

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 123 «Комп'ютерна інженерія»

Освітня програма: «Обслуговування комп'ютерних систем і мереж»

Група: 4КС-55

Дипломний проект

**здобувача освіти денної форми навчання
КС.55.22.000.ДП**

***СПАТАРЬ ПАВЛА
КОСТЯНТИНОВИЧА***

**м. Одеса
2022 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНАХТ»

Спеціальність: **123 «Комп'ютерна інженерія»**

Освітня програма: **«Обслуговування комп'ютерних систем і мереж»**

Група: **4КС-55**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Створення ігрового тренажеру
для вивчення роботи цифрових пристроїв

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

Дипломник _____ (Спатарь П.К)

Керівник _____ (Скорнякова О. 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 ДКК № _____

Оцінка ДКК _____

Секретар ДКК _____

ЗМІСТ

ВСТУП	5
1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ	7
1.1. Вивчення існуючих підходів до використання комп'ютерних ігор в освітньому процесі	7
1.2 Аналіз предметної області	9
1.3 Вибір засобів реалізації	12
1.4 МДЕ: Механіка, динаміка та естетика	18
1.5 Розробка програмного коду	20
1.5.1 Створення головного меню	20
1.5.2 Створення меню вибору рівня	24
1.6 Додавання бази даних	25
1.6.1 використання бази даних	26
1.7 Створення рівня	27
1.7.1 Створення ігрових механізмів	28
1.7.2 Створення гравця та кристалу	32
1.7.3 Алгоритм роботи та дій користувача	35
2 ЕКОНОМІЧНИЙ РОЗДІЛ	39
3 ОХОРОНА ПРАЦІ	44
3.1 Вступ	44
3.2 Аналіз та безпека умов праці працівника на робочому місці	44
3.3 Організація робочого місця	45
3.4 Мікроклімат	46
3.5 Освітлення	47
3.6 Електробезпека	48

					КС.55.22.00.ДП ПЗ		
<i>Изм.</i>	<i>Лист</i>	<i>№ докум.</i>	<i>Підпись</i>	<i>Дата</i>			
<i>Розробив</i>		Спатарь П.К.			<i>Лит.</i>	<i>Лист</i>	<i>Листов</i>
<i>Перевірів</i>		Скорнякова О.В.			3	2	
<i>Рецензент</i>					ВСП ОТФК ОНТУ ΔКС-55		
<i>Н. Контр.</i>		Петрашова В.І.					
<i>Затвердив</i>		Скорнякова О.В.					

3.7 Пожежна безпека	48
ВИСНОВКИ	51
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	52
ДОДАТОК А	

					КС.55.22.00.ДП ПЗ	Лист
						4
Изм.	Лист	№ докум.	Підпись	Дата		

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та Ш
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма «Обслуговування комп'ютерних систем і мереж»

ЗАТВЕРДЖУЮ:

Заст. дир. з НВР _____

“ _____ ” _____ 2022 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти

Спатарь Павло Костянтинович

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи): Створення ігрового тренажера для вивчення роботи цифрових пристроїв

затверджена наказом по коледжу від “30” січня 2021 р. № 306-А2-ОД

2. Термін здачі закінченого проекту (роботи) _____

3. Вихідні данні до проекту (роботи): 1. Міжплатформенне середовище розробки комп'ютерних ігор Unity;

2. Microsoft visual studio community edition; 3. Вбудована СУБД SQLite;

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

ВСТУП.

1. ТЕХНОЛОГІЧНИЙ РОЗДІЛ

2. ЕКОНОМІЧНИЙ РОЗДІЛ

3. ОХОРОНА ПРАЦІ

4. ВИСНОВКИ

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

Презентаційний матеріал, кількість слайдів 10

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

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Вступ, 1	Скорнякова О.В.		
2	Копайгородська Т.Г.		
3	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		

7. Дата видачі завдання _____

Керівник

(підпис)

Завдання прийняв до виконання

(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1.	Вступ.	5.04.2022	
2.	Технологічний розділ. Вивчення існуючих підходів до використання комп'ютерних ігор в освітньому процесі.	8.04.2022	
3.	Технологічний розділ. Аналіз предметної області.	11.04.2022	
4.	Технологічний розділ. Вибір засобів реалізації.	21.04.2022	
5.	Технологічний розділ. МДЕ: Механіка, динаміка та естетика.	29.04.2022	
6.	Технологічний розділ. Розробка програмного коду.	7.05.2022	
7.	Технологічний розділ. Додавання бази даних	11.05.2022	
8.	Технологічний розділ. Створення рівня гри	17.05.2022	
9.	Економічна частина.	20.05.2022	
10.	Охорона праці	1.06.2022	

Дипломник

(підпис)

Керівник

(підпис)

ВСТУП

Гра - це діяльність або вид спорту, зазвичай вимагає навичок, знань або випадку, в якому ви слідуєте фіксованим правилам і намагаєтеся перемогти противника або вирішити головоломку .

Хоч і комп'ютерні ігри сьогодні можна знайти в будинках по всьому світу, насправді вони зародилися в дослідницьких лабораторіях вчених.

Наприклад, в 1952 році британським професором А. С. Дугласом було створено ОХО, також відоме як хрестики-нулики, в рамках своєї докторської дисертації в Кембриджському університеті. А в 1958 році Вільям Хігінботам створив "теніс для двох" на великому аналоговому комп'ютері та підключеному екрані осцилографа для щорічного Дня відвідувачів Брукхейвенської національної лабораторії в Аптоне, штат Нью-Йорк.

У 1962 році Стів Рассел з Массачусетського технологічного інституту винайшов Spacewar!, комп'ютерну відеогру для космічних боїв для PDP-1 (Programmed Data Processor-1), в той час ультрасучасного комп'ютера, який в основному використовувався в університетах. Це була перша відеогра, в яку можна було грати на декількох комп'ютерах.

Метою дипломного проекту є створення ігрового тренажера для вивчення роботи цифрових пристроїв. Сьогодні все більш широкого поширення набувають нові методи прийому і передачі знань. ігри стали застосовуватися не тільки у формальному, але і в неформальному навчанні, торгівлі, сфері охорони здоров'я та армії.

У проекті буде розглянуто створення ігрового тренажера за допомогою міжплатформенного середовища розробки комп'ютерних ігор – Unity.

В роботі розглянуті такі питання:

- вивчення існуючих підходів до використання комп'ютерних ігор в освітньому процесі;
- вибір платформи і засобів реалізації ігрового тренажера
- реалізація ігрового навчального тренажера

					КС.55.22.00.ДП ПЗ	Аркуш
						5
Зм.	Аркуш	№ докум.	Підпис	Дата		

- написання скриптів ігрових об'єктів мовою С#
- визначено алгоритм дій користувача
- розглянуто питання економічної доцільності розробки;
- розглянуто питання охорони праці.

					КС.55.22.00.ДП ПЗ	Аркуш
						6
Зм.	Аркуш	№ докум.	Підпис	Дата		

1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

1.1 Вивчення існуючих підходів до використання комп'ютерних ігор в освітньому процесі

У наш час ігри перестали бути лише розвагою і дозвіллям. Сьогодні їх використовують як унікальний інструмент для навчання всіляких навичок. Раніше ігри в освітньому процесі не були настільки популярні, але в даний час вони набувають все більшого поширення, оскільки широко використовуються для розвитку навичок і популяризації сфер діяльності людини, полегшення роботи вчителів і викладачів.

Аналізуючи застосування комп'ютерних ігор, можна відзначити наступний феномен. Сьогодні Звичайний користувач сучасних медіатехнологій починає занурюватися у віртуальний ігровий світ не тільки для розваги, але і для реалізації своїх ідей у віртуальному світі шляхом застосування різних програмноапаратних засобів. Крім того, він використовує комп'ютерні ігри для самонавчання (наприклад, стрільба, попередження на реакцію, розвиток аналітичного мислення, вивчення мов і т.д.). Все це може забезпечити позитивні результати в сфері освіти, особливо якщо для цього будуть використані освітні віртуальні ресурси (віртуальна лабораторія, віртуальний клас і т.д.). Перевагою таких ігрових форм є те, що можна мотивувати її учасника (учня) на успішне проходження гри, заохочуючи його ігровими бонусами, очками і досвідом.

Все це буде підігрівати інтерес гравця у вирішенні різних ситуаційних завдань з максимально ефективним результатом, який відповідним чином відіб'ється на його успішності. Практично будь-яку відеогру можна розглядати як серйозну. Це залежить від актуальності використання гри і сприйняття гравця. Основним завданням серйозних ігор є застосування нових ігрових технологій в тренінгових або освітніх цілях.

Віртуальні тури, лабораторії, тренажери та Симулятори активно використовуються у вищих професійних установах як в нашій країні, так і за кордоном. Застосування авіаційних і автомобільних симуляторів економічно

					КС.55.22.00.ДП ПЗ	Аркуш
						7
Зм.	Аркуш	№ докум.	Підпис	Дата		

вигідно за рахунок того, що вони виконують функції тренажерів, замінюючи літальні апарати і автомобілі відповідно. Завдяки тому що при неправильних діях учня відбувається автоматична зупинка або блокування віртуального симулятора, не завдається шкода оточуючим і не відбувається поломка найдорожчого обладнання. Використання віртуальних лабораторій дозволяє проводити різні дослідження (наприклад, з хімії або фізики), не застосовуючи дорогого обладнання і не піддаючи гравця (учня) технологічним небезпекам, які можуть виникнути в ході проведених експериментів.

Як підсумок в даний час прогрес використання комп'ютерних ігор досить відчутний. Це дає можливість освітнім технологіям слідувати в ногу з технологічними інноваціями, що дозволяють за рахунок яскравої графіки, занурення в атмосферу гри створювати простір, в якому гармонійно поєднуються навчання і розвага.

1.2 Аналіз предметної області

Розглянемо існуючі аналоги. Наприклад, гра під назвою «логіка». Суть гри в тому, щоб крок за кроком передавати кристал з верхньої платформи на нижню. Подаючи 1 або 0 на вхід механізмів в правій частині схеми можна виставляти платформи на шляху кристала якщо на вхід механізму подати 0 платформа забирається.

Щоб управляти механізмами використовуються перемикачі в лівій частині схеми. Стан яких змінюється натисканням мишки. Якщо перемикач включений на логічні схеми надходить 1 якщо вимкнений 0. Логічні схеми перетворюють вхідні сигнали за такими правилами:

- схема **не** інвертує вхідний сигнал (0 замість 1 і 1 замість 0)
- схема **і** 1 на виході тільки якщо на всіх входах 1
- схема **або** якщо хоча-б на одному вході 1 то на виході буде 1
- схема **виключає або** 1 тільки якщо 1 тільки на одному вході (1 0 або 0 1)

					КС.55.22.00.ДП ПЗ	Аркуш
						8
Зм.	Аркуш	№ докум.	Підпис	Дата		

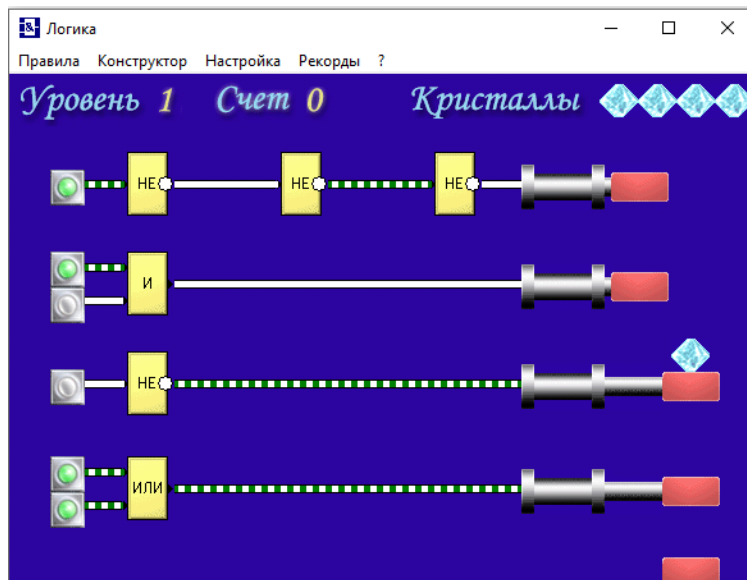


Рисунок 1.1 – Гра «Логика»

Якщо кристал передати через більш ніж один проміжок за раз він розіб'ється почавши рівень заного. Всього дається 5 кристалів на гру. Якщо їх всіх розбити гра закінчиться.

Гра while True: learn () - це головоломка/симулятор, присвячена: машинному навчанню, неймережам, штучному інтелекту, великим даним, а також ІІІ.

Гра ставить гравця на місце фахівця з неймереж і машинного навчання, але потім ви дізнаєтеся про те що ваш кіт справляється з роботою краще. І вам доводиться вирішувати головоломки, щоб сконструювати перекладач з котячого.

Гра заснована на реально застосовуваних принципах машинного навчання: від незручних експертних систем до могутніх згорткових неймереж, здатних передбачати майбутнє.

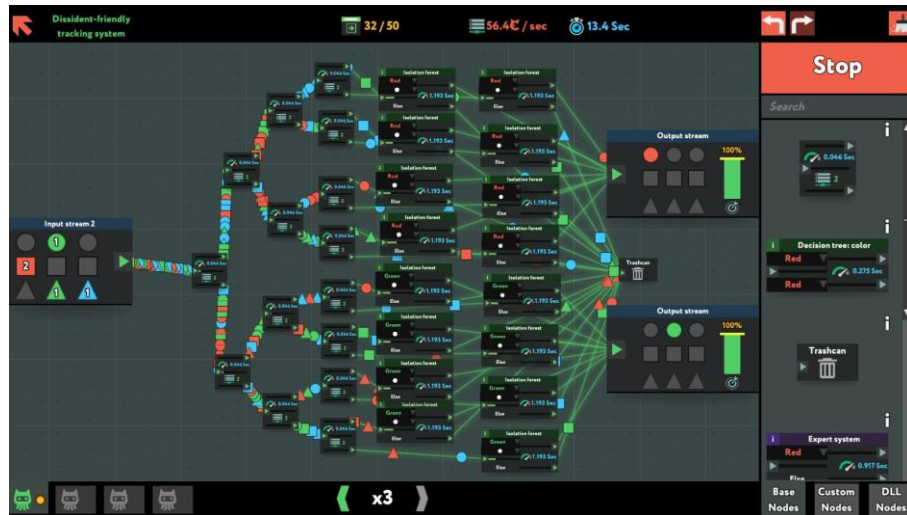


Рисунок 1.2 – Гра «while True: learn()»

Minecraft: Education Edition-це ігрова навчальна платформа, яка допомагає розвивати креативність, навички вирішення завдань і колективної роботи шляхом занурення в цифрове середовище. Педагоги в усьому світі використовують Minecraft: Education Edition для підвищення інтересу учнів до різних предметів і для наочного представлення абстрактних понять.

Основною перевагою Minecraft Education Edition є створення віртуальних аудиторій, які дозволяють учневі «пережити» події, що відбуваються або взяти участь у вкрай інтерактивному уроці. Надана грою свобода захоплює гравця і дозволяє вчителю відтворити у віртуальному просторі безліч різних видів активностей, наприклад:

- Провести віртуальну екскурсію.
- Зробити квест.
- Провести історичну реконструкцію.
- Розіграти сценку або спектакль.
- Провести творчий конкурс (як правило, Minecraft тут виступає в якості своєрідного «воксельного графічного редактора»).
- Читання лекцій.
- Практичні та лабораторні заняття.

					КС.55.22.00.ДП ПЗ	Аркуш
						10
Зм.	Аркуш	№ докум.	Підпис	Дата		

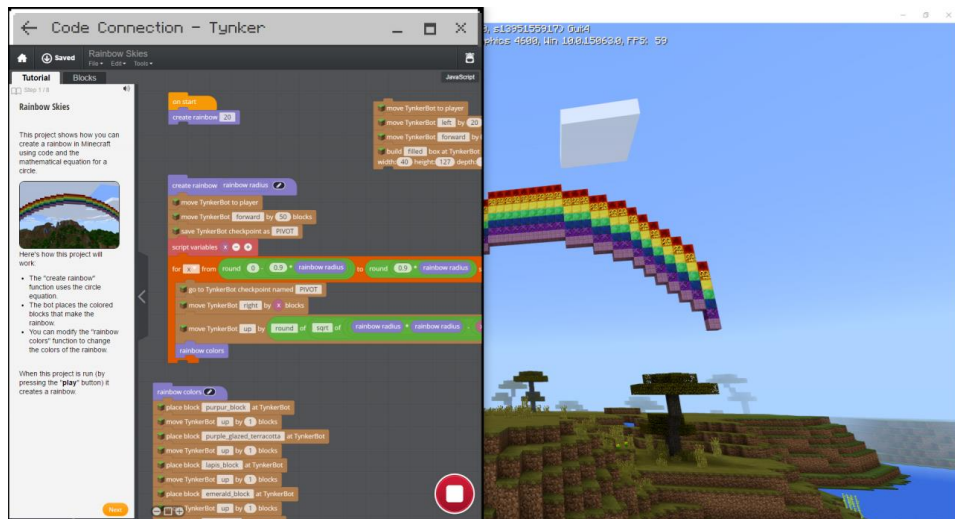


Рисунок 1.3 – Гра «Minecraft Education Edition»

1.3 Вибір засобів реалізації

Unity

Ігрові движки надають безліч функцій, які корисні в різних іграх. Гра, реалізована з використанням певного движка, отримає всі ці функції, додавши при цьому Користувальницькі художні ресурси і ігровий код, специфічний для цієї гри. У Unity є моделювання фізики, карти нормалей, оклюзія навколишнього простору екрану (SSAO), динамічні тіні... і цей список можна продовжувати. Багато ігрових движків можуть похвалитися такими функціями, але Unity володіє двома основними перевагами перед аналогічними передовими інструментами розробки ігор: надзвичайно продуктивний візуальний робочий процес і високий ступінь кроссплатформенної підтримки.

Візуальний робочий процес являє собою досить унікальний дизайн, що відрізняється від більшості інших середовищ розробки ігор. У той час як інші інструменти розробки ігор часто являють собою складну мішанину розрізнених частин, які необхідно контролювати, або, можливо, бібліотеку, яка вимагає від вас налаштування власного інтегрованого середовища розробки (IDE), ланцюжок збірки і тому подібного, робочий процес в Unity прив'язаний до ретельно продуманого візуального редактора. Редактор використовується для компоновання сцен у вашій грі і для їх об'єднання художніх ресурсів і коду в

					КС.55.22.00.ДП ПЗ	Аркуш
						11
Зм.	Аркуш	№ докум.	Підпис	Дата		

інтерактивні об'єкти. Принадність цього редактора в тому, що він дозволяє швидко і ефективно створювати ігри професійної якості, надаючи розробникам інструменти для неймовірно продуктивної роботи, в той же час використовуючи великий список новітніх технологій в відеоіграх.

Редактор особливо корисний для швидкої ітерації, відточування гри за допомогою циклів Прототипування і тестування. Ви можете налаштовувати об'єкти в редакторі і переміщати предмети навіть під час гри. Крім того, Unity дозволяє налаштовувати сам редактор, написавши сценарії, які додають нові функції і меню в інтерфейс.

Крім значних переваг редактора в продуктивності, інший головною сильною стороною набору інструментів Unity є високий ступінь крос-платформної підтримки. Unity мультиплатформенна не тільки з точки зору цілей розгортання (ви можете виконувати розгортання на ПК, в Інтернеті, на мобільних пристроях або консолях), але це також мультиплатформенно з точки зору інструментів розробки (ви можете розробити гру на Windows або macOS). Ця незалежна від платформи природа багато в чому пояснюється тим, що Unity починалася як Програмне забезпечення тільки для Mac і пізніше було портовано на Windows. Перша версія була випущена в 2005 році, але зараз Unity випустила свою п'яту основну версію (з великою кількістю часто випускаються незначних оновлень). Спочатку Unity підтримувала лише Mac як для розробки, так і для розгортання, але протягом декількох місяців Unity була оновлена для роботи і в Windows. Наступні версії поступово додавали нові платформи розгортання, такі як кросплатформенний веб-плеєр у 2006 році, iPhone в 2008 році, Android в 2010 році і навіть ігрові консолі, такі як Xbox і PlayStation. Зовсім недавно вони додали розгортання в WebGL, нову платформу для 3d-графіки в веб-браузерах, і є навіть підтримка VR-платформ, таких як Oculus Rift і Vive. Деякі ігрові движки підтримують стільки цілей розгортання, як Unity, і ні один з них не робить розгортання на декількох платформах таким простим.

					КС.55.22.00.ДП ПЗ	Аркуш
						12
Зм.	Аркуш	№ докум.	Підпис	Дата		

На додаток до цих основних переваг, третя, більш тонка перевага виходить від модульної системи компонентів, використовуваної для створення ігрових об'єктів. У компонентній системі, компоненти являють собою комбіновані пакети функціональних можливостей, а об'єкти створюються як набір компонентів, а не як сувора ієрархія КЛАСІВ.

Компонентна система - це інший (і зазвичай більш гнучкий) підхід до об'єктно-орієнтованого програмування (ООП), де ігрові об'єкти створюються за допомогою композиції, а не Успадкування.

Одна з найбільших переваг Unity 3D-вбудована аналітика; ви можете знайти його в інструменті. Це дозволяє програмістам дізнатися більше про продуктивність гри, створювати звіти, використовуючи інформацію про гру. Іншими словами, це дозволяє вам дізнатися більше про гру і внести потрібні зміни в потрібний момент, щоб зробити її краще для користувачів.

Одне з сильних місць Unity-скрипти на JS і C#. Вам не потрібно вивчати вбудовану мову скриптування з його логікою і обмеженнями — ви просто берете будь-який з цих двох мов програмування і пишете що вам потрібно. Для чого потрібна мова C#.

На скрипти можна повісити що завгодно: поява ворогів з-за рогу, вміст скрині зі скарбами або відповіді другорядного героя. Якщо ви вмієте програмувати, то зможете зробити настільки складну і цікаву гру, наскільки захочете самі.

В цілому, Unity 3D-це комплексний пакет, який спрощує, прискорює і робить розробку ігор більш ефективною. В даний час це один з кращих інструментів для створення ігор малого та середнього розміру. З позитивного боку, його також дуже легко освоїти і він має доступну цінову політику.

Наступне питання - це вибір мови програмування для реалізації завдання дипломного проекту.

У світі існує багато різних мов програмування, кожен з яких має свої переваги і недоліки. Ось деякі особливості, якими володіє C#:

- компільована мова;

					КС.55.22.00.ДП ПЗ	Аркуш
						13
Зм.	Аркуш	№ докум.	Підпис	Дата		

- керований код;
- сувора підтримка типів;
- заснований на функціях;
- об'єктно-орієнтований.

Зазвичай люди пишуть комп'ютерні програми на мові, незрозумілому комп'ютера. Насправді всі комп'ютерні мікропроцесори, наявні на ринку, використовують свої, зрозумілі їм набори простих команд, відомі як машинний мову.

Команди на цій мові виконуються мікропроцесором дуже, дуже швидко, але для людини ця мова надзвичайно складний. Наприклад, рядок на машинній мові: 00001 01100 10000 11110 безумовно має якесь значення для відповідного мікропроцесора, але майже нічого не означає для людини. У машинному коді використовуються тільки два символи - 0 і 1. Тому всі складні типи даних - числа, букви і т. д. - повинні перетворюватися в послідовності бітів (тобто нулі і одиниці).

Щоб людям було легше писати програми, були придумані зрозумілі людині мови програмування-їх іноді називають мовами розробки. Мова розробки можна вважати проміжним, що виконує роль посередника між людиною і комп'ютером. Мови розробки, такі як C#, мають логічну структуру і легко інтерпретуються комп'ютером, але при цьому досить близькі до писемним людським мовам, щоб програмісти могли легко читати і розуміти програми, написані на них.

Мови розробки також діляться на компільовані, такі як BASIC, C++, C# і Java, і інтерпретовані, такі як JavaScript, Perl, PHP і Python.

В інтерпретованих мовах розробка і виконання коду включає два етапи:

- програміст пише програму;
- потім кожен раз, коли хтось із гравців грає в гру, код на мові розробки перекладається на машинний мову прямо під час виконання.

					КС.55.22.00.ДП ПЗ	Аркуш
						14
Зм.	Аркуш	№ докум.	Підпис	Дата		



Рисунок 1.4 – Мови програмування

Інтерпретовані мови мають високу переносимість коду, тому що код на мові розробки може інтерпритуватися в машинні команди для конкретного комп'ютера, на якому він виконується. Однак за таку гнучкість доводиться платити зниженням швидкості виконання через необхідність інтерпретувати код на мові розробки, не оптимізованому для пристрою, на якому він виконується, а також по ряду інших причин. Так як інтерпретований код повинен виконуватися на всіх оптимізація мови для конкретного пристрою просто неможлива.

У компільованій мові, такій як C#, процес програмування включає три окремих етапів:

- програміст пише програму на мові розробки, такій як C#;
- компілятор перетворює код на мові розробки в код на машинній мові для комп'ютера конкретного типу;
- комп'ютер виконує скомпільовану програму.

Тут додається проміжний етап компіляції, в ході якого код на мові розробки перетворюється в виконуваний код (тобто в додаток), який комп'ютер може виконувати безпосередньо, без проміжної інтерпретації. Оскільки компілятор має повне уявлення про програму і платформу, на якій вона повинна виконуватися, він може застосовувати безліч різних оптимізацій. В іграх ці оптимізації дозволяють отримати більш високу частоту кадрів, використовувати більш детальну графіку і забезпечити більш швидку реальну дію на дії користувача. Більшість високобюджетних ігор пишеться на

компільованій мові, що дає високу швидкість за рахунок оптимізації, але це також означає, що для кожної платформи доводиться компілювати різний виконуваний код.

Більш традиційні компільовані мови, такі як BASIC, C++ і Objective-C, вимагають від програміста явно управляти розподілом пам'яті комп'ютера, ви потребуючи його вручну виділяти і звільняти пам'ять, коли потрібно створити або нищити змінну. У таких мовах якщо програміст вручну не звільнить раніше виділений фрагмент ОЗУ, в його програмі виникне «витік пам'яті». Він буде продовжувати виділяти пам'ять в ОЗУ, поки не вичерпає її всю, що викличе збій в роботі комп'ютера.

Оскільки, C# - керована мова, виділення і звільнення пам'яті в ньому виконується автоматично. У керованому коді все ще можна створити витік пам'яті, але зробити це ненавмисно набагато складніше.

Мова C# є строго типізованою мовою: це означає, що будь-яка спроба передати параметр неправильного типу в якості аргументу або привласнити значення змінної, яка не може бути неявно перетворена, викличе помилку компіляції. Це дозволяє уникнути багатьох помилок, які відбуваються тільки під час виконання в інших мовах.

Об'єктно-орієнтоване програмування - ООП) - це модель програмування, в якій програми організовані навколо об'єктів і даних, а не дій і логіки.

ООП дозволяє розкласти проблему на ряд сутностей, званих об'єктами, а потім будувати дані і функції навколо цих об'єктів.

Клас-це ядро будь-якої сучасної об'єктно-орієнтованої мови програмування, такої як C#. У мовах ООП обов'язково створення класу для представлення даних. Клас-це план об'єкта, який містить змінні для зберігання даних і функції для виконання операцій з даними. Клас не займає місця в пам'яті і, отже, є тільки логічним поданням даних.

Ігрологія, або людологія, - так називається наука, що вивчає ігри та проектування ігор. За останнє десятиліття екологи запропонували різні способи

					КС.55.22.00.ДП ПЗ	Аркуш
						16
Зм.	Аркуш	№ докум.	Підпис	Дата		

аналізу ігор, щоб спростити розуміння та обговорення структури, основних елементів ігор та їх впливу на гравців і суспільство в цілому.

1.4 МДЕ: Механіка, динаміка та естетика

Вперше запропонований на конференції Game Developers Conference в 2001 році і формалізований у 2004 році, у статті " MDA: a Formal Approach to Game Design and Game Research " 1, мде є найбільш широко використовуваним методом аналізу в геології. Ключовими елементами мде є визначення хутра Ніки, динаміки та естетики; розуміння різних точок зору, з яких дизайнери і гравці розглядають гру; і припущення, що дизайнери повинні спочатку підходити до створення гри з позиції естетики, а потім звертатися до динаміки і механіці, що генерує цю естетику.

Механіка: конкретні компоненти гри на рівні представлення даних і алгоритмів

Динаміка: поведінка компонентів механіки під час виконання у відповідь на введення гравця і виведення один одного.

Естетика: бажані емоційні реакції, що виникають у гравця, коли він взаємодіє з ігровою системою.

Технологія ітераційної розробки характеризує життєвий цикл проекту програмного забезпечення, що включає кілька ітерацій. Ітерація складається з приблизно послідовного набору завдань, обраних з усіх без винятку розділів розробки програмного забезпечення.

- *Аналіз:* етап аналізу допомагає зрозуміти, де ви знаходитесь і куди прагнете. Ви повинні чітко розуміти проблему, яку намагаєтеся вирішити (або можливість, яку намагаєтеся використовувати) в ході проектування. Ви повинні також розуміти, які ресурси зможете привнести в проект і скільки часу ви зможете виділити на проектування
- *Проектування:* тепер, маючи більш чітке уявлення про те, де ви находитесь і до чого прагнете, створіть проект, який вирішить вашу

					КС.55.22.00.ДП ПЗ	Аркуш
						17
Зм.	Аркуш	№ докум.	Підпис	Дата		

проблему, с використанням доступних вам ресурсів. Цей етап починається з мозкового штурму і завершується конкретним планом реалізації

- *Реалізація:* Реалізація лише невеликої частини гри перед
- тестування часто допомагає зосередити увагу на більш вузькому колі проблем, ніж реалізація великими фрагментами. Закінчивши реалізацію, можна запустити пробну гру
- *Тестування:* Тестування допомагає з'ясувати який ефект викликають проєктовані вами ігрові механіки. тестування на самих ранніх етапах, дозволяє поки це можливо все поміняти і повернутися на правильний шлях.



Рисунок 1.5 – Ітераційна розробка

1.5. Розробка програмного коду

1.5.1 Створення головного меню

На вкладці Scene ми робимо наш рівень гри. Тут можна бачити, як виглядає світ гри і переміщати об'єкти у сцені. Також на сцені вже спочатку додана камера. Приклад показаний на рисунку 1.4.

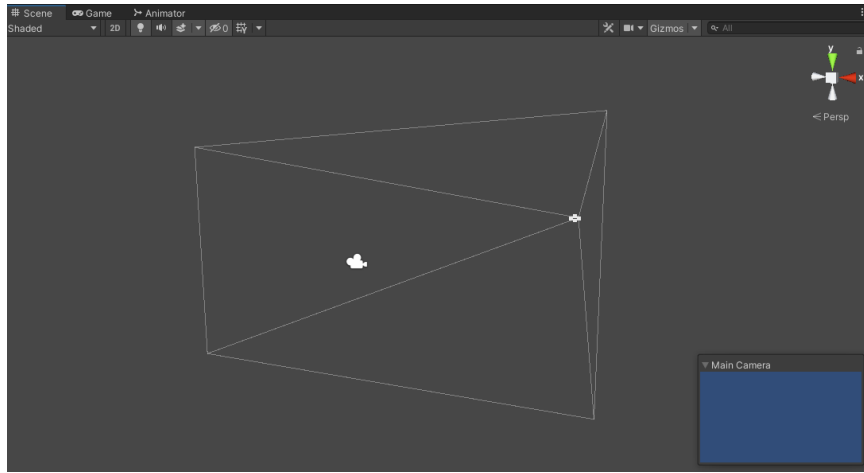


Рисунок 1.6 - Робочий простір Unity

Для початку зробимо бекграунд використовуючи Tilemap.

Tilemap дозволяє швидко створювати і компоувати 2d рівні за допомогою комбінації спрайтів і контролювати такі властивості як порядок шарів і тайлмапніе коллайдери. Інтерфейс схожий на paint, де ми можемо малювати плитку на полотні, використовуючи будь-які доступні зображення, які ми завантажуюємо в проект.

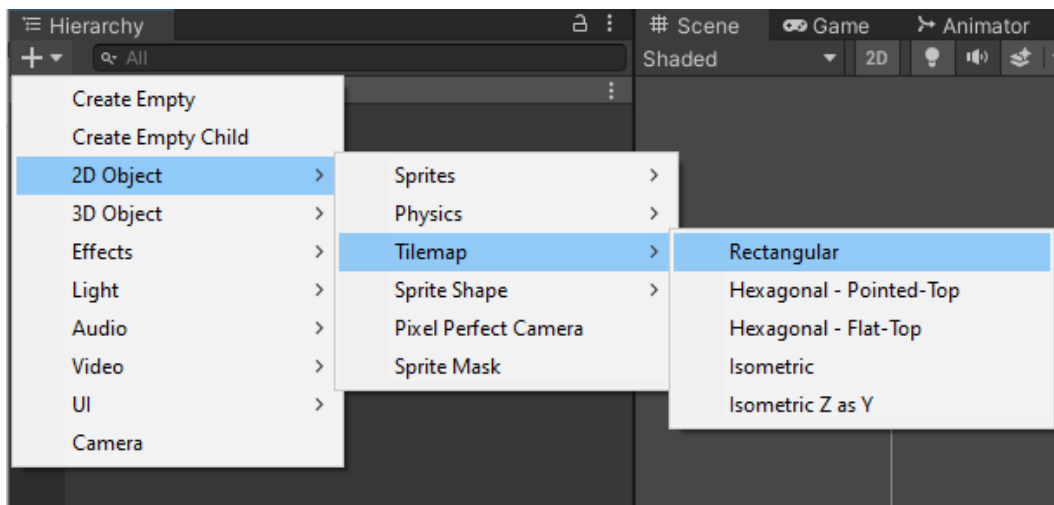


Рисунок 1.7 – Додання Tilemap

Тепер мені потрібна Tile Palette, яку я міг би використовувати для малювання на тайлмапі. Для цього в моєму проекті повинен бути встановлений пакет *2D Tilemap Editor*. У вікні Tile Palette я вибираю "Create New Palette". Даю палітрі ім'я і зберігаю її.

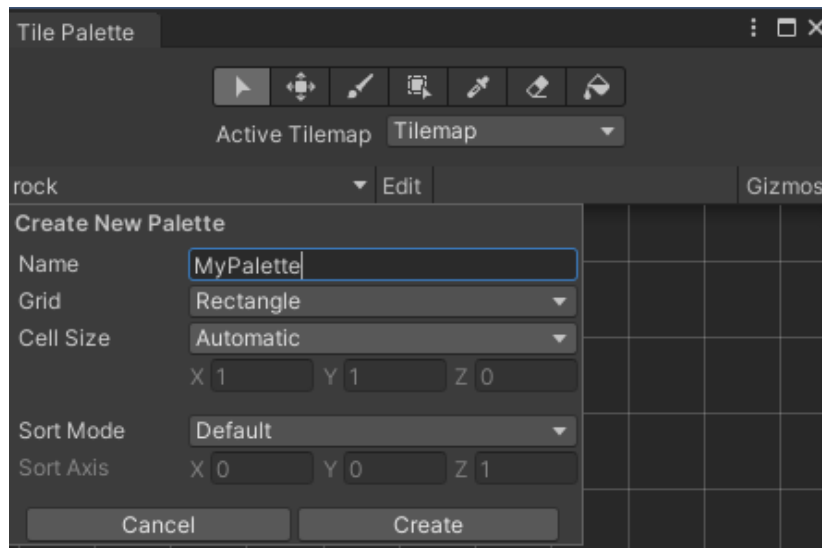


Рисунок 1.8 – Створення TilePalette

Тепер, коли у мене є палітра плиток, її потрібно заповнити плитками для малювання. Ці плитки зроблені з спрайтів.

У мене є підготовлений аркуш спрайтів, який потрібно нарізати за допомогою редактора спрайтів і перетягнути у вікно Tile Palette.

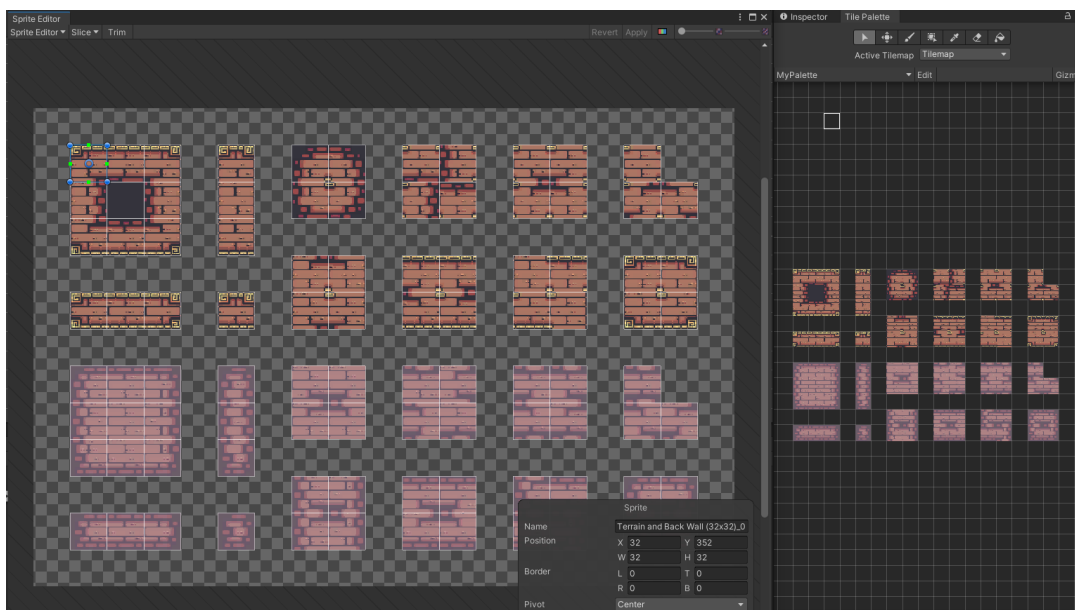


Рисунок 1.9 – Налаштування TilePalette

Тепер, коли все готово, можна приступати до фарбування.

					КС.55.22.00.ДП ПЗ	Аркуш
Зм.	Аркуш	№ докум.	Підпис	Дата		20

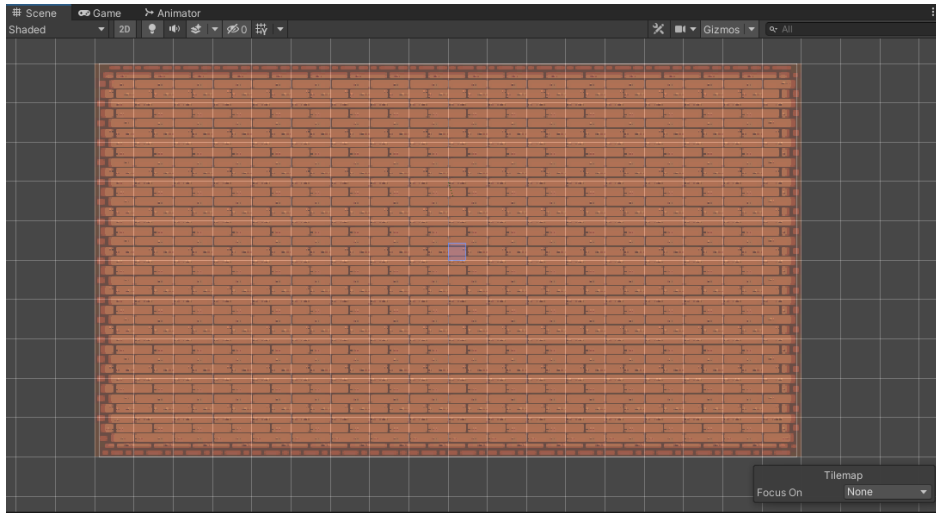


Рисунок 1.10 – Background

Наступний етап-це створення меню.

Засоби відображення абстрактних взаємодій називають призначеним для користувача інтерфейсом (UI) або графічним інтерфейсом користувача (GUI). Хоча UI потрібно будь-якому програмному забезпеченню, тому що користувач просто не зможе користуватися додатком без нього, GUI в іграх використовується дещо по-іншому. У грі текст і кнопки накладаються на ігровий простір за допомогою проєкційного дисплея.

Проєкційним дисплеєм називається індикатор для виведення важливої інформації безпосередньо під час гри. Його прообразом став індикатор на лобовому склі, який використовувався спочатку у військовій авіації і дозволяв пілотам бачити навігаційну і спеціальну інформацію, не відволікаючись на приладову панель.

У додатку використовується метод `OnGUI ()`. Подібно до класів `Start ()` і `Update ()`, всі представники класу `MonoBehaviour` автоматично відповідають на метод `OnGUI ()`. Він запускається в кожному кадрі після візуалізації двовимірної сцени, надаючи місце для команд малювання GUI.

Створюємо об'єкт `Canvas`. На `Canvas` створюю об'єкт `Panel` називаємо його `MainMenu`. Він буде відповідати за наше меню. Створюємо на цьому об'єкті кнопки нашого меню, використовуючи вкладку `Button`. Створюємо

кнопки головного меню - "play", "options" і "exit". Приклад наведено на малюнках .

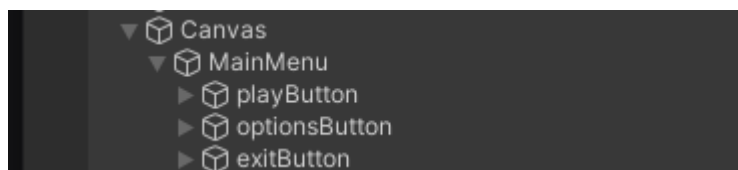


Рисунок 1.11 – Створені кнопки у вікні Hierarchy

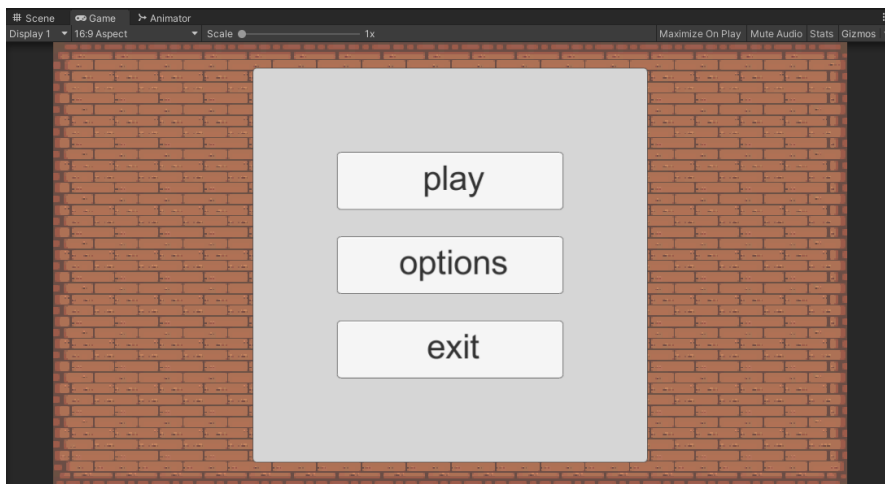


Рисунок 1.12 – Прототип головного меню

1.11
Додаю підготовлені спрайти на кнопки. Приклад наведено на рисунку

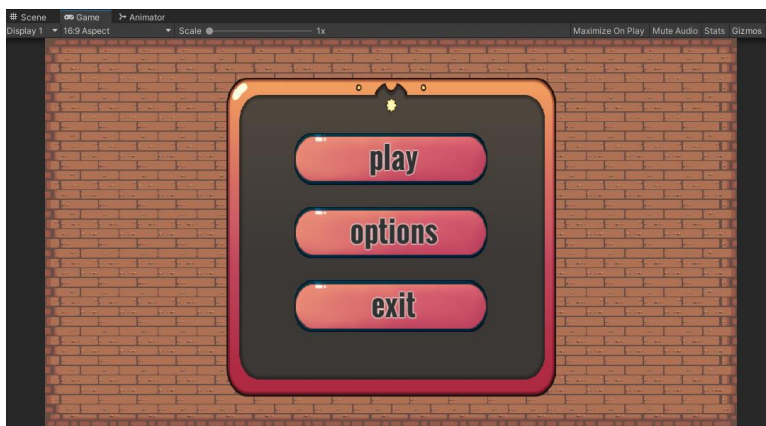


Рисунок 1.13 – Підсумковий вид головного меню

Пишу код для кнопок.

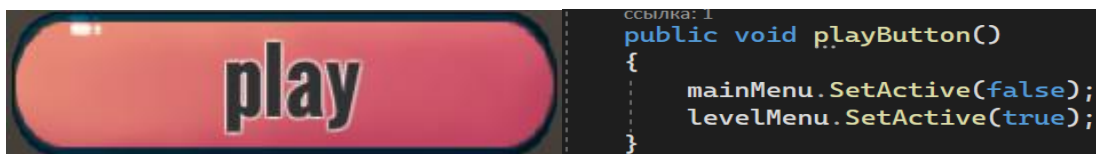


Рисунок 1.14 – Код кнопки "play"

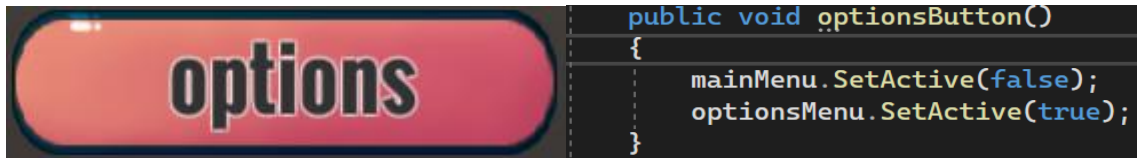


Рисунок 1.15 – Код кнопки “options”



Рисунок 1.16 – Код кнопки “exit”

1.5.2 Створення меню вибору рівня

Створюю на canvas ще одну panel називаю її "Select Level" і додаю на неї кнопки рівнів. Приклад наведено на рисунках 1.15 та 1.16

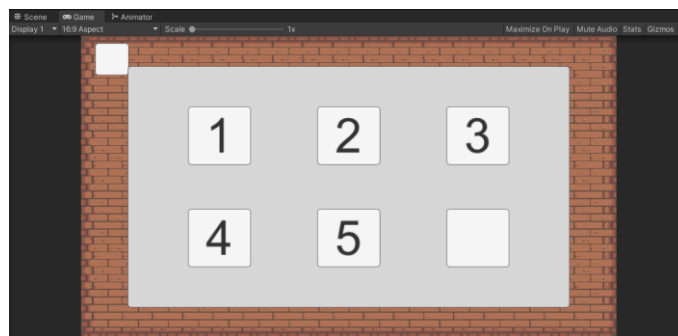


Рисунок 1.17 – Прототип меню вибору рівня

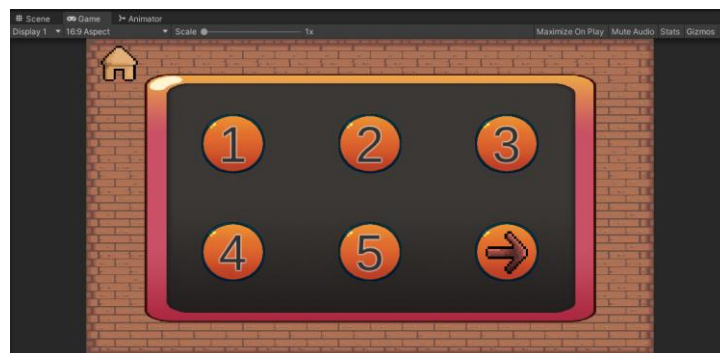


Рисунок 1.18 – Підсумковий вид меню вибору рівня

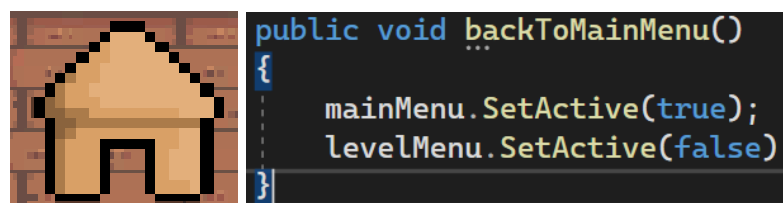


Рисунок 1.19 – Код кнопки повернення в головне меню



```
public void loadLevel(int levelIndex)
{
    SceneManager.LoadScene(levelIndex);
}
```

Рисунок 1.20 – Код кнопки вибору рівня

1.6 Додавання бази даних

SQLite - це програмна бібліотека, яка реалізує автономний, безсерверний, що не вимагає налаштування транзакційний механізм бази даних SQL. SQLite-це найбільш поширений в світі движок бази даних SQL. Вихідний код SQLite знаходиться у відкритому доступі.

Механізм SQLite не є автономним процесом, як інші бази даних, ви можете пов'язати його статично або динамічно відповідно до ваших вимог з вашим додатком. SQLite безпосередньо звертається до своїх файлів сховища.

- SQLite не вимагає окремого серверного процесу або системи для роботи (без сервера).
- SQLite поставляється з нульовою конфігурацією, що означає відсутність необхідності в налаштуванні або адмініструванні.
- Повна база даних SQLite зберігається в одному крос-платформному дисковому файлі.
- SQLite дуже маленький і легкий, менше 400 КБ в повній конфігурації або менше 250 КБ з опущеними додатковими функціями.
- SQLite є автономним, що означає відсутність зовнішніх залежностей.
- Транзакції SQLite повністю сумісні з ACID, що забезпечує безпечний доступ з декількох процесів або потоків

Створюю нову папку для бази даних і додаю туди:

Mono.Data.Sqlite.dll, sqlite3.dll, System.Configuration.dll

System.EnterpriseServices.dll, System.Security.dll.

Створюю нову базу даних і створюю в ній нову таблицю

					КС.55.22.00.ДП ПЗ	Аркуш
						24
Зм.	Аркуш	№ докум.	Підпис	Дата		

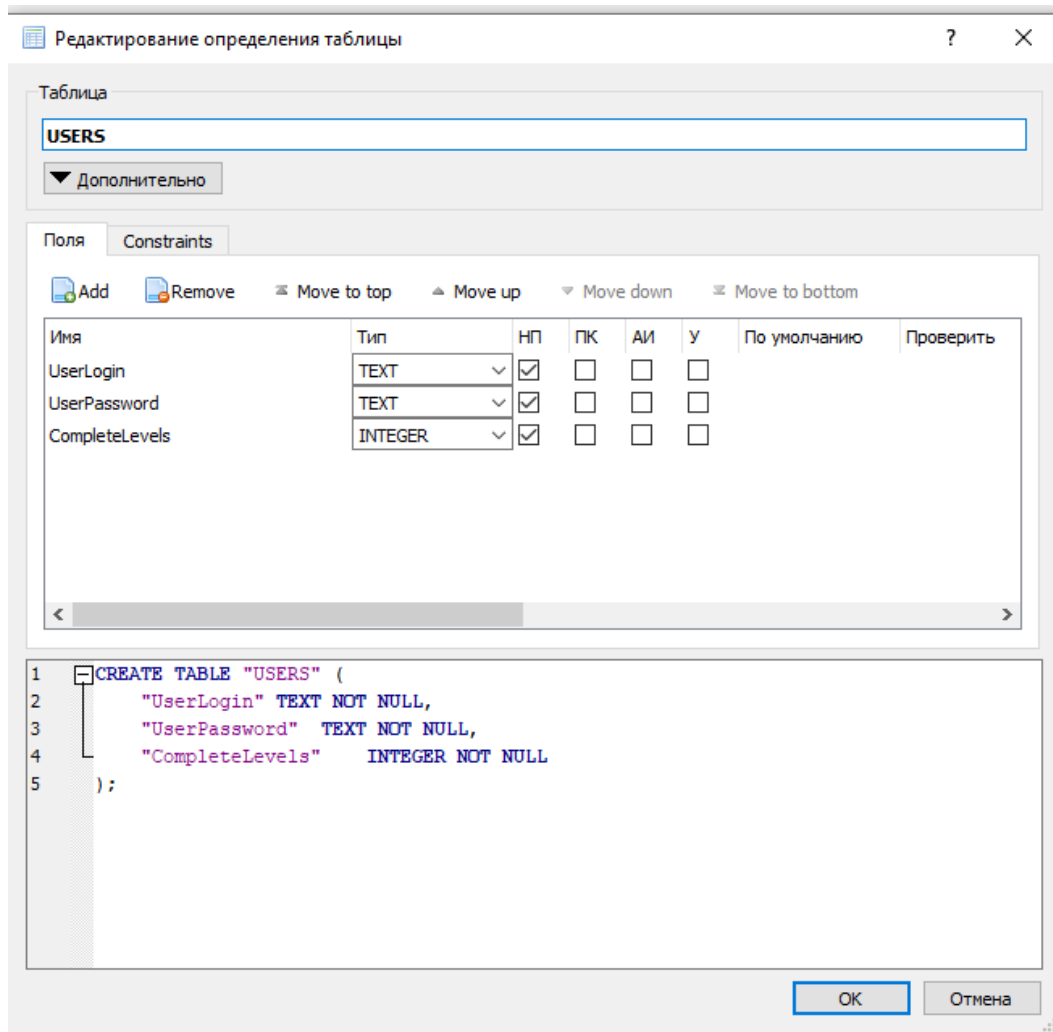


Рисунок 1.21 – створення нової таблиці в базі даних

```
using Mono.Data.Sqlite;
using System.Data;
using System;
```

Рисунок 1.22 – підключення бібліотек в коді

1.6.1 Використання бази даних

Тепер використовуючи додану базу даних можна реалізувати збереження прогресу окремо для кожного користувача.

Використовуючи створену базу даних зчитую з таблиці кількість пройдених рівнів у активного користувача і роблю кнопки невідкритих рівнів неактивними.

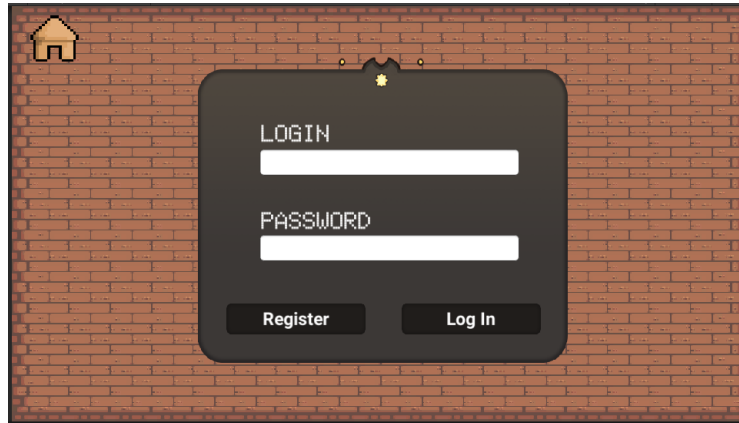


Рисунок 1.23 – Вікно входу в профіль гравця

```

void Start()
{
    path = Application.dataPath + "/SqDB/DB.db"; //путь к базе данных
    dbconnection = new SqliteConnection("URI=file:" + path);
    dbconnection.Open(); //открыть подключение
    if (dbconnection.State == ConnectionState.Open) // если подключение открыто
    {
        SqliteCommand cmd = new SqliteCommand();
        cmd.Connection = dbconnection;
        cmd.CommandText = "SELECT CompleteLevels FROM [USERS] WHERE UserLogin = '" + PlayerPrefs.GetString("User") + "'";
        SqliteDataReader reader = cmd.ExecuteReader(); //считывает пройденные уровня у активной учетки
        while (reader.Read())
        {
            levelUnLock = int.Parse(String.Format("{0}", reader[0]));
        }

        for (int i = 0; i < levelUnLock; i++)
        {
            _buttons[i].interactable = true;
        }
    }
}

```

Рисунок 1.24 – Використання БД

1.7 Створення рівня гри

Створюю нову сцену назвавши її "lv10". Додам на сцену два тайлмапа один для заднього фону другий для підлоги і стін. Заповнюю тайлмапи використовуючи створену до цього Time Palette. Міняю значення Order in layer другого шару з 0 на 1 щоб він відображався перед фоном і додаю на нього компонент Tilemap Collider 2D . Приклад наведено на рисунках 1.25 та 1.26 .

					КС.55.22.00.ДП ПЗ	Аркуш
						26
Зм.	Аркуш	№ докум.	Підпис	Дата		

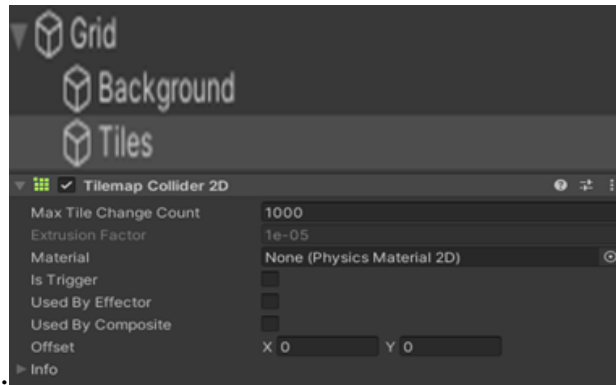


Рисунок 1.25 – компонент Tilemap Collider 2D

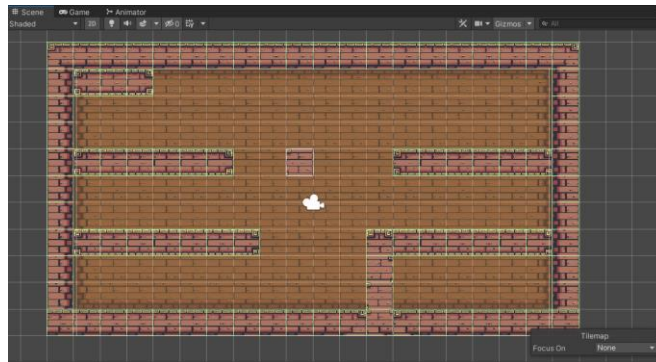


Рисунок 1.26 - підлоги та стіни з додаванням Tilemap Collider 2D

1.7.1 Створення ігрових механізмів

Кнопка. Створюю новий об'єкт називаю його " Button "і новий скрипт"Btn".У компоненті Sprite Renderer додаю спрайт кнопки. Додаю два компоненти BoxCollider2D і в 1 з них ставлю галочку is Trigger.

При вході будь-якого об'єкта в тригері кнопки спрайт кнопки зміниться на натиснуту а на вихід буде подана логічна одиниця і навпаки при виході об'єкта з тригера спрайт зміниться на віджатую а на вихід буде поданий логічний нуль

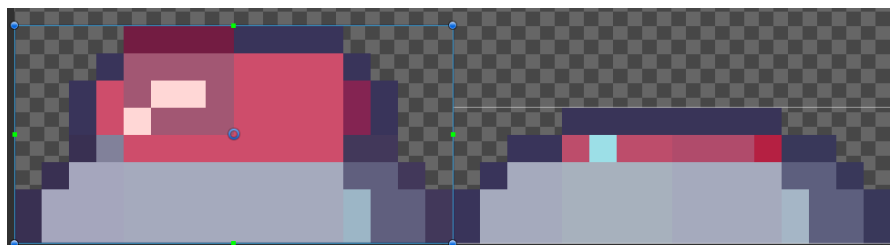


Рисунок 1.27 – Спрайти кнопки

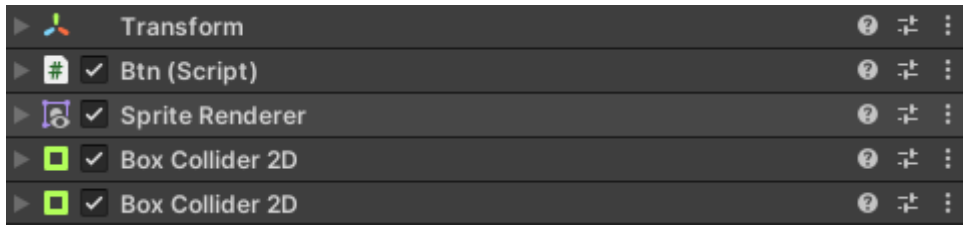


Рисунок 1.28 – Компоненти об'єкта button

```

6 public class Btn : MonoBehaviour
7 {
8     public Sprite[] sprites = new Sprite[2]; //Массив спрайтов кнопка нажата/отжата
9     private SpriteRenderer _spriteRenderer;
10    public UnityEvent<int> MyEvent; //Ивент для передачи значения 1 или 0 в другой скрипт
11    private Color[] _isActive = { new Color(0.44f, 0.378f, 0.378f, 1f), new Color(0.971f, 0.261f, 0.261f, 1f) };
12    public SpriteRenderer[] Wires; // массив спрайтов проводов
13    private int _objects = 0; // Объекты на кнопке
14
15    // Сообщение Unity | Ссылка: 0
16    void Start() //вызывается перед вызовом какого либо метода Update
17    {
18        _spriteRenderer = GetComponent<SpriteRenderer>();
19        MyEvent.Invoke(0);
20    }
21
22    // Сообщение Unity | Ссылка: 0
23    private void OnTriggerEnter2D(Collider2D collision) //вызывается когда игрок/объект
24    { //входит в триггер
25        _objects++;
26        buttonPressed(1);
27    }
28
29    // Сообщение Unity | Ссылка: 0
30    private void OnTriggerExit2D(Collider2D collision) //вызывается когда игрок/объект
31    { //входит в триггер
32        _objects--;
33        if (_objects == 0) //если на кнопки нет объектов
34            buttonPressed(0);
35    }
36
37    // Ссылка: 2
38    private void buttonPressed(int value)
39    {
40        _spriteRenderer.sprite = sprites[value]; //меняет спрайт кнопки на нажатую/отжатую
41        MyEvent.Invoke(value); //передает 1 или 0
42        for (int i = 0; i < Wires.Length; i++) //меняет цвет проводов
43        {
44            Wires[i].color = _isActive[value];
45        }
46    }
47 }

```

Рисунок 1.29 – Код компонента “Btn”

Важіль. Створюю новий об'єкт називаю його "Lever" і новий скрипт "Lever". У компоненті Sprite Renderer додаю спрайт важіля . Додаю компонент BoxCollider2D і ставлю галочку is Trigger.

При вході гравця в тригер спрайт важіля зміниться а на вихід буде подана логічна одиниця або логічний нуль.

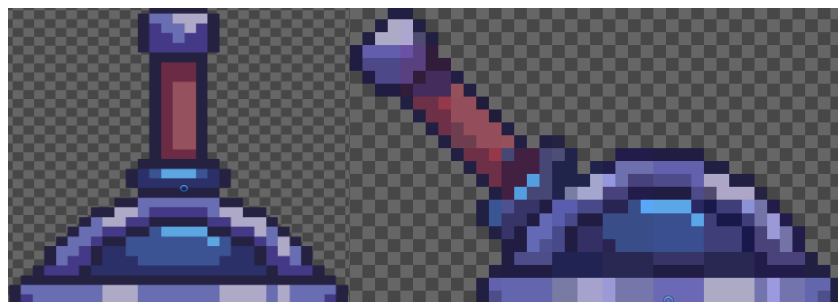


Рисунок 1.30 – Спрайти важіля

```
private void OnTriggerEnter2D(Collider2D collision)//вызывается когда игрок/объект
{
    //входит в триггер
    {
        if(collision.name == "Player") //если объект вошедший в триггер игрок
        {
            if (switcher == 1)
            {
                spriteRenderer.sprite = sprites[0]; //меняет спрайт рычага
                switcher = 0;
                MyEvent.Invoke(switcher); //передает логический 0
                for (int i = 0; i < Wires.Length; i++) //меняет цвет проводов
                {
                    Wires[i].color = disabled;
                }
            }
            else
            {
                spriteRenderer.sprite = sprites[1]; //меняет спрайт рычага
                switcher = 1;
                MyEvent.Invoke(switcher); //передает логическую 1
                for (int i = 0; i < Wires.Length; i++) //меняет цвет проводов
                {
                    Wires[i].color = active;
                }
            }
        }
    }
}
```

Рисунок 1.31 – Код важіля

Висувні списи. Створюю новий об'єкт називаю його "Spears" і новий скрипт "Spear". У компоненті Sprite Renderer додаю спрайт важіля . Додаю компонент BoxCollider2D.

При подачі одиниці списи будуть підніматися а при нулі опускатися змінюючи розмір колайдера на відповідний

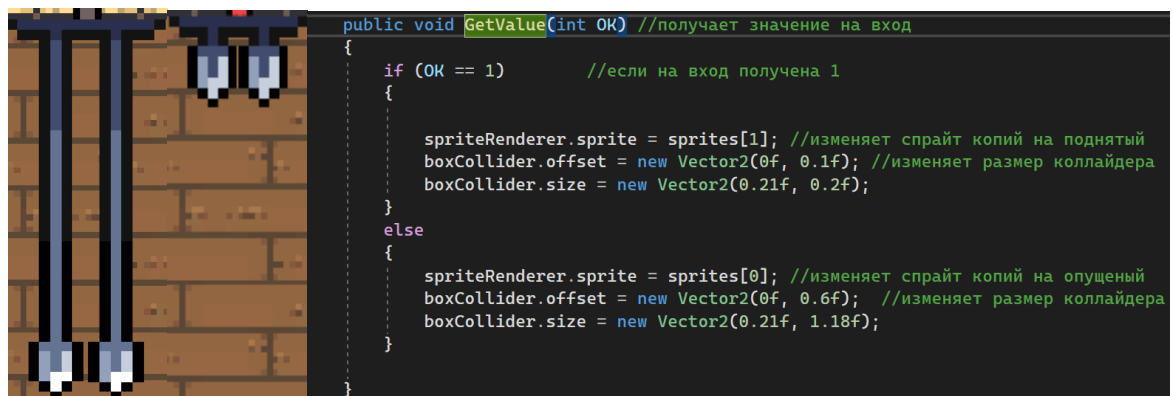


Рисунок 1.32 – Код і спрайти висувних спис

Логічні елементи. Логічний елемент - це базовий елемент цифрової схеми виконує елементарну логічну операцію перетворюючи таким чином множини вхідних логічних сигналів у вихідний логічний сигнал.



Рисунок 1.33 – Спрайти базових логічних елементів

					КС.55.22.00.ДП ПЗ	Аркуш
						29
Зм.	Аркуш	№ докум.	Підпис	Дата		

Логічне АБО. У логічного елемента або (диз'юнктора) на виході буде 1 коли або на один або на інший або на обидва входи будуть подані 1.

Таблиця 1.1 – Таблиця істинності елемента АБО

X1	X2	Y
0	0	0
1	0	1
0	1	1
1	1	1

```

public void GetValue1(int get) //получает значение на первый вход
{
    value1 = get;
    result = value1 | value2;
    MyEvent.Invoke(result); //передает на выход логический 0 или 1
    changeWiresColor(result); //меняет цвет проводов
}

Ссылка: 0
public void GetValue2(int get) //получает значение на второй вход
{
    value2 = get;
    result = value1 | value2;
    MyEvent.Invoke(result); //передает на выход логический 0 или 1
    changeWiresColor(result); //меняет цвет проводов
}
    
```

Рисунок 1.34 – Код елемента АБО

Логічне І. Логічний елемент І реалізує функцію множення (кон'юнкцію). У цього елемента 1 на виході буде тільки тоді коли на обидва входи подані 1.

Таблиця 1.2 – Таблиця істинності елемента І

X1	X2	Y
0	0	0
1	0	0
0	1	0
1	1	1

```

Ссылка: 0
public void GetValue1 (int get) //получает значение на первый вход
{
    value1 = get;
    result = value1 & value2;
    MyEvent.Invoke(result); //передает на выход логический 0 или 1
    changeWiresColor(result); //меняет цвет проводов
}

Ссылка: 0
public void GetValue2(int get) //получает значение на второй вход
{
    value2 = get;
    result = value1 & value2;
    MyEvent.Invoke(result); //передает на выход логический 0 или 1
    changeWiresColor(result); //меняет цвет проводов
}
    
```

Рисунок 1.35 – Код елемента І

Елемент НЕ. Логічний елемент НЕ (інвертор) це єдиний елемент має тільки 1 вхід. Якщо на вході 1 то на виході 0 і навпаки якщо на вході 1 то на виході 0.

Таблиця 1.3 – Таблиця істинності елемента НЕ

X	Y
0	1
1	0

```

public void GetValue(int get) //получает значение на вход
{
    result = (get + 1) % 2; //меняет 1 на 0 или 0 на 1
    MyEvent.Invoke(result); //передает на выход инверсное значение
    if (result == 1)
    {
        for (int i = 0; i < Wires.Length; i++) // //меняет цвет проводов
        {
            Wires[i].color = active;
        }
    }
    else
    {
        for (int i = 0; i < Wires.Length; i++) //
        {
            Wires[i].color = disabled;
        }
    }
}

```

Рисунок 1.36 – Код елемента НЕ

1.7.2 Створення гравця та кристалу

Створюю об'єкт назвавши його "Player". Додаю на нього три компоненти Rigidbody2D, Capsule Collider 2D і Animator. Приклад наведено на рисунку 1.37. Створюю Animator Controller і прикріплюю його до створеного аніматора.

Компонент Rigidbody дає ігровому об'єкту фізичну присутність в сцені, оскільки він може реагувати на сили. Можна прикладати сили безпосередньо до ігрового об'єкта або дозволити йому реагувати на зовнішні сили, такі як гравітація або інше жорстке тіло, що вдаряє по ньому.

Компоненти коллайдера визначають форму об'єкта для цілей фізичних зіткнень. Невидимий колайдер не обов'язково повинен бути точно такою ж формою, як сітка об'єкта, і насправді грубе наближення часто більш ефективно і невиразно в ігровому процесі.

Animator Controller керує логікою анімованого ігрового об'єкта. Всередині контролера Animator є стани і Подсостояния, які пов'язані один з

одним за допомогою переходів. Стани це представлення анімаційних кліпів в Аніматорі.

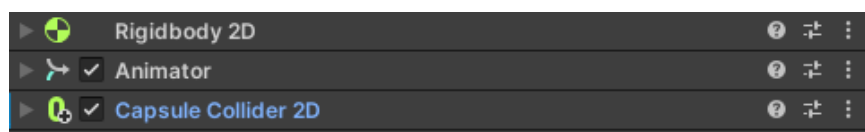


Рисунок 1.37 – Компоненти об'єкта "Player"

Тепер потрібно створити 4 анімації для гравця:

1. Анімація спокою
2. Анімація пересування
3. Анімація стрибка
4. Анімація смерті

У вікні анімація створюю 4 анімаційних кліпу "Player_Idle", "Player_run", "Player_Jump", "Player_Death"

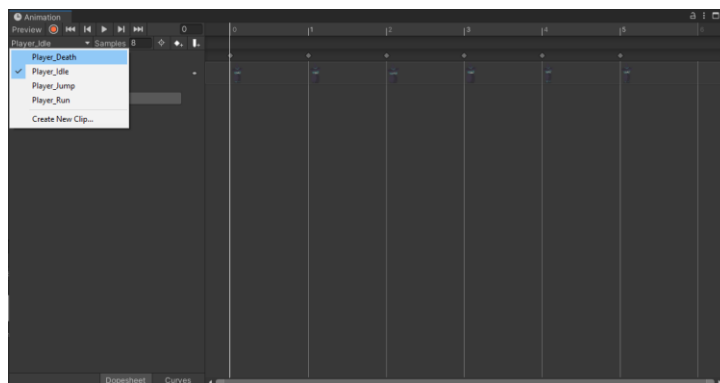


Рисунок 1.38 – створення анімації



Рисунок 1.39 – Спрайти анімації спокою

У вкладці Parameters створюю три змінні :

змінну типу float " speed " та булеві змінні " IsJumping " і " IsAlive ".

Налаштовую переходи так щоб при початку рівня запускала анімація Player_Idle. З анімації Player_Idle якщо змінна speed > 0.1 запускала анімація Player_Run і з Player_Run назад в Player_Idle якщо швидкість == 0. З будь-

якого стану перехід до Player_Jump якщо IsJump == true і з Player_Jump до Player_Idle. якщо IsJump == false і швидкість == 0 або до Player_Run якщо IsJump == false і швидкість > 0.1. Приклад наведено на рисунку 1.40.

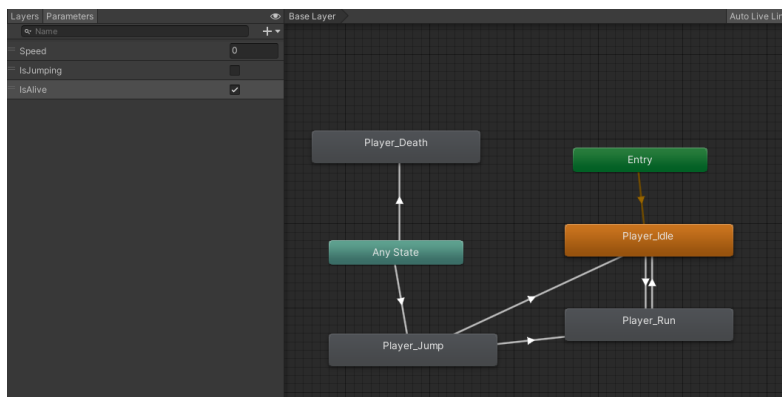


Рисунок 1.40 – Налаштування Animator Controller

```

void Update()
{
    horizontalMove = Input.GetAxisRaw("Horizontal") * runSpeed;
    animator.SetFloat("Speed", Mathf.Abs(horizontalMove));
    if (Input.GetButtonDown("Jump"))
    {
        jump = true;
        animator.SetBool("IsJumping", true);
    }
}

Сообщение Unity | Ссылка: 0
void FixedUpdate()
{
    controller.Move(horizontalMove * Time.fixedDeltaTime, false, jump);
    jump = false;
}

Ссылка: 0
public void OnLanding()
{
    animator.SetBool("IsJumping", false);
}

```

Рисунок 1.41 – PlayerController

Кристал. У грі щоб завершити рівень потрібно буде зібрати кристал

```

private void OnTriggerEnter2D(Collider2D collision) //вызывается когда объект входит в триггер
{
    if (collision.name == "Player") //если объект вошедший в триггер игрок
    {
        ui.SetActive(true); //включает окно конца уровня
        canvas.SetActive(false); //отключает меню паузы
        Destroy(gameObject); //уничтожает кристал
        Time.timeScale = 0; //ставит паузу
        UnlockLevel(); //вызывает метод UnlockLevel
    }
}

ссылка: 1
public void UnlockLevel()
{
    int currentLevel = SceneManager.GetActiveScene().buildIndex; // получает индекс текущей сцены

    if (currentLevel >= PlayerPrefs.GetInt("Levels"))
    {
        PlayerPrefs.SetInt("levels", currentLevel + 1); //открывает доступ к следующему уровню
    }
}

```

Рисунок 1.42 – Код код об'єкта кристал



Рисунок 1.43 – Вікно кінця рівня

1.7.3 Алгоритм роботи та дій користувача

Запускаємо гру через папку з грою натискаючи на її іконку.

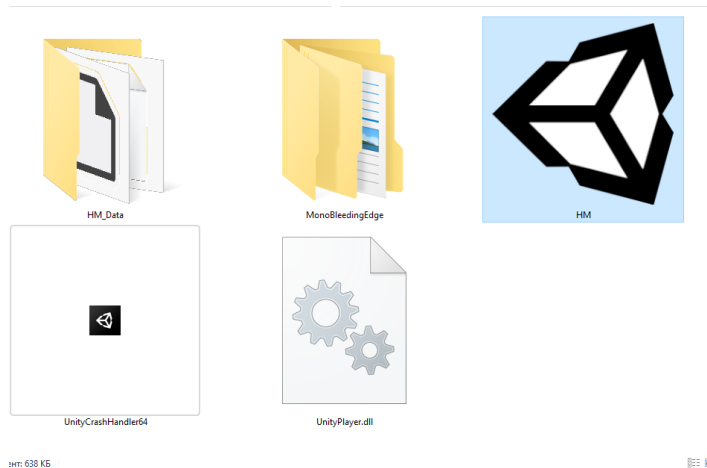


Рисунок 1.44 – папка з грою

Користувач після запуску гри бачитиме її головну сторінку з переліком кнопок: «play», «options», «exit» та кнопку профіля.

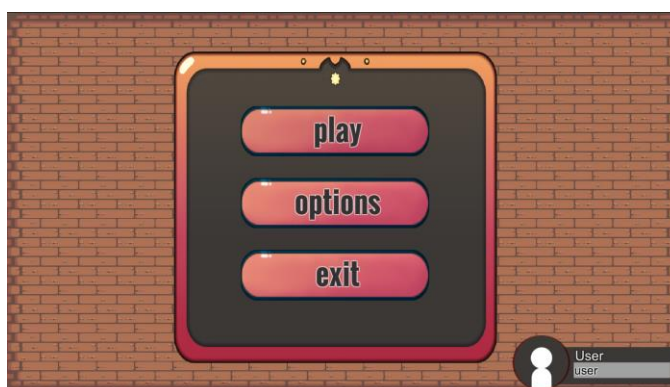


Рисунок 1.45 – Головне меню

Натискаючи на кнопку профіля, гравцеві буде запропоновано меню входу в аккаунт.

На рисунку 1.46 представлено меню для входу в профіль гравця.

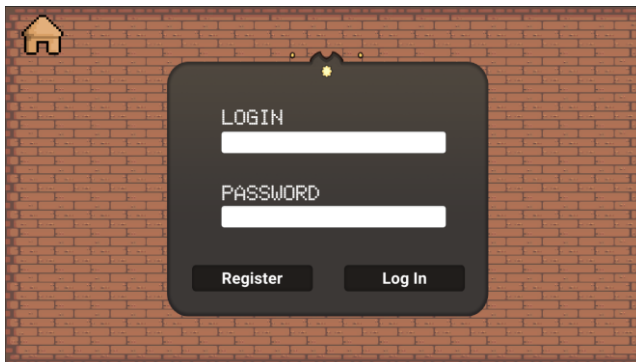


Рисунок 1.46 – Меню входу в профіль гравця

Гравець має змогу обрати рівень для проходження гри. Однак він може або повторно пройти рівні які вже проходив, або пройти наступний рівень. Всі решта кнопок для вибору рівнів є неактивними.

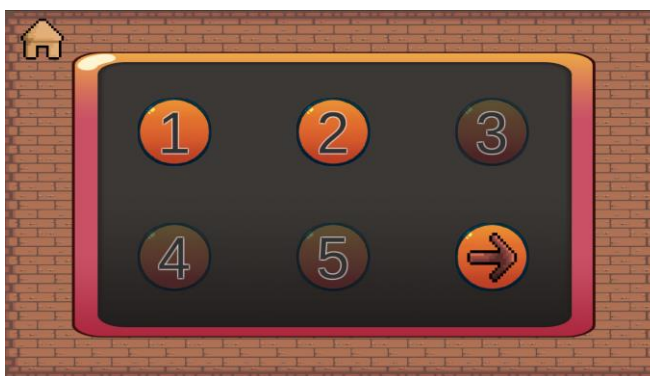


Рисунок 1.47 – Меню вибору рівнів

На рисунку 1.48 представлено скриншот робочого вікна для проходження першого рівня гри.

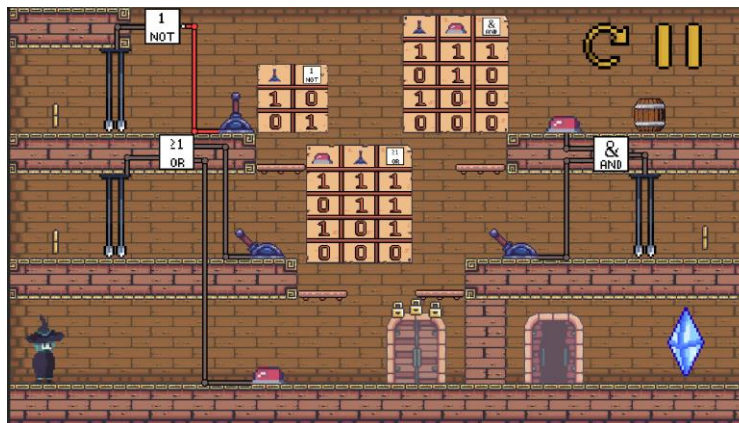


Рисунок 1.48 – Перший рівень

Якщо гравцю потрібно прервати гру, то він може скористатися паузою, натиснувши кнопку Esc або натиснути на кнопку паузи в правому верхньому куті екрану.



Рисунок 1.49 – Меню паузи

Після проходження рівня буде показано вікно кінця рівня. І гравець може повернутися в меню натиснувши кнопку «menu» або перейти на наступний рівень натиснувши кнопку «next level»



Рисунок 1.50 – Меню при проходженні рівня

Після смерті персонажа буде показано вікно поразки і гравець зможе почати рівень спочатку натиснувши кнопку «try again» або повернутися в меню натиснувши кнопку «menu»



Рисунок 1.52 – Меню при смерті гравця

Висновок

Отже в цьому розділі було проаналізовано існуючі технології створення ігор та представлено опис створюваного в данному дипломному проекті ігрового тренажера для вивчення роботи логічних елементів.

					КС.55.22.00.ДП ПЗ	Аркуш
						37
Зм.	Аркуш	№ докум.	Підпис	Дата		

2 ЕКОНОМІЧНА ЧАСТИНА

Метою даних розрахунків є обчислення вартості виконання науково-дослідної роботи «Створення ігрового тренажеру для вивчення роботи цифрових пристроїв» У дипломній роботі був запропонован ігровий тренажер для вивчення роботи базових логічних елементів .

Даний вид проекту відноситься до науково-дослідницької розробки. Оцінка якості розробленого проекту включає визначення трудомісткості і вартості його створення.

Розрахунок трудомісткості НДР здійснений в наступній послідовності:

- 1) Складений перелік всіх етапів і видів робіт, які необхідно виконати в ході даної НДР. Після узгодження з керівником проекту допущено виключення, доповнення, об'єднання окремих етапів і видів робіт;
- 2) По кожному виду робіт визначений кваліфікаційний рівень виконавців. Перелік етапів і робіт, що виконуються при проведенні НДР, приведений в таблиці 2.1.

Таблиця 2.1 - Перелік етапів і робіт, що виконуються при проведенні НДР

Етап проведення НДР	Вигляд робіт	Посада виконавця
Розробка технічного завдання (ТЗ)	1.Складання і затвердження ТЗ для НДР по розробці «ігрового тренажера»	Дипломник, керівник
Вибір напрямку дослідження	1. Збір і вивчення науково-технічної літератури. 2. Формулювання можливих напрямів вирішення завдань, поставлених в технічному завданні НДР і їх порівняльна оцінка. 3. Вибір напрямку проведення досліджень 4. Розробка плану проведення досліджень для подальшої розробки.	Дипломник керівник
Теоретичні і експериментальні дослідження	1.вивчення існуючих підходів до використання комп'ютерних ігор в освітньому процесі	Дипломник керівник консультанти

	2. Аналіз предметної області 3. Вибір засобів реалізації 4. МДЕ: Механіка, динаміка та естетика 5. Створення головного меню 6. Створення меню вибору рівня 7. Додавання та використання бази даних 8. Створення рівня 9. Створення ігрових механізмів 10. Створення гравця та кристалу 11. Алгоритм роботи та дій користувача	
Узагальнення і оцінка результатів досліджень	1. Узагальнення результатів 2. Оцінка повноти вирішення поставлених завдань. 3. Складання і оформлення звіту. Розгляд результатів проведеною НДР і прийняття результатів в цілому.	Дипломник керівник консультанти

Оцінка тривалості виконання робіт розраховується на основі вірогідних оцінок робіт, що задаються виконавцями.

Таблиця 2.2 - Очікувана трудомісткість робіт

Вигляд роботи	Очікуваний час виконання (дні)
1. вивчення існуючих підходів до використання комп'ютерних ігор в освітньому процесі	2
2. Аналіз предметної області	3
3. Вибір засобів реалізації	2
4. МДЕ: Механіка, динаміка та естетика	2
5. Створення головного меню	3
6. Створення меню вибору рівня	3
7. Додавання та використання бази даних	4
8. Створення рівня	3
9. Створення ігрових механізмів	3
10. Створення гравця та кристалу	3
11. Алгоритм роботи та дій користувача	2

Розрахунок собівартості і ціни виконання НДР. Виходячи з особливостей створення науково – технічної продукції і її залежності від інтелектуальної праці, розрахунок собівартості і ціни виконання НДР включає наступні статті витрат: витрати на матеріали, основна і додаткова заробітна плата, відрахування до єдиного соціального фонду страхування, витрати на роботи, що виконуються сторонніми організаціями, і деякі інші.

1) Витрати на матеріали складають 80 грн. (Папір А4)

2) До витрат «Основна заробітна плата» відносяться оплата праці виконавців, безпосередньо притягнених до її виконання. Розмір основної зарплати встановлюється виходячи з чисельності різних категорій виконавців, трудомісткості, що витрачається ними на виконання різних видів робіт, а також їх середньої заробітної плати (ставки) за один робочий день. Відповідно до статті 8 «Закону про Державний бюджет України на 2021» встановлено мінімальну заробітну плату у місячному розмірі з 1 січня 2022 року - 6500 гривень; мінімальну погодинну тарифну ставку – 39,26 грн.

Середня зарплата за один робочий день для кожного виконавця визначена по формулі:

$$Зден = п.т.с. * 8;$$

де п.т.с – погодинна тарифна ставка, грн.;

8 – тривалість робочого дня, год.

$$Зден дипломника = 39.26 * 8 = 314,08 \text{ грн.}$$

$$Зден керівника = 65 * 8 = 520 \text{ грн.}$$

$$Зден консультантів = 55 * 8 = 440 \text{ грн.}$$

Витрати на основну заробітну плату, НДР, що включаються в собівартість, приведені в таблиці 2.3.

Таблиця 2.3 - Витрати на основну заробітну плату

Виконавець	Погодинна тарифна ставка, грн	Денна ставка, грн	Трудомісткість робочих днів	Сума основної зарплати, грн
Дипломник	39,26	314,08	30	9422,4

Керівник	65	520	1	520
Консультант по економічній частині	55	440	0,25	110
Консультант по охороні праці	55	440	0,25	110
Нормоконтроль	56	448	0,25	112
Всього (Зо)				10274,4

3) Витрати на додаткову заробітну плату визначаються у відсотках від основної. У наукових закладах додаткова заробітна плата складає 10-12% від основної заробітної плати.

$$Зд=10\% * 10274,4;$$

$$Зд= 1027,44\text{грн}$$

4) До складу собівартості НДР включаються податки, збори і інші обов'язкові платежі, встановлені системою оподаткування що діє. Відрахування до єдиного соціального внеску складає:

$$Зесв=0,22*(10274,4 + 1027,44);$$

$$Зесв = 2486,4\text{грн.}$$

5) До накладних витрат відносять витрати на управління і господарське обслуговування, що відноситься до всіх виконуваних НДР. У наукових закладах накладні витрати складають 40 -120% від основної і додаткової заробітної плати.

$$Рнакл= (1027,44 + 10274,4)*0,4;$$

$$Рнакл = 4 520,73\text{грн.}$$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому НДР за формою, приведеною в таблиці 2.4.

Таблиця 2.4 - Калькуляція планової собівартості

Статті витрат	Сума, грн.
1. Матеріали	80
2. Основна заробітна плата	10274,4
3. Додаткова заробітна плата	1027,44
4. Відрахування до єдиного соціального внеску	2486,4

					КС.55.22.00.ДП ПЗ	Аркуш
						41
Зм.	Аркуш	№ докум.	Підпис	Дата		

5. Накладні витрати	4 520,73
Планова собівартість (Спл)	18 388,97

Плановий прибуток визначений по формулі:

$$Ппл = 0,1 * 18\,388,97 = 1\,838,89 \text{ грн}$$

Де 0,1 – норматив, який враховує граничний рівень рентабельності, встановлений чинним законодавством для науково-технічної продукції.

Договірна ціна визначається по формулі:

$$Цнір = 18\,388,97 + 1\,838,89 = 20\,227,86 \text{ грн.}$$

Ціну реалізації встановлюємо з урахуванням ПДВ

$$ПДВ = 0,2 * 20\,227,86 = 4\,045,57 \text{ грн.}$$

Звідси ціна реалізації становить:

$$Цр = 20\,227,86 + 4\,045,57 = 24\,273,43 \text{ грн.}$$

					КС.55.22.00.ДП ПЗ	Аркуш
						42
Зм.	Аркуш	№ докум.	Підпис	Дата		

3 ОХОРОНА ПРАЦІ

3.1 Вступ

Охорона праці - це система заходів, покликана забезпечити збереження життя і здоров'я людей в процесі виконання ними своїх трудових обов'язків.

Охорона праці містить безліч аспектів таких як норми освітлення, рівня шуму, мікроклімату, розмір робочого місця і безліч інших, не менш важливих пунктів.

Нині комфорт та безпеку робітників забезпечують нормативно- правові акти, закони та положення. Адже, якщо працівник працюватиме у негативній для нього обстановці або під впливом негативних факторів і на робочому місці не забезпечуватиметься належні умови праці, тоді цей працівник буде швидше втомлюватися, допускати помилки, що може стати причиною розвитку професійної хвороби або виробничої травми.

В дипломному проєкті охорона праці розглядається з точки зору безпеки праці на робочому місці програміста, який займається розробкою програми з використанням персонального комп'ютера.

3.2 Аналіз та безпека умов праці працівника на робочому місці

Шкідливими факторами на робочому місці цього учня можуть стати чинники що приведенні у таблиці нижче

Таблиця 3.1 - Шкідливі чинники

Джерело	Чинники	
	Шкідливі	Небезпечні
1.Робота за персональним комп'ютером.	1.Підвищена або знижена температура повітря робочої зони 2.Підвищена або знижена рухливість повітря	1.підвищене значення напруги в електричному ланцюзі, замикання якої може статися через тіло людини

	3. підвищена або знижена іонізація повітря 4. статичні перевантаження кістково-м'язового апарату і динамічні локальні перевантаження м'язів кистей рук 5. перенапруження зорового аналізатора 6. монотонність праці	
--	--	--

Ця робота не вимагає будь-яких фізичних навантажень і належить до категорії 1а. Під час виконання роботи використовується персональний комп'ютер.

3.3 Організація робочого місця

Робочий стіл з урахуванням характеру виконуваної роботи повинен мати достатній розмір для раціонального розміщення монітора (дисплея), клавіатури, іншого використовуваного обладнання та документів, поверхня, що володіє низькою відбивною здатністю. Клавіатура розташовується на поверхні столу таким чином, щоб простір перед клавіатурою було достатнім для опори рук працівника (на відстані не менше ніж 300 мм від краю, зверненого до працівника). Щоб забезпечувалося зручність зорового спостереження, швидке і точне зчитування інформації, площина екрану монітора розташовується нижче рівня очей працівника переважно перпендикулярно до нормальної лінії погляду працівника (нормальна лінія погляду - 15° вниз від горизонталі).

Для виключення впливу підвищених рівнів електромагнітних випромінювань відстань між екраном монітора і працівником має становити не менше 500 мм (оптимальне 600-700 мм). Робочий стілець (крісло) повинен бути стійким, місце сидіння повинно регулюватися по висоті, а спинка сидіння - по

					КС.55.22.00.ДП ПЗ	Аркуш
						44
Зм.	Аркуш	№ докум.	Підпис	Дата		

висоті, кутах нахилу, а також відстані спинки від переднього краю сидіння. Для тих, кому це зручно, передбачається підставка для ніг.

3.4 Мікроклімат

Згідно до ГОСТ 12.1.005-88, СН 4088-86. Мікроклімат приміщення повинен відповідати нормативам.

Для приміщення з комп'ютером існують певні вимоги до температури, вологості і наявності пилу. Температура повинна знаходитися на рівні 22-25 °С, відносна вологість – 40-60 %, рівень аероіонів – від 400-600 до 50 000 (оптимальний – 1500-5000).

Таблиця 3.2 - Мікроклімат приміщень

Пора року	Температура повітря у градусах Цельсія.	Відносна вологість повітря у відсотках.	Швидкість руху повітря у метрах за секунду.
Холодна	22-24	40-60	0,1
Тепла	23-25	40-60	0,1

Це оптимальні умови для забезпечення максимально комфортного теплового балансу температури тіла людини і його терморегуляції. Якщо температура вище норми, кровоносні судини розширюються і тепловіддача в навколишнє середовище зростає. При зниженні температури кровоносні судини відповідно звужуються приплив крові до тіла сповільнюється і тепловіддача зменшується.

На терморегуляцію організму впливає також вологість повітря. Занадто висока вологість (більше 85 %) ускладнює терморегуляцію, а занадто низька (менше 20%) викликає пересихання слизових, причому не тільки дихальних шляхів, але і очей.

Не менш важлива оптимальна вологість в приміщенні: чим вона вища, тим слабкіше вплив електростатичних і електромагнітних полів, рівень випромінювання яких в приміщенні, де встановлений комп'ютер, завжди підвищений.

Рівень іонів у повітрі що повинні відповідати санітарно-гігієнічним нормам № 2152-80.

Таблиця 3.3 - Рівень іонів у повітрі

Рівні	Кількість на сантиметр кубічний.	
	п+	п-
Мінімальне	400	600
Оптимальне	1500-300	3000-5000
Максимальне	50000	50000

Для підтримки мікроклімату слід використовувати кондиціонери та зволожувачі повітря або інші прилади.

3.5 Освітлення

Робоче місце з використанням ПК розміщується таким чином, щоб природне світло падало збоку (бажано зліва). Для зниження яскравості в полі зору при природному освітленні застосовуються регульовані жалюзі, щільні штори. Світильники загального і місцевого освітлення повинні створювати нормальні умови освітленості і відповідний контраст між екраном і навколишнім оточенням з урахуванням виду роботи і вимог видимості з боку працівника.

Освітленість на поверхні столу в зоні розміщення робочого документа повинна становити 300-500 люкс. Можливі заважають відображення і відблиски на екрані монітора і іншому обладнанні усуваються шляхом відповідного розміщення екрану, обладнання, розташування світильників місцевого освітлення. При рядному розміщенні робочих столів розташування екранів відеомоніторів назустріч один одному через їх взаємне відображення не допускається

У разі використовувани штучного освітлення слід використовувати люмінесцентні лампи. Допускається використання металогалогенних ламп потужністю 250Вт. Також допускається використання ламп розжарювання для місцевих світильників. Використовування будь яких світильників без розсіювачів заборонено.

3.6 Електробезпека

Конструктивні заходи забезпечують захист від випадкового дотику до струмопровідних частин за допомогою захисних оболонок і ізоляції струмоведучих частин.

Схемно-конструктивні заходи призначені для забезпечення захисту від ураження електричним струмом при дотику до металевих оболонок, які можуть опинитися під напругою в ре-зультаті аварії. У даному приміщенні в комп'ютерах застосовується занулення. Біля монітора передбачена подвійна ізоляція. Заходи захисту

Необхідно дотримуватися правил техніки безпеки при роботі з високою напругою і наступних запобіжних засобів:

- монтаж, обслуговування, ремонт і наладка ЕОМ, заміна деталей, пристосувань, блоків повинна здійснюватися тільки при повному відключенні живлення;
- заземлення конструкції приміщення мають бути надійно захищені діелектричними щитками або сітками від випадкового дотику.

3.7 Пожежна безпека

Пожежна безпека входить в комплекс заходів з охорони праці, і організаційна робота в цій сфері на об'єктах господарювання включає широкий спектр заходів, а саме:

- створення умов для безпечної праці,
- мінімізації ризику виникнення пожеж,

					КС.55.22.00.ДП ПЗ	Аркуш
						47
Зм.	Аркуш	№ докум.	Підпис	Дата		

- своєчасне і повноцінне забезпечення технічними засобами для запобігання займання та усунення самих пожеж та їх наслідків,
- контроль дотримання протипожежних вимог і норм законодавства, розробка і впровадження регламентів по гасінню пожеж, евакуації та порятунку з місць пожежі й задимлення людей і майна (матеріальних цінностей),
- внутрішнє і зовнішнє навчання співробітників.

Первинні засоби пожежогасіння застосовуються для боротьби з пожежами на початковій стадії. До них належать: пожежні кран-комплекти, вогнегасники, пожежний інвентар (резервуари з водою, ящики з піском, пожежні відра, лопати), а також різний переносний пожежний інструмент (кирки, сокири, багри, ломы і т. ін.). Коли від пожежі захищаються приміщення з персональними комп'ютерами, то слід враховувати специфіку вогнегасних речовин у вогнегасниках, які призводять під час гасіння до псування обладнання. Ці приміщення рекомендується оснащувати вуглекислотними вогнегасниками з урахуванням гранично допустимої концентрації вогнегасної речовини.

Вуглекислотні вогнегасники вважаються найефективнішими для гасіння електрообладнання та електроприладів. Ліквідація полум'я відбувається за рахунок низької температури вогнегасної речовини, яка дозволяє не тільки збити вогонь, але і остудити тліючі ділянки ізоляції. З недоліків вуглекислоти можна відзначити тільки шкідливе випаровування цієї речовини. Саме тому забороняється гасити електроустановки в закритих приміщеннях.

– Вуглекислота не залишає слідів після випаровування і в той же час не пошкоджує запалене електрообладнання. Це особливо важливо при гасінні комп'ютерної техніки або загорівся телевізора.

– Вуглекислотними вогнегасниками можна гасити електроустановки під напругою до 10000 Вольт (10 кВ).

Також при роботі з ПЕОМ, на робочому місці забороняється мати вогненебезпечні речовини. Перш за все, не можна допускати накопичення паперових відходів поблизу ПЕОМ і їх несвоєчасного прибирання. У

					КС.55.22.00.ДП ПЗ	Аркуш
						48
Зм.	Аркуш	№ докум.	Підпис	Дата		

приміщеннях забороняється: запалювати вогонь; включати електрообладнання, якщо в приміщенні пахне газом; курити; сушити що-небудь на опалювальних приладах; закривати вентиляційні отвори в електроапаратурі. Підключення ПЕОМ до електромережі необхідно проводити через мережеві фільтри. Мережеві фільтри повинні підключатися до електромережі через розетки з заземлюючими контактами.

Джерелами займання у виробничих приміщеннях з ПЕОМ можуть бути: Іскра при розряді статичної електрики; іскри від електрообладнання; іскри від удару і тертя; відкрите полум'я.

3.7 Висновки

Охорона праці відіграє важливу роль у трудовому житті людини. Правильна організація праці значно підвищує його продуктивність та дозволяє зробити виробничий процес більш чітким, що підвищить рентабельність самого підприємства.

					КС.55.22.00.ДП ПЗ	Аркуш
						49
Зм.	Аркуш	№ докум.	Підпис	Дата		

ВИСНОВКИ

В процесі роботи я провів аналіз використання існуючих підходів використання комп'ютерних ігор в освітньому процесі. Проаналізував сучасні існуючі технології створення ігор.

Спроектував та реалізував ігровий тренажер для вивчення роботи логічних елементів використовуючи міжплатформенне середовище розробки комп'ютерних ігор Unity

					КС.55.22.00.ДП ПЗ	Аркуш
						50
Зм.	Аркуш	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Троелсен Ендрю, Джетікс Філіп. Мова програмування С# 7 і платформи .NET і .NET Core, 2018. Видавництво Вільямс. 1328 с.
2. Посібник користувача Unity 2021.3 (LTS)
<https://docs.unity3d.com/ru/530/Manual/> (дата звернення: 10.06.2022)
3. <https://stackoverflow.com/> (дата звернення: 9.06.2022)
4. <https://www.cyberforum.ru/unity> (дата звернення: 14.05.2022)
5. Paris Buttfield-Addison Unity Game Development Cookbook: Essentials for Every Game, 2019. O'Reilly Media; 1st edition. 408 с.
6. <http://unity3d.ru/> (дата звернення: 25.05.2022)
7. <https://qna.habr.com/> (дата звернення: 28.05.2022)
8. Jeremy Gibson Bond. Introduction to Game Design, Prototyping, and Development, 2014. Addison-Wesley Professional; 1st edition 1024 с.
9. Jared Halpern, Developing 2D Games with Unity: Independent Game Programming with C#, 2018. Apress; 1st ed. Edition. 410 с.
10. Alan Beaulieu, Learning SQL: Master SQL Fundamentals, 2009. O'Reilly Media; 2nd edition 338 с.
11. ГОСТ 27331-87 Пожарная техника. Классификация пожаров
12. ГОСТ 12.003-74 ССБТ. Опасные и вредные производственные факторы. Классификация
13. ГОСТ 12.1.030-81 ССБТ. Электробезопасность. Защитное заземление, зануление.
14. ДСН 3.3.6.042-99. Санітарні норми мікроклімату виробничих приміщень
15. Бойчик І. М. Економіка підприємства : навчальний посібник для студентів економічних спеціальностей вищих навчальних закладів I-IV рівнів акредитації. Третє видання, випр. і доп. / І. М. Бойчик, П. С. Харів., М. І. Холчан, Ю. В. Піча. – К. : Каравела, 2016. – 328 с

					КС.55.22.00.ДП ПЗ	Аркуш
						51
Зм.	Аркуш	№ докум.	Підпис	Дата		