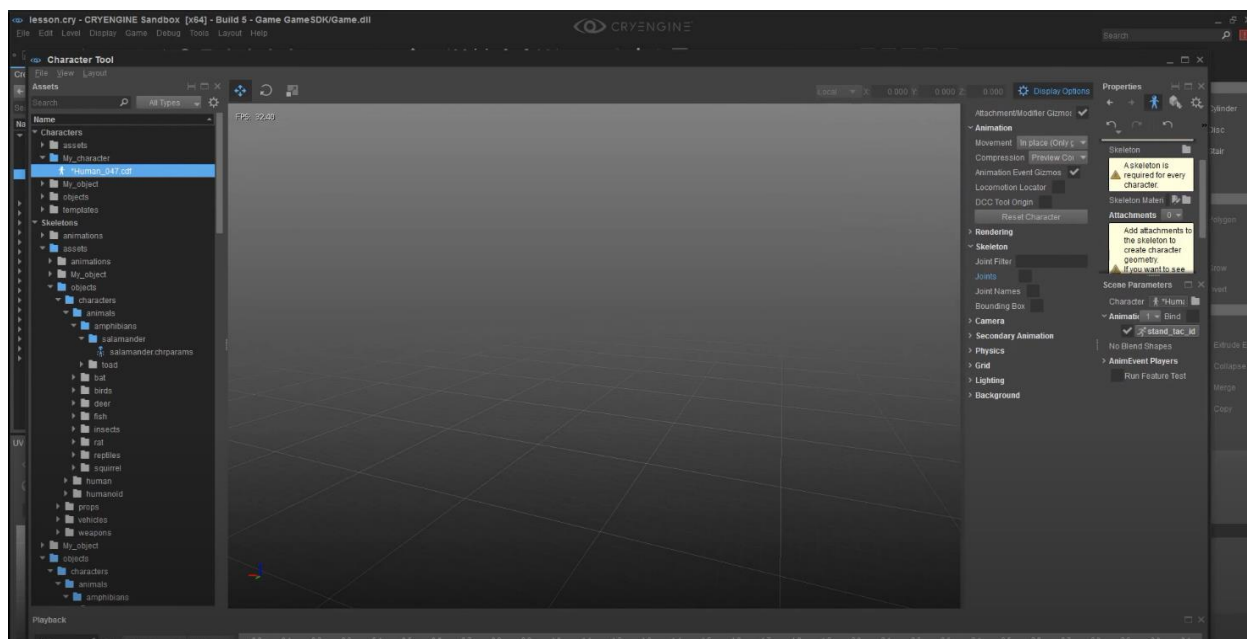


Рис. 1.4.2 – Робоче вікно ігрового движка CryEngine

Unreal Engine — ігровий «движок», що розробляється і підтримується компанією Epic Games. Надає великий набір інструментарію для створення 2D і 3D проєктів. Починаючи з четвертої версії — поширюється безкоштовно. Проте до тих пір, поки розробник не випустить свій перший комерційний продукт на основі UE4. Є головним движком більшості сучасних ігор. Робоче вікно ігрового «движка» Unreal Engine представлено на рисунку 1.4.3.

Переваги:

- високі показники графіки;
- кросплатформність;
- великі і регулярні оновлення;
- різноманітний інструментарій.

Недоліки:

- високі технічні вимоги;

- складність в освоєнні;
- мала кількість російськомовної документації;
- комерційна ліцензія.

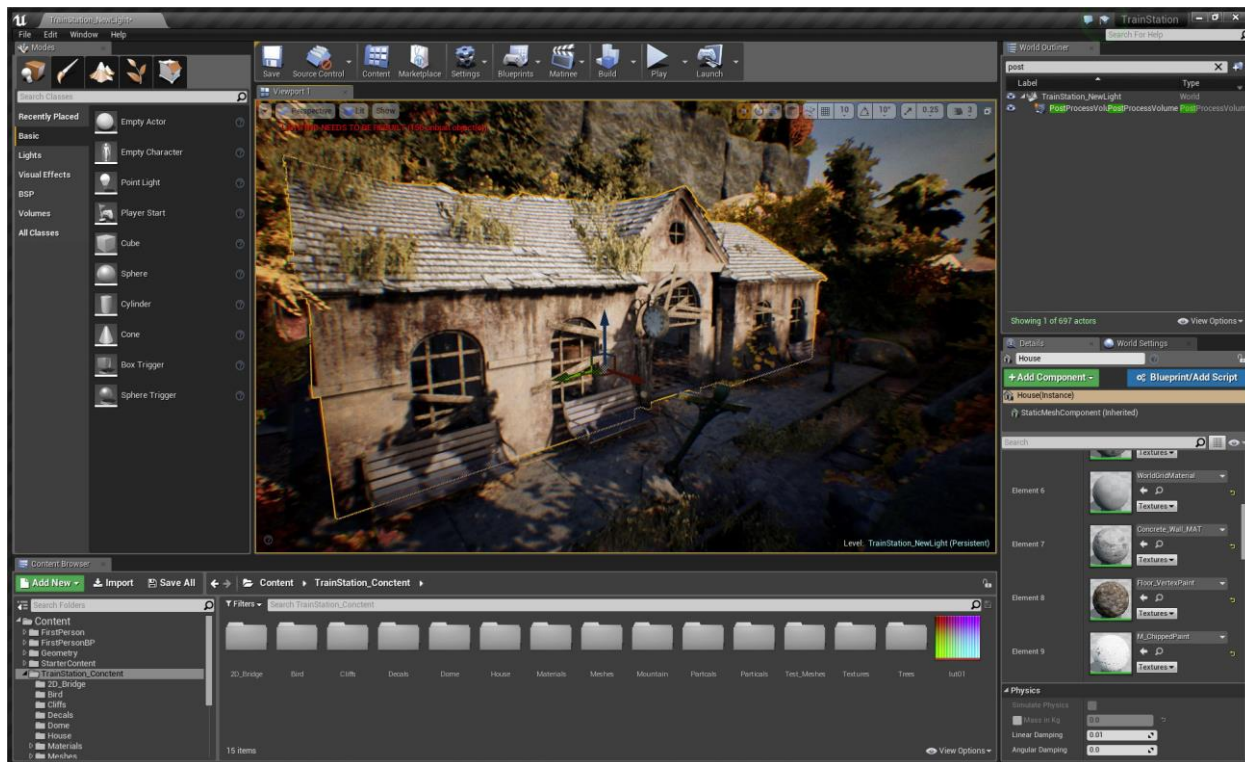


Рис. 1.4.3 – Робоче вікно ігрового движка Unreal Engine

Всі розглянуті ігрові «движки» в основному призначені для створення 3D додатків і вимагають високі показники технічних вимог, тому використовувати їх недоцільно (у багатьох користувачів ПК відсутні сучасні і потужні комплектуючі). Перейдемо до розгляду іншої групи програмного забезпечення для створення ігрового додатку.

2) бібліотеки для мов програмування (API).

Для початку варто визначитися, яку мову програмування використовувати. На сьогоднішній день виділяють наступні мови програмування як основи подальшої розробки:

1. C#

Дозволяє стартувати розробку швидше, а це дозволяє швидше отримати

прототип рішення. Швидкість розробки на початкових етапах проекту значно вище в порівнянні з іншими мовами.

C# спроектований бути кросплатформним, однак його розвиток не пішов в цьому напрямку, тому під Windows утворилася досить повна .net інфраструктура, а на інших платформах рівноцінної інфраструктури не з'явилося.

При розробці невеликих проектів продуктивність C # не поступається іншим мовам програмування, однак при збільшенні вихідного коду, алгоритмів і т.д. – швидкість роботи додатків значно падає.

C# володіє пристойною кількістю бібліотек зі старту, що істотно полегшує розробку.

2. HTML и JavaScript

Є незамінними при створенні простих веб-додатків та ігор, проте на цьому переваги даних засобів закінчуються.

3. C++

Є основною мовою програмування при розробці ігор, оскільки дозволяє здійснити повний контроль над засобами і логікою. Є абсолютним рекордсменом за продуктивністю, за кросплатформністю, за сумісністю і за кількістю існуючих бібліотек серед інших мов програмування. Однак багато програмістів виділяють головний його недолік – складність освоєння даної мови.

Зведемо отримані дані в таблицю 1.3.

Таблиця 1.3 – Порівняння мов програмування

Критерій Мова програмування	C#	JavaScript/HTML	C++
Продуктивність	★★★★☆	★★☆☆☆	★★★★★ (максимальна)
Кіл-ть бібліотек	★★★★☆	★★☆☆☆	★★★★★
Зручність програмування	★★★★☆ (легка)	★★☆☆☆	★★☆☆☆
Складність освоєння	★★★★☆ (легка)	★★★★☆	★★☆☆☆
Кросплатформність	★★★★☆	★★★★★ (повна)	★★★★☆

Кожна мова програмування добре себе показує в певних завданнях. Наприклад: веб-розробка (JavaScript) не має прихильності до операційної системи; якщо в пріоритеті розробки є швидкість (отримання прототипу додатку), то слід використовуватися C#. В якості мови програмування для створення ігор було вирішено використовувати мову C++, оскільки вона підтримує величезну кількість бібліотек і має гарну продуктивність.

На мові C++ існує два основних API для створення графіки:

1) SFML

Проста і кросплатформна мультимедійна бібліотека. Включає в себе модулі для програмування ігор і мультимедійних засобів. Складається з п'яти модулів:

1. System – модуль управляє часом і потоками. Є базовим, тобто необхідний для всіх модулів

2. Window – управління вікнами и вивід інформації користувачеві.

3. Graphics – виробляє висновок інформації (зображень, геометричних фігур). Для використання потрібно підключити модуль Window.

4. Audio – бібліотека для роботи зі звуком.

Саме вона була обрана як середовище розробки проекту. Через її простоту та зручність в розробці та менеджменті проекту.

2) Qt Graphics

Більш потужний інструмент (у порівнянні з SFML). Входить до складу бібліотеки Qt, яка вдає із себе інструментарій розробки ПЗ на мові програмування C++ і володіє кросплатформністю. Qt надає програмісту не тільки зручний набір бібліотек класів, а й певну модель розробки додатків, певний каркас їх структури, істотно знижує появу помилок в додаток (витоку пам'яті, незакриті файли і т.д.). Бібліотека Qt володіє власною IDE (середовищем розробки) – Qt Creator, що істотно спрощує розробку додатків. Оболонка Qt Creator представлена на малюнку 2.4.4.

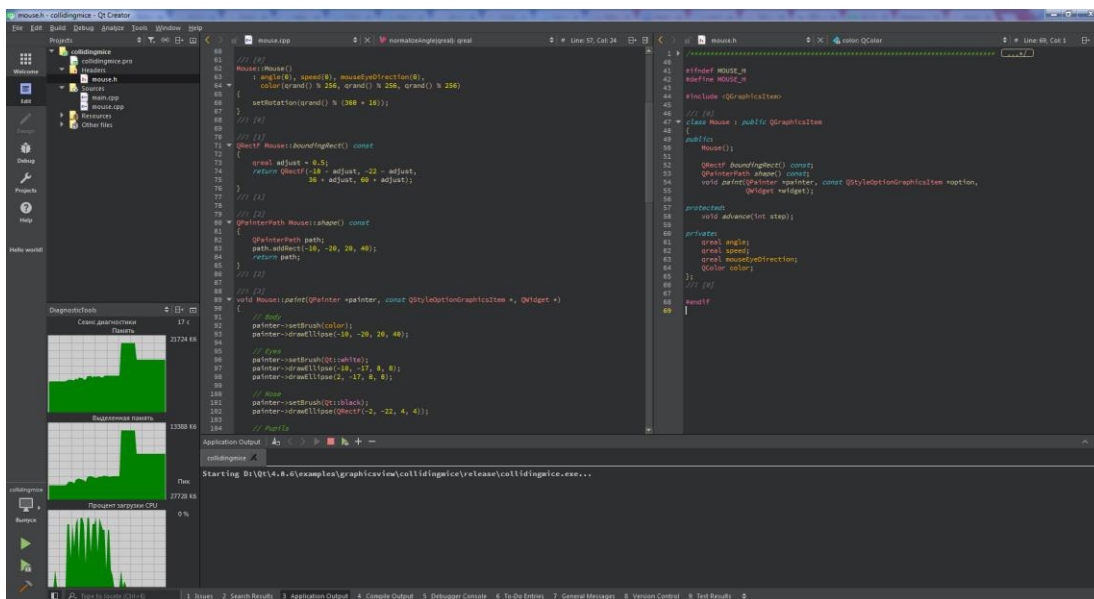


Рис. 2.4.4 – IDE Qt Creator

Висновки до розділу 1. Таким чином, можна вивести основні вимоги до гри:

1. платформа створення – Персональний комп'ютер;
2. операційне середовище – Windows 7 і вище;
3. жанр гри – аркада;
4. використовувана мова програмування – C++;
5. використовувана API – SFML ;

2 ОГЛЯД АНАЛОГІВ

На сьогоднішній день існує безліч схожих ігрових додатків жанру аркадний платформер. Розглянемо найпоширеніші з них.

2.1 Super Mario Bros

Найпопулярнішою подібною грою цьому жанрі є Super Mario Bros. (рис. 2.1.1). Мета гри — врятувати принцесу. Метою кожного окремого рівня було досягти флагштоку з прапором щоб перейти на новий рівень. (рис. 2.1.2).

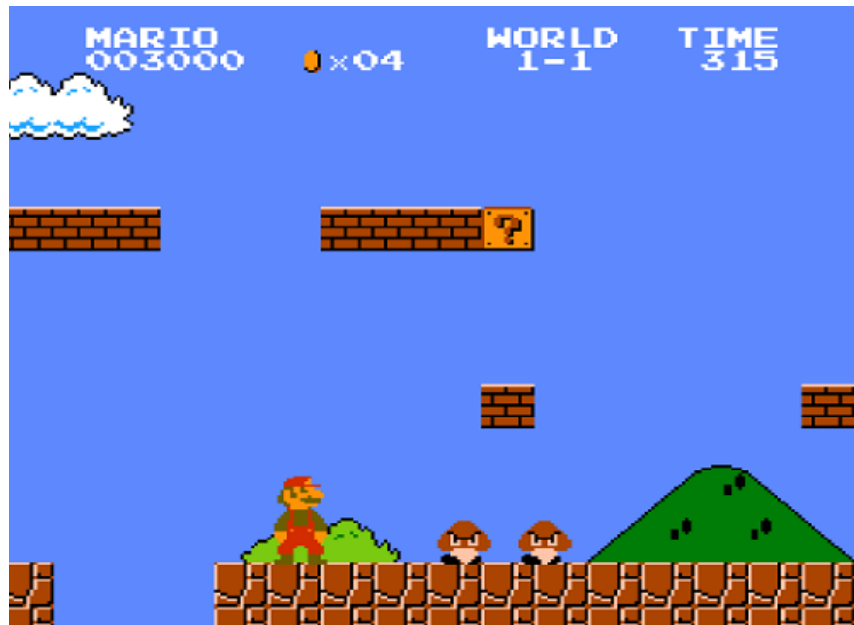


Рис. 2.1.1. - Скріншот гри Super Mario Bros.

Недоліки:

- користувач не має можливості змінювати ігрові моделі;
- користувач не має можливості змінювати логіку роботи гри (зміна складності рівня, зміна траєкторії пересування ворогів);
- гра має примітивні моделі.

Переваги:

- незвичайний сетинг;
- присутнє наростання складності в міру проходження.

Недоліки:

- в грі присутні тільки стандартні однотипні моделі;
- складність присутня, однак, у користувача немає можливості вибрати її самому;

2.3 Fairy

«Fairy» — Це аркадний хоррор-платформер в жанрі хоррор з виглядом збоку. Гравцю потрібно відправитися в старий маєток, в якому перебуває потойбічне зло. Пробиваючись через нечисть, гравцю потрібно вижити і знайти вихід. У грі присутня адаптивна складність (складність збільшується з рівнем).



Рисунок 2.3 – Скріншот гри «Fairy»

Переваги:

- присутнє наростання складності під час проходження.

Недоліки:

- складність є, однак у користувача немає можливості вибрати її самому.

Висновки до розділу 2. У результаті огляду вищеперерахованих ігрових рішень були виділені їх функціональні складові і винесені в таблицю 2.6. При аналізі розглянутих ігор були виділені наступні особливості жанру:

- аркади включають у себе або набір рівнів, або нескінченний режим гри;
- щоб збільшити показник очок потрібно знищити модель ворога або зібрати деякі речі, наприклад монети.
- майже у всіх аркадах є показник здоров'я, який зменшується при попаданні гравцем на голову суперника

Порівняння особливостей ігор жанру «Аркада»

Таблиця 2.6

Назва гри	Графіка	Сюжет	Можливість змінювати моделі	Можливість змінювати поведінку ворогів
Super Mario Bros.	2D	Набір місій	-	-/+ Присутнє збільшення складності
The Tomb	2D	Набір місій	-	-/+ Присутнє збільшення складності
Fairy	2D	Набір місій	-	-/+ Присутнє збільшення складності

З таблиці 2.6 видно, що більшість аркад не пропонують користувачеві можливість змінювати поведінку ворогів або відкрито задавати рівень складності, а також змінювати моделі в грі.

3 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

Розробка гри складалася з чотирьох етапів:

3.1 Розробка концептуальної схеми

3.1.1 Формування цілей і завдання проекту

На цьому етапі продумується функціонал і можливості. Формується жанр та інші особливості гри. Як жанр гри було обрано напрямок «Аркада», оскільки він досить поширений. Таким чином, на даному етапі були сформовані наступні твердження:

- гра буде представляти з себе платформер
- у грі присутні гравець, вороги;
- у грі врахована система рівнів;
- мета гри – пройти всі рівні.

3.1.2 Розробка UML-діаграм

Для проектування програми була використана мова графічного опису об'єктного моделювання UML [18], так як вона дозволяє продумати і описати архітектуру проекту без програмного коду, при цьому не вимагає особливих знань для їх розуміння. В якості основного засобу опису проекту були обрані і побудовані діаграми використання, які показують функціональність майбутньої системи. Варіанти використання показують можливі взаємини між системою і користувачем. Вони відображають зовнішній інтерфейс системи і вказують форму того, що система повинна зробити. Дійова особа (актор) – це елемент, який не є системою, а взаємодіє з нею, через особливий інструмент – варіант В ході проектування був виділений наступний актор:

Гравець – користувач додатка, що володіє повним доступом до функцій відеоігри. Може брати участь у грі, управляти налаштуваннями додатку. Загальна діаграма використання представлена на рисунку 3.1.2.1.

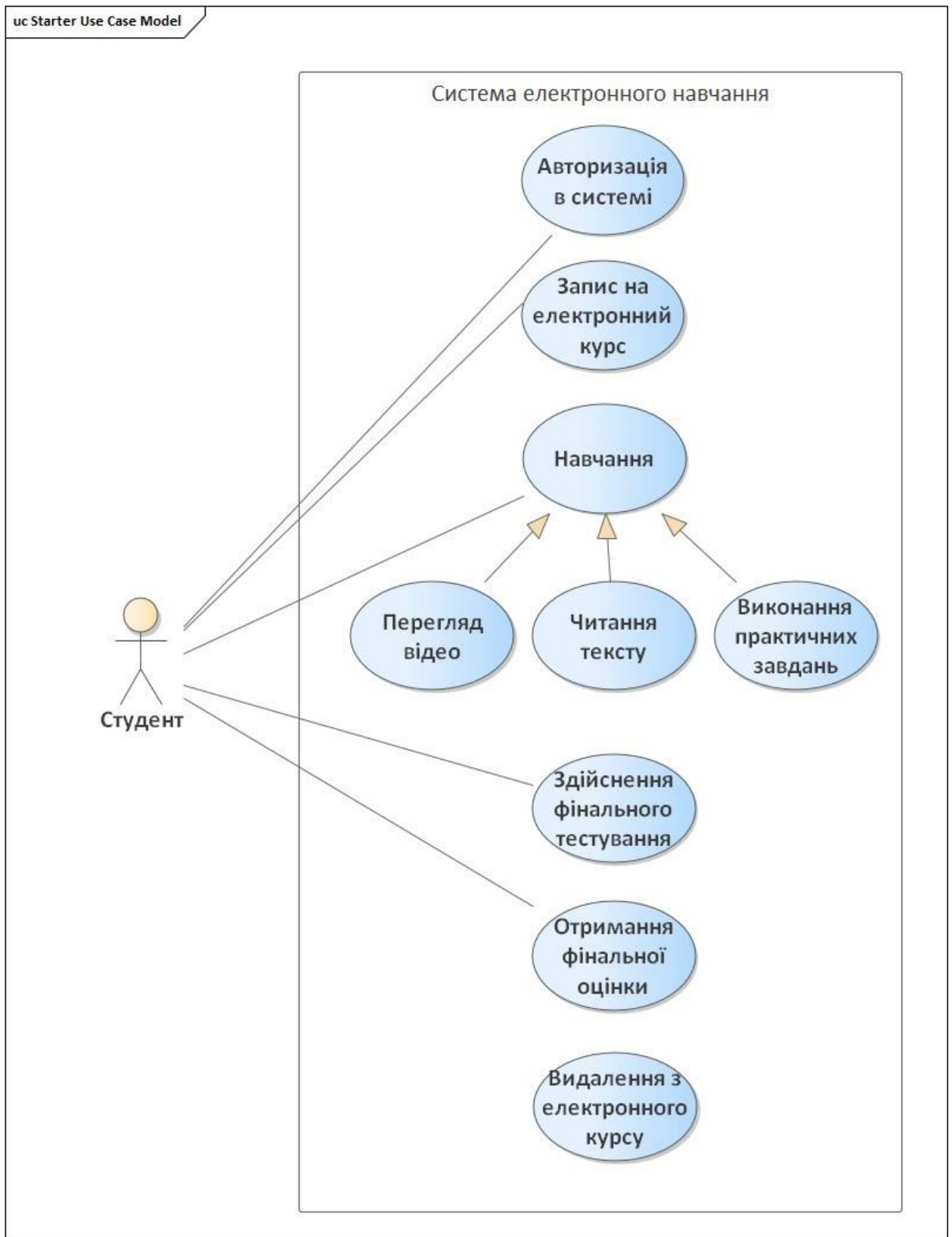


Рис. 3.1.2.1 – Загальна UML діаграма використання

3.1.3 Розробка бізнес сутностей і логіки

У результаті проектування програми, були виділені бізнес сутності і логіка, що реалізує роботу відповідної гри.

3.2 Розробка гри з використанням стандартних об'єктів

3.2.1 Створення Scene и View

Одновимірний масив `TileMap` із кількістю елементів `[H]` що означає висоту нашого рівня, створює шаблон оточення для великої кількості графічних 2D елементів для заповнення `TileMap`-ом. Для реалізації двовимірного оточення гри використовується `Tile-based graphics`, це спеціальний метод створення двовимірної графіки. Суть цього метода полягає в створенні графіки із набору однотипних зображень (`Tileset`), які утворюють своєрідну сітку заповнюючи окремі її частини необхідними текстурами, таким чином ми будемо рівень.

- `offsetX` – координата x прямокутника;
- `offsetY` – координата у прямокутника;
- `W` – ширина прямокутника;
- `H` – висота прямокутника.

Використовуючи клас `Texture` та його метод `loadFromFile` завантажуюмо `Tileset` який буде використовуватися для створення графіки. Також за допомогою методу `tile` цього ж класу, ми вказуємо який саме `Tileset` буде використовуватися грою, в нашому випадку той що ми отримали від методу `loadFromFile`.

3.2.2 Графічне представлення в бібліотеках

Графічне представлення в бібліотеках є декартовою системою координат. Таким чином, щоб змінити положення елементів і їх геометрію потрібно скористатися двома величинами: x-координати і у-координати (рисунок 3.2.2).

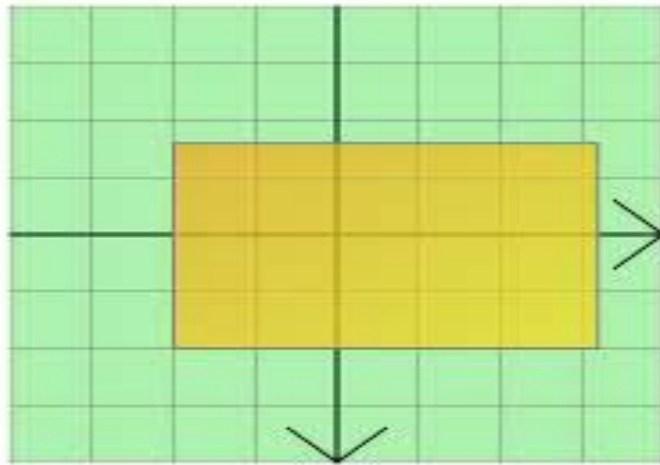


Рис. 3.2.2 – Система координат графічного представлення

В якості моделі ворога і головного персонажа будуть виступати спрайти, які представляють собою елементи Tileset-у.

Вороги повинні з'являтися у верхній частині екрану гри. Поява ворога заздалегідь відома, він має власний простий штучний інтелект.

За допомогою функції `setPosition (double x, double y)` – встановлюється позиція елемента, де:

- x - координата x встановлюваного елемента;
- y – координата y встановлюваного елемента.

3.2.3 Обробка колізій

Для реалізації колізії в грі було використано власноруч написану функцію. Стандартна функція не могла повністю виконати поставлені задачі, тому вона була замінена іншою власноруч заміненою функцією. При взаємодії (перетині) моделі ворога і гравця, модель гравця повинна почервоніти, сповіщаючи про попадання ворога по гравцю.

3.2.4 Створення таймерів

Мета гри – знищити ворогів, та дійти до кінця рівня. Для оновлення стану екрану та елементів на ньому нам треба викликати функцію Update, використовуючи таймер, ми повинні вказати періодичність оновлення екрану. наприклад void update(float time)

3.3 Завантаження текстур, музики, моделей

Ми вже завантажили TileSet в гру, тепер нам необхідно дістати спрайти, а також з інших джерел треба отримати:

- музика
- різні ігрові звуки

1) Sprites

- Enemy – стандартна модель ворога. Являє собою 2D-спрайт черепахи
- Player – стандартна модель гравця (користувача). Являє собою набір спрайтів, які утворюють анімації персонажа

- BackGround – задній фон гри. В якості фону було обрано небо.

2) Music & Sounds

- Jump – звук, який з'являється в результаті прижку головного персонажа.
- Mario_Theme – музика, яка звучить в якості фону.

3.4 Розробка типів пересування

Пересування по прямій траєкторії;

Для пересування об'єкта по прямій траєкторії слід зміщувати її з координування Y через особливості системи координат. Таким чином, щоб перемістити модель, скористаємося такою функцією:

```
setPosition (x(),y() + offset);
```

де offset – величина зсуву відносно координати Y

Висновки до розділу 3.

У результаті процесу розробки були виконані наступні завдання:

- виділені основні положення гри (тип гри, жанр тощо);
- вироблені основні цілі та завдання гри;
- продумані основні можливості та особливості гри;
- складені UML – діаграми функціональної складової програмної системи;
- вивчені технології роботи з графічними компонентами;
- для роботи зі звуковими компонентами системи були вивчені відповідні

бібліотеки;

- продумані логічні компоненти системи;
- розроблені різні типи пересування,
- були знайдені і використані безкоштовні моделі для програмної системи;
- розроблені інтерфейси для взаємодії з користувачем.

4 КЕРІВНИЦТВО КОРИСТУВАЧА

4.1 Загальний опис

«Маріо» – це легендарна гра, в якій гравцеві потрібно виживати, знищувати якомога більше ворогів і заробити якомога більше очок.

Під час запуску програми «Маріо» користувач бачить основне меню програми, представлене на рисунку 4.1.1. Взаємодія з елементами меню здійснюється за допомогою комп'ютерної миші – наведіть курсор на елемент «Почати гру» і натисніть на ліву кнопку миші.

При натисканні користувачем на кнопку «Почати гру» з'являється вікно, в якому здійснюється основний процес гри (рис. 4.1.2).

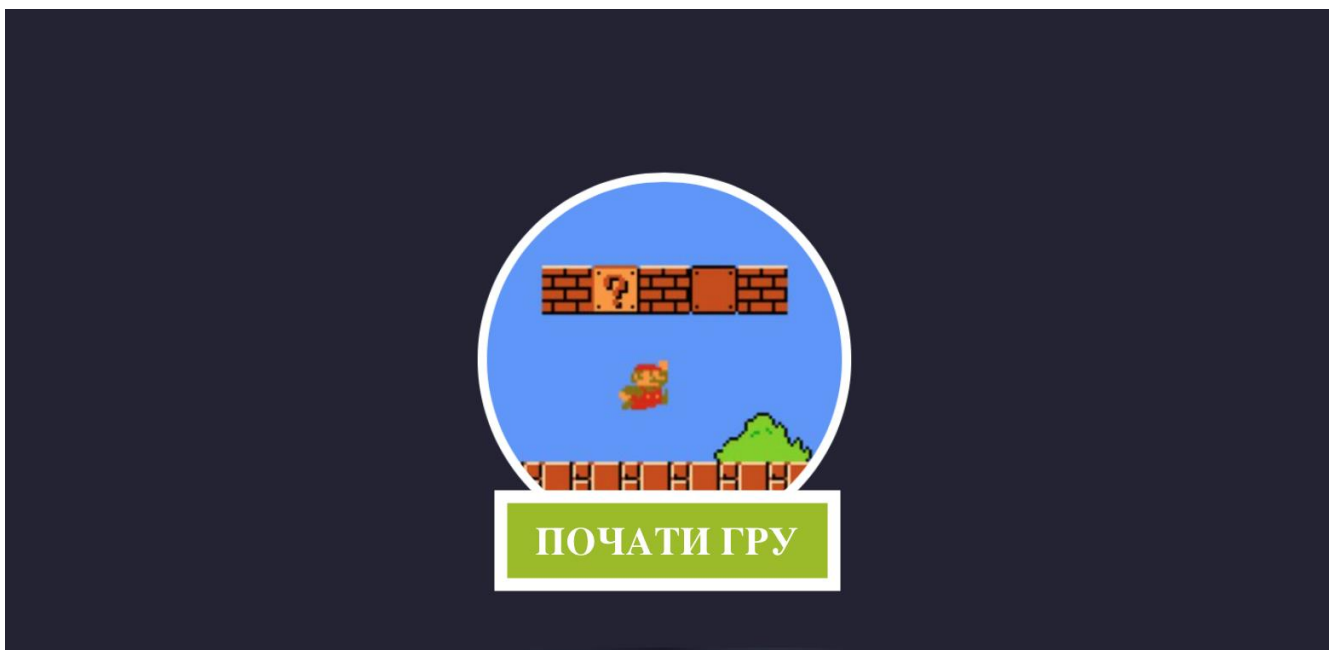


Рис. 4.1.1 – Головне меню

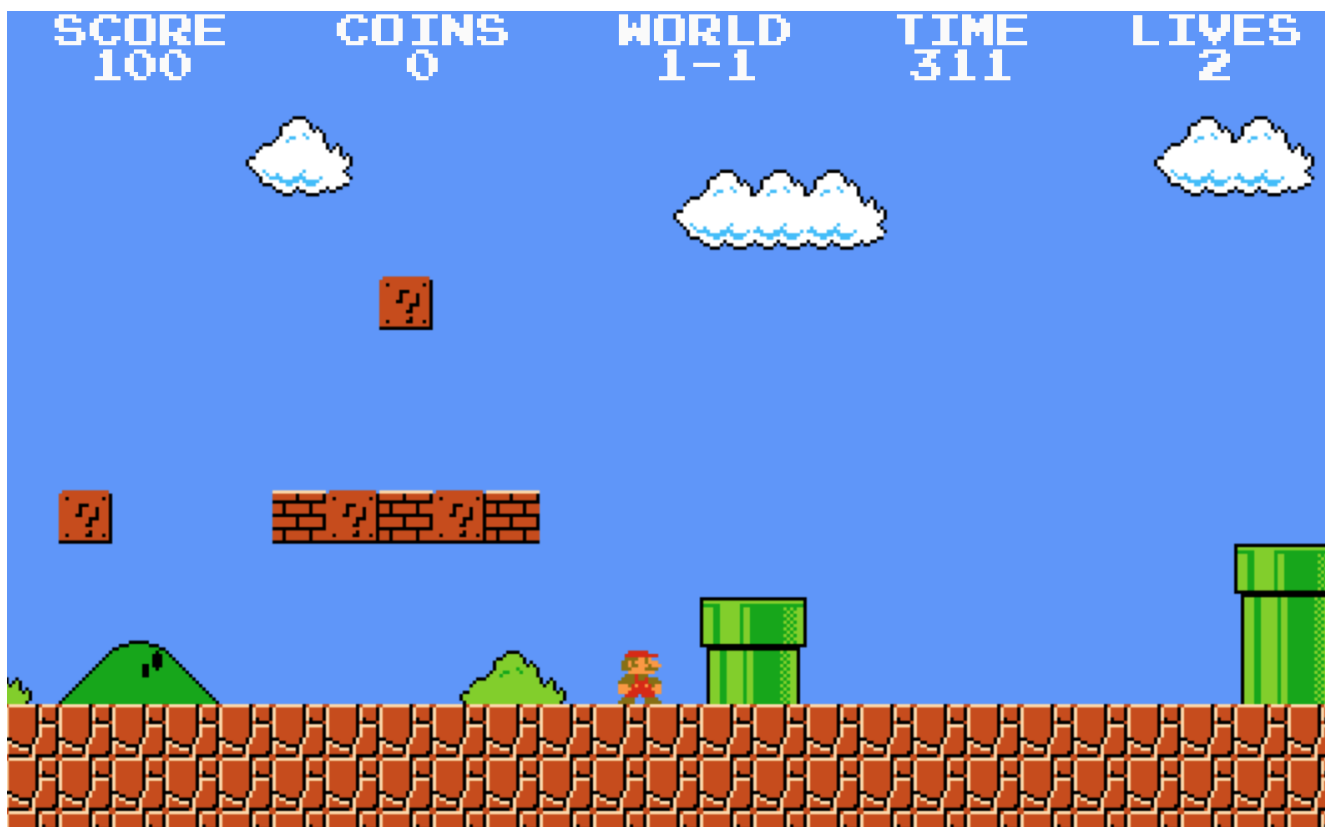


Рис. 4.1.2 – Основне вікно гри (перший рівень)

Основні елементи управління:

- клавіша «Ліворуч» – пересування моделі гравця вліво;
- клавіша «Праворуч» – пересування моделі гравця вправо;
- клавіша «Вгору» – стрибок.

У верхній частині екрану представлено п'ять показників.

- “Score” – рахунок гравця (показник очок);
- “Coins” – кількість зібраних монет;
- “World” – поточний рівень складності;
- “Times” – таймер, відлік часу на проходження рівня;
- “Lives” – показник кількості спроб (життів персонажа).

Під час проходження рівня гравцеві необхідно проходити перешкоди, які показано на рис. 4.1.3.

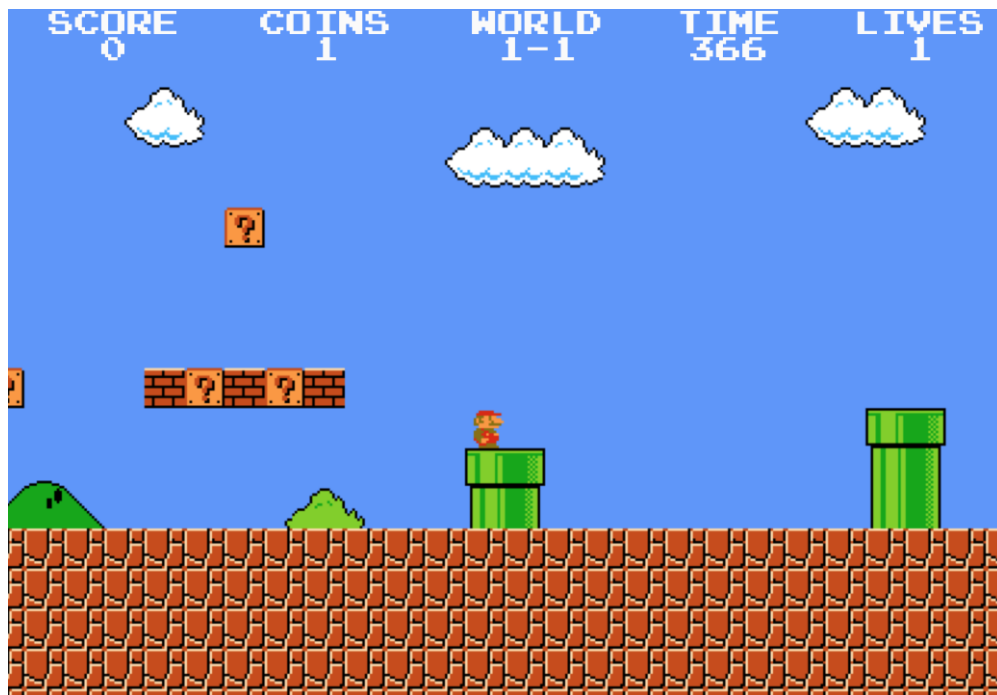


Рис. 4.1.3 – Здолання перешкоди

Щоб пройти дану перешкоду гравцеві необхідно використати «стрибок», натиснувши клавішу «вгору», анімація якого зображена на рис. 4.1.4.

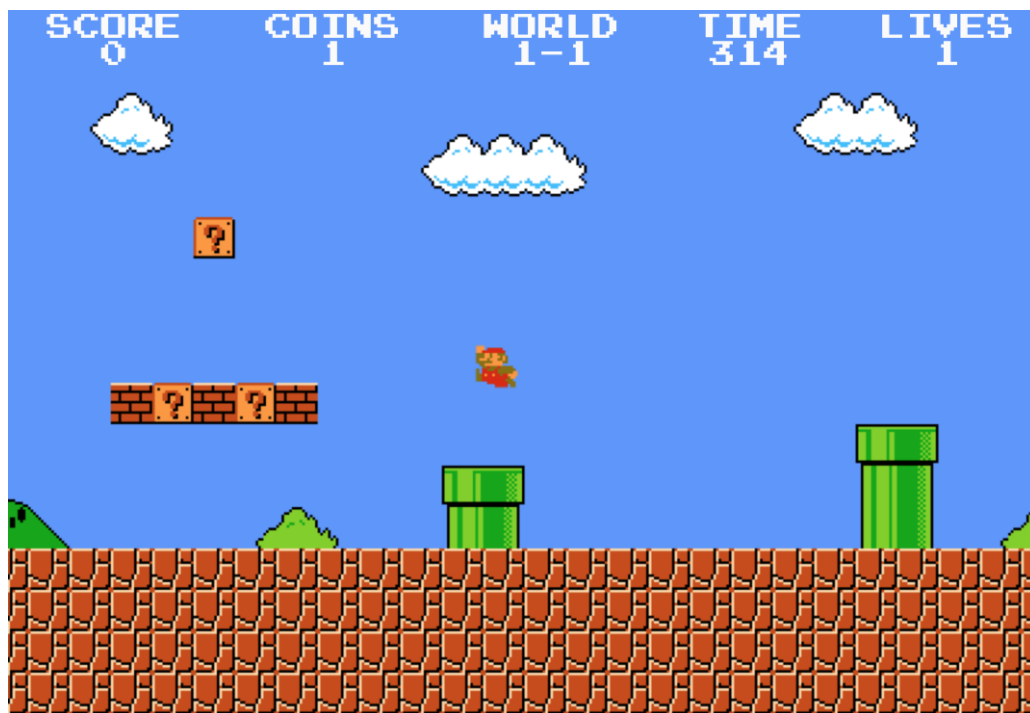


Рис. 4.1.4 – Відображення анімації виконання стрибка

Протягом всього процесу гри з'являються вороги. Зустрічі з ними необхідно уникати, інакше гравець втратить одне ігрове життя і почне проходження рівня з самого початку. Проте ворога можна здолати за допомогою стрибка і влучного попадання прямо в нього, за це користувачу будуть нараховані додаткові бали, приклад на рис. 4.1.5.

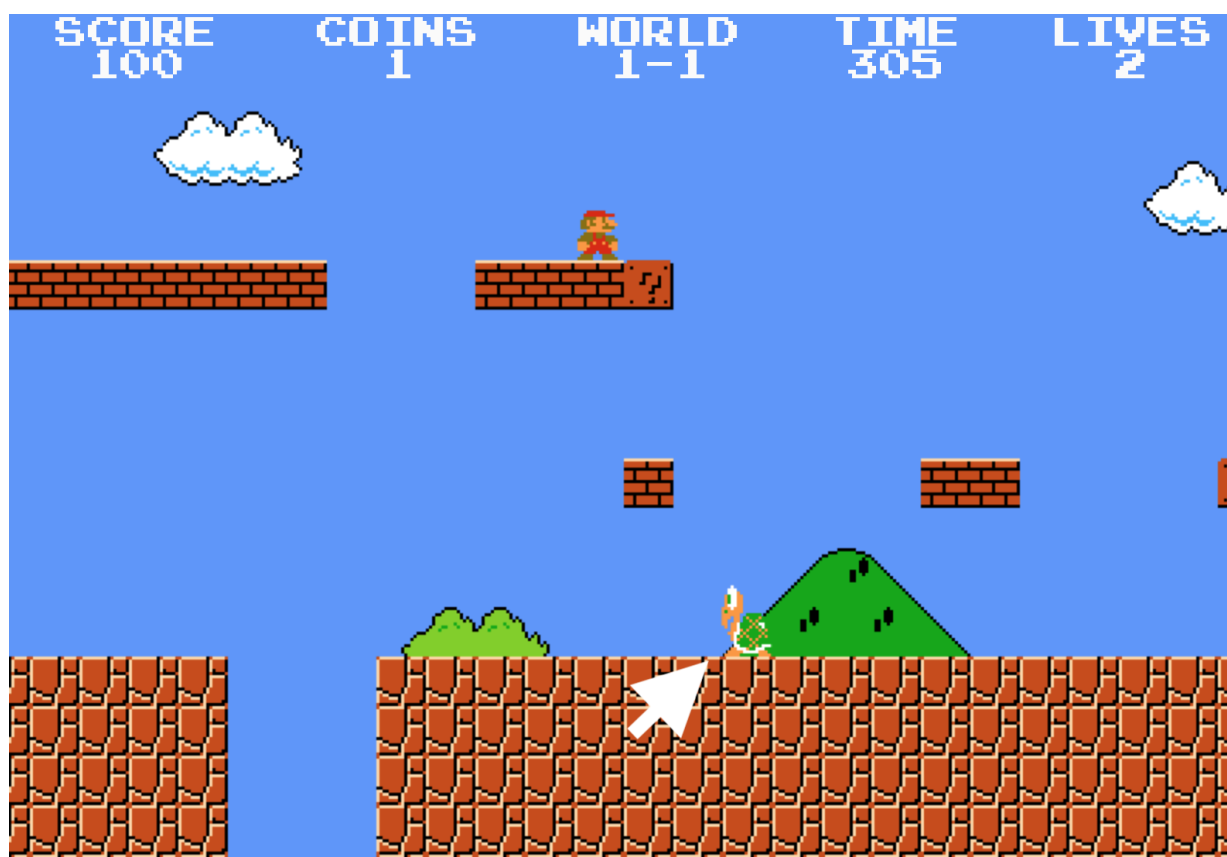


Рис. 4.1.5 – Здолання ворога, нарахування додаткових балів

Під час проходження рівня гравець має можливість заробляти монети, збиваючи за допомогою стрибка бонусні комірочки, позначені «?», проте інколи до них досить важко дістатися (рис. 4.1.6)

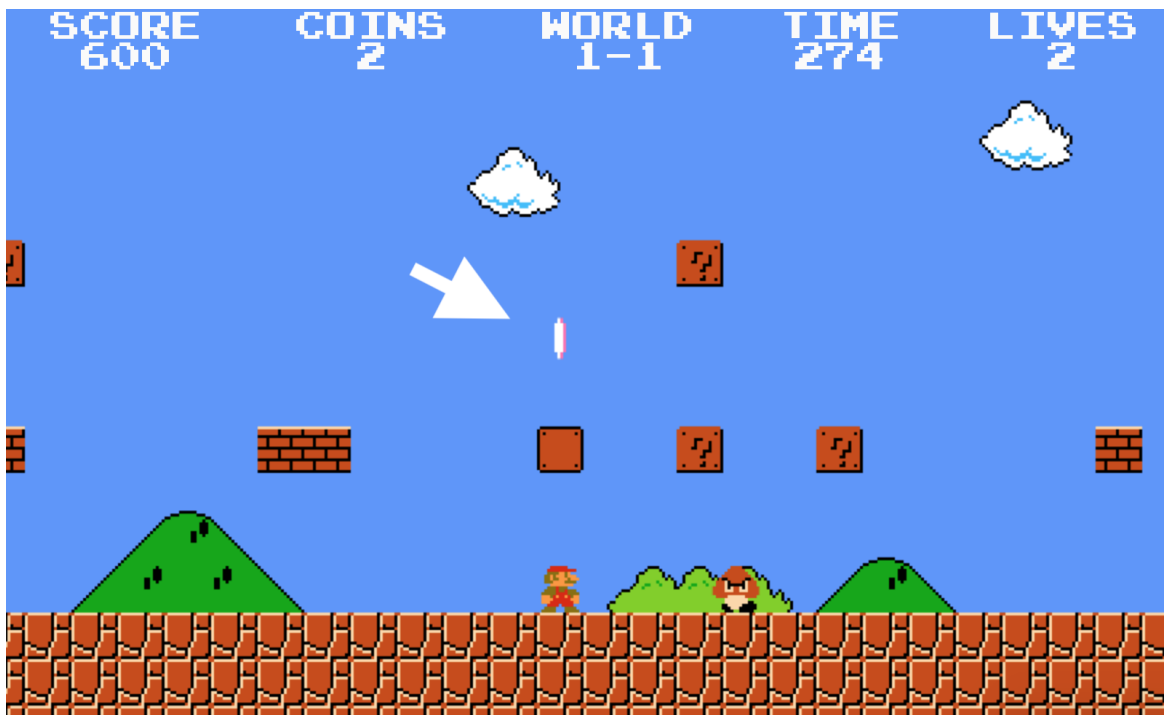


Рис. 4.1.6 – Накопичення монет з бонусних комірок

Через певний час, пройшовши низку випробувань, гравець бачить заповітну межу фінішу, відображену на рис. 4.1.7.

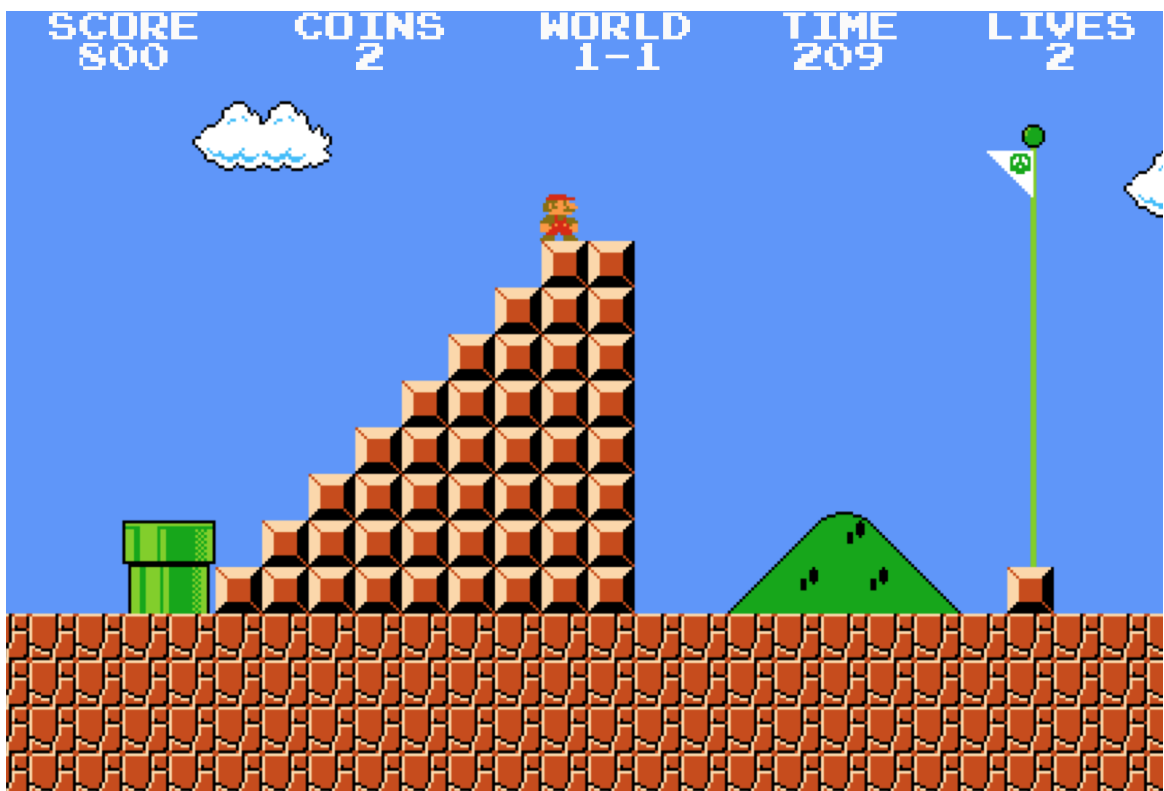


Рис. 4.1.7 – Проходження фінішної межі

Після проходження фінішу, весь не витрачений час на рівень перераховується у бали, завдяки чому користувач замотивований підвищувати моторику рухів та тренувати логіку, щоб проходити рівні якомога швидше (рис. 4.1.8)

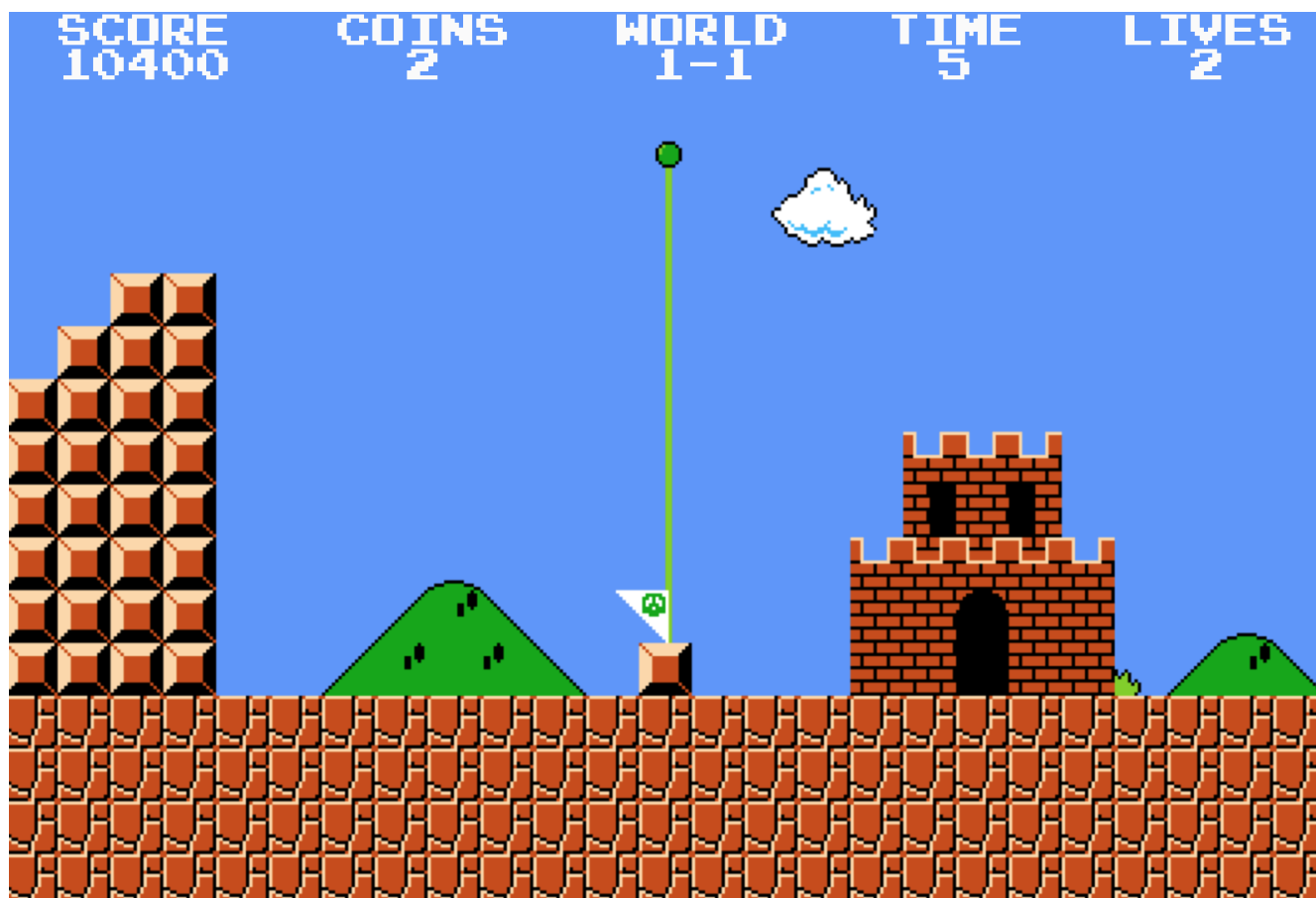


Рис. 4.1.8 – Перерахування залишеного часу у бали

Здолавши рівень, головний герой гри «Маріо» потрапляє у трубу, яка направляє його на наступний рівень, де він зіштовхнеться з більшою кількістю ворогів та складністю мапи (рис 4.1.9).

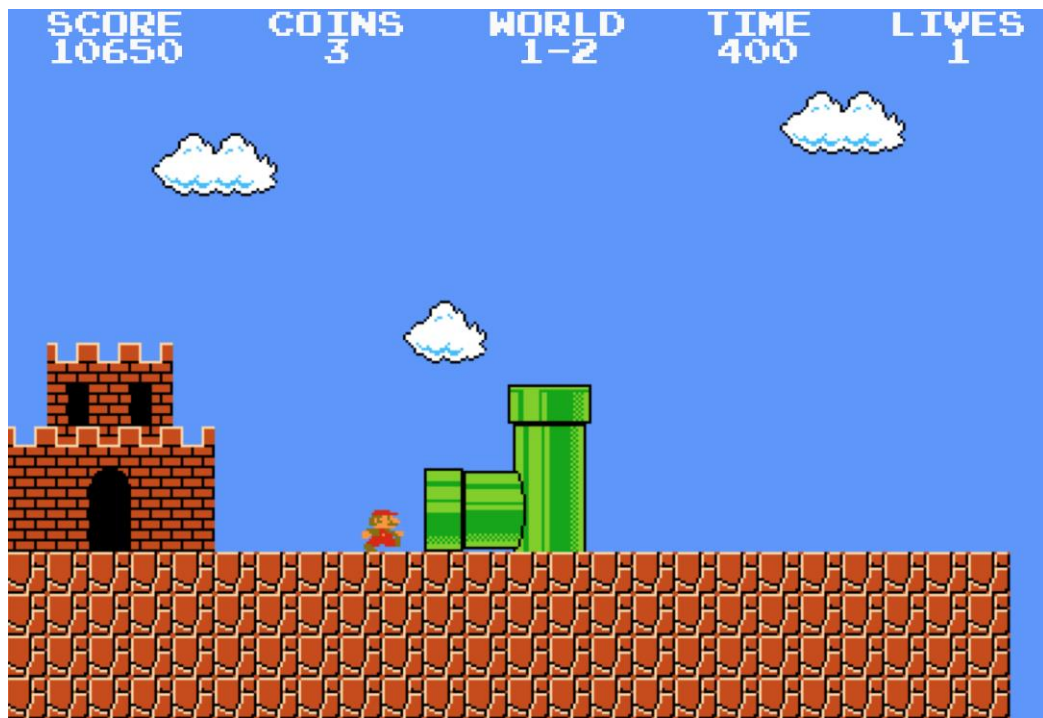


Рис. 4.1.9 – Перехід на наступний рівень

Новий рівень має іншу промальовану мапу з більшою складністю, нових ворогів, в яких тепер не одне ігрове життя, а декілька. Також новий рівень супроводжується перешкодами підвищеної складності (рис. 4.1.10).



Рис. 4.1.10 – Відображення нового рівня

Коли кількість життів (Lives) гравця дорівнює нулю – гра завершується, про що користувача сповіщає наступне вікно, зображене на рис. 4.1.11.



Рис. 4.1.11 – Вікно зі сповіщенням про кінець гри

Після завершення гри, через певний відлік часу гравець може спробувати заново пройти гру, набравши ще більше балів і витративши менше часу на проходження рівнів і здолання ворогів.

ВИСНОВКИ

Частка користувачів ПК та інтернет аудиторії зростає щодня. В основному користувачі використовують ПК в якості розважальної платформи. Таким чином, ігрова аудиторія продовжує зростати, тим самим мотивуючи розробників створювати нові ігрові проекти.

В рамках випускної кваліфікаційної роботи була вивчена і опрацьована технологія розробки ігор шляхом створення флеш ігрового додатку «Mario» в жанрі «Аркада». Розробка велася на мові програмування C ++ з використанням бібліотек SFML.

Для досягнення даної мети були вирішені наступні завдання:

- здійснена постановка ігрової задачі;
- проведений аналіз аналогів і програмних засобів розробки комп'ютерної гри;
- проведений пошук і підбір TileSet;
- повторені і поглиблені знання в мові програмування C++;
- вивчена робота з деякими бібліотеками SFML ;
- спроектовано додаток;

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Як Україна виглядає на світовому ринку розробки ігор — 8 графіків [Електронний ресурс]. – Режим доступу: <https://businessviews.com.ua/ru/tech/id/virobnictvo-igor-v-ukrajini-1948/>
2. Україна на світовому ІТ-ринку: 10 найцікавіших фактів — інфографіка [Електронний ресурс]. – Режим доступу: <https://biz.nv.ua/ukr/economics/ukrajina-na-svitovomu-rinku-it-top-10-najtsikavishih-faktiv-infografika-2353974.html>
3. Report on ICT and Digital Game Industry Technology Enhanced Learning and Service at Partner Country Ukraine [Електронний ресурс]. – Режим доступу: https://gamehubcbhe.deusto.es/wpcontent/uploads/2016/10/GameHub_D1.2.pdf
4. Worldwide Digital Games Market [Електронний ресурс]. – Режим доступу: <https://twitter.com/SuperData/status/675337478514024448>
5. Top Operating System Software [Електронний ресурс]. – Режим доступу: <http://www.jegsworks.com/lessons/ComputerBasics/lesson8/lesson8-3.html>
6. Українська ігрова індустрія [Електронний ресурс]. – Режим доступу: <https://vertigo.com.ua/ukrainska-ihrova-lokalizatsiia-skladnoshchi-uspikhy-i-perspektyvy>
7. CRYENGINE - The complete solution for next generation game development [Електронний ресурс]. – Режим доступу: <https://www.cryengine.com>.
8. What is Unreal Engine 4 ? [Електронний ресурс]. – Режим доступу: <https://www.unrealengine.com/>
9. Game programming [Електронний ресурс]. – Режим доступу: goo.gl/rxJ4xzc
10. What is SFML ? [Електронний ресурс]. – Режим доступу: <https://www.sfml-dev.org/grl-whatis>.
11. Super Mario Bros. [Електронний ресурс]. – Режим доступу: <https://ru.wikipedia.org/>.
12. The Tromb. [Електронний ресурс]. – Режим доступу:

<http://www.hexag.net/radiant/>.

13. Fairy. [Электронный ресурс]. – Режим доступа:
https://ru.wikipedia.org/wiki/Luxor_Game/.


```

sprite.setPosition(rect.left - offsetX, rect.top - offsetY);

dx=0;
}

void Collision(int num)
{
for (int i = rect.top/16 ; i<(rect.top+rect.height)/16; i++)
    for (int j = rect.left/16; j<(rect.left+rect.width)/16; j++)
        {
            if ((TileMap[i][j]=='P') || (TileMap[i][j]=='k') ||
(TileMap[i][j]=='0') || (TileMap[i][j]=='r') || (TileMap[i][j]=='t'))
                {
                    if (dy>0 &&
num==1)
                        { rect.top =
i*16 - rect.height; dy=0; onGround=true; }
                    if (dy<0 &&
num==1)
                        { rect.top =
i*16 + 16; dy=0;}
                    if (dx>0 &&
num==0)
                        { rect.left =
j*16 - rect.width; }
                    if (dx<0 &&
num==0)
                        { rect.left =
j*16 +16;}
                }

            if (TileMap[i][j]=='c') {
//
TileMap[i][j]=' ';
}
}
};

class ENEMY

```

```

{

public:
float dx,dy;
FloatRect rect;
Sprite sprite;
float currentFrame;
bool life;

    void set(Texture &image, int x, int y)
    {
sprite.setTexture(image);
rect = FloatRect(x,y,16,16);

        dx=0.05;
currentFrame = 0;
life=true;
    }

    void update(float time)
    {
rect.left += dx * time;

        collision();

        currentFrame += time * 0.005;
        if (currentFrame > 2) currentFrame -= 2;

        sprite.setTextureRect(IntRect(18*int(currentFrame), 0, 16,16));
        if (!life) sprite.setTextureRect(IntRect(58, 0, 16,16));

sprite.setPosition(rect.left - offsetX, rect.top - offsetY);

    }

    void collision()
    {
for (int i = rect.top/16 ; i<(rect.top+rect.height)/16; i++)
    for (int j = rect.left/16; j<(rect.left+rect.width)/16; j++)
        if ((TileMap[i][j]=='P') || (TileMap[i][j]=='0'))
            {
                if (dx>0)

```



```

j*16 - rect.width; dx*=-1; }

j*16 + 16; dx*=-1; }

}

};

```

```

{ rect.left =
else if (dx<0)
{ rect.left =
}
}
};

```

```

int main()
{
    Renderwindow window(VideoMode(400, 250), "SFML works!");

    Texture tileSet;
    tileSet.loadFromFile("Mario_Tileset.png");

    PLAYER Mario(tileSet);
    ENEMY enemy;
    enemy.set(tileSet,48*16,13*16);

    sprite tile(tileSet);

    soundBuffer buffer;
    buffer.loadFromFile("Jump.ogg");
    sound sound(buffer);

    Music music;
    music.openFromFile("Mario_Theme.ogg");
    music.play();

    clock clock;

    while (window.isOpen())
    {
        float time = clock.getElapsedTime().asMicroseconds();
        clock.restart();

        time = time/500; // здесь регулируем скорость игры

        if (time > 20) time = 20;
    }
}

```

```

Event event;
while (window.pollEvent(event))
{
    if (event.type == Event::Closed)
        window.close();
}

if (Keyboard::isKeyPressed(Keyboard::Left))    Mario.dx=-0.1;

if (Keyboard::isKeyPressed(Keyboard::Right))   Mario.dx=0.1;

if (Keyboard::isKeyPressed(Keyboard::Up)) if (Mario.onGround) { Mario.dy=-
0.27; Mario.onGround=false;  sound.play();}

Mario.update(time);
enemy.update(time);

if (Mario.rect.intersects( enemy.rect ) )
{
    if (enemy.life) {
        if (Mario.dy>0) { enemy.dx=0; Mario.dy=-0.2; enemy.life=false;}
        else Mario.sprite.setColor(Color::Red);
    }
}

if (Mario.rect.left>200) offsetX = Mario.rect.left-200;
//смещение

window.clear(Color(107,140,255));

for (int i=0; i<H; i++)
    for (int j=0; j<W; j++)
    {
        if (TileMap[i][j]=='P') tile.setTextureRect( IntRect(143-
16*3,112,16,16) );

        if (TileMap[i][j]=='k') tile.setTextureRect(
IntRect(143,112,16,16) );

```

```

        if (TileMap[i][j]=='c') tile.setTextureRect( IntRect(143-
16,112,16,16) );

        if (TileMap[i][j]=='t') tile.setTextureRect(
IntRect(0,47,32,95-47) );

        if (TileMap[i][j]=='g') tile.setTextureRect( IntRect(0,16*9-
5,3*16,16*2+5) );

        if (TileMap[i][j]=='G') tile.setTextureRect(
IntRect(145,222,222-145,255-222) );

        if (TileMap[i][j]=='d') tile.setTextureRect(
IntRect(0,106,74,127-106) );

        if (TileMap[i][j]=='w') tile.setTextureRect(
IntRect(99,224,140-99,255-224) );

        if (TileMap[i][j]=='r') tile.setTextureRect( IntRect(143-
32,112,16,16) );

        if ((TileMap[i][j]==' ') || (TileMap[i][j]=='0')) continue;

        tile.setPosition(j*16-offsetX,i*16 - offsetY) ;
        window.draw(tile);
    }

    window.draw(Mario.sprite);
    window.draw(enemy.sprite);

    window.display();
}

return 0;
}

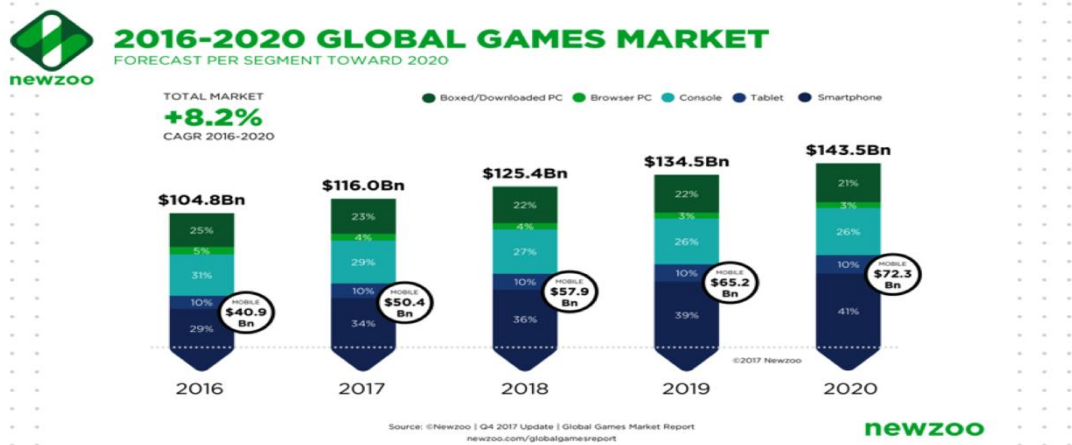
```

Розробка комп'ютерної гри жанру флеш на уважність мовою програмування C++

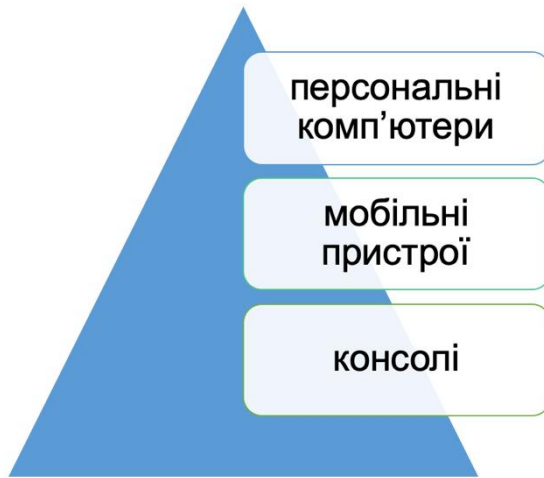


Макогонюк Максим Олегович
студент групи ПП3-52

Світовий ігровий ринок за підсумками 2020 року



Ігрові платформи сьогодні

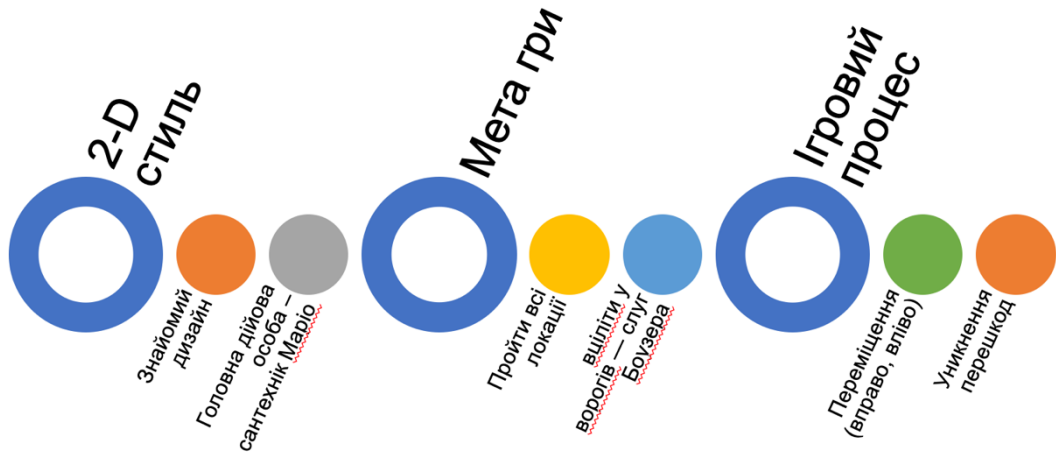


Жанри ігор сьогодні



- | | | | |
|-----------|-------------|-------------|--------------|
| ■ баталії | ■ стратегії | ■ азартні | ■ пазл |
| ■ екшени | ■ сімейні | ■ кежуал | ■ rpg |
| ■ пригоди | ■ аркади | ■ спортивні | ■ симулятори |

Розробка мультимедійного ігрового додатку



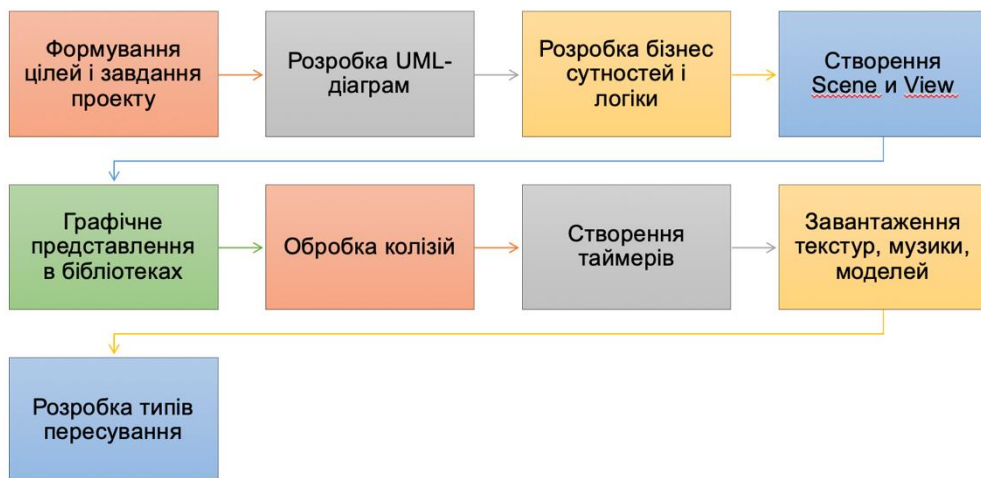
Чому аркада?



Вибір програмної складової



Розробка програмної системи



годні

Дякую за увагу!