ДЕРЖАВНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук та інформаційних технологій

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Розробка гри-платформи з використанням рушія Unity»

Студента 4 курсу, 8 групи, Іслам Загідула Раджауловича

спеціальності
122 «Комп'ютерні науки» _____
підпис студента

Науковий керівник
кандидат педагогічних наук, доцент _____ Базурін Віталій Миколайович
підпис керівника

Гарант освітньої програми
кандидат технічних наук, професор _____ Демідов Павло Георгійович
підпис керівника

Київ 2023

Державний торговельно-економічний університет

Факультет інформаційних технологій
Кафедра комп'ютерних наук та інформаційних систем
Спеціальність 122 «Комп'ютерні науки»

Затверджую
Зав. кафедри _____Пурський О.І.
«12» грудня 2022р.

Завдання
на випускну кваліфікаційну роботу (проект) студенту

Ісламу Загідула Раджауловича
(прізвище, ім'я, по батькові)

1. Тема випускної кваліфікаційної роботи (проекту)
«Розробка гри-платформи з використанням рушія Unity»
Затверджена наказом ректора від *«09» грудня 2022 р. № 3332*
2. Строк здачі студентом закінченої роботи *30 травня2023 року*
3. Цільова установка та вихідні дані до роботи
Мета роботи: створення і написання демонстраційної гри "Гра-платформа" із жанром платформер на ігровому рушії Unity.
Об'єкт дослідження: гра на ігровому рушії Unity
Предмет дослідження: процес розробки гри,створення моделей,написання коду та створення потрібних моделей та персонаж
4. Перелік графічного матеріалу_____
_____
_____
_____
_____

5. Консультанти по роботі із зазначенням розділів, за якими

здійснюється консультування:

Розділ	Консультант (прізвище, ініціали)	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Базурін В.М.	15.12.2022 р.	15.12.2022 р.
2	Базурін В.М.	15.12.2022 р.	15.12.2022 р.
3	Базурін В.М.	15.12.2022 р.	15.12.2022 р.

6. Зміст випускної кваліфікаційної роботи (проекту) (перелік питань за кожним розділом)

ВСТУП

РОЗДІЛ 1. СЕРЕДОВИЩЕ РОЗРОБКИ ГРИ

1.1. Аналіз проблем під час розробки

1.2. Unity

1.3 Огляд існуючих ігор на платформі Unity

РОЗДІЛ 2. РОЗРОБКА МОДЕЛІ ГРИ

2.1 Загальна концепція та алгоритми обчислень

2.2 Модель обчислення

2.3. Жанр гри та її види

РОЗДІЛ 3. КОМ'ЮТЕРНА ГРА НА ПЛАТФОРМІ UNITY

3.1 Розробка інформаційно-логічної моделі

3.2 Програмна реалізація гри

3.3 Технологія використання гри

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТКИ

7. Календарний план виконання роботи

№ пор.	Назва етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	
		За планом	Фактично
1	2	3	4
2	Вибір теми випускної кваліфікаційної роботи	04.10.2022	04.10.2022
3	Розробка та затвердження завдання на випускну кваліфікаційну роботу	15.12.2022	15.12.2022
4	Вступ	03.02.2023	03.02.2023
5	Розділ 1. Середовище розробки гри	28.02.2023	28.02.2023
6	Розділ 2. Розробка моделі гри	06.04.2023	06.04.2023
7	Розділ 3. Комп'ютерна гра на платформі Unity	12.05.2023	12.05.2023

8	Висновки	15.05.2023	15.05.2023
9	Здача кваліфікаційної роботи на кафедру науковому керівнику	30.05.2023	30.05.2023
10	Попередній захист випускної кваліфікаційної роботи	31.05.2023-01.06.2023	31.05.2023-01.06.2023
11	Виправлення зауважень, зовнішнє рецензування випускної кваліфікаційної роботи	02.06.2023	02.06.2023
12	Представлення готової зшитої випускної кваліфікаційної роботи на кафедру	05.06.2023	05.06.2023
13	Публічний захист випускної кваліфікаційної роботи	За розкладом ЕК	

8. Дата видачі завдання «30» грудня 2022 р.

Керівник випускної кваліфікаційної роботи (проекту) Краскевич В.Є.
(прізвище, ініціали, підпис)

Гарант освітньої програми Демідов П.Г.
(прізвище, ініціали, підпис)

Завдання прийняв до виконання студент-дипломник Іслам З. Р.
(прізвище, ініціали, підпис)

(прізвище, ініціали, підпис)

9. Відгук керівника випускної кваліфікаційної роботи (проекту)

_____
_____
_____
_____
_____
_____
_____
_____
_____
_____

Керівник випускної кваліфікаційної роботи (проекту)
_____ 30.05.2023 р.
(підпис, дата)

10. Висновок про випускну кваліфікаційну роботу

Випускна кваліфікаційна робота студента Іслам З.Р.
(прізвище, ініціали)
може бути допущена до захисту в екзаменаційній комісії.

Гарант освітньої програми_____ Демідов П.Г.
(підпис, прізвище, ініціали)

Завідувач кафедри _____ Пурський О.І.
(підпис, прізвище, ініціали)

«_____»_____2023 р.

Анотація

Метою роботи є створення прототипу тривимірної гри у жанрі платформер, розрахованого на одного користувача. Для досягнення

мети були проаналізовані сучасні тенденції у розробці комп'ютерних ігор, досліджені засоби розробки комп'ютерних ігор і проекти зроблені на рушії Unity. На основі цих елементів був створений прототип тривимірної гри у жанрі платформер, розрахованого на одного користувача, що містить в собі один рівень з реалізованою основною ігровою механікою. Результатом роботи є тестова програма в середовищі розробки Unity яка використовує розроблені скрипти, написані за обраним методом. Також є можливості для розширення функціоналу та напрямки для подальшого розвитку.

Ключові слова: Платформер, гра, розробка гри, ігровий рушій Unity.

Anotation

The purpose of the work is to create a prototype of a three-dimensional game in the platformer genre, designed for one user. In order to achieve the goal, modern trends in the development of computer games were analyzed, tools for the development of computer games were researched, and projects were made using the Unity engine. On the basis of these elements, a prototype of a three-dimensional game in the platformer genre was created, designed for one user, containing one level with implemented basic game mechanics. The result of the work is a test program in the Unity development environment that uses the developed scripts written according to the chosen method. There are also opportunities for expanding functionality and directions for further development.

Keywords: Platformer, Game, Game Development, Unity Game Engine

ЗМІСТ

Вступ

З 1947 до 1950 року, було представлено перший комп'ютер для розваги - Nimatron.Також,була написана перша в світі теоретична програма для гри в шахи на комп'ютері, хрестики-нулики,теніс та шашки.В основному це були пікселі на дуже маленькому екрані,попри гігантські розміри машини.Не кажучи також про величезну суму грошей,яку треба було на той час оплатити щоб мати ті “комп’ютери”.Тож,в основному,такі машини мали лише інститути,в яких вивчалися та досліджувалися комп’ютери - їх робота та можливості.Розробники того часу здебільшого прагнули показати можливості та процес роботи на той час комп'ютерів.Але з початку 60-х,коли деякі компанії почали бачити та бажали зробити з цього прибутковий бізнес у сфері розваг,почали з’являтися перші ігрові автомати,де за монети ви мали змогу зіграти у новітні на той час ігри,що були доступні вже для всіх бажаючих,на відміну від старих машин,на яких протестувати програми мали змогу тільки на виставках.

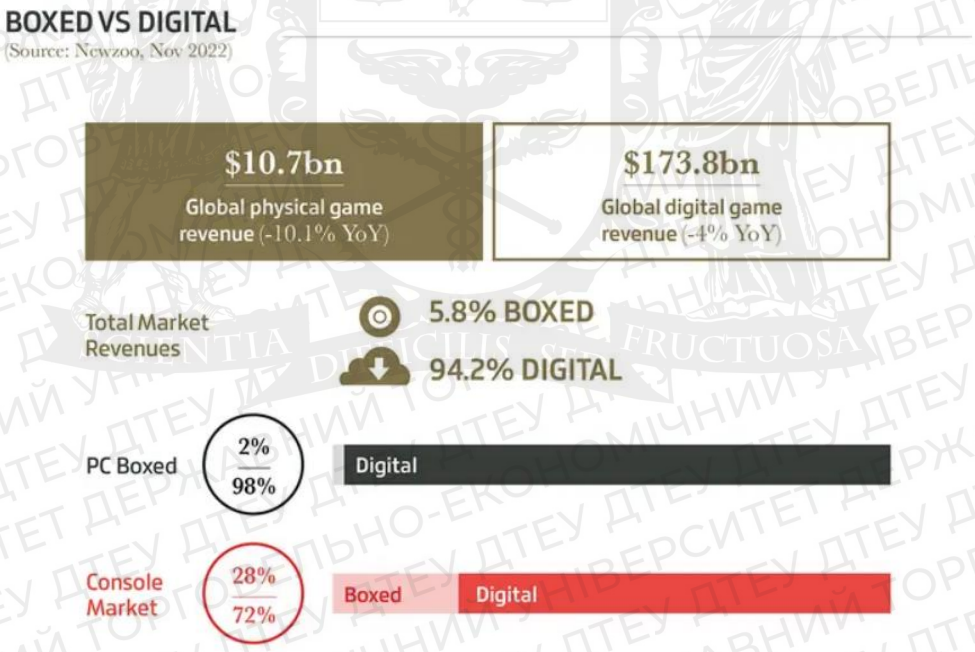
А з початку 70-х з’явилася перша домашня ігрова приставка з популярною грою pong та 11 різновидів гри у неї.Роками з'являлися усе більш нові домашні ігрові приставки,у тому числі портативні приставки, що потребували батарейки та давали змогу грати де завгодно, без потреби у постійному підключенні живлення до розетки,також нові автомати що мали змогу вже продемонструвати повноцінний кольоровий світ.Всі ці пристрої випускаються й до сьогодення,з новим більш сучасним нутром всередині,що дає можливість розробки нових ігор на них з новими та цікавими ігровими механіками.

З технологічним прогресом з'являлися нові пристрої,більш компактними та мають більше функцій,на відміну від тих гігантських комп'ютерів на початку їх зародження,та використовуються як для обчислень та вирішення тих чи інших сучасних проблем,так й для розваг.Комп'ютер став невід'ємною частиною життя майже кожної людини.Можливо,дехто не знає,але той же сучасний мобільний телефон є “кишеньковим комп'ютером”,в якому є можливість виконувати майже ті самі задачі,для яких потребувався персональний комп'ютер.

І як вже очікувано,на сьогодні існує доволі широка бібліотека,ігор на будь-який смак та на будь які пристрої,навіть старі ігри тих часів,що були на ігрових автоматах,ви можете відтепер завантажити за лічені секунди та зіграти на тому ж мобільному телефоні.

Можна сказати, що ігри - це певний вид мистецтва,що заповнюється своїми видовищними жанрами. Вони можуть нести не тільки розважальний характер, але і змушувати задуматися, переживати, піднімати серйозні глобальні або психологічні питання та дарують емоцій не менше, ніж перегляд кінофільму або театральної постанови.А їх розробкою займаються величезні команди,які іноді перевищують 400 осіб(включаючи сценаристів,дизайнерів,програмістів,акторів озвучення а також дубляжу,тощо),та мають бюджети на мільярди доларів.Дохід від ігрового блокбастеру може перевершувати збори від прокату фільмів,та й подарувати емоцій не менше них.Іншими словами це сучасний вид мистецтва,і як в кожному виді мистецтва в основному,вам потрібно закласти частину своєї душі,щоб зачепити та зацікавити споживача,що придбає копію вашої гри.

З потребою нових можливостей з’явилися потреба у відкритих ігрових рушіях,що дають змогу кожному програмісту можливість розробити та втілити у життя власну гру.Не одні лише ААА проекти від величезних компаній мали змогу отримати величезну популярність.Є також програмісти-любителі,або ж інді-розробники,які розроблюють невеликі програми, яким хоч не притаманна висококласна графіка, але вони мають нові та цікаві функції,що приносять деякі інновації в ігроладі,або мають цікавий сюжет та можуть грати на ностальгії гравця.Ці ігрові програми також мають велику популярність та приносять не малий дохід її авторам.



Статистика згідно якій,більший відсоток прибутку розробникам надходить саме від цифрових копій ігор[1]

Згідно підсумкам за 2022 рік,ринок ігор сягає 184 млрд доларів,де 50% займають мобільні ігри.Станом на сьогодні,згідно статистикам,найпопулярніша платформа серед геймдева в Україні залишається Unity(понад 69%).**Актуальність** теми даної роботи

полягає в тому, що з постійним розвитком IT, необхідно створювати нові програмні продукти, що будуть розвивати ігровий ринок та індустрію розваг в цілому. На сьогоднішній день є багато технічних можливостей для створення кросплатформних ігор, що дозволяє запускати гру на багатьох платформах,особливо у сфері ігрового ринку,який росте та розвивається швидше усіх.[1]

Також,має бути відповідь на дане запитання а саме:"Чому вона має дивну назву "Гра -Платформа"?". На це питання можна відповісти дуже коротко - власне бажання автора.Так,доволі дивна назва,але гра є лише демонстраційною - вона не буде розроблена повноцінно та викладена у публічній доступ.Це лише прототип,у якому автор реалізує власні можливості та аналізує у даній роботі функціонал сучасних ігрових рушіїв сьогодення.У даній роботі будуть розглянуті також певні теоретичні матеріали,наприклад: поняття жанру "платформер",його види та приклади ігор платформерів що були розроблені на даному рушії.

Метою нашого дослідження є створення і написання демонстраційної гри "Гра-платформа" із жанром платформер на ігровому рушії Unity.

Об'єктом роботи є гра-прототип на рушії Unity.

Предметом роботи виступає процес розробки гри.Концепти,написання коду,розробка потрібних моделей предметів та персонажа.

Практичне завдання дослідження полягає у аналізі теорій, реалізації прототипу та перевірки тез,що Unity є кращим варінтом для новачка у виборі рушія для розробки ігор.

Структура роботи: Дана робота складається зі вступу, трьох розділів основної частини, висновків, списку використаної літератури та додатків.

РОЗДІЛ 1. СЕРЕДОВИЩЕ РОЗРОБКИ ГРИ

1.1. АНАЛІЗ ПРОБЛЕМ ПІД ЧАС РОЗРОБКИ

Як вже було згадано раніше,"нинішні відеоігри - це вид мистецтва",і як у кожного виду мистецтва - чудова гра іноді також потребує "крові та поту".

Працюючи у компанії,вам та вашій команді(якщо ви працюєте над проектом не один) потрібно розробити "прототип" проекту та продемонструвати видавцю чи інвесторам.Якщо їм сподобається ідея - отримати фінансування та почати повноцінну розробку гри.Також,інвестори чи видавець,може потребувати додати чи зробити деякі зміни у грі,адже ви робите гру за рахунок їх фінансування.

Здавалось би,маючи знання програмування,робота над грою займе не так багато часу та взагалі це легке діло. Новачок у цій сфері може вважати,що маючи готовий рушій робота буде доволі швидкою та без проблем. На справді не так. Рушій,як правило,"відбирає лише десяток валіз із сотні,яку потрібно нести".

Недооцінка є серйозною проблемою проекту. Наприклад: планування пов'язані з оптимістичною оцінкою, як правило, через ті завдання, які розробникам здавались легкими та швидкими,тим самим виділяючи менше часу та увагу до даних елементів,що могло понести за собою серйозні наслідки або й скорочення контенту та ідей,для того,щоб встигнути за графіком. Вони повинні виділяти правильно час,щоб зробити «все правильно» та домогтися більш «солідної гри». Таким чином, цілі і терміни виконання повинні бути визначені достроково і, при необхідності, часто переглядатися під час виробництва. Крім того, вони повинні витрачати більше часу на

оцінку ризиків під час підготовки до виробництва, виділяючи більше часу на кожну деталь гри.[2]

Закон Гофштадтера стверджує, що «це завжди займає більше часу, ніж ви очікуєте».Оцінка програмного забезпечення є добре вивченою сферою, яка використовує попередні дані для оцінки зусиль (і вартості) проекту. Він використовує різні методи, як-от COSMIC або Agile' Story Points, і методи, як-от аналіз експертів, машинне навчання тощо. Проте оцінка програмного забезпечення в розробці ігор часто виконується вручну з використанням досвіду старших розробників. Розробники, особливо старші, окрім використання своїх досвіду, також повинні документувати свій досвід на майбутнє, наприклад, для створення моделей ML. Оцінка гри, як і для будь-якого іншого програмного проекту, різниться в різних ігрових проектах. Команди переходять від однієї гри до іншої та повинні адаптуватися до технологічного прогресу та різних вимог, що ускладнює оцінювання. Вони повинні інвестувати в довгі етапи підготовки до виробництва, щоб дослідити та зрозуміти нові технології, інструменти та дизайн ігор. Краща оцінка приходить із кращою інформацією про попередні проекти. Проте закритий характер більшості ігор ускладнює обмін інформацією. Посмертні дослідження є важливим, але недостатнім джерелом інформації. Розробники повинні збирати метадані про минулі ігрові проекти та застосовувати/розширювати традиційні методи оцінювання.[3]

Нечітке бачення ігрового дизайну впливає на весь ігровий проект, включаючи управління та тестування. Незважаючи на те, що бачення ігрового дизайну пов'язане з ігровим дизайном і мистецтвом, його має сприймати вся команда, і, отже, це також проблема управління. Команди повинні розуміти бачення проекту, щоб уникнути марної роботи. Їм слід витрачати менше часу на

визначення статичних документів, які швидко застарівають, і більше часу на підготовчу роботу, доки не стане ясною основна механіка та цікавий фактор. Вони повинні створити прототип і відтворити. Нарешті, вони повинні тримати творчий контроль над проектами, наскільки це можливо.

1.2. UNITY

Unity - це ігровий рушій, що дозволяє створювати гри під більшість популярних платформ. За допомогою цього рушія можна розробити ігри на персональних комп'ютерах (які працюють під Windows, MacOS, Linux), на смартфонах і планшетах (iOS, Android, Windows Phone),та ігрових консолях (PlayStation 3-5, Xbox(One,Series X|S),Nintendo Wii,WiiU,Switch,Google Stadia).[4]

Був випущений у 2005 році,спочатку як ігровий рушій для пристроїв на Mac OS,але згодом з явилася можливість розробляти ігри й у користувачів Windows,Linux, а також на ігрові консолі та мобільні телефони.Більшість мобільних ігор у які ви грали або граєте були розроблені як раз на цьому рушії.У вас є можливість розроблювати як 3д,так і 2д ігри.Функціонал програми легкий у вивчені,а згодом,у користуванні.Unity має такі особливості як:

-Простий робочий процес, що дозволяє розробникам швидко збирати сцени в інтуїтивно зрозумілому робочому просторі редактора.

-Спеціальні інструменти для створення 2D- і 3D-ігор зі спільними правилами, які полегшують роботу розробникам.

-Дуже унікальна та гнучка система анімації для створення якісної анімації за дуже короткий час.

-Можливість скоротити час розробки, використовуючи вже створені ресурси, доступні у величезному Asset Store.

Ще одна його перевага це поєднання редактора сцен (в комплексі загального редактора) з редактором ігрових об'єктів і редакторів скриптів.Додатково додаються генератори дерев і террейнів.Більш кращі можливості скриптингу.

Для написання коду ігрових програм на рушії Unity вам в основному потрібні знання C#,але також є підримка JavaScript.Після встановлення,Unity може автоматично знайти та використовувати Visual Studio під час написання скриптів.Якщо з певних причин цього не сталося,ви можете вказати у налаштуваннях.Під час написання скриптів,ви матимете готову колекцію команд рушія,що автоматично з'являтимуться - достатньо лише написати пару літер.Це дуже корисна функція та спрощує роботу у написанні коду.

Як для кожного ігрового рушія - чим краще залізо у комп'ютера тим зручніше та легше користуватися Unity.Але точні мінімальні вимоги для Unity,ви можете знайти на офіційному сайті рушія.

This section lists the minimum requirements to run the Unity Editor. Actual performance and rendering quality might vary depending on the complexity of your project.

Minimum requirements	Windows	macOS	Linux
Operating system version	Windows 7 (SP1+), Windows 10 and Windows 11, 64-bit versions only.	High Sierra 10.13+ (Intel editor) Big Sur 11.0 (Apple silicon Editor)	Ubuntu 20.04, Ubuntu 18.04, and CentOS 7
CPU	X64 architecture with SSE2 instruction set support	X64 architecture with SSE2 instruction set support (Intel processors) Apple M1 or above (Apple silicon-based processors)	X64 architecture with SSE2 instruction set support
Graphics API	DX10, DX11, and DX12-capable GPUs	Metal-capable Intel and AMD GPUs	OpenGL 3.2+ or Vulkan-capable, Nvidia and AMD GPUs.
Additional requirements	Hardware vendor officially supported drivers	Apple officially supported drivers (Intel processor) Rosetta 2 is required for Apple silicon devices running on either Apple silicon or Intel versions of the Unity Editor.	Gnome desktop environment running on top of X11 windowing system, Nvidia official proprietary graphics driver or AMD Mesa graphics driver. Other configuration and user environment as provided stock with the supported distribution (Kernel, Compositor, etc.)
	For all operating systems, the Unity Editor is supported on workstations or laptop form factors, running without emulation, container or compatibility layer.		

Системні вимоги для Unity Editor[5]

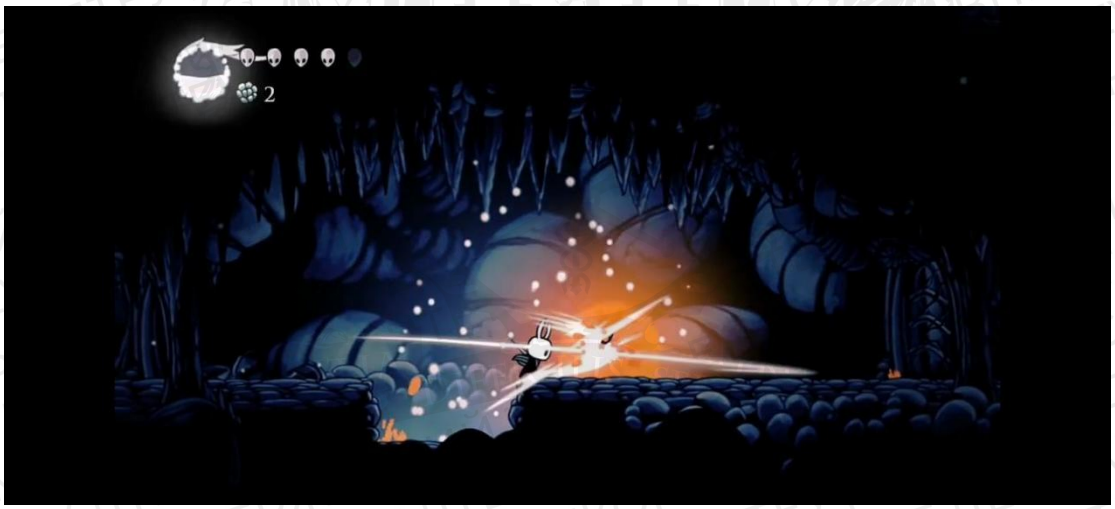
1.3. ОГЛЯД ІСНУЮЧИХ ІГОР НА ПЛАТФОРМІ UNITY

На рушії Unity було розроблено немалу кількість відомих ігор як на ПК,ігрові консолі,також й на мобільні телефони.Наврядчи вас цікавило чи ви знали,що майже 50% мобільних ігор в основному розроблені на рушії Unity.Такі відомі ігри які ви грали(або точно знаєте про них) як Hearth Stone,Subway Surfers,Temple Run,Angry Birds,Pokemon Go,тощо.[4]

Так,як у даній робота розробляється гра у жанрі "платформер",тож приклади тут будуть саме платформери розроблені на рушії Unity ,вони здобули велику популярність та рекомендуються гравцями й по сьогодення.

1)Hollow Knight - це гра жанрів platform / adventure, розроблена і видана Team Cherry.Сама гра є метроїдванією(це піджанр пригодницького бойовика що має механіку гри як у ігр серії metroid та castelvania),та сайд-скролер, дія якої відбувається в Геллоунесті, вигаданому підземному королівстві. Під час дослідження підземного світу гравець керує мовчазним безіменним «Лицарем», схожим на комаху. Лицар володіє типом меча, який називається цвяхом та який використовується як у бою, так і у взаємодії з навколишнім середовищем.У більшості областей гри гравці стикаються з ворожими жуками та іншими істотами. Рукопашний бій передбачає використання цвяха, щоб вражати ворогів з короткої відстані. Гравець також може вивчати заклинання, що дозволяє здійснювати далекі атаки. Переможені вороги викидають валюту під назвою Geo. Лицар починає з обмеженою кількістю очок життя, які представлені масками.Так звані «Осколки маски» можна збирати протягом усієї гри, щоб збільшити максимальну кількість масок гравця. Вражаючи ворогів, лицар отримує душу, яка зберігається в посудині душі. Якщо всі маски втрачено, Лицар гине, а ворог Тіні з'являється там, де вони

загинули. Гравець втрачає весь Geo і може мати зменшену кількість Soul. Гравцям потрібно перемогти ворога Тіні, щоб повернути втрачену валюту та мати звичайну кількість душі. Гра продовжується з останньої відвіданої лавки, на якій сидів персонаж, які розкидані по всьому ігровому світу та діють як точки збереження та місця, де гравець може змінити свої чари. Спочатку гравець може використовувати Душу лише для «Фокусування» та регенерації масок, але в ході гри гравці відкривають і збирають кілька наступальних заклинань, які споживають Душу. Додаткові судини душі, які використовуються для зберігання більшої кількості душ, можна отримати протягом гри.



Hollow Knight

2)Cuphead – це платформер, який використовує механіки жанру «run and gun». Головний герой гри Капхед не ставить перед собою завдання порятунку світу, він просто вибиває весь дух зі своїх ворогів і намагається повернути програну у Диявола ставку. Гра володіє цікавою стилістикою, як у мультфільмах 1930-х років.

Розробкою Cuphead займалася студія StudioMDHR Entertainment, яка акцентує свою увагу на інді-проектах. Також вони

взяли на себе роль видавця. Під час створення гри, розробники надихалися мультфільмами від студій Fleischer Studios, Disney, а також мультиплікаторами Абом Айверксом, Грімом Натвіком та Уіллардом Боускі.

Це перша гра розробників та свій початок розробки гра має ще у 2010 році.Вперше гру показали у 2014 році.У цей же рік планувався реліз,але його відклали через бажання розробників довести гру до ідеалу.Очікування виправдані - гра вийшла дуже чудовою,дуже незвичною завдяки стилю мультів старих років та ,місцями, дуже важкою,через важких босів,що не дають вам шансу видихнути.

Cuphead

3)Ori And The Blind Forest - ще одна чудова гра,випущена у 2015 році.На початку вам могло б здатися що це короткий, приємний артхаусний платформер, де ігровий процес існує в основному для

підтримки розкішної графіки та емоційної тяги історії. Але Ori - це вимоглива, хитро заплутана пригода в стилі Metroidvania, в якій швидкий тригерний палець і ідеальний час мають значення майже так само, як і вивчення його середовищ.

Ori and the blind forest

4)Fall Guys - онлайн гра у жанрі платформер батл рояль.Ви граєте за створінь,схожих на боби з ручками та ніжками,та ваша задача - пройти усі перешкоди,фінішувати у топі,вижити та залишитися єдиним з 60 гравців.У ваших можливостях є не тільки біг та стрибок - ви можете схопитися за суперника та викинути його з поля гри.

”Battle Royale”(переклад з англійського “королівська битва”) - це жанр відеоігор, який поєднує в собі елементи виживання,де переможцем є остання людина,що вижила. Це популярний жанр відеоігор, який набув великої популярності в останні роки.

Fall Guys Ultimate Knockout

5)My Friend Pedro - 2.5д гра у жанрі platformer-shooter.Розроблена DeadToast Entertainment та видається Devolver Digital.

Ви граєте за мовчазного хлопця у масці,заради свого божевільного друга-банана відправляється убивати “паганих хлопців”.Як очікувано від такого роду ігор - усе це подається у трешевій та гумористичній формі.

Геймплей базується на проходженні рівнів і вбивстві ворогів.Варіантів вбивства противників предостатньо: стрілянина з двох рук, звисаючи на мотузці і в польоті, в стрибках від стіни, використовуючи підручні предмети для відволікання і вбивства ворогів, використовуючи рикошет. При цьому можна використовувати комбінацію подвійного націлювання й уповільненого руху. Гравці отримують бонусні очки, за будь-які з додаткових трюків.

Присутні також погоні на різному транспорті. У грі реалізована незвичайна механіка стрілянини – стріляючи з двох пістолетів, гравець керує двома прицілами одночасно, і сам вирішує куди стріляти. Приціл стабільний поки не буде натиснута спеціальна кнопка, після чого основний приціл фокусується в обраній точці, а другий можна вільно обертати як вам завгодно. При натисканні на спусковий гачок, стрілянина відбувається відразу з двох знарядь, спрямованих в різні боки.

Також,гра має автоматичний генератор GIF-зображень, ними можна буде ділиться в соціальних мережах демонструючи найцікавіші моменти в ході проходження.

My friend Ped

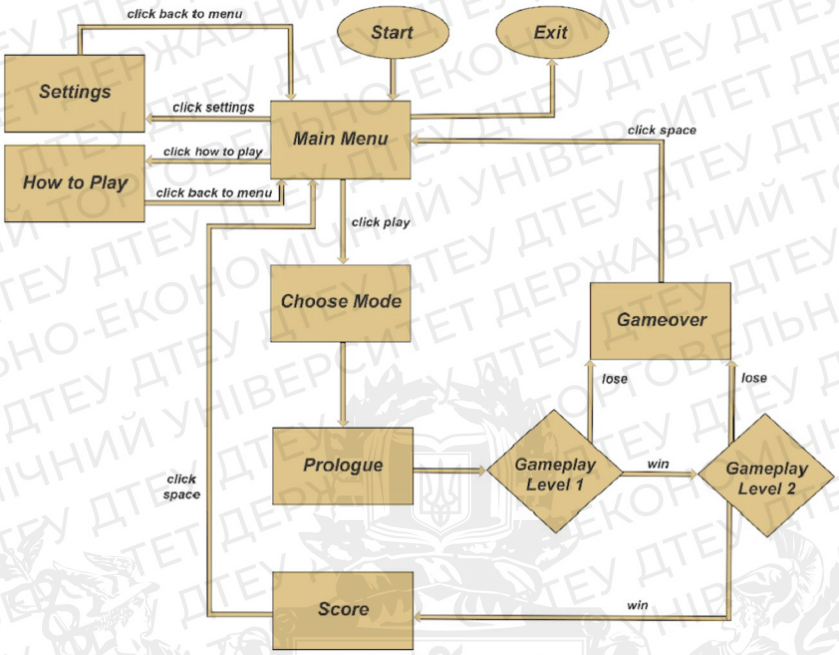
РОЗДІЛ 2. РОЗРОБКА МОДЕЛІ ГРИ

2.1. АЛГОРИТМИ ОБЧИСЛЕНЬ

Алгоритми - це набір інструкцій, що описують порядок дій виконавця для досягнення результату розв'язання задачі за скінченну кількість дій; система правил виконання дискретного процесу, яка досягає поставленої мети за скінченний час. Для її візуалізації часто використовують блок-схеми.[6]

Цей термін був започаткований ще у 12 столітті перським математиком Аль-Хорезмі.Є припущення,що слово є невдалим перекладом імені самого математика.У 19 столітті,британський математик Ада Лавлейс розробила першу в історії програму,реалізовану для виконання на електронно-обчислювальних машинах.А з початку 1930 почався бурхливий розвиток,завдяки якому з’являються нові технології які лягли в основу інформатики.[6]

Про це можна розписати одну величезну статтю,але зупинимося на цьому.Алгоритми та блок-схеми потрібні для зображення вже повної та логічної картини певних части нашої гри,як наприклад .



Приклад блок-схеми гри[7]

Для даного проекту було створено дві блок-схеми: блок-схема гравця та ворога(додаток 2.1.1 та додаток 2.1.2).Ці блок-схеми дають нам можливість розуміти що саме ми хочемо розробити,у якій послідовності це все має працювати та що будемо писати у коді цих об’єктів.

Основна функція гравця у платформерній грі - можливість керувати ним,тоді як для ворога - бути перешкодою для гравця у досягнені фінішу та проходження рівня.Для обох об’єктів маємо вказати у коді їх швидкість(Accelaration).Вона потрібна для регулювання швидкості об‘єктів.

Забіжимо наперед: рушій Unity має можливість показувати ті елементи,що вказані публічно у скрипті,при умові що скрипт прив’язаний до об’єкту та регулювати їх під час тестового запуску проекту,що в свою чергу допомагає у таких ситуаціях.Окрім швидкості та силі стрибку,ми можемо також змогу розписати у скрипті позначення як кількість здоров’я,сила удару,перевірити умову як наприклад “чи видимий гравець ворогу”,тощо.

Як можемо побачити по блок-схемі гравця(додаток 2.1.1),він має також так звану “Силу Стрибка”(Jump Force).Його ми будемо використовувати для регулювання стрибку гравця,тобто чим вище показник - тим вище стрибатиме гравець.

Для ворога розроблена блок-схема більш-менш проста та невеличка.Для початку потрібно щоб скрипт ворогу без усіляких проблем працював під час запуску та переслідував гравця.Тому запропонована така ідея - у його випадку було вписано дві умови - “патрулювання” та “переслідування”.Для “патрулювання” потрібно створити окремий скрипт та об’єкт,що виступатиме як “patrol point”(точка патрулювання),за якою ворог буде спочатку слідкувати.Досягнувши її,точка на тому місці зникне та рандомно з‘явиться у іншому місці.Ворог після доторкання до тієї точки стоятиме 5 секунд а потім слідкуватиме за тою,що з’явилась у іншому місці.Таким чином буде розроблене так зване “патрулювання”.У другій умові,якщо гравець потрапить у радіус(об’єкт,що буде навколо ворога,це може бути як невидима сфера у якій знаходитиметься ворог,або великий прямокутник),ворог перерве патрулювання та слідкуватиме за гравцем до поки той не зникне у полі зору,яким виступатиме “радіус”.Таким чином ми розробимо штучний інтелект ворога на початковій стадії.

Початкове зображення усіх ідей або послідовностей роботи програм у схемах є корисним та важливим елементом під час розробки будь-якого програмного забезпечення.Цим не можна нехтувати під час розробки проектів,адже ви маєте розуміти хоча б логічну роботу того чи іншого об’єкту.

2.2. ЗАГАЛЬНА КОНЦЕПЦІЯ

Як зазначалося раніше - розробка відеогри є доволі довгим та виснажливим процесом.Здебільшого,вона потребує команду з фахівців,де кожен виконує своє спеціалізоване завдання.

Етап розробки гри ми можемо розділити на 4 основні частини:концепція,підготовка розробки,розробка та постпродакшен. Тож повернемося до концепту.

Концепція гри,а якщо казати терміном у анатомії живих істот є мозком.Так ви матимете кістки - рушій,шкіра - моделі,м’ясо та м’язи - скрипти,а також сердце - сценарій.Але як без мозку так і без концепту у грі це все не буде функціювати.Тому визначення що з себе буде представляти гра є важливою частиною у розробці відеогри.Для створення концепту є потреба ідеї.У данному випадку ідея є основою концепту.Розробники можуть черпати натхнення з будь чого та з будь звідки.Сама ідея гри може також мінятися протягом розробки гри.

Прикладом можна виставити гру Resident Evil 4(2005),яка нещодавно отримала ремейк(тобто оновлена та перевидана версія відеогри,що має ті самі аспекти оригінала,але з покращення деяких з них),що вийшов 24 березня.Ще задовго до виходу 11 січня 2005 року,свій початок розробки вона почала у 1999 році та за час розробки мала більше одної ідеї,що змінювалися роками.Наприклад,першою ідеєю гри був проект з назвою “туман”/“замок”,що був представлений у 2002 році,де гравцеві потрібно було досліджувати замок та тікати від істоти у вигляді чорного диму.Розписувати детально про ці ідеї буде доволі недоречно,але додам,що частина того списку відхилених ідей та концепцій для цієї гри були повторно використані розробниками з

Капкому та сприяли появі абсолютно нових серій ігор,що також здобули велику популярність.[8]

Джерелом натхнення для даного проекту є три гри.Усі вони мають спільний елемент - фіксована камера.Тобто у гравця нема можливостей нею керувати та вона не крутиться навколо гравця.Рівні у грі це поєднання 2д-скролінгу з вільним 3д пересуванням по локації.

Гра Super Mario 3D World.Це гра у жанрі платформер,є шостою 3д грою у серії та бцла випущена на домашні консолі Nintendo Wii U у 2013 році.Геймплей представляє зз себе величезний лінійний рівень,де ціль гравця дійти до фінішу.Також є елементи від старих 2д ігор про Маріо,як наприклад обмеження у часі та величезний флагшток,що слугує фінішом.[9]

Super Mario 3D World

Це не єдина гра серед платформерів,що має фіксовану камеру.Перша трилогія Crash Bandicoot є платформером з піджанром пригодницький бойовик.Подібно Super Mario 3D World,дана трилогія є 3д іграми,що мають фіксовану камеру та лінійні

рівні.Гравець керує головним персонажем Креш,який повинен пройти всі рівні та перемогти злого вченого Нео Кортекса.Певні рівні, які вимагають від нього тікати від валуна, що котиться, змінюють цю перспективу, тоді як на інших рівнях відтворюється традиційна перспектива бічної прокрутки. Креш може рухатися в усіх напрямках; окрім руху ліворуч і праворуч, він може відходити від гравця або до нього, а елементи керування не змінюються залежно від його позиції. Його дві основні форми атаки,подібно Маріо,полягають у стрибках на ворогів і виконанні обертової атаки, яка викидає ворогів з екрана. Підбиті вороги можуть зачепити та завдати ударів іншим ворогам, які зараз знаходяться на екрані.[10]

Crash Bandicoot 1

Також можна додати сюди перші ігри Lego,як Lego Indiana Jones,Lego Star wars 1 та 2.Усі ці ігри є також платформерами з піджанром пригодницький бойовик.Усі грі слідують події у фільмах, хоча деякі сцени з фільму були змінені, щоб знизити вікову категорію та дозволити спільну багатокористувацьку гру,як битви з

босами,або просто надати гравцеві комічне полегшення у вигляді комічних катсцен.[11]

Lego Indiana Jones The original adventure

Це не єдині ігри,але саме вони надихнули до розробки та є концепцією даного проекту.У висновку можна зазначити,що ціль та концепція гри - розробити трьохвимірний платформер з фіксованою камерою,що показує вид зверху.

2.3. ЖАНР ГРИ ТА ЇЇ ВИДИ

Коли розробник береться за ідею гри,він у першу чергу,після ідеї, визначає її жанр.Без визначення жанру ти не зможеш розробити гру.Наприклад: якщо ідея гри полягає у рукопашній битві між гравцями у арені - її жанр визначається як файтинг.Якщо у розгадані логічних завдань - гра-головоломка.

Визначення **жанру відеогри** використовується для класифікації відеоігор відповідно до інтерактивних ігрових дій гравця. Він не залежить від сценарію або змісту уявного ігрового

світу, на відміну від творів літератури чи кіномистецтва. Єдиної класифікації жанрів відеоігор не існує, тому в різних джерелах одну й ту саму відеогру можуть відносити до різних. Також можливі змішання жанрів, коли гру неможливо віднести до одного жанру.

Попри це, розробники відеоігор послуговуються усталеними поняттями про різновиди відеоігор (не завжди жанрові), що дозволяє з анонсів при розробці та у рекламно-маркетингових кампаніях повідомити гравцям суть гри.

Як уже раніше зазначалось,мета цієї роботи - розробити демонстраційну гру “Гра-платформа” на рушії Unity.Як можна уже зрозуміти, жанр гри є action-adventure(пригодницький екшн) платформер.Але є потреба у аналізу що з себе представляє даний жанр відеоігор,перш ніж перейти до практичної частини.

Платформер (англ. Platformer), також відомий як Платфо́рмна гра (англ. Platform game) - жанр відеоігор, ігровий процес в якому складається зі стрибків персонажа по різноманітним платформам (звідси й назва) та через перешкоди, збирання предметів, зазвичай необхідних для проходження рівня.[12]

На початку зародження,ігри жанру платформер було важко назвати платформером.Спочатку вони були обмежені статичним екраном а ваша ціль в основному це набрати більше балів.На приклад,перші грі які вважають зародженням платформ виходили спочатку на аркадні автомати.Їх задача в основному поляга у набиранні більшої кількості балів у грі та побиття рекоду,наприклад: Space Panic(1980) де ваша задача заключається у проходжені рівнів, шляхом знищення усіх прибульців на рівні та набрати найбільшу кількість балів.Згодом,з прогресом у ігрових консолях та можливістю показувати не різнокольорові квадрати а повноцінний мальовничий світ,хоч і у 8-біт,більша частина розробок відеоігор

перейшла саме до домашніх консолей.Можна сказати з упевненістю,що більша частина ігор на тій же 8-бітній консолі Nintendo Entertainment System - це саме сайд-скролерні платформери.[14]

Дуже величезна кількість платформерів вийшла за увесь час,через що їх почали поділять на види та піджанри:

Двухвимірний платформер - перший вид платформерів.У традиційних двовимірних платформерах персонаж рухається зліва направо, деякі ігри мають взагалі вид зверху,що дозволяє рухатися у будь які напрями.Є також ігри або моменти,де рух можливий тільки вперед і не залежить від бажання гравця (автоскроллінг).

Тривимірний платформер з'явився завдяки зростанню потужності ігрових консолей та аркадних машин.У більшості двовимірних платформерів гравцеві потрібно було досягти на рівні тільки однієї мети - дійти з точки А до точки Б.Однак у багатьох тривимірних платформерах,кожен рівень може бути як одною тропою,так і великим світом,де вам потрібно прочісувати кожен куточок для отримання бонусів за повноцінне проходження гри, збираючи фрагменти головоломок, зірки чи предмети для прокачки персонажа.[13]

Звісно не треба також забувати про ігри з відкритим світом,де у вас була можливість не йти по одній лише дорозі(лінії) задля проходження гри.Гарним прикладом є гра Spyro the Dragon (1998),що мала нелінійне проходження і справжнє вільне переміщення в трьох вимірах в рамках рівня.Звісно ж,сюди можна було би включити метроїдванію,але це "жанр відкритого світу" поки виключно для ігор з 2х або 2.5 виміром.

З часом,з'явився також мікс 2д та 3д платформеру - **2.5D платформер**,суть якого полягає у переміщенні персонажа в одній

площині та вид збоку, але сам рівень та моделі є тривимірними.Прикладами таких ігор є вище згаданий My Friend Pedro,Megaman X8 та Sonic 4 episode 1 та 2 та ще купа інших ігор.[16]

Megaman X8(2004)- приклад того,як зазвичай виглядає 2.5D гра

Як кожен жанр у відеоіграх,жанр платформер має також так звані піджанри.

Нескінченні ранери - гра платформер,де персонаж є у постійному русі по “нескінченному” світі гри.Управління гри обмежене можливістю лише стрибка персонажа, атаки, або виконання спеціальних дій. Мета цих ігор - пробігти якомога далі, перш ніж персонаж помре. Ігри з цим жанром добре підходять для невеликого набору елементів управління, частіше - обмежується одним натисканням екрану для стрибків. Нескінченні бігалки зазнали особливого успіху на мобільних платформах. Особливо на

рушії Unity,адже більша частина таких ігор зроблена саме на ньому,як Temple Run,Cookie Run,Subway Surf та ін.[15]

Також,не можна не згадати ще платформери з під-жанром **метроїдванія**.Ці ігри зазвичай мають велику взаємопов'язану та величезну карту світу, яку гравець має дослідити.Здебільшого,деякі частини світу будуть недоступні гравцеві на початку гри,до поки він не виконає певні умови(поки не переможе босса,не знищить споруду/об’єкт або поки не придбає спеціальні предмети, інструменти, зброю, здібності або знання у грі). Придбання покращень(апгрейдів) також може полегшити гравцеві проходження гри,з легкістю здолати більш складних ворогів або знайти ярлики та секретні області.Гравець також часто у таких іграх має змогу відстежувати свої кроки по карті.[17]

Також,окрім,метроїдванії та ранерів,існують ще такі піджанри як **біжи та стріляй**(run-and-gun platform),**платформи-головоломки**(puzzle-platform),**кінематографічний платформер**(),**комічні екшн-ігри**(comical action game),**ізометричні**(isometric platform) та **пригодницькі**(platform-adventure).Це ще не весь список,адже піджанри можуть також зливатися між собою або з іншими жанрами,як наприклад платформер-горор (horror-platformer) та пригодницький екшн(action adventure).

З пояснення можна тепер зазначити,що наш проект матиме action-adventure жанр,подібно іграм,зазначеним у концепції.

ДТЕУ ДТЕУ ДТЕУ

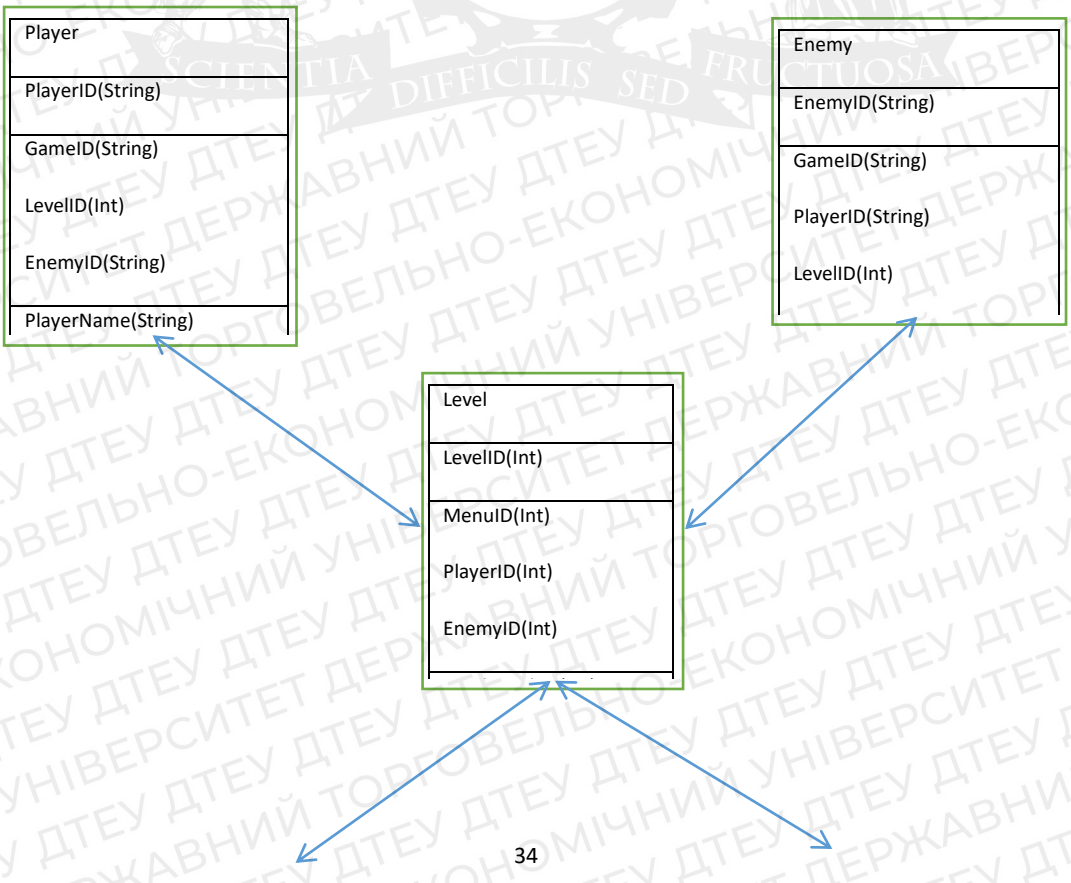
РОЗДІЛ 3. КОМП'ЮТЕРНА ГРА НА ПЛАТФОРМІ UNITY

3.1. РОЗРОБКА ІНФОРМАЦІЙНО-ЛОГІЧНОЇ МОДЕЛІ

Перш ніж почати програмну реалізацію розробимо останню модель для нашого проекту.Вона також є важливою складовою у будь-якому проекті,та потрібна для більш кращого розуміння що ми хочемо у даному проекті.

Інформаційно-логічна модель(ілм) це сукупність інформаційних об'єктів та зв'язків між ними.Її розроблюють ще на початкових стадіях проекту,де узагальнюють основні властивості та поєднують елементи що будуть пов'язані між собою.

Для створення ілм,нам потрібно виконати основні етапи,а саме збір інформації про об'єкти,формулювання знань та структуризація у схемах для подальшого його використання.



модель гри

Дана схема є інформаційно-логічною моделлю нашої гри,яка складатиметься з 5 об'єктів: головного меню,нашого рівня,гравця,ворога та гри взагалом.Кожен з цих об'єктів має власні певні набори параметрів,які описують певну частину області яку представляє об'єкт.Також кожен об'єкт має ключові параметри.Кожен ключовий параметр виступатиме в ролі простих або складних ключів відповідних таблиць.

Наприклад - гравець пов'язаний з грою,рівнем та ворогом.Швидкість,сила стрибка та здоров'я гравця - це дані що будуть розписані у скрипті гравця під час програмної реалізації гри.

Майже кожен об'єкт у інформаційно-логічної моделі буде зв'язаний з іншим по деяким параметрам.Ці зв'язки мають показати які та як саме зв'язані між собою об'єкти та які спільні властивості їм притаманні.

Ще одним етапом підготовки до створення проекту є ER-діаграми.Це модель даних,що описує концептуальні схеми за допомогою узагальнених блоків.Її зручно використовувати під час проектування інформаційних систем.Це одна з найпростійших візуальних моделей,що дозволяє описати структуру моделі у загальних рисах.

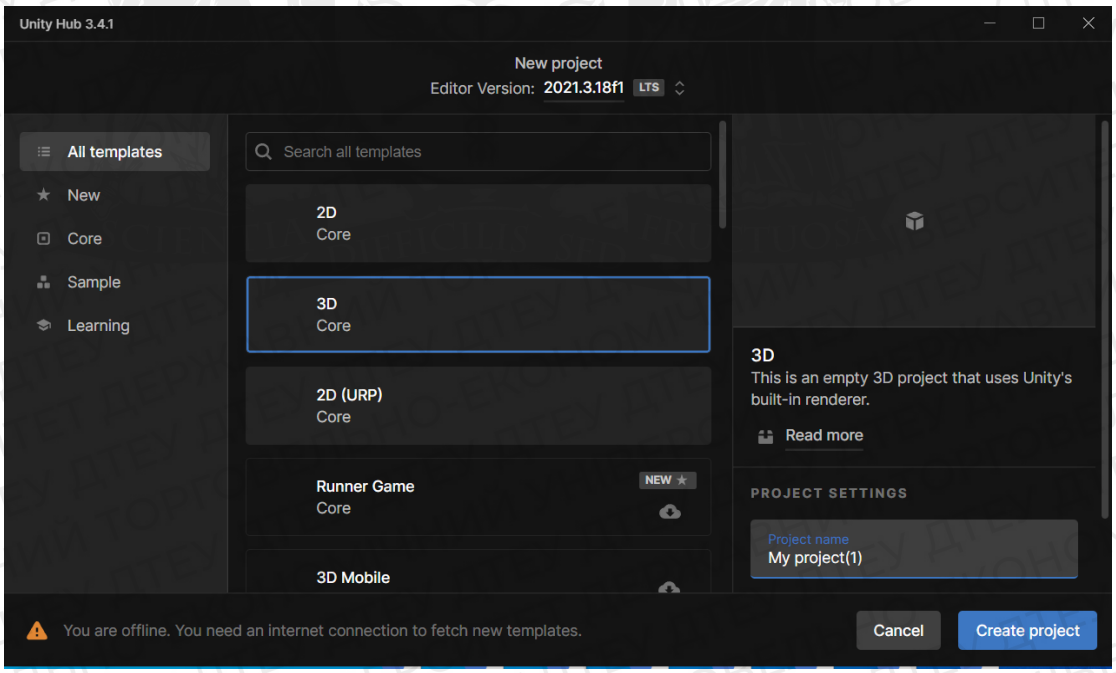
Звісно,під час розробки повноцінної гри,розробники створюють не одну інформаційно-логічну модель,але ми

обмежимося лише одною моделлю,оскільки вже маємо блок-схеми,де розкладали усі потрібні дані для даного проекту.

3.2. ПРОГРАМНА РЕАЛІЗАЦІЯ ГРИ ТА ЇЇ ТЕСТУВАННЯ

Тепер до головного.

Маючи вже початкову ідею,поняття що ми бажаємо зробити та схеми,ми можемо нарешті перейти до програмної реалізації проекту за допомогою ігрового рушія Unity.[18]

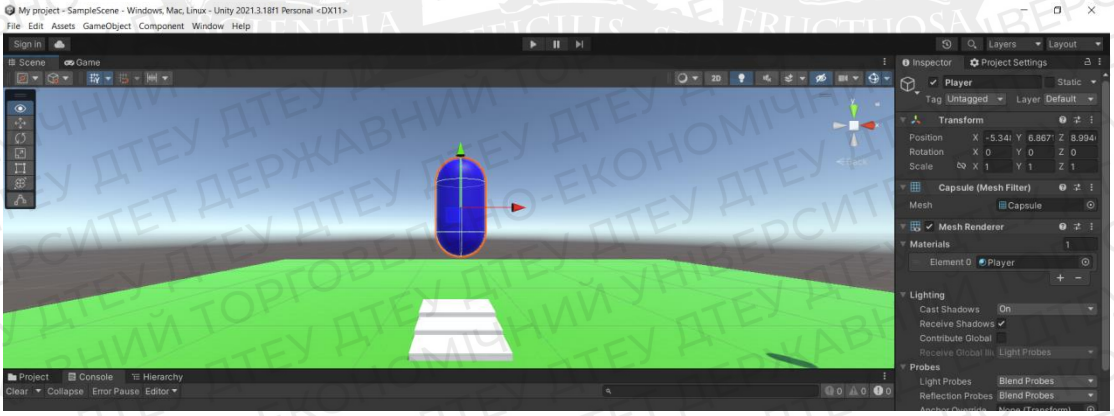


Обираємо ядро та назву проекту і починаємо

Почнемо з основи будь якого платформеру - гравця та платформ.При створенні проекті у вас автоматично створюється

сцена та додаються основні об’єкти: головна камера та світло.Це є основою у будь-якому ігровому рушії.Подальші вже дії зі сценої, додавання або видалення певних об’єктів звідти вже по вашому бажанню.Навіть якщо ви,наприклад,випадково видалите світло у сцені,ви можете його легко повернути за допомою славнозвісної комбінації Ctrl+Z,що є стандартної комбінацією для повернення до попереднього стану.Або ж через вкладу Game Object,про який ми зараз і поговоримо.

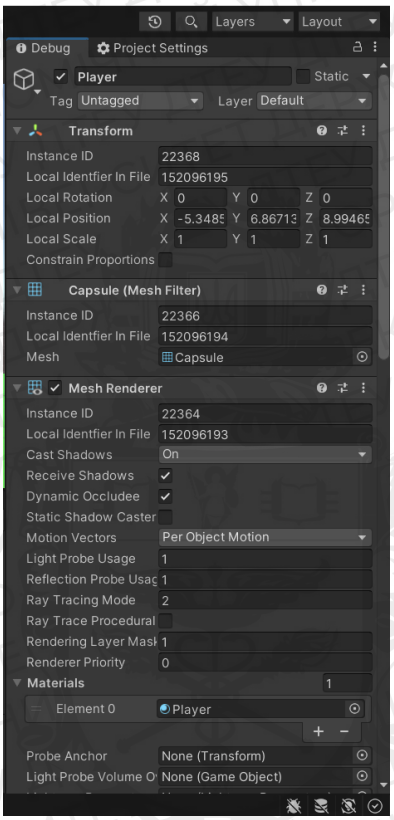
Для початку ми йдемо в Game Object > 3D Object та обираємо будь який об’єкт,що буде нашим персонажем гри.Куб,шар,капсула - що завгодно може бути “гравцем” на початку розробки гри.Додаємо також інші об’єкти: об’єкт plane(площа,яка виступатиме основним полем),cubes (куби,що виступатимуть платформами) а також решту інших об’єктів на ваше бажання.



Наш гравець

В інспекторі ви можете редагувати характеристики об’єктів,змінювати їх розташування,положення,колір,місце та розмір,додавати до них компоненти,текстуру,скрипти,тощо.Також

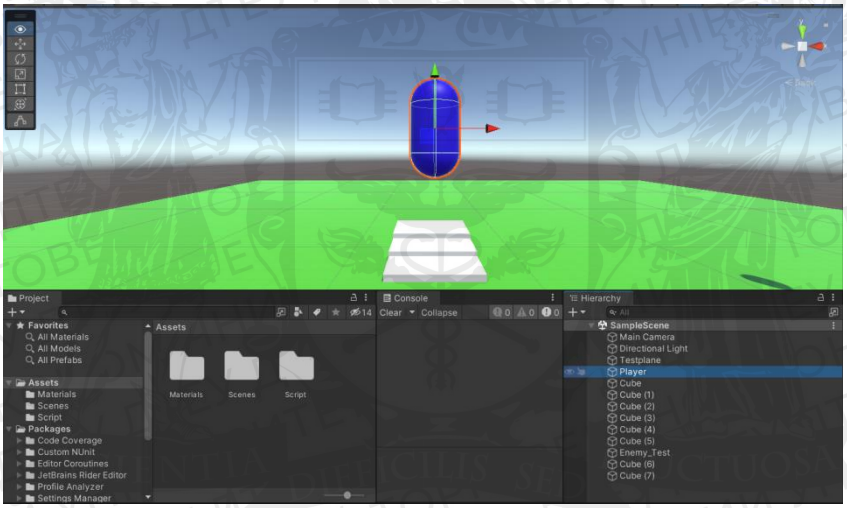
ви можете переключитися зі звичайного режиму у режим дебаг.У цьому режимі набагато більше опцій та можливостей ніж у звичайного інспектора.



Інспектор у режимі дебаг.

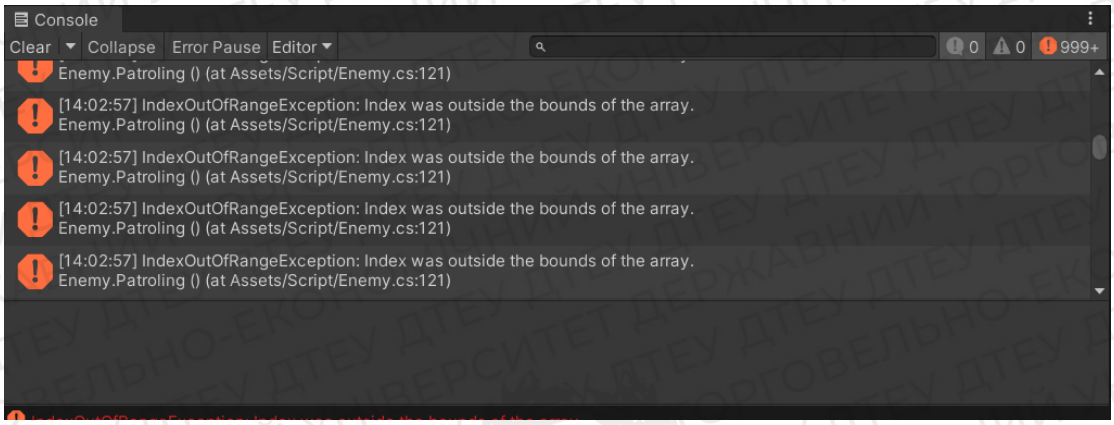
Все, що створюється в рамках проекту (файли, скрипти, текстури, моделі і т.д.), можна знайти в області Project. Це панель, на якій потрібно відображаються усі активи і структури проекту. Unity відображає в області Project усі папки та файли нашого проекту що знаходяться на жорсткому диску. Для того щоб подивитися детальніше увесь вміст проекту що зберігається на вашому комп'ютері натисніть ліву кнопку миші(лкм) та оберіть "show in explorer".[19]

Не забуваємо також про ієрархію.Усі створені вам об‘єкти на сцені з’являються в ієрархії.Саме об’єкти,а не вміст проекту.Під час створення стандартного проекту у Unity,на сцені автоматично створюються лише два об’єкта - основна камера та світло.Якщо ви бажаєте створити пусту сцену,без усіляких об’єктів,така можливість у вас також є.Також ви можете копію моделі перенести у будь-яку папку проекта,якщо у вас на це є потреба.Звідти,ви також можете перетягувати об’єкти до сцени.[20]



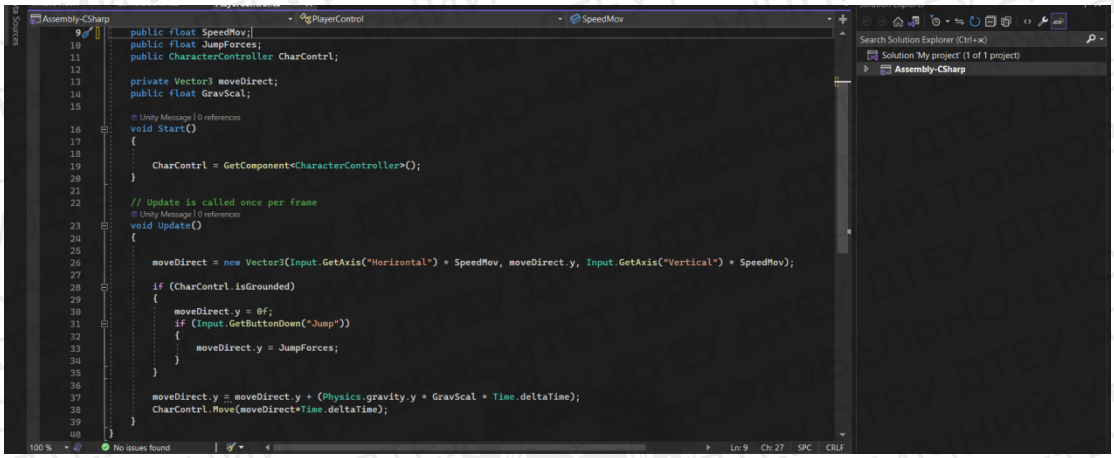
Ієрархія(hierarchy),консоль(console) та проект(project)

Будь-які помилки під час взаємодії з проектом будуть з’являтися у консолі.Заодно воно вкаже у якому саме рядку є помилка у скрипті,що дуже зручно та скорочує пошук помилок.



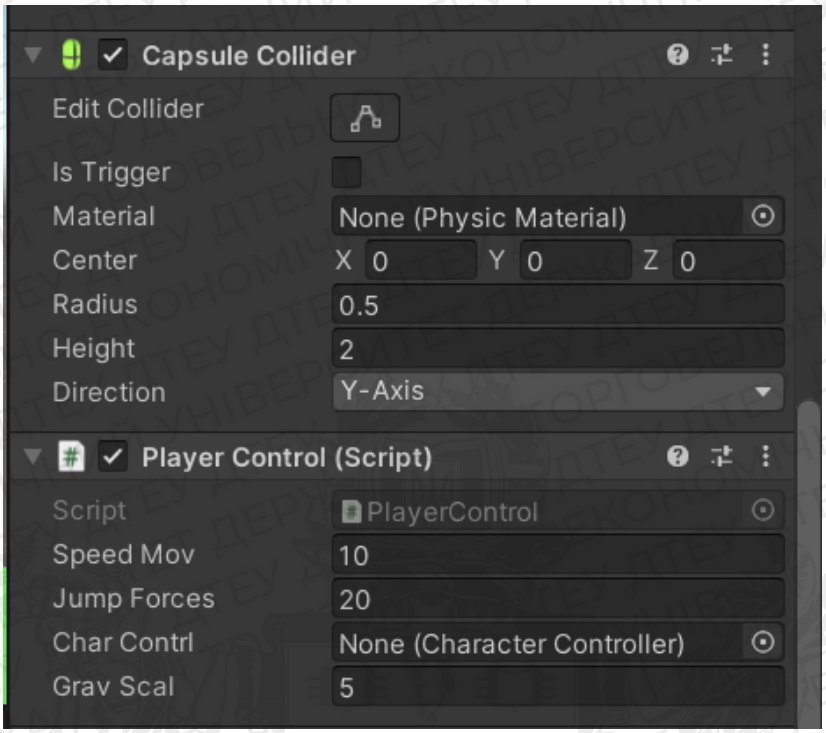
Консоль відображає що є помилка у скрипті ворога,через що скрипт не працює

Unity дає вам можливість писати скрипти через Visual Studio.Але для початку вам потрібно на налаштуваннях додати саму програму.Це потрібно для відкриття доступу до усіх можливих команд,що є у Unity.Звісно,не забувайте також зберігати постійно код у Visual Studio,щоб Unity мав можливість завантажити написаний вами код.Використовуючи також інспектор,ми маємо можливість розписати усі потрібні нам елементи гри(рух,силу стрибка,взаємодію з іншими об’єктами та гравітацію,щоб камера слідкувала за персонажем,тощо),і в інспекторі коригувати самі значення.



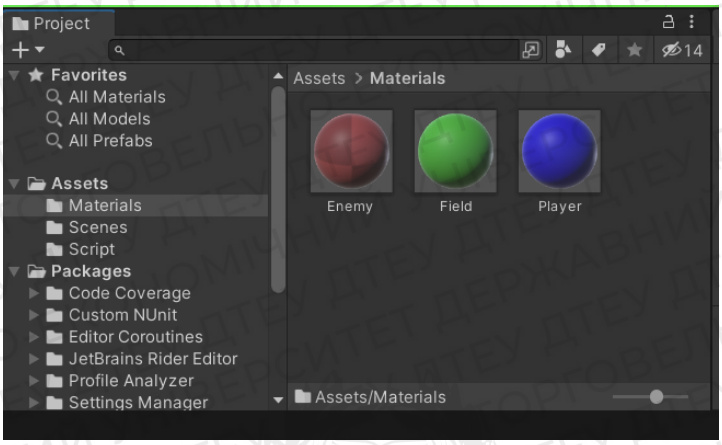
Скрипт що відповідає за рух гравця

Усі елементи у коді,що мають специфікатор public, будуть показані у інспекторі та дає змогу вас коригувати характеристики за вашим бажанням.Ви також можете налаштовувати біг,стрибок та решту навіть під час тестового запуску сцени.Елементи що матимуть специфікатор private не доступні нікому,окрім їх класу та дочірним елементам.[21]



Усі елементи які є публічними у скрипті,з'являтимуться у інспекторі

Усі ваші об'єкти по стандарту будуть білими.Для текстури гравця,поля та інших об'єктів,ми створимо матеріали,та надаємо їм кольори.Створюємо "матеріали",позначаємо потрібними кольорами та притягуємо до потрібного об'єкту і готово.Ваші об'єкти мають хоча б якусь текстуру.



Матеріали

Тепер перейдемо до гравця - капсула створена,додані усі до нього компоненти,а саме “character controller”,який спрощує нам завдання щодо керування гравцем.Також наш скрипт,у якому ми розписали можливість рухатися та стрибати.Публічні флоути у скрипті відповідають за певні характеристики гравця,як швидкість руху та сила стрибку.Як згадувалося раніше - ми маємо змогу редагувати ці показники навіть під час тестового запуску сцени.

```
public class PlayerControl : MonoBehaviour
{
    // Start is called before the first frame update

    public float SpeedMov;
    public float JumpForces;
    public CharacterController CharContrl;
```

```
{

    private Vector3 moveDirect;
    public float GravScal;

    void Start()
    {

        CharContrl = GetComponent<CharacterController>();
    }

    // Update is called once per frame
    void Update()
    {

        moveDirect = new Vector3(Input.GetAxis("Horizontal") *
SpeedMov, moveDirect.y, Input.GetAxis("Vertical") * SpeedMov);

        if (CharContrl.isGrounded)
        {
            moveDirect.y = 0f;
```

```
        if (Input.GetButtonDown("Jump"))
        {
            moveDirect.y = JumpForces;
        }
    }

    moveDirect.y = moveDirect.y + (Physics.gravity.y * GravScal *
Time.deltaTime);
    CharContrl.Move(moveDirect*Time.deltaTime);
  }
}
```

Код для керування гравцем

Time.deltaTime ми вписуємо щоб гравець не летів по світу на високій швидкості,через те,що його швидкість пов’язана з частотою кадрів у секунду.Визначивши показники запустимо сцену та превіримо чи все працює як потрібно та чи маємо ми змогу стрибати та стояти на платформах.

Основна робота зроблена,гравець має змогу рухатися та стрибати по платформам.Тепер ми можемо перейти до головного меню та додаткових функцій.

Головне меню гри є одним із найпростіших задач у даній роботі.Воно не потребує здебільшого великої кількості часу.Все,що нам потрібно на даний момент це можливість завантажити рівень та вийти з гри.Можна також додати налаштування,та змогу,наприклад змінювати гучність.

У першу чергу ми створюємо нову сцену.Та створюємо у ній зображення(або ж полотно).По дефолту,ми матимемо невеличкий білий квадрат у величезній прозорій рамці.У інспекторі ви можете розширити його під рамку.

Якщо ви бажаєте імпортувати та прив’язати до вашого полотна власне зображення,спершу ви маєте виставити у налаштуванні вашого зображення тип текстури Sprite(2D and UI).Після ви можете спокійно перетягувати ваше зображення до розділу Source Image.

Переконайтеся що під час створення кнопок меню,вони пов’язані з полотном.У інспекторі ви налаштовуєте кнопки на ваш власний розсуд.Ви можете обрати кольори які будуть під час натискання,чи коли курсор просто наведений на кнопку.Також можете розробити та завантажити власні спрайти для кнопок меню.

Для функцій “грати” або “вийти”,створюємо скрипт.У функції PlayGame ми завантажимо нашу сцену з рівнем,та QuitGame - щоб вийти з гри.

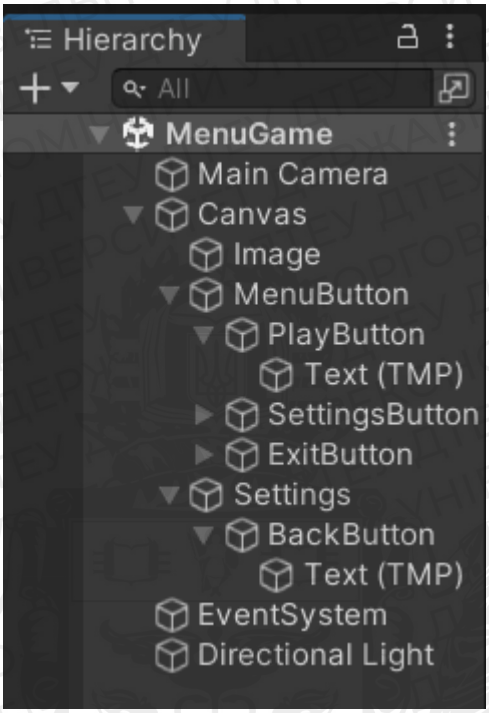
```
public class MainMenu : MonoBehaviour
{
    public void PlayGame()
    {
        SceneManager.LoadScene(1);
    }

    public void QuitGame() { Application.Quit(); }
}
```

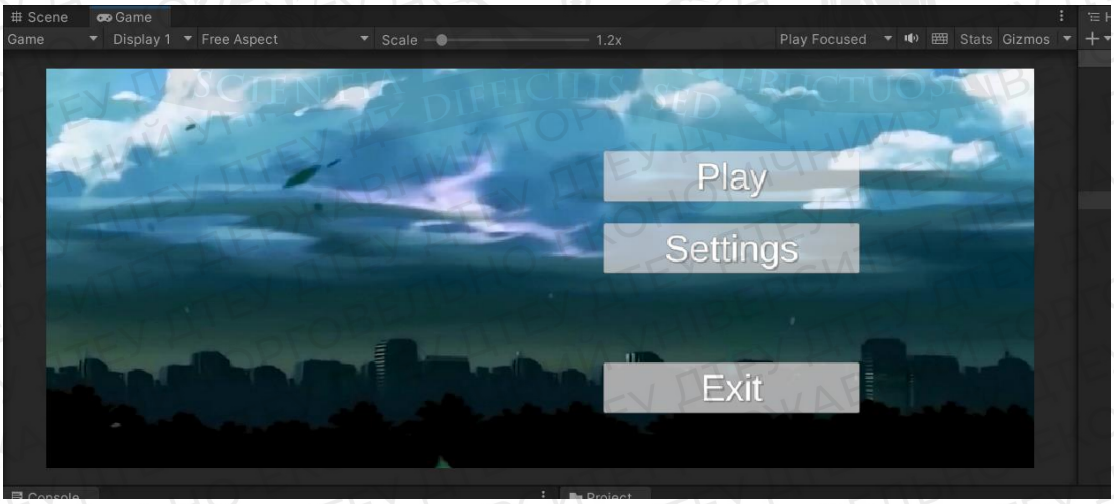
Код головного меню

Щоб завантажити потрібну нам сцену ми також можемо вписати назву потрібної нам сцени.

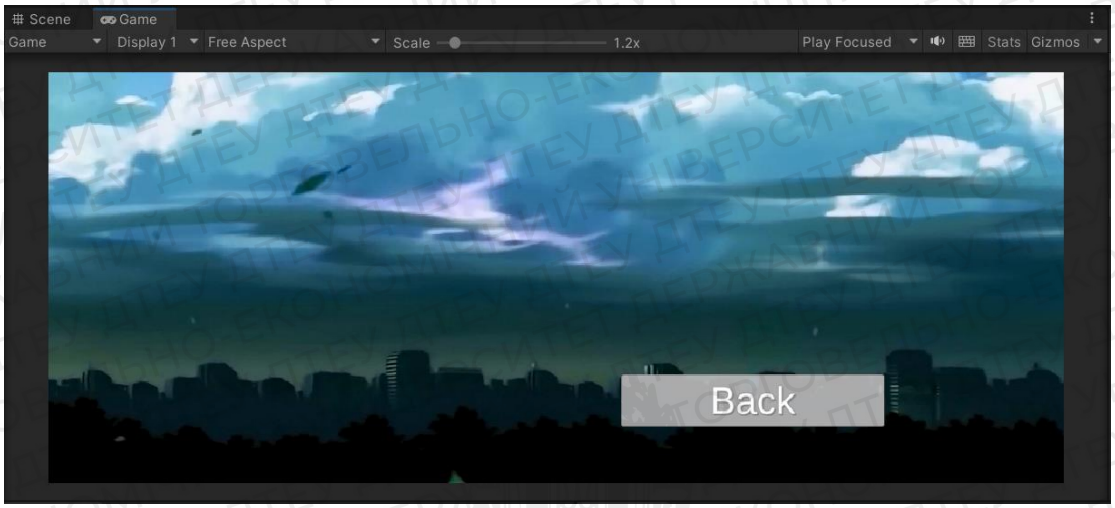
Для кнопок у меню ми робимо окремі об’єкти до яких приєднуємо кнопки та наш скрипт.Також,додамо налаштування де додамо кнопку “назад”.Тепер у нас є головне меню,у якому натискаючи на play - завантажуємо рівень та розпочнемо гру,на

settings - зайдемо на налаштування,та exit - вихід.Кнопка назад у налаштуваннях також з’являється та працює вона без проблем.



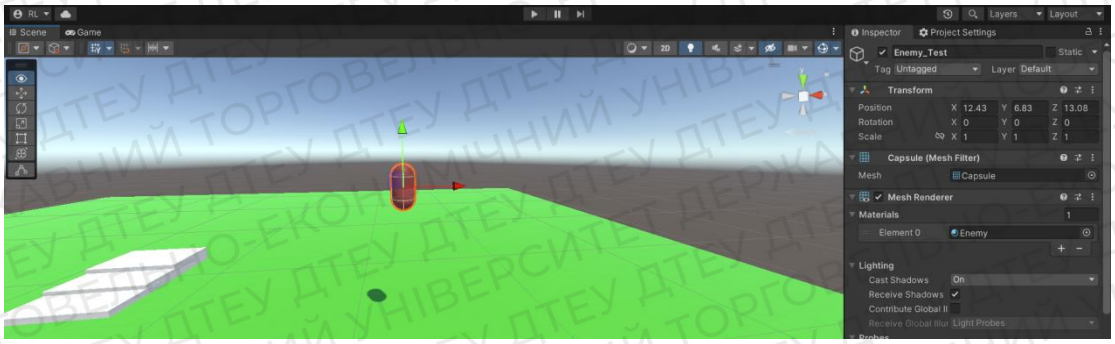


Головне меню



Налаштування

Головне меню також готове.Рівні завантажуються та програма закривається без усіляких проблем.Перейдемо тепер до ворогів.Робимо ще одну капсулу,яка буде нашим ворогом.Міняємо параметри у інспекторі за власним бажанням.Додаємо йому компоненти та переходимо до скрипту.Скрипт ворога буде значно більшим ніж у гравця.Нам потрібно розписати йому штучний інтелект - можливість доганяти та атакувати гравця.



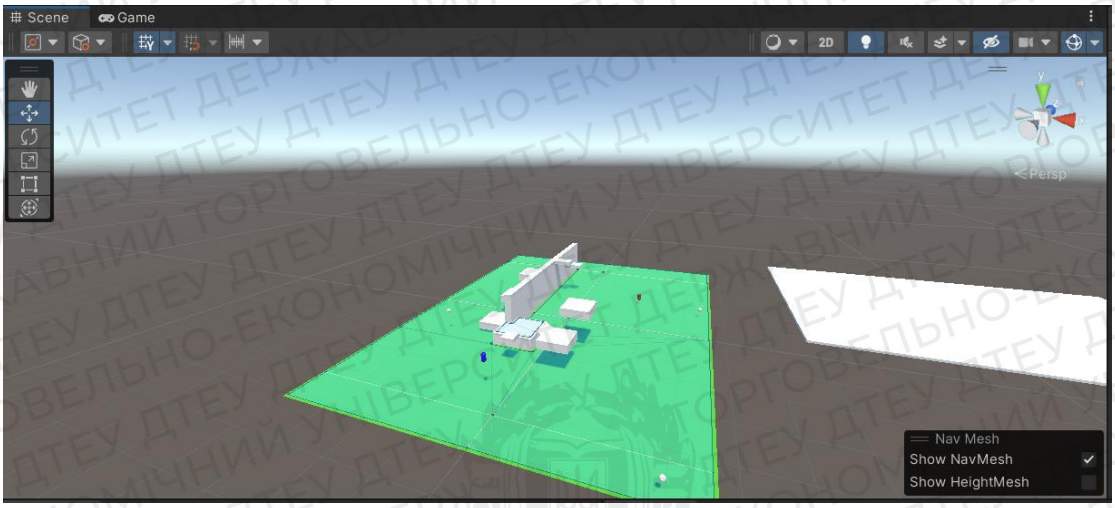
Наш ворог

Вороги є перешкодою для гравця що заважають йому дійти фінішу або ж виконати завдання для проходження рівня.Є багато

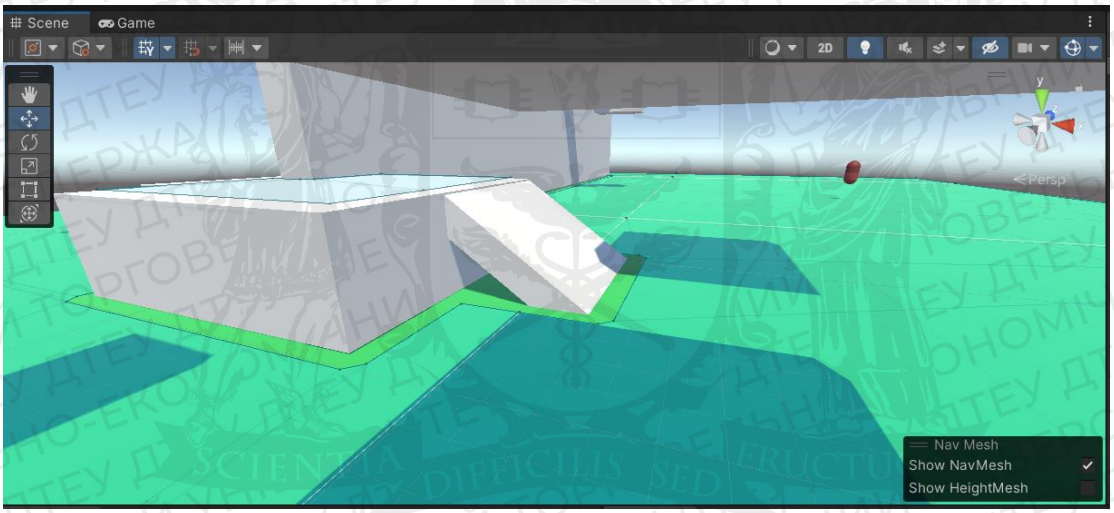
варіантів як краще розписати штучний інтелект ворога,але ми зупинимося на можливосts переслідувати його.Це більш зручний варіант для даного етапу.Також встановимо ворогу компонент RigidBody та NavMeshAgent.

Компонент RigidBody використовується для надання об’єктам фізики.Тобто завдяки ньому ці об’єкти можуть мати силу гравітації та можливість пересувати цей об’єкт,в залежності від його ваги.Він не буде намертво прикріплений до повітря,а падатиме одразу на землю як тільки гра почнеться.Цей компонент також чудово підходить і для декоративного плану.

NavMeshAgent використовується для розпізнавання об’єктом землі та пересування по ній.Для цього нам потрібно перейти у Window > Ai > Navigation,що увімкнути додатковий розділ налаштування,що знаходитиметься там же де й інспектор.Обираємо нашу величезну платформу та в вкладці “запекти”,натискаємо на “запекти”.З’явиться напів-прозоре “поле” що покриє платформу - таким чином помічається де об’єкти,що використовують navMesh,можуть пересуватися.Також ви можете у вкладці navigation вказати усі перешкоди,що є на вашому рівні - навколо них зникне синє покриття а об’єкти,що мають компонент агента,оминатимуть ці місця.



Поле,по якому може ходити об'єкт що має компонент NavMesh Agent.

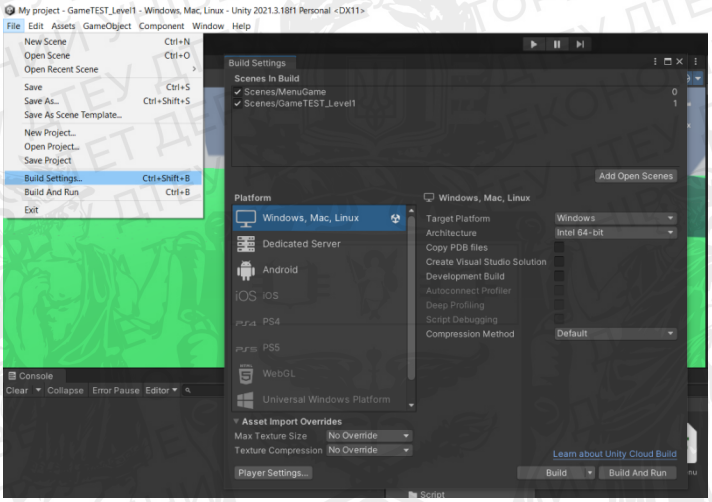


Усі об'єкти що позначені як "перешкода",будуть оминатися об'єктом

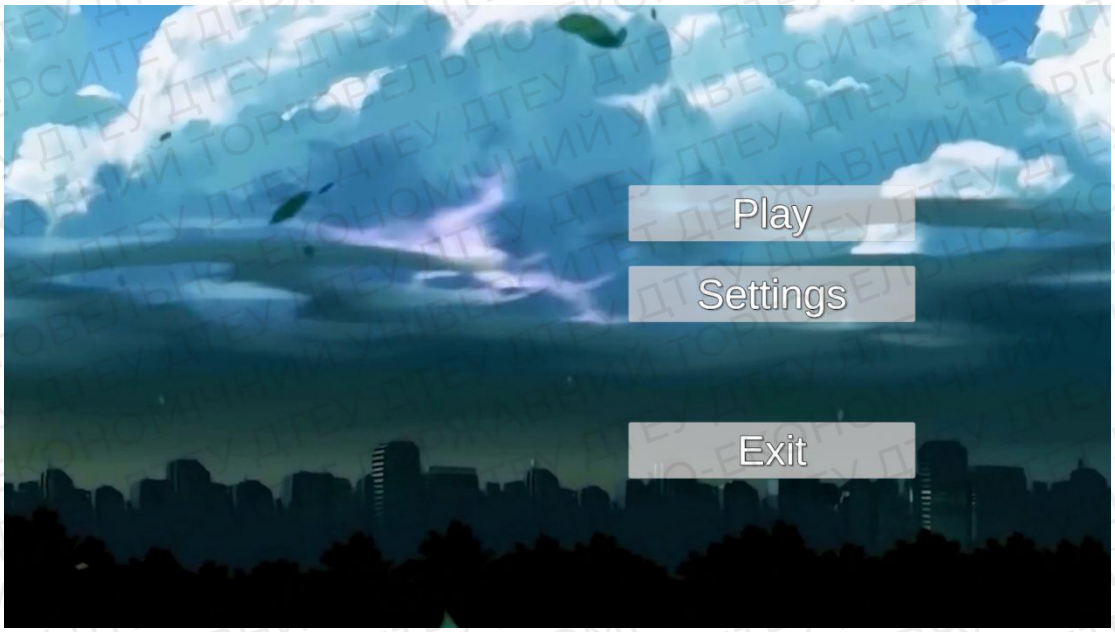
Якщо гравець стане на платформу,що позначена як перешкода,ворог стоятиме до поки гравець знову не буде на платформі по якій ворог може пересуватися.

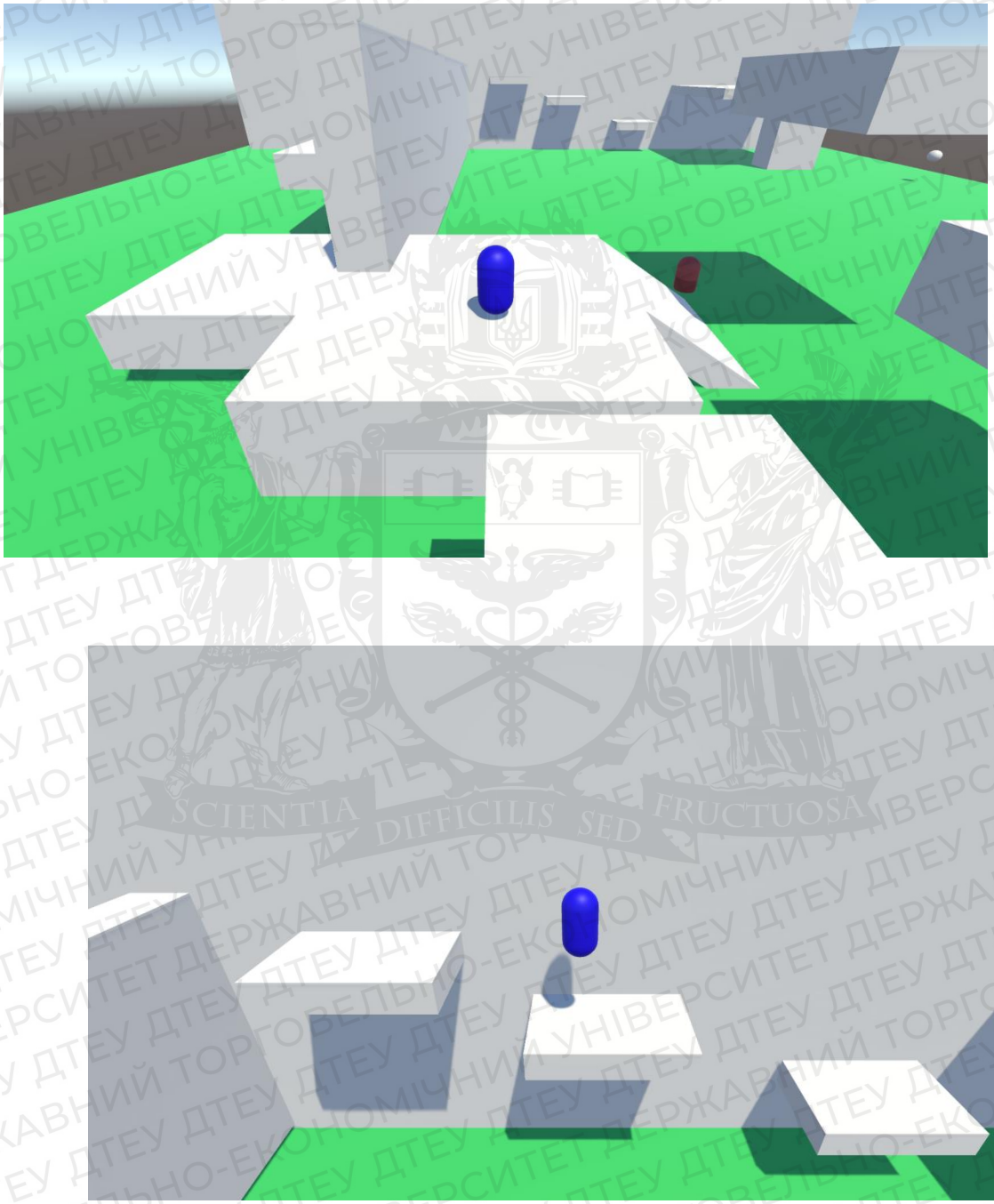
3.3. ТЕХНОЛОГІЯ ВИКОРИСТАННЯ ГРИ

Після виконання основних завдань,потрібно побудувати даний проект у повноцінну програму.Для цього ми заходимо у File > Build Settings.Обираємо потрібну нам платформу,розряд оперативної системи(32 або 64).Потім ви обираєте папку де зберігаєте пз та будуєте ігрове програмне забезпечення.



Програма запускається без проблем.На сучасних комп’ютерах з кращими характеристиками,програма завантажується за лічені секунди.Гра працює без фатальних помилок.Програма запускається у повноекранному режимі.





Скріншоти гри “Гра - платформа”: головне меню та тестовий рівень

Звісно,це не можна назвати повноцінною грою.Наразі можна сказати,що це є початковий прототип.Є можливість керувати гравцем та взаємодіяти з об’єктами,але залишається ще купа роботи

для допрацювання.Але цілі,що були зазначені у роботі були виконані.Навіть попри те,що у цьому прототипі було мало що зроблено - він має як потенціал,так і обширну кількість направлень та ідей.

Обмеження за віком є законодавчою політикою у низці держав стосовно творів масової культури, в основному: фільмів, телепередач та відеоігор. Система рейтингів «Обмеження за віком» безпосередньо пов'язана з такою політикою: вона використовується для інформування батьків про доцільність перегляду певного твору дітьми.[22]

Система **PEGI** – Велика Британія та більшість країн Європи

PEGI 3	Для всіх вікових груп. Гра не містить звуки та зображення, що можуть злякати маленьких дітей. Не допускається вживання нецензурних слів. Приклади: Just Dance, FIFA.
PEGI 7	Для дітей віком від 7 років. Допускаються помірні форми насильства, деякі сцени можуть налякати дітей. Приклади: Minecraft Dungeons, Рослини проти зомбі.
PEGI 12	Для підлітків віком від 12 років. Гра може містити більш реалістичні сцени насильства. Приклади: Fortnite, Sims 4.
PEGI 16	Для підлітків віком від 16 років. Сцени насилля стають більш реалістичними та схожими на реальне життя. Можуть містити нецензурну лайку та вживання алкоголю чи наркотиків. Приклади: Ark: Survival Evolved, Destiny 2.
PEGI 18	Для людей віком від 18 років. Ця класифікація застосовується для ігор з надзвичайним рівнем насильства та необґрунтованим вбивством. Гра також може містити сцени із наркотиками, азартними іграми та діями сексуального характеру. Приклади: Grand Theft Auto V, Fallout 4.

Рейтинги вікових обмежень[23]

Зазвичай,рейтинг вікових обмежень визначається лише коли гра вже на стадії “полірування” та готова до релізу,чого не можна поки казати про даний проект.Але,згідно з вищезазначеним списком,даний проект мав би рейтинг “PEGI 7”.

Висновки

У випускній кваліфікаційній роботі представлено результати теоретичних і прикладних досліджень, що полягають тематиці даної роботи.В результаті проведених досліджень були отримані такі висновки:

Аналіз наукових статей та енциклопедій дає розуміння термінологій,що таке рушій, та аналіз проблем,що згідно науковим статтям є частими серед розробників.У даній роботі проаналізовано також існуючі ігри у жанрі платформер,що розроблялися на рушії Unity.

Для даного проекту було аналізовано та створено концепцію, та моделі певних елементів,що мають ключову роль у даному проекті.Завдяки ним,формується наша гра.Детально аналізували поняття жанру даного проекту.Визначили які саме піджанри матиме даний проект.

Розроблено початковий прототип,що має основні елементи будь-якої гри у жанрі платформер.Також було проведено аналіз рушія Unity також було доведено що Unity надає можливості сучасному розробнику в короткий термін розробляти ігрові продукти.

Список використаних джерел

[1]–Спасюк Д. смартфони продовжують домінувати, а консолі втрачають прибуток,2022

[2]-Cristiano Politowskia,Fabio Petrillob,Gabriel C. Ullmanncand та Yann-Gaël Guéhéneuc. Game Industry Problems: an Extensive Analysis on the Gray Literature ст.7

[3]-Cristiano Politowskia,Fabio Petrillob,Gabriel C. Ullmanncand та Yann-Gaël Guéhéneuc. Game Industry Problems: an Extensive Analysis on the Gray Literature ст.12-13

[4]-Unity (ігровий рушій) [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Unity_(game_engine)

[5]- Системні вимоги для Unity 2021 LTS [Електронний ресурс]. – Режим доступу: https://docs.unity3d.com/Manual/system-requirements.html

[6]-Алгоритм [Електронний ресурс].- Режим доступу: https://uk.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC

[7] - Fig 1 - available via license: Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International [Електронний ресурс].- Режим доступу: https://www.researchgate.net/figure/Game-Flowchart-2D-and-3D_fig1_336201432

[8]-Biohazard 4 Prototype [Електронний ресурс].- Режим доступу: https://residentevil.fandom.com/wiki/Biohazard_4_Prototype#Development_rundown

[9]-Super Mario 3D World [Електронний ресурс].- Режим доступу: https://en.wikipedia.org/wiki/Super_Mario_3D_World

[10]-Crash Bandicoot (video game) [Електронний ресурс].- Режим доступу: https://en.wikipedia.org/wiki/Crash_Bandicoot_(video_game)

[11]-List of Lego video games [Електронний ресурс].- Режим доступу: https://en.wikipedia.org/wiki/List_of_Lego_video_games

[12]-Платформер [Електронний ресурс].- Режим доступу https://uk.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D1%80

[13]-Тривимірні платформери [Електронний ресурс].- Режим доступу https://uk.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D1%80#%D0%A2%D1%80%D0%B8%D0%B2%D0%B8%D0%BC%D1%96%D1%80%D0%BD%D1%96_%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D1%80%D0%B8

[14]-Становлення жанру платформерів [Електронний ресурс].- Режим доступу https://uk.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D1%80#%D0%A1%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B6%D0%B0%D0%BD%D1%80%D1%83

[15]-Нескінченні ранери [Електронний ресурс].- Режим доступу https://uk.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D1%80#%D0%9D%D0%B5%D1%81%D0%BA%D1%96%D0%BD%D1%87%D0%B5%D0%BD%D0%BD%D1%96_%D1%80%D0%B0%D0%BD%D0%B5%D1%80%D0%B8

[16]-Двох з половиною вимірний опис об'єкта [Електронний ресурс].- Режим доступу https://en.wikipedia.org/wiki/2.5D

[17]–Метроїдванія [Електронний ресурс].- Режим доступу https://en.wikipedia.org/wiki/Metroidvania

[18]-Mike Geig Unity 2018 Game Development in 24 hours, 2018 ст.28

[19]-Mike Geig Unity 2018 Game Development in 24 hours, 2018 ст.32

[20]-Mike Geig Unity 2018 Game Development in 24 hours, 2018 ст.35

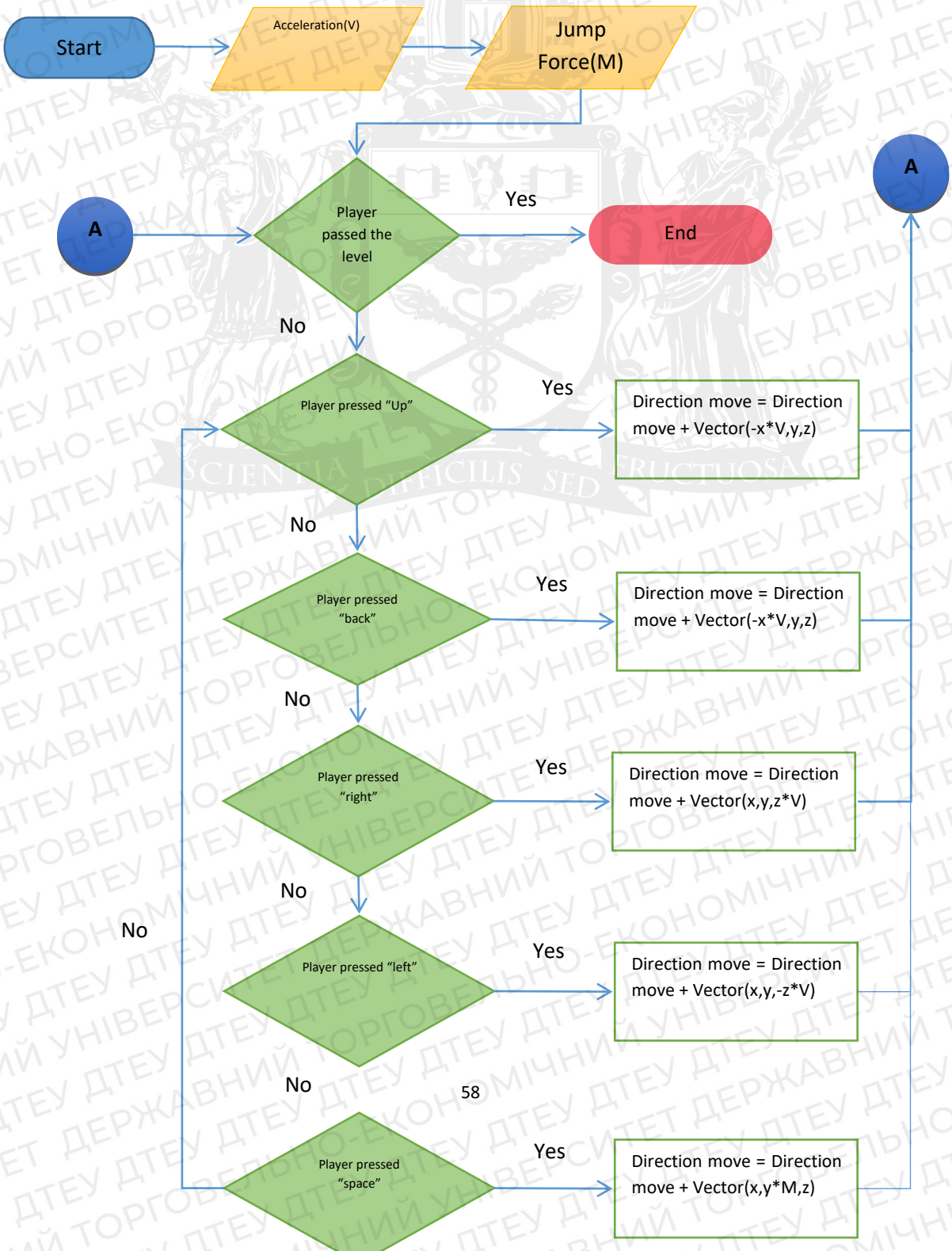
[21]-Mike Geig Unity 2018 Game Development in 24 hours, 2018 ст.156

[22]-Обмеження за віком [Електронний ресурс].- Режим доступу https://uk.wikipedia.org/wiki/%D0%9E%D0%B1%D0%BC%D0%B5%D0%B6%D0%B5%D0%BD%D0%BD%D1%8F_%D0%B7%D0%B0_%D0%B2%D1%96%D0%BA%D0%BE%D0%BC

Додатки

Додатки для 2.1 "Основи концепції та алгоритми обчислень":

2.1.1 Блок-схема керування гравцем



2.1.2 Блок схема роботи штучного інтелекту ворога