

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КОЛЕДЖ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА
ЗЕМЛЕВПОРЯДКУВАННЯ НАУ**

Кафедра _____ Розробка програмного забезпечення _____

ДИПЛОМНИЙ ПРОЕКТ
(ПОЯСНЮВАЛЬНА ЗАПИСКА)

**ВИПУСКНИКА ОСВІТНЬО-КВАЛІФІКАЦІЙНОГО РІВНЯ
"МОЛОДШИЙ СПЕЦІАЛІСТ"**

Тема: «Ігровий модуль у стилі «Shooter»

Виконавець: Рабін І. Р. _____

Керівник: Нечипорук В. В. _____

Нормоконтролер: Проценко С. М. _____

Рецензент: _____

**КОЛЕДЖ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА
ЗЕМЛЕВПОРЯДКУВАННЯ НАУ**

Кафедра _____ Розробка програмного забезпечення _____

Напрямок (спеціальність) 5.05010301 «Розробка програмного забезпечення» _____

ЗАТВЕРДЖУЮ
Голова комісії

_____ Апенько Н. В.

« _____ » _____ 2017 р.

**ЗАВДАННЯ
НА ВИКОНАННЯ ДИПЛОМНОГО ПРОЕКТУ СТУДЕНТА**

Рабіна Ігора Романовича
(прізвище, ім'я, по батькові)

1.Тема проекту (роботи): «Ігровий модуль у стилі «Shooter»

затверджена наказом директора від «16» січня 2017 року №02/СТ

2.Термін виконання проекту (роботи): з 15 травня 2017р. по 17 червня 2017р.

3.Вихідні дані до проекту (роботи): системна програмна документація, мова програмування C#, кросплатформений інструмент для розробки дво- та тривимірних ігор Unity3D, програма 3D- моделювання Blender.

4.Зміст пояснювальної записки (перелік питань, що підлягають розробці):

а) аналіз предметної області та постановка завдання проектування;

б) структура комп'ютерної гри;

в) проектна частина та методи;

г) керівництво користувача;

д) охорона праці.

5.Перелік обов'язкового графічного матеріалу:

6. Календарний план

<i>№ п/п</i>	<i>Етапи виконання дипломного проекту</i>	<i>Термін виконання етапів</i>	<i>Примітка</i>
1	<i>Аналіз предметної області та постановка завдання проектування</i>	<i>15.05.2017 – 18.05.2017</i>	
2	<i>Проектування інтерфейсу користувача та модулів</i>	<i>19.05.2017 – 23.05.2017</i>	
3	<i>Створення моделей та матеріалів</i>	<i>24.05.2017 – 31.05.2017</i>	
4	<i>Програмна реалізація контролера та інших скриптів</i>	<i>01.06.2017 – 06.06.2017</i>	
5	<i>Тестування ігрового модуля</i>	<i>07.06.2017 – 09.06.2017</i>	
6	<i>Написання пояснювальної записки</i>	<i>12.06.2017 – 16.06.2017</i>	

7. Дата видачі завдання 16 січня 2017 р.

Керівник дипломного проекту _____
(підпис)

Завдання прийняв до виконання _____
(підпис студента)

РЕФЕРАТ

Пояснювальна записка до дипломного проекту «Ігровий модуль у стилі «Shooter»: 60 с., 38 рис., 45 літературних джерел.

С#, МЕТОД, МОДУЛЬ, UNITY3D, BLENDER, ГРА, SHOOTER, КОНТРОЛЛЕР, СКРИПТ

Об'єкт проектування – ігрова індустрія.

Предмет проектування – ігровий модуль у стилі «Shooter».

Мета дипломного проекту – розробка ігрового модуля у стилі «Shooter».

Методи проектування – мова програмування С#, кросплатформенний інструмент для розробки дво- та тривимірних ігор Unity3D, програма 3D-моделювання Blender.

Прогнозні припущення щодо розвитку об'єкта дослідження – створення робочого зразка модуля та використання його в комп'ютерних системах для розвитку дрібної моторики.

Результати проектування – ігровий модуль, який розроблено у стилі «Shooter», призначений для розвитку дрібної моторики, швидкого реагування та зорової уваги користувача.

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ ПРОЕКТУВАННЯ	8
1.1 Аналіз сучасних ігор у стилі «Shooter».....	8
1.2 Аналіз методів і технологій в галузі розробки комп'ютерних ігор.....	11
1.3 Постановка завдання проектування.....	17
1.4 Висновки до розділу.....	18
РОЗДІЛ 2. СТРУКТУРА КОМП'ЮТЕРНОЇ ГРИ.....	19
2.1 Функціональні вимоги.....	19
2.2 Структура гри.....	20
2.3 Висновки до розділу.....	23
РОЗДІЛ 3. ПРОЕКТНА ЧАСТИНА ТА МЕТОДИ.....	24
3.1 Первинні налаштування.....	24
3.2 Основні скрипти.....	31
3.3 Висновки до розділу.....	37
РОЗДІЛ 4. КЕРІВНИЦТВО КОРИСТУВАЧА.....	38
4.1 Вимоги до програмного продукту.....	38
4.2 Установка гри.....	38
4.3 Ігровий процес.....	40
4.4 Тестування.....	45
4.5 Висновки до розділу.....	46
РОЗДІЛ 5. ОХОРОНА ПРАЦІ	47
Вступ	47
5.1 небезпечні та шкідливі виробничі фактори, що діють у робочій зоні.....	49
5.2 Організаційні і технічні заходи для виключення небезпечних і зменшення рівня шкідливих виробничих факторів	51
ВИСНОВКИ	55
ВИКОРИСТАНА ЛІТЕРАТУРА	57

ВСТУП

Науково-технічний прогрес набрав до кінця XX ст. запаморочливу швидкість, спричинив появу такого чуда сучасності як комп'ютер і комп'ютерні технології. З вдосконаленням комп'ютерів удосконалювалися і ігри, залучаючи все більше і більше людей. На сьогоднішній день комп'ютерна техніка досягла такого рівня розвитку, що дозволяє програмістам розробляти дуже реалістичні ігри з гарним графічним і звуковим оформленням.

Грають не тільки діти, а й дорослі. І грають, як правило, в робочий час. Зараз персональні комп'ютери поширені повсюди. Напевно, немає жодного учня, який би не любив грати в комп'ютерні ігри. Деякі з нас готові грати день і ніч. Батьки ж навпаки, забороняють грати в комп'ютерні ігри або встановлюють для гри дуже маленький проміжок часу, хоча й самі часом не проти пограти.

Поява комп'ютерних ігор можна віднести до моменту, коли комп'ютери зі сфери експериментальної і майже секретної (адже на них повинні були розраховуватися траєкторії снарядів і ракет під час військових дій) почали переходити в світ науковий і практичний. Це сталося в кінці 60-х рр. XX ст.

Комп'ютер став володіти якимось більш-менш дружнім користувачеві інтерфейсом – замість лампочок і загадкових індикаторів з'явилися алфавітно-цифрові дисплеї. Звичайно, ні про яку графіку не могло йти й мови. Але за комп'ютерами працюють люди і ніщо людське їм не чуже. І ось в один прекрасний вечір після важкого трудового дня молодий програміст (а кому ще могла прийти в голову ідея використовувати комп'ютер не за прямим призначенням) вирішив написати невелику програму, яка грала б з ним в якусь не надто складну гру, наприклад «Бики і корови» і звичайно ж, така ідея прийшла в голову не тільки йому одному. Незабаром програми для розваги почали з'являтися все частіше і частіше і навіть стали входити до складу пакетів програм, що поставляються разом з комп'ютерами.

Зі створенням комп'ютерної графіки і появою справжніх домашніх комп'ютерів ігрова індустрія стрімко зростає. Ігри випускалися тисячними тиражами, не рахуючи піратських копій. Приблизно за десять років для

домашнього комп'ютера ZX-Spectrum фірми Sinclair Research було випущено більше 6 тис. найменувань ігор.

Зараз ігрова індустрія є однією з точок опори, на яких стоїть індустрія персональних ЕОМ.

У кожному жарті є частка правди, і світ, напевно, не побачив би «комп'ютерної революції», якби не комп'ютерні ігри.

Перед користувачами постає завдання обрати найбільш зручний інструмент з багатим функціоналом, який буде інтуїтивно зрозумілим та не буде навантажувати систему.

Об'єктом дипломного проекту є ігрова індустрія.

Мета проектування – розробка програмного модуля у стилі «Shooter» для розвитку дрібної моторики.

Предмет – комп'ютерна гра.

Дипломний проект розділений на 5 частин.

У першій частині проводиться аналіз сучасних видів ігор, зокрема «Shooter» версії ігор, їх функціональні можливості та особливості, а також визначається завдання проектування.

Друга частина відповідає за структуру програмного продукту.

У третій частині відбувається розбір алгоритмів та пояснення їх роботи.

В четвертій частині наводиться керівництво користувача і інші навчальні матеріали.

П'ята частина – охорона праці.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ ПРОЕКТУВАННЯ

1.1 Аналіз сучасних ігор у стилі «Shooter»

До найближчих аналогів розробленої гри можна віднести DOOM, Quake, Half-Life та Counter Strike.

DOOM.

DOOM – відеогра, випущена компанією «id Software» 10 грудня 1993 року, один з найбільш відомих та популярних шутерів від першої особи. Вона багато в чому здійснила визначальний ідейний вплив на подальший розвиток жанру шутерів [1].

Особливості:

- перший у своєму роді шутер;
- рух без зупинки;
- складні рівні з скарбницями та рогадженнями.

Недоліки:

- дуже кривава гра, що не добре впливає на дітей та психіку;
- мультиплеєр – не оптимізований.

Інтерфейс даної гри зображено на Рис. 1.1:



Рис. 1.1 – Інтерфейс гри «DOOM»

Кафедра РПЗ				КІТЗ НАУ 17 47.000 ПЗ			
Виконав	Рабін І.Р			АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАВДАННЯ ПРОЕКТУВАННЯ	Літера	Аркуш	Аркушів
Керівник	Нечипорук В. В.				Д	8	60
Консулт.					РПЗ – 48		
Норм. контр.	Проценко С. М.						
Голова комісії.	Апенько Н.В.						

Quake.

Quake – відеогра жанру шутера від першої особи, розроблена «id Software» і випущена 22 червня 1996 року. Quake здійснила прорив у 3D-технологіях, повністю реалізувавши тривимірний світ замість двовимірної карти з інформацією про висоту (як було в DOOM). Quake була однією із небагатьох ігор, що дозволяла грати через Інтернет, а не лише в локальній мережі, що значною частиною зумовило її популярність [2].

Особливості:

- сюжет;
- мультиплеєр;
- музичний супровід;

Інтерфейс гри Quake зображено на рис. 1.2:



Рис. 1.2 – Інтерфейс гри «Quake»

Half-Life.

Half-Life – відеогра жанру науково-фантастичного шутера від першої особи, розроблена «Valve Software» і видана «Sierra Studios» 20 листопада 1998 року для

ПК. Гра базується на значно переробленому ігровому рушієві Quake від «id Software» [3].

Особливості:

- наявність першого штучного інтелекту у своєму стилі;
- динамічний сюжет;
- реалістичний світ;

Інтерфейс гри зображено на рис. 1.3:



Рис. 1.3 – Інтерфейс гри «Half-Life»

Counter Strike.

Counter Strike – серія комп'ютерних ігор в жанрі командного шутера від першої особи, заснована на рушію GoldSrc, спочатку з'явилася як модифікація гри Half-Life. Всього в основній серії вийшло п'ять версій гри, найпопулярніша з яких – Counter-Strike 1.6. Існують і окремі частини, наприклад Counter-Strike: Condition Zero (мультиплеєр з одноразовими завданнями) і Condition Zero: Deleted Scenes (сюжетна гра), засновані на рушію GoldSource, а також Counter-Strike: Source, заснована на більш сучасному рушію Source. 14 серпня 2011 року Valve офіційно презентувала Counter-Strike: Global Offensive, яка вийшла 21 серпня 2012 року.[4]

Особливості:

- мультиплеєр;
- нестійкість текстур.

Інтерфейс гри Counter Strike зображено на рис. 1.4:



Рисунок 1.4 – Інтерфейс гри «Counter Strike»

1.2 Аналіз методів і технологій в галузі розробки комп'ютерних ігор

При розробці ігор використовуються різні методи і засоби, але їх всі можна умовно розділити на декілька складових:

1. Мова програмування. Залежить від системи, під яку створюється гра, та можливості виконання поставлених задач.
2. Бібліотеки.

3. Загальні поняття. Платформа. Unity3D, Blender. Вибір найбільш зручного інтерфейсу користувача.

Огляд мови програмування та середовища розробки

C# – об'єктно-орієнтована мова програмування, створена для роботи у середовищі Microsoft .NET Framework. Мова C# була розроблена з урахуванням сильних і слабких особливостей інших мов, зокрема Java і C++. Специфікація мови C# була написана Андерсом Гейлсбергом, Скотом Вілтамутом та Пітером Гольде.

Ключові особливості мови C#:

- компонентна орієнтованість;
- код зібраний воедино;
- уніфікована система типів і їх безпечність;
- автоматична і мануальна робота з пам'яттю;
- використання єдиної бібліотеки класів – CLR.

C# є дуже близьким родичем мови програмування Java. Мова Java була створена компанією Sun Microsystems, коли глобальний розвиток інтернету поставив задачу розсосереджених обчислень. Взявши за основу популярну мову C++, Java виключила з неї потенційно небезпечні речі (типу вказівників без контролю виходу за межі). Для розсосереджених обчислень була створена концепція віртуальної машини та машинно-незалежного байт-коду, свого роду посередника між вихідним текстом програм і апаратними інструкціями комп'ютера чи іншого інтелектуального пристрою. Java набула чималої популярності, і була ліцензована також і компанією Microsoft. Але з плином часу Sun почала винуватити Microsoft, що та при створенні свого клону Java робить її сумісною виключно з платформою Windows, чим суперечить самій концепції машинно-незалежного середовища виконання і порушує ліцензійну угоду. Microsoft відмовилася піти назустріч вимогам Sun, і тому з'ясування стосунків набуло статусу судового процесу. Суд визнав позицію Sun справедливою, і зобов'язав Microsoft відмовитися від позаліцензійного використання Java.

У цій ситуації в Microsoft вирішили, користуючись своєю вагою на ринку, створити свій власний аналог Java, мови, в якій корпорація стане повновладним господарем. Ця новостворена мова отримала назву C#. Вона успадкувала від Java концепції віртуальної машини (середовище .NET), байт-коду (MSIL) і більшої безпеки вихідного коду програм, плюс врахувала досвід використання програм на Java. Нововведенням C# стала можливість легшої взаємодії, порівняно з мовами-попередниками, з кодом програм, написаних на інших мовах, що є важливим при створенні великих проектів. Якщо програми на різних мовах виконуються на платформі .NET, .NET бере на себе клопіт щодо сумісності програм (тобто типів даних, за кінцевим рахунком).

Станом на сьогодні C# визначено флагманською мовою корпорації Microsoft, бо вона найповніше використовує нові можливості .NET. Решта мов програмування, хоч і підтримуються, але визнані такими, що мають спадкові прогалини щодо використання .NET.

Середовище розробки.

MonoDevelop – відкрите інтегроване середовище розробки для платформ Linux, Mac OS X та Microsoft Windows, передусім націлене на розробку програм, які використовують і Mono, і Microsoft .NET Framework. На даний момент підтримуються мови C#, Java, Boo, Python, Vala, C та C++. Також MonoDevelop підтримує такі технології, як ASP.NET MVC, Silverlight, MonoMac и MonoTouch.

Наприкінці 2003 року кілька розробників зі спільноти Mono почали портувати SharpDevelop, успішний відкритий .NET IDE з System.Windows.Forms під Windows на Gtk# під Linux. Почавшись як відгалуження SharpDevelop, MonoDevelop архітектурно відрізняється від останніх релізів SharpDevelop.

З часом проект MonoDevelop був занурений в решту частин проекту Mono і активно підтримується Novell і спільнотою Mono. Починаючи від Mono 1.0 Beta 2, MonoDevelop включений в комплект релізів Mono.

Роль платформи .NET

Microsoft .NET – програмна технологія, запропонована фірмою Microsoft як платформа для створення як звичайних програм, так і веб-застосунків. Одною з ідей .NET є сумісність служб, написаних різними мовами.

Можна виділити наступні її основні риси:

1. Підтримка декількох мов. Основою платформи є загальномовне середовище виконання Common Language Runtime (CLR), завдяки чому .NET підтримує кілька мов: поряд з C# це також VB.NET, C++, F#. При компіляції код на будь-якому з цих мов компілюється в збірку спільною мовою CIL (Common Intermediate Language).

2. Кросплатформеність.

3. Потужна бібліотека класів. .NET Представляє єдину для всіх підтримуваних мов бібліотеку класів.

4. Різноманітність технологій. Загальномовне середовище виконання CLR і базова бібліотека класів є основою для цілого стека технологій, які розробники можуть задіяти при побудові тих чи інших додатків.

Бібліотеки.

Використовувались такі бібліотеки C# Unity3D для розробки скриптів:

UnityEngine; //Включає основні функції Unity3D

System; //Дозволяє працювати з математичними виразами і т.д.

System.Collections; //Включає основні розширення (колекції)

System.Collections.Generic; //Підключає додаткові колекції

System.IO; //Дозволяє працювати з зчитуванням і записом в файл

System.Linq; //Дозволяє працювати з динамічними масивами і т.д.

Загальні поняття. Платформа. Unity3D, Blender.

Шутер (Shooter) – це жанр відеоігор, піджанр action, основу ігрового процесу якого складає стрільба зі зброї по цілях, зазвичай вороже налаштованих. Часто основним завданням гравця є знищення усіх цілей на рівні та набір найбільшої кількості очок, а не проходження сюжету. Шутери бувають різних типів, наприклад:

Шутер від першої особи.

Ігри цього жанру, де основна частина ігрового процесу – це знищення ворогів із різноманітної вогнепальної зброї. Характерною особливістю жанру є вид «з очей» головного героя. Шутери від першої особи виникли тоді, коли з'явилися можливості показувати на плоскому екрані картинку, що відтворювала тривимірний світ, 2.5D або в повноцінному 3D. Саме з цього підвиду з'явився жанр шутерів.

Шутер від третьої особи.

Ігри цього жанру мають багато подібностей з шутером від першої особи, але відрізняються від них розташуванням ігрової камери – тут вона знаходиться позаду персонажа. Шутери від третьої особи мають інтерфейс (HUD). Більшість ігор відображають на екрані інформацію про гравця: зброя/обладнання, боєприпаси, здоров'я персонажа, ключові точки (зазвичай стрілкою в напрямку об'єкта), список завдань тощо. У наш час перспективу обирає шутер від першої особи та здебільшого кооперативні (онлайн), які зосереджуються на якомусь одному режимі багатокористувацької гри.

Unity3D – це інструмент для розробки двох і тривимірних додатків та ігор, що працює під операційними системами Windows і OS X. Створені за допомогою Unity програми працюють під операційними системами Windows, OS X, WindowsPhone, Android, AppleiOS, Linux, а також на ігрових приставках Wii, PlayStation 3, PlayStation 4, Xbox 360, XboxOne. Є можливість створювати додатки для запуску в браузері за допомогою спеціального модуля Unity (UnityWebPlayer), а також за допомогою реалізації технології WebGL. Раніше була експериментальна підтримка реалізації проектів в рамках модуля AdobeFlashPlayer, але пізніше команда розробників Unity прийняла складне рішення щодо відмови від цього.

В сучасних умовах актуальним є впровадження інформаційно-комп'ютерних технологій в систему освіти країни. Тим більше що сьогодні більшість молодих людей вільно володіють персональним комп'ютером і вміло використовують відомості, отримані з Інтернету, а їм часто зручніше заглянути в глобальну

мережу, ніж шукати необхідний матеріал в традиційній друкованій навчальній літературі.

Інформатизація освіти і зростаючі вимоги до якості і кількості висококваліфікованих фахівців призводять до необхідності розробки і впровадження інноваційних освітніх методик і технологій, що сприяють формуванню нових форм навчання, що необмежені просторово-часовими рамками.

Blender – пакет для створення тривимірної комп'ютерної графіки, що включає засоби моделювання, анімації, вимальовування, після обробки відео, а також створення відеоігор.

Особливостями пакету є малий розмір, висока швидкість вимальовування, наявність версій для багатьох операційних систем – FreeBSD, GNU/Linux, Mac OS X, Microsoft Windows, SkyOS, MorphOS та Pocket PC. Пакет має такі функції, як динаміка твердих тіл, рідин та м'яких тіл, систему гарячих клавіш.

У Blender, як і у всіх 3D-редакторах, є доступ до різних базових об'єктів (фігур), які допоможуть почати роботу.

З цього меню користувач може застосовувати різні інструменти для зміни форми і поліпшення моделі. Наприклад, якщо потрібно взяти куб і створити з нього тривимірний символ «плюс», то можна використовувати один з найпопулярніших інструментів моделювання: Витискання (Extrude). Стандартний куб, з виділеної верхньою межею, в якому верхня грань видавлена вгору, як зображено на рис. 1.4:

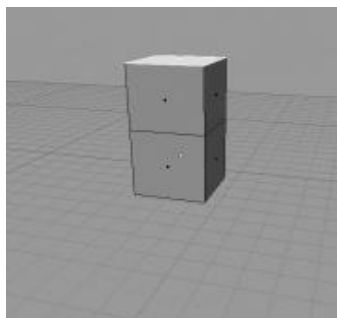


Рисунок 1.4 – Куб з видавленою верхньою гранню

Далі користувач має змогу зробити фігуру, наприклад «плюс». Тепер він, можливо, захочете змінити форму символу, збільшити ширину середньої частини кожного боку «плюса». Щоб зробити щось подібне, необхідно застосувати інший популярний інструмент – Loop Cut, як зображено на рис 1.5:

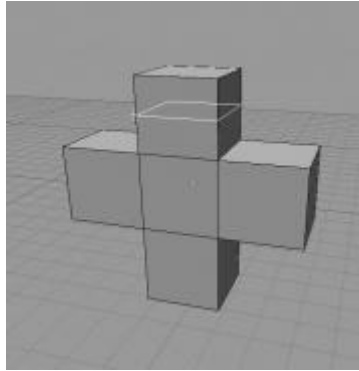


Рисунок 1.5 – Застосування інструменту Loop Cut для зміни масштабу

1.3 Постановка завдання проектування

Відповідно до мети в проекті поставлені наступні завдання:

1. Проаналізувати сучасні комп'ютерні ігри, які орієнтовані на розвиток дрібної моторики та уваги користувача.
2. Розробити інтерфейс ігрового модуля.
3. Створити контроллер персонажу для управління переміщенням у просторі.
4. Розробити структуру гри відповідно до функціональних вимог.
5. Реалізувати методи відображення та редагування параметрів контролю персонажу.
6. Протестувати ігровий модуль на різних пристроях та при різних навантаженнях.

1.4 Висновки до розділу

Проаналізовано існуючі аналоги ігор, які орієнтовані на розвиток дрібної моторики та уваги користувача. Усі вони відмінні між собою, тому було поставлено завдання створити ігровий модуль, що буде ще одним аналогом ігор у сфері «Shooter». Для роботи з контроллером персонажа було обрано мову програмування C# та платформу Unity3D. Для створення моделі середовища та його компонентів було обрано програму 3D-моделювання Blender.

РОЗДІЛ 2. СТРУКТУРА КОМП'ЮТЕРНОЇ ГРИ

2.1 Функціональні вимоги

До гри поставлені наступні функціональні вимоги:

1. Ігровий модуль повинен бути реалізований на ПК. Використання візуальних засобів повинно бути погоджено з викладачем.
2. Відображення і можливість редагування параметрів персонажу, наприклад швидкості.
3. Ігровий модуль повинен візуалізувати ігрове поле, на якому відбуваються зміни (знищення супротивника, зроблені постріли, зроблені постріли противника).
4. Ігровий модуль повинен реалізовувати певний оптимальний метод гри комп'ютера. При цьому, в кожній новій грі комп'ютер повинен грати по-різному, незалежно від розташування супротивника, тобто його ходи не повинні бути передбачувані.
5. В процесі гри повинен вестись протокол гри, в який будуть заноситися інформаційні повідомлення про зроблені ходи супротивників.
6. Зручна можливість запуску гри.
7. Зручний та приємний інтерфейс користувача.
8. Оптимізація гри під менш потужні ПК.

Вимоги до інтерфейсу користувача:

1. Головне меню. При відкритті гри повинне відображатись головне меню з управлінням персонажу та деякими функціональними кнопками.
2. Кнопка змінення режиму гри: Мультиплеєр або боти.

Кафедра РПЗ				КІТЗ НАУ 17 47.000 ПЗ			
Виконав	Рабін І.Р.			ПЕРЕЛІК ВИМОГ ДО ПРОГРАМНОГО ПРОДУКТУ	Літера	Аркуш	Аркушів
Керівник	Нечипорук В. В.				Д		19 60
Консульт.					РПЗ – 48		
Норм. контр.	Проценко С. М.						
Голова комісії	Апенько Н.В.						

3. Наявність кнопки «Опції», яка дозволяє змінювати чутливість миші, зміни складності ботів, включення аудіосупроводу до гри, зміни параметрів графіки.

4. Наявність кнопки «Пауза та «Повернутися у гру».

5. Можливість зміни «Нікнейму».

6. Наявність кнопки «Вихід з гри».

2.2 Структура гри

Структура гри (рис. 2.1) описана трьома модулями: управління, анімацій та відображення.



Рисунок 2.1 – Структура гри

Модуль управління містить в собі основні методи гри (скрипти) для управління подіями на сцені та контроллер персонажу.

Модуль анімацій містить в собі опис анімацій та ефектів, що будуть зображені на сцені.

Модуль відображення містить у собі всі статичні об'єкти гри та координати їх положення у просторі.

Структурна схема ігрового модуля зображена на рис. 2.2:

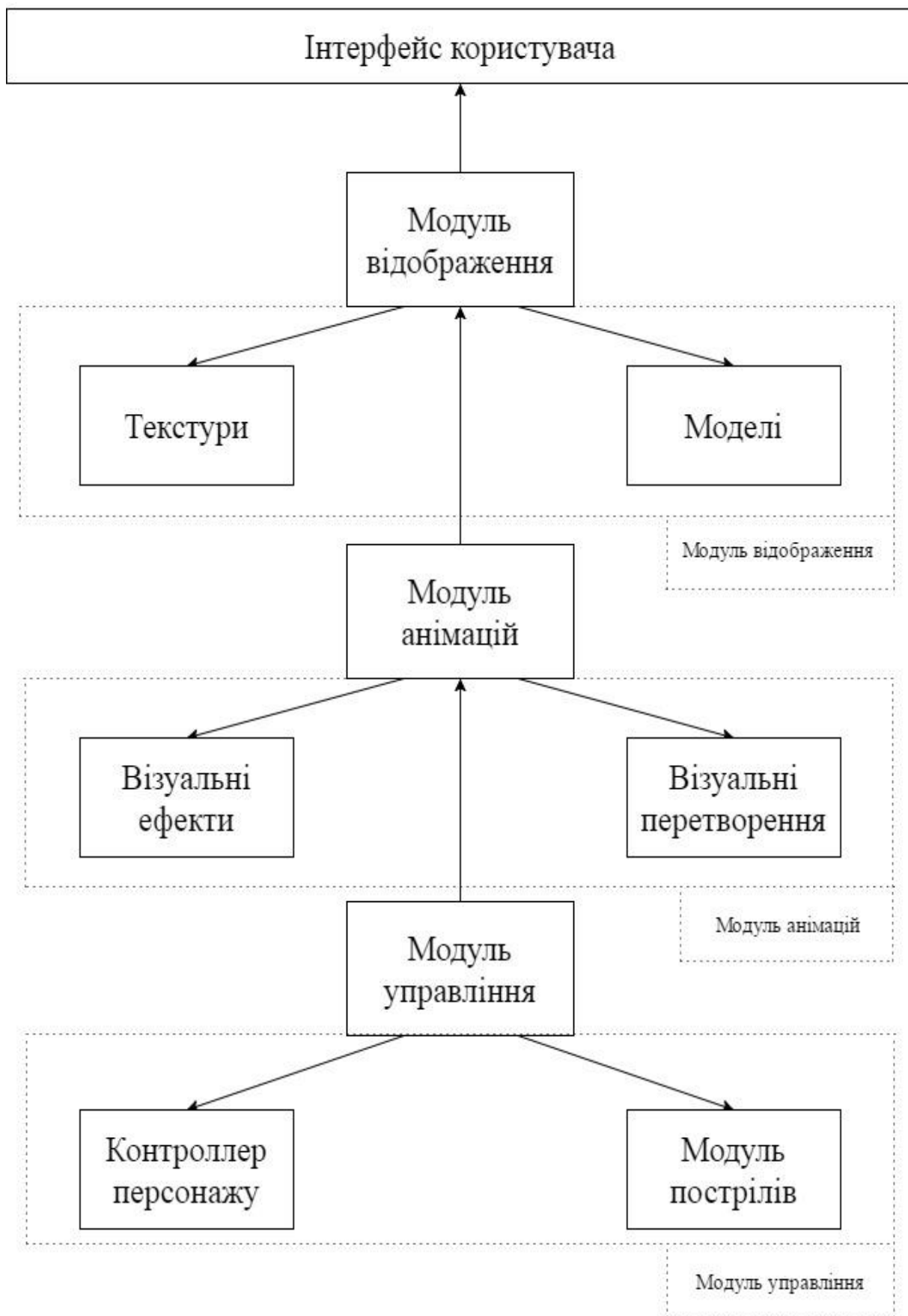


Рисунок 2.2 – Структурна схема ігрового модуля

Дана схема включає:

- модуль відображення. Він містить моделі та текстури, що лягають в основу відображення елементів та компонентів ігрового процесу. Текстурами виступають вбудовані матеріали Blender-a та Unity3D, моделями – вбудовані та створені елементи середовища;
- модуль анімації. Містить у собі стандартні візуальні ефекти та перетворення середовища розробки Unity3D, які були адаптовані під створені компоненти та моделі;
- модуль управління. Складається з контролеру персонажу та модуля пострілів, які описані скриптами, створеними за допомогою мови програмування C#.

Для початку гри користувач обирає початкові параметри в головному меню. Цими параметрами виступають режим гри (мультиплеєр або офлайн з ботами), налаштування миші, вибір складності та графіки, а також можна вказати бажаний нікнейм.

Всі ці параметри буде розроблено у наступних версіях гри, а на даний момент часу ігровий модуль представляє з себе тир, у якому користувачу потрібно ходити по середовищу та влучати у цілі, розставлені на сцені.

Ігровий процес починається з натиснення на кнопку «Start». Після натиснення відкривається головна сцена, що складається з моделі поверхні та компонентів середовища (цілей).

Ігровий процес закінчується, коли користувач захоче вийти з сцени гри до головного меню за допомогою кнопки «Escape»

Діаграма використання можливостей користувача зображена на рис. 2.3:

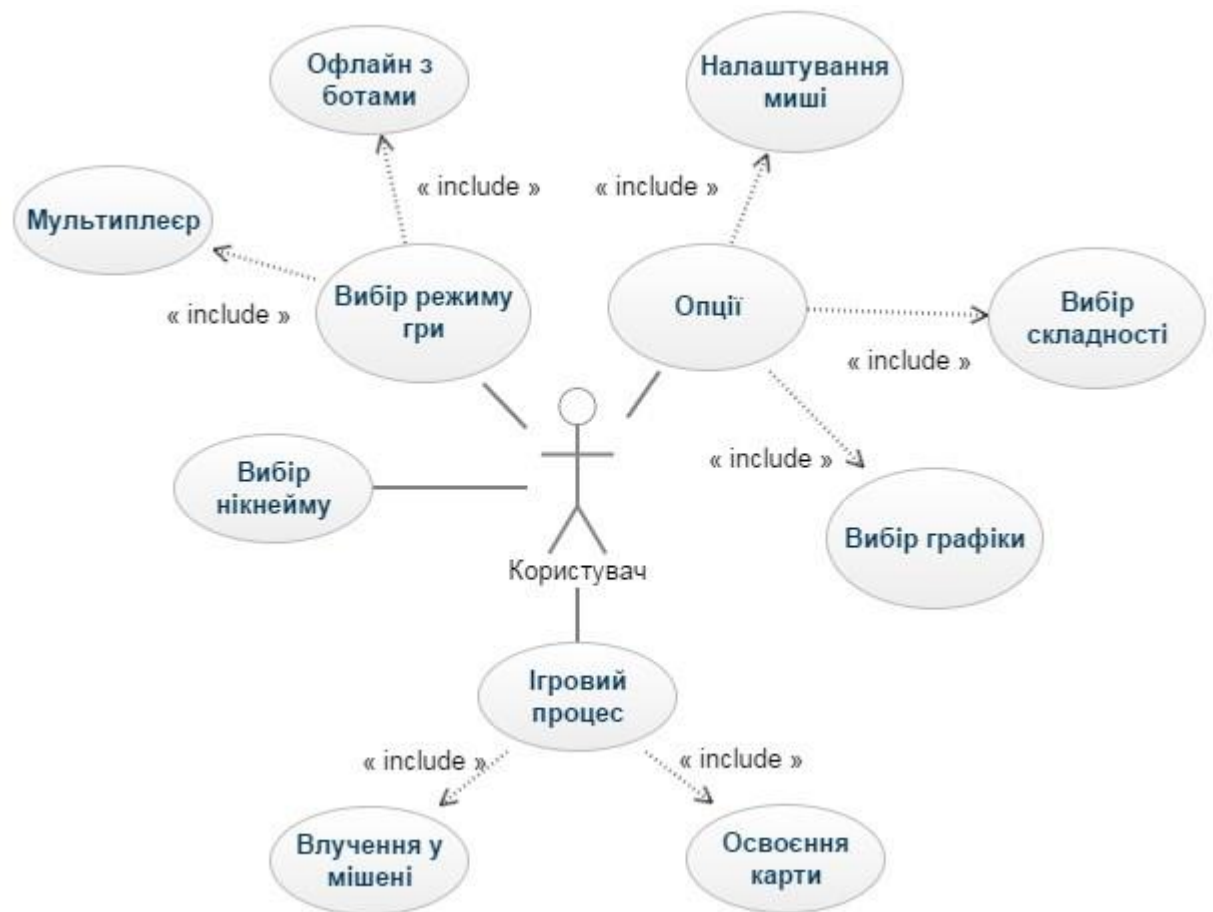


Рисунок 2.3 – Діаграма використання можливостей користувача

Інформаційне забезпечення.

Інформаційне забезпечення контролера персонажу зображено на рис. 2.4. Вхідними даними для контролера є дії користувача – натиснені клавіші та рухи миші. Після обробки даних на вихід подаються координати для руху персонажу у просторі.

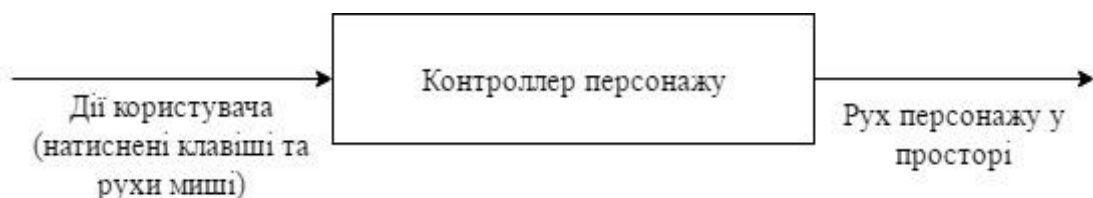


Рисунок 2.4 – Інформаційне забезпечення контролера персонажу

2.3 Висновки до розділу

1. Визначено функціональні вимоги до комп'ютерної гри.
2. Розроблено структуру гри та її модулі.
3. Розроблено діаграму використання можливостей користувача, яка відображає основні дії при запуску ігрового модуля.

РОЗДІЛ 3. ПРОЕКТНА ЧАСТИНА ТА МЕТОДИ

1.1 Первинні налаштування

Створення моделі середовища.

Модель середовища створювалася у програмі 3d-моделювання Blender:

В першу чергу було створено модель Plane (по стандарту 1x1 в метрах) та розширено її на 25 метрів, як показано на рис. 3.1:

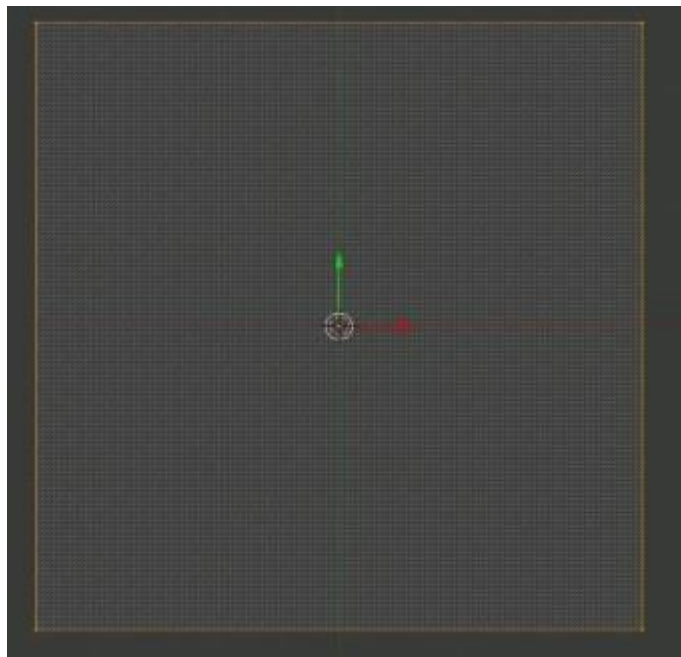


Рисунок 3.1 – Модель Plane

Далі було використано функцію Subdivide, щоб розділити її по областним міркам 10x10 частин та виділення деяких з них для зміщення вниз, щоб створити 3d-модель, як показано на рис. 3.2:

Кафедра РПЗ				КІТЗ НАУ 17 47.000 ПЗ						
Виконав	Рабін І.Р.			ПРОЕКТНА ЧАСТИНА ТА АЛГОРИТМИ			Літера	Аркуш	Аркушів	
Керівник	Нечипорук В. В.						Д		25	60
Консульт.							РПЗ – 48			
Норм. контр.	Проценко С. М.									
Голова комісії.	Апенько Н.В.									

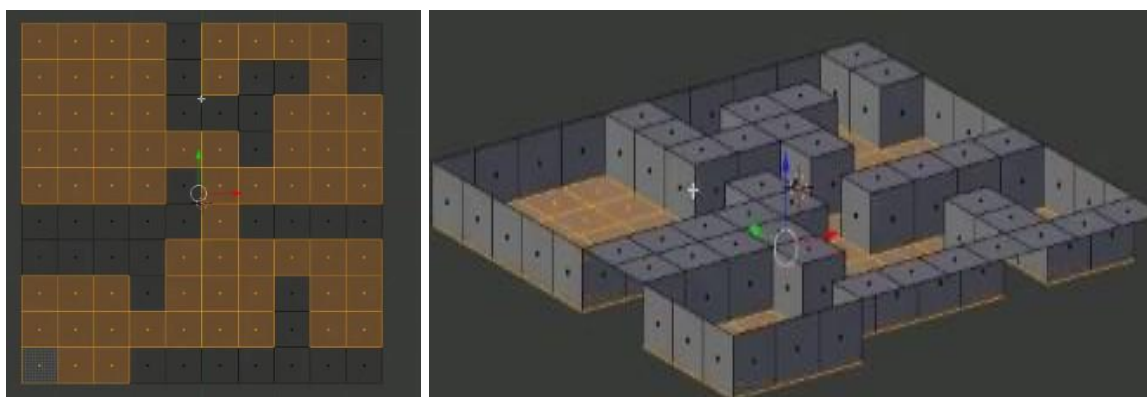


Рисунок 3.2 – Створення кімнат

На цьому рисунку показано створення кімнати за допомогою функції Extrude (зміщення вниз виділених областей).

Після цього модель було збережено у форматі *.blend та інтегровано його у Unity3D, як показано на рис. 3.3:

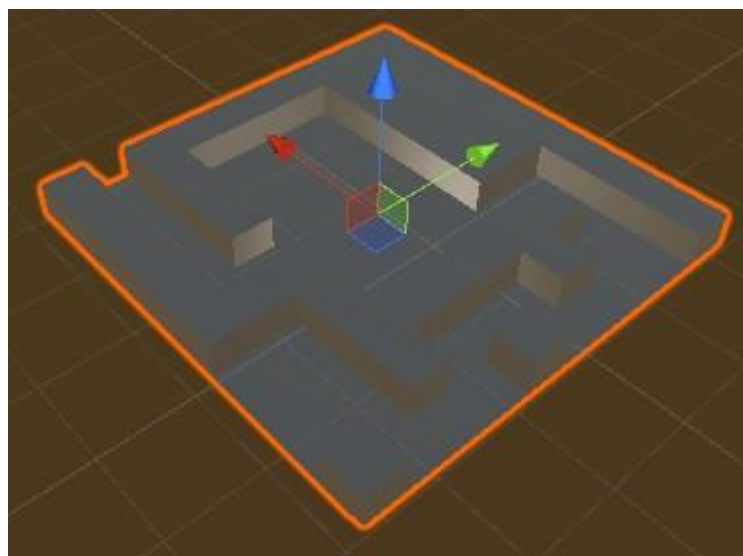


Рисунок 3.3 – Модель інтегрована у Unity3D

Створення матеріалів.

За допомогою Blender-а було створено лампу, яка потім буде ціллю для пострілів, як показано на рис. 3.4:



Рисунок 3.4 – Лампа

Потім було створено модель кулі у Unity3D шляхом використання компоненту «Сфера» з додаванням кольору та вбудованого світла, як показано на рис. 3.5:

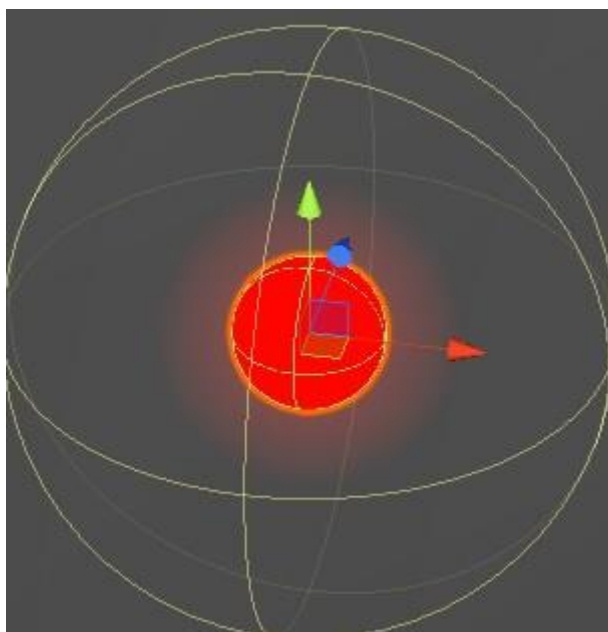


Рисунок 3.5 – Модель кулі

Налаштування ефектів.

Об'єкт загорання fireEffect було попередньо встановлено з стандартних наборів Unity3D та налаштовано. Ефект складається з трьох частин – джерела світла, основної частини та диму. Налаштування джерела світла зображено на рис. 3.6:



Рисунок 3.6 – Налаштування джерела світла

Налаштування основної частини зображено на рис. 3.7:

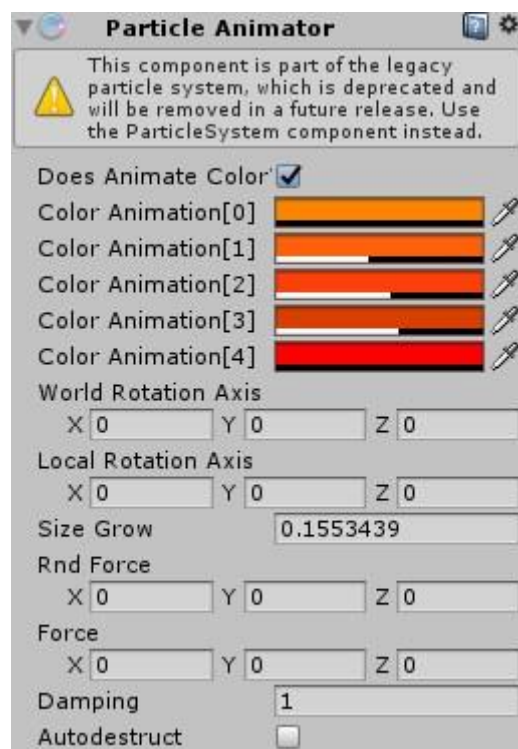


Рисунок 3.7 – Налаштування основної частини

Налаштування диму зображено на рис. 3.8:

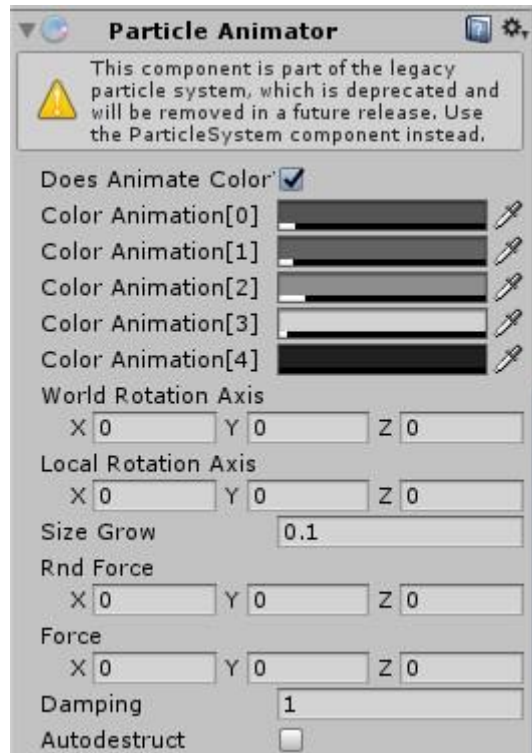


Рисунок 3.8 – Налаштування диму

Кінцевий результат ефекту вогню зображено на рис. 3.9:

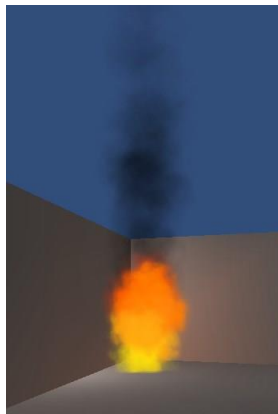


Рисунок 3.9 – Ефект вогню

Для освітлення сцени був використаний компонент «Point light» (рис. 3.10), який потім було розміщено по всій сцені.

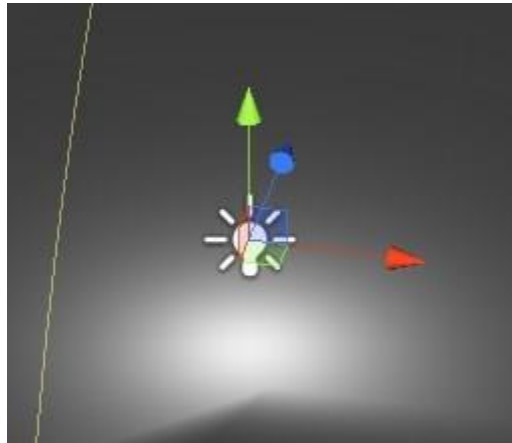


Рисунок 3.10 – Компонент Point light

Створення головного меню.

Головне меню було створено за допомогою нових сцен Unity3D, на яких було встановлено наступні компоненти (рис. 3.11):

- Plane (фон головного меню);
- Spotlight (для освітлення фону);
- камери;
- текст.

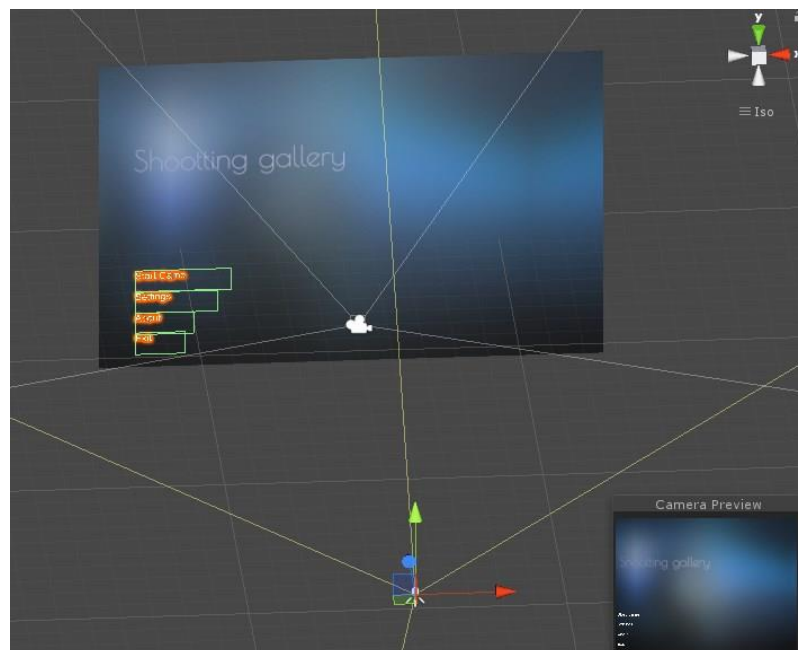


Рисунок 3.11 – Сцена з компонентами

Для інтерактивності головного меню було створено скрипт «MainMenu» в якому вказувалися реакція кнопок на їх наведення, а саме зміна кольору, як зображено на рис. 3.12:

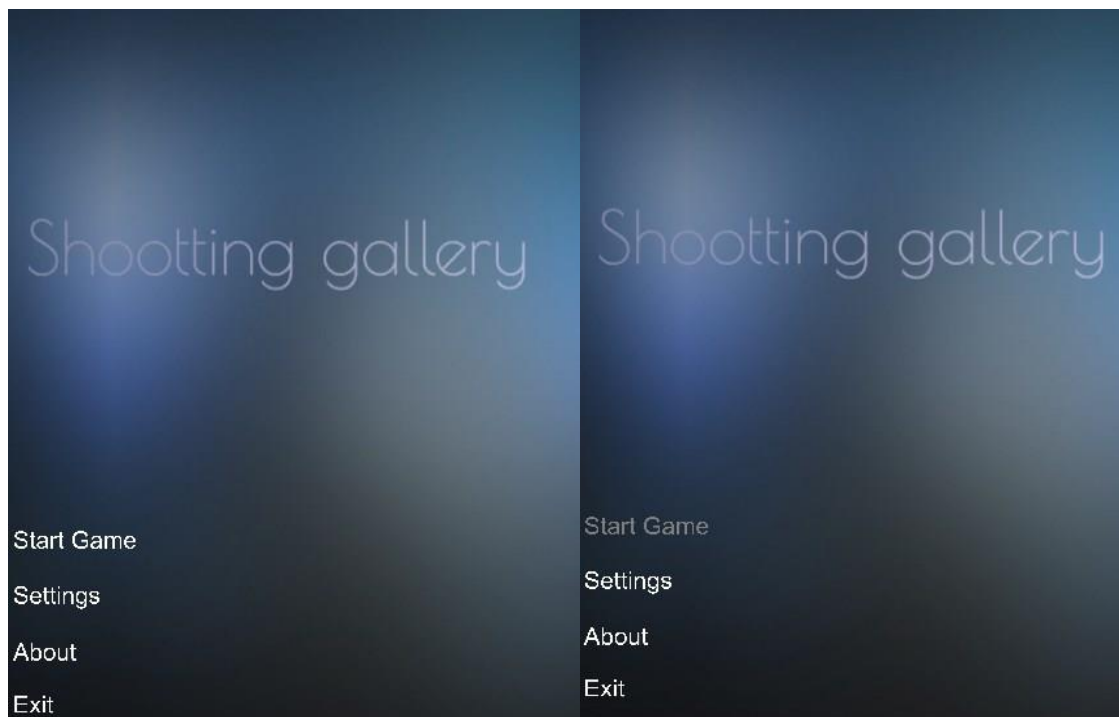


Рисунок 3.12 – Зміна кольору при наведенні

Відповідно до функціональних вимог було створено кнопки головного меню: «Start Game», «Settings», «About» та «Exit». Далі під кожную кнопку меню було створено скрипти навігації для зручного переходу між сценами, як показано на рис. 3.13:

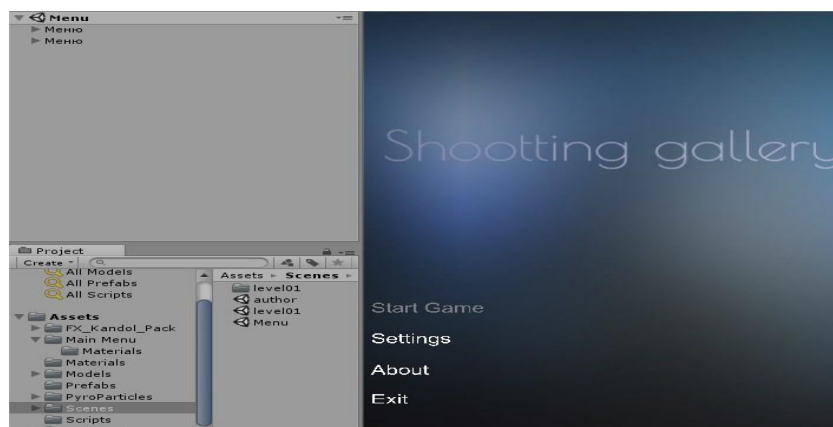


Рисунок 3.13 – Сцена головного меню «Menu»

Коли користувач натискає на кнопку «Start Game» – його переносить зі сцени «Menu» на сцену «level01», як показано на рис 3.14:

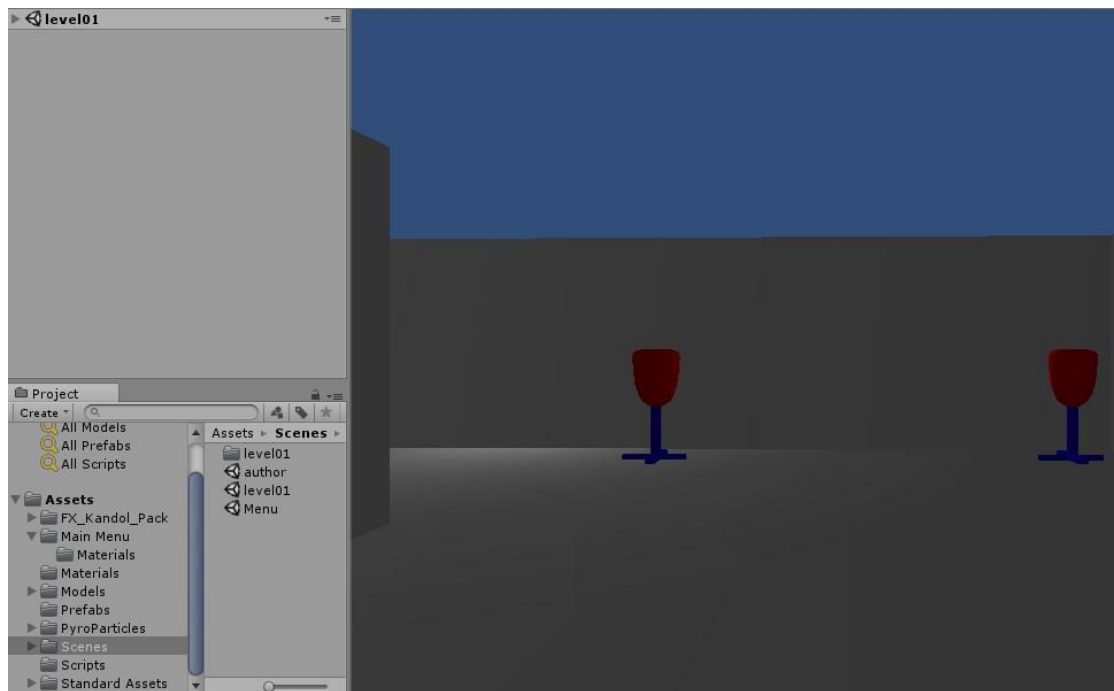


Рисунок 3.14 – Перехід на сцену «level01» за допомогою кнопки «Start Game»

3.2 Основні скрипти

Основним скриптом гри є контролер персонажу (Character Controller), наведений нижче.

Персонажу в грі від першої або третьої особи часто потрібна деяка фізика заснована на зіткненнях, щоб він не падав крізь підлогу або не ходив крізь стіни. Зазвичай, все ж прискорення і рух персонажа фізично нереалістичні, таким чином він може прискорюватися, гальмувати і змінювати напрямок миттєво, не наражаючись впливу імпульсу.

У 3D фізики цей тип поведінки можна створити використовуючи Character Controller. Цей компонент дає персонажу простий коллайдер в формі капсули, який завжди знаходиться у вертикальному положенні.

```
float rotLeftRight = Input.GetAxis ("Mouse X") * mouseSensitivity;
```



```

transform.Rotate (0, rotLeftRight, 0);
verticalRotation -= Input.GetAxis ("Mouse Y") * mouseSensitivity;
verticalRotation = Mathf.Clamp (verticalRotation,
upDownRange, upDownRange);
Camera.main.transform.localRotation = Quaternion.Euler(verticalRotation, 0, 0);
float forwardSpeed = Input.GetAxis("Vertical") * movementSpeed;
float sideSpeed = Input.GetAxis("Horizontal") * movementSpeed;
Vector3 speed = new Vector3(sideSpeed, 0, forwardSpeed);
speed = transform.rotation * speed;
CharacterController cc = GetComponent<CharacterController>();
cc.SimpleMove(speed);

```

У цьому скрипті задаються вхідні дані у вигляді дій користувача, наприклад, положення миші та натиснених клавіш, що регулюють положення камери та рух об'єкту у просторі, при цьому контролюючи швидкість, чуттєвість миші та максимальний кут огляду. Для прийняття положення миші використовується вбудований метод `Input.GetAxis`:

```

float rotLeftRight = Input.GetAxis ("Mouse X") * mouseSensitivity;
verticalRotation -= Input.GetAxis ("Mouse Y") * mouseSensitivity;

```

Рух персонажу здійснюється за допомогою методу `cc.SimpleMove(speed)`, де `cc` – об'єкт персонажа на сцені, а `speed` – обчислена оптимальна швидкість переміщення, що включає в себе переміщення по горизонталі, вертикалі та обертання:

```

Vector3 speed = new Vector3(sideSpeed, 0, forwardSpeed);
speed = transform.rotation * speed;

```

Далі була поставлена задача задати гравітацію у параметр Jump для здійснення прижків по сцені:

```
verticalVelocity += Physics.gravity.y * Time.deltaTime;  
if( characterController.isGrounded && Input.GetButton("Jump") )  
verticalVelocity = jumpSpeed;
```

Параметри швидкості, чуттєвості миші, швидкості стрибка та максимальний кут огляду можна редагувати (збільшити чи зменшити, як захоче користувач), як зображено на рис. 3.1:



Рисунок 3.1 – Параметри для редагування

Наступний скрипт задає параметри кулі для стрільби від першої особи. Спочатку було створено шаблон кулі для його подальшого використання при стрільбі:

```
public GameObject bullet_prefab;  
float bulletImpulse = 20f;
```

Сам постріл відбувається при натисненні кнопки «LeftCtrl», після чого встановлюється початкове положення кулі, враховуючи положення камери:

```
void Update ()  
{  
if( Input.GetButtonDown("Fire1") )
```

```

    {
    Camera cam = Camera.main;
    GameObject thebullet = (GameObject)Instantiate(bullet_prefab,
    cam.transform.position + cam.transform.forward, cam.transform.rotation);
    thebullet.rigidbody.AddForce( cam.transform.forward * bulletImpulse,
    ForceMode.Impulse);
    }
}

```

Далі була поставлена задача ініціалізувати шаблон кулі з параметрами сили та рухом у просторі.

Наступний скрипт задає значення періоду життя куль після пострілу та реагує на об'єкти сцени з тегом «Enemy», покриваючи їх полум'ям:

```

float lifespan = 3.0f;
public GameObject fireEffect;
void Update ()
{
    lifespan -= Time.deltaTime;
    if(lifespan <= 0)
    Destroy(gameObject);
}
void OnCollisionEnter(Collision collision)
{
    if(collision.gameObject.tag == "Enemy")
    {
        collision.gameObject.tag = "Untagged";
        Instantiate(fireEffect, collision.transform.position, Quaternion.identity);
        Destroy(gameObject);
    }
}
}

```

Наступний скрипт MainMenu було створено для зміни кольору кнопок при наведенні курсора на них:

```
Renderer clr;  
void OnMouseEnter()  
{  
    clr = GetComponent<Renderer>();  
    clr.material.color = Color.red;  
}  
void OnMouseExit()  
{  
    clr = GetComponent<Renderer>();  
    clr.material.color = Color.white;  
}
```

Далі було створено скрипти, прив'язані до кнопок для зручної навігації по сценам. Для переходу на сцену гри до кнопки «Start Game» було прив'язано наступний скрипт:

```
void OnMouseDown ()  
{  
    SceneManager.LoadScene("level01");  
}
```

Тепер для переходу у параметри гри до кнопки «Settings» було прив'язано наступний скрипт:

```
public Camera camera1;  
public Camera camera2;  
void OnMouseDown ()
```

```
{  
    camera1.enabled = false;  
    camera2.enabled = true;  
}
```

До наступної кнопки «About», яка буде застосована для переходу на сцену з даними розробника, було використано скрипт:

```
void OnMouseDown ()  
{  
    SceneManager.LoadScene("author");  
}
```

Потім до кнопки виходу зі гри «Exit» було створено наступний скрипт:

```
void OnMouseDown ()  
{  
    Application.Quit ();  
}
```

Останньою кнопкою до якої було прив'язано скрипт стала кнопка «Back», за допомогою якої користувач мав змогу повернутися з меню параметрів у головне меню:

```
public Camera camera1;  
public Camera camera2;  
void OnMouseDown ()  
{  
    camera1.enabled = true;  
    camera2.enabled = false; }  

```

3.3 Висновки до розділу

1. Створено модель середовища за допомогою Blender-a.
2. Створено компоненти середовища та візуальні ефекти перетворення.
3. Реалізовано контроллер персонажу, який задає первинні параметри.
4. Реалізовано метод FP_Shooting, який управляє пострілами при натисненні заданої клавіші.
5. Реалізовано метод для управління життєвим циклом кулі та її детонування при попаданні в об'єкти певного типу.
6. Створено навігацію між сценами та швидкий вихід до головного меню.

РОЗДІЛ 4. КЕРІВНИЦТВО КОРИСТУВАЧА

4.1 Вимоги до ігрового модуля

Для правильного функціонування ігрового модуля потрібно установити операційну систему Windows 7 з архітектурою x64. Мінімальні вимоги до апаратного забезпечення:

Мінімальні системні вимоги:

1. Процесор – 1.0 ГГц.
2. ОЗУ – 2 Гб.
3. Вільний простір на диску – 150 Мб.
4. Звукова карта – Conexant SmartAudio HD.
5. Microsoft .NET Framework 4.5.
6. Мишка.
7. Клавіатура.
8. Монітор.
9. Відеокарта NVIDIA GeForce 635M.

Для запуску також необхідно мати установлений Microsoft .NET Framework версії 4.5 або вище.

4.2 Установка гри

Для установки гри користувачу необхідно зайти в архів з грою та розпакувати його (рис 4.1) у заздалегідь встановлену папку, після чого можна починати гру, подвійно натиснувши на файл Shooting gallery.exe (рис 4.2).

Кафедра РПЗ				КІТЗ НАУ 17 47.000 ПЗ				
Виконав	Рабін І.Р.			КЕРІВНИЦТВО КОРИСТУВАЧА	Літера	Аркуш	Аркушів	
Керівник	Нечипорук В.В.				Д		39	60
Консульт.					РПЗ – 48			
Норм. контр.	Проценко С. М.							
Голова комісії.	Апенько Н.В.							

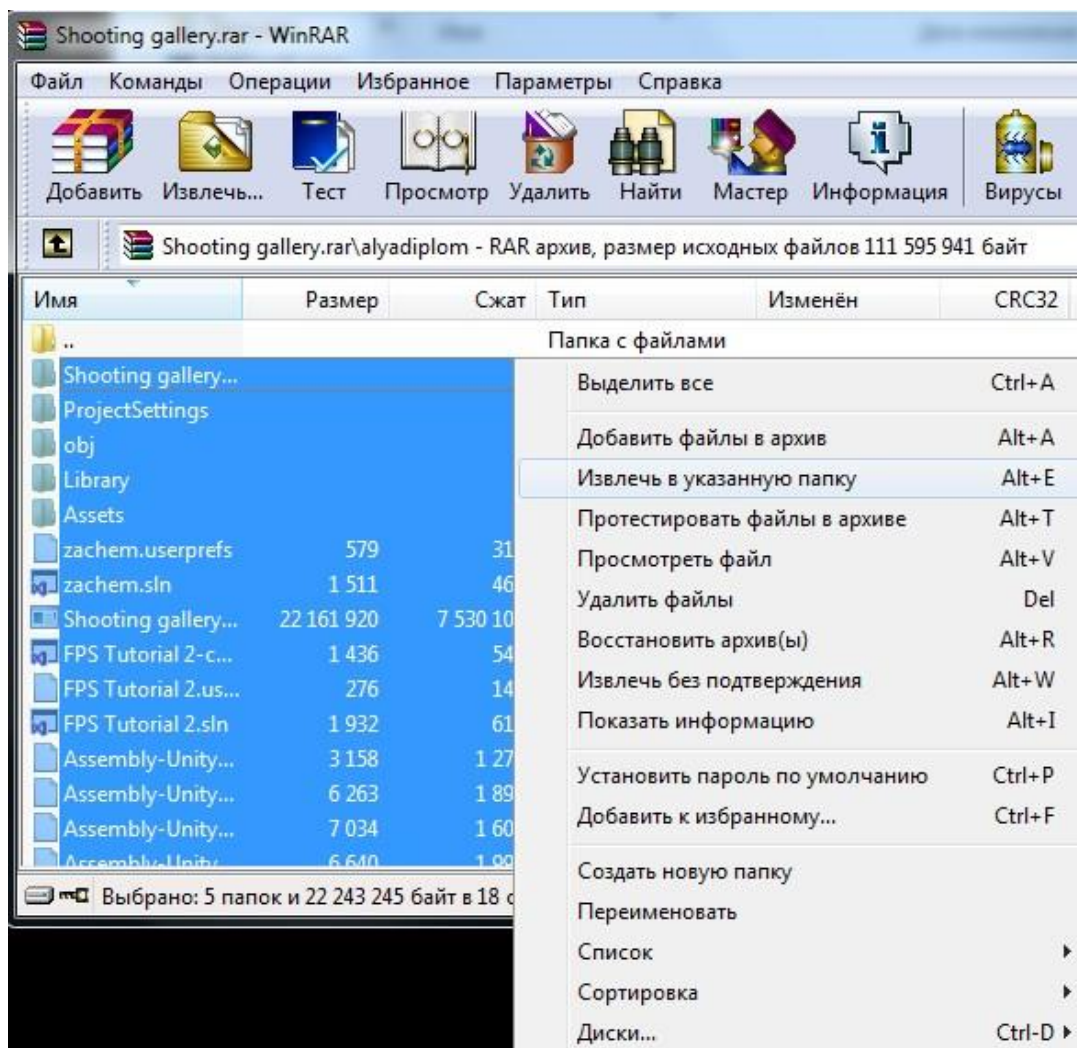


Рисунок 4.1 – Установка гри

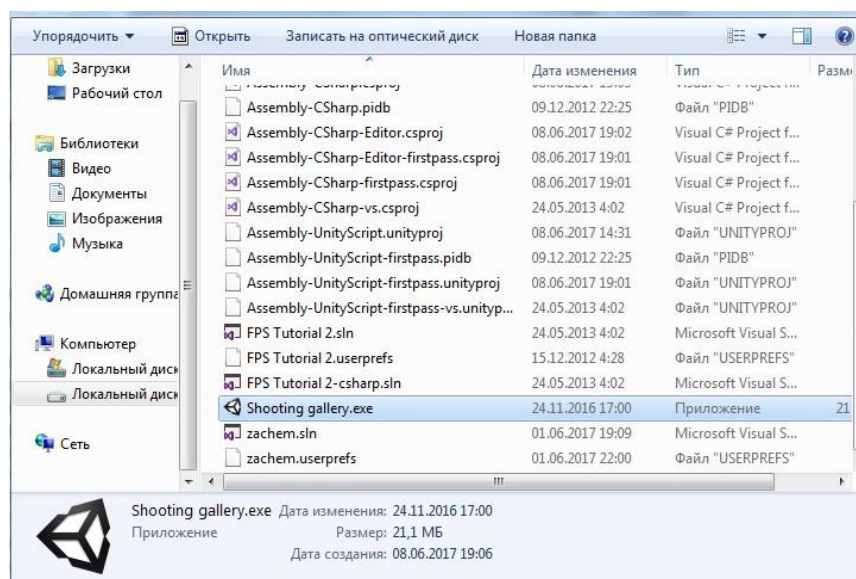


Рисунок 4.2 – Запуск гри

4.3 Ігровий процес

Запуск гри відбувається після подвійного натиснення на файл Shooting gallery.exe, після чого на екрані буде відображено вікно початкових налаштувань, в якому можна буде вибрати розширення екрану, якість зображення і необхідний монітор (рис. 4.3):

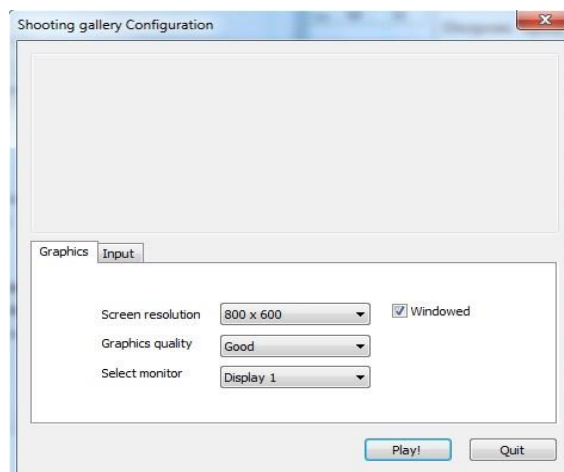


Рисунок 4.3 – Вікно початкових налаштувань

Після вибору необхідних налаштувань та завантаження гри буде відображено головне меню користувача (рис. 4.4):

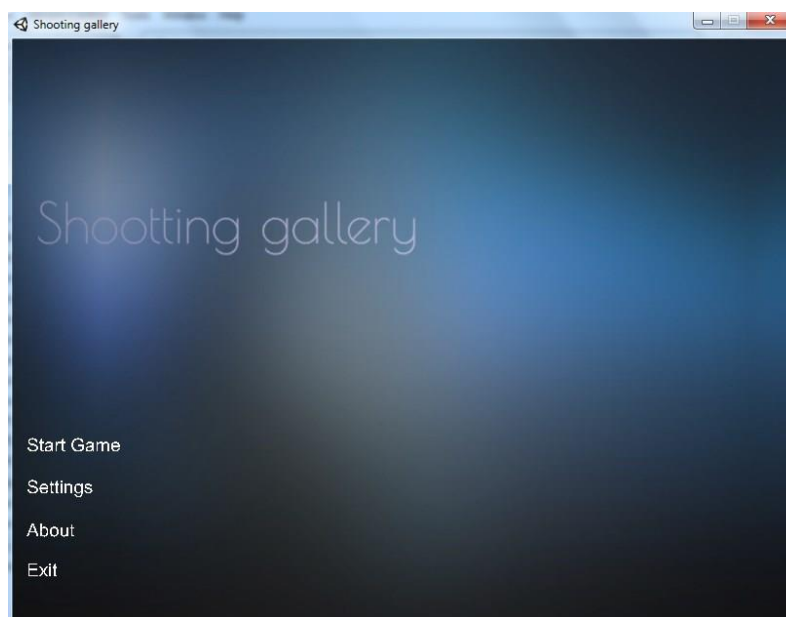


Рисунок 4.4 – Головне меню користувача

Після цього користувач може вибрати налаштування під його комп'ютер та відповідно до його вподобань, подивитись інформацію про розробника продукту та версії, почати гру або вийти з неї, як показано на рис. 4.5, рис. 4.6 та рис.4.7:

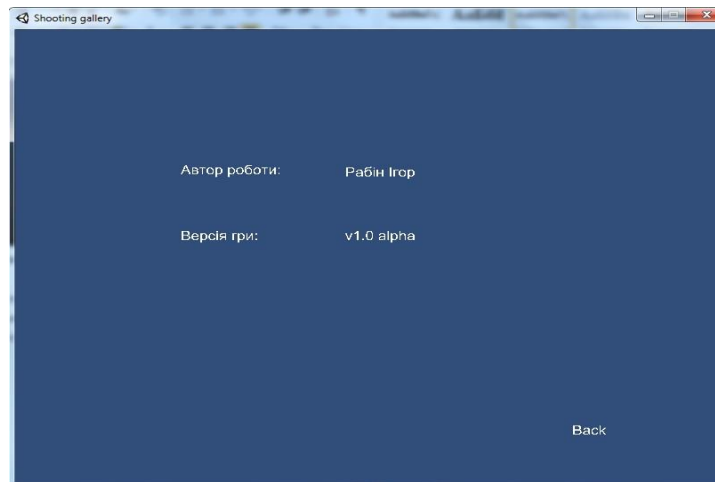


Рисунок 4.5 – Інформація про розробника та версію гри

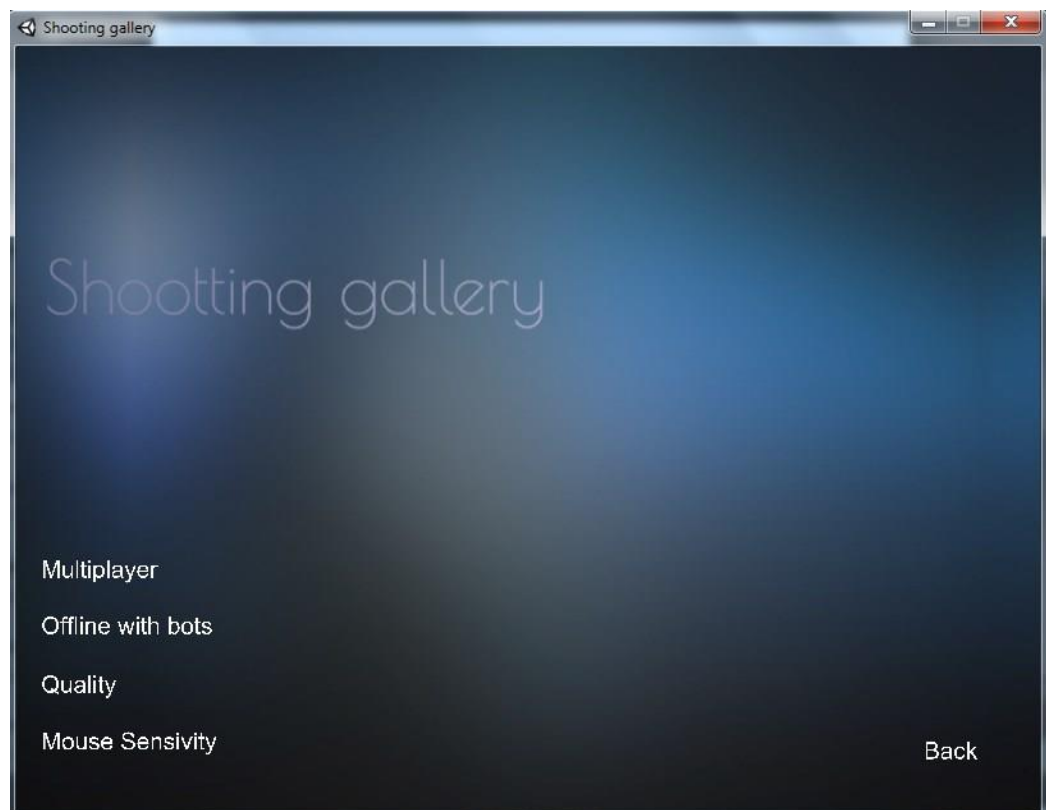


Рисунок 4.6 – Меню налаштувань

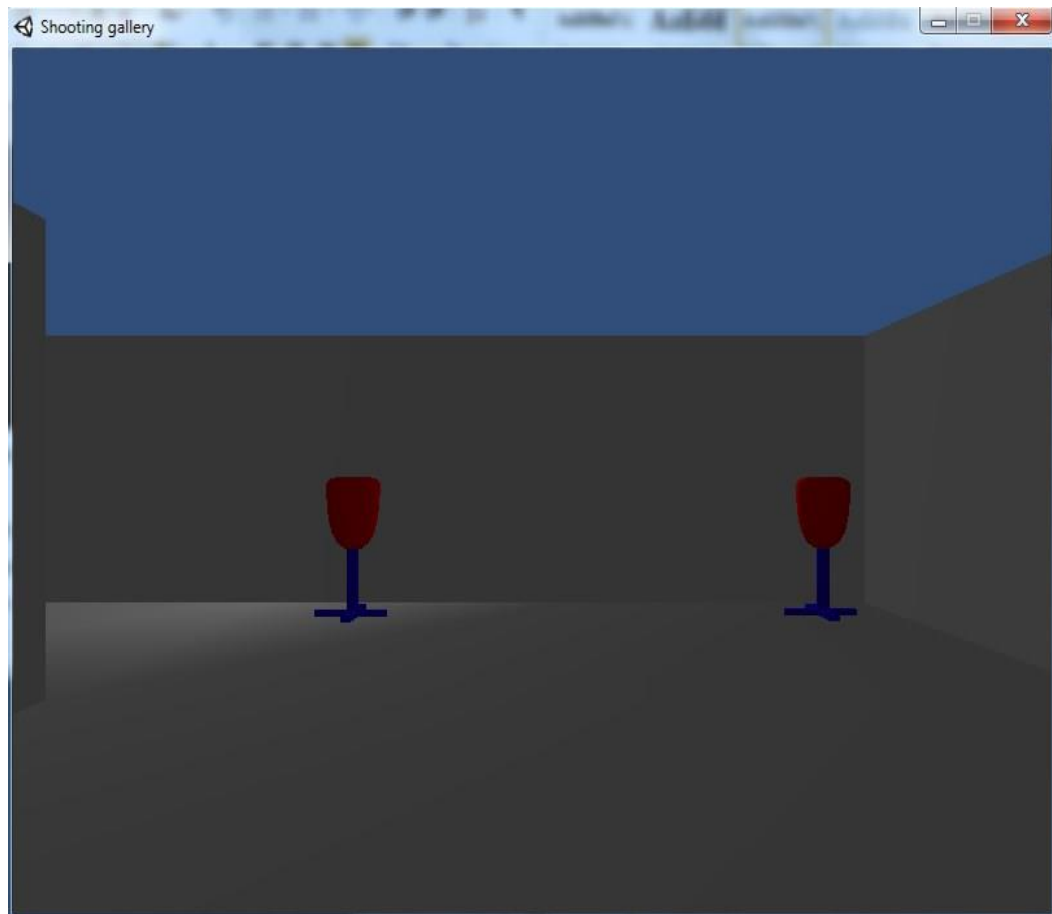


Рисунок 4.7 – Початок гри

Для переміщення по сцені використовуються кнопки W, S, A, D – відповідно вперед, назад, вліво та вправо. Також є кнопка для прижків – Space. Переміщення по середовищу зображено на рис. 4.8 та 4.9:

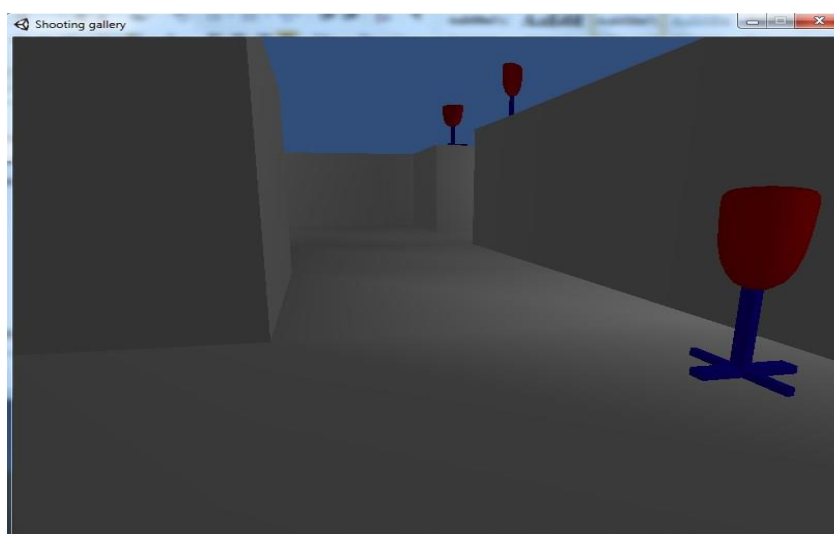


Рисунок 4.8 – Початкове положення персонажу

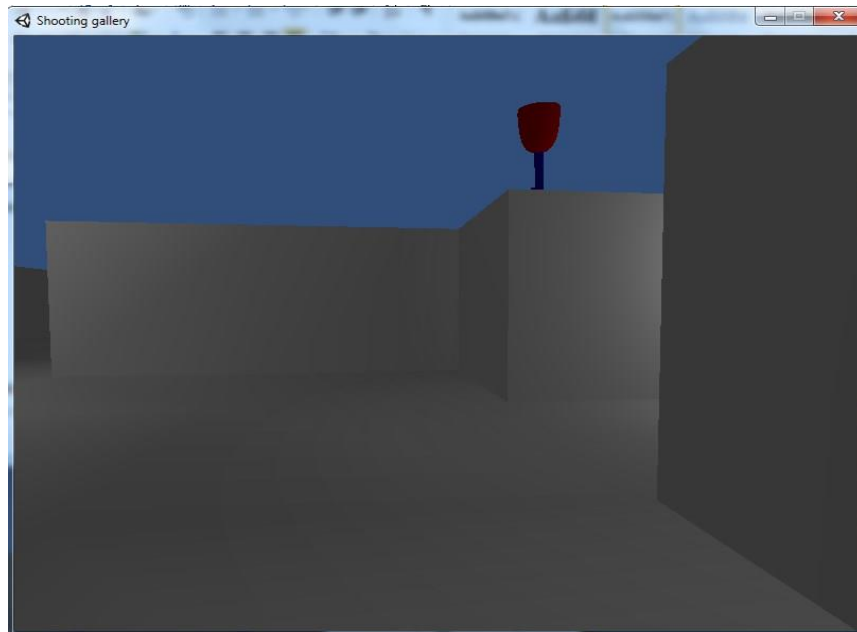


Рисунок 4.9 – Положення персонажу після переміщення по середовищу

Ігровий процес заключається в тому, щоб користувач ходив по ігровому середовищу та влучав у мішені, які зображено на рис. 4.10:

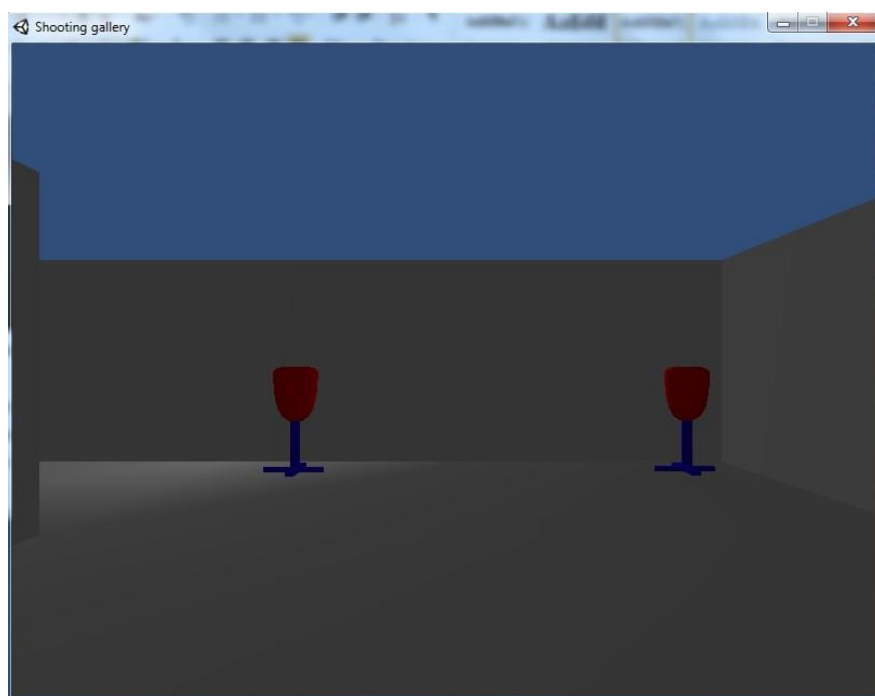


Рисунок 4.10 – Мішені

Після влучення у мішені вони будуть загоратися, що засвідчує влучне попадання. Приклад загорання мішені зображено на рис. 4.11:

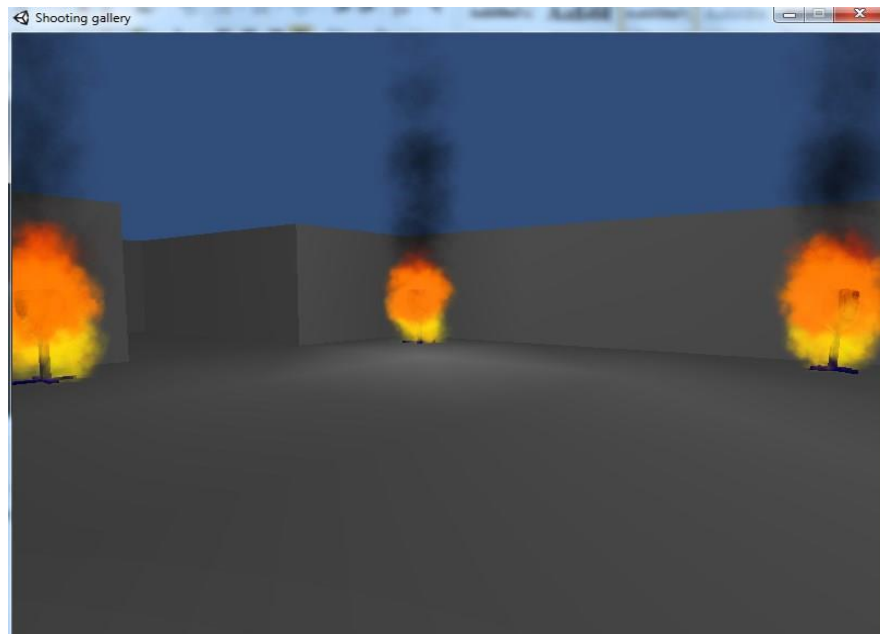


Рисунок 4.11 – Мішені після влучення у них

Сам постріл відбувається при здійсненні натискання на ліву кнопку миші, після чого вилетить куля, як зображено на рис. 4.12:

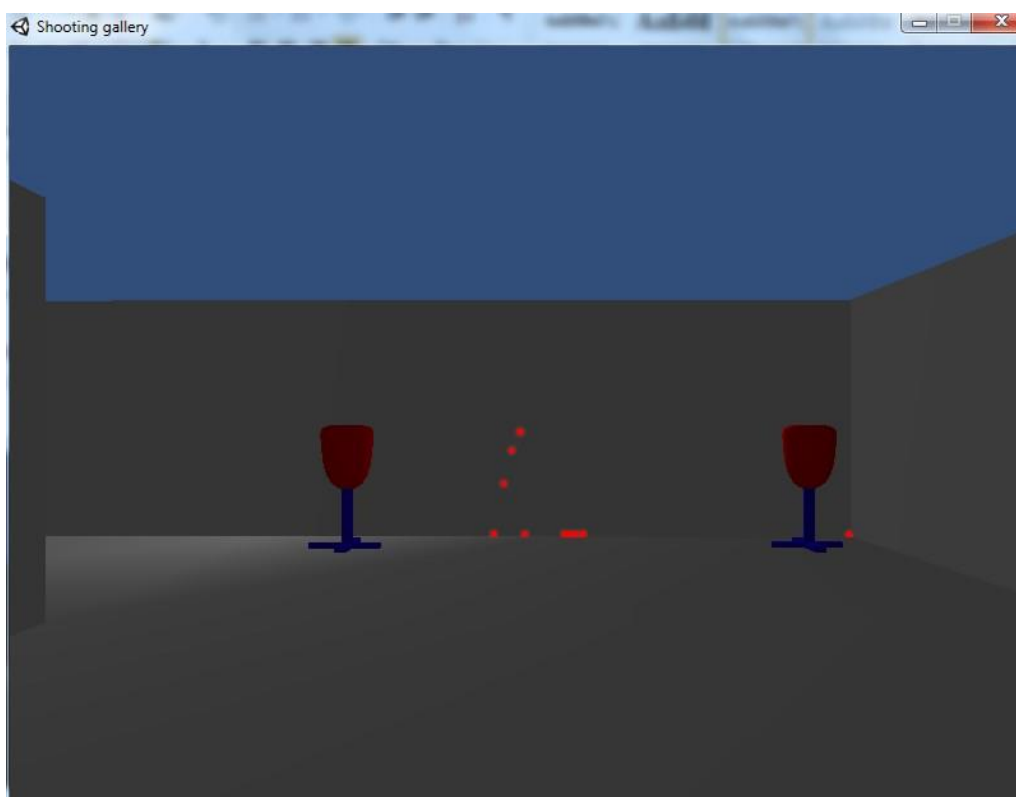


Рисунок 4.12 – Постріл

4.4 Тестування

Тестування ігрового модуля відбувалося за допомогою відкритих опитувань в соціальних мережах. У тестуванні брало участь 89 осіб.

Тестування було проведено за такими критеріями:

- інтерфейс користувача;
- функціональні можливості;
- зручність;
- стабільність;
- бажання користуватись.

Статистика відгуків відображена на рис. 4.13:

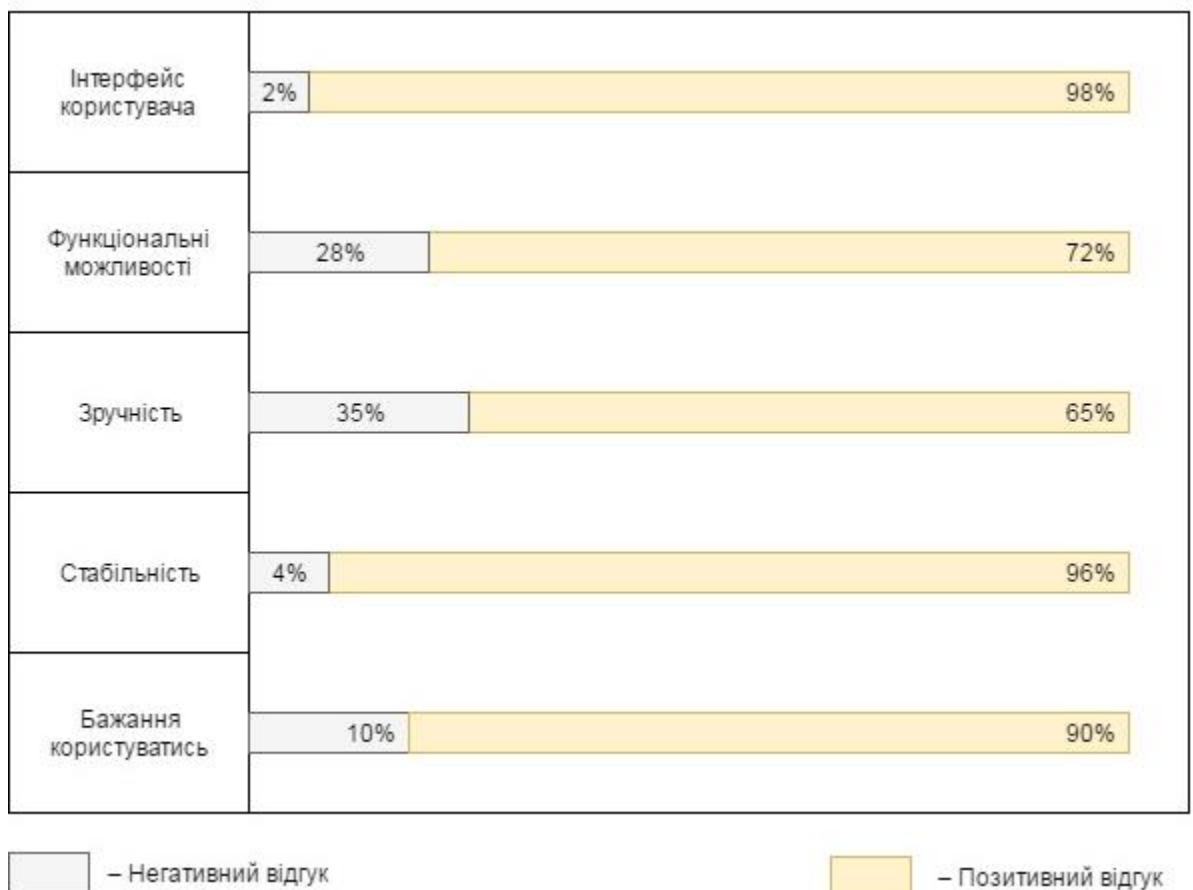


Рисунок 4.13 – Статистика тестування

Інтерфейс користувача, як довело тестування, є відмінним, але є мінус у тому, що більшість ще в процесі розробки.

Функціональні можливості ігрового модулю не так добре задовольнили користувачів, як інтерфейс, тому що в грі нема прицілу.

Зручність отримала оцінку «Задовільно», але ще є над чим міркувати та дороблювати.

Стабільність відмінна. Помилки та збоїв не було виявлено.

Бажання користуватись є не у всіх, тому що гра дуже однотипна та швидко набридає, проте є попит на майбутній версії гри.

В цілому тестування пройшло успішно і отримало оцінку «Задовільно».

Надалі гра буде вдосконалюватись за допомогою цих відгуків.

4.5 Висновки до розділу

1. Визначено мінімальні системні вимоги до ігрового модуля.
2. Протестовано ігровий модуль на різних пристроях при різних навантаженнях. Після тестування було підтверджено вимоги стабільності, точності, та зручності.
3. Результатами тестування є зміни в інтерфейсі користувача, функціональних можливостях та зручності.
4. Визначено основні напрями змін для майбутніх версій ігрового модуля.

РОЗДІЛ 5. ОХОРОНА ПРАЦІ

Вступ

Охорона праці робітників здобуває в даний час усе більш важливе значення. Це виражається необхідністю підвищення ефективності виробництва, обумовленою продуктивністю праці.

Згідно ДСТУ 2293-99, що набув чинності від 2000-01-01, визначено наступні терміни та визначення основних понять:

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження здоров'я і працездатності людини в процесі праці.

Поняття охорони праці фактично розкриває головні напрямки, які створюють систему забезпечення безпеки життя і здоров'я працівників у процесі їх трудової діяльності, тобто ця система вміщує в собі заходи, які поодинокі або в сукупності спрямовані на створення умов праці, що відповідають вимогам збереження життя та здоров'я працівників в процесі трудової діяльності.

Безпечні умови праці, безпека праці – стан умов праці, за якого вплив на працівника небезпечних і шкідливих виробничих чинників усунуто, або вплив шкідливих виробничих чинників не перевищує гранично допустимих значень.

Небезпечний (виробничий) чинник – виробничий чинник, вплив якого на працівника в певних умовах призводить до травм, гострого отруєння або іншого раптового різкого погіршення здоров'я або до смерті.

Кафедра РПЗ				КІТЗ НАУ 17 47.000 ПЗ				
Виконав	Рабін І.Р.			ОХОРОНА ПРАЦІ	Літера	Аркуш	Аркушів	
Керівник	Нечипорук В.В.				Д		48	60
Консульт.	Лазарчук Є.П				РПЗ – 48			
Норм. контр.	Проценко С.М							
Голова комісії.	Апенько Н.В.							

Шкідливий (виробничий) чинник – виробничий чинник, вплив якого за певних умов може призвести до захворювання, зниження працездатності і (або) негативного впливу на здоров'я нащадків.

Примітка. Залежно від кількісної характеристики (рівня, концентрації тощо) і тривалості впливу шкідливий виробничий чинник може стати небезпечним.

Травма – порушення анатомічної цілісності організму людини або його функцій внаслідок дії чинників зовнішнього середовища.

Виробнича травма – травма що сталася внаслідок дії виробничих чинників.

Нещасний випадок – непередбачений збіг обставин і умов, за яких заподіяна шкода здоров'ю, або настала смерть людини.

Нещасний випадок на виробництві – раптове погіршення стану здоров'я чи настання смерті працівника під час виконання ним трудових обов'язків внаслідок короткочасного (тривалістю не довше однієї робочої зміни) впливу небезпечного або шкідливого чинника.

Умови праці – сукупність чинників виробничого середовища і трудового процесу, які впливають на здоров'я і працездатність людини під час виконання нею трудових обов'язків.

Медичний огляд – огляд працівників спеціальною комісією лікарів з обов'язковими лабораторними, клінічними і функціональними дослідженнями з метою визначення можливості допущення до виконання конкретної роботи (до професії) за станом здоров'я.

Працездатність – здатність людини виконувати певну роботу, яка визначається рівнем її фізичних і психофізіологічних можливостей, а також станом здоров'я і професійною підготовленістю.

Непрацездатність – повна чи часткова втрата загальної або професійної працездатності внаслідок захворювання, нещасного випадку або вродженої фізичної вади.

5.1 Небезпечні та шкідливі виробничі фактори, що діють у робочій зоні

Метеорологічні умови.

Мікроклімат виробничих помешкань і стан повітряного середовища в робочій зоні – головні чинники, що визначають умови праці. Основні параметри метеорологічних умов – температура, швидкість руху і запиленість повітря, барометричний тиск впливають на теплообмін і загальний стан організму людини.

Робочою зоною вважається простір висотою до 2 м над рівнем підлоги або майданчику, де знаходиться робоче місце. Метеорологічні умови в робочій зоні виробничих помешкань повинні задовольняти вимогам ГОСТ 12.1.005-88, ССБП, «Санітарно-гігієнічні вимоги до повітря робочої зони». До помешкання диспетчерського пункту висуваються наступні вимоги:

1. Температура повинна бути в межах від 16-25 °С.
2. Оптимальні значення відносної вологості 40-50%, припустимі - не більш 75%.
3. Помешкання повинно бути сухим, з постачанням вентиляції.
4. Повітря не повинне містити часток пилу й інших шкідливих домішок.

Відносна вологість характеризує вологість повітря при визначеній температурі. Вологість повітря впливає на теплообмін в організмі людини, в основному, на віддачу тепла випаром. Середній рівень відносної вологості 40-60% відповідає умовам метеорологічного комфорту при спокої або при дуже легкій фізичній роботі.

Небезпека ураження електричним струмом.

Електричний струм робить наступні види впливу на організм людини: термічне, електрохімічне, біологічне і механічне.

У різних приладах небезпека поразки струмом різна, тому що параметри електроенергії, умови експлуатації електроустаткування, характер середовища помешкань, у яких воно встановлено, можуть бути дуже різноманітними.

Мікроклімат приміщення, в якому знаходиться ПК, впливає на електроізоляцію приладів, електричний опір тіла людини може створювати умови для ураження обслуговуючого персоналу електричним струмом.

В цьому відношенні розрізняють виробничі приміщення з підвищеною безпекою, особливо небезпечні, без підвищеної безпеки.

Аналізуючи характеристики приміщення, в якому встановлено ПК приходимо до висновку, що це приміщення без підвищеної безпеки. Електричні установки, до яких відноситься практично все обладнання, що являє для людини потенційну небезпеку, бо в процесі експлуатації або профілактичних робіт людина може торкнутися частин, які знаходяться під напругою, в результаті пошкодження (пробою) ізоляції, не подає будь-яких сигналів, що попереджали б людину про небезпеку.

Реакція людини на електричний струм виникає лише при його протіканні крізь тіло людини.

У відповідності з вимогами ГОСТ 12.1 019-19 «Электробезопасность. Общие требования» і ГОСТ 12.1 038-82 «Гранично допустимые уровни напряжений прикосновений токов» захист від небезпечного впливу експлуатації ПК забезпечується:

- застосування захисного заземлення;
- ізоляцією струмопровідних частин;
- огорожа струмопровідних частин, що відвертають випадковий дотик до них;
- вибором проводів, їх згуповкою, трасировкою і так далі.

Обслуговуючий персонал повинен вивчати правила по ТБ і вміти виявляти першу допомогу постраждалому від поразки електричним струмом.

Шум.

Шум – безладне сполучення небажаних для людини звуків, що заважають трудовій діяльності або відпочинку. Шум негативно впливає на нервову систему або ушкоджує слуховий апарат. У умовах шуму підвищується кров'яний тиск, спостерігається підвищена дратівливість, що приводить до зниження фізичної працездатності, притупленню уваги.

Іонізуючі випромінювання.

Іонізуюче випромінювання – будь-яке випромінювання, взаємодія якого із середовищем приводить до утворення електричних зарядів різних знаків. По своїй природі іонізуючі випромінювання можуть бути корпускулярними (альфа, бета, нейтронне), або електромагнітними (гама, рентгенівське, космічне).

Джерелами електромагнітних полів є: атмосферна електрика, радіовипромінювання, електричні і магнітні поля Землі, штучні джерела (монітори обчислювальних комплексів, радіомодеми, телебачення і т.п.). Джерелами іонізуючих випромінювань – електровакуумні прилади.

5.2 Організаційні і технічні заходи для виключення небезпечних і зменшення рівня шкідливих виробничих факторів

Нормалізація освітлення.

Щоб виключити найбільш небезпечний фактор – враження людини електричним струмом у розробленому пристрої основні вузли та деталі вкриті кожухами, електропроводка виконується у відповідності до умов навколишнього середовища.

Січення проводів обирається за допустимою густиною струму. У пристрої ізоляція проводів відповідає номінальній напрузі в сіті, а захисна оболонка – умовам навколишнього середовища та способу прокладки проводів.

При технічному обслуговуванні усі роботи виконуються з повністю знятою напругою живлення. При цьому необхідно прийняти міри запобігаючи помилковому вимкненню апаратури. До початку роботи необхідно впевнитися у відсутності напруги. Для цього використовують вольтметр типу ТП-4М, вказівник напруги типу УНН-10 з діапазоном вимірюваної напруги 150 В.

Особливу увагу необхідно приділити важливому з точки зору виробничій санітарії питанню освітлення на робочому місці.

Виробниче освітлення регулюється нормативно-технічним документом СНтаП II-4-79. Освітлення на робочому місці повинно бути поєднаним (штучне та природне світло). Природне освітлення повинно бути боковим. При виконанні робіт з категорії високої зорової точності коефіцієнт природної освітленості

повинен відповідати нормативним рівням по СНтаП II-4-79 (не нижче 1,5), при зоровій роботі середньої точності – не нижче 1.

Штучне освітлення слідую здійснювати у вигляді комбінованої системи освітлення з використанням люмінесцентних джерел світла в світильниках загального освітлення. Вони повинні забезпечувати рівномірне освітлення за допомогою відображеного або розсіяного світло розподілення.

Визначимо норму загального штучного освітлення (кількість необхідних світильників) для забезпечення необхідної освітленості приміщення, застосувавши метод коефіцієнта використання світлового потоку. Загальна розрахункова формула має вигляд:

$$F = \frac{E_{\min} \cdot S \cdot K_z \cdot z}{N \cdot \eta \cdot n}, \quad (5.1)$$

де F – світловий потік лампи в світильнику, Лм;

E_{\min} – норма (мінімум) освітленості, Лк;

S – площа приміщення, м²;

K_z – коефіцієнт запасу, $K_z=1,3$;

z – коефіцієнт нерівномірності освітленості; $z=0,45$;

N – число світильників, яке визначається з умови рівномірності освітлення;

η – коефіцієнт використання світового потоку;

n – кількість ламп в світильнику.

Коефіцієнт використання світлового потоку η залежить від типу світильника, коефіцієнтів відображення від підлоги, стін, стелі та індексу приміщення λ , який визначається по формулі:

$$\lambda = \frac{A \cdot B}{H_p \cdot (A + B)}, \quad (5.2)$$

де A та B – довжина та ширина приміщення, м;

H_p – висота підвісу світильника над робочою поверхнею, м.

H_p обчислюється по формулі:

$$H_p = H - h_C - h_N, \quad (5.3)$$

де H – висота приміщення, м;

h_c – висота світильника, м;

h_N – висота робочої поверхні, м.

Для розрахункового випадку маємо: довжина робочого кабінету складає 6 м; ширина – 4 м; висота – 3 м.

Приміщення освітлюється світильниками типа ЛСП01, в кожному з котрих знаходиться по дві лампи ЛБ. Висота робочої поверхні складе 0,8 м, висота світильника – 0,275 м.

Визначимо висоту підвісу світильників, підставивши вихідні значення в формулу (8.3):

$$H_p = 3.0 - 0.275 - 0.8 = 1.925 \text{ (м)}.$$

Далі визначимо значення індексу приміщення λ , підставляючи в формулу (5.2) значення H_p : $\lambda=1,2$.

По індексу приміщення та коефіцієнтам світлового потоку від підлоги – 30% (0,3), від стін – 40% (0,4) та від стелі – 60% (0,6) визначаємо для світильника ЛСП01 значення коефіцієнта використання світлового потоку (η). $\eta=0,52$.

Згідно вимогам СНтаП II-4-79 норма (мінімум) освітленості при загальному в системі комбінованого освітлення складає 400 лк. Світловий потік лампи ЛБ-80 складає 4960 лм.

З формули (5.1) виразимо значення кількості світильників N та підставляючи відомі значення в вираз одержимо $N=1,1$. Округляючи значення до більшої цілої цифри, отримуємо, що вимагається 2 світильника.

Висновок: при достатньому природному освітленні (світла пора доби, ясна погода) кількості та сумарної площі світлових щілин достатньо для забезпечення необхідної освітленості робочого приміщення.

У випадку недостатності природного освітлення необхідно задіяти джерела штучного освітлення (розрахунок показав, що достатньо мати 2 світильника з лампами ЛБ-80 та чистими плафонами).

Підвищений рівень шуму на робочому місці.

Підвищений рівень шуму негативно діє на організм людини та призводить до швидкої втоми, зниження уваги, що може призвести до травмування працівника. Рівень шуму не повинен перевищувати 65 ДБ(ГСТ 12.1.003-83).

Захист від іонізуючого випромінювання.

Основна задача при забезпеченні радіаційної безпеки перебуває в тому, щоб не допустити можливість випромінювання вище гранично припустимої. Вона вирішується шляхом застосування комплексу організаційних і технічних заходів, у тому числі захистом: часом, відстанню, екранами і кількістю.

Захист часом досягається в результаті відповідної підготовки та організації робіт, упорядкування і дотримання графіків, при яких час контакту з джерелом випромінювання мінімально, а продуктивність праці залишається на досить високому рівні. Але найбільш ефективним заходом захисту є екранування. Систематичний вплив електромагнітних полів на працюючих, із рівнями, що перевищують припустимі, приводить до порушення стану їхнього здоров'я. При цьому можуть виникати порушення в нервовій, серцево-судинній, ендокринній і інших системах організму людини.

ВИСНОВКИ

1. Проаналізовано існуючі аналоги ігор які орієнтовані на розвиток дрібної моторики та уваги користувача. Усі вони відмінні між собою, тому було поставлено завдання створити ігровий модуль, що буде ще одним аналогом ігор у сфері «Shooter». Для роботи з контроллером персонажа було обрано мову програмування C# та платформу Unity3D. Для створення моделі середовища та його компонентів було обрано програму 3D-моделювання Blender.
2. Було визначено функціональні вимоги до комп'ютерної гри.
3. Розроблено структуру гри та її модулі.
4. Розроблено діаграму використання можливостей користувача, яка відображає основні дії при запуску ігрового модуля.
5. Створено модель середовища за допомогою Blender-a.
6. Створено компоненти середовища та візуальні ефекти перетворення.
7. Реалізовано контроллер персонажу, який задає первинні параметри.
8. Реалізовано метод FP_Shooting, який управляє пострілами при натисненні заданої клавіші.
9. Реалізовано метод для управління життєвим циклом кулі та її детонування при попаданні в об'єкти певного типу.
10. Створено навігацію між сценами та швидкий вихід до головного меню.
11. Визначено мінімальні системні вимоги до ігрового модуля.
12. Протестовано ігровий модуль на різних пристроях та при різних навантаженнях. Після тестування було підтверджено вимоги стабільності, точності, та зручності.
13. Результатами тестування є зміни в інтерфейсі користувача, функціональних можливостях та зручності.
14. Визначено основні напрями змін для майбутніх версій ігрового модуля.
15. Було здобуто навички та досвід створення 3D-моделей у середовище моделювання Blender.

16. В процесі проектування було удосконалено навички роботи з Unity3D, створенні моделей та ефектів, а також їх програмна реалізація.

17. Практичне значення результатів проектування полягає у наданні можливості розвитку дрібної моторики, швидкого реагування та зорової уваги користувача.

ВИКОРИСТАНА ЛІТЕРАТУРА

1. <https://doom.com/>
2. <https://quake.bethesda.net/>
3. <http://store.steampowered.com/app/70/HalfLife/>
4. <http://counter-strike.net/>
5. <https://metanit.com/sharp/tutorial/2.1.php>
6. <https://metanit.com/sharp/tutorial/3.1.php>
7. <https://metanit.com/sharp/tutorial/3.29.php>
8. <https://metanit.com/sharp/tutorial/1.1.php>
9. <https://github.com/Unity-Technologies>
10. <https://github.com/mono/monodevelop>
11. <https://habrahabr.ru/post/143861/>
12. Генри С. Уоррен мл. Алгоритмические трюки для программистов / Вильямс, 2014 – 512с.
13. Ликнесс Д. Приложения для Windows 8 на C# / Питербург: БХВ-Питербург , 2013 – 368 с.
14. Генри С. Уоррен мл. Алгоритмические трюки для программистов / Вильямс, 2014 – 512с.
15. Хейлсберг А., Торгерсен М., Вилтамут С., Голд П . Язык программирования C#. Классика Computers Science. 4-е изд. / Петербург: БХВ-Петербург , 2011 – 784 с.
16. Н.Вирт. Алгоритмы и структуры данных.-М.: "Мир", 1989.
17. Joe Hoking Unity in Action: Multiplatform Game Development in C# with Unity 5.
18. ГОСТ Р МЭК 60447-2000. Интерфейс человеко-машинный. Принципы приведения в действие. М., 2000. 20 с.
19. Нестеров А. Л. Проектирование АСУТП. Методическое пособие. Книга 2. –СПб. : ДЕАН, 2009. – 944 с.
20. Unity 5.x Shaders and Effects Cookbook Книга, Алан Заккони и Кеннет Ламмерс.

21. Introduction to Game Design, Prototyping, and Development: From Книга, Джереми Гибсон.
22. <https://www.youtube.com/watch?v=THnivyG0Mvo>
23. https://www.youtube.com/watch?v=UK57qdq_lak
24. <https://www.youtube.com/watch?v=KfxVFjNPE3A>
25. <https://www.youtube.com/watch?v=f8tAI0ITV6s>
26. <https://www.youtube.com/watch?v=0fGB2H1AGP8>
27. <https://www.youtube.com/watch?v=-FE4zfu-V-c>
28. Unity 5 Game Optimization Книга, Крис Дикинсон.
29. Mastering Unity Scripting Книга, Алан Торн.
30. Building an FPS Game with Unity Книга, John P. Doran.
31. Getting Started with Unity 5 Книга, Эдуард Лавьери.
32. Unity Animation Essentials Книга, Алан Торн.
33. Procedural Content Generation for Unity Game Development Книга, Райан Уоткинс.
34. Mastering Unity Shaders and Effects Книга, Джейми Дин.
35. Unity Character Animation with Mecanim Книга, Джейми Дин.
36. http://b3d.mezon.ru/index.php/Blender_Basics_4-rd_edition
37. http://b3d.mezon.ru/index.php/Blender_Basics_4-rd_edition/Chapter_1_-_The_Blender_Interface
38. http://b3d.mezon.ru/index.php/Blender_Basics_4-rd_edition/Chapter_2_-_Working_with_Viewports_%28windows%29
39. http://b3d.mezon.ru/index.php/Blender_Basics_4-rd_edition/Chapter_3_-_Creating_and_Editing_Objects
40. http://b3d.mezon.ru/index.php/Blender_Basics_4-rd_edition/Chapter_4_-_Materials_and_Textures
41. http://b3d.mezon.ru/index.php/Blender_Basics_4-rd_edition/Chapter_5_-_Setting_Up_a_World
42. http://b3d.mezon.ru/index.php/Blender_Basics_4-rd_edition/Chapter_6_-_Lighting_and_Cameras

43. http://b3d.mezon.ru/index.php/Blender_Basics_4-rd_edition/Chapter_7-_Render_Window_Settings
44. Blender. 3D-моделирование и анимация. Руководство для начинающих, Андрей Прахов.
45. Инструменты моделирования в Blender, Артём Слаква.