

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ  
УНІВЕРСИТЕТ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-05

# Дипломний проект

здобувача освіти денної форми навчання

РП.05.04.000.ДП

***ГОВЕРИ ОЛЕКСІЯ  
ОЛЕКСІЙОВИЧА***

м. Одеса  
2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ВСП «ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ ОНТУ»

Спеціальність: **121 «Інженерія програмного забезпечення»**

Освітня програма: **«Розробка програмного забезпечення»**

Група: **4РП-05**

## ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Розробка ігрового додатку класичного жанру

Проектний матеріал складається з пояснювальної записки на \_\_\_\_\_ сторінках та графічного (презентаційного) матеріалу на \_\_\_\_\_ аркушах (слайдах).

Дипломник \_\_\_\_\_ (Говера О.О.)

Керівник \_\_\_\_\_ (Іванова Л.В.)

### Консультанти:

з економічної частини \_\_\_\_\_ (Копайгородська Т.Г.)

з охорони праці \_\_\_\_\_ (Чорновол Н.І.)

з дотримання вимог ЄСКД \_\_\_\_\_ (Петрашова В.І.)

старший консультант \_\_\_\_\_ (Скорнякова О.В.)

### До захисту допущений

Голова циклової комісії \_\_\_\_\_ (Скорнякова О.В.)

Завідувач відділення \_\_\_\_\_ (Суліма Ю.Ю.)

Захист « \_\_\_\_ » \_\_\_\_\_ 2022 р.      Протокол ЕК № \_\_\_\_\_

Оцінка ЕК \_\_\_\_\_

Секретар ЕК \_\_\_\_\_

## **Анотація**

Тема дипломного проекту «Розробка ігрового додатку класичного жанру».

У цій роботі описані принципи розробки сучасної комп'ютерної гри за жанром horror. У ігровому додатку застосовується цікаві механіки гри.

У аналітичній частині приведені технічне завдання до проекту, системні характеристика комп'ютера щоб грати з комфортом. В технологічній частині приведені інструменти та детальний опис програмного коду та описання праці ігрових об'єктів ігрового додатку. У програмному розділі описані ігрові двигуни їх різновиди, також описані популярні жанри ігрових додатків. Останніми розділами стали охорона праці під час розробки та конструювання додатку та економічне обґрунтування.

Ключові слова: ігровий двигун, код, розробка, C#, Unity3D, ігровий об'єкт, Скрипт, UI.

## **Annotation**

The theme of the graduation project is "Development of the game addendum to the classic genre."

This robot describes the principles of development of modern computer grime behind the horror genre. At the game addendum using interesting mechanics gri.

At the analytical part, the technical specifications for the project are given, the system characteristics of the computer to increase comfort. In the technological part, a detailed description of the program code and the description of the practice of the game objects of the game appendix is given. The software section has descriptions of game engines and their differences, as well as descriptions of popular genres of game add-ons.

The remaining divisions were the protection of practical work and the hour of construction, the construction of an addendum, and the economical priming.

Keywords: game engine, code, development, C#, Unity3D, game object, script, UI.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ВСП «ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та III

Спеціальність 121 «Інженерія програмного забезпечення»

Освітня програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР Беркань І.В.

“ \_\_\_\_\_ ” \_\_\_\_\_ 2022 р.

**ЗАВДАННЯ**

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти Говера Олексій Олексійович

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка ігрового додатку класичного жанру

затверджена наказом по коледжу від “ \_\_\_\_\_ ” \_\_\_\_\_ 202 \_\_\_\_\_ р. № \_\_\_\_\_

2. Термін здачі закінченого проекту (роботи) 20.06.2022р.

3. Вихідні данні до проекту (роботи) \_\_\_\_\_

1. *Концепт документ гри, ігрове середовище;*

2. *Аналіз програмного забезпечення; ігрові двигуни*

3. *Вимоги до функціоналу, вимоги до результативності;*

4. *Засоби тестування ігрових проектів*

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

*Вступ. Аналітичний огляд ігрової індустрії. Розробка концепт-документу для ігрового проекту. Розробка та реалізація ігрового додатку в Unity . Тестування та публікація ігрового додатку. Економічні розрахунки . Охорона праці. Висновок. Список використаних джерел інформації.*

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)

1. *Структурна схема ігрового середовища*

2. *Алгоритм роботи ігрового додатку*

3. *Схема взаємодії ігрового контенту*

4. *Графічний дизайн ігрового проекту*

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний	Іванова Л.В.	30.11.2021	17.05.2022
Економіка	Копайгородська Т.Г.	4.05.2022	17.05.2022
Охорона праці	Чорновол В.І.	4.05.2022	17.05.2022
Нормоконтроль	Петрашова В.І.	4.05.2022	17.05.2022
Старший консультант	Скорнякова О.В.	4.05.2022	17.05.2022

7. Дата видачі завдання \_\_\_\_\_ 30.11.2021 \_\_\_\_\_

Керівник

Іванова Л.В.

(підпис)

Завдання прийняв до виконання

Говера О.О.

(підпис)

#### КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1.	Передмова	4.05.2022	
2.	Аналітичний огляд ігрової індустрії	8.05.2022	
3.	Розробка концепт-документу для ігрового проекту	10.05.2022	
4.	Розробка та реалізація ігрового додатку в Unity	15.05.2022	
5.	Тестування та публікація ігрового додатку	17.05.2022	
6.	Економічні розрахунки	22.05.2022	
7.	Розділ охорони праці	26.05.2022	
8.	Висновок	28.05.2022	
9.	Перелік літератури	31.05.2022	
10.	Оформлення пояснювальної записки	2.06.2022	
11.	Оформлення графічної частини	10.06.2022	
12.	Малий захист кваліфікаційної роботи	15.06.2022	
13.	Захист кваліфікаційної роботи	18.06.2022	

Дипломник

Говера О.О.

(підпис)

Керівник

Іванова Л.В.

(підпис)



## ЗМІСТ

СТОР

<b>ПЕРЕДМОВА</b>	<b>6</b>
<b>1. Аналітичний огляд ігрової індустрії</b>	<b>10</b>
<b>1.1. Класифікація ігрових жанрів</b>	<b>11</b>
<b>1.2. Вибір ігрового двигуна</b>	<b>15</b>
<b>1.3. Етапи проектування ігрових додатків</b>	<b>20</b>
<b>2. Розробка концепт-документу для ігрового проекту (тех. завдання)</b>	<b>25</b>
<b>3. Розробка та реалізація ігрового додатку в Unity</b>	<b>27</b>
<b>3.1 Створення проекту та його налаштування</b>	<b>27</b>
<b>3.2 Інтерфейс розробника</b>	<b>27</b>
<b>3.3 Створення ігрового миру</b>	<b>29</b>
<b>3.4 Програмування камери</b>	<b>30</b>
<b>3.5 Додавання моделей до ігрового світу</b>	<b>32</b>
<b>3.6 Накладання текстур на тривимірну модель</b>	<b>34</b>
<b>3.7 Програмування ігрових об'єктів</b>	<b>36</b>
<b>3.8 Створення ігрового меню та переходу до ігрового світу</b>	<b>41</b>
<b>4. Тестування та публікація ігрового додатку</b>	<b>43</b>
<b>5. Економічні розрахунки</b>	<b>45</b>
<b>6. Охорона праці</b>	<b>51</b>
<b>Висновок</b>	<b>54</b>
<b>Список використаних джерел інформації</b>	<b>55</b>

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

## *ПЕРЕДМОВА*

Комп'ютерні ігри – це складні програми, які служать для навчання чи розваги і розраховані різні вікові групи. Є два види комп'ютерних ігор — навчальні та для розваг.

Розважальні або екшн-ігри — це шутери (від англ. shoot — стріляти), платформери (від англ. platform game — гра, де персонаж перестрибує на різні платформи) та файтинги (від англ. fight — битися, боротися). Такі ігри вимагають швидкості реакції та координації рухів рук і очей. Тому безліч дослідників намагаються зрозуміти, чим такі ігри можуть бути корисні.

Люди, що захоплюються екшн-іграми, демонструють високу зорову увагу та просторову орієнтацію. Про це у своїй праці «Exercising your brain» пишуть професори Женевського університету С. Грін та Д. Бавальє. Також такі ігри розвивають уміння перемикатись між завданнями, виявляти дрібні предмети та фокусуватись тривалий час на чомусь одному. У досліджах на здатність концентруватися на різних чинниках діти, що перед цим грали в динамічний шутер, показали кращий результат, ніж діти, що не грали.

У відеоіграх поступово збільшується навантаження — швидкість або складність. Таким чином навчання стає індивідуальним і кожен грає на межі своїх можливостей. Також екшн-ігри потребують постійної адаптації, оскільки у вищих рівнях виникають нові середовища, завдання та вимоги. Це формує звичку «вчитися навчатись», адже гравцю доводиться самостійно пристосовуватись до нового. Помилковою є думка, що відеоігри роблять дітей більш роздратованими та агресивними. Навпаки, вони можуть відволікати від болісних спогадів та негативних емоцій, а також знижувати тривожність. Ще одним позитивним ефектом екшн-ігор є навичка планування. Для того щоб успішно грати, потрібно навчитись прогнозувати наслідки своїх дій та стратегічно мислити. А якщо гра динамічна, то рішення доводиться приймати досить швидко. Усі ці вміння працюють не лише у віртуальному світі, а й переносяться в реальне життя.

Але захоплення комп'ютерними іграми має і негативний бік - підлітки переважно не бажають відволікатися від комп'ютерних ігор заради інших видів діяльності та спілкування, або ж прагнуть відразу ж до них повернутися, не

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

бажаючи прислухатися до думки педагога. Гра починає здаватися їм найцікавішим заняттям, вони намагаються максимально замінити нею реальність, багато в чому на шкоду міжособистісному спілкуванню. Подібна ситуація може викликати негативні наслідки у процесі соціального розвитку підлітку та потребує спостереження і корекції соціальних педагогів, батьків, педагогів, психологів.

Метою дипломного проекту є розробка ігрового додатку, який відноситься до таких жанрів як Інді-ігри. До цього жанру відносяться відеоігри, створені незалежно від фінансової підтримки великих видавців. Також у ігровому додатку буде присутній жанр Adventure, який передбачає вирішення поставлених завдань шляхом їх обдумування, уважного пошуку підказок і схованих деталей. Такі ігри можуть як мати одного чи кількох головних персонажів, так і не мати їх взагалі. Також у грі будуть присутні елементи Psychological horror – це психологічні прийоми, покликані налякати гравця емоційними, психічними чи психологічними станами, а не монстрами чи страхами.

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

## **РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД ІГРОВОЇ ІНДУСТРІЇ**

Індустрія відеоігор або індустрія інтерактивних розваг — економічний сектор, пов'язаний з розробкою, просуванням та продажем відеоігор. У неї входить велика кількість спеціальностей, за якими працюють тисячі людей по всьому світу.

Сьогодні індустрія комп'ютерних ігор є сектор глобальної економіки, що швидко розвивається, пов'язаний із розробкою, просуванням та продажем ігор. А самі ігри тепер позиціонуються та сприймаються користувачами як якісне багатожанрове інтерактивна розвага, яка поступово стає все популярнішою.

На думку експертів, саме ігрова індустрія стає провідником ідей сучасного творчості та новаторства. Комп'ютерні ігри зараз присутні у різних сегментах масової культури, а деякі навіть стали її феноменами (наприклад, Mario чи Final Fantasy).

У деяких країнах, наприклад, у США, комп'ютерні ігри були визнані окремим видом мистецтва. Крім того, було започатковано нову субкультуру зі своєю мовою, іконами, унікальними подіями та заходами. Формується сильна екосистема. Геймінг стає високооплачуваною професією. Згідно з експертними оцінками, понад 35% усіх геймерів у світі готові зробити геймінг своїм основним родом занять.

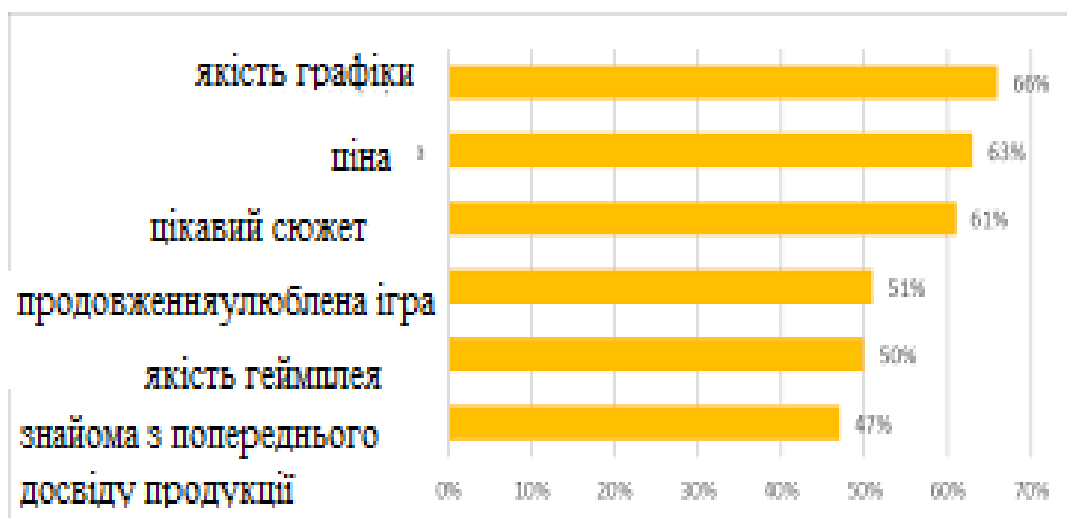
З'являються не лише спеціальні журнали, сайти та ТВ-програми, присвячені іграм, а й в даний час включаються в розважальні та бізнес-видання (наприклад, англійська SUN або американський WallStreetJournal). За сюжетом комп'ютерних ігор давно знімаються анімаційні та кінострічки (наприклад, "Смертельна битва" (Mortal Combat, 1995), "Лара Крофт: Розкрадача гробниць» (Lara Croft: Tomb Raider, 2001) та ін.). А в основі сюжетів самих ігор лежать політичні, економічні та соціальні події (наприклад, на основі подій Другої світової війни – Codename Panzers, Phase Two (2005) або Blitzkrieg 3 (2017)).

Про зростаючу роль ігор у житті сучасного суспільства також свідчить активне проникнення ігрових практик у неігрові процеси – гейміфікація (gamification) – до ряду традиційних галузей (наприклад, ритейл, реклама, банківський сектор, освіта, цивільні авіаперевезення та ін.) з метою залучення, утримання та навчання цільової аудиторії.

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

В даний час активно розвиваються розраховані на багато ігрові проекти, що функціонують одночасно як і соціальні простори, де учасники можуть грати та спілкуватися під час гри (наприклад, The Crew, Minecraft). Експерти вважають, що наступні проекти подібного роду будуть використовувати ідею розрахованих на багато користувачів ігор вже скоріше як соціальний, а не просто ігровий досвід. Крім того, спостерігається та інтеграція комп'ютерних ігор із популярними соцмережами (Facebook, Twitter та ін.). Видавці вбудовують у геймплей комп'ютерних ігор та в ігрові консолі шерінгові опції, що дозволяють ділитися враженнями – робити скріншоти, записувати відеоролики та розміщувати їх у своїх соцакаунтах і при цьому не залишати гру.

Найпомітнішим трендом розвитку ігрової індустрії є підвищення якості графіки комп'ютерних ігор. Але тут є один нюанс: у короткостроковому періоді прориви не такі помітні, але якщо розширити часові рамки, то прогрес очевидний. Більше того, сьогодні саме якість графіки є основним фактором, провокуючим користувачів на покупку гри.



**Рис.1.1 – Фактори, які впливають на вибір користувачами комп'ютерних ігор**

На думку галузевих експертів, у найближчі 10–20 років графіка у комп'ютерних іграх стане невідмінною від кіно, що у поєднанні з пристроями віртуальної та доповненої реальності забезпечить максимально можливий ефект занурення.

## 1.1 Класифікація ігрових жанрів

Щоб класифікація комп'ютерних ігор за жанрами не здавалася надто складною, варто визначити три класи, до яких можна віднести більшу частину ігрових програм:

**Динамічні ігри** Від геймера потрібно максимальна швидкість реакції та точність. Мінімум інтелектуальних завдань.

**Ігри планування.** У них головне – розвиток та оцінка ситуації. У цьому думати доводиться як про справжній стан справ, а й у тому, що може статися наступних ходах і які переваги можна отримати у майбутньому. Найближча і найнаочніша паралель - шахи.

**Сюжетні ігри** Вони можуть бути й елементи двох вищеописаних класів, проте метою є просування по сюжету, а чи не перемога над противником.

**Аркади** - один із найстаріших жанрів. Головна їхня ознака - найпростіше управління. Наприклад, геймеру не потрібно нічого знати про те, як керувати машиною в реальному житті. Досить натиснути кнопку зі стрілкою, щоб повернути.

Однак це зовсім не означає, що в аркаді легко виграти. Багато розробників слідує золотому правилу: легко освоїти, важко перемогти.

В основі сучасних розподілів відеоігор на жанри лежить вид активності, який найчастіше здійснює гравець в іграх даного жанру. Так відеоігри в загальному можуть поділятися на ігри руху, планування і сюжету або спілкування, дії та контролю. В багатьох класифікаціях визначення жанру відбувається за кількома осями. Наприклад, за двома осями сюжет — свобода дії, або трьома абстракція — симуляція — свобода. Проте найчастіше використовуваною класифікацією, хоч і не прийнятою усіма, жанри з якої зустрічаються в більшості існуючих, є наведена нижче, яка виключає осі або багаторівневі поділи:

**Action.** В іграх такого жанру необхідно використовувати рефлексі та швидкість реакції для подолання ігрових обставин. Це один із базових жанрів і водночас найпоширеніший. Як правило екшн-ігри пов'язані із агресивними діями щодо противників або оточення. Персонаж гравця повинен битися, стріляти, переслідувати ціль чи самому уникати переслідування. Стосовно екшн-ігор, де наявні значні елементи пригодницьких ігор, застосовується термін Action-

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

adventure. З-поміж них часто виділяється напрям аркадних ігор, ігровий процес яких вирізняється простотою та легкістю освоєння.

Цей жанр поділяється на велику кількість піджанрів, серед яких основними є:

Шутери — вимагають від гравця боротися з противниками шляхом стрілянини. Залежно від перспективи, поділяються на шутери від першої (Wolfenstein 3D) чи третьої особи (Макс Пейн). Існують різновиди як тактичні, в яких ігровий персонаж діє у складі команди (SWAT 4), аркади (Alien Shooter), стелс-екшн, метою якого є приховані дії для виконання завдань, без прямого знищення противників (серія Hitman). Щодо ігор, де основою ігрового процесу є знищення великих кількостей ворогів, а сама стрілянина в цьому переважає над тактикою і влучністю, застосовується термін Shoot 'em up (R-Type, Touhou, Contra).

Файтинги — імітують ближній бій, як правило один на один, на спеціальних аренах. Приклади: серія Mortal Kombat, Street Fighter.

Beat 'em up — подібні на файтинги, з тою різницею, що персонажі вільно переміщуються ігровим світом (а не спеціальною ареною) і борються проти багатьох противників одночасно. Приклади: Golden Axe, Battletoads, Double Dragon.

Платформери — персонаж мусить рухатися, стрибаючи по платформах, та долати перешкоди. Приклади: Mario, Spyro the Dragon, Megaman.

Лабіринти — персонаж рухається лабіринтом з метою знайти вихід, зібрати предмети і/або уникнути пасток і небезпек. Часто в іграх цього жанру є обмеження на час. Приклади: Pac-Man, Boulder Dash.

**Ігри – тренажери (симулятори):** різного роду гонки, військові та космічні ігри. Зазвичай у них гравець сидить у кабіні літака або автомобіля з екранами, важелями і кнопками. Звичайно, їздити в таких автомобілях та літати на таких літаках простіше, ніж у справжніх. Але смак можна відчути.

Найбільше нароблено ігор-автоперегонів (The Need For Speed, Test Drive); є літакові симулятори (Microsoft Flight Simulator, Red Jets); є навіть космічні кораблі та роботи (Mechwarrior, Wing Commander). У симуляторах теж важлива швидка реакція, оскільки їзда та польоти проходять з високою швидкістю, а бій – взагалі справа для моторних. Але не плутайте аркадні гоночки і польоти з симуляторами,

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

тому що в симулятори грати складніше, та й ігровий процес відрізняється більшим реалізмом (фізика в таких симуляторах максимально наближена до реальності, тому трохи позуєшся - машину занесе та інше).

**Спортивні симулятори** (NBA, FIFA, NHL) – імітації спортивних змагань із футболу, баскетболу, гольфу тощо. Щоправда, управління таким складним об'єктом, як людина, яка грає у футбол, поки не дуже вдається програмістам. Та й користуватися для цього мишкою не дуже зручно. Тому в такі ігри простіше грати із джойстиком.

**Стратегічні ігри (стратегії)** – ви будете міста, країни і навіть цілі планети, керуючи їх розвитком, будуючи будинки та дороги, проводячи електрику, обкладаючи мешканців податками, укладаючи союзи та оголошуючи війни. Суть ігрового процесу – добування деяких найважливіших ресурсів – енергії, територій, води, грошей, дерева, їжі, золота тощо. У таких іграх ви самі не берете участі в діяльності підпорядкованих вам територій або планет. Працюють інші, а ви їхній лідер та мозковий центр – король, президент, генерал, верховний маг. З погляду правил скоєння ходів стратегії ділять на покрокові (TBS – turn based strategy), де ходи здійснюються строго по черзі, як і шахах, і стратегії реального часу (RTS – real time strategy), де кожен гравець робить хід тоді, коли вважає за потрібне. Найвідоміші стратегії: Warcraft, Starcraft, Age of Empires, Command & Conquer. Втім, є і такий різновид стратегій, у яких ви й самі трішки бігаєте та стріляєте.

Якщо у такому фантастичному світі ви не верховний правитель і навіть не генерал, а рядовий учасник – воїн, маг, космічний торговець, то це вже називають **рольовою грою або RPG (role playing game)**.

А якщо крім вас і комп'ютера в цю ж гру грають на якомусь інтернет-сервері ще тисяча (або сто тисяч) людей, то такі забави називають вже розрахованими на багато користувачів ролевими іграми: MUG або MMORPG. У рольовій грі дуже важливо не тільки, яким персонажем ви граєте (які у нього здібності, сильний він чи, навпаки, розумний, воїн чи чаклун), а й те, яку ви вибрали для нього зброю та обладунки. У кожного виду зброї та обладунку свої тактико-технічні дані, своя забійна сила, ступінь захисту та довговічність. У міру розвитку гри ваш персонаж набирає очки. Після досягнення деякої магічної кількості окулярів він набуває

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

наступного ступеня могутності і майстерності: стає сильнішим, швидшим, може тягати більше всяких речей і приладдя. Одні з найвідоміших та найпопулярніших RPG – Diablo, Fallout, Lineage тощо.

**ММОРПГ** - Масова розрахована на багато користувачів рольова онлайн-гра (від англ. MMORPG - Massively multiplayer online role-playing game), або масові розраховані на багато користувачів рольові онлайн-ігри - вид рольових комп'ютерних ігор, в яких реальні гравці взаємодіють один з одним у віртуальному світі. Як і в більшості звичайних рольових ігор у ММОРПГ гравець керує персонажем, покращуючи його характеристики та взаємодіючи з іншими гравцями. Головні відмінності від РПГ (див. нижче) полягають у необмеженому числі учасників та безперервності ігрового процесу, що протікає цілодобово, тоді як гравці відвідують віртуальний ігровий світ у міру можливості. До речі, як правило, грати можна в такі ігри онлайн безкоштовно, якщо звичайно ви не почнете прокачувати свого персонажа за реальні гроші.

## **1.2. Вибір ігрового двигуна**

Движок гри (game engine) – це її основне ядро, базове програмне забезпечення, на основі якого будуються всі інші складові гри. Програмний код, який може використовуватися для створення варіацій гри, аддонів до неї або навіть зовсім нового ігрового світу.

Середньостатистичний ігровий двигун надає розробникам спосіб додавати такі речі, як: фізика, керування, рендерінг, необхідні скрипти, виявлення колізій, штучний інтелект, і т.п. без необхідності їх програмування

### **Огляд популярних середовищ для розробки ігрових проєктів**

#### **1. Unreal Engine**

Один з найпопулярніших ігрових двигунів – Unreal Engine від Epic Games. Оригінальна версія була випущена в 1998 році, а через 21 рік вона все ще продовжує використовуватися для великої кількості ігор щороку.

Найвідомішими іграми, створеними разом з Unreal Engine, є серія Gears of War, серія Mass Effect, серія Bioshock та серія Batman: Arkham.

Сила Unreal Engine полягає в тому, що його достатньо легко можна

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

модифікувати, щоб кожна гру можна було зробити дуже унікальною. Остання версія, Unreal Engine 4, вважається найпростішою у використанні в руках професіонала. Однак є й інші двигуни, які легші для нових дизайнерів.

## 2. Unity

Один з них – Unity, багатоплатформенний ігровий рушій, який дозволяє легко створювати інтерактивний 3D-контент. Дуже багато Інді-розробників віддають перевагу Unity за його відмінні функціональні можливості, якісний контент та можливість його використання майже для будь-яких ігор.

Останні відомі ігри, що створені використовуючи Unity, це Lara Croft Go, Her Story, Pillars of Eternity, та Kerbal Space Program. Найкраще, що можна сказати про Unity 5 – це Персональне видання, яке безкоштовне для завантаження. Це видання включає в себе рушій з усіма функціями і може (здебільшого) використовуватися для створення ігор на будь-якій платформі. Проблема полягає в тому, що Професійне видання, яке має безліч чудових інструментів, вимагає сплати щомісячної плати. Ці функції включають бета-доступ, звітування про ефективність ігор, можливість налаштувати екран заставки, ліцензію команди тощо.

## 3. GameMaker

На відміну від більшості інших ігрових рушії, GameMaker: Studio набув більш широкого поширення, оскільки для його використання не потрібні знання з програмування. Натомість користувачі можуть “обирати та клацати”, та створювати ігри набагато простіше та швидше, ніж кодувати рідними мовами.

Відомі ігри, що зроблені за допомогою GameMaker, це Spelunky, Hotline Miami, Super Crate Box та майбутній Hyper Light Drifter.

GameMaker популярний тим, що ви можете робити гру, не вивчаючи спочатку яку-небудь мову програмування. А ті, хто має досвід кодування, можуть використовувати цей рушій для покращення своїх ігор. Проблема з GameMaker та іншими двигунами “обирай та клацай”, полягає в тому, що розробники значно більш обмежені у функціях, ніж ті, хто використовує інші двигуни. І хоча існує безкоштовна версія, але щоб використати GameMaker на максимум буде необхідно придбати або професійну, або версію Master Collection.

## 4. Godot

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Рушій Godot чудово підходить для створення 2D та 3D ігор. Рушій пропонує величезний набір поширених інструментів, тому ви можете просто зосередитись на створенні своєї гри, не вигадуючи велосипед. Він безкоштовний у користуванні та з відкритим кодом через ліцензію MIT. Ні роялті, ні передплати, ні приховані виплат – все, що ви розробляєте за допомогою рушія Godot, – все ваше.

У Godot є спільнота, яка постійно виправляє помилки у роботі рушія та розробляє нові функції, що завжди є хорошим знаком. Активна спільнота також означає, що ви отримаєте відповіді навіть на ваші найбільш конкретні питання, пов'язані з Godot.

Godot також посилається на свої інші Інтернет-хаби, включаючи форуми Reddit, групи Facebook, спільноту у Steam, форуми Godot тощо.

## 5. AppGameKit

AppGameKit – це легкий у вивченні механізм розробки ігор, ідеально підходить для початківців, любителів та інді-розробників. Це схоже на вас? AppGameKit зосереджується на швидкому кодуванні та створенні додатків та можливості використання їх на різних платформах. Ця швидка ітерація та обмін між платформами орієнтована на розвиток мобільних ігор, але працює з більшістю платформ: iPhones and iPads, Linux, HTML5 браузері, Raspberry Pi, MacOS, Windows, Android.

AppGameKit коштує 79,99 дол. США з можливістю поєднати базовий продукт з додатками, такими як Visual Editor, щоб заощадити гроші на обох продуктах.

## 6. CryEngine

Платформа CryEngine безкоштовна у використанні. Ви отримуєте повний вихідний код рушія та всі його функції, не сплачуючи жодних ліцензійних платежів, роялті чи інших прихованих платежів.

CryEngine – надзвичайно приголомшливий рушій. Графіка гарна, а персонажі – життєві. CryEngine можна використовувати для розробки ігор для різних платформ: Xbox One, PlayStation 4, ПК з Windows, Oculus Rift.

Так, ви правильно зрозуміли – якщо ви хотіли зануритися у світ створення ігор віртуальної реальності, CryEngine покриває ваші потреби. Крім самого

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

двигуна, CryEngine також пропонує безліч безкоштовних навчальних ресурсів. Підручники, форуми та документація дають вам інструменти, необхідні для початка. Через потужність рушія, навчання працювати з ним є абсолютно необхідним. Цьому вам допоможе YouTube та офіційні ресурси CryEngine.

Якщо ви шукаєте ігрові ресурси, ви, ймовірно, зможете легко знайти їх на ринку. Ринок Cryengine пропонує такі пакети, як «гірський котедж» або «космічний саундтрек» в обмін на гроші. Також є доступні ресурси безкоштовно, як-от “Пакет для початківців CryEngine V” та “Вибухові речовини”.

## 7. Amazon Lumberyard

Amazon Lumberyard – це рушій, побудований Amazon (що зрозуміло випливає з назви). За даними сайту, Lumberyard – це “Безкоштовний. Потужний. Повністю налаштований”. Ігровий рушій не має “роялті або прихованих виплат, легка інтеграція з Twitch та AWS, а також багато іншого на підході”.

Починаючи від електронних книг, до фізичних продуктів, та навіть до продуктів харчування – все, до чого торкається Amazon, здається, стає простішим? Зручнішим?

AWS розшифровується як веб-сервіс Amazon – захищена хмарна платформа, що створена та підтримується Amazon. Інтеграція через Lumberyard означає, що ви можете легко побудувати онлайн-гру.

Єдина проблема – послуги AWS від Amazon, коштують грошей. Тим не менш, ви можете вибрати свій план (або взагалі не використовувати у своїй грі режим онлайн).

Якщо багатокористувацькі ігри – це те, що ви хочете, у Lumberyard є всі інструменти, щоб це зробити легко. Існує серія відео “Посібник з експлуатації” для

Lumberyard, яка допоможе вам швидко і легко пройти шляхом створення першого прототипу гри з цим рушієм.

## 8. RPG Maker

Найкраще, що можна сказати про RPG Maker – це те саме, чому він час від часу отримує погані оцінки – майже будь-хто може використовувати його для створення гри. Програмне забезпечення створене для того, щоб ви могли скласти повну гру від початку до кінця (навіть якщо ви нічого не знаєте про програмування

									Арк.
Змін.	Арк.	№ докум.	Підпис	Дата					

*РП 05.04.000.00 ДП*

ігор).

“Серія RPG Maker дозволяє налаштувати кожен аспект вашої гри за допомогою простого у користуванні інтерфейсу, що робить її ідеальною для початківців, але досить потужною для експертів”. Існує ряд продуктів RPG Maker. RPG Maker MV коштує \$ 79,99. Для користувачів Windows є безкоштовна пробна версія.

Досвідчені користувачі можуть скористатися своїми знаннями Javascript, щоб “змінити гру на свій смак, починаючи від вигляду битв та закінчуючи інтерфейсом меню. Користувачі без знань JavaScript обмежуватимуться лише тим, що пропонує рушій, але це все одно більш ніж достатньо для створення гри.

## 9. LibGDX

Libgdx – це рушій для розробки ігор Java, яка пропонує єдиний API, який працює на всіх підтримуваних платформах. API – це аббревіатура для інтерфейсу програмування прикладних програм – наявність єдиного API робить libGDX чудовим вибором для розробки на різних платформах. Не має значення, на яку платформу ви орієнтовані: Windows, iOS, Linux, Mac OS тощо. Всі використовують один і той же API.

LibGDX дозволяє запускати та відладжувати гру на самому своєму робочому столі. Це дозволяє легко генерувати швидкі ітерації вашої гри та швидко протестувати зміни (оскільки для перевірки цих змін вам не потрібно буде запускати iOS / Android тощо). Форуми libGDX також живі, з темами, які висвітлюють розробку за допомогою libGDX та підручники для початку роботи.

Існує також “посібник з джерелами” з демонстраціями та прикладами, якими ви можете скористатися.

## 10. Urho3D

Urho3D – це безкоштовний легкий, кросплатформенний 2D та 3D ігровий рушій, реалізований на C++ та випущений під ліцензією MIT. Вікі Urho3D містить усе необхідне, щоб розпочати роботу з рушієм, включаючи інструкції щодо налаштування: Windows із Visual Studio, Windows з MinGW, Android, Linux

Існують посібники для створення вашого першого проекту та ряд інших тем форуму, починаючи від того, як вирізати дірки у місцевості до розробки основних

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

матеріальних ефектів для візуалізації.

Urho3D зараз на версії 1.7, і як ми вже згадували, завантажувати його можна абсолютно безкоштовно. На веб-сайті є повний перелік функцій, включаючи версію, в якій ці функції були вперше реалізовані.

Urho3D також описує обмеження у його використанні, включаючи той факт, що вам знадобляться навички “С++ для написання коду та вдосконалення існуючих підсистем, таких як мережа, фізика та анімація, залежно від ваших потреб”. Короткий список обмежень у використанні рушія доступний у повному обсязі на їх веб-сайті.

### 1.3. Етапи проектування ігрових додатків

Процес створення мобільної гри можна грубо розділити на 2 етапи — препродакшн та продакшн: підготовка до випуску ігрового контенту та саме виробництво. Однак, щоб розібратися у важливих тонкощах, варто копнути глибше. У більшості випадків при створенні гри з нуля продукт проходить 7 стадій розробки: постановка мети, пошук ідеї, створення концепту, прототипування, закрите бета-тестування або плейтест, софт-лонч, реліз

Ціль - це перше, з чим необхідно визначитися перед початком розробки. Мета має бути конкретною, вимірною, досяжною та реалістичною. І не варто забувати про те, що завжди будуть обмеження часу. Забути про мету це все одно, що піти в похід без карти або почати будівництво будинку без плану. Постановка мети гарантує, що ви не тільки знаєте, що збираєтеся робити, а як ви збираєтеся це робити. Коли ціль сформульована, можна переходити до наступного, не менш важливого етапу.

**Препродакшн** – це процес підготовки до створення відеоігри. Етап препродакшна включає 3 підетапи: пошук ідеї, створення концепту та прототипування. Мета препродакшна — визначити ідею гри, продумати, що зробить вашу гру унікальною і чим вона відрізнятиметься від усіх інших, як і чому вона стане хітом. Для цього необхідно підготувати основну документацію (геймдизайн-документ) та пропрацювати всі слабкі місця перед запуском гри в продакшн. На цьому етапі ми створюємо документацію, яка детально описує гру. Так усі розуміють, що належить зробити на етапі продакшну.

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

Грамотно спланований препродакшн скорочує тривалість наступного етапу та економить бюджет. За рахунок чого? — Якщо враховано ризики, виявлено фактори успішності проекту, а економіка гри, унікальні фічі та механіка доведено до досконалості, глобальних правок під час продакшну буде менше.

### **Пошук ідеї**

Можна створити десятки проектів зі зрозумілими механіками, пізнаваними сюжетними поворотами та чудовою графікою, і при цьому не отримати визнання гравців. Щоб проект став справді успішним, у ньому має бути «родзинка». Потрібно викликати у гравців емоції, які хоч трохи відрізнятимуться від того, що можна випробувати у реальному світі. Але при цьому необхідно дотримуватися балансу між знайомим для гравця світом і чимось незвичайним, адже приходять у гру саме за новими враженнями. В ігровій промисловості є багато різних жанрів та сеттингів. Наприклад, такі жанри, як FPS/TPS, RTS, Horror та всілякі симулятори. А сеттинг може бути історичний, фентезі, стімпанк чи кіберпанк. На відміну від сеттинга далеко не всі жанри підходять для мобільних ігор. Тому вже на етапі пошуку ідеї ви повинні уявляти, як ваша гра виглядатиме в результаті. Наприклад, уявіть собі ситуацію: Ваш гравець після важкого робочого дня їде додому на метро. Він їде стоячи, а у вагоні сильна товкучка. Щоб трохи відсторонитися від сірої реальності і пощекотати нерви, він тягнеться за своїм телефоном і хоче насолодитися вашим Хоррором. Абсурд? Правильно! Мобільний пристрій не дозволяє досягти того ж рівня занурення, що PC, приставка або VR.

Продумайте класні та, головне, доречні для мобільного геймінгу ідеї до початку розробки гри.

### **Створення концепту**

Це підготовчий етап, який має на меті відшліфувати ідею майбутньої гри та зібрати базову геймдизайнерську документацію (Pitch Document або Concept Document). Вони викладають короткий, але при цьому ємний опис гри з ключовими аспектами. У процесі створення концепт-документу ми відповідаємо на запитання: що за гру робимо і для яких платформ, про що ця гра, для кого призначено фінальний продукт та хто наші конкуренти. У Concept Document ми описуємо сюжет, сеттинг, основну механіку гри. Такий документ використовують

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

усередині компанії і представляють видавцеві.

Важливий момент: всі документи, що геймдизайнери, повинні регулярно актуалізуватися і бути доступні всім співробітникам. Це необхідно для того, щоб кожен член команди, будь то художник, програміст або сценарист, могли в будь-який момент звернутися до документа і зрозуміти, чи проміжний результат сходиться з тим, що було заплановано.

### **Прототипування**

На цьому етапі команда розробників гри прототипує свій концепт: перевіряє ідеї та гіпотези, втілюючи їх у невеликих простих прототипах. Наприклад, створює 2-3 варіанти візуального оформлення рівнів, щоб у результаті вибрати найвдаліше. Те саме стосується механік та поведінки персонажів. На прототип зазвичай витрачають мінімум часу — тестують лише те, що потрібно, і не звертають увагу на його «красу». Складний у реалізації та «причесаний» прототип ніяк не вплине на фінальний вид гри, тому витрачати ресурси на його підготовку не варто.

### **Продакшн**

Найбільш ресурсозатратний етап створення гри. Завдання - випустити повноцінну версію гри для внутрішнього перегляду продюсерами, закритого бета-тестування, техлончу або софт-лончу. На цьому етапі створюється гра. Команда готує необхідний арт, анімує, налаштовує ігровий баланс, програмує та багато тестує. Результат продакшну — перша версія гри.

### **Закрите бета-тестування або плейтест**

На етапі ЗБТ готову гру демонструють нову, обмежену за кількістю гравців аудиторію. Це допомагає знайти геймдизайнерські помилки, виявити проблеми та побачити загальну картину того, як гравці взаємодіють із фінальним продуктом. Розробники відеоігор аналізують дані проведених плейтестів і роблять висновки: що в грі добре, а що погано, які потрібні зміни та доопрацювання. Але навіть під час таких тестів продакшн на великих проектах триває: доопрацьовуються фічі, гру призводять до фінального вигляду та зустрічі з більш чисельною аудиторією.

### **Софт-лонч**

У цей момент починається справжнє "життя" гри. На етапі софт-лончу продукт представляють ширшій загальній, наприклад, відкривають доступ до гри в

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

декількох країнах. Здійснюють перевірку технічних метрик і параметрів, ваги гри, швидкості завантаження та якості роботи на різних платформах, оптимізують гру під прийом трафіку з постійним його збільшенням. Йде збір статистики, обробка геймерського фідбеку. Виходячи з даних програмісти шліфують знайдені недоробки, продюсер і геймдизайнери аналізують статистику.

### Реліз

На цьому етапі гру просувають, розвивають ком'юніті і, якщо необхідно, розширюють сервіс техпідтримки, запускають маркетингові заходи, ведуть роботу над новим контентом. Продюсер та геймдизайнери проробляють статистичні дані, шукають шляхи покращення показників (наприклад, показники утримання гравців) та впроваджують нові фічі – настає період Live-Ops. У цей час у грі з'являються заздалегідь заготовлені події, змагання, пропозиції про знижки – робота над продуктом продовжується. Після релізу розробники отримують більше фідбеку від гравців, а отже, можуть вивчити аудиторію, її інтереси та потреби. Все подальше виробництво зводиться до покращення продукту, регулярних оновлень, наповнення гри новим контентом та можливостями, які стимулюватимуть геймерів грати частіше і довше.

Мій ігровий додаток відноситься до таких жанрів як: Інді-ігри (відеоігри, створені незалежно від фінансової підтримки великих видавців), Adventure (жанр відеоігор, основу ігрового процесу яких складає вирішення поставлених завдань шляхом їх обдумування, уважного пошуку підказок і схованих деталей - такі ігри можуть як мати одного чи кількох головних персонажів, так і не мати їх взагалі) також має елементи Psychological horror(Психологічні ігри жахів покликані налякати гравця емоційними, психічними чи психологічними станами, а не монстрами чи страхами).

На початку гри ви прокидаєтесь у невідомому вам місці та робите усі зусилля щоб покинути це огиднене місце.

Загалом у виборі технологій зі свого погляду я керуюсь двома основними підходами: технологічний та бізнесовий.

Під технологічним я маю на увазі яка технологія найкраще підходить під проект. Допустимо на ній є необхідні бібліотеки, більшість необхідного

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		



## **2. РОЗРОБКА КОНЦЕПТ-ДОКУМЕНТУ ДЛЯ ІГРОВОГО ПРОЕКТУ**

Ідея гри: Ти граєш за дівчину якій сниться поганий сон, вона прокидається у якомусь невідомому місці, здається це сховище біля тебе кровать та ще кілька предметів, тобі подрібно догадатися що робити щоб покинути цю кімнату. Це все відбувається у наш час, гра закінчується коли ця дівчина з ім'ям Вероніка прокинеться.

Жанр і аудиторія

Жанр – Adventure(жанр відеоігор, основу ігрового процесу яких складає вирішення поставлених завдань шляхом їх обдумування, уважного пошуку підказок і схованих деталей.

Широка аудиторія, вік 18+.

Платформа – Windows.

Сеттінг – нічний кошмар який відбувається у підвалі, хоррор, фентезі.

Особливості гри

1. Наявність сюжетної лінії з історичною прив'язкою.
2. Гравець керує тільки головним героєм (звичайна людина).
3. Людина читає записки.
4. Записки дають шлях та підказки

### **Опис гри**

Пекло — це експеримент, який ви не можете вижити в Nightmares Veronicy, гри жахів на виживання від першої особи.

При відкритті гри ти знаходишся у головному меню де тобі показуються кнопки взаємодії з предметами регулятор гучності музики та кнопка старти (початок гри) та кнопка вихід.

Ти граєш за дівчину та прокидаєшся у сховищі якого ти ніколи не бачив тобі не відомо де ти, ти опиняєшся у безвихідній ситуації тобі потрібно за допомогою твоєї логіки здогадатися як покинути цю кімнату, шукаєш будь-які варіанти вибратися звідти, шукаєш будь-які деталі. Це гра про вивчення, життя та занурення в атмосферу нічного кошмару. Це те, від чого у вас точно побіжать мурашки по шкірі. Ви не отримуєте жодної допомоги із зовнішнього світу.

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

Все, що у вас є – голова та руки.

На протязі усієї гри ти будеш у постійному страху та будеш шукати підказки, тобто записки, кров яка показує шлях та ін. На протязі гри головна героїня буде бачити своє ім'я та фото в деяких місцях зі свого прошлого. Усе це змішано у цій грі. Також в грі присутнє звукове супроводження, в головному меню - це музика. У грі вона застосовується для створення атмосфери та нагнетіння моментів.

Почули звук чийось кроків? Чи це просто гра вашого розуму? Відчуйте...

Із застосуванням світу, що діє за всіма законами фізики, 3D-графіки та динамічного звукового супроводу, цей світ поступово поглине вас. Контроль за грою повністю у ваших руках. Тут немає внутрішньоігрових відео, стрибків у часі – все, що станеться з вами, буде викликане саме вашими діями.

Гра завершується коли головна героїня прокидається при проходженні гри. У Кінці гравцю виражається велика подяка за проходження гри.

Отже побачимо чи зможеш пройти кімнати та завершити гру чи не зможеш.

Порівняння з аналогами.

1. Outlast – Сеттінг, Схожі локації та атмосфера, жанр
2. The Bunker – Атмосфера, дуже схожа локація.
3. Amnesia – Сеттінг, жанр та атмосфера, спроба надавити на гравця

психологічно

Основні відмінності:

1. Двері відчиняються за допомогою ключів які ти знаходишь
2. Місія одна.
3. Видумана історія з елементами хоррору

Засоби розробки

Ігровий двигун – Unity.

Мова програмування – C#

Додаткове ПЗ: Blender, Visual Studio, ПЗ для звукового супроводу.

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Ресурси (програмні та апаратні)

1. ОС: Windows 10, тільки 64-разрядні версії; intel core i3 8100: Geforce gtx 1050Ti. 8gb CPU

2. Системні характеристики: Операційна система: XP, Vista, Windows7.

Процесор: 2 GHz Intel Pentium 4 або AMD Athlon.

Оперативна пам'ять: 1 Гб.

HDD: 250 MB на жорсткому диску

Відеокарта: 1 GB Graphics card з 1 GB пам'яті

Версія DirectX: 9.0.

А також: Клавіатура, миша

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

### 3. РОЗРОБКА ТА РЕАЛІЗАЦІЯ ІГРОВОГО ДОДАТКУ В UNITY

#### 3.1 Створення проекту та його налаштування

Для того щоб установити Unity переходимо за посиланням: [unity3d.com](http://unity3d.com): у верхньому правому куті натисніть на «Купити Unity» виберіть Personal (тиснемо на «Спробувати Personal») під кнопкою завантаження натисніть на одне з двох: "Виберіть Windows" або "Виберіть Mac" після вибору поставте галочку з підтвердженням та завантажте.

Відкриваємо настановний файл. - Натискаємо "Далі". - Чекаємо на встановлення з'єднання -Приймаємо умови-ліцензійної угоди -Відзначаємо необхідні параметри, які вказані на картинках -Вибираємо директорію установки або залишаємо як є-Чекаємо на закінчення завантаження та установки - Відкриваємо програму-при необхідності створюємо акаунт у Unity3D.

Для того щоб створити наш ігровий проект потрібно відчинити Unity Hub.

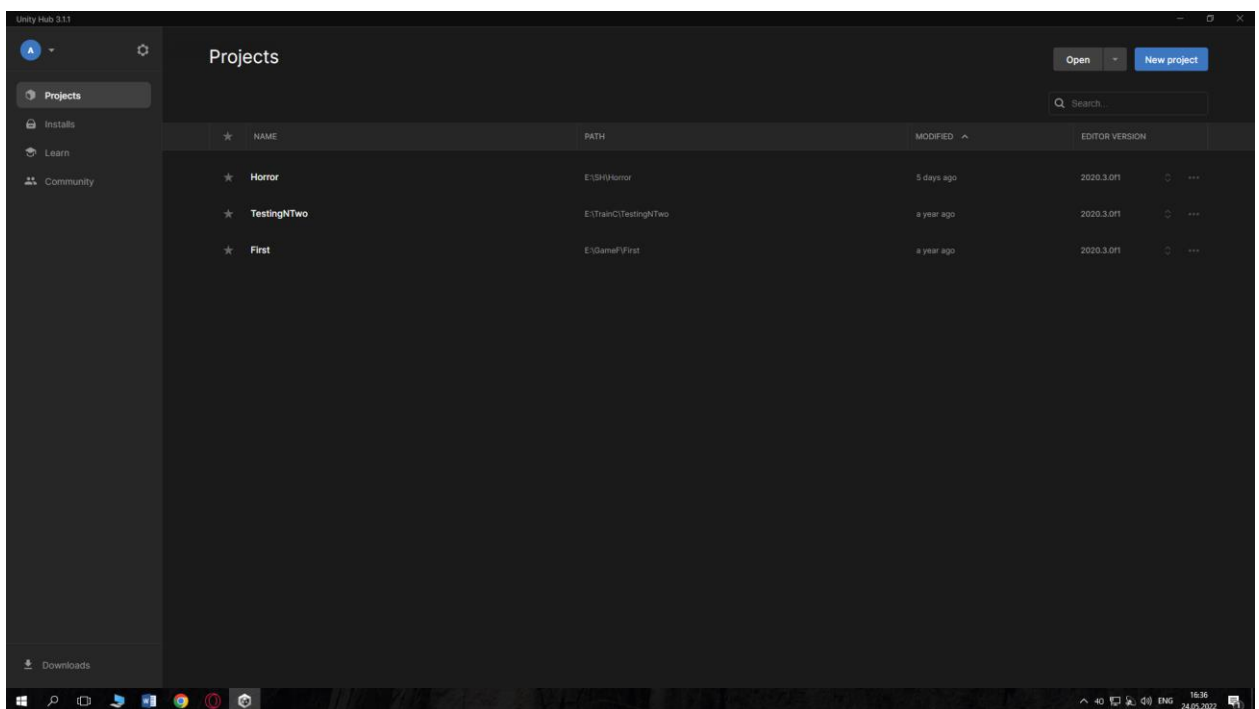


Рис.3.1. –Головне меню Unity

У вас відкриється меню в якому ви можете створити ваш проект. Для початку натисніть справа зверху кнопку New project.

Після цього у вас з'явиться панель для вибору чого саме ви хочете і де саме ви будете зберігати ваш проект. Після вашого вибору ви натискаєте Create project. Ось і з'явився ваш проект.

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

## 3.2 Інтерфейс розробника

Головне окно редактора складається з кількох вкладок, що називаються Видами ( Перегляди ). У Unity є кілька видів - все вони призначені для конкретних цілей, описаних у цьому розділі.

Це означає, що зовнішній вигляд редактора може відрізнятись від одного проекту до іншого та від одного розробника до іншого, залежно від особистих уподобань та типу роботи, яку ви виконуєте.

Розташування вікон за замовчуванням дає практичний доступ до найпоширеніших вікон. Якщо ви ще не знайомі з різними вікнами в Unity, ви можете визначити їх за назвою на вкладці. Найпоширеніші та найкорисніші вікна показані в позиціях за замовчуванням нижче

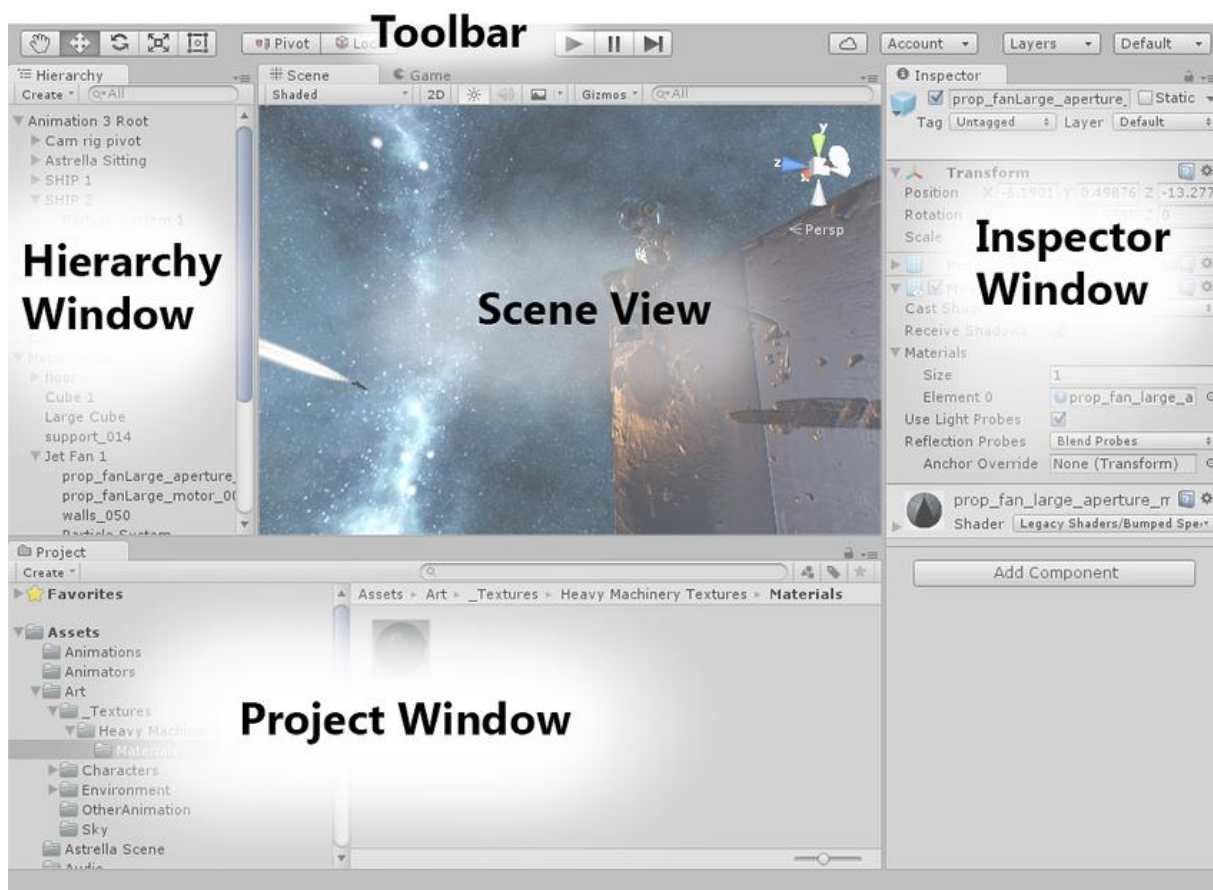


Рис.3.2. – Помітки інтерфейсу

Змін.	Арк.	№ докум.	Підпис	Дата

## Вікно проекту

У вікні проекту відображається ваша бібліотека ресурсів, які доступні для використання у вашому проекті. Коли ви імпортуєте активи в проект, вони з'являються тут.

## Вид сцени

Перегляд сцени дозволяє візуально переміщатися та редагувати сцену. Перегляд сцени може показувати 3D або 2D перспективу, залежно від типу проекту, над яким ви працюєте.

## Вікно ієрархії

Вікно ієрархії — це ієрархічне текстове представлення кожного об'єкта в сцені. Кожен елемент сцени має запис в ієрархії, тому два вікна за своєю суттю пов'язані. Ієрархія розкриває структуру того, як об'єкти приєднуються один до одного. Ієрархія (Hierarchy) містить всі об'єкти (GameObject) у поточній сцені. Деякі з них є прямими примірниками файлів ассетів, таких як 3D-моделі, а інші - екземпляри префабів, об'єктів користувача, з яких складається більша частина вашої гри. Ви можете вибрати об'єкти в ієрархії, і перетягнути один об'єкт на інший, щоб створити батьківський зв'язок (Parenting). При додаванні та видаленні об'єктів у сцені вони також будуть з'являтися та зникати з Ієрархії.

## Вікно інспектора

Вікно інспектора дозволяє переглядати та редагувати всі властивості поточного вибраного об'єкта. Оскільки різні типи об'єктів мають різні набори властивостей, макет і вміст вікна інспектора буде відрізнятися.

## Панель

## інструментів

Панель інструментів надає доступ до найважливіших робочих функцій. Ліворуч він містить основні інструменти для маніпулювання переглядом сцени та об'єктами в ній. У центрі розташовані елементи керування відтворенням, паузою та кроком. Кнопки праворуч надають вам доступ до ваших хмарних служб Unity та вашого облікового запису Unity, за якими слідує меню видимості шарів і, нарешті, меню макета редактора (яке надає деякі альтернативні макети для вікон редактора та дозволяє вам зберігати власні власні макети).

Панель інструментів не є вікном і є єдиною частиною інтерфейсу Unity, яку

					РП 05.04.000.00 ДП	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

ви не можете змінити.

### 3.3 Створення ігрового світу

Для створення ігрового світу нам потрібно встановити менеджер з ім'ям ProBuilder він дуже полегшить роботу із створенням нашого світу.

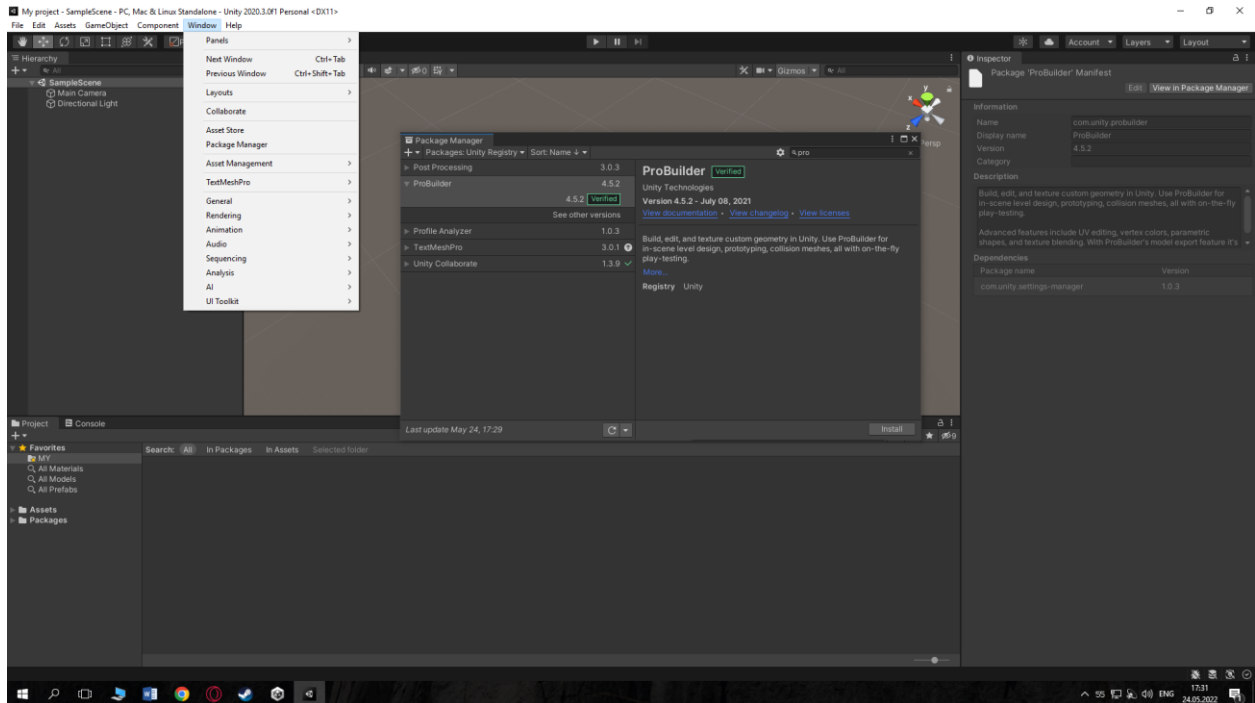


Рис.3.3 – Встановлення менеджера ProBuilder

Коли ми його відкриємо у нас з'явиться ось така панель. За допомогою цього менеджера я створив мій ігровий світ.

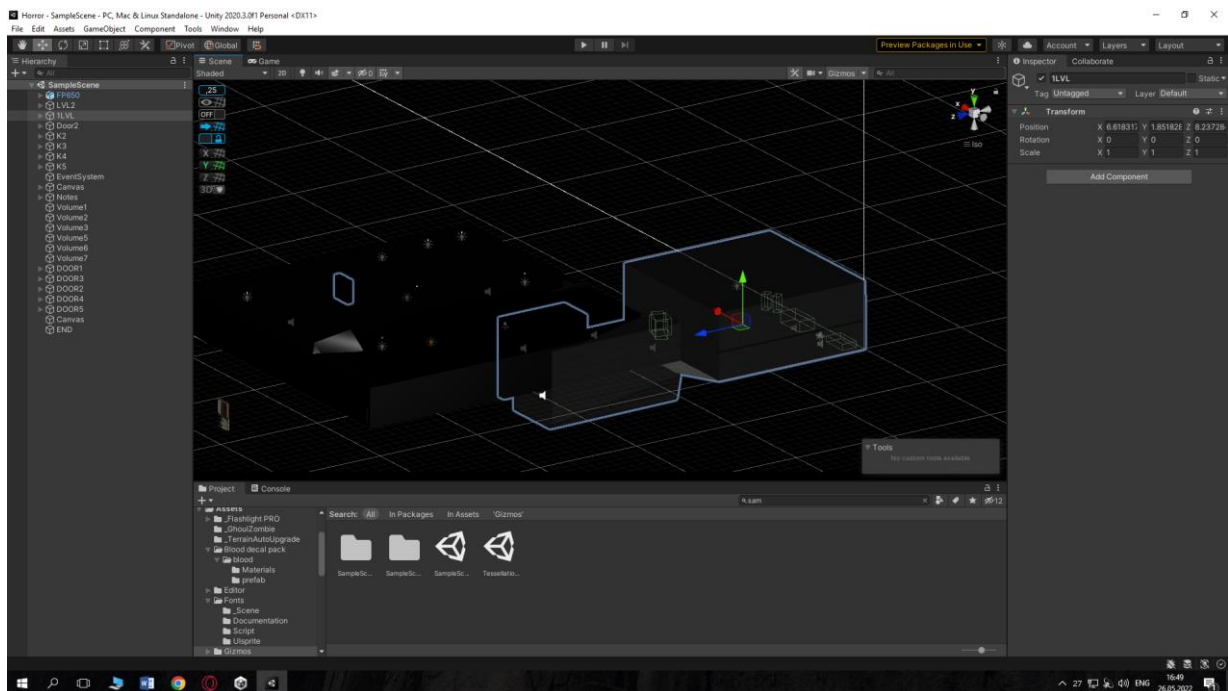


Рис.3.4. – Мій ігровий світ

						Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		
<i>РП 05.04.000.00 ДП</i>						

З'явився ось такий тривимірний мир за допомогою цього менеджера. В загальному на створення миру було витрачено приблизно 2 місяця. Весь мир був імпровізований та створений за своїми побажаннями.



**Рис.3.5. – Перша локація**

За допомогою ассетів та ProBuildera були додані ігрові об'єкти. На малюнку ми бачимо декілька моделей (Шафа, доска, стіл та ін.). Ігрові моделі були встановлені з asset store.

### 3.4 Програмування камери

Ось такий код написан у методі старт(Код виконується коли гра запускається). За допомогою коду знизу ми зможемо керувати нашою камерою та дивитися по осі: x, y, z. Зможемо пригати, приземлитися та бігти.

Також буде чути шаги та наш біг. Отримаємо нормаль для переміщення по ній поверхні, якої торкаються.

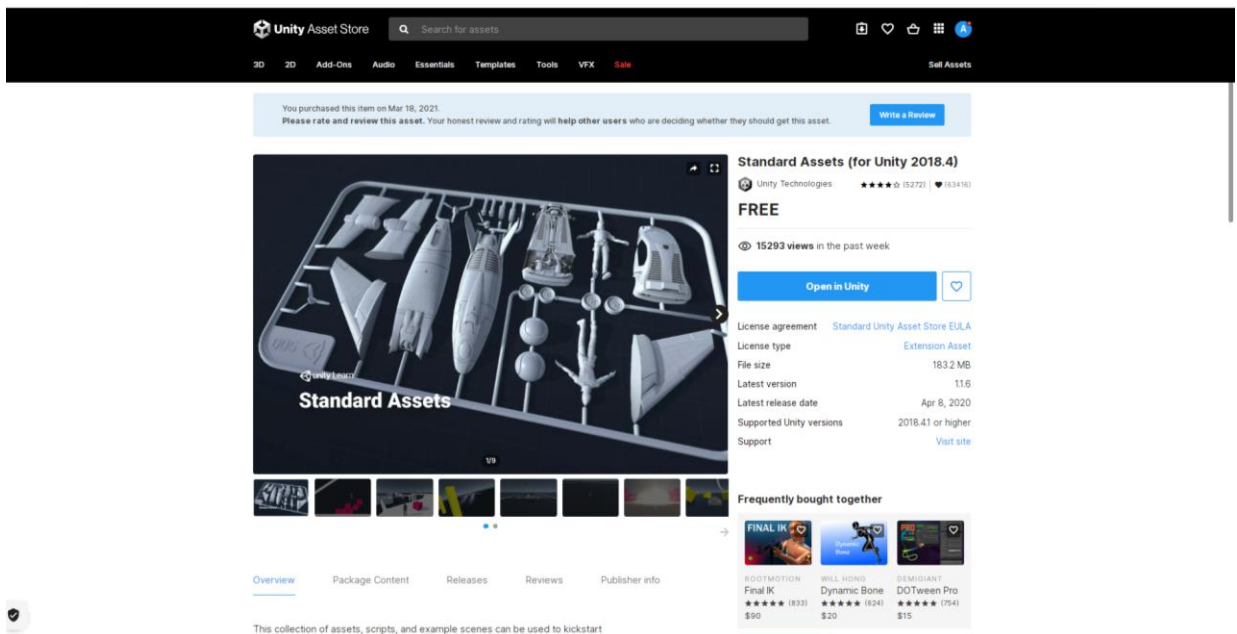
На 99 рядку прямо сказано - завжди рухатися вздовж камери вперед, оскільки це напрямок, на який вона спрямована.

Все що нижче ми отримаємо нормаль для переміщення по ній поверхні, якої торкаються.

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		



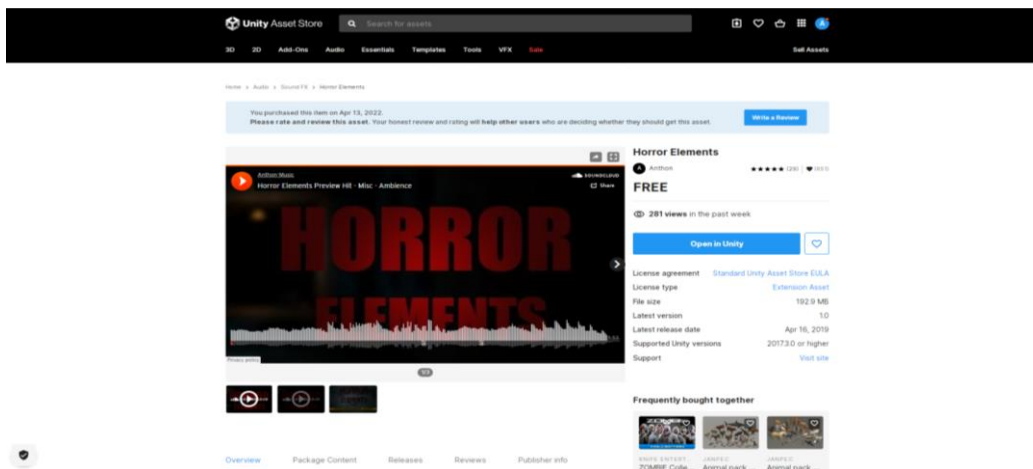




**Рис.3.9. – Standart Assets**

Цю колекцію асетів, сценаріїв і прикладів сцен можна використовувати для початку навчання Unity або використовувати як основу для ваших власних проєктів.

По-друге, це **Horror Elements**.



**Рис.3.10. – Horror Elements**

Майже усі звуки які додають атмосферу до цієї гри, взяті з цього асету (Музика з головного меню, напружені звуки)

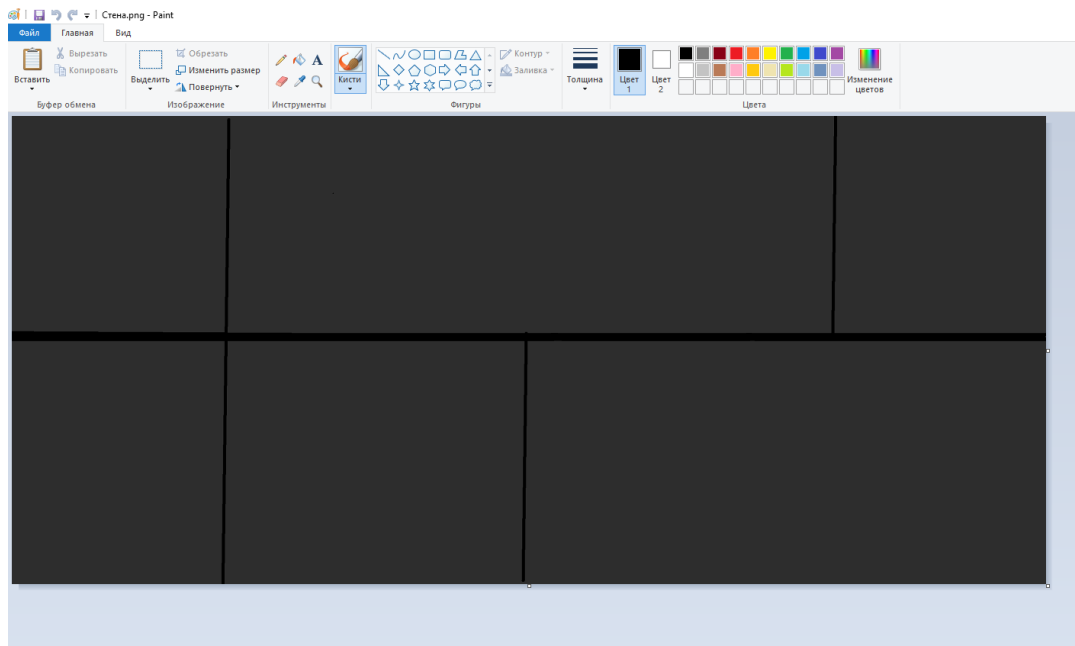
По-третьє, це **Hospital Horror Pack**.

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		



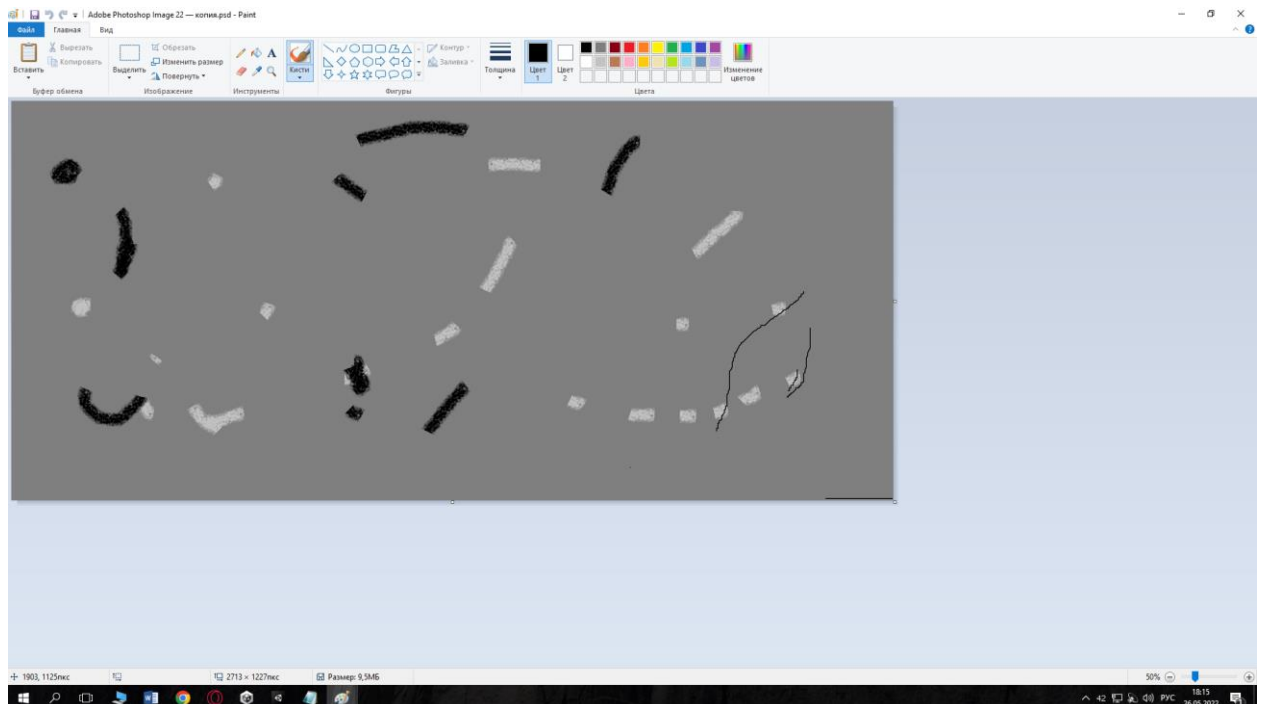
Зараз ви бачите стіну, яка була намальована в звичайному Paint.

До цього малюнку застосовувалися звичайні прямі лінії.



**Рис.3.13. –текстура стіни у Paint**

Цей малюнок був імпортований до Unity і вже потім накладений на ігрову модель, в нашому випадку це стіна.



**Рис.3.14. –текстура підлоги у Paint**

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

На цьому малюнку вже намальована підлога, яку ми намалювали у Paint.



**Рис.3.15. – Приклад накладання текстури на підлогу**

На цьому ж прикладі ми маємо бетону, брудну підлогу яка не блищить, бо це бетон.

Ось так виглядає вже з відкорегованим матеріалом наша підлога.

За допомогою поганого освітлення усе погано видно та присутня якась інтрига.

### **3.7 Програмування ігрових об'єктів**

Скриптинг – необхідна складова всіх ігор. Навіть найпростіші ігри потребують скриптів для реакції на дії гравця та організації подій геймплею. Крім того, скрипти можуть бути використані для створення графічних ефектів, управління фізичною поведінкою об'єктів або реалізації користувальницької ІІ системи для персонажів гри.

Почнемо зі скрипту коли гравець входить в поле дії цього скрипту і відбувається грання аудио звуку.

Ігровому об'єкту додаємо Box Collider, Audio Source. Додаємо скрипт.

Компоненти Box Collider визначають форму об'єкта для цілей фізичних зіткнень. У разі зіткнення, система скриптингу може це виявити та виконати дії, вказані у функції OnCollisionEnter. Однак ви також можете використовувати фізичний рушок просто для виявлення того, що один коллайдер входить в простір іншого, без створення колізії.

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class SoundsScript : MonoBehaviour
6 {
7     public float TimerForDestroy;
8     [Header("Ігрок")]
9     public Collider MyPlayer;
10    [Header("звук")]
11    public AudioClip MySound;
12    public GameObject mySoundSource;
13    [Header("console")]
14    public string MyMessage;
15    void OnTriggerEnter(Collider MyPlayer)
16    {
17        mySoundSource.GetComponent<AudioSource>().PlayOneShot(MySound);
18        Debug.Log(gameObject);
19        Debug.Log(MyMessage);
20        Destroy(gameObject, TimerForDestroy);
21    }
22 }
23
24
25

```

**Рис.3.15. – Скрипт прогання аудіо**

Коллайдер, настроений як триггер (за допомогою властивостей Is Trigger), не веде себе як твердий об'єкт і просто буде пропускати інші коллайдери скрізь себе. Коли інший коллайдер входить «на територію» цього коллайдера, триггер викликає функцію OnTriggerEnter у скриптах об'єкта, до якого приєднано триггер.

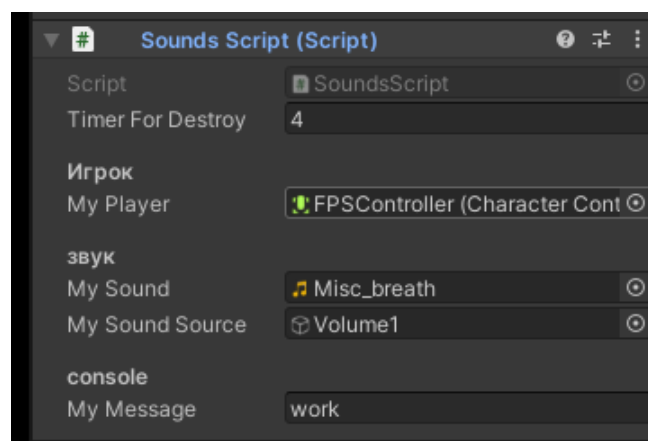
З 15 рядка у нас описується метод, коли гравець входить у цей об'єкт.

17 рядок – програтися аудіо вибране нами.

18-19 рядки вивід у консоль повідомлення задля перевірки чи відбувся метод.

20 рядок – видалення ігрового об'єкту через потрібний нам час (задля того, щоб завжди аудіо не повторювалося коли гравець входить у цей об'єкт).

Ось такі параметри ми можемо вводити у наш об'єкт.

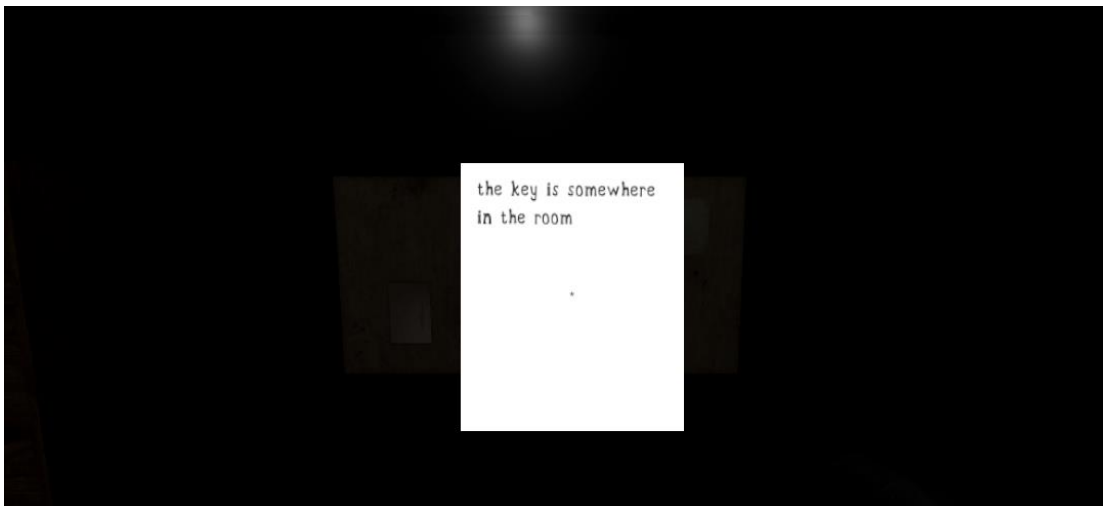


**Рис.3.16. – Панель прогання аудіо**

Наступне – це читання записок на яких написан текст.

У My Player ми переміщуємо нашу камеру, у My Sound ми переміщуємо нашу звукову доріжку, у My Sound Source ми переміщуємо наш об'єкт, у My Message виводиться повідомлення яке ми напишемо.

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		



**Рис.3.17. – Приклад читання записки**

Для цього нам потрібно створити Canvas (Компонент Canvas є абстрактним простором, в якому проводиться налаштування та відмальовування UI. Усі UI-елементи мають бути нащадками ігрових об'єктів, до яких приєднано Canvas. Коли ви створюєте UI-елемент із пункту меню (GameObject > Create UI), Canvas буде додано автоматично, якщо його немає у сцені).

Наступний крок – додати Image, на нього Text та додати тег.

Після цього приєднати скрипт на нашу камеру. Після нашого коду ми зможемо зажимати клавішу E та читати приближений спрайт. А коли ми його дочитаємо нам залишиться відпустити клавішу.

```

12 void Start()
13 {
14     ...
15 }
16
17 // Update is called once per frame
18 void Update()
19 {
20     ...
21     if (!isread)
22     {
23         RaycastHit hit;
24         if (Physics.Raycast(Camera.main.transform.position, Camera.main.transform.forward, out hit, 2f, mask))
25         {
26             if (hit.collider.tag == "letter")
27             {
28                 ...
29                 if (Input.GetKeyDown(KeyCode.E))
30                 {
31                     letter = hit.transform;
32                     isread = true;
33                 }
34             }
35         }
36     }
37     else
38     {
39         letter.position = Camera.main.transform.position + Camera.main.transform.forward * .5f;
40         letter.LookAt(Camera.main.transform);
41         if (Input.GetKeyUp(KeyCode.E))
42         {
43             isread = false;
44             Destroy(letter_gameObject);
45         }
46     }
47 }
48
49
50

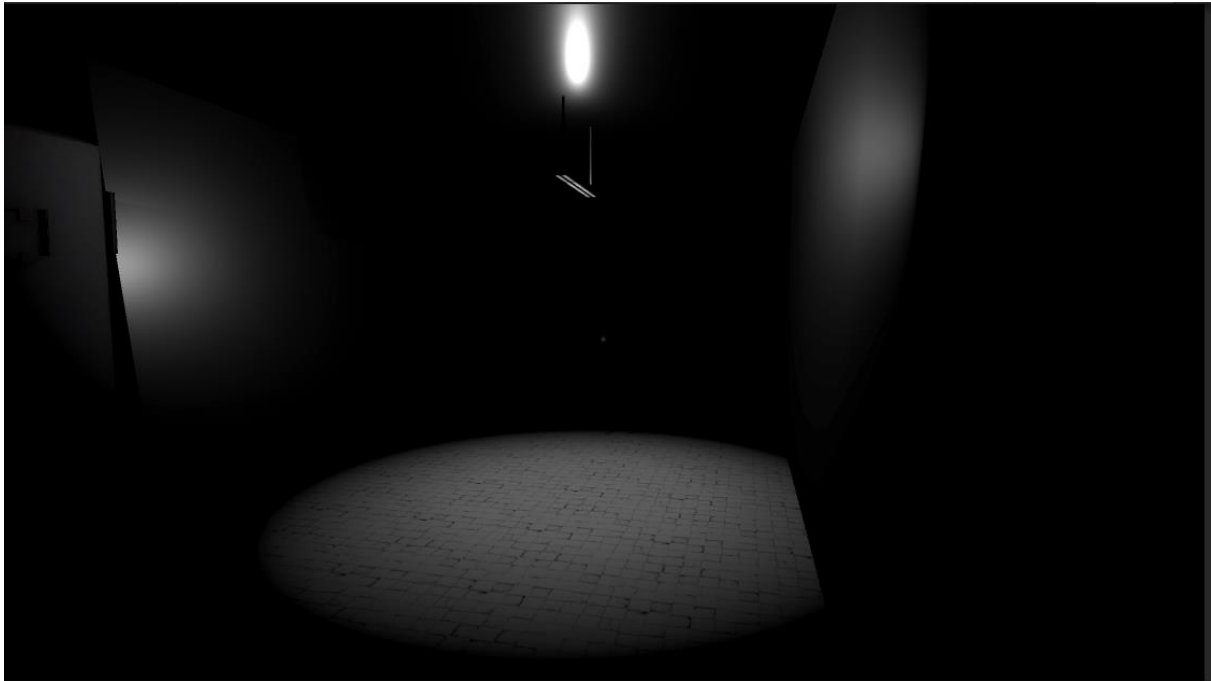
```

**Рис.3.18. – Скрипт читання записки**

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

У 21 рядку відбувається перевірка за якою з середини екрану вистрілює промінь, та якщо він потрапляє у об'єкт з тегом **letter** то відбувається наближення камери за нажимання кнопки E.

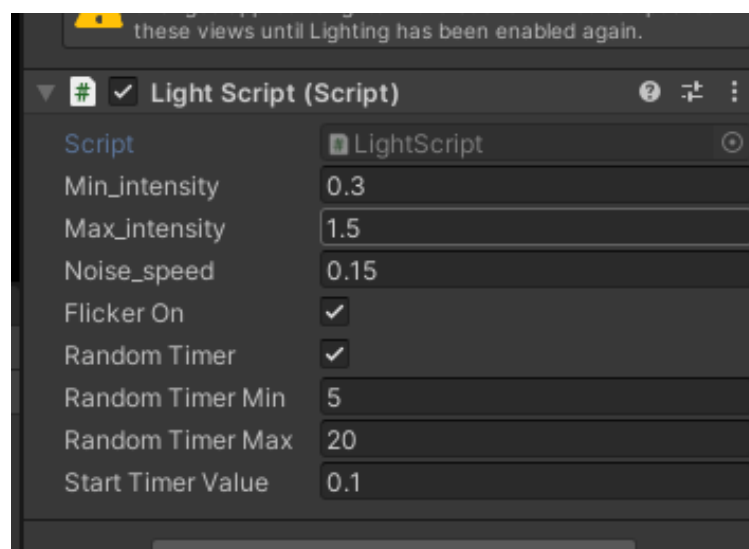
На наступному скрипті ми робимо мерехтливе світло на лампі.



**Рис.3.19. – Приклад світла з лампи**

Додається компонент Light до ігрового об'єкта.

На цьому малюнку ми беремо потрібний час інтервалу до наступного мерехтлива світла, та в методі Update це відбувається кожен кадр.



**Рис.3.20.– Панель скрипту для мерехтливого світла**

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		



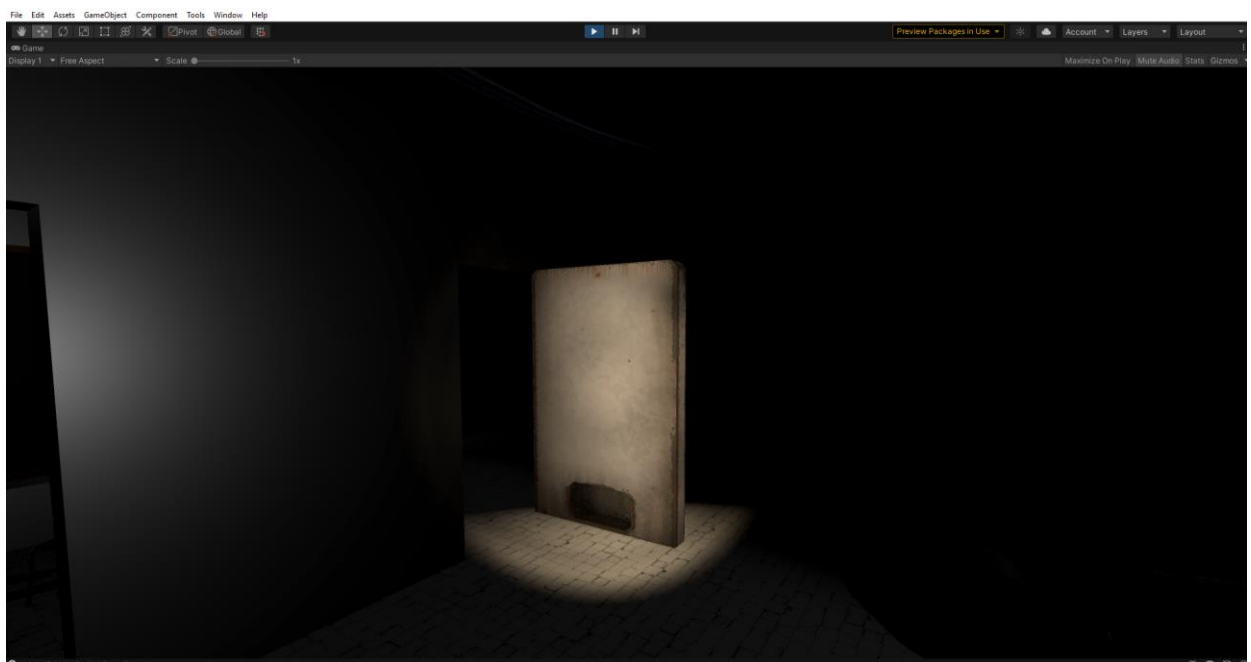
```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 [SerializeField]
6 public class Door : MonoBehaviour
7 {
8     [SerializeField]
9     float openDoor;
10    [SerializeField]
11    float closeDoor;
12    [SerializeField]
13    float speed = 1;
14
15    public bool isOpen;
16    public bool isLocked;
17    public int id;
18
19    void Update()
20    {
21        if (isOpen)
22        {
23            OpenDoor();
24        }
25        else
26        {
27            CloseDoor();
28        }
29    }
30
31    void OpenDoor()
32    {
33        transform.rotation = Quaternion.Slerp(transform.rotation, Quaternion.Euler(transform.rotation.x, openDoor, transform.rotation.z), speed * Time.deltaTime);
34    }
35
36    void CloseDoor()
37    {
38        transform.rotation = Quaternion.Slerp(transform.rotation, Quaternion.Euler(transform.rotation.x, closeDoor, transform.rotation.z), speed * Time.deltaTime);
39    }
40
41 }

```

**Рис.3.23. – Скрипт для відчинення двері**

Саме це ми додаємо до двері. На цьому скрипті ми у методі OpenDoor та CloseDoor крутимо нашу дверь по осі x. Але нам потрібно буде змінити наш, так званий, центр тяжіння, для того щоб крутіння не відбулося просто по центру. Ми його пересунемо до правої сторони. Після цього розміщуємо його у двірний отвір та підбираємо масштаб.



**Рис.3.24. Відчинення двері**

На цьому скриншоті ми бачимо що двір успішно відчиняється та зачиняється у грі. Ми можемо налаштувати швидкість відчинення та зачинення двері. Або градуси відчинення двері. Дверь відчиняється та зачиняється натискаючи кнопку “E”.

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

Та на фініші я вам покажу перехід до іншої сцени (кінець).

```
taskboxFog.shader Level.cs Compiled-Standard.shader SoundScript.cs MonoBehaviour.cs Reader.cs
Assembly-CSharp
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.SceneManagement;
5
6 public class NextScene : MonoBehaviour
7 {
8     void OnTriggerEnter(Collider MyPlayer)
9     {
10         Debug.Log("Finish");
11         SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
12     }
13 }
14
```

Рис.3.25. – Скрипт для переходу на іншу сцену

Коли гравець заходить у об'єкт йде перехід до наступної сцени. Коли закінчується завантаження сцени ми успішно переходимо до неї.

### 3.8 Створення ігрового меню та переходу до ігрового світу

Для початку нам потрібно зробити задній фон для атмосфери.



Рис.3.26. – Головне меню без інтерфейсу

Ось так саме виглядає головне меню без інтерфейсу. Як ми можемо бачити, лише плоска поверхня, декілька дерев та задній фон, який знаходиться в далечині.

Коли ми його вже додали ми створюємо об'єкт у **Canvas** та кріпимо до нього скрипт. Саме він нам допоможе переходити з кліку на наступну сцену.

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

```

UltraSkyboxFog.shader  Level.cs  Compiled-Standard.shader  SoundsScript.cs  MotionBlur.cs  Reader
Assembly-CSharp
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class Level : MonoBehaviour
7  {
8      public void PlayGame()
9      {
10         SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
11     }
12     public void QuitGame()
13     {
14         Debug.Log("Quit");
15         Application.Quit();
16     }
17 }
18

```

**Рис.3.27. – Скрипт для переходу на наступні сцени**

Додаємо Button до об'єкту та вибираємо. Після цього ми додаємо текст до Button і фантазуємо.

Так як в моєму меню буде програватися музика, я додам ще регулятор звуку. За допомогою цього повзунка ми регулюємо гучність звуку.

Отже, ми маємо меню з кнопками старт та вихід. Маємо регулятор гучності, та маємо опис кнопок для потрібної нам дії.



**Рис.3.28. – Головне меню**

Ось що вийшло взагалом. Мінімалістичне меню з доброю музикою. Та напруженим заднім фоном що задає загадковості гравцю який зайшов пограти у цю гру.

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## РОЗДІЛ 4. ТЕСТУВАННЯ ГРИ ТА КЕРІВНИЦТВО КОРИСТУВАЧА

Після вдалої установки Zip файла потрібно розархівувати файл Horror.exe. Та відкрити його. Після чого гра відкриється у повному екрані.

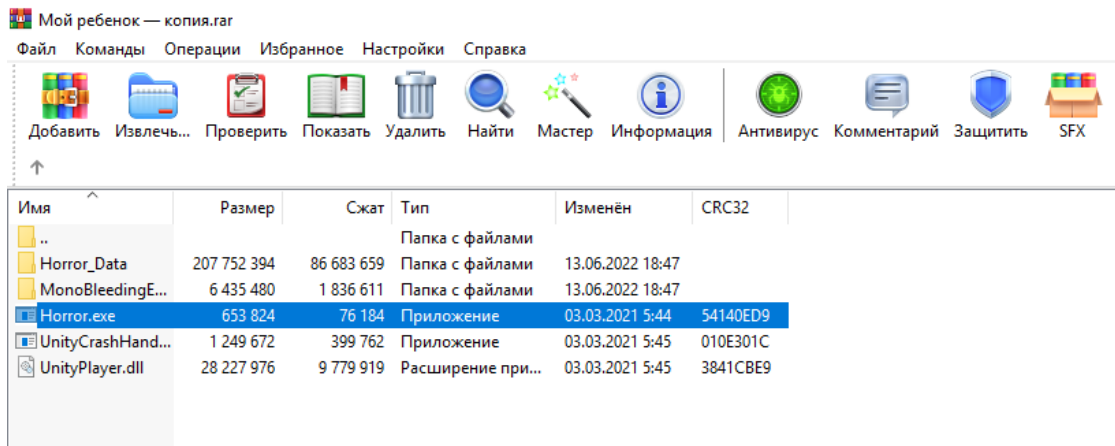


Рис.4.1. Вигляд Zip файлу

Після відчинення гри у вас відкриється головне меню із варіантами вхід та вихід якщо користувач натискає play - гра починається, якщо exit – відбувається вихід з самої гри.

Unity Test Runner — це інструмент, який перевіряє ваш код як у режимі редагування, так і в режимі відтворення, а також на цільових платформах, таких як Standalone , Android або iOS.

Відбулася перевірка усієї програми за якою ми визначили що, відсутні помилки у кодї

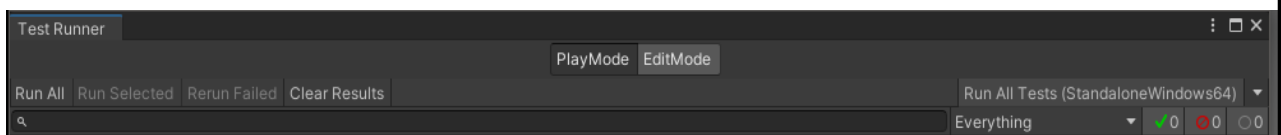


Рис.4.2. Перевірка коду на помилки

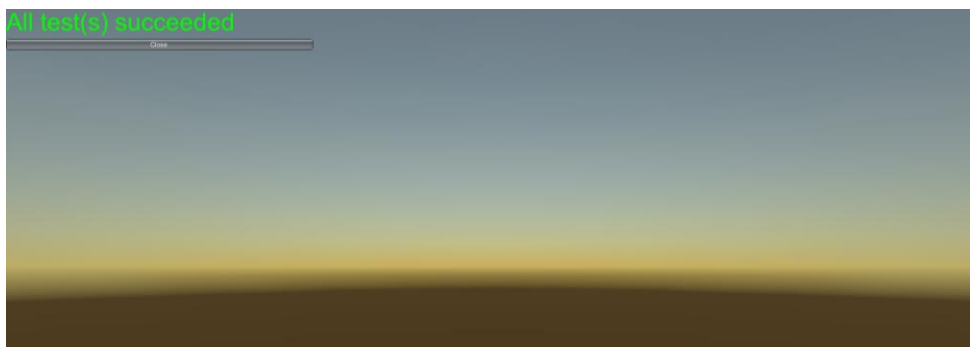
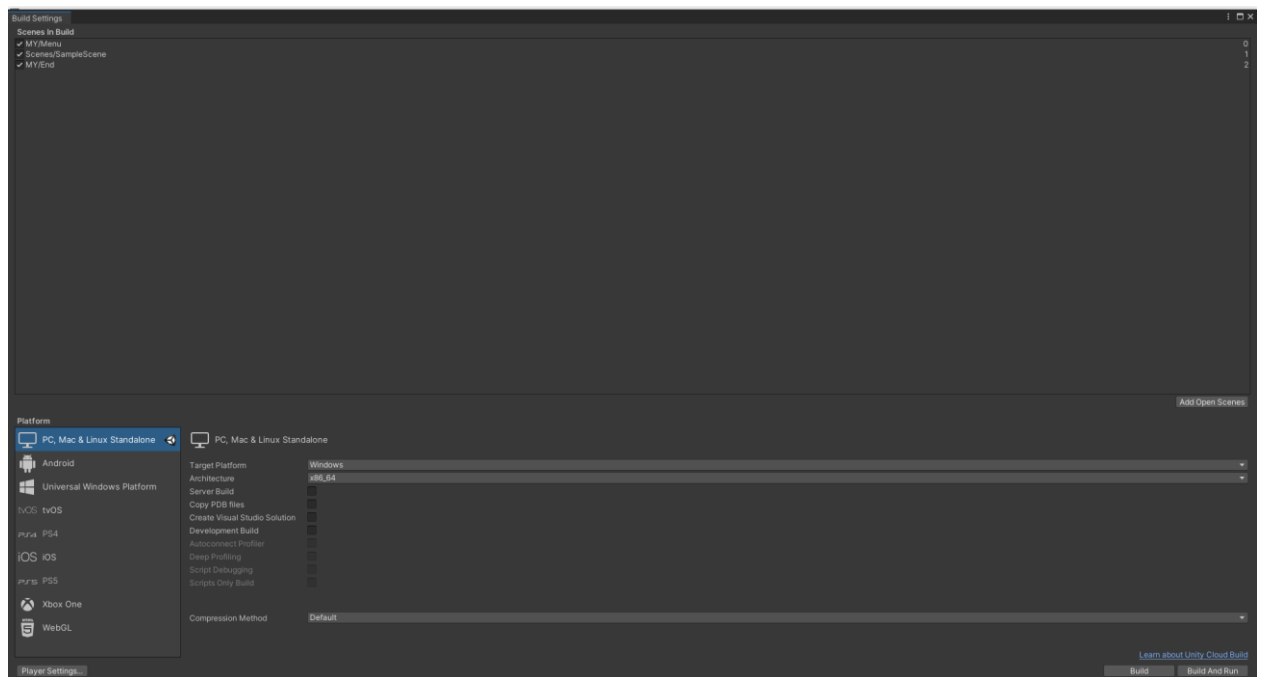


Рис.4.3. Перевірка проекту на помилки

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Доступ до Unity Test Runner можна отримати через Window > Test Runner .

Для того щоб створити наш ігровий додаток в окремому вікні нам потрібно перейти у розділ в unity Files та натиснути Build Settings.



**Рис.4.4. Конструювання гри**

У нас відчинилось меню із налаштуванням, де ми вибираємо те, що нам потрібно. Слід зазначити, кожна зі сцен у списку має свій індекс. Scene 0 – це перша сцена, яка буде завантажена у складання. Якщо ви бажаєте завантажити іншу сцену під час гри, використовуйте у коді метод `Application.LoadLevel()`. Коли я додав більше однієї сцени і хотів змінити її порядок, просто перетягував сцени в списку до тих пір, поки не досягну необхідного порядку.

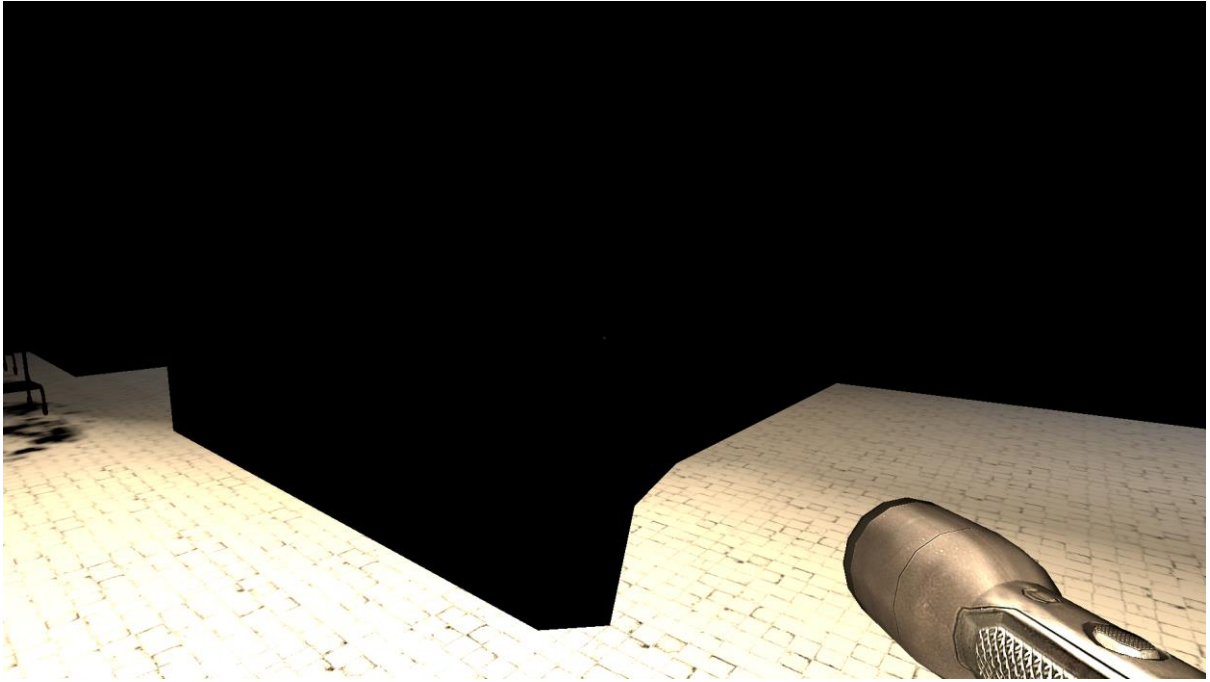
Нам відображаються платформи та інші не менш важливі настрої, які ми можемо змінити, але не всі (платформу ми вибирали на самому початку). Після цього ми натискаємо `Bild and run` та очікуємо час збирання нашої гри.

Перейдемо до тестування гри. Під час тестування з багами у головному меню я не зіштовкнувся усе працювало так як було задумано. Проблеми з'явилися з освітленням у грі та ліхтариком який працював не зрозуміло як. Стіни не було видно, хоча на них попадало світло. Ліхтарик освітляв не напрямом, а як звичайна лампа. Було прийняте рішення видалити ліхтарик та додати в неосвітлених місцях лампи. Також був на екрані видно курсор window. Його потрібно було теж видалити. Вирішив цю проблему я за допомогою строки в методі `void start`.

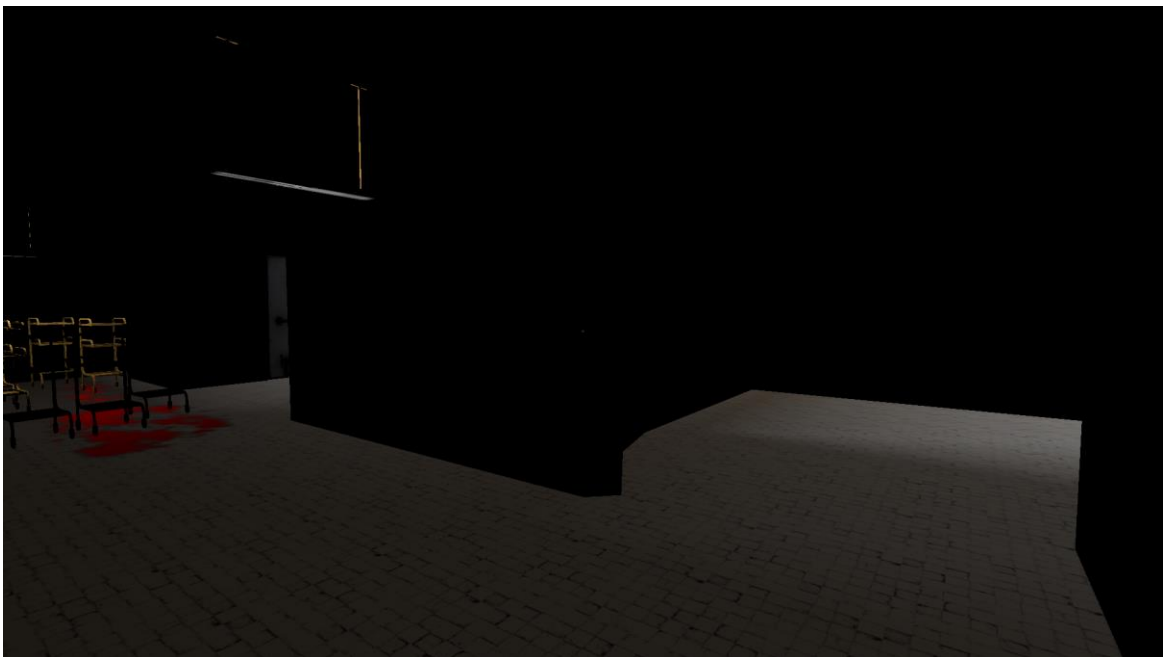
									Арк.
Змін.	Арк.	№ докум.	Підпис	Дата					

*РП 05.04.000.00 ДП*

Cursore.visible=false; Вона додається у canvas.



**Рис.4.5. Освітлення «До»**



**Рис.4.6. Освітлення «Після»**

Отже, після видалення ліхтарика та додавання більшої кількості ламп, на мою думку, відчулися покращення, але присутні у деяких комнатах баги з освітленням які я не зміг вирішити.

Після цього – не було помічено критичних багів які б руйнували гру. Отже ми маємо звук без багів, двері усі відчанняються, переходи до сцен працюють.

За моїм висновком моїй гріне вистачає досвідченої праці з освітленням.

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## 5. ЕКОНОМІЧНА ЧАСТИНА

### 5.1 Резюме

В даному дипломному проекті розроблене:

Ефективність кожного програмного продукту визначається його якістю та ефективністю процесу розробки. Якість ПП визначається наступними складовими: з точки зору користувача; з позиції використання ресурсів; виконання вимог до програмного забезпечення.

Оцінка якості програмного продукту з точки зору користувача визначається необхідним на стадії функціонування розміром оперативної пам'яті ЕОТ, витратами машинного часу, пропускнуою спроможністю каналів передачі даних. Оцінка якості програмного продукту включає визначення трудомісткості і вартості його створення.

### 5.2. Визначення трудомісткості розробки програмного забезпечення

Тривалість розробки програмного продукту залежить від його обсягу, трудомісткості розробки, кваліфікації виконавців, а також планових термінів, визначених умовами ринку. Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тисячах умовних машинних команд програми аналога.

Каталог аналогів

Таблиця 5.1

Найменування ПП	Обсяг функції ПП – $V_0$ , усл. машинних командах.
1. ПП автоматизованих розрахунків	1300 – 8600
2. ПП загальної математики і ПП імітаційного моделювання	1800 – 8800
3. ПП введення інформації	1060 – 5750

У таблиці 5.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт. Для нашого варіанта виділено сірим кольором.

Вибравши аналог ПП, що містить  $V_0$  в умовних машинних командах, трудомісткості визначати на основі табл.5.2.

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

Таблиця.5.2

Обсяг ПП, тис.умов.машинних команд	Норма часу, люд/год
1.00	229
2.00	244
3.00	262

На підставі отриманого значення, по довіднику, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера,  $K_k=0,7\div 0,8$ ):  $T^a = 229 \times 0,8 = 183.2$  (люд/годин).

Трудомісткість програмного продукту визначається по кожному етапу розробки окремо на підставі трудомісткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_{T3} = T^a p \times L_1 \times K_H \quad (4.1)$$

$$T_{TP} = T^a p \times L_2 \times K_H \quad (4.2)$$

$$T_{RP} = T^a p \times L_3 \times K_H \times K_T \quad (4.3)$$

Для розрахунку необхідні наступні коефіцієнти:

$L_i$  – питома вага і-го етапу розробки (див. табл. 5.2.);

$K_H$  – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 5.3.);

$K_T$  – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 5.4.).

Таблиця 5.3.

Значення питомих коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПП.

Код стадії	Ступінь новизни		
	А	Б	В
T3 ( $L_1$ )	0,15	0,12	0,12
TP ( $L_2$ )	0,16	0,15	0,11
RP ( $L_3$ )	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		







## **6. ОХОРОНА ПРАЦІ**

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

Одним з резервів підвищення ефективності виробництва є вдосконалення методів забезпечення безпеки праці, тому що травматизм визначає істотну частину непродуктивних втрат робочого часу, а боротьба з травматизмом, крім гуманістичного спрямування, має чітко виражений економічний аспект.

Безпека праці виступає і як один з факторів, які забезпечують високу продуктивність праці. Доведено, що висока продуктивність праці може бути досягнута тільки в умовах, коли забезпечена її безпека.

В розділі охорона праці дипломного проекту вирішується питання безпеки праці програміста при розробці ним ігрового додатку класичного жанру.

### **1. Аналіз та безпека умов праці працівника на робочому місці**

Забезпечення безпечних і здорових умов праці в значній мірі залежить від правильної оцінки небезпечних та шкідливих виробничих факторів. Однакові по складності зміни в організмі людини можуть бути викликані різними причинами. Це можуть бути фактори виробничого середовища, надмірне фізичне і розумове навантаження, нервово-емоційна напруга, а також різне сполучення цих причин.

Оператори і програмісти зіштовхуються із впливом таких фізично небезпечних і шкідливих виробничих факторів, як підвищений рівень шуму, підвищена температура зовнішнього середовища, відсутність або недостатня освітленість робочої зони, електричний струм, статична електрика тощо. До основних шкідливих факторів при роботі з комп'ютером відносять: тривале сидяче положення, електромагнітне випромінювання, навантаження на зір

На робочому місці програміста повинні бути створені умови для безпечної та високопродуктивної праці.

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

## 2. Розробка заходів з охорони праці

### 6.1 Виробничі приміщення.

Вимоги ДСанПІН 3.3.2.007-98 визначають об'ємно-планувальні рішення будівель та приміщень для роботи з ВДТ. Розміщення робочих місць з ВДТ ЕОМ і ПЕОМ у підвальних приміщеннях, на цокольних поверхах заборонено. Площа на одне робоче місце становить не менше 6,0 м<sup>2</sup>, а об'єм – не менше ніж 20,0 м<sup>3</sup>. У приміщеннях слід щоденно робити вологе прибирання. Вони повинні бути оснащені аптечками першої медичної допомоги. При приміщеннях мають бути обладнанні побутові приміщення для відпочинку.

При кольоровому оформленні виробничих і допоміжних приміщень необхідно враховувати орієнтацію їхніх вікон стосовно частин світу і використовувати гармонійне сполучення кольорів. Для стін і робочих поверхонь використовують мало насичені (основні) кольори, для невеликих помешкань або ділянок, що рідко потрапляють у поле зору працюючих, а також для створення контрастності – кольори середньої насиченості (допоміжні), для маленьких по площі поверхонь – насичені (акценти) – як функціональне фарбування. Стелі у всіх приміщеннях повинні бути білими. Поверхні устаткування в приміщеннях повинні бути матовими або напівматовими, для виключення випадку відблисків світла в очі працюючого, а стіни бути пофарбованими фарбами пастельних тонів.

### 6.2 Освітлення.

Приміщення для роботи з ПК повинні мати природне та штучне освітлення, відповідно до ДБН В.2.5-28-2006. У приміщеннях, призначених для роботи з відео терміналами, доцільно, щоб вікна були орієнтовані на північ або північний захід. На вікнах повинні бути штора або жалюзі, що регулюють рівень освітленості і захищають від прямого влучення сонячних променів на робоче місце.

Для штучного освітлення у приміщенні використовуються люмінесцентні лампи типу ЛБ, які в порівнянні з лампами розжарювання мають ряд істотних переваг: за спектральним складом світла вони близькі до природного світла, мають підвищену світлову віддачу (у 2-5 разів вищу, ніж у ламп розжарювання); мають триваліший термін служби – до 10 тис годин.

### 6.3 Параметри повітря робочої зони.

									Арк.
Змін.	Арк.	№ докум.	Підпис	Дата	РП 05.04.000.00 ДП				

Для підтримки в приміщеннях нормального, що відповідає гігієнічним вимогам складу повітря, видалення з нього шкідливих газів, пилу використовують вентиляцію. Механічна вентиляція (кондиціонери, вентилятори і т.д.) залежно від напрямку руху повітряних потоків, може бути витяжною, припливною і припливно-витяжною. При природній вентиляції (за допомогою вікон) повітря надходить у приміщення і видаляється з нього внаслідок різниці температур і тиску.. Механічна вентиляція забезпечується вентиляторами, що забирають повітря зовні і направляє його до будь-якого робочого місця. або устаткування, а також видаляють забруднене повітря.

У виробничих приміщеннях на робочих місцях мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря – ГОСТ 12.1.005-88, СН 4088-86.

Параметри мікроклімату	Взимку	влітку
Температура, С <sup>0</sup>	22-24	23-25
Відносна вологість, %	40-60	40-60
Швидкість руху повітря, м/с	0,1	0,1-0,2

#### **6.4 Шум і вібрація, ЕМВ.**

При розумовій праці, яка вимагає зосередженості припустимий рівень шуму становить 50дБ. Для зменшення шуму й вібрації в приміщенні устаткування, апарати й прилади встановлюють на спеціальні прокладки, що амортизують. Якщо стіни в приміщенні є джерелами шумоутворення, вони повинні бути облицьовані звуковбирним матеріалом.

У разі надмірного шуму чи вібрації технічного обладнання, роботодавець повинен забезпечити працівників антивібраційними килимками.

Кількісно вплив електромагнітного поля на людину оцінюється величиною поглинутої її тілом електромагнітної енергії, W,Вт, або питомої енергії, що поглинається Wп, Вт/кг. Основні заходи захисту від ЕМВ — це захист часом, захист відстанню, екранування джерел випромінювання, зменшення випромінювання в самому джерелі випромінювання, виділення зон випромінювання , екранування робочих місць, застосування ЗІЗ.

#### **6.5 Організація робочих місць з ПК.**

					<i>РП 05.04.000.00 ДП</i>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		

Роботодавець, який використовує найману працю робітників, повинен забезпечити відповідність їхніх робочих місць комфортним та безпечним умовам. Розмір одного робочого місця має становити не менше 6 квадратних метрів. При необхідності, суміжні робочі місця співробітників, що працюють з комп'ютером, слід розділити перегородками висотою до 2 метрів.

При визначенні достатнього розміру приміщення і робочого місця на одну особу необхідно додатково враховувати шафи, сейфи, тумби або інші предмети меблів чи обладнання, які знаходяться в кімнаті.

На столі працівника можливо розмістити допоміжні для роботи пристрої (принтери, колонки, сканери), а також місця для зберігання документів, за умови, що це не обмежуватиме видимість екрану і не заважатиме працівнику.

Робочий стілець співробітника має бути підйомно-поворотним, легко регульованим за висотою та забезпечувати належну підтримку та зручне положення спини і хребта особи. Щодня необхідно проводити вологе прибирання приміщення, та очищати робоче місце та безпосередньо монітор комп'ютера від запиленості.

Мають бути чітко встановлені перерви для відпочинку працівників (окрім обідньої), як правило, тривалістю 10-15 хвилин раз на годину або дві, в залежності від складності роботи. У будь-якому випадку, роботодавець повинен передбачити такий розпорядок роботи на підприємстві, щоб час неперервної роботи з комп'ютером був не більше ніж 4 години.

### **6.3 Пожежна безпека**

Під пожежною безпекою розуміють систему державних і суспільних заходів, спрямованих на охорону від вогню людей і власності. Пожежна безпека приміщень, що мають електричні мережі, регламентується ГОСТ 12.1.033-81, ГОСТ 12.1.004-85. Робота оператора ЕОМ повинна вестися в приміщенні, що відповідає категорії Д пожежної безпеки (негорючі речовини й матеріали в холодному стані).

Пожежна безпека об'єкта забезпечується:

- Системою запобігання пожежі;

	-	Системою протипожежного захисту;			Арк.
		РП 05.04.000.00 ДП			
Змін.	Арк.	№ докум.	Підпис	Дата	

- Організаційно-технічними заходами.

Всі приміщення повинні бути забезпечені первинними засобами пожежогасіння: пожежним водопостачанням (пожежні крани ПК), пожежні щити з набором пожежного інструменту, вуглекислотними або порошковими вогнегасниками. У випадку виникнення пожежі необхідно відключити електроживлення, викликати по телефону 101 пожежну команду, евакуювати людей із приміщення відповідно до плану евакуації і приступити до ліквідації пожежі.

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

## ***ВИСНОВОК***

Мета цієї дипломної роботи полягає у створенні комп'ютерної гри, для того щоб людина могла себе почувати в ролі головного героя гри. Гра була створена на ігровому двигуні Unity3D.

Для досягнення поставленої мети було сформульовано такі завдання:

1. Ознайомитись із двигуном Unity та розібратися із його механікою праці.
2. Підтянути знання із мовою програмування С#.
3. Спланувати сетинг гри.
4. Створити ігровий мир.
5. Написати скрипти.
6. Здійснити розрахунок економічних показників ефективності створення ігрового додатку.

У першому розділі був огляд ігрової індустрії та перелік ігрових двигунів.

У другому та третьому розділі був проведений аналіз та пред'явлені вимоги до Ігрового додатку, докладно описана кожна механіка праці ігрових об'єктів та створення миру в якому відбувається гра.

У четвертому розділі дипломної роботи розглядався перший щапук гри та пошук перших проблем та багів з якими я зіштовкнувся. Були знайдені та проаналізовані та представлені способи підвищення потоку клієнтів за допомогою мобільного додатка.

У п'ятому розділі дипломної роботи розглядався розрахунок економічних показників ефективності створення ігрового додатку.

У шостому розділі було проаналізовано умови труда та їх вплив на людину. Під час вирішення завдання було вивчено теоретично методи пред'явлення та аналізу вимог до програмного забезпечення. Було розроблено додаток, який відповідає пред'явленим вимогами дипломною роботою, та функціонує згідно з описаними вимогами. Також були закріплені та отримані нові навички роботи в середі розробки гри Unity.

Таким чином, завдання були вирішені в повному обсязі, ціль досягнуто.

					<i>РП 05.04.000.00 ДП</i>	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		

## **СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ ІНФОРМАЦІЇ**

- 1 Шилдт Герберт - C# 4.0: повне керівництво 329с.
- 2 Майк Гейг-Розробка ігор на Unity 2018 за 24 години 10-100 с.
- 3 Скриптинг –[Електронний ресурс]. Режим доступу:  
<https://docs.unity3d.com/ru/530/Manual/ScriptingSection.html>
- 4 ProBuilder[https://www.youtube.com/watch?v=lCdF4O4hkM8&t=1482s&ab\\_channel=KorgiBit](https://www.youtube.com/watch?v=lCdF4O4hkM8&t=1482s&ab_channel=KorgiBit)[Електронний ресурс]. Режим доступу:
- 5 Умови праці -[Електронний ресурс]. Режим доступу:  
<http://referatss.com.ua/work/umovi-praci-na-pidpriemstvi-stan-problemi-ta-zahodi-pokrashhennja/>
- 6 Хокінг Джозеф - Unity у дії. Мультиплатформна технологія на C#. 2-е межд. Видання. -38 с.
- 7 Ферроне Харрісон - Вивчаємо C# через розробку ігор Unity. 5-те видання
- 8 Ассети та ін. <https://assetstore.unity.com/> Інформаційні системи. Визначення інформаційної системи. - 122 с.
- 9 Книга, Філіп Джепікс та Ендрю Троелсен - Мова програмування C# 7 та платформи .NET та .NET Core. 28 с.

					<b>РП 05.04.000.00 ДП</b>	<i>Арк.</i>
<i>Змін.</i>	<i>Арк.</i>	<i>№ докум.</i>	<i>Підпис</i>	<i>Дата</i>		