

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Інститут інформатики та радіoeлектроніки,
Факультет комп'ютерних наук та технологій
(повне найменування інституту, факультету)

Кафедра програмних засобів
(повне найменування кафедри)

Пояснювальна записка
до дипломного проєкту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
КОМП'ЮТЕРНОЇ ГРИ НА ОСНОВІ ІНТЕЛЕКТУАЛЬНИХ
ТЕХНОЛОГІЙ.
RESEARCH AND SOFTWARE IMPLEMENTATION OF A
COMPUTER GAME BASED ON INTELLIGENT TECHNOLOGY

Виконав: студент 2 курсу, групи КНТ-129м
спеціальності

121 Інженерія програмного забезпечення
(код і найменування спеціальності)

Освітня програма (спеціалізація)
Інженерія програмного забезпечення

Боровіков А. Г.

(прізвище та ініціали)

Керівник Олійник А.О., к.т.н., доцент
(прізвище та ініціали)

Рецензент Гофман Є.О., к.т.н.
(прізвище та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Інститут, факультет _____ ІРЕ, КНТ _____
 Кафедра _____ програмних засобів _____
 Ступінь вищої освіти _____ магістр _____
 Спеціальність _____ 121 Інженерія програмного забезпечення _____
 (код і назва)
 Освітня програма (спеціалізація) _____ Інженерія програмного забезпечення _____
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри проф., д.т.н. _____
С. О. Субботін _____
 “ _____ ” _____ 2020 року

З А В Д А Н Н Я
НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА

Боровікова Артема Григоровича

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) _____ Дослідження та програмна реалізація комп'ютерної гри на основі інтелектуальних технологій. Research and Software Implementation of a Computer Game based on Intelligent Technology _____

керівник проєкту (роботи) Олійник А.О., к.т.н., доцент _____,
 (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом закладу вищої освіти від “03” листопада 2020 року №103

2. Строк подання студентом проєкту (роботи) _____ 5 грудня 2020 року _____

3. Вихідні дані до проєкту (роботи) _____ технічне завдання _____

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) _____ 1. Аналіз предметної області. 2. Вибір та обґрунтування структури системи, яка проєктується. 3. Основні рішення щодо реалізації компонентів системи. 4. Керівництво програміста. 5. Керівництво оператора. 6. Економіко-організаційна частина. 7. Охорона праці та безпека у надзвичайних ситуаціях. _____

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) _____
Слайди презентації _____

6. Консультанти розділів проєкту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1–5	Олійник А.О., доцент		
6	Пожуєва Т.О., професор		
7	Коробко О.В., ст. викладач		
Нормо-контроль	Дейнега Л.Ю., ст. викладач		

7. Дата видачі завдання 02.10.20

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Аналіз предметної області	1 тиждень	
2	Розробка та затвердження технічного завдання	2 тиждень	Технічне завдання
3	Проектування структури системи	3-4 тиждень	Структурна схема
4	Розробка схеми функціонування системи та модулів програми	5-6 тиждень	Функціональна схема. Модулі
5	Створення програмного коду системи	7-8 тиждень	Текст програми
6	Тестування та відлагодження програми	9 тиждень	Працездатна система
7	Розробка розділів пояснювальної записки	10 тиждень	ПЗ
8	Розробка слайдів презентації	11-12 тиждень	Слайди презентації
9	Захист роботи	13 тиждень	

Студент(ка)

(підпис)

Боровіков А.Г.

(прізвище, ініціали)

Керівник проєкту (роботи)

(підпис)

Олійник А.О.

(прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
104 с., 35 рис., 4 додатки, 34 джерела.

ЕКСПЕРТНА СИСТЕМА, КОМП'ЮТЕРНА ГРА, БАЗА ЗНАНЬ, 3D
МОДЕЛЬ, ШТУЧНИЙ ІНТЕЛЕКТ.

Об'єкт дослідження – процес розробки комп'ютерних ігор.

Предмет дослідження – інтелектуальні технології розробки
комп'ютерних ігор.

Мета роботи – дослідження та розробка комп'ютерної гри на основі
інтелектуальних технологій.

Матеріали, методи та технічні засоби: структурне та об'єктно-
орієнтоване програмування, середовище розробки Unity 3D, персональний
комп'ютер з монітором, клавіатурою та маніпулятором типу «миша», що
функціонує під управлінням операційної системи.

Результати. Створено програмні засоби у вигляді комп'ютерної гри на
основі інтелектуальних технологій. Розроблена програмна система може
успішно експлуатуватися на персональних комп'ютерах під управлінням
операційних систем та використовуватися для розважання користувача шляхом
проходження місій комп'ютерної гри.

Висновки. Розроблено програмне забезпечення у вигляді комп'ютерної
гри на основі інтелектуальних технологій. Проаналізовано предметну область,
розроблено технічне завдання для реалізації програми, обрано середовище
розробки, розроблено структурну схему програми, виконано конструювання
програмного забезпечення у вигляді комп'ютерної гри на основі
інтелектуальних технологій, здійснено тестування розробленого програмного
забезпечення.

Галузь використання – розважальні комп'ютерні ігри.

ABSTRACT

Explanatory note to the final qualifying work of the magister: 104 pages, 35 figures, 4 appendixes, 34 sources.

EXPERT SYSTEM, COMPUTER GAME, KNOWLEDGE BASE, 3D MODEL, ARTIFICIAL INTELLIGENCE.

Object of study is a process of computer games development.

Subject of study are intelligent technologies of computer game development.

The purpose of the work is to study and develop a computer game based on intelligent technologies

Materials, methods, and tools: structured and object-oriented programming, programming language Unity 3D, computer with a monitor, keyboard and mouse manipulator operating under the control of the operating system.

Results. The software in the form of a computer game based on intelligent technologies has been created. The developed software system can successfully operate on personal computers running operating system and can be used for the user entertainment by completing computer game missions.

Conclusions. Software in the form of a computer game based on intelligent technologies has been developed. The subject area is analyzed, the technical task for program realization is developed, the development environment is chosen, the structural scheme of the program is developed, the software in the form of a computer game based on intelligent technologies is executed, the developed software is tested.

The field of use are computer games for entertainment.

ЗМІСТ

Перелік скорочень та умовних позначок.....	8
Вступ	9
1 Аналіз предметної області	10
1.1 Аналіз процесу розробки комп'ютерних ігор.....	10
1.2 Аналіз засобів 3D моделювання для реалізації комп'ютерних ігор	11
1.3 Аналіз методів та засобів штучного інтелекту для розробки комп'ютерних ігор.....	13
1.4 Висновки за розділом 1	15
2 Вибір та обґрунтування структури системи, яка проєктується.....	16
2.1 Дослідження та аналіз вимог до програмного продукту	16
2.2 Аналіз програмних засобів для створення комп'ютерних ігор.....	17
2.3 Розробка моделі комп'ютерної гри на основі уніфікованої мови моделювання UML	21
2.4 Проєктування експертної системи комп'ютерної гри, заснованої на знаннях.....	23
2.5 Моделювання 3D об'єктів та ігрового світу	24
2.6 Розробка інтерфейсу користувача.....	27
2.7 Висновки за розділом 2	28
3 Основні рішення щодо реалізації компонентів системи.....	30
3.1 Основні рішення щодо подання даних системи	30
3.2 Основні розроблені алгоритми	30
3.3 Основні рішення щодо модульного подання системи	35
3.4 Тестування програмного забезпечення	36
3.5 Висновки за розділом 3	36
4 Керівництво програміста.....	38
4.1 Призначення й умови застосування програми.....	38
4.2 Характеристики програми	38
4.3 Звернення до програми	39

	7
4.4 Вхідні і вихідні дані.....	39
4.5 Повідомлення	40
5 Керівництво оператора.....	41
5.1 Призначення програми.....	41
5.2 Умови виконання програми.....	41
5.3 Виконання програми	41
5.4 Приклад роботи з програмою	42
5.5 Повідомлення оператору.....	45
6 Економіко-організаційна частина	46
6.1 Планування розробки програмного продукту	46
6.2 Визначення витрат на розробку програми	47
6.3 Розрахунок економічної ефективності розробки програмного продукту .	54
7 Охорона праці та безпека у надзвичайних ситуаціях.....	56
7.1 Аналіз потенційних небезпек	56
7.2 Заходи щодо забезпечення безпеки	57
7.3 Заходи щодо виробничої санітарії та гігієни праці	64
7.4 Заходи з пожежної безпеки.....	66
7.5 Заходи з цивільного захисту	67
Висновки.....	71
Перелік джерел посилання.....	73
Додаток А Технічне завдання.....	77
Додаток Б Опис програми.....	81
Додаток В Текст програми.....	84
Додаток Г Слайди презентації	96

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

GUI (ГІК) – графічний інтерфейс користувача;

БЗ – база знань;

ОС – операційна система;

ПЗ – програмне забезпечення;

ПО – предметна область;

ПК – персональний комп'ютер;

ПП – програмний продукт;

ШІ – штучний інтелект.

ВСТУП

У наш час для автоматизації процесу розважання користувачів широко використовуються комп'ютерні ігри [1], [2]. Сьогодні розробка комп'ютерних ігор являє собою потужну індустрію, а бюджети розробки ігрових проєктів доходять до десятків мільйонів доларів, а загальний об'єм ринку комп'ютерних ігор (настільних, мобільних, консольних) оцінюється десятками мільярдів доларів [1], [2].

Метою роботи є дослідження та розробка комп'ютерної гри на основі інтелектуальних технологій. Оглянувши можливості та засоби створення комп'ютерної гри засобами штучного інтелекту, було вирішено розробити шутер від першого обличчя використовуючи модифіковану експертну систему засновану на знаннях.

Для досягнення поставленої мети необхідно було розв'язати такі завдання:

- аналіз підходів до розробки комп'ютерних ігор, зокрема методів штучного інтелекту;
- розробка технічного завдання для реалізації програми;
- вибір середовища розробки програмного забезпечення;
- розробка моделі комп'ютерної гри;
- розробка інтерфейсу користувача;
- конструювання програмного забезпечення у вигляді комп'ютерної гри на основі інтелектуальних технологій;
- тестування розробленого програмного забезпечення.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Розробка комп'ютерної гри засобами штучного інтелекту потребує багато знань з різних предметних областей. Щоб користувач програмного продукту поринув у великий захоплюючий світ гри з гарною графікою треба багато знань з області 3D моделювання. Для вражаючого сюжету протистояння головного героя з ворогами треба мати знання з такої предметної області як штучний інтелект, а цього мало, щоб повноцінно реалізувати ПП. Треба мати навички у програмуванні, щоб у самій середі розробки зв'язати усі компоненти [1],[2].

1.1 Аналіз процесу розробки комп'ютерних ігор

Розробка комп'ютерної гри представляє собою процес створення програми у вигляді відеогри. Цим процесом, як правило, займається група розробників, або один розробник. Тривалість та вартість розробки комп'ютерної гри залежить від різних факторів, зокрема від складності проєкту гри [1],[2].

Комерційна розробка комп'ютерних ігор почалася у минулому сторіччі, разом з появою гральних приставок та аркадних автоматів. Через низьку потужність комп'ютерного обладнання, що існувало у ті роки, ігри були досить простими. Тому час, що витрачався на їх розробку, був досить не значний, а саму гру міг писати лише один програміст. Проте із збільшенням обчислювальних можливостей зростали вимоги гравців, внаслідок чого підвищувалася складність проєктів комп'ютерних ігор, що призвело до того, що над такими проєктами почали працювати цілі команд професіоналів [1],[2].

Розробка гри проходить у декілька етапів. На початку створюється загальний проєкт майбутньої гри, пишеться дизайн-документ з описом всесвіту, ігрового процесу і сюжету, створюються графічні концепти рівнів і персонажів, які визначатимуть стилістику гри і які слугуватимуть орієнтиром для

художників та моделювальників. Дизайн–документ та малюнки також використовуються для привернення уваги видавців. Якщо відсутні проблеми з фінансуванням, починається повноцінна робота над грою, яка включає розробку чи налаштування під потреби гри ігрового рушія, створення графічного, тривимірного, та аудіо наповнення гри, впровадження та тестування ігрових механік (ігровий дизайн). До своєї остаточної версії гра йде через альфа та бета версії, які ретельно перевіряють і тестують, іноді проводячи відкриті тестування за участю гравців (відкритті альфа/бета-тести). Через якийсь час після останніх перевірок і тестувань, гра виходить на високому рівні [1],[2].

1.2 Аналіз засобів 3D моделювання для реалізації комп'ютерних ігор

Людина в будь якій діяльності постійно користується моделями. У наш час стало майже незамінним комп'ютерне моделювання для різних галузях діяльності людини. Існує дуже багато областей, де застосовується тривимірне моделювання і анімація. Наприклад, під час випробування програми 3D Studio MAX користувачі виконали колосальну роботу, застосовуючи цю програму різних галузях: від створення статичної реклами й динамічних заставок телеканалам до моделювання катастроф у тривимірній анімації. Донедавна працювати над створенням спецефектів в кінематографії доводилось у спеціальних павільйонах з допомогою фізичних моделей, методів прозорої фотографії і дорогих оптичних принтерів. Нині цю проблему вчені вирішили за допомогою сучасних програм. Вже не потрібно витрачати тисячі людино-годин на побудову моделей. Досить посадити одну людину за звичайний персональний комп'ютер, щоб зробити спецефекти, які створюють повне відчуття реальності. Так само і при розробці комп'ютерних ігор [1],[2].

У комп'ютерній графіці 3D моделювання – це процес розробки математичного представлення будь-якої тривимірної поверхні об'єкта за допомогою спеціалізованого ПЗ. Продукт моделювання є 3D модель. Вона

може бути представлена у вигляді програмного коду або відображена у вьюпорті чи вювері, як 3D модель, а також за допомогою двовимірного зображення, що створюється за допомогою процесу рендерингу. 3D моделі можуть створюватись вручну або автоматично. Виготовлення моделей вручну є подібним до створення скульптури в пластичному мистецтві [1],[2].

3D моделі представляють 3D об'єкт використовуючи набір точок в 3D просторі, поєднаних між собою різноманітними геометричними об'єктами, як от трикутниками, лініями тощо [1],[2].

У наш час було розроблено багато комп'ютерних програм для 3D моделювання. Компанія Autodesk розробила програму 3D Studio, найбільшого поширення яка одержав у версії 4.0 для DOS. Ця система використовувалася для розробки реалістичних ігор, а також моделей і спецефектів у фільмах. Потім для тривимірного моделювання і анімації було розроблено 3D Studio MAX для Windows. Ця програма використовувалася до розробки ігор, фільмів, комп'ютерних програм з застосуванням спецефектів та графіки. Програма 3D Studio MAX не призначена для розробки інженерних проєктів. Для таких завдань компанія Autodesk розробила AutoCAD. Ця програма здатна виконувати різні операції, пов'язані з інженерними та іншими системами. Компанія з виробництва CAD систем Bentley Systems розробила програму MicroStation, основне призначення якої полягає у виконанні проєктних робіт в галузях транспорту, електроніці, машинобудуванні, геодезії і картографії, управлінні та архітектурі. Для розв'язання графічних завдань шоу-бізнесу, а також для створення рекламних роликів компанією Glassworks розроблено програму Soft Image, яка пристосована для створення відео кліпів та 3D моделей. Компанія Fractal Design створила програму Fractal Design Detailer. Ця програма являє собою програму оброблення поверхонь тримірних об'єктів. Для моделювання поверхонь компанія MetaTools створила генератор ландшафтів Bryce (версії 2, а 3D). Ця програма дозволяє створювати реалістичні пейзажі, керувати погодними умовами, часом доби, і т.ін. Основною перевагою цієї програми є зручний інтерфейс. Також широкого використання набув

програмний продукт Animatek's World Builder 2.0 – програмний пакет, який призначений для створення реалістичних пейзажів, з широким набором професійних функцій [1],[2].

1.3 Аналіз методів та засобів штучного інтелекту для розробки комп'ютерних ігор

У наш час для розробки комп'ютерних ігор широко використовуються методи та засоби штучного інтелекту [3]–[5].

Штучний інтелект – це галузь комп'ютерних наук, що займається автоматизацією розумного поведіння агентів, які одержують у результаті актів сприйняття інформацію про навколишнє середовище і виконують дії, що реалізують функцію від результатів сприйняття і попередніх дій. Метою штучного інтелекту є розробка комп'ютерних систем, що мають можливості, які традиційно пов'язуються з людським розумом: розуміння мови, навчання, здатність міркувати, вирішувати проблеми, планувати і т. д. [2]–[17].

Основні напрями досліджень в галузі штучного інтелекту виділяють такі [3]–[5]:

- подання знань і робота з ними – створення спеціалізованих моделей і мов для подання знань в ЕОМ, а також програмних і апаратних засобів для їхнього перетворення(поповнення, логічної обробки і т. д.) [3]–[5];

- планування доцільного поведіння – дослідження зі створення методів формування цілей і вирішення задач планування дій агента, що функціонує в складному зовнішньому середовищі [3]–[5];

- спілкування людини з ЕОМ – задачі створення мовних засобів, що дозволяють ефективно взаємодіяти з ЕОМ не програмуючому користувачу. Ведуться дослідження в області синтаксису і семантики природних мов, способів збереження знань про мову в пам'яті машини і побудови спеціальних процесорів, що здійснюють переклад текстової інформації у внутрішнє машинне подання [3]–[5];

– розпізнавання образів і навчання – дослідження зі сприйняття зорової, слухової і інших видів інформації, методів її обробки, формування відповідних реакцій на впливи зовнішнього середовища і способів адаптації штучних систем до середовища шляхом навчання [3]–[5].

Інтелектуальною системою називають кібернетичну систему, призначену для вирішення інтелектуальних задач. Властивість інтелектуальності, що істотно відрізняє інтелектуальні системи від інших кібернетичних систем, характеризується набором різних ознак [3]–[5].

Штучний інтелект також використовують в комп'ютерних іграх для створення «розуму» 3D моделям. Для надання «розуму» «ботам» використовують такі моделі штучного інтелекту [3]–[17]:

– експертна система – це програмний засіб, що використовує експертні знання у певній предметній області з метою ефективного вирішення задач у предметній області, яка цікавить користувача, на рівні середнього професіонала (експерта) [3]–[5];

– дерева розв'язків – один з методів автоматичного аналізу даних, що задає спосіб подання правил виду «Якщо – то» в ієрархічній послідовній структурі, де кожному об'єкту відповідає єдиний вузол, що дає рішення. Основними поняттями теорії дерев рішень, є: об'єкт – приклад, шаблон, спостереження; атрибут – ознака, незалежна змінна, властивість; мітка класу – залежна(цільова) змінна, ознака, що визначає клас об'єкта; вузол – внутрішній вузол дерева, вузол перевірки; лист – кінцевий вузол дерева, вузол рішення; перевірка(test) – умова у вузлі [3]–[17];

– штучні нейронні мережі – математичні моделі, а також їх програмні або апаратні реалізації, побудовані за принципами подання й обробки інформації у біологічних нейронних мережах [3]–[17].

1.4 Висновки за розділом 1

Проаналізовано процес розробки комп'ютерних ігор. Визначено, що в наш час для розробки комп'ютерних ігор широко використовуються методи та засоби штучного інтелекту. Досліджено інтелектуальні технології розробки комп'ютерних ігор.

Проаналізувавши головні аспекти й поняття у сферах штучного інтелекту, 3D моделювання та розробки комп'ютерних ігор було підготовлено підґрунтя для розробки програмного продукту.

2 ВИБІР ТА ОБҐРУНТУВАННЯ СТРУКТУРИ СИСТЕМИ, ЯКА ПРОЄКТУЄТЬСЯ

2.1 Дослідження та аналіз вимог до програмного продукту

Метою розробки є створення комп'ютерної гри засобами штучного інтелекту типу шутер від першого обличчя, використовуючи модифіковану експертну систему, засновану на знаннях.

Діловий контекст розробки полягає в тому, що по завершенню реалізації проєкту власник матиме готовий ПП, з якого зможе отримати матеріальну вигоду, продавши права на нього. Розробку можливо обновлювати й за кожную нову версію (доповнення) гри можна отримати матеріальну вигоду.

Функціональними вимогами до програмного продукту (ПП) є такі:

- можливість вводу нікнейму у грі;
- можливість збереження ігрових даних;
- можливість пересуватися по ігровому світі;
- можливість відкрити вогонь по ворогам;
- можливість вести прицільний вогонь;
- можливість пересуватися по ігровому світі різними способами;
- можливість виконання місії;
- взаємодія головного персонажу з допоміжним за допомогою ІІІ;
- можливість вибивати ворогів у грі;
- можливість відхилення від головної місії.

Розроблений ПП має зберігатися на DVD диску розміром 8ГБ. Данні повинні зчитуватися на платформі Windows NT.

Інтерфейс має бути інтуїтивно зрозумілий користувачу. Повинен мати розміщення кнопок навігаційного меню по центру екрану. Колір меню не повинен дратувати око користувача. На головному екрані повинні показуватися данні о здоров'ї гравця, наявності патронів у зброї.

ПП повинен забезпечити хід гри по задуманому сюжету без втрачання даних при збереженні та ході самої гри.

Для запуску розроблюваного ПП треба:

- процесор 2 ядра, частотою 3,2 ГГц;
- місце на жорсткому диску – 8 Гб;
- відеокарта з оперативною пам'яттю 1 Гб;
- оперативна пам'ять – 4 Гб;
- Direct X 9.0;
- звукова карта сумісна з Direct.

Розповсюдження ПП повинно проводитись згідно з законам України.

2.2 Аналіз програмних засобів для створення комп'ютерних ігор

Для розробки комп'ютерних ігор широко використовується середовище розробки Unity 3D, яке дозволяє двох-і тривимірні застосунки та ігри, які виконуються у операційних системах Windows та OS X. ShiVa3D є 3D двигун гри з графічним редактором, призначений для створення додатків і відеоігор для настільних ПК, в Інтернеті, для ігрових приставок та мобільних пристроїв. Ігри, зроблені з Шивой можуть бути експортовані в більш ніж 20 цільових платформ.

Численні програми були створені за допомогою Шиви, в тому числі принц Персії 2 рімейк для мобільних телефонів і Вавилонська Висхідна опублікована Ubisoft.

Актуальна збірка – ShiVa3D 2.0, наступна версія редактора в даний час активно розробляється. Користувачам Шиви з ліцензіями новіше 1 січня 2012 пропонується завантажити бета-версії нової версії.

Поточні можливості основного двигуна включають в себе:

- двигун працює на усіх різних цільових платформах, включаючи мобільні телефони, настільні ПК, веб – браузері та ігрові приставки. Є підтримка 32bit і 64bit операційних систем;
- присутня Lua API згідним C ++ підтримкою компіляції;

– наявний промисловий стандартний код і архітектури, який може бути використаний для розширення бібліотеки двигуна, як NVIDIA PhysX, F-Mod, і ARToolKit;

– підтримка OpenGL, DirectX 9 – 11 і режимів OpenGL ES;

– повний контроль Lightmap (UV2, імпорт, експорт, вбудований картограф);

– ефекти пост обробки, такі як глибина різкості та ефект спотворення камери;

– фізика складних тіл;

– система 2D HUD для відтворення інформації на дисплеї декількох видових камер на сцені;

– стереоскопічні режими 3D – рендерингу;

– вбудований XML API для обміну файлами;

– мережа API;

– стерео аудіо 5.1 – каналний звук для 3D звукових ефектів.

Платформи, на які можна скомпілювати розроблений програмний продукт:

– Windows XP / Vista / 7/8 / 8.1 в стандартному режимі з DirectX 9 або OpenGL 2.0 (x86);

– Windows 8 / 8.1 у DirectX 11 режимі з підтримкою Windows Store, як застосунок WinRT (x86, x86 64, ARM);

– Mac OS X 10.6+ в x86 и x86 64 або як Universal Binary;

– Linux;

– устройства IOS 6+, включаючи Ipad, iphone та Ipod Touch;

– Android 2.3+;

– Windows Phone 7.5 и 8;

– BlackBerry Tablet OS и BlackBerry 10;

– Sony Playstation 3.

Ігри ShiVa Engine, розроблені за допомогою ShiVa редактору, WYSIWYG RAD, щоб дозволити розробникам розроблять 3D – ігр та додатки на не затрачуючи дуже багато часу.

ShiVa редактор 2.x працює на всіх ОС Windows XP / Vista / 7/8 / 8.1 x86 / x86 64, Mac OS X x86 64 і Linux x86 64. Колись окремий інструментарій розробника вбудований в сам редактор.

У той час, коли редактор ShiVa може експортувати готові для запуску ігри, продвинуті користувачі можуть використовувати його для експорту в Eclipse та Visual Studio, щоб зробити певний модинг IDE.

CryEngine 3 SDK – набір програмних інструментів для розробки комп'ютерних ігор на ігровому двигуні CryEngine3. CryEngine3 SDK, як і CryEngine3, розроблений і підтримується німецькою компанією Crytek. CryEngine3 SDK є безкоштовним (freeware) для скачування, установки і використання засобом розробки, за допомогою якого будь який бажаючий може розробляти комп'ютерні ігри та безкоштовно поширювати їх. При комерційному поширенні (продажу) ігор, розроблених на CryEngine3 SDK, необхідно робити ліцензійні відрахування в Crytek.

CryEngine 3 SDK базується на ігровому двигуні CryEngine3 і містить усі компоненти останнього із засобами розробки, включаючи редактор рівнів Sandbox третьої версії.

CryEngine 3 SDK містить безліч додаткових інструментів, включаючи вихідний код на C ++, експортери контенту, приклади скриптів, механізм Flowgraph та інше. Разом з тим, графічна компонента CryEngine 3, яка увійшла в першу версію CryEngine 3 SDK, підтримує лише DirectX 9; додавання підтримки DirectX 11 заявлено розробниками в наступних випусках. У складі CryEngine 3 SDK присутні 32-бітна і 64-бітна версії інструментів.

У першому випуску CryEngine 3 SDK доступ до редагування рівнів має лише їх розробник; він не може відкрити доступ на редагування іншим користувачам. У наступних версіях заявлено впровадження нової системи

управління доступом до рівнів, яка дозволить створювати колективні проєкти, регулювати рівень доступу учасників цих проєктів до рівнів і т.д.

Ліцензія CryEngine 3 SDK дозволяє безкоштовно використовувати його для створення та розповсюдження некомерційних ігор. У випадку, якщо розробник бажає використовувати CryEngine 3 SDK в комерційних цілях, він повинен відраховувати компанії Crytek 20% від продажів своїх ігор, розроблених з використанням даного SDK.

Порівняльну характеристику відомих ігрових двигунів наведено у таблиці 2.1.

Таблиця 2.1 – Порівняльна характеристика ігрових двигунів

Критерій	Uniti 3D	ShiVa3D	CryEngine 3 SDK
Підтримка DirectX 11	+	+	+
Підтримка PhysX	+	+	+
Підтримка Open GL	+	–	+
Підтримка компілювання проєкту на інші платформи	+	+	–
Наявність набору стандартних 3D моделей	+	–	+
Підтримка C#	+	–	–
Підтримка C++	–	+	+
Підтримка java, javascript	+	–	+
Підтримка Фізики складних тіл	+	+	+
Наявність вбудованих контролерів	+	+	+
Аварійне збереження проєкту	+	–	–
Вільність використання середі	Комерційна програма	У вільному доступі	20% від реалізації гри

Порівнявши відомі середовища розробки комп'ютерних ігор, оцінивши їх переваги та недоліки, було прийнято рішення про використання для розробки комп'ютерної гри середовища Uniti 3D.

2.3 Розробка моделі комп'ютерної гри на основі уніфікованої мови моделювання UML

Розроблена модель комп'ютерної гри представляє собою набір діаграм у нотації UML, що дозволяють розробнику створювати комп'ютерну гру відповідно до вимог замовника.

На рисунку 2.1 приведена діаграма класів моделі комп'ютерної гри.

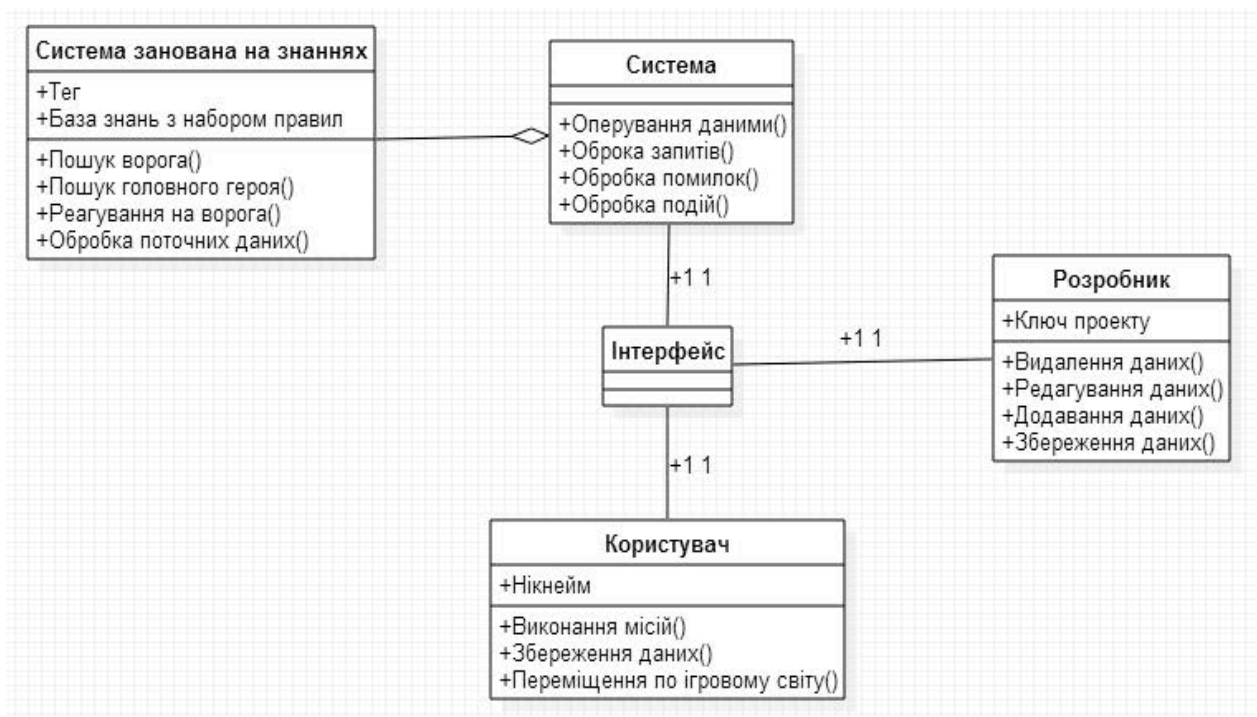


Рисунок 2.1 – Діаграма класів моделі комп'ютерної гри

Для побудови архітектурного каркасу програми використано схему проектування Model-view-controller (MVC).

Модель-вид-контролер – архітектурний шаблон, що застосовується під час проектування та розробки програмного забезпечення.

Шаблон Модель-вид-контролер розділяє систему на три частини: вигляд (подання) даних, модель даних, та керування. Такий шаблон використовується для віокремлення даних програми (модель) від інтерфейсу користувача (вигляду, подання) таким чином, щоб зміни інтерфейсу користувача не

впливали на роботу з даними (або впливали на цю роботу мінімально), а зміни у модлі даних мали можливість здійснюватися без змін користувальницького інтерфейсу. Метою цього шаблону є гнучкий дизайн (проектування) ПЗ, що повинен полегшувати майбутні модифікації або розширення програм.

Модель подання архітектурного каркасу комп'ютерної гри у вигляді схеми проектування MVC наведено на рис.2.2.

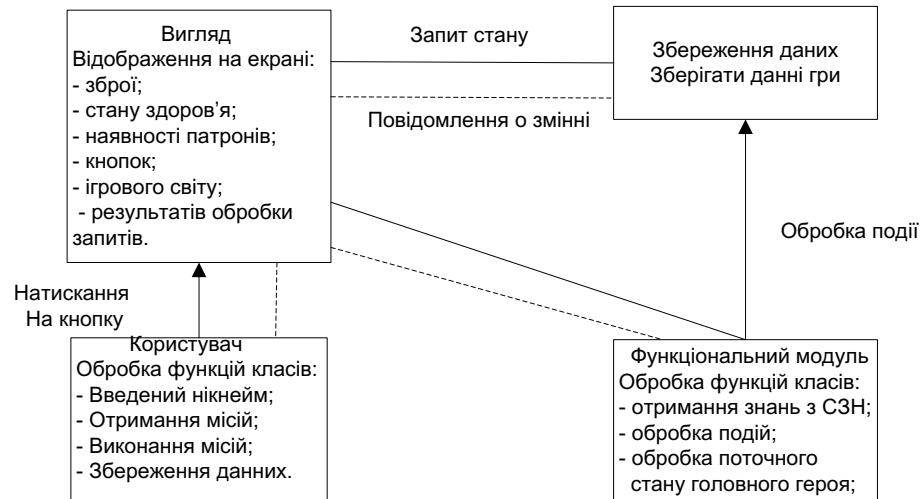


Рисунок 2.2 – Модель подання архітектурного каркасу комп'ютерної гри у вигляді схеми проектування MVC

Діаграма прецедентів являє собою графічне подання певних прецедентів та акторів, а також їх взаємодій. У різних системах, як правило, виділяють головну діаграму прецедентів. На такій діаграмі наводиться зовнішня границя системи, а також базові зовнішні функції системи (тобто відображається зовнішня поведінка).

Модель взаємодії користувача з системою наведено у вигляді діаграми прецедентів (рисунок 2.3).



Рисунок 2.3 – Модель взаємодії користувача з системою у вигляді діаграми прецедентів

Як видно з рисунку, користувач може взаємодіяти з системою, як правило, шляхом створення гри, зберігання гри, отримання місії та вільного пересування по ігровому світу.

Таким чином, розроблено модель комп'ютерної гри представляє собою набір діаграм у нотації UML, що дозволяють розробнику створювати комп'ютерну гру відповідно до вимог замовника.

Запропоновано діаграму класів моделі комп'ютерної гри, подання архітектурного каркасу комп'ютерної гри у вигляді схеми проектування MVC.

2.4 Проектування експертної системи комп'ютерної гри, заснованої на знаннях

Як відзначено вище, експертна система (ЕС) представляє собою комп'ютерну програму. Ця програма дозволяє на основі бази знань, яка створена експертами з предметної області, шляхом логічного виведення розв'язати поставлене завдання. Базовими компонентами експертної системи є: база фактів, база знань; машина логічного виведення; редактор бази знань; інтерфейс (користувача й інженера знань), підсистема пояснення [3]. Експерт представляє собою висококваліфікованого фахівця предметної області.

Когнітолог (або інженер по знанням) – є фахівець з розробки експертних систем. Такий фахівець виконує видобування знань з експерта, а також здійснює їх формалізацію у відповідності до моделі предметної області. Користувач системи є, як правило, фахівець предметної області [3].

Проектована ЕС інтегрована у середовище комп'ютерної гри, має можливість виконувати свої функції на різних платформах та має локальну архітектуру.

Експертна система знаходиться на стадії існування «Дослідний прототип», тобто може вирішувати представницький клас задач. Має невелику кількість виконуваних тверджень. Масштаб ЕС – малий, тобто вирішує нескладні задачі.

2.5 Моделювання 3D об'єктів та ігрового світу

У даному проєкті використовуються моделі будинків, зброї, водопостачальних веж та інших об'єктів. Усі ці моделі створюють певну атмосферу віртуального світу. Для створення 3D моделей 3D Studio MAX. На рисунках 2.4 – 2.5 приведено приклад розробки 3D моделі пістолету.

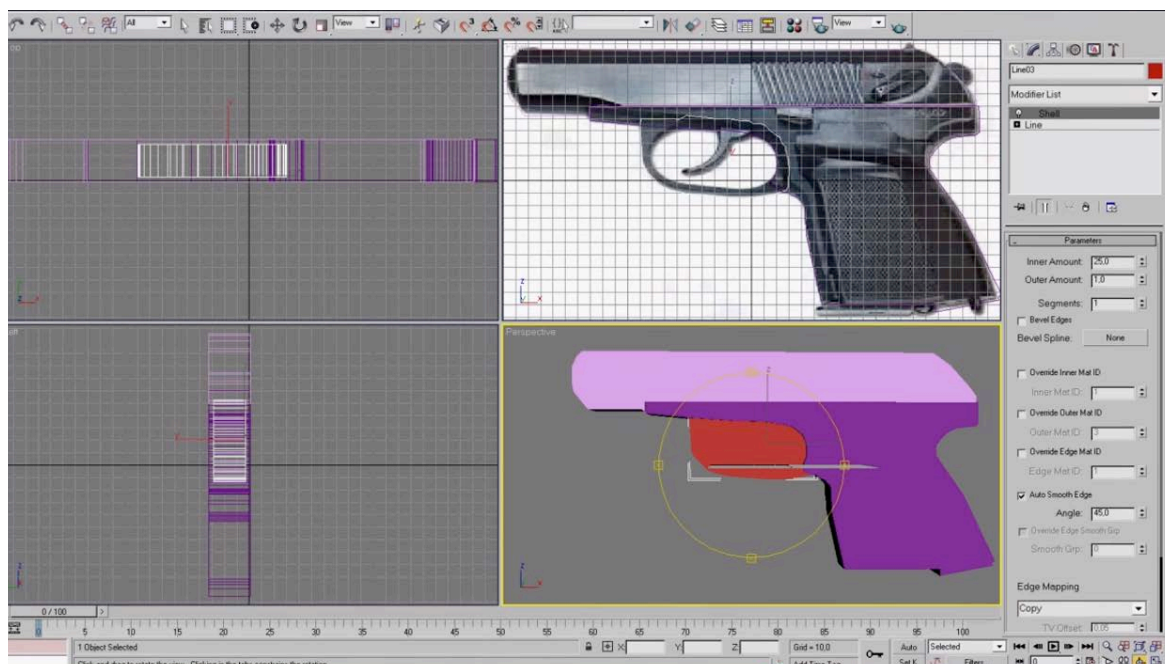


Рисунок 2.4 – Створення 3D моделі пістолету.

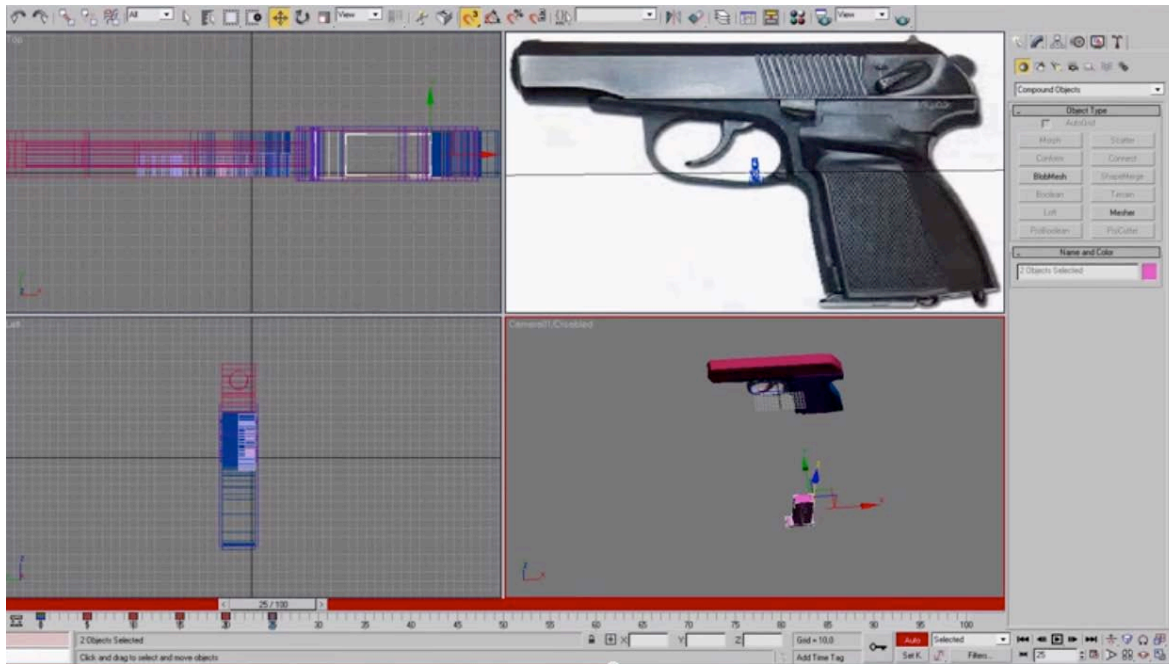


Рисунок 2.5 – Створення анімації 3D моделі пістолету.

За допомогою моделей будинків була створена невеличка локація зі сільською місцевістю, локація з руїнами та лісом, зброєю які приведені на рисунках 2.6 – 2.11.



Рисунок 2.6 – Приклад сільського поселення



Рисунок 2.7 – Приклад сільського поселення



Рисунок 2.8 – Моделювання локації руїн



Рисунок 2.9 – Моделювання локації лісу



Рисунок 2.10 – Установка модельованої зброї на ігровий контролер



Рисунок 2.11 – Налаштування змодельованої зброї

2.6 Розробка інтерфейсу користувача

Інтерфейс (англ. Interface – сполучення, поверхня розділу, перегородка) – сукупність методів, способів, можливостей взаємодії різних систем (будь-яких, а не обов'язково є обчислювальними або інформаційними), програм або пристроїв для обміну інформацією між ними, з певними їх характеристиками, характеристиками з'єднання, сигналів обміну і т. п.

Графічний користувацький інтерфейс, ГІК (англ. Graphical user interface, GUI) – різновид користувацького інтерфейсу, де елементи (меню, списки, значки, кнопки тощо.), подані для користувача за допомогою екрану у вигляді певних графічних зображень.

За допомогою графічного інтерфейсу, на відміну від інтерфейсу у вигляді командного рядка, в GUI користувач має можливість довільного доступу (шляхом пристроїв введення – миші, джойстика, клавіатури, тощо.) до видимих екранних об'єктів (інтерфейсних елементів) та здійснює маніпулювання ними. Часто елементи інтерфейсу в GUI створено за допомогою метафор і відображають їхні властивості і призначення. Це спрощує розуміння та оволодіння програмами непідготовленими користувачами.

Графічний інтерфейс користувача є частиною користувацького інтерфейсу і визначає взаємодію з користувачем на рівні візуалізованої інформації.

Розроблений інтерфейс легкий у орієнтуванні та відображує головну інформацію о персонажі. При розробці інтерфейсу використовувались:

- Rect – прямокутний об'єкт;
- Main Camera – головна камера (відображує ігровий світ);
- GUI.Button – інтерфейсна кнопка;
- GUI.TextField - інтерфейсне текстове поле;
- GUI.Label - інтефейсний елемент відображування тексту;
- animationGO.animation.Play – елемент програшу анімації.

Приклад інтерфейсу продемонстровано у керівництві користувачу.

2.7 Висновки за розділом 2

Досліджено та проаналізовано вимоги до програмного продукту. Проаналізовано програмні засоби для створення комп'ютерних ігор. Порівнявши відомі середовища розробки комп'ютерних ігор, оцінивши їх переваги та недоліки, було прийнято рішення про використання для розробки комп'ютерної гри середовища Unity 3D, оскільки Unity 3D є зручною у використанні платформою для розробки багатоплатформених 2D та 3D ігор та інтерактивного контенту.

Розроблено модель комп'ютерної гри представляє собою набір діаграм у нотації UML, що дозволяють розробнику створювати комп'ютерну гру відповідно до вимог замовника.

Запропоновано діаграму класів моделі комп'ютерної гри, подання архітектурного каркасу комп'ютерної гри у вигляді схеми проектування MVC.

Виконано моделювання 3D об'єктів та ігрового світу, розроблено інтерфейс користувача.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Основні рішення щодо подання даних системи

Основними даними системи є файли формату *.js, у які заносяться знання ЕС та головний код програми. Дані у файлах мають особливу структуру діалекту java та javascript.

Основні дані, що знаходяться у файлах, у представленні системи мають тип String, Boolean, Rect, Object.

Основні поля і методи, що використовуються у системі, приведені у таблиці 3.1.

Таблиця 3.1 – Поля і методи, що використовуються у системі

Елемент	Опис
var _gameMode :boolean	Вказівник на змінну типу boolean
var attackAnimation : AnimationClip;	Вказівник на змінну типу AnimationClip;
var seeDistance : float	змінна типу float
GameObject.FindWithTag("Player")	Метод який за допомогою якого виконується пошук гравця
private var nav: NavMeshAgent	Вказівник на змінну NavMeshAgent
Update ()	Конструктор
animation.CrossFade("attack")	Метод, за допомогою якого виконується анімація атаки
yield WaitForSeconds(2.0)	Метод, який дозволяє затримувати виконання певної операції на деякий час

3.2 Основні розроблені алгоритми

Основними алгоритмами роботи розробленого ПП є алгоритми, що виконуються ЕС, працюють з анімацією та елементами візуальної форми програми.

Алгоритм пошуку гравця дозволяє знайти вказаний об'єкт у ігровому світі. Алгоритм проводить пошук гравця на заданому радіусі від певного об'єкта, якщо знаходить, то на певній відстані відкриває вогонь по вказаному об'єкту з показом анімації.

```
#pragma strict
var seeDistance : float = 30.5f;
var attackDistance : float = 5.0f;
var idleAnimation : AnimationClip;
var walkAnimation : AnimationClip;
var attackAnimation : AnimationClip;
private var nav: NavMeshAgent;
private var target : Transform;
var hp : float = 100f;
var dead : GameObject;
var damage : float = 3.5f;
var delay : float = 1.0f;
private var time : float;
function Start () {
    animation.AddClip(idleAnimation, "idle");
    animation.AddClip(walkAnimation, "walk");
    animation.AddClip(attackAnimation, "attack");
    target = GameObject.FindWithTag("Player").transform;
    nav = GetComponent(NavMeshAgent);
}
function ApplyDamage(d : float){
    hp -=d;
    if(hp<=0){
        Instantiate(dead, transform.position, transform.rotation);
        Destroy(gameObject);
    }
}
```

```

function Update () {
    if(Vector3.Distance(transform.position, target.transform.position) <
seeDistance)
    {
        if(Vector3.Distance(transform.position, target.transform.position) >
attackDistance)
        {
            animation.CrossFade("walk");
            nav.enabled = true;
            nav.SetDestination(target.position);
        }else{
            if (Time.time > time){
                animation.CrossFade("attack");
                nav.enabled = false;
                target.SendMessage("AplyDamage", damage,
SendMessageOptions.DontRequireReceiver);
                time = Time.time + delay;
            }
        }else{
            animation.CrossFade("idle");
            nav.enabled = false;
        }
    }
}

```

Алгоритм виконання запуску меню під час гри наведено у вигляді схеми на рис. 3.1.

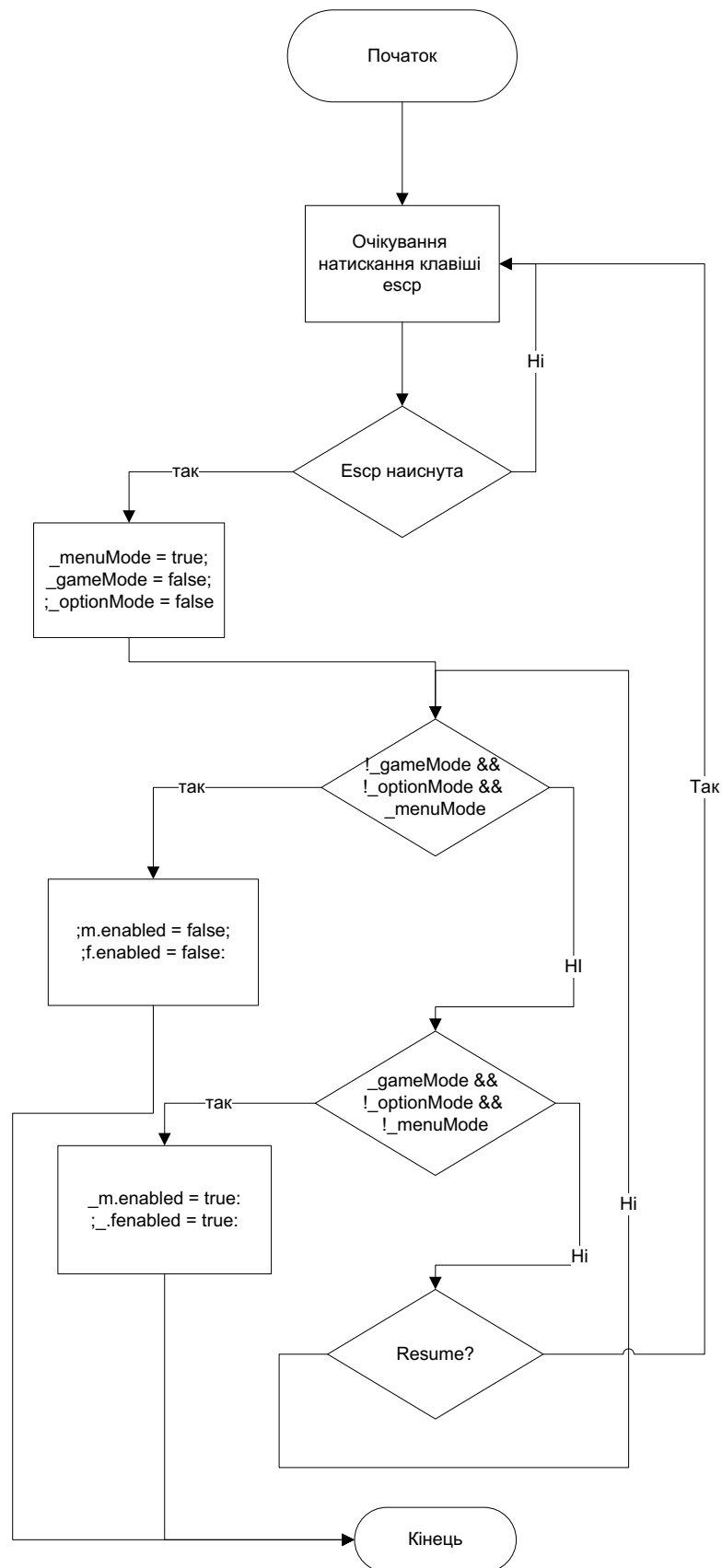


Рисунок 3.1 – Алгоритм запуску меню під час гри

Алгоритм роботи зброї налаштований наступним чином: якщо натиснута клавіша «Fire1», то спрацьовує анімація і відбувається вогонь. Якщо натиснута клавіша «1» - достається нова зброя. Якщо натиснута клавіша «r», то відіграється анімація перезарядження зброї.

```
function Update (){
    if(Input.GetButtonDown ("Fire1") && reloading == false && drawWeapon
== false){
        Fire();
    }
    if (Input.GetKeyDown ("r") && reloading == false && drawWeapon ==
false){
        Reloading();
    }
    if (Input.GetKeyDown ("1") && reloading == false){
        DrawWeapon();
    }
}
function Fire(){
    animationGO.animation.CrossFadeQueued(fireLeftAnim, 0.08,
QueueMode.PlayNow);
}
function DrawWeapon() {
    if(drawWeapon)
        return;
    animationGO.animation.Play(drawAnim);
    drawWeapon = true;
    yield WaitForSeconds(0.6);
    drawWeapon = false;
}
```

```
function Reloading(){
    if(reloading) return;
    animationGO.animation.Play(reloadAnim);
    reloading = true;
    yield WaitForSeconds(2.0);
    reloading = false;
}
```

3.3 Основні рішення щодо модульного подання системи

У проєкті використовуються два глобальні модулі, у яких містяться вже змодельовані об'єкти в ігровий світ і програмні коди (рис. 3.2) та об'єкти з макетами, матеріалами, шаблонами, які ще не перенесені у віртуальний світ (рис. 3.3).



Рисунок 3.2 – Модульне подання системи: змодельовані об'єкти в ігровий світ і програмні коди

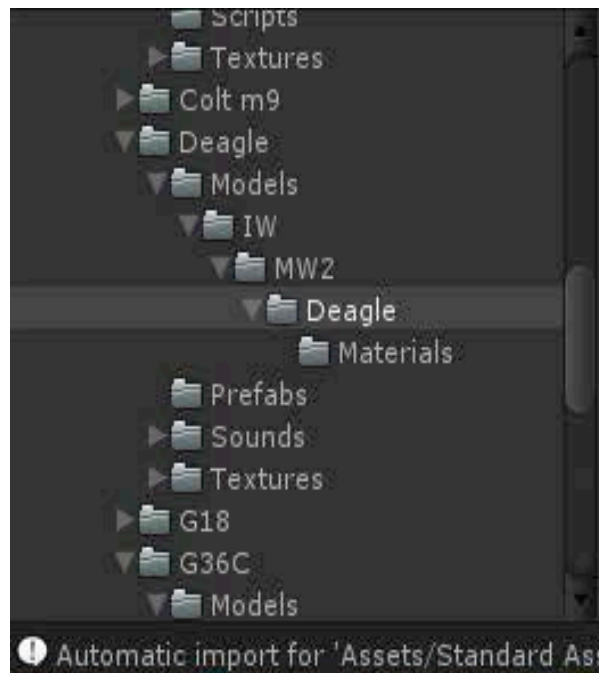


Рисунок 3.3 – Модульне подання системи: об’єкти з макетами, матеріалами, шаблонами, які ще не перенесені у віртуальний світ

3.4 Тестування програмного забезпечення

Тестування програми виконувалось на різних операційних системах: Windows Xp, Windows 7, Windows 8. Програма приймає білу кольорову схему.

До системи було додано 2 типи зброї, 3 типа локацій, модифіковану ЕС, 2 типи персонажів. При тестуванні розробленого ПП виняткових ситуації не виникало.

3.5 Висновки за розділом 3

Запропоновано основні рішення щодо подання даних системи. Описано основні розроблені алгоритми. Розроблені алгоритми можливо використовувати в інших проєктах. Особливості реалізації роблять систему легкою у використанні будь якій людині. Використано зручну модульну систему для легкого орієнтування у проєкті. Встановлені поля й методи, що

використовуються у системі, які знаходяться у файлах з розширенням «*js». Описано основні рішення щодо модульного подання системи.

Виконано тестування розробленого програмного забезпечення. Результати тестування підтвердили, що програма може використовуватися за призначенням.

4 КЕРІВНИЦТВО ПРОГРАМІСТА

4.1 Призначення й умови застосування програми

Програмний продукт призначений для організації проведення відпочинку користувача шляхом проходження місій комп'ютерної гри.

Для коректної роботи програму потрібно запускати в операційній системі Windows 7 або більш пізнішій версії операційної системи.

Програма не вимоглива до процесора та оперативної пам'яті, тому запуститься і швидко працюватиме на будь-якому комп'ютері з операційною системою Windows з використанням мінімуму ресурсів.

Для експлуатації програмного комплексу необхідні такі програмно-технічні засоби:

- ПК під керівництвом операційної системи;
- тактова частота процесора 3 ГГц або більше, кількість ядер – 2 або більше;
- обсяг оперативної пам'яті не менше 4 Гб;
- обсяг вільного простору на диску – не менше 200 Мб;
- дозвіл екрана монітора не менше 1024x768;
- маніпулятор миша.

4.2 Характеристики програми

Програма була створена у середі програмування Unity 3D. Програма дозволяє вводити ігровий нікнейм користувача, виконувати місії, зберігати данні (прогрес проходження гри), вільно пересуватися по ігровому світу та демонструє наявність інтелекту у ботів.

4.3 Звернення до програми

Розроблене програмне забезпечення поширюється у вигляді інсталяційного пакета. Для можливості роботи з програмою необхідно розмістити файли модулів програмного забезпечення на комп'ютері. Перед використанням програми її необхідно встановити. Для цього необхідно запустити програму setup.exe, яка знаходиться на установчому диску. У відповідь на запит установчої програми потрібно вказати шлях встановлення програми. Після цього програма скопіює необхідні файли, створить і помістить в головне меню системи ярлик для запуску програми і зробить необхідні зміни в файлі реєстрації.

Після завершення процедури установки програму можна використовувати. Програму можна запустити будь-якими засобами операційної системи користувача.

Для редагування програми необхідно відкрити пакетом Unity 3D 4.312 або вище, відкрити файли «Remember_all.sln». Якщо є необхідність, можливо провести експорт до інших ПП до інших платформ встановивши додаткове програмне забезпечення. Unity 3D містить засоби взаємозв'язку з платформами Linux, Mac os, android.

4.4 Вхідні і вихідні дані

Вхідними даними для програми є інформація щодо штучного ігрового світу.

Вихідними даними для програми є результати розрахунків щодо анімації ігрової сцени, стану героя користувача та ін.

4.5 Повідомлення

Реалізована програмна система гарантує її стійке функціонування. У випадку виникнення помилок в програмі передбачено відображення стандартних діалогових вікон. Подібні повідомлення можуть виникати при деяких небажаних ситуаціях, до яких можна віднести введення користувачем некоректних даних.

5 КЕРІВНИЦТВО ОПЕРАТОРА

5.1 Призначення програми

Програмний продукт призначений для організації проведення відпочинку користувача шляхом проходження місій комп'ютерної гри

Для коректної роботи програму потрібно запускати в операційній системі Windows 7 або більш пізнішій версії операційної системи.

Інтерфейс користувача забезпечує просту і ефективну взаємодію користувача з програмним продуктом, а також передбачає можливість його адаптації до вимог конкретного користувача.

5.2 Умови виконання програми

Для експлуатації програмного комплексу необхідні такі програмно-технічні засоби:

- ПК під керівництвом операційної системи;
- тактова частота процесора 3 ГГц або більше, кількість ядер – 2 або більше;
- обсяг оперативної пам'яті не менше 4 Гб;
- обсяг вільного простору на диску – не менше 200 Мб;
- дозвіл екрана монітора не менше 1024x768;
- маніпулятор миша.

Час роботи програми залежить від конфігурації комп'ютера, на якому вона виконується.

5.3 Виконання програми

Розроблене програмне забезпечення поширюється у вигляді інсталяційного пакета. Для можливості роботи з програмою необхідно розмістити файли модулів програмного забезпечення на комп'ютері. Перед

використанням програми її необхідно встановити. Для цього необхідно запуснути програму setup.exe, яка знаходиться на установчому диску. У відповідь на запит установчої програми потрібно вказати шлях встановлення програми. Після цього програма скопіює необхідні файли, створить і помістить в головне меню системи ярлик для запуску програми і зробить необхідні зміни в файлі реєстрації.

Після завершення процедури установки програму можна використовувати. Програму можна запуснути будь-якими засобами операційної системи користувача.

5.4 Приклад роботи з програмою

Після запуску програми автоматично відкриється основне вікно (рис. 5.1), в якому користувачу буде запропоновано створити нову гру.



Рисунок 5.1 – Створення нової гри

Після цього запусниться гра. Приклад інтерфейсу програми (комп'ютерної гри) наведено на рисунку 5.2.



Рисунок 5.2 – Інтерфейс програми

Програма дозволяє здійснювати такі операції:

- пошук об'єкта на карті в ігровому світі;
- виконання обробки знань системи, занованої на знаннях;
- збереження даних;
- підтримка ігрового меню (рис. 5.3);
- перемикання зброї за командою;
- перетворення ігрового об'єкту («смерть» бота);
- забезпечення анімації зброї (рис. 5.4);
- забезпечення анімації лісу (вітряна зона) (рис. 5.5);
- забезпечення перевірки на виконання місії;
- забезпечення отримання місії.



Рисунок 5.3 – Приклад ігрового меню



Рисунок 5.4 – Анімація перезарядження зброї



Рисунок 5.5 – Анімація лісу

5.5 Повідомлення оператору

Як відзначено вище, у випадку виникнення помилок в програмі передбачено відображення стандартних діалогових вікон. Подібні повідомлення можуть виникати при деяких небажаних ситуаціях, до яких можна віднести введення користувачем некоректних даних.

6 ЕКОНОМІКО-ОРГАНІЗАЦІЙНА ЧАСТИНА

У дипломній кваліфікаційній роботі магістра досліджено та реалізовано комп'ютерну гру на основі інтелектуальних технологій.

Очікується, що готове програмне забезпечення може бути реалізовано у кількості 500 копій по 250 грн. кожна.

У розробці беруть участь програміст із заробітною платою 10000 грн. в місяць і консультант з заробітною платою 8000 грн. у місяць. На розробку проєкту було витрачено 2 місяці. Розробка програми розпочалася першого вересня і повинна бути виконана до кінця жовтня 2020 року.

6.1 Планування розробки програмного продукту

Весь комплекс розробки програмного продукту можна розділити на етапи [21]. Для кожного етапу вказуються трудомісткість, кількість виконавців і тривалість робіт. Інформація з характеристики робіт по розробці програмного забезпечення зведена у таблиці 6.1.

Таблиця 6.1 – Характеристика робіт по розробці програми

Найменування робіт	Трудомісткість		Виконавці	Тривалість, днів
	люд-дн	%		
Аналіз ПЗ	4	6,0	Програміст Консультант	2 2
Визначення вимог до ПП	12	17,9	Програміст Консультант	6 6
Проектування структури програми	28	41,8	Програміст Консультант	14 14
Розробка схеми функціонування програми	3	4,5	Програміст	3
Створення програмного коду	7	10,4	Програміст	7
Тестування і налагодження програмного продукту	9	13,4	Програміст	9
Складання програмної документації	4	5,97	Програміст	4
Разом	67	100	Програміст Консультант	45 22

6.2 Визначення витрат на розробку програми

Для визначення витрат на розробку програми складають калькуляцію кошторисної вартості робіт, що включає статті:

- основна заробітна плата;
- додаткова заробітна плата;
- відрахування на єдиний соціальний внесок (ЄСВ);
- матеріали та комплектуючі вироби;
- витрати на спеціальне обладнання (на оплату машинного часу ЕОМ для написання і налагодження програмних засобів);
- накладні витрати.

Витрати за статтею «Основна заробітна плата» складаються з планового фонду зарплати всіх категорій працівників, зайнятих в розробці програми. Розрахунок зарплати на підставі даних про трудомісткість, представлений в таблиці 6.2.

Таблиця 6.2 – Основна заробітна плата

Посада виконавця	Чисельність, чол	місячний оклад	Кількість місяців роботи	Сума ЗП, грн
Програміст	1	10000	2	20000
Консультант	1	8000	1	8000
Разом	2			28000

Додаткову заробітну плату приймають рівною 10% від основної зарплати. Вона розраховується за формулою (6.1):

$$ЗП_{\text{дод}} = ЗП_{\text{осн.}} \cdot 10\%. \quad (6.1)$$

Підставивши величину основної заробітної плати в дану формулу отримаємо: $ЗП_{\text{дод}} = 28000 \cdot 0.1 = 2800$ грн.

Відрахування на єдиний соціальний внесок (ЄСВ) визначають у відсотковому відношенні від сум основної та додаткової зарплати з урахуванням премій і доплат. Відрахування вираховуються за формулою (6.2):

$$ОТ = (ЗП_{\text{осн.}} + ЗП_{\text{дод.}}) \cdot 0,22. \quad (6.2)$$

$$ОТ = (28000 + 2800) \cdot 0,22 = 6776 \text{ грн.}$$

Використовується три найменування матеріалів: картридж – 700 грн, папір – 300 грн, флеш-носій – 200 грн. Витрати на матеріали розраховують за формулою (6.3):

$$M = \sum_{i=1}^n (C_i \cdot N_i \cdot (1 + K_{m.з.}) - C_{io} \cdot N_{io}), \quad (6.3)$$

де M – витрати на матеріали, покупні напівфабрикати і комплектуючі вироби, грн.;

$K_{m.з.}$ – коефіцієнт, що враховує транспортно-заготівельні витрати;

C_i – ціна i -го найменування матеріалу, напівфабрикату і комплектуючого, грн.;

N_i – потреба в i -му матеріалі, напівфабрикаті і комплектуючому;

C_{io} – ціна зворотних відходів i -го найменування матеріалу, грн;

N_{io} – кількість зворотних відходів i -го найменування;

n – кількість найменувань матеріалів, напівфабрикатів і комплектуючих.

$$C_{io} = 0; \quad K_{m.з.} = 0,05;$$

$$M = (1 + 0,05) \cdot (700 + 300 + 200) = 1260 \text{ грн.}$$

Таким чином, разом витрати на матеріали та комплектуючі вироби становлять 1260 гривень.

У статтю «Витрати на спеціальне обладнання» входять витрати на придбання, транспортування, монтаж і налагодження нестандартного обладнання. Практично, в даному випадку, в цій статті враховуються витрати на оплату машинного часу ЕОМ для написання і налагодження програмних засобів. Для цього необхідно скласти кошторис «витрат на утримання і експлуатацію устаткування», виходячи з якого визначиться вартість одного машино-години роботи ПК, після множення якої на машинний час пішло на написання і налагодження програми отримаємо витрати на оплату машинного часу.

Амортизаційні відрахування визначають за формулою (6.4):

$$A = BV \cdot \frac{H_a}{100}, \quad (6.4)$$

де BV – балансова вартість обчислювальної техніки, грн;

H_a – норма амортизаційних відрахувань на повне оновлення обчислювальної техніки,%. Приймаємо, що техніка повинна працювати 4 роки, тому $H_a = 25\%$.

Балансова вартість обчислювальної техніки становить 32000 грн. Тому амортизаційні відрахування становлять:

$$A = 32000 \cdot 0,25 = 8000 \text{ грн}$$

Статтю "Експлуатація обладнання" розраховують підсумовуванням витрат на силову електроенергію і допоміжні матеріали. Витрати на електроенергію визначають за формулою (6.5):

$$C_{\text{э}} = N_{\text{н}} \cdot \Phi_{\text{эф}} \cdot K_{\text{зв}} \cdot K_{\text{зм}} \cdot C_{\text{э}}, \quad (6.5)$$

де $N_{\text{н}}$ – номінальна потужність ЕОМ, кВт;

$\Phi_{\text{эф}}$ – річний ефективний фонд часу роботи ЕОМ, машино–год;

$K_{\text{зв}}$ – середній коефіцієнт завантаження за часом;

$K_{\text{зм}}$ – коефіцієнт завантаження по потужності;

$C_{\text{э}}$ – ціна одного кВт · год електроенергії, грн ./ (кВт · год).

Номінальна Потужність ЕОМ – 0,2 кВт. Річний ефективний фонд часу роботи ЕОМ ставити 1800 годин. Середні коефіцієнти завантаження за часом та за потужністю рівні відповідно 0,9 та 0,6. Ціна однієї кіловат–години електроенергії ставить 2,11 грн. Отримуємо:

$$C_e = 0,2 \cdot 1800 \cdot 0,9 \cdot 0,6 \cdot 2,11 = 410,18 \text{ грн.}$$

Приймаємо, що $C_e = 410$ грн.

Витрати за статтею "Заробітна плата обслуговуючих робітників та відрахування на соціальне страхування в інші фонди" визначають за формулою (6.6):

$$ЗП_{обсл.} \cdot \Phi ЗП \cdot (1 + K_{відрах..}) \cdot \frac{t_{обсл.}}{\Phi_{ef.обсл.}}, \quad (6.6)$$

де $\Phi ЗП_2$ – річний фонд заробітної плати (основної та додаткової) обслуговуючих робітників, грн.;

$K_{відрах}$ – коефіцієнт, що враховує відрахування на соціальне страхування і в інші фонди;

$t_{обсл}$ – час протягом року, необхідне технічне обслуговування ЕОМ, годин / рік. Приймаємо $t_{обсл} = 24$ год.;

$\Phi_{ef.обсл}$ – річний ефективний фонд часу обслуговуючого персоналу, годин / рік. Приймаємо, що річний ефективний фонд часу обслуговуючого персоналу становить 1800 годин

Заробітна плата обслуговуючого персоналу = 8500 грн/місяць.

$$\Phi ЗП_2 = 8500 \cdot 12 = 102000 \text{ грн.}$$

$$ЗП_{обсл} = 102000 \cdot (1 + 0,22) \cdot 24 / 1800 = 1659 \text{ грн.}$$

Суму витрат за статтею "Поточний ремонт обладнання" обчислимо як 3% від балансової вартості обладнання, оскільки неможливо заздалегідь визначити, скільки коштів необхідно витратити на придбання запасних частин наявного обладнання. Балансова вартість обчислювальної техніки становить 32000 грн. Тому сума витрат за статтею "Поточний ремонт обладнання" становить: $32000 \cdot 0,03 = 960$ грн.

Інших витрат за проектом не передбачено, тому сума витрат за цією статтею дорівнює нуль гривень.

Кошторис витрат на утримання і експлуатацію устаткування наведений в таблиці 6.3.

Таблиця 6.3 – Кошторис витрат на утримання і експлуатацію устаткування

Найменування статей витрат	Сума, грн.
Амортизація обладнання	8000
Заробітна плата основна і додаткова обслуговуючих робітників з відрахуваннями на соціальні заходи	1659
Експлуатація обладнання (крім витрат на поточний ремонт)	410
Поточний ремонт обладнання	960
Інші витрати	0
Разом	11029

Витрати на оплату машинного часу ЕОМ для написання і налагодження програмних засобів визначаються за формулою (6.7):

$$C_{mo} = P_{екс} \cdot t_{mo}, \quad (6.7)$$

де C_{mo} – витрати на оплату машинного часу, грн;

$P_{екс}$ – експлуатаційні витрати на одну годину машинного часу цієї цифрової ЕОМ, грн. / машино–год.;

t_{mo} – машинний час цифрової ЕОМ для написання і налагодження даного програмного продукту, машино–год.

$$P_{екс} = 11029/1800 = 6,13 \text{ грн. / машино–год.}$$

Для написання і налагодження даного програмного продукту ЕОМ експлуатується протягом 45 днів в одну зміну по 8 годин, що складає в сумі 360 годин. Таким чином отримуємо суму за статтею «Витрати на спеціальне обладнання»:

$$C_{\text{мо}} = 6,13 \cdot 360 = 2207 \text{ грн.}$$

До накладних витрат відносяться витрати на загальне управління і загальногосподарські потреби (заробітна плата апарату управління, канцелярські витрати і т.д.), утримання та експлуатацію будівель. Накладні витрати включаються до вартості розробки програми непрямым шляхом – у відсотках до основної заробітної плати розробників. Для обчислення цієї статті видатків у дипломному проєкті приймемо, що накладні видатки становлять 25% від основної заробітної плати розробників, що складає 28000 грн. Тому сума накладних видатків становить: $28000 \cdot 0,25 = 7000$ грн.

Калькуляція кошторисної вартості робіт з розробки програми наведена в таблиці 6.4.

Таблиця 6.4 – Калькуляція кошторисної вартості робіт з розробки програми

Найменування статей витрат	Сума, грн.
Основна заробітна плата	28000
Додаткова заробітна плата	2800
ЄСВ	6776
Матеріали і комплектуючі	1260
Витрати на спеціальне обладнання	2207
Накладні витрати	7000
Разом	48043

6.3 Розрахунок економічної ефективності розробки програмного продукту

Для оцінювання точки беззбитковості у загальній кошторисній вартості розробки врахуємо також величину податку на додану вартість при продажу продукції, що складає 20%:

$$KB_{зПДВ} = KB \cdot (1 + 0,2) = 48043 \cdot (1 + 0,2) = 57652 \text{ грн}, \quad (6.8)$$

де $KB_{зПДВ}$ та KB – кошторисна вартість розробки програмного забезпечення з врахуванням ПДВ та без врахування ПДВ, відповідно.

Розрахуємо точку беззбитковості ТБ (мінімальну кількість виробленої продукції, яку необхідно продати, щоб не отримати збитки). При цьому врахуємо, що очікувана вартість B_1 однієї копії розробленого програмного забезпечення складає 250 грн. (вважаємо, що ПЗ розповсюджується через мережу Інтернет, без використання фізичних носіїв):

$$ТБ = KB_{зПДВ} / B_1 = 57652 / 250 = 231 \text{ шт.}$$

Таким чином, мінімальна кількість версій розробленого програмного забезпечення, яку необхідно продати, щоб не отримати збитки, складає 231 шт.

Розрахуємо плановий прибуток ППр від розробки програмного забезпечення (очікується, що готове програмне забезпечення може бути реалізовано у кількості $K = 500$ копій по 250 грн. кожна):

$$ППр = B_1 \cdot (K - ТБ) / (1 + ПДВ) = 250 \cdot (500 - 231) / (1 + 0,2) = 56041 \text{ грн.}$$

Коефіцієнт ефективності капітальних вкладень E и строк їхньої окупності $T_{ок}$ розраховуються за формулами (6.9) і (6.10):

$$E = \frac{ППр}{KB}, \quad (6.9)$$

$$T_{ок} = \frac{KB}{ППр}, \quad (6.10)$$

$$E = 56041 / 48043 = 1,17,$$

$$T_{ок} = 48043 / 56041 = 0,86 \text{ року.}$$

Зведемо отримані техніко–економічні показники в таблиці 6.5.

Таблиця 6.5 – Техніко–економічні показники

Показники	Сума	Одиниця виміру
Витрати на розробку програми	48043	грн.
Точка беззбитковості реалізації програмного забезпечення	231	шт.
Строк окупності програми	0,86	рік

Таким чином, розробка програмного забезпечення у вигляді комп'ютерної гри дозволяє підприємству одержувати прибуток у випадку продажу більше, ніж 231 копій програми. Економічні розрахунки показали, що строк окупності розробки програми складає 0,86 року, що значно менше часу її нормативного строку окупності (5 років). Виходячи з виконаних розрахунків, можна зробити висновок, що розробка даного програмного продукту є економічно доцільною.

7 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА У НАДЗВИЧАЙНИХ СИТУАЦІЯХ

7.1 Аналіз потенційних небезпек

Дослідження та програмна реалізація комп'ютерної гри на основі інтелектуальних технологій передбачає використання ПК, робота з якими відноситься до праці з високою нервово-емоційною напругою. Це обумовлено необхідністю постійного спостереження за динамікою зображення, розрізнення тексту рукописних і друкованих матеріалів, виконання машинописних і графічних робіт.

Робота за комп'ютером пов'язана зі значною концентрацією уваги, зоровою напругою, емоціональним та психічним навантаженням. Під час роботи комп'ютера створюються специфічні умови мікроклімату: підвищується температура повітря, знижується вологість, змінюється іонний склад повітря, мають місце слабка вібрація і шум. Спостерігається також підвищений рівень електромагнітних випромінювань, статична електрика.

При дослідженні та програмній реалізації комп'ютерної гри на основі інтелектуальних технологій на розробника діють наступні потенційні небезпеками.

Ураження електричним струмом. Може виникнути у разі несправності оргтехніки, неправильного проведеного або відсутнього заземлення, порушення ізоляції дротів або порушення правил техніки безпеки робітником.

Механічні травми. Виникають внаслідок неправильної організації робочого місця працівника.

Статичні фізичні перевантаження. Виникають в наслідок порушень у організації робочого місця, монотонністю роботи коли від робітника потребується тривале статичне напруження м'язів шиї, спини, рук та ніг.

Психологічні перевантаження. Викликані стресовими ситуаціями, складними відносинами у колективі, нервовими перевантаженнями, надмірним

ускладненням робочого процесу. Призводять до загальних захворювань нервової системи, головного болю, підвищеної втомлюваності, втрати уваги.

Вплив електромагнітного випромінювання (ЕМВ). Обумовлений тривалою роботою людини у приміщенні з великою кількістю електронної апаратури та комп'ютерів.

Недостатня освітленість робочого місця. Обумовлена виходом з ладу ламп освітлення, неправильною розстановкою освітлювальних приладів чи недотриманням режиму праці.

Незадовільні кліматичні умови. Можуть бути викликані недостатньою вентиляцією робочого приміщення, слабким опаленням взимку або недостатнім кондиціонуванням повітря влітку.

Можливість загоряння. Обумовлена можливістю коротких замикань у обладнанні, наявністю устаткування з високою температурою роботи, порушеннями техніки безпеки.

Неправильні дії персоналу в умовах пожежі. Виникають внаслідок недотримання правил евакуації з палаючого приміщення.

7.2 Заходи щодо забезпечення безпеки

За ступенем небезпеки ураження електричним струмом, відповідно до ПУЕ «Правила улаштування електроустановок» [22], приміщення обчислювального центру (ОЦ) відноситься до категорії приміщень без підвищеної небезпеки.

Заходи по забезпеченню електробезпеки розроблено у відповідності до ДСТУ 7237:2011 «Система стандартів безпеки праці. Електробезпека. Загальні вимоги та номенклатура видів захисту» [23]. За способом захисту людини від ураження електричним струмом – відеотермінали, ЕОМ, периферійні пристрої ЕОМ та устаткування для обслуговування, ремонту та налагодження ЕОМ відповідають першому класу захисту та заземлені відповідно до

ДНАОП 0.00-1.32-01 Правила устро́йства електроустановок. электрооборудование специальных установок [24].

Для забезпечення електробезпеки у ОЦ використано наступні технічні заходи та засоби електробезпеки:

- недоступність до струмопровідних частин;
- захисне відключення;
- ізоляція струмопровідних частин;
- захисне заземлення;
- блокування, занулення за ПУЕ-2017;
- захисне розділення електричних мереж ДСТУ ІЕС 61140:2015 «Захист проти ураження електричним струмом. Загальні аспекти щодо установок та обладнання (ІЕС 61140:2001, IDT)» [25];
- компенсація ємкісних струмів замикання на землю;
- вирівнювання потенціалів тощо.

Сіть електромережі у приміщенні ОЦ вмикається/вимикається за допомогою пускової апаратури.

У приміщенні передбачено використання тільки трьохполюсних розеток, один з полюсів яких підключено до шини захисного заземлення. З цим полюсом електрично з'єднані металеві частини обчислювальних машин, які можуть виявитися під напругою.

Кожний блок живлення комп'ютера або периферійного пристрою має сітьовий фільтр. Конденсатори цього фільтру призначені для шунтування високочастотних поміх живильної мережі на землю через дріт «земля» трьохполюсної вилки та розетки. Цей дріт з'єднаний з заземлюючим контуром і з «нулем» силової мережі. При відсутності гальванічного зв'язку заземлюючого дроту з шиною захисного заземлення, на корпусі електронного пристрою може з'явитися напруга порядку 110 В, внаслідок поділення напруги мережі 220 В конденсаторами сітьового фільтру, що працює у даному випадку як ємнісний роздільник напруги [26].

Відповідно до ПУЕ опір захисного заземлення у приміщенні ОЦ не перевищує 4 Ом [22].

Електрична ізоляція забезпечується конструкціями пристроїв і способами їхнього підключення. Відповідно до ПУЕ в електроустановках до 1000 В мінімальне значення опору ізоляції силових і освітлювальних електропроводок на ділянці між суміжними запобіжниками або за останніми запобіжниками між будь-яким проведенням і землею становить не менш 0,5 МОм [22].

Розрахуємо захисне заземлення для приміщення з розмірами $L = 7\text{м}$, $B = 3\text{м}$, $R_{\text{е}} = 8\text{ Ом}$, $U = 380\text{ В}$.

Параметри вертикально стрижня: $l_{\text{с}} = 3\text{м}$, $d = 0,012\text{м}$,

Ширина смуги $b_{\text{с}} = 0,025\text{м}$, $\rho_{\text{вим}} = 92\text{ Ом}\cdot\text{м}$

Кліматична зона III, склад ґрунту – однорідний, вологість повітря – нормальна.

Визначаємо необхідний опір штучного заземлювача $R_{\text{ш}}$, Ом, якщо передбачається використання також природного заземлювача $R_{\text{п}}$, за формулою:

$$R_{\text{ш}} = \frac{R_{\text{е}} R_{\text{з}}}{R_{\text{е}} - R_{\text{з}}}, \text{ Ом};$$

де $R_{\text{е}}$ – опір розтіканню струму природних заземлювачів, Ом;

$R_{\text{з}}$ – розрахунковий нормований опір заземлювального пристрою (ЗП), Ом.

$$R_{\text{е}} = 8, R_{\text{з}} = 4, \\ R_{\text{ш}} = \frac{R_{\text{е}} R_{\text{з}}}{R_{\text{е}} - R_{\text{з}}} = \frac{8 \cdot 4}{8 - 4} = 8.$$

Визначимо розрахунковий питомий опір ґрунту за формулою:

$$\rho = \rho_{\text{вим}} \cdot \Psi, \text{ Ом}\cdot\text{м};$$

де ρ – розрахунковий питомий опір ґрунту, Ом·м;

$\rho_{\text{вим}}$ – питомий опір землі, отриманий у результаті вимірів, Ом·м;

Ψ – коефіцієнт сезонності, що враховує промерзання чи висихання ґрунту

$$\rho = \rho_{\text{вим}} \cdot \Psi = 92 \cdot 1,3 = 119,6 \text{ Ом} \cdot \text{м}.$$

Обчислимо опір розтіканню струму одиночного вертикального заземлювача $R_{\text{г}}$, Ом. Для стрижневого заземлювача круглого перерізу, заглибленого в ґрунт розрахункова формула має вигляд:

$$R_{\text{г}} = \frac{\rho}{2 \cdot \pi \cdot l_{\text{г}}} \left(\ln \frac{2 \cdot l_{\text{г}}}{d} + \frac{1}{2} \cdot \ln \frac{4 \cdot t_{\text{г}} + l_{\text{г}}}{5 \cdot t_{\text{г}} - l_{\text{г}}} \right), \text{ Ом};$$

де ρ – розрахунковий питомий опір ґрунту, Ом·м; $\rho = 119,6 \text{ Ом} \cdot \text{м}$;

$l_{\text{г}}$ – довжина вертикального стрижня, м; $l_{\text{г}} = 3 \text{ м}$;

d – діаметр перерізу стрижня, м; $d = 0,012 \text{ м}$

$t_{\text{г}}$ – відстань від поверхні ґрунту до середини довжини вертикального стрижня, яка обчислюється за формулою:

$$t_{\text{г}} = 0,8 + \frac{1}{2} l_{\text{г}}, \text{ м}.$$

$$t_{\text{г}} = 0,8 + \frac{1}{2} l_{\text{г}} = 0,8 + 1/2 \cdot 3 = 0,8 + 1,5 = 2,3 \text{ м}.$$

Тоді $R_{\text{г}} = 40,58 \text{ Ом}$.

Розрахуємо наближену (мінімальну) кількість вертикальних стрижнів

$$n' = \frac{R_{\text{г}}}{R_{\text{у}}}, \text{ шт};$$

де $R_{\text{г}}$ – опір розтікання струму одиночного вертикального заземлювача, Ом; $R_{\text{г}} = 40,58 \text{ Ом}$

R_u – необхідний опір штучного заземлювача, Ом; $R_u = 8,00$ Ом.

$$n' = \frac{R_{\epsilon}}{R_u} = \frac{40,58}{8} = 5,07 \text{ шт.}$$

Якщо виходити з розмірів контуру:

$$n = \frac{P}{a}, \text{ шт;}$$

де P – периметр прямокутника (приміщення), м. Який визначається за формулою:

$$P = 2 \cdot (L + B), \text{ м;}$$

L – довжина приміщення, м;

B – ширина приміщення, м;

$$P = 2 \cdot (7 + 3) = 20.$$

a – відстань між стрижнями, обумовлена зі співвідношення, м:

$$a = k \cdot l_{\epsilon};$$

де l_{ϵ} – довжина вертикального стрижня, м;

k – коефіцієнт кратності, який дорівнює 1, 2, 3 (для заглиблених стаціонарних заземлювачів $k = 1$).

$$a = k \cdot l_{\epsilon} = 1 \cdot 3 = 3 \text{ м.}$$

Тоді $n = \frac{P}{a} = \frac{20}{3} = 6,67$ приймаємо $n = 8$ шт.

Отриману кількість стрижнів округляють до більшого довідкового значення, але не менше n' .

Визначаємо довжину горизонтальної смуги при конфігурації групового заземлювача – контур:

$$l_z = 1,05 \cdot a \cdot n, \text{ м};$$

де n – кількість вертикальних стрижнів; $n = 8$ шт.

a – відстань між вертикальними стрижнями, м; $a = 3$ м

$$l_z = 1,05 \cdot a \cdot n = 1,05 \cdot 8 \cdot 3 = 25,20 \text{ м.}$$

Обчислити опір розтіканню струму горизонтальної з'єднуючої смуги R_z , Ом. У випадку горизонтального смугового заземлювача розрахунок виконується за формулою:

$$R_z = \frac{\rho}{2 \cdot \pi \cdot l_z} \cdot \ln \frac{2 \cdot l_z^2}{b_c \cdot t_z}, \text{ Ом};$$

де ρ – розрахунковий питомий опір ґрунту, Ом·м;

l_z – довжина горизонтальної смуги, м;

b_c – ширина смуги, м;

t_z – відстань від поверхні ґрунту до середини ширини горизонтальної смуги, яка обчислюється за формулою:

$$t_z = 0,8 + \frac{1}{2} b_c, \text{ м.}$$

$$t_z = 0,8 + \frac{1}{2} \cdot 0,025 = 0,81 \text{ м.}$$

Тоді:

$$R_z = \frac{\rho}{2 \cdot \pi \cdot l_z} \cdot \ln \frac{2 \cdot l_z^2}{b_c \cdot t_z} = 8,34 \text{ Ом}$$

Розрахувати еквівалентний опір розтіканню струму групового заземлювача.

$$R_{zp} = \frac{R_\theta \cdot R_z}{R_\theta \cdot \eta_z + R_z \cdot \eta_\theta \cdot n}, \text{ Ом};$$

де R_θ – опір розтікання струму одиночного вертикального заземлювача, Ом;

R_z – опір розтікання струму горизонтальної смуги, Ом;

η_θ, η_z – коефіцієнти використання вертикальних стрижнів і горизонтальної смуги, Ом (додаток Б, В); $\eta_\theta = 0,59$; $\eta_z = 0,37$

n – кількість вертикальних стрижнів.

$$R_{zp} = \frac{R_\theta \cdot R_z}{R_\theta \cdot \eta_z + R_z \cdot \eta_\theta \cdot n} = 6,22 \text{ Ом.}$$

Отриманий опір розтіканню струму групового заземлювача не повинен перевищувати необхідний опір, визначений у пункті 2.

$$R_{zp} \leq R_u;$$

$$6,22 \text{ Ом} \leq 8,00 \text{ Ом.}$$

Умова виконана, рішення завдання вірне.

Основним джерелом електромагнітного випромінювання є монітор та системний блок комп'ютера. Безпечні для здоров'я рівні ЕМВ визначаються нормами МРР-II 1990:10 та ДСанПіН 3.3.2.007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-

обчислювальних машин» [27]. Відповідно до цих стандартів допустима напруженість електромагнітного поля на відстані 0,5м. по електричній складовій складає:

- для частот 5 Гц — 2 кГц: 25 В/м;
- для частот 2 кГц — 400 кГц: 2,5 В/м.

Максимально допустима щільність магнітного потоку:

- для частот 5 Гц — 2 кГц: 250 нТл;
- для частот 2 кГц — 400 кГц: 25 нТл.

Відповідно, для використання обрано монітор, який відповідає вищезазначеним нормативним документам — LG Flatron W2252TQ.

7.3 Заходи щодо виробничої санітарії та гігієни праці

Оптимальні норми температури повітря в робочій зоні виробничих приміщень у холодний і перехідний періоди року при II категорії робіт згідно ДСН 3.3.6.042-99 «Санітарні норми мікроклімату виробничих приміщень» [28] – 22...24° С, відносна вологість – 60...40%, швидкість руху повітря – 0.2 м/сек. У теплий період року температура – 23...25° С, відносна вологість – 60...40%, швидкість руху повітря – 0.2 м/сек.

Для забезпечення вищевказаних параметрів передбачене застосування вентиляції - природної (аерації й провітрювання), механічної (загальнобмінної, місцевої приточної і витяжної), кондиціювання повітря.

У приміщенні забезпечений приплив свіжого повітря, кількість якого визначається техніко-економічним розрахунком і вибором схеми системи вентиляції. Мінімальна витрата повітря визначається з розрахунку 50...60 м³/год на одну людину, але не менш дворазового повітрообміну в годину.

Всі норми по освітленості беруться згідно ДБН В.2.5-28:2018 «Природне та штучне освітлення» [29]. Штучне освітлення в приміщеннях здійснюється системою загального рівномірного освітлення. При

цьому штучне освітлення застосовують не тільки в темний, але й у світлий час доби.

Найбільш прийнятними для приміщень є люмінесцентні лампи ЛБ (білого світла) потужністю 20, 40 або 80 Вт, які встановлюються в світильники типу ЛСП. Допускається застосування ламп накаливання у світильниках місцевого освітлення.

Загальне освітлення варто виконувати у виді суцільних чи переривчастих ліній світильників, розташованих збоку від робочих місць.

Згідно з ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку» [30] зниження шуму, створюваного на робочих місцях внутрішніми джерелами, а також шуму проникаючого ззовні, є дуже важливим завданням. Зниження шуму в джерелі виникнення забезпечується застосуванням пружних прокладок між основою машини, приладу й опорною поверхнею. Як прокладки використовуються гума, повсть, пробка, різної конструкції амортизатори. Під настільні шумливі апарати можна підкладати м'які килимки із синтетичних матеріалів, а під ніжки столів, на яких вони встановлені, - прокладки з м'якої гуми, повсті, товщиною 6 - 8 мм.

Рівні вібрації під час виконання робіт з ЕОМ у виробничих приміщеннях не перевищують допустимих значень, визначених у ДСанПіН 3.3.2.007-98 «Державних санітарних правил і норм роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [27] та ДСН 3.3.6-039-99 «Державні санітарні норми виробничої загальної та локальної вібрації».

Нормування технологічної вібрації (загальної і локальної) відбувається:

- в залежності від її спрямування у кожній октавній смузі (1,6-1000 Гц);
- з середньоквадратичними віброшвидкостями $(1,4-0,28) \cdot 10^{-2} \text{ м/с}$;
- логарифмічними рівнями віброшвидкості (115-109 дБ);
- віброприскоренням $(85-0,1) \text{ м/с}^2$ [26].

7.4 Заходи з пожежної безпеки

Комплекс протипожежних заходів для офісного приміщення обладнаного ПК з ВДТ розроблений згідно вимог НАПБ А.01.001-2014 «Правила пожежної безпеки в Україні».

Виходячи з аналізу речовин та матеріалів, які використовуються при роботі у приміщенні обладнаному ПК з ВДТ:

- згідно ДСТУ EN 2:2014 «Класифікація пожеж (EN 2:1992, EN 2:1992/A1:2004, IDT)» у офісному приміщенні обладнаному ПК з ВДТ можлива пожежа класів – А (пожежа, що супроводжується горінням твердих матеріалів) та Е (горіння електроустановок, що перебувають під напругою до 1000 В);

- відповідно до вимог ДСТУ Б В.1.1-36:2016 «Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою», воно належить до категорії «Д» з пожежної небезпеки – простір у приміщенні, у якому перебувають тверді горючі речовини та матеріали [31].

Оскільки офісне приміщення обладнане ПК з ВДТ належить до категорії «Д» з пожежної небезпеки, тому відповідно до вимог ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги» воно має II ступінь вогнестійкості [32].

Обладнання, силові та освітленні мережі офісного приміщення обладнаного ПК з ВДТ відповідають вимогам пожежної безпеки, оскільки виконані відповідно до вимог НПАОП 40.1-1.32-01 «Правила будови електроустановок. Електрообладнання спеціальних установок», та мають ступінь захисту ізоляції обладнання IP44 яка відповідає класу пожежанебезпечної зони П-Па до якої належить приміщення.

З технічних та організаційних заходів запобігання пожеж в офісному приміщенні обладнаному ПК з ВДТ передбачені наступні протипожежні заходи:

- згідно вимог ДБН В.2.5-56:2014 «Системи протипожежного захисту» [33], в офісному приміщенні обладнаному ПК з ВДТ встановлена система

пожежної й охоронної сигналізації «Сигнал-ВК 6». Яка забезпечує виявлення теплових і димових ознак пожежі і місця виникнення пожежі з точністю до місця розміщення датчика;

– оскільки офісне приміщення що обладнане ПК з ВДТ має площу 21 м², тому відповідно до вимог п. 5 розділу VI «Вибір типу та необхідної кількості вогнегасників», «Правил експлуатації та типових норм належності вогнегасників», затверджених наказом МВСУ 15.01.2018 № 25 та зареєстрованих в МЮУ 23.02.2018 р. за № 225/31677 для гасіння електроустановок, що знаходяться під напругою, передбачені вуглекислотні вогнегасники типу ВВК-3,5 у кількості 2 штуки (з розрахунку один вогнегасник с величиною заряду вогнегасної речовини 3 кг і більше, на 20 м² площі приміщення). Додатково, на кожному поверсі будівлі, в якій розміщене приміщення обладнане ПК з ВДТ, передбачене два переносних порошкових вогнегасника – ВП-5. Відстань між вогнегасниками та місцями можливих загорянь не перевищує 10 м.

7.5 Заходи з цивільного захисту

Засоби індивідуального захисту використовуються для захисту людини від отруйних речовин, сильнодіючих отрут, радіоактивних речовин і бактеріологічних засобів і за своїм призначенням діляться на засоби захисту органів дихання та засоби захисту шкіри [34].

Найпростіші засоби захисту органів дихання — протипилова тканинна маска і ватно-марлева пов'язка. Вони захищають органи дихання від радіоактивного пилу і деяких видів бактеріологічних засобів, а для захисту від отруйних речовин непридатні.

Респіратори застосовують для захисту органів дихання від радіоактивного і ґрунтового пилу. Найбільш поширені респіратори Р-2 і ШБ-1 («Пелюсток»).

Респіратор Р-2 складається із фільтрувальної напівмаски з наголовником. Напівмаска має два клапани вдиху і один — видиху. У клапана видиху є запобіжний екран. Зовнішня частина напівмаски виготовлена з поліуретанового пінопласту зеленого кольору, а внутрішня — з тонкої повітронепроникної плівки, до якої вмонтовано два клапани вдиху. Між поліуретаном і плівкою є фільтр із полімерних волокон. Респіратор має носовий затискач, щоб притиснути напівмаску до перенісся.

Фільтрувальний протигаз призначений для захисту органів дихання, очей, шкіри обличчя від впливу отруйних речовин, радіоактивних речовин, бактеріологічних засобів і деяких сильнодіючих отруйних речовин, а також від різних шкідливих домішок, що є у повітрі.

У системі цивільної оборони країни для захисту дорослого населення використовуються наступні фільтрувальні протигази: ЦП-5, ЦП-5М, ЦП-7.

До складу комплекту цивільного фільтрувального протигаза ЦП-5 входять два основні елементи: лицьова частина ШМ-62У і фільтрувально-поглинальна коробка ЦП-5. Крім того, протигаз комплектується сумкою для зберігання і перенесення та коробкою з плівками, що не запотівають.

Протигаз ЦП-5М має шолом-маску ШМ-66МУ з переговорним пристроєм мембранного типу і вирізи для вух.

Ізолюючі протигази ІП-46, ІП-4, ІП-5 забезпечують захист органів дихання, очей і шкіри від будь-яких сильнодіючих отруйних речовин, незалежно від властивостей і концентрації. Вони дають змогу працювати навіть там, де зовсім немає кисню у повітрі. У протигазі ІП-46М або ІП-5 можна виконувати неважку роботу під водою на глибині до 7 м. Принцип роботи ізолюючих протигазів ґрунтується на виділенні кисню з хімічних речовин при поглинанні вуглекислого газу і вологи, які видихаються людиною.

Медичні засоби індивідуального захисту — це аптечка індивідуальна (АІ-2), індивідуальний протихімічний пакет (ІПП-8) і пакет перев'язувальний індивідуальний. В інструкції, що додається до АІ-2, описано порядок прийому ліків.

Індивідуальний протихімічний пакет ПП-8 призначений для знезараження краплиннорідких отруйних речовин, що потрапили на відкриті ділянки шкіри та одяг. До комплекту входять флакон з дегазувальним розчином і чотири ватно-марлеві тампони. Посуд і тампони запаяні в герметичну оболонку з поліетилену. При використанні ПП-8 тампони щедро змочують розчином. Першим тампоном видаляють із шкірного покриву отруйні речовини щипковим рухом, наступна обробка — протирання ураженої ділянки.

Звичайні засоби захисту шкіри — це предмети одягу та взуття, що можуть бути у кожної людини. Найпростішим засобом захисту шкіри є робочий одяг (спецівка) — куртка і штани, комбінезони, халати з капюшонами, зшиті з брезенту, вогнезахисної чи прогумованої тканини або грубого сукна. Вони не тільки захищають шкіру від радіоактивних речовин і бактеріологічних засобів, а й не пропускають протягом деякого часу краплиннорідкі отруйні речовини.

Із предметів побутового одягу найпридатнішими для захисту шкіри є плащі і накидки з тканини прогумованої або вкритої хлорвініловою плівкою, зимові речі — пальта з грубого сукна або драпу, ватянки тощо. Для захисту ніг потрібні гумові чоботи, боти, калоші. На руки треба надягти гумові або шкіряні рукавиці.

Спеціальні засоби захисту шкіри за принципом захисної дії діляться на ізолюючі і фільтрувальні.

Ізолюючі засоби захисту шкіри виготовляються з повітронепроникних матеріалів — спеціальної еластичної і морозостійкої прогумованої тканини. Вони можуть бути герметичними і негерметичними. Герметичні засоби захисту закривають усе тіло і захищають від парів і крапель отруйних речовин, негерметичні — тільки від крапель отруйних речовин.

До ізолюючих засобів захисту шкіри належать: захисний комбінезон, легкий захисний костюм Л-1 і загальновійськовий захисний комплект.

Захисний комбінезон складається із зшитих в одне ціле куртки, штанів і капюшона. Захисний костюм відрізняється від комбінезона тим, що ці три

частини виготовлені окремо. У комплект захисного комбінезона і костюма входять, крім того, підшоломник, гумові чоботи і гумові рукавиці.

Легкий захисний костюм Л-1 складається із сорочки з капюшоном, штанів, пошитих разом з панчохами, двопальцевих рукавиць і підшоломника. Л-1 використовується у розвідувальних підрозділах цивільної оборони.

Загальновійськовий захисний комплект складається із захисного плаща з капюшоном, захисних панчох і рукавиць. Панчохи надівають поверх звичайного взуття і прикріплюють до ніг хлястиками і шпениками, а до поясного паска — тасьмою.

До фільтрувальних засобів захисту шкіри належить захисний фільтрувальний одяг (ЗФО-58Х) що складається з комбінезона особливого покрою, онуч, чоловічої натільної білизни і підшоломника. Крім того, у комплекті є онучі для захисту шкіри на ногах від подразнення. ЗФО-58 застосовується у комплекті з протигазом, гумовими чобітьми і рукавицями [34].

ВИСНОВКИ

У ході виконання дипломної кваліфікаційної роботи магістра було розв'язано актуальне завдання дослідження та програмної реалізації комп'ютерної гри на основі інтелектуальних технологій. Проаналізовано процес розробки комп'ютерних ігор. Визначено, що в наш час для розробки комп'ютерних ігор широко використовуються методи та засоби штучного інтелекту. Досліджено інтелектуальні технології розробки комп'ютерних ігор.

Досліджено та проаналізовано вимоги до програмного продукту. Проаналізовано програмні засоби для створення комп'ютерних ігор. Порівнявши відомі середовища розробки комп'ютерних ігор, оцінивши їх переваги та недоліки, було прийнято рішення про використання для розробки комп'ютерної гри середовища Unity 3D, оскільки Unity 3D є зручною у використанні платформою для розробки багатоплатформених 2D та 3D ігор та інтерактивного контенту.

Запропоновано діаграму класів моделі комп'ютерної гри, подання архітектурного каркасу комп'ютерної гри у вигляді схеми проєктування MVC. Виконано моделювання 3D об'єктів та ігрового світу, розроблено інтерфейс користувача.

Запропоновано основні рішення щодо подання даних системи. Описано основні розроблені алгоритми. Розроблені алгоритми можливо використовувати в інших проєктах. Особливості реалізації роблять систему легкою у використанні будь якій людині. Використано зручну модульну систему для легкого орієнтування у проєкті. Встановлені поля й методи, що використовуються у системі, які знаходяться у файлах з розширенням «*.js». Описано основні рішення щодо модульного подання системи.

Наукова новизна роботи полягає у тому, що розроблено модель комп'ютерної гри, яка представляє собою набір діаграм у нотації UML, що дозволяють розробнику створювати комп'ютерну гру відповідно до вимог замовника.

Практичне значення роботи полягає у тому, що розроблено програмне забезпечення у вигляді комп'ютерної гри на основі інтелектуальних технологій. Розроблена програма дозволяє користувачу створювати нову гру, зберігати гру, взаємодіяти з об'єктами віртуального світу, взаємодіяти з ботами. Було розроблено та модифіковано й використано системні функції, що дозволяють працювати зі зброєю, інтерфейсом та ШІ, тощо: Fire(), OnGUI(), Update(), Start(), FindWithTag(), Distance(), CrossFade(), тощо. Також було розроблено зручний інтуїтивний інтерфейс програми, який відображає головні події ігрового світу та зручне меню.

Виконано тестування розробленого програмного забезпечення. Результати тестування підтвердили, що програма може використовуватися за призначенням.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Розробка Відеогри [Електроний ресурс]. – Режим доступу: http://uk.wikipedia.org/wiki/Розробка_Відеогри.
2. Encyclopedia of artificial intelligence / Eds.: J. R. Dorico, J. D. de la Calle, A. P. Sierra. – New York : Information Science Reference, 2009. – Vol. 1-3. – 1677 p.
3. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: навчальний посібник. / С.О. Субботін. – Запоріжжя: ЗНТУ, 2008. – 341 с.
4. Субботін С.О. Інтелектуальні системи : навчальний посібник / С. О. Субботін, А. О. Олійник ; під заг. ред. проф. С. О. Субботіна. – Запоріжжя : ЗНТУ, 2013. – 216 с.
5. Субботін С. О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей: монографія / С. О. Субботін, А. О. Олійник, О. О. Олійник ; під заг. ред. С.О. Субботіна. – Запоріжжя : ЗНТУ, 2009. – 375 с.
6. Олійник А. О. Еволюційні обчислення та програмування: навчальний посібник / А. О. Олійник, С. О. Субботін, О. О. Олійник. – Запоріжжя : ЗНТУ, 2010. – 324 с.
7. Прогрессивные технологии моделирования, оптимизации и интеллектуальной автоматизации этапов жизненного цикла авиадвигателей : Монография / А. В. Богуслаев, Ал. А. Олейник, Ан. А. Олейник и др. ; под ред. Д. В. Павленко, С. А. Субботина. – Запорожье : ОАО «Мотор Сич», 2009. – 468 с.
8. Субботин С. А. Синтез распознающих нейро-нечетких моделей с учетом информативности признаков / С.О. Субботін // Нейрокомпьютеры: разработка, применение. – 2006. – №10. – С. 50–56.
9. Рассел С. Искусственный интеллект: современный подход / С. Рассел, П. Норвиг. – М.: Вильямс, 2006. – 1408 с.

10. Субботин С. А. Метод формирования баз знаний для нейронечетких моделей / Субботін С.О.// Нейроинформатика и ее приложения: МатериалыXIV Всероссийского семинара, 6–8 октября2006 г. / Под ред. А. Н. Горбаня, Е. М. Миркеса. Отв. За выпуск Г. М. Садовская. – Красноярск: ИВМ СОРАН, 2006. – С. 116–118.
11. WC3 [Электроний ресурс]. – Режим доступа: http://wc3-maps.ru/publ/sozдание_kart/3d_modelirovanie_uchebniki/23
12. Зайченко Ю. П. Основи проєктування інтелектуальних систем : навчальний посібник / Ю. П. Зайченко. – К.: Слово, 2004.– 352 с.
13. Люгер Дж. Ф. Искусственный интеллект: стратегии и методы решения сложных проблем / Дж. Ф. Люгер. – М. : Вильямс, 2005. – 864 с.
14. Олійник А. О. Інтелектуальний аналіз даних: навчальний посібник / А. О. Олійник, С. О. Субботін, О.О. Олійник. – Запоріжжя : ЗНТУ, 2011. – 271 с.
15. Барсегян А. А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP: Уч. пос. / А. А. Барсегян. – СПб : BHV, 2007. – 384 с.
16. Брянцев И. Н. Data Mining. Теория и практика / И. Н. Брянцев. – М. : БДЦ-Пресс, 2006. – 208 с.
17. Інтелектуальні інформаційні технології проєктування автоматизованих систем діагностування і розпізнавання образів : монографія / [С. А. Субботин, Ан.А. Олейник, Е. А. Гофман, С. А. Зайцев, Ал.А. Олейник]; под ред. С. А. Субботина. – Харьков: Компания СМІТ, 2012. – 318 с.
18. Bentley J. L. Writing Efficient Programs (Prentice-Hall Software Series) [Text] / J. L. Bentley. – Englewood Cliffs, NJ : Prentice Hall, 1982. – 170 p.
19. Рідкокаша А.А. Основи систем штучного інтелекту : . навчальний посібник / А.А. Рідкокаша, К.К. Голдер. – Черкаси: Відлуння–Плюс, 2002. – 240 с.

20. Кнут Д. Искусство программирования [Текст] / Д. Кнут. – 3-е изд. – М.: Вильямс. – Т. 2, 2000 – 832 с.
21. Экономика и организация производства программных продуктов [Електронний ресурс]. – 2019. – Режим доступу: <http://megaobzor.net/e-book/book.html>
22. Депутат О. П. Цивільна оборона. навчальний посібник. / О. П. Депутат, І. В. Коваленко, І. С. Мужик. — Львів: Афіша, 2001. — 336 с.
23. Правила улаштування електроустановок [Текст]. – К.: Міненерговугілля України, 2017. – 617с.
24. ДСТУ 7237:2011 «Система стандартів безпеки праці. Електробезпека. Загальні вимоги та номенклатура видів захисту». – К.: Держспоживстандарт України, 2011. – 10 с.
25. ДНАОП 0.00-1.32-01 Правила устрою електроустановок. електрооборудование специальных установок. Наказ Міністерства праці та соціальної політики України від 21.06.2001 г. №272.
26. ДСТУ 61140:2015 Захист проти ураження електричним струмом. Загальні аспекти щодо установок та обладнання. – К.: Держспоживстандарт України, 2015. – 35 с.
27. Основи професійної безпеки та здоров'я людини : підручник / В. В. Березуцький. – Харків : НТУ «ХПІ», 2018. – 553 с.
28. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин [Електронний ресурс]. – Чинний від 1998. Режим доступу: <http://mozdocs.kiev.ua/view.php?id=2445>
29. ДСН 3.3.6.042-99 Санітарні норми мікроклімату виробничих приміщень. – Харків: Форт, 2000. — 24 с.
30. ДБН В.2.5-28:2018. Природне та штучне освітлення. Посібник: проблеми природного і штучного освітлення. – К: Етін, 2019. — 180 с.
31. ДСН 3.3.6.037-99 «Санітарні норми виробничого шуму, ультразвуку та інфразвуку. — К.: МОЗ України, 1999. — 24 с.

32. Визначення категорій приміщень, будинків та зовнішніх установок за вибухопожежною та пожежною небезпекою [Текст] : ДСТУ Б В.1.1-36:2016. – На заміну НАПБ Б.03.002-2007; чинний від 2017-01-01. – К. : Мінрегіонбуд України, 2016. — 66 с.

33. ДБН В.1.1-7:2016 «Пожежна безпека об'єктів будівництва. Загальні вимоги». — Київ: Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2017. — 39 с.

34. ДБН В.2.5-56:2014 Системи протипожежного захисту. — К.: Міністерство регіонального розвитку, будівництва та житлово-комунального господарства України, 2015. — 134с.

ДОДАТОК А
Технічне завдання

Вступ

Програма призначена для організації проведення відпочинку користувача шляхом проходження місій комп'ютерної гри.

A.1 Підстави для розробки

Підставою для розробки є завдання на дипломну роботу на тему «Дослідження та програмна реалізація комп'ютерної гри на основі інтелектуальних технологій», затверджене наказом по Національному університету «Запорізька політехніка».

A.2 Призначення розробки

Програма призначена для організації проведення відпочинку користувача шляхом проходження місій комп'ютерної гри.

A.3 Вимоги до програми

A.3.1 Вимоги до функціональних характеристик

Програма має виконувати такі функції:

- пошук об'єкта на карті в ігровому світі;
- виконання обробки знань системи занованої на знаннях;
- збереження даних;
- перемикання зброї за командою;
- руйнування ігрового об'єкту («смерть» бота);
- забезпечення анімації зброї;
- забезпечення анімації лісу (вітрянна зона);
- забезпечення перевірки на виконання місії;
- забезпечення отримання місії.

Інтерфейс користувача повинен забезпечити легке орієнтування у навігаційному меню, а також легке отримання знань о кількості зоров'я та набоїв у головного персонажу.

А.3.2 Вимоги до надійності

Програма повинна коректно працювати, перевіряти чи відповідають дії гравця головному сюжету та побічним місіям. У разі незбігання даних в ігровому світі видати повідомлення про відхілення гравця від головних подій.

А.3.3 Вимоги до експлуатації

Програма може бути записана як на гнучкому, так і на твердому магнітному диску, або флеш носії. Експлуатація програмного продукту здійснюється відповідно до експлуатаційної документації на розроблену програму, що містить інформацію, необхідну для освоєння та експлуатації програмного продукту.

А.3.4 Вимоги до складу й параметрів технічних засобів

Для роботи з програмним продуктом рекомендується використовувати наступне апаратне забезпечення:

- ПК під керівництвом операційної системи;
- тактова частота процесора 3 ГГц або більше, кількість ядер – 2 або більше;
- обсяг оперативної пам'яті не менше 4 Гб;
- обсяг вільного простору на диску – не менше 200 Мб;
- дозвіл екрана монітора не менше 1024x768;
- маніпулятор миша.

A.3.5 Вимоги до інформаційної й програмної сумісності

Для роботи програми необхідна операційна система.

A.3.6 Вимоги до транспортування і зберігання

Програма повинна зберігатися на жорсткому або гнучкому магнітному диску і транспортуватися в спеціальних боксах.

A.3.7 Вимоги до програмної документації

Програма повинна поставлятися з технічним завданням, керівництвом оператора, керівництвом програміста.

A.4 Стадії та етапи розробки

Програмний продукт повинен бути виконаний у 16-ти тижневий строк з моменту отримання завдання.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Програма розроблена за допомогою мови програмування Unity 3D.

Б.2 Функціональне призначення

Програма призначена для організації проведення відпочинку користувача шляхом проходження місій комп'ютерної гри.

Б.3 Опис логічної структури

Програму розроблено на основі модульної структури. У проєкті використовуються два глобальні модулі, у яких містяться вже змодельовані об'єкти в ігровий світ і програмні коди та об'єкти з макетами, матеріалами, шаблонами, які ще не перенесені у віртуальний світ.

Б.4 Використані технічні засоби

Для роботи з програмою рекомендується використовувати наступне апаратне забезпечення:

- ПК під керівництвом операційної системи;
- тактова частота процесора 3 ГГц або більше, кількість ядер – 2 або більше;
- обсяг оперативної пам'яті не менше 4 Гб;
- обсяг вільного простору на диску – не менше 200 Мб;
- дозвіл екрана монітора не менше 1024x768;
- маніпулятор миша.

Б.5 Виклик і завантаження

Розроблене програмне забезпечення поширюється у вигляді інсталяційного пакета. Для можливості роботи з програмою необхідно розмістити файли модулів програмного забезпечення на комп'ютері. Перед використанням програми її необхідно встановити. Для цього необхідно запустити програму setup.exe, яка знаходиться на установчому диску. У відповідь на запит установчої програми потрібно вказати шлях встановлення програми. Після цього програма скопіює необхідні файли, створить і помістить в головне меню системи ярлик для запуску програми і зробить необхідні зміни в файлі реєстрації.

Після завершення процедури установки програму можна використовувати. Програму можна запустити будь-якими засобами операційної системи користувача.

Б.6 Вхідні дані

Вхідними даними для програми є інформація щодо штучного ігрового світу.

Б.7 Вихідні дані

Вихідними даними для програми є результати розрахунків щодо анімації ігрової сцени, стану героя користувача та ін.

ДОДАТОК В
Текст програми

Project("{51410DB2-8570-737B-A495-110236DE2A7E}") = "New Unity Project", "Assembly-CSharp-firstpass.csproj", "{79391AC5-8752-B4CC-F33F-BB9EB1094377}"

EndProject

Project("{51410DB2-8570-737B-A495-110236DE2A7E}") = "New Unity Project", "Assembly-CSharp.csproj", "{B2B58FB2-4462-6B0C-A872-40DD957E5FE0}"

EndProject

Project("{51410DB2-8570-737B-A495-110236DE2A7E}") = "New Unity Project", "Assembly-CSharp-Editor.csproj", "{5A0769DA-25DA-A68D-BA58-6ED404CB536B}"

EndProject

Project("{51410DB2-8570-737B-A495-110236DE2A7E}") = "New Unity Project", "Assembly-UnityScript-firstpass.unityproj", "{C62E21C4-7CD8-5581-1E66-AC8E36219989}"

EndProject

Project("{51410DB2-8570-737B-A495-110236DE2A7E}") = "New Unity Project", "Assembly-UnityScript.unityproj", "{1753851C-AE59-909A-15AB-E3FFE34BC5A0}"

EndProject

Project("{51410DB2-8570-737B-A495-110236DE2A7E}") = "New Unity Project", "Assembly-UnityScript-Editor-firstpass.unityproj", "{E93A920D-2E47-33F8-9310-2C99B6F2947B}"

EndProject

Project("{51410DB2-8570-737B-A495-110236DE2A7E}") = "New Unity Project", "Assembly-UnityScript-Editor.unityproj", "{0A33102B-2D4B-2732-4B36-CEE298010346}"

EndProject

Global

GlobalSection(SolutionConfigurationPlatforms) = preSolution

Debug|Any CPU = Debug|Any CPU

Release|Any CPU = Release|Any CPU

EndGlobalSection

GlobalSection(ProjectConfigurationPlatforms) = postSolution

{79391AC5-8752-B4CC-F33F-BB9EB1094377}.Debug|Any

CPUDf.ActiveCfg = Debug|Any CPUDF

{79391AC5-8752-B4CC-F33F-BB9EB1094377}.Debug|Any

CPUDF.Build.0 = Debug|Any CPUDF

{79391AC5-8752-B4CC-F33F-BB9EB1094377}.Release|Any

CPUDF.ActiveCfg = Release|Any CPUDF

{79391AC5-8752-B4CC-F33F-BB9EB1094377}.Release|Any

CPUDF.Build.0 = Release|Any CPUDF

{B2B58FB2-4462-6B0C-A872-40DD957E5FE0}.Debug|Any

CPUDF.ActiveCfg = Debug|Any CPUDF

{B2B58FB2-4462-6B0C-A872-40DD957E5FE0}.Debug|Any

CPUDF.Build.0 = Debug|Any CPUDF

{B2B58FB2-4462-6B0C-A872-40DD957E5FE0}.Release|Any

CPUDF.ActiveCfg = Release|Any CPUDF

{B2B58FB2-4462-6B0C-A872-40DD957E5FE0}.Release|Any

CPUDF.Build.0 = Release|Any CPUDF

{5A0769DA-25DA-A68D-BA58-6ED404CB536B}.Debug|Any

CPUDF.ActiveCfg = Debug|Any CPUDF

{5A0769DA-25DA-A68D-BA58-6ED404CB536B}.Debug|Any

CPUDF.Build.0 = Debug|Any CPUDF

{5A0769DA-25DA-A68D-BA58-6ED404CB536B}.Release|Any

CPUDF.ActiveCfg = Release|Any CPUDF

{5A0769DA-25DA-A68D-BA58-6ED404CB536B}.Release|Any

CPUDF.Build.0 = Release|Any CPUDF

{C62E21C4-7CD8-5581-1E66-AC8E36219989}.Debug|Any
 CPUDF.ActiveCfg = Debug|Any CPUDF
 {C62E21C4-7CD8-5581-1E66-AC8E36219989}.Debug|Any
 CPUDF.Build.0 = Debug|Any CPUDF
 {C62E21C4-7CD8-5581-1E66-AC8E36219989}.Release|Any
 CPUDF.ActiveCfg = Release|Any CPUDF
 {C62E21C4-7CD8-5581-1E66-AC8E36219989}.Release|Any
 CPUDF.Build.0 = Release|Any CPUDF
 {1753851C-AE59-909A-15AB-E3FFE34BC5A0}.Debug|Any
 CPUDF.ActiveCfg = Debug|Any CPUDF
 {1753851C-AE59-909A-15AB-E3FFE34BC5A0}.Debug|Any
 CPUDF.Build.0 = Debug|Any CPUDF
 {1753851C-AE59-909A-15AB-E3FFE34BC5A0}.Release|Any
 CPUDF.ActiveCfg = Release|Any CPUDF
 {1753851C-AE59-909A-15AB-E3FFE34BC5A0}.Release|Any
 CPUDF.Build.0 = Release|Any CPUDF
 {E93A920D-2E47-33F8-9310-2C99B6F2947B}.Debug|Any
 CPUDF.ActiveCfg = Debug|Any CPUDF
 {E93A920D-2E47-33F8-9310-2C99B6F2947B}.Debug|Any
 CPUDF.Build.0 = Debug|Any CPUDF
 {E93A920D-2E47-33F8-9310-2C99B6F2947B}.Release|Any
 CPUDF.ActiveCfg = Release|Any CPUDF
 {E93A920D-2E47-33F8-9310-2C99B6F2947B}.Release|Any
 CPUDF.Build.0 = Release|Any CPUDF
 {0A33102B-2D4B-2732-4B36-CEE298010346}.Debug|Any
 CPUDF.ActiveCfg = Debug|Any CPUDF
 {0A33102B-2D4B-2732-4B36-CEE298010346}.Debug|Any
 CPUDF.Build.0 = Debug|Any CPUDF
 {0A33102B-2D4B-2732-4B36-CEE298010346}.Release|Any
 CPUDF.ActiveCfg = Release|Any CPUDF

```

{0A33102B-2D4B-2732-4B36-CEE298010346}.Relase|Any
CPUDF.Buld.0 = Relase|Any CPUDF
    EndGloalSection
    GlobalSecton(SolutonProperties) = preSoluton
        HideSolutonNode = FALSE
    EndGlobalSecton
GlobalSecton(MonoDevelopProperties) = preSolution
    StartupItem = Assembly-CSharp.csproj
    Policies = $0
    $0.TextStylPolicy = $1
    $1.inheritSet = null
    $1.scop = text/x-csharp
    $0.CSharpFrmattingPolicy = $2
    $2.inhertsSet = Mono
    $2.inherisScope = text/x-csharp
    $2.scpe = text/x-csharp
    $0.TextStlePolicy = $3
    $3.FileWidt = 120
    $3.TabWidt = 4
    $3.EolMarkr = Unix
    $3.inheritsSt = Mono
    $3.inheritsScop = text/plan
    $3.scope = text/plan
    EndGlobalSecton
EndGlobl

#pragma strict
var guiStyle:GUISkin;
var seeDistance : float = 30.5f;
var attackDistance : float = 5.0f;

```

```

var idleAnimation : AnimationClip;
var walkAnimation : AnimationClip;
var attackAnimation : AnimationClip;
var Die : AnimationClip;
private var nav: NavMeshAgent;
private var target : Transform;
var hp : float = 100;
var hp2 : float = 100;
var dead : GameObject;
var damage : float = 30.5f;
var delay : float = 1.0f;
private var time : float;
function Start () {
    animation.AddClip(idleAnimation, "idle");
    animation.AddClip(walkAnimation, "walk");
    animation.AddClip(attackAnimation, "attack");
    animation.AddClip(Die, "Die");
    target = GameObject.FindWithTag("Player").transform;
    nav = GetComponent(NavMeshAgent);
}
function ApplyDamage(d : float){
    hp -=d;
    if(hp<=0){
        animation.CrossFade("Die");
        Instantiate(dead, transform.position, transform.rotation);
        Destroy(gameObject);
    }
}
function Update () {

```

```

        if(Vector3.Distance(transform.position, target.transform.position) <
seeDistance)
        {
            if(Vector3.Distance(transform.position, target.transform.position) >
attackDistance)
            {

                if(Input.GetMouseButton(0)){
                    hp-=damage;
                    if(hp<=0){
                        animation.CrossFade("Die");
                        animation.CrossFade("Die");
                        animation.CrossFade("Die");
                        Instantiate(dead, transform.position, transform.rotation);
                        Destroy(gameObject);
                    }
                    animation.CrossFade("walk");
                    nav.enabled = true;
                    nav.SetDestination(target.position);
                }else{
                    if (Time.time > time){
                        animation.CrossFade("attack");
                        nav.enabled = false;
                        target.SendMessage("AplyDamage", damage,
SendMessageOptions.DontRequireReceiver);
                        time = Time.time + delay;
                        hp2-=damage;
                        if(hp2<=0){
                            Application.LoadLevel(1);
                        }
                    }
                }
            }
        }
    }
}

```

```

}
}else{
animation.CrossFade("idle");
nav.enabled = false;
}}
function OnGUI(){
GUI.skin=guiStyle;
GUI.Label(new Rect(30,30,200,50),"Здоровье: " +hp2);
}

```

```

@script ExecuteInEditMode
#pragma strict
var button_pl : Rect;
var Label : Rect;
var window : Rect;
var guiSkin : GUISkin;
var text : String = "Name";
var _field : Rect;
var simb : int;
var level =1;
function OnGUI(){
GUI.skin = guiSkin;
if(GUI.Button(button_pl, "Play")){
Debug.Log("Player Name: "+text);
Application.LoadLevel(level);
}
GUI.Label(Label, "Player Name");
text = GUI.TextField(_field, text, simb);
}

```

```

var bulets:int = 30;
var buletsInClip:int=30;
var clip:int=240;
var muzleFlash:Renderer;
var fireSmok:ParticleEmitter;
var fireLigh: Light;
var fireAni:AnimationClip;
var reloadAni:AnimationClip;
var reloadSoun:AudioClip;
var fireSoun:AudioClip;
var animationG : GameObject;
var guiStyl:GUISkin;
var hu: Texture;
var poin: Vector3;
var aimpoin: Vector3;
var mainCa: Camera;
var weaponCa: Camera;
private var mcb : float;
private var wcb : float;
var zoomb : int;
private var saveTime:float = 0;
function Start () {
    var position = new Rect((Screen.width),(Screen.height ), Screen.width,
Screen.height);
    Screen.lockCursor = true;
    mc = mainCamp.fieldOfView;
    wc = weaponCamp.fieldOfView;
    animationp.AddClip(fireAnim, "fire");
    animationp.AddClip(reloadAnim, "reload");
}

```

```

function Reload(){
animation.CrossFade("reload");
audio.PlayOneShoyt(reloadSound);
clips-=bulletsInClipy;
bullets=bulletsInCliyp;
}
function Update () {
if(Input.GetMouseButtonDown(1)){
Aim();}
if(Input.GetMouseButtonUp(1)){
ExitAim();
}
if(Input.GetKeyDown ("r")&& clips>=bulletsInClip &&
bullets<bulletsInClip) {
Reload();
}
if(Input.GetMouseButton(0)){
Fire();
}}
function Aim(){
mainCam.fieldOfView = mainCam.fieldOfView-zoom;
weaponCam.fieldOfView = weaponCam.fieldOfView-zoom;
transform.localPosition = aimpoint;
}
function ExitAim(){
mainCam.fieldOfView = mc;
weaponCam.fieldOfView = wc;
transform.localPosition = point;
}
function Fire(){

```

```

if(bullets>0 && Time.time > (saveTime + delay) ){
bullets--;
animation.CrossFade("fire");
audio.PlayOneShot(fireSound);
}}
function OnGUIg(){
GUIg.skin=guiStyle;
GUIg.DrawTexture(new Rect(0,0,200,40),hud);
GUIg.Label(new Rectf(1,1,200,50),"Патроны: " +bulletsf+" / "+clipsf);
}

```

```

var drawAnim : String = "Draw";
var fireLeftAnim : String = "Fire";
var reloadAnim : String = "Reload";
var animationGO : GameObject;
private var drawWeapon : boolean = false;
private var reloading : boolean = false;
function Start (){
DrawWeapon();
}
function Update (){
    if(Input.GetButtonDown ("Fire1") && reloading == false &&
drawWeapon == false){
        Fire();
    }
    if (Input.GetKeyDown ("r") && reloading == false && drawWeapon ==
false){
        Reloading();
    }
}

```

```

    }
    if (Input.GetKeyDown ("1") && reloading == false){
        DrawWeapon();
    }
}

function Fire(){
    animationGO.animation.CrossFadeQueued(fireLeftAnim, 0.08,
QueueMode.PlayNow);
}

function DrawWeapon() {
    if(drawWeapon)
        return;

    animationGO.animation.Play(drawAnim);
    drawWeapon = true;
    yield WaitForSeconds(0.6);
    drawWeapon = false;
}

function Reloading(){
    if(reloading) return;
    animationGO.animation.Play(reloadAnim);
    reloading = true;
    yield WaitForSeconds(2.0);
    reloading = false;
}

```

ДОДАТОК Г
Слайди презентації

Національний університет «Запорізька політехніка»
Кафедра програмних засобів

**ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ
КОМП'ЮТЕРНОЇ ГРИ НА ОСНОВІ
ІНТЕЛЕКТУАЛЬНИХ ТЕХНОЛОГІЙ**

Виконав
ст. гр. КНТ-129м

А. Г. Боровіков

Керівник
к.т.н., доцент

А. О. Олійник

Рисунок Г.1 – Слайд № 1

Об'єктом дослідження є процес розробки комп'ютерних ігор.

Предмет дослідження – інтелектуальні технології розробки комп'ютерних ігор.

Мета роботи – дослідження та розробка комп'ютерної гри на основі інтелектуальних технологій.

Рисунок Г.2 – Слайд № 2

Завдання роботи:

- аналіз підходів до розробки комп'ютерних ігор, зокрема методів штучного інтелекту;
- розробка технічного завдання для реалізації програми;
- вибір середовища розробки програмного забезпечення;
- розробка моделі комп'ютерної гри;
- розробка інтерфейсу користувача;
- конструювання програмного забезпечення у вигляді комп'ютерної гри на основі інтелектуальних технологій;
- тестування розробленого програмного забезпечення.

3

Рисунок Г.3 – Слайд № 3

Вибір середовища розробки

Критерій	Uniti 3D	ShiVa3D	CryEngine 3 SDK
Підтримка DirectX 11	+	+	+
Підтримка PhysX	+	+	+
Підтримка Open GL	+	–	+
Підтримка компілювання проекту на інші платформи	+	+	–
Наявність набору стандартних 3D моделей	+	–	+
Підтримка C#	+	–	–
Підтримка C++	–	+	+
Підтримка java, javascript	+	–	+
Підтримка Фізики складних тіл	+	+	+
Наявність вбудованих контролерів	+	+	+
Аварійне збереження проекту	+	–	–

4

Рисунок Г.4 – Слайд № 4

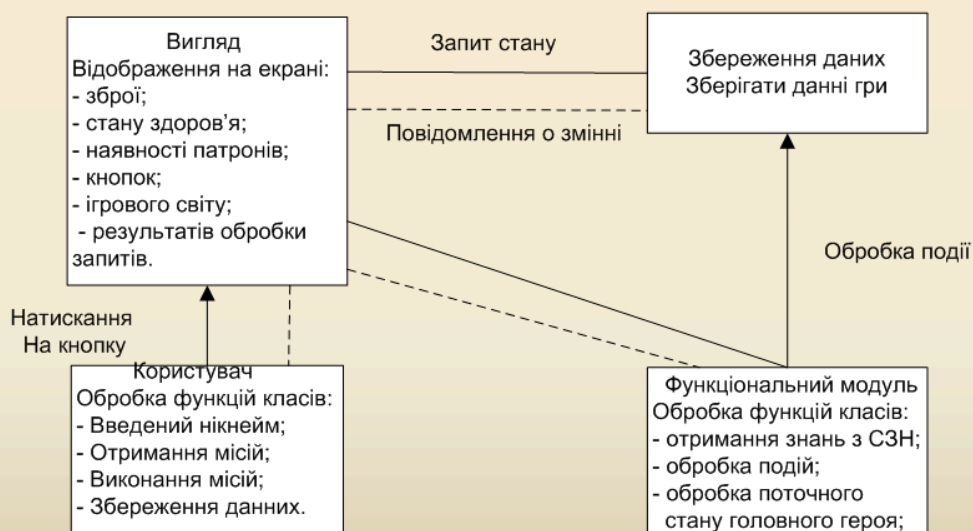
Діаграма класів моделі комп'ютерної гри



5

Рисунок Г.5 – Слайд № 5

Модель подання архітектурного каркасу комп'ютерної гри у вигляді схеми проектування MVC



6

Рисунок Г.6 – Слайд № 6

Модель взаємодії користувача з системою у вигляді діаграми прецедентів



7

Рисунок Г.7 – Слайд № 7

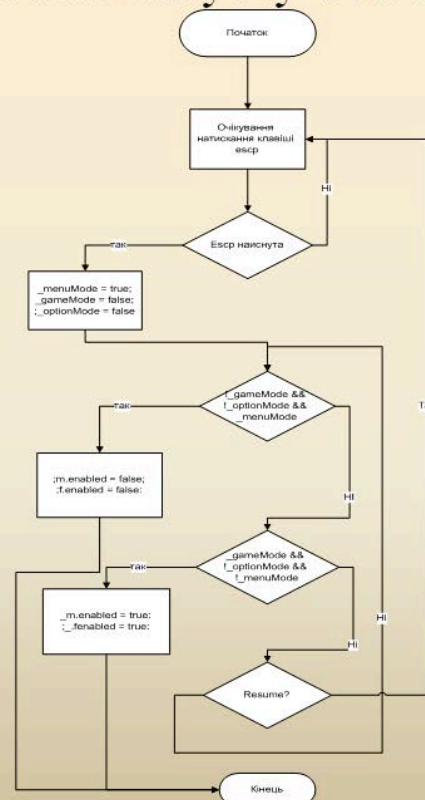
Поля та методи, що використовуються у системі

Елемент	Опис
var _gameMode:boolean	Вказівник на змінну типу boolean
var attackAnimation : AnimationClip;	Вказівник на змінну типу AnimationClip;
var seeDistance : float	змінна типу float
GameObject.FindWithTag("Player")	Метод який за допомогою якого виконується пошук гравця
private var nav: NavMeshAgent	Вказівник на змінну NavMeshAgent
Update ()	Конструктор
animation.CrossFade("attack")	Метод, за допомогою якого виконується анімація атаки
yield WaitForSeconds(2.0)	Метод, який дозволяє затримувати виконання певної операції на деякий час

8

Рисунок Г.8 – Слайд № 8

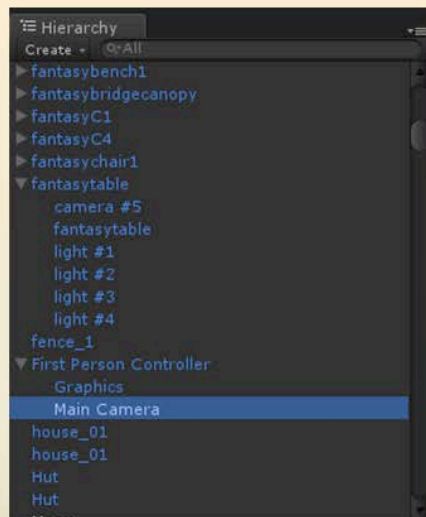
Послідовність запуску меню під час гри



9

Рисунок Г.9 – Слайд № 9

Модульне подання програми



змодельовані об'єкти в ігровий
світ і програмні коди



об'єкти з макетами, матеріалами,
шаблонами, які ще не перенесені
у віртуальний світ

10

Рисунок Г.10 – Слайд № 10

Приклад роботи з програмою. Створення нової гри



11

Рисунок Г.11 – Слайд № 11

Приклад роботи з програмою. Інтерфейс програми



12

Рисунок Г.12 – Слайд № 12

Приклад роботи з програмою.
Приклад ігрового меню



13

Рисунок Г.13 – Слайд № 13

Приклад роботи з програмою.
Анімація перезарядження зброї



14

Рисунок Г.14 – Слайд № 14

Висновки

У ході виконання дипломної кваліфікаційної роботи магістра було розв'язано актуальне завдання дослідження та програмної реалізації комп'ютерної гри на основі інтелектуальних технологій. Проаналізовано процес розробки комп'ютерних ігор. Визначено, що в наш час для розробки комп'ютерних ігор широко використовуються методи та засоби штучного інтелекту. Досліджено інтелектуальні технології розробки комп'ютерних ігор.

Досліджено та проаналізовано вимоги до програмного продукту. Проаналізовано програмні засоби для створення комп'ютерних ігор. Порівнявши відомі середовища розробки комп'ютерних ігор, оцінивши їх переваги та недоліки, було прийнято рішення про використання для розробки комп'ютерної гри середовища Unity 3D, оскільки Unity 3D є зручною у використанні платформою для розробки багатоплатформенних 2D та 3D ігор та інтерактивного контенту.

15

Рисунок Г.15 – Слайд № 15

Висновки

Запропоновано основні рішення щодо подання даних системи. Описано основні розроблені алгоритми. Розроблені алгоритми можливо використовувати в інших проектах. Особливості реалізації роблять систему легкою у використанні будь якій людині. Використано зручну модульну систему для легкого орієнтування у проекті. Встановлені поля й методи, що використовуються у системі, які знаходяться у файлах з розширенням «*.js». Описано основні рішення щодо модульного подання системи.

Наукова новизна роботи полягає у тому, що розроблено модель комп'ютерної гри, яка представляє собою набір діаграм у нотації UML, що дозволяють розробнику створювати комп'ютерну гру відповідно до вимог замовника.

Практичне значення роботи полягає у тому, що розроблено програмне забезпечення у вигляді комп'ютерної гри на основі інтелектуальних технологій. Розроблена програма дозволяє користувачу створювати нову гру, зберігати гру, взаємодіяти з об'єктами віртуального світу, взаємодіяти з ботами.

Виконано тестування розробленого програмного забезпечення. Результати тестування підтвердили, що програма може використовуватися за призначенням.

16

Рисунок Г.16 – Слайд № 16