

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: «Розробка програми для створення та редагування ігрових моделей  
з використанням Autodesk 3D Max, Blueprin»

Виконав(ла): студент(ка) IV курсу, групи СПс-42  
спеціальності \_\_\_\_\_

121 «Інженерія програмного забезпечення»

(шифр і назва спеціальності)

Сабіров Є.О.  
(підпис) (прізвище та ініціали)

Керівник \_\_\_\_\_  
(підпис) (прізвище та ініціали)

Нормоконтроль \_\_\_\_\_  
(підпис) (прізвище та ініціали)

Завідувач кафедри \_\_\_\_\_  
(підпис) (прізвище та ініціали)

Рецензент \_\_\_\_\_  
(підпис) (прізвище та ініціали)

## РЕФЕРАТ / ABSTRACT

Кваліфікаційна робота бакалавра містить: с., рис., табл., джер.  
ДОСЛІДЖЕННЯ, АКТУАЛЬНІСТЬ, ЗАВДАННЯ, ОБ'ЄКТ, АНАЛІЗ,  
ПРОГРАМНА СИСТЕМА, ВИМОГИ, UNREAL ENGINE, C++,  
КОНСТРУЮВАННЯ, ТЕСТУВАННЯ

Метою представленої бакалаврської роботи є розробити програмний продукт призначенням якого є створення та редагування ігрових моделей, анімацій та логіки з використанням Autodesk 3D Max, Blueprint.

Метою роботи є це створення ігрової моделі призначенням якої є розвиток логіки користувача і можливостей удосконалювати штучний інтелект у комп'ютерних іграх. Це означає, що розроблена гра повинна працювати і бути побудована з використанням різних технологій і платформ.

RESEARCH, TOPIC ACTUALITY, TASK, OBJECT, ANALYSIS,  
SOFTWARE SYSTEM, REQUIREMENTS, UNREAL ENGINE, C++, DESIGN,  
TESTING

The purpose of the presented bachelor thesis is to develop a software product whose purpose is to create and edit game models, animations and logic using Autodesk 3D Max, Blueprint.

The purpose of the bachelor thesis is to create a game model, the purpose of which is to develop user logic and opportunities to improve artificial intelligence –AI for computer games. This means that the developed game must work and be built using different technologies and platforms.

## ЗМІСТ

ВСТУП	7
1 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ	9
1.1 Аналіз предметної області та вимоги до програмної системи	9
1.1.2 Аналіз існуючих рішень	13
1.1.3 Постановка задачі та опис можливих рішень	14
1.1.4 Пошук актантів та варіантів використання	15
1.1.5 Опис ключових варіантів використання	17
1.1.6 Опис специфіки, аналіз ролей та посад що використовуються в розробці відеогри	18
1.2 Етапи розробки відеогри	20
1.2.1 Проєктування (пре-продакшин)	20
1.2.2 Особливості виробництво програмного продукту типу відеогра	21
1.2.3 Процес розробки контенту	22
1.2.4 Процес розробки механіки гри	24
1.2.5 Етап тестування	24
1.2.6 Етап випуску та реалізації	25
1.2.7 Процес підтримки	26
1.3 Конструювання програмної системи	27
1.3.1 Обґрунтування редактору Unreal Engine	27
1.3.2 Обґрунтування застосування Blueprint Interface	28
1.3.3 Робота з освітленням	30
1.3.4 Використання інструментів і моделі SpeedTree в додатку Unreal Engine 4	31
1.3.5 Імпорт моделей в UE 4	32
1.4 Апаратні засоби та програмне забезпечення	36

2 ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ	37
2.1 Тестування можливостей графічного рушія типу Unreal Engine	37
2.2 Тестування процесорів за максимальних налаштувань по якості	42
3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	46
3.1 Безпека життєдіяльності	46
1 3.1.1 Запобігання виникненню надзвичайних ситуацій	46
2 3.1.2 Аварії з викидом шкідливих речовин	49
3.2 Основи охорони праці	52
ВИСНОВКИ	56
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ	58
ДОДАТКИ	62
Додаток А Слайди презентації	
Додаток Б Відгук	
Додаток В Рецензія	

## ВСТУП

Стандартом першого (бакалавра) рівню вищої освіти (ОПП за назвою „121 Інженерія програмного забезпечення„) та відповідно до нього розробленою однойменною освітньо-професійною програмою передбачено цілий ряд компетентностей та відповідних їм програмних результатів які я, як здобувач вищої освіти, набуваю, та можу їх та відповідно передбачені використати при написанні роботи кваліфікаційного рівня бакалавра. Крім цього в репозитарії ТНТУ оприлюднене положення що регламентує та роз'яснює кваліфікаційні роботи студентів ТНТУ ім.І.Пулюя на здобуття першого першого (бакалавра) рівню вищої освіти якого я дотримуюсь і в якому вказано, що в бакалаврській КР слід зробити акцент на вирішенні спеціалізованої задачі яка супроводжується практичними проблеми у певній галузі в сфері професійної діяльності, також передбачається застосування обраних теорій та методів та підходів відповідних суміжних попередньо вивчених дисциплін, важливими є такі характеристики як комплексність та невизначеність умов, відповідно до вимог стандарту вищої освіти.

В процесі та під час виконання роботи бакалаврської роботи я, відповідно до кваліфікаційних вимог, планую:

- проаналізувати та осмислити проблемну область, здійснити пошук та проаналізувати інформаційні ресурси та джерела, на предмет обраної тематики шляхом аналізу існуючих чи подібних методик вирішення поставлених задач;

- провести бібліографічну роботу використавши для цього етапу принципи та підходи сучасних інформаційних технологій та відповідні інструменти, користуватись методичними, можливо при потребі статистичними та відповідними законодавчими нормативними матеріалами, які є у вільному доступі, критично оцінити сучасні існуючі відповідно до обраної тематики

концепції та підходи, врахувати економічні аспекти доцільності створення програмного продукту, парадигм безпеки життєдіяльності та охорони праці;

- сформулювати мету і актуальність обраної тематики дослідження;

- використати сучасні методи та інструментів для інженерії програмного забезпечення,

- проаналізувати отримані результати, які використовувались та отримані в процесі роботи;

- оцінити можливість застосування отриманих результатів у, для мого випадку, практичній діяльності;

- продемонструвати також вміння та знання що стосуються сучасних саме інформаційних технологій призначених для проведення дослідження в обраному напрямку;

- кваліфіковано оформити отримані результати у вигляді записки до кваліфікаційної роботи, з дотриманням принципів самостійності та вимог відповідних ДСТУ щодо оформлення технічної документації.

## 1 РОЗРОБКА ПРОГРАМНОЇ СИСТЕМИ

### 1.1 Аналіз предметної області та вимоги до програмної системи [1-7]

Unreal Engine 4 (UE4) і Cry Engine 3, є двома найбільш популярними і затребувані ігровими рушіями, які доступні для широкого загалу [8]. На даний момент багато студій використовують власні рушії гри, також є ринок для інди-розробників і навіть великі організації, яким потрібні потужні ігрові рушії, які б допомогли їм при створенні своєї гри. Зокрема, такі ігри, як Tom Clancy's Splinter Cell (всі частини), Postal 2, Infinity Blade, Crysis 1-3 розроблялись саме на цих ігрових системах.

Можна сказати що Unreal Engine і Cry Engine 3 чудовими рушіями гри, але в залежності від того, що ви плануєте розробити, природно, що один може бути

кращим варіантом ніж інший. Можливо це залежить від інтерфейсу, чи може від мови програмування. У будь-якому випадку обома можна користуватися, однак я би хотів більш детально розглянути кожен із ігрових рушіїв і проаналізувати їх сильні сторони, щоб мати змогу вирішити, яких із них буде краще виконувати задуману задачу [9-11].

Перш за все вважаю, що варта почати з того, що потрібно зрозуміти який конкретно потрібен ігровий рушій, як ви збираєтеся його використовувати, і які конкретно ігри ви хочете створювати, зокрема: 2D-платформер або 3D гру, можливо планується поєднання 2D і 3D, можливо гра для веб-браузера або для мобільної платформ, можливо потрібні консольні платформи. Також важливим моментом є те, яким чином планується заробіток від створених продуктів, чи передбачена купівля через додаток і ще багато багато інших важливим моментів. Наприклад, для створення 3D-гри, Cry Engine 3 буде більш прийнятним варіантом, оскільки він є дуже потужним. І хоча графічно він не на тому ж рівні, що Unreal Engine 4, але якщо не планується створення Next Gen графіки, то його можливостей буде абсолютно достатньо [12].

Крім вище переліченого виборі ігрового рушія в значній мірі може залежати від того, яку саме гру заплановано, на якій платформі, та фінансові та матеріальні можливості. Нас сьогодні Cry Engine 3 пропонується в безкоштовній версії, якої цілком достатньо для початківця для ознайомлення. Однак вже для тестування усіх можливостей обраного рушія, потрібно придбати мінімальну одномісячну підписку, яка вартує в межах 10 у.о. Для Unreal Engine 4 вже такого немає, оскільки він абсолютно безкоштовний, зручний, з повним відкритим доступом. Однак, вже UE4 передбачає ліцензійний платіж у розмірі 5%, тому, слід пам'ятати, що з будь-якої суми заробленої створеним вами проектом, Epic Games отримає 5% включаючи купівлі через додаток, розміщених оголошень, та вартості гри. Важливим також вважаю те, що Unreal Engine 4 безкоштовний для коледжів та університетів. Отже, обидва рушії гри є відносно доступними, з безкоштовними альтернативами.

Далі вважаю, настав час вибору мови програмування. Гра як і будь-який інший програмний продукт потребує знань мов програмування, від вибору якої залежить можливість вирішення який рушій оптимальніше обрати UE4 або CryEngine 3.

В Unreal Engine 4 використовується C++, а в Unity в основному C# або JavaScript. Таким чином вирішивши, на мою думку все зводиться до особистих вподобань. Часто я чув думку, що C++ є більш архаїчною. Також Unreal Engine 4 має Blueprint візуальний сценарій. Блок на основі методу сценаріїв знаходиться безпосередньо в середині UE4, таким чином технічно не потрібно писати жодного рядка коду. Це дуже зручно для швидкого створення прототипів рівнів, і за допомогою Blueprint є можливість створювати цілі ігри [13,14].

Природно що існують обмеження на можливості Blueprint, їх цілком достатньо для отримання навиків у випадку створення ігор. Також якщо планується використання Unreal Engine 4 для візуалізації та архітектурних проходжень Blueprint є хорошим рішенням оскільки не доведеться писати жодного рядка коду, що є зручно для швидкого створення ігри.

Також важливо, що Unreal Engine 4 виходить на перше місце за розміром активів свого магазину. Є можливість завантаження різних ігрових активів, зокрема символів, реквізитів та звукових ефектів.

В ігрових консолях наступного покоління передбачено можливості для створення графіки на одному рівні з іграми. Зокрема в CryEngine 5 графічні можливості суттєво розширено.

Не лишнім буде зазначити, що все таки графіка, не є головною ціллю розробки, оскільки сьогодні для зацікавлених користувачів та розробників вона не є ключовим аспектом в грі, і часто немає сенсу акцентуватись при створенні продукту конкурентноспроможного до графіки надвисокого рівня, оскільки не це є метою. В Unreal Engine 4 передбачено можливість створити будь-який тип візуалізацію, стилю зокрема це і 2D або і 3D. І на мою думку завжди краще мати



можливість вдосконалити свою роботу, зокрема в графіці, що і можна буде вже вважати модернізацією.

Якщо є говорити про простоту, тобто яка програма є простішою у використанні, то тут перевагу варта надавати лише особистим вподобанням на мою думку. CryEngine 3, як правило, розглядається як більш інтуїтивно і простіше зрозумілий ігровий рушій, однак, Unreal Engine 4 після вдосконалення і доопрацювання UI вже також достатньо легко зрозуміти користувацький інтерфейс, і це не займе багато часу [15].

Вважаю, що для того, щоби зрозуміти який же все таки варіант обрати, варта спробувати обидва. Отже, пропоную стислий огляд, який я наведу в результаті аналізу:

CryEngine 3 є безкоштовним, повнофункціональним рушієм в базовій своїй версії. Для отримання розширеної версії потрібно платити 10\$ кожного місяця, і тоді можливості та функціонал будуть більшими, зокрема:

1. Підтримка C#: нове API дозволить розробникові з досвідом C# розпочати роботу з скриптами в CryEngine 3 одразу [16].
2. Покращена продуктивність на сучасних ПК в застосунках зі складною графікою.
3. Підтримка DirectX 12 з можливістю глибокої оптимізації.
4. Покращена тривимірна система оптимізована під VR, дає можливість досягти кращої якості з мінімальними впливом на продуктивність.
5. Нова система: майже все обробляє GPU.
6. Підтримка FMOD Studio для гнучкості в області аудіо.

Unreal Engine 4 попередньо коштував \$ 19 на місяць, проте станом на сьогодні Unreal є безкоштовним, таким чином можна отримати вихідний код C++ абсолютно безкоштовно. Однак, після того, як гра почне приносити прибуток, з автора на користь Epic Games стягуватиметься 5% від квартального валового прибутку на продукт [17].

Обидва двигуни мають дуже активні спільноти на дискусійних форумах і відповіді концентратори, а також об'ємну документацію, керівництва та вікі-сторінки.

Не можу не відмітити і те, що документація Unreal дуже добре написана, містить детальні та зрозумілі пояснення та скріншоти. У той самий час документи CryEngine 3 також непогані, проте вважаю, їх варта доопрацювати.

Важливо, що CryEngine 3 має величезну кількість відео-уроків по [www.digitaltutors.com](http://www.digitaltutors.com), також є підручники по даному рушію, однак англійською, але для мене це не є проблемою. Також є сотні професійних Tuts завантажень призначених як для початківців так і для просунутих користувачів, які охоплюють всі аспекти процесу створення гри.

Стосовно платформ, то для Unreal Engine це: Windows PC, Mac OS X, IOS, Android, VR, Linux, SteamOS, HTML5, Xbox One і PS4.

Платформи Cry Engine 3 це: Windows PC, Mac OS X, VR, Linux, SteamOS, Xbox One і PS4 [18].

Якщо говорити про інтерфейси, то обидва дуже схожі, з панелями інструментів і настройками.

Інтерфейс Unreal, на мій погляд, є достатньо роздутим і складним, ніж це могло би бути. Багато часу витрачається на імпорт і зберігання, прості завдання вимагають додаткових, непотрібних кроків.

Cry Engine 3 швидший, інтерфейс швидкий і зрозумілий. Він настільки легкий, що може працювати на Windows XP (SP2), в той час як UE4 вимагає мінімум, Windows 7 64-біт. Хоча кінцевий продукт може виглядати краще в Unreal, ніж в Cry Engine 3, проте йде більше часу і набагато більше зусиль, особливо для початківців.

В CryEngine 3 більшість розробників для розробки використовують або JS або C #. Однак, не обов'язково використовувати тільки якусь одну. Можна використовувати одну або іншу, або обидві. Unreal Engine поставляється з

системою Візуалізатора сценаріїв, які можуть бути використані, щоб розробити всю гру, або використовувати в поєднанні з трохи швидшим C ++.

Epic Games не має так багато сервісів, але було створено так званий фонд розвитку на п'ять мільйонів доларів з цільовим призначенням "фінансові гранти на інноваційні проекти, побудовані в Unreal Engine 4 попередньої та наступної версій".

### 1.1.2 Аналіз існуючих рішень

Steam вважається сервісом за допомогою якого здійснюється цифрова дистрибуція компанії Valve, який вважається одним із відомих розробників відеоігор. Steam зазвичай використовують з метою завантаження та активації ігор через мережу інтернет, щоб отримати автоматичні оновлення та новини, стосується продуктів самої Valve, так і інших розробників, серед яких варта назвати: Activision, Codemaster, Eidow Interactive, Epic Games, GSC, Ubisoft, THQ [19-23].

Однак користуватися сервісом можуть лише зареєстровані розробники, реєстраційний внесок у розмірі 100 у.о. Не зважаючи на потужні можливості ціна доволі висока.

Origin (попередньо до 3 червня 2011 року EA Store) є системою призначенням якої є цифрова дистрибуція компанії ElectronicArts, яка дає змогу користувачеві придбати комп'ютерні ігри через мережу Інтернет та вантажити їх при допомозі клієнту Origin (попередньо EA DownloadManager, EA Downloader та EA Link ) [24].

В Origin використовуються і деякі соціальні функції, зокрема:

- можливість керування профілем;
- можливість спілкуватися з однодумцями та друзями в чаті, та під час гри.

Додатковий функціонал робить Origin конкурентоспроможним по відношенню до Steam. Станом на сьогодні передбачено можливість автоматичного оновлення ігор, синхронізацію зберігання ігор за допомогою хмарного сховища, можливість комунікації у вбудованому чаті.

Для ідентифікації користувачі Origin передбачено об'єднаний акаунт, яким замінюється акаунт EA і може вилучатись, якщо його не використовували впродовж 24 місяців та на ньому не було зареєстровано жодної придбаної гри. Крім цього передбачено можливість змінювати ім'я облікового запису.

### 1.1.3 Постановка задачі та опис можливих рішень

Завданням представленої бакалаврської роботи є розробити програмний продукт призначенням якого є створення та редагування ігрових моделей, анімацій та логіки з використанням Autodesk 3D Max, Blueprint [1,25].

Задачу можна сформулювати наступним чином – це створення ігрової моделі призначенням якої є розвиток логіки користувача і можливостей удосконалювати штучний інтелект у комп'ютерних іграх. Це означає, що розроблена гра повинна працювати і бути побудована з використанням різних технологій і платформ [26,27].

Таким чином фіналізуючи вище сказане, основним завданням є:

1- проведення поєднання ігрових можливостей до більш загальних функціональних можливостей, за допомогою яких можна задовольнити більшість варіантів використання.

2- варта звернути увагу на можливості реалізації вузькоспеціалізованих функцій. Це можна зробити не накладаючи жорстких обмежень на варіанти використання.

3- забезпечити підтримку різних сучасних технологій та платформ, для цього необхідно обрати обидва протоколи для передачі даних:

загальнодоступний та поширений на багатьох платформах формат повідомлень. Цей підхід дасть змогу отримати результати для покращення роботи штучного інтелекту.

4- беручи до уваги цільову аудиторію, на яку розрахована гра, вона повинна мати відповідну складність, бути цікавою, з непередбачуваними варіантами проходжень, достатньо простою для впровадження до програмного забезпечення штучного інтелекту та з врахуванням можливості підтримки малих команд.

#### 1.1.4 Пошук актантів та варіантів використання

Після огляду існуючих рішень та ознайомлення із профільними ігровими ресурсами, я склав перелік найбільш затребуваних варіантів використання комп'ютерних ігр у програмному забезпеченні, чим і визначив варіанти використання для свого проєкту [28]. Зокрема це:

- авторизація користувача;
- скачування контенту даної гри;
- отримання початкових даних користувача;
- пізнання для себе чогось нового;
- отримання даних про управління грою для користувача;
- отримання різних досягнень.

Також варта провести розділення функціоналу для розробника і розроблюваного ним програмного забезпечення. Так функції, що відповідають за обробку даних кінцевого користувача, розробник може використовувати у своєму ПЗ, в той час як реєстрацію програмного забезпечення у системі як споживача сервісу доцільно проводити у окремому шарі, який назву AI.

Таким чином результуючий список актантів набув наступного вигляду:

##### 1. Користувач

## 2. Розробник

## 3. AI

Далі пропоную детальніше розглянути варіанти використання кожного з цих актантів.

Користувач має доступ до своїх даних. Для цього користувачу слід авторизуватись у системі, чому має передувати реєстрація. Таким чином, список варіантів використання для користувача пропоную наступний:

- реєстрація;
- вхід;
- вихід;
- налаштування гри;
- отримання досягнень;
- отримання досягнень за грою;
- взаємодія зі штучним інтелектом.

Для подальшого використання користувачем гри, слід залучити розробника, адже саме ПЗ третьої сторони взаємодію з даними користувачів. Для цього розробнику слід отримати ключі доступу, за якими його ПЗ буде ідентифіковано і допущено до роботи з користувачем. Перелік варіантів використання для розробника:

- вхід;
- вихід;
- створення гри;
- налаштування штучного інтелекту;
- налаштування гри;
- створення досягнень.

### 1.1.5 Опис ключових варіантів використання [29-32]

При отриманні таблиці досягнень користувач звертається до сервісу із запитом, а у відповідь отримує таблицю з назвами досягнень як своїх так і відсортованих по порядку користувачів.

При отриманні переліку ПЗ третьої сторони (далі візьму конкретний випадок – ігри) користувач звертається до гри із запитом про те, що він знайшов всі зазначені предмети, а у відповідь отримує таблицю, що містить коротку інформацію про знайдені предмети, та про ті які ще потрібно буде знайти. Даний випадок представлений у наступній таблиці 1.1.

Таблиця 1.1

№	Актор	Назва	Опис
1	Розробник	Створення гри	Для одного із раніше зареєстрованих користувачів
2	Розробник	Налаштування гри	Надає змогу користувачу налаштувати гру
3	Розробник	Отримати перелік предметів	Користувач отримує перелік здобутих предметів
4	Розробник	Отримати перелік досягнень	Користувач отримує перелік здобутих досягнень
6	Користувач	Вхід	Користувач ідентифікує себе

Продовження таблиці 1.1

№	Актор	Назва	Опис
7	Користувач	Вихід	Користувач закінчує роботу з грою
8	Користувач	Налаштування гри	Налаштування гри під себе
9	Користувач	Взаємодія із штучним інтелектом	Вступає в взаємодію із штучним інтелектом
10	AI	Звернення уваги на гравця	Перевіряє чи здобув користувач вибране досягнення
11	AI	Взаємодія з користувачем	Замічає користувача

1.1.6 Опис специфіки, аналіз ролей та посад що використовуються в розробці відеогри

Перш за все це гейм дизайнер який відповідає за концепцію гри та її основні, відмінні від інших, риси. Ним проектується базова ігрова механіка, безпосередньо сам процес, образи задуманих персонажів. Він займається і коригуванням впродовж розробки. Зазвичай добре, якщо гейм дизайнер вже має досвід роботи розробки відеоігор, на іншій посаді чи посадах. Часто в розробці приймають участь декілька гейм дизайнерів, один з яких виконує обов'язки провідного дизайнера, інші є несуть відповідальність за дизайн окремих напрямків: персонажі, рівні, механіка.

Менеджер, учасник команди розробників, в обов'язки якого входить контроль та координація роботи людей, які залучені до процесу створення гри. Він займається розподілом обов'язків, встановленням строків виконання кожного із видів робіт. Саме менеджери за специфікою розділяють функції генерального директора, головного менеджера та продюсера.

Для написання та вдосконалення програмного коду та інших видів робіт пов'язаних із ним є програмісти. Зокрема він займається створенням базової ігрової механіки, файлової архітектури проєкту. Можливий наступний розподіл



видів робіт: технічного директора, програміста-проектувальника, програміста інтерфейсу, графічного рушія та інше.

Також не обійтись без художника, який створюватиме заставки, спрайт, текстуру, тривимірні моделі, займається оформленням інтерфейса. Часто передбачено розподіл, в якому існує розподіл коли один із художників займається текстурами, інший відповідає за спецефекти, анімацію і так далі. Роботу всіх художників координує головний художник - арт-директор.

Ще одним із ключових працівників є звукорежисер, який займається записом та обробкою звукових ефектів, музичного супроводу та мови. Саме він продумує доцільність та відповідність звуків відповідно до ігрових подій та ситуацій.

Композитор займається написанням музики, яку можуть виконувати як окремі музиканта так і гурти на різних музичних інструментах та поєднаннях.

Звуковими ефектами та їх синтезуванням займається відповідальний за цей напрямок.

Сценарист займається складанням сценарію гри, загальним сюжетом і окремими подіями, написанням текстів діалогів, описом предметів та персонажами.

В створенні рухів персонажів, зйомкою відеороликів займається актор.

Актор озвучування займається озвучуванням мови персонажів, закадровим голосом.

Дизайнер рівнів займається розстановкою вже готових об'єктів, створенням сюжетних подій, заданням рухів об'єктів та персонажів по рівню. Рекламою гри, поширенням її реклами, проведенням презентації займається піар-менеджер.

Тестувальник програє різні версії гри, тестує її безперебійність в роботі в різних ситуаціях, він виявляє, специфікує та документує некоректності та помилки, недоопрацювання і недоліки.

## 1.2 Етапи розробки відеогри

Весь процес розробки гри можна поділити на декілька етапів. На початковому створюють загальний проєкт, пишуть дизайн-документ із описом всесвіту, всього ігрового процесу та сюжету, створюють графічні концепти для рівнів та персонажів, за якими визначатиметься стилістика гри та які будуть слугувати орієнтиром як для художників так і для тих хто займається моделюванням. Створюється дизайн-документ та зображення, які, зазвичай, використовують для привернення уваги видавців.

Залишається справа за фінансуванням, і якщо це питання вирішене, розпочинається повноцінна робота над розробкою гри, яка включає в себе і розробку чи налаштування наявного, ігрового рушія, також створення графічного, три вимірної та аудіо-забезпечення гри, введення з обов'язковим тестуванням ігрових механік, ігрового дизайну. Для отримання прикінцевої версії гра проходить перевірку та тестування її альфа та бета версій, часом проводяться відкриті тестування за допомогою гравців (так звані відкритті альфа чи бета-тести). Вже після всіх цих етапів гру можна тиражувати для продажу. Після виходу гри ще є етапи супроводу під час яких її можна вдосконалити та доповнювати.

### 2.2.1 Проєктування (пре-продакшин)

Зазвичай, якщо я маю і оприлюднюю ідею будь-чого, не виключенням є і розробки гри, я і стаю керівником проєкту. Далі йде процес розподілу ролей та відповідальностей, чим можна привернути увагу гравців; чи гра дасть змогу привнести щось кардинально нове чи ж використовуватимуться вже відомі та перевірені сучасні технології, вибір тематики. Вже враховуючи заплановані

переважаючи дії гравця, обирають жанр, наприклад, стратегію в реальному часі чи, можливо, шутер від першої особи. Важливо зазначити, що під час роботи жанр може бути змінено і це потрібно обов'язково врахувати в концепції, яку потрібно переробити, можливо навіть суттєво.

Гейм дизайнер приймає рішення стосовно майбутніх образів персонажів, яким буде в цілому ігровий світ, ним розробляються попередні малюнки – так званий концепт-арт. Гра отримує робочу попередню назву, яку, швидше за все буде змінено.

Відбирається команда однодумців – це: розробники, програмісти, сценаристи, художники, композитори, менеджери та інші. Вже вони шляхом обговорень чи мозкових штурмів визначають строки виконання для кожного із видів робіт, орієнтовну дату виходу вже готової гри, скільки та яке обладнання, об'єм фінансування, котрі для цього необхідні.

Саме результатом цього етапу є складання концепт-документу, яким обґрунтовується доцільність створення задуманої гри. Дизайн-документом описується гра в цілому, задум її початкового сюжету, весь ігровий процес, в ньому міститься концепт-арт. Також має створюватись прототип, спрощена початкова версія гри, по якій вже можна уявити бачення ігрового процесу.

### 2.2.2 Особливості виробництва програмного продукту типу відеогра

Як всі програмні продукти функціонал відеогри, технічні можливості, контент як і весь ігровий процес, забезпечує програмний код. Таким чином процес розробки містить такі самі етапи, які передбачені для розробки програмного забезпечення, крім цього більше часу слід приділити контенту та створенню ігрових механік.

Важливим моментом є те, що сучасні ігри часто складаються із готових програмних модулів, це так звані ігрові рушії, в яких вже реалізовано базові

функції, за допомогою яких поєднується графіка зі звуком об'єктами та їх рухами. Проте, щоби налаштувати рушій під реалізацію конкретного задуму програмісти проводять доопрацювання його, для отримання потрібних функцій. Існують як повністю безкоштовні ігрові рушії, так і такі, для використання яких потрібно отримати ліцензію. Крім цього для різних рушіїв можуть бути різні ліцензії.

Є різні типи рушіїв, є такі, що розраховані для створення ігор певного жанра, а є універсальні. Не всі рушії може забезпечити однакові внутрішньоігрові можливості та однаковий рівень графіки. За допомогою рушіїв можна створювати ігри під різні платформи, наприклад Unreal DevelopmentKit підтримується розробка інтерактивних продуктів для PC, Xbox360, PlayStation3, Wii,Android.

Певний клас ігор можна створювати у спеціально призначених програмах, в яких відразу передбачено початкові базові ресурси та дії, які не потребують ґрунтовного знання мов програмування, наприклад програми GameMaker, Construct,RPG Maker.

### 2.2.3 Процес розробки контенту

Традиційно у відеоіграх передбачено створення графіки, звукових ефектів та внутрішньо ігрових текстів. Концепт-арт виконують на папері чи за допомогою комп'ютера, і часто він складається із кількох варіантів. На їх основі художники затверджують і створюють дво- чи тьох- вимірні моделі персонажів, предметів і декорацій. Робиться це за допомогою програм, призначених саме для робіт пов'язаних із графікою.

Для того щоби надати моделям рухової активності, їх анімують за допомогою вже іншого спеціалізованого програмного забезпечення. Для цього створюють різні набори з рухів, які будуть відтворюватись в залежності від конкретних дій гравця при допомозі програмного коду. У випадку із дво-

вимірною графікою – це набори спрайтів, в яких кожна картинка є у вигляді окремого кадру. Для того, щоби рухи та емоції були більш реалістичними застосовуються моделі рухів живих акторів. Після того як рух зафіксовано за допомогою датчиків їх переносять на комп'ютерний персонаж.

За допомогою візуальних ефектів гра стає більш видовищніша та набуває певного стилю. Серед тих, що надають реалістичності є ефект тіні, заломлення світла, також постріли чи вибухи. За допомогою інших позначаються стани та дії персонажу, якими визначається стиль виконання гри. Деякі ігри повністю стилізуються як комікси деякі у вигляді кіно. Для реалізації картинок та звуку використовуються спеціальні графічні і звукові рушії.

Для виконання звукового оформлення зазвичай використовується чи пишеться музика та проводиться озвучування ігрових персонажів. Крім цього, щоби отримати повноцінне звукове оформлення використовуються звукові ефекти, наприклад звуки які імітують кроки чи постріли. Зазвичай їх можна обрати в безкоштовних бібліотеках, або ж записати самостійно. Існують спеціальні композитори, які спеціалізуються на створенні ігрової музики та звукових ефектів. Також музичний супровід може виконуватись як оркестрово так і пісенними групами. Діалоги персонажів озвучують спеціальні актори на студіях звукозапису.

Часом буває, що в іграх є відео- вставки, які створюються за допомогою програм дво- чи трьох- вимірної анімації. Часом для реалістичності для відео вставок знімають живих акторів, які на цьому спеціалізуються та будують декорації. А вже сюжети, діалог, додаткові деякі тексти пишуть сценаристи і відповідальні за це письменники.

#### 2.2.4 Процес розробки механіки гри

Саме механіка визначає насиченість всього процесу, та обумовлює правила, з дотриманням яких відбувається процес гри в цілому. Під цим терміном розуміють сукупність ігрових об'єктів, зокрема персонажів, об'єктів, із якими вони можуть взаємодіяти, декорацію. Частиною для механіки вважається керування за допомогою якого гравець може управляти як персонажем так і усім ігровим світом. Крім цього на етапі розробки ігрової механіки створюють і користувацький інтерфейс, за допомогою якого гравець інформується та може взаємодіяти з ігровим світом.

Як правило ігри діляться на локаційні рівні. Це робиться для того, щоби комп'ютер не перенавантажувати опрацюванням всього ігрового світу. Дизайнер рівнів розташовує готові об'єкти у світі гри та продумує їх рухову активність. За допомогою компонування рівнів визначається наскільки гра буде цікава та якими можливостями для вирішення конкретної ситуації володітиме гравець.

Фізичний рушій, у свою чергу, «відповідає» за процес взаємодії між об'єктами, який відбувається поза контролем гравця. Наприклад, ним реалізуються закон інерції, гравітація, поведінка рідин та властивості предметів. А вже сам штучний інтелект «займається» поведінкою персонажів та за їх реакцію на дії гравців. Велика кількість подій у грі реалізується за скриптами. Безпосередньо самі ж події вигадуються сценаристами, а вже скриптами займаються програмісти.

#### 2.2.5 Етап тестування [33]

Після етапу роботи над кодом, опрацювання контенту та механіки, без яких гра не зможе працювати, зазвичай проводять її допрацювання. Тобто, так звана

альфа-версія гри ще не є «зібраною», але в неї вже можна грати. Зрозуміло, що у такій версії можуть бути помилки та не доопрацювання, наприклад, відсутність деяких можливостей, музичного наповнення чи об'єктів. В загальному для виявлення проблем, працюють тестувальники, обов'язком яких є проходження гри, при цьому вони повинні по максимуму користуватись всіма передбаченими можливостями. Також вже на цьому етапі розробниками записуються рекламні трейлери в яких детально показують ігровий процес на відео, таким чином це дає уявлення для потенційних гравців уявлення про продукт.

На дещо пізнішому етапі випускається так звана бета-версія. До її тестування вже варта залучати потенційних покупців розробки. На етапі бета-версії продовжується процес виявлення помилок, перевірки коректності взаємодії між об'єктами ігрового світу, її керування. Також можна вносити зміни до оформлення, в ігровий баланс, можливості персонажів.

#### 2.2.6 Етап випуску та реалізації

За весь процес та проходження процесу продажу розроблюваного продукту зазвичай відповідальність несе видавець гри, вже безпосередньо від котрого розробник отримує частину виручених коштів. У випадку коли як обидвоє – розробник так само як і видавець є частиною однієї компанії, розподіл отриманого прибутку здійснюється за затвердженими керівництвом правилами. В складі фахівців сторони видавця є піар-менеджери та ігрові журналісти, котрі знають як та відповідно оприлюднюють інформацію про взагалі існування гри, задіяні та проводять рекламну кампанію, викладають відео матеріали, публікують новини у спеціалізованих джерелах, демонструють гру на популярних виставках відеоігр.

До обов'язків видавця вже входить локалізація гри, тобто переклад її тексту і звукового супроводу, адаптація до законів і культурних особливостей країни, на

аудиторію якої продукт видаватиметься. Цим процесом займається команда фахівців спеціалістів, котрі, окрім зазначеного ще й адаптують гру під гравців з конкретних країн. Характерною особливістю сучасних ігор є те, що текст та звуки є окремими файлами, і це значно спрощує процес локалізації.

Вже коли роботу закінчено, гру локалізовано її потрібно записати на носії, чи завантажити в хмару і надати потрібно доступи, при цьому версії для завантаження поширюють через сервіси для цифрового поширення, наприклад подібні до Steam.

### 2.2.7 Процес підтримки

Після того як гра випущена у ній все ще можуть виявити до тепер не знайдені помилки. Також в протестованій грі користувачів можуть не задовільняти ті чи інші аспекти, наприклад, складність. В таких випадках розробниками випускаються патчі, за допомогою яких можна усунути помилки і таким чином якісно вдосконалити продукт. Самі по собі і розробка так само як і випуск патчів не є джерелами додаткового прибутку, проте за допомогою них є можливість збільшення продажів гри, бо вже враховуються побажання користувачів та корегуються не виявлені попередньо помилки.

Для того, щоби підтримувати інтерес до гри до неї можна випускати доповнення, в яких, окрім виправлених помилок, додаються нові можливості, доповнення, додатково розвивають сюжетну лінію. Зазвичай доповнення можуть бути і платні і безкоштовні. Через мережу Інтернет розповсюджують доповнення типу DLC(Downloadable content), які у своїй більшості будуть платними і в яких суттєвих змін гри не передбачено, проте є нові можливості та варіанти оформлення персонажу.



## 2.3 Конструювання програмної системи

Проект – набір ресурсів, контенту і коду, які працюють повідомляючи і видають фінальний результат для кінцевого користувача. Всі ці ресурси зберігаються на жорсткому диску в тій же самій системі, що і всередині запущеного проекту і в Content Browser'е.

На жорсткому диску проект представляється папка з усіма його ресурсами, так само файломuproject, який визначає, які ресурси з ним пов'язані і які завантажувати в редактор для подальшої роботи. Простою мовою, це як .exe файл, який завантажує ресурси програми, тільки в нашому випадку спеціально для редактора UE4.

### 2.3.1 Обґрунтування редактору Unreal Engine

В основі редактора Unreal Engine знаходиться редактор рівня зображень. Такий режим є доступним по замовчуванню та відображається як головний редактор, саме по цій причині його вивчення відіграє важливу роль при вивченні рушія. Для початку редактор, разом із редакторами контенту, оскільки редактор займає забагато місця і не «поміщається» не лише на одному але й на 2-х екранах, стискають до розміру кількох вікон. Слід зазначити, що окремі розділи редактору показані лише тоді, коли виникає необхідність в редагуванні чогось конкретного. Інші розділи стають доступними тоді, коли виникає необхідність змінити потрібні властивості обраного об'єкта. В загальному існує декілька основних навігаторів контентного вмісту розроблюваної гри завдяки яким є можливість отримання доступу до якого-небудь об'єкта або до його властивостей, передбаченим розроблюваним проектом.

Таблиця 1.2 – Таблиця навігаторів контентного вмісту гри

ProjectBrowser	Дозволяє отримати доступ до ваших проєктів. Відкрити існуючий чи створити новий
ContentBrowser	Панель, за допомогою якої отримується доступ до вашого контентного вмісту
GlassViewer	Список усіх класів, доступних у розроблюваному проєкті
Global AssetPicker	Інструмент зі швидким доступом до об'єктів, передбачених у розроблюваному проєкті.

Також в UE4 передбачено декілька редакторів властивостей, які я перелічив далі у таблиці 1.3.

Таблиця 1.3 – Таблиця передбачених в UE редакторів властивостей

PropertyMatrix	Для зміни властивостей одразу декількох об'єктів передбачено редактор для властивостей об'єктів
ColorPicker	Панель призначена для вибору кольору.
CurveEditor	Редактор ламаних

Крім цього в наявності є кілька вікон для налаштування, у яких можна міняти як параметри безпосередньо самого редактора так і інших ваших проєктів, див. табл.1.4

Таблиця 1.4 – Вікна для налаштувань

EditorSettings	Вікно із налаштуваннями редактору UE4
ProjectSettings	Вікно з призначенням для змін налаштувань проєкту
KeyBinding	Вікно для керування гарячими клавішами.

### 2.3.2 Обґрунтування застосування Blueprint Interface

Використання блюпрінт інтерфейсів дозволяє абсолютно різним типам об'єктів використовувати якусь із або загальну функціональність. Наприклад, можна мати абсолютно різні типи об'єктів (дерева, машини, меблі), проте всі вони будуть наслідувати якусь певну можливість, наприклад їх можна знищити і/чи

зруйнувати. При створенні блупрінт-інтерфейсу з функцією `ontakewearonfire`, функція може бути викликана і використана, якщо в блупрінти (машина, дерево і т.д.) впроваджено даний інтерфейс.

Blueprint Interface можна створити через Content Browser, клікнувши на кнопку New, або через контекстне меню. Даний тип блупрінта знаходиться в підменю Blueprints, там само де і Macro Library.

Після створення, потрібно відкрити Інтерфейс в Блюпрінт редакторі і налаштувати функції.

Блюпрінт Інтерфейси редагуються в спеціальній версії Блюпрінт редактора. Оскільки я не можу створювати свої змінні, графіки або компоненти, процес редагування Інтерфейсів сильно відрізняється від редагування звичайних Блупрінтів.

Після створення, Blueprint Interface не матиме жодних функцій.

Функції – головний елемент Блюпрінт Інтерфейсів, які не мають реалізації самі по собі. Вони існують лише з набором вхідних і вихідних контактів. Це може бути використано для передачі інформації через Інтерфейс, для зміни значень в Блупрінтах, в які впроваджений Інтерфейс.

Для того щоб додати функцію, потрібно натиснути на кнопку «дати функцію» на категорії функцій «Категорія функцій», після чого ввести ім'я нової функції. Далі створиться новий графік разом з функцією. Важливо акцентувати увагу на те, що графік не активний і не має жодних контактів. Всю функціональність (функції) потрібно додавати в конкретному об'єкті, до якого вона буде застосовуватись.

Оскільки на даному етапі, функція не має ніякої функціональності та реалізації, все, що можна зробити, це створити вхідні і вихідні контакти:

1. В панелі Details, в розділі Input (вхід) або Output (вихід) натиснути кнопку (new)

2. Ввести назву параметра і встановити тип. Також параметр можна розкрити кнопкою, що б встановити стандартне значення. Воно буде використовуватися, якщо до контакту нічого не додасться.

Для створення великих ландшафтів, вбудована в UE4 система працюватиме ефективніше, ніж СтатікМеш. Система ландшафтів використовує 4 байти на кожен вертекст, а СтатікМеш використовує від 24-х до 28-ми.

Таким чином, Ландшафт потребує приблизно в 6 або 7 разів менше пам'яті при такій самій щільності вертексів. Сам ландшафт зберігає інформацію у вигляді текстури, і «вміє» вивантажувати невживані LOD'и на дальніх дистанціях. Завдяки цьому, ландшафт завантажуються з диска в міру приближення до його окремої частини.

### 2.3.3 Робота з освітленням

Для того, щоби сцена виглядала привабливо, може знадобитись навик правильно працювати з освітленням і Unreal Engine 4 надає широкий набір інструментів для роботи зі світлом: від простих джерел світла, які є основою створення освітлення сцени, до прорахунку статичного світла та тіней, завдяки чому можна отримати найвищу якість освітленості картки або рівня.

Освітлення сцени створюється завдяки розставленим джерелам світла там, де це необхідно. Завдяки цим джерелам, UE4 «розуміє», які ділянки потрібно освітлювати, а які затемнювати. За тим самим принципом, залежно від типу джерел, визначати, якої якості та яким повинне бути освітлення, який тип тіней використовувати і так далі.

У Unreal Engine 4 передбачено чотири типи джерел світла:

- Directional (направлене)	в основному, використовується для вуличного освітлення, від віддаленого джерела, наприклад, сонця
----------------------------	---

- Point (точкове джерело світла)	– звичайна «лампочка». Випромінювання світла у всі сторони
- Spot (прожекторний)	подібно до попереднього також випускає світло з однієї точки, але в одну сторону, наприклад як ліхтарик
- Sky (небесний)	джерело світла копіює фон сцени і застосовує його як загальне освітлення для мешів на відповідному рівні. Іноді називають Ембієнт освітленням

Для кожного джерела світла на вкладці «Transform» знаходиться властивість, яка називається «Mobility» («Рухливість»).

В загальному доступними є три параметри: статичний, стаціонарний, рухливий (Static, Stationary і Movable, відповідно). Кожен із них досить суттєво впливає на роботу джерела світла, що у свою чергу відображається на продуктивності.

Види рухливості джерел світла:

Статичне	повністю статичне, яке розраховується в окремому процесі і майже не витрачає ресурсів комп'ютера в процесі гри
Стаціонарне	може змінювати свій колір і насиченість в реальному часі, але не може переміщатися, обертатися або масштабувати силу світла. Прораховується окремо, проте може відкидати тіні від динамічних об'єктів
Рухоме	повністю динамічне, здатне змінювати всі свої характеристики в режимі реального часу і створювати повністю динамічне освітлення

#### 2.3.4 Використання інструментів і моделі SpeedTree в додатку Unreal Engine 4

Процес використання SpeedTree з UE4 можна поділити умовно на три кроки:

1- створення або придбання моделі SpeedTree. В будь-якому випадку: чи то використання моделі з існуючої бібліотеки чи то створення з нуля використовуючи SpeedTree Modeler, для того щоби розпочати роботу потрібна модель. Моделі складаються із файлів SpeedTree (.spm, SRT), текстур, компонентів сіток(.obj, .fbx) які з'єднуються між собою.

Моделі SpeedTree імпортуються в UE4 , як і будь-який інший актив. Вони приходять у вигляді статичної сітки з додатковими атрибутами для реалізації і гладкою LOD.

2- місце моделі у вашій сцені. Після імпорту моделі SpeedTree можуть бути розміщені або розфарбовані, як і будь-який інший елемент статичної сітки.

3- Отримання моделі для статичного освітлення. Якщо планується використання статичного освітлення, необхідно налаштувати lightmapping координати УФ - випромінювання. SpeedTree автоматично створить набір координат текстури для моделі, придатної для використання з Lightmass UE4. Якщо не потрібно використовувати статичне освітлення для моделі цей крок можна пропустити, в іншому ж випадку, віднести потрібно відповідально.

### 2.3.5 Імпорт моделей в UE 4

При збереженні моделі в версії UE4 з Modeler потрібно мати .srt файл, який представляє саме мою модель. Це файл, який потрібно буде імпортувати в редактор UE4. Таким чином модель SpeedTree буде імпортовано в звичайний статичний об'єкт сітки.

Імпорт файлу .srt, як і інших активів. Щоб зберегти отримані активи можна зробити папку для моделі, натиснувши кнопку "Імпорт".

В цілому є кілька варіантів при імпорті, які представлені в діалозі імпорту. По-перше, є вибір між 3D геометрією. Якщо дерево буде використовуватися тільки крупним планом, то "герой" – це дерево, в іншому ж випадку виникають

проблеми з найнижчою LOD (як це іноді буває динамічні тіні падають на плоску геометрію), тут як варіант можна вибрати «ігнорувати сітку». Або, якщо дерево буде використовуватися лише далеко, і є бажання зберегти завантаження 3D-геометрії, можна імпортувати лише рекламний щит.

У діалоговому вікні імпорту, також можна вибрати варіант щоб включити деталі і блиск, в матеріалах, використовуваних в 3D дереві. Можна також додати обробку вершин до матеріалів, яка ставиться особливим вузлом SpeedTree в матеріал, який може обробляти вітер, billboarding і інші ефекти.

Таким чином, після завершення імпорту мають бути в наявності такі нові активи:

1. Статична сітка, яка представляє модель з можливістю змінити параметри на цей та інші активи статичної сітки (наприклад, LOD діапазони, лайтмап дозвіл і т.д.).
2. Текстура активів по кожному атласі, щиті, і заповнення текстури карти моделі.
3. Матеріальні цінності для кожного компонента (малювати) виклик моделі.

Слід також зазначити, що матеріали мають особливе значення і є можливість редагування генерованих матеріалів, які є базовими, але можуть бути поліпшені і адаптовані відповідно до запланованих сцен.

Після того, як модель було імпортовано як актив, є можливість помістити її у сценах, як і інші статичні меші. Найбільш поширеними є:

1- розміщення в ручному режимі. Для цього достатньо просто перетягнути актив і помістити його в сцену для того щоб помістити модель. Це надає гнучкості. Потрібно відзначити їх як "статичні" (для цього потрібен хороший Lightmap щоб була можливість використання з Lightmass або "рухомі" для динамічного активу.

2- розфарбовування. Потрібно перетягнути актив в області листя і пофарбувати моделі.

3- додавання вітру на сцені. Додати вихідний об'єкт «напрямок вітру» (для цього потрібно обрати "Вікно перегляду → клас", пошук "WindDirectionalSource", і перетягнути його на сцену). "Швидкість" за замовчуванням 1.0, саме вона вважається максимальною для вітру для моделей SpeedTree. Чим нижче це значення тим нижча сила вітру, початкове значення 0.1 – спокійний вітер.

4- рівень деталізації. LOD переходи – космічний напрям в UE4. Ці значення редагуються шляхом редагування статичного активу сітки, пов'язаними з моделлю SpeedTree. Значення за замовчуванням вибираються на імпорт та працюють для типових ситуацій.

5- обчислення LightMap UV координат. Для того, щоб використовувати статичне освітлення (Lightmass), в UE4 потрібно обчислити набір UV координат для моделі. Це УФ відображення для моделі згадується як Lightmap. Отримання хорошого Lightmap є частиною науки і частиною мистецтва.

6- перехід до відображення щільності LightMap. Використання на панелі інструментів кнопки "Рендер"-перемикання активізується режим рендеринга для "Lightmap щільності".

За допомогою цієї візуалізації можна показати, наскільки добре лайтмеп-дозвіл розподіляється по моделі і призначений точно відповідати одному і тому ж режиму візуалізації в UE4 ("Показати → Lightmap щільність"). Зелені ділянки моделі являють собою ідеальну щільність. Синій колір вказує, що дозвіл лайтмеп є занадто низьким і може показати артефакти, коли лайтмап побудований в UE4. Червоний вказує на лайтмеп-дозвіл занадто високий і не оптимальний для цієї геометрії.

Сам лайтмеп функція відображається в нижньому лівому кутку. Потрібно обрати частину моделі і тоді можна побачити, де вона представлена в Lightmap. Можна також вибрати трикутники в Lightmap, щоб побачити, яка частина моделі задіяна.



Рівень застосування Lightmap залежить від вимог до проєкту і складності. Проте метою є використання мінімального, аж до базового рівня, застосування Lightmap, проте з достатнім охопленням місця розміщення моделі у сцені.

Потрібно встановити цільовий дозвіл Lightmap, вибравши генератор "дерево" і редагування: властивість "Lightmap Resolution". Цілком можливо, що модель може бути занадто складною, щоб відповідати обраному доступу. В такому разі у вікні виводу з'явиться попередження і дозвіл буде перепросимасо до базового, яким передбачено принаймні один тексель в Lightmap при збереженні одного текстеля кордону, потрібного для уникнення візуалізації артефактів.

Властивість "Lightmap: Display мультиплікатор" покаже щільність Lightmap або більшу, ніж цільової розмір. Ця опція використовується для того, щоб помістити Увс в меншу роздільну здатність карти і переглядати те, як це буде виглядати для великих карт. Такий підхід дозволить розмістити Lightmap Увс в невеликий простір і як і раніше експериментувати з більш високою роздільною здатністю.

7- налаштування LightMap. Після того як було обрано дозвіл наступним кроком потрібно налаштувати, як лайтмеп розподілятиметься по моделі. Це потрібно для того, щоби переконатися у наявності необхідної щільності у частині моделі, де це необхідно перш за все. Зазвичай найбільша щільність є бажаною у, наприклад, стовбурі, нижніх гілках та корінні. На цю область гравець найчастіше звертає увагу і яка, у свою чергу, найбільш схильна до очевидних артефактів, якщо щільність занадто мала. Для цього варта звернути увагу на комбінацію з наступних інструментів для тонкого налаштування лайтмеп розподілу щільності:

Скидання	Виберіть "Інструменти → Reset лайтмеп", щоб встановити всі Lightmap скаляри до 1,0, за винятком листя. Базово вони встановлюються на 0.0, таким чином кожен лист отримує рівно один тексель
Матеріал Ваги	Кожен матеріал має властивість, так звану зване "Unwrap масштабу". Збільшення або зменшення цього значення призведе до зміни відносного розміру кожної з частин геометрії.

Генератор	Кожна комбінація генератор / вузол, який може вмістити геометрія має "Lightmap: Масштаб" властивість. Зміна цього атрибуту призведе до зміни розміру простору обраного об'єкта в Lightmap.
-----------	--

## 2.4 Апаратні засоби та програмне забезпечення

Як і будь-яке інше спеціалізоване програмне забезпечення UnrealEngine не є виключенням, і має деякі специфічні вимоги до апаратного та програмного забезпечення для запуску гри і редактора або розробки із рушієм.

Таблиця 1.5 – Таблиця системних вимог

Рекомендоване обладнання	
Операційна система-	Windows 7/8/10 64-bit
Процесор-	Intel i7 or AMD, 3.5 GHz чи вище
Пам'ять-	8 GB RAM
Відео карта-	DirectX11

Нижче наведені вимоги для запуску двигуна або редактора.

Таблиця 1.6 – Таблиця мінімальних вимог для запуску рушія

Запуск двигуна Engine	
Операційна система-	Windows7/8/10 64-bit
DirectX час виконання-	DirectX End-User
VisualStudio Version-	VisualStudioProfessional чи Visual Studio Community: – Pro – Community
iOS Development-	

Вимоги, що пред'являються для оптимальної роботи фахівця, який працює з рушієм перераховані в таблиці 1.6

## 2 ТЕСТУВАННЯ ПРОГРАМНОЇ СИСТЕМИ

### 2.1 Тестування можливостей графічного рушія типу Unreal Engine

На цьому етапі я провів претестування нового графічного рушія типу Unreal Engine версія 4. Попередньо я розглянув можливості рушія та версії Daylight і зараз я представляю результат мого дослідження, зокрема ефективність роботи відеокарти в більш потужних ігрових додатках, таких як представлено на рисунку 2.1.



Рисунок 2.1 – Логотип рушія

В якості основи для тестування я обрав гру, яку побудував з його використанням– це Bioshock Infinity.

Перевагою UnrealEngine є те, що він підтримує основні операційні системи по типу версій Windows. Основні системні вимоги для візуалізації максимальної якості представлено на рисунку 2.2. Стосовно інших операційних систем: від розробників на сьогодні даних немає, але, цілком можливо, над цим ведеться робота.

	<b>Рекомендовані (FullHD)</b>	<b>Ефективні (UltraHD)</b>	<b>Оптимальні (4K)</b>
<b>Процесор на базі платформи Intel</b>	Core i 3 2100 3,1 ГГц	Core i 3 2100 3,1 ГГц	Core i 5 2500K 3,3 ГГц
<b>Процесор на базі платформи AMD</b>	AMD FX-6300 3,5 ГГц	AMD FX-8350 3,6 ГГц	AMD FX-9570 5,0 ГГц
<b>Об'єм оперативної пам'яті</b>	3 гігабайти	4 гігабайти	4 гігабайти
<b>Відеокарна на базі відеопроеесора NVIDIA</b>	EFORCE GTX 770	EFORCE GTX 780	
<b>Відеокарна на базі відеопроеесора AMD</b>	Radeon R9 280X	Radeon R9 290X	
<b>Необхідний об'єм відеопам'яті</b>	1280 мегабайт	2048 мегабайт	3076 мегабайт
<b>DirectX</b>	DX11	DX11	DX11
<b>Місце на жорсткому диску</b>	5 гігабайт	5 гігабайт	5 гігабайт
<b>Операційна система</b>	Windows Vista	Windows	Windows

Рисунок 2.2 – Системні вимоги  
для максимальної якості візуалізації у Unreal Engine

Пріоритетним та, я вважаю, основним графічним API для Unreal Engine версії 4 є DirectX11. Оскільки, Unreal Engine версії4 в цілому вважається ігровим рушієм, який розроблено і підтримкою якого займається компанія EpicGames.

Перша гра, створена за допомогою на рушія Unreal побачила світ ще у далекому 1998 році. Будучи призначеним, в першу чергу, для «шутерів», від 1-ої особи, рушій добре себе зарекомендував і при створенні іншожанрових ігор.

Мовою, на якій написаний рушій є C++. Рушій Unreal Engine версії 4 дає можливість створювати ігри під більшість операційних як систем так і платформ: MicrosoftWindows,Linux, MacOS і MacOS X, консолі X box, X box360, Play Station (версій 2,3), Play Station Portable,Wii, Dreamcast і NintendoGame Cube.

Для того, щоби спростити портування в рушії використано варіант модульної системи залежних компонент: підтримуються різноманітні системи рендерингу, зокрема, такі як:Direct3D,OpenGL,Pixomatic, засоби для голосового

відтворення тексту, розпізнавання мови, модуля призначені для роботи із мережею та підтримка різноманітних пристроїв введення.

Для мережевих ігор передбачено підтримку технологій типу WindowsLive, XboxLive, а також GameSpy, з можливістю до 64-ох гравців одночасно. Незважаючи на те, що в кошторис на розробку не передбачав багаточисельності користувачів на одному сервері, також рушій було використано і при створенні ігор типу MMORPG.

Таблиця 2.1 – Тестова частина

Тестова конфігурація	
Тестові стенди	Тестовий стенд №1 на базі платформи AMD Sокет AM3 Тестовий стенд №2 на базі платформи AMD Sокет AM3+
Мультимедійне обладнання	Монітор Dell U3010 Монітор Asus PQ321QE
Програмна конфігурація	
Операційна система	Window 8.1/10
Графічний драйвер	Nvidia GeForce/ION Driver Release 337/50
Програма моніторингу	Fraps GPU Mist

Усі відео карти було протестовано за максимальних показників графічної якості із використанням програми MSI Afterburner. Метою тестування було визначити, як саме поведуть себе відео карти різних виробників за однотипних умов експлуатації. За показник продуктивності приймався середній та мінімальний FPS. Дозвіл технологічної демонстрації прописується вручну шляхом редагування файлу GameUserSettings.ini, який викликається за допомогою edit settings.bat. Результати тестування за максимальних налаштувань якості memoryGPU представлено на рисунку 2.3

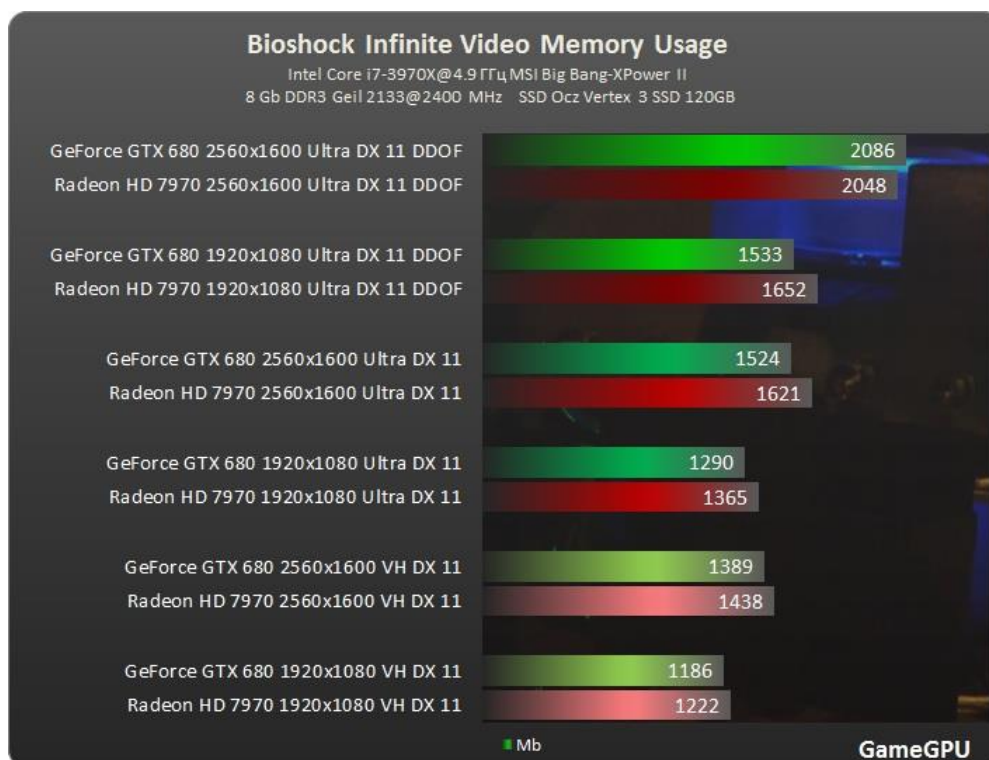


Рисунок 2.3 – Тестування процесора і відеокарти

За рекомендований обсяг використання відео пам'яті для розширення 1920x1080 є 1024Мб відео пам'яті, для розширення 2560x1600 є 1524Мб відео пам'яті і для розширення 3840x2160 близько 2048Мб відео пам'яті представлена на рисунку 2.4, зокрема тестування за максимальних налаштувань якості 1920 x1080, 2560x1600, 5760x1080.

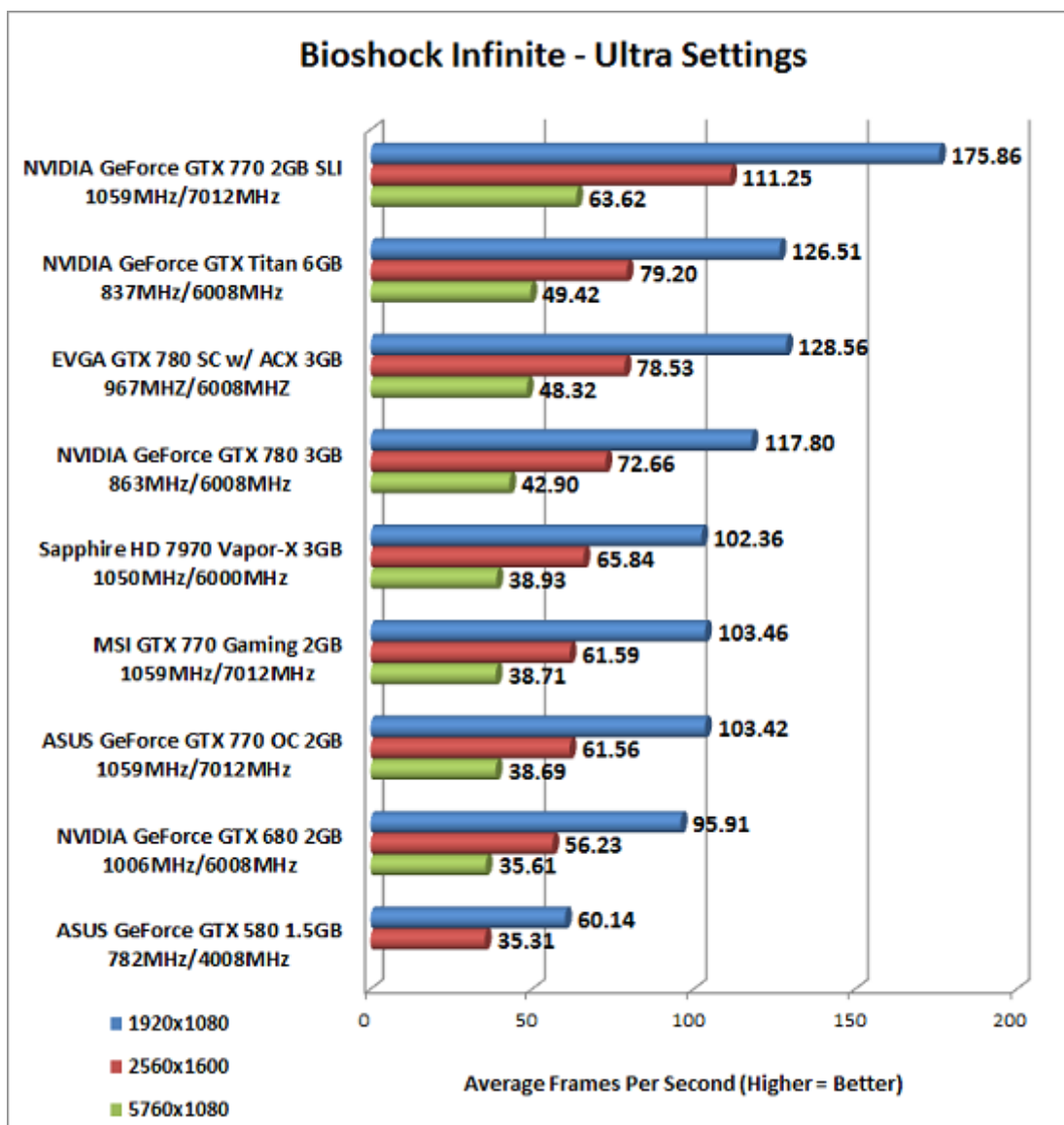


Рисунок 2.4 – Тестування відеокарти  
при розширенні 1920x1080,2560x1600,5760x1080

Таким чином, для вказаних налаштувань прийнятним показником FPS відзначено відео карти рівнів RadeonHD6870 або GeForceGTX580. Оптимальними ж, на мою думку, відео картами стануть RadeonHD7970 або GeForceGTX770 Gaming і вищих версій.

## 2.2 Тестування процесорів за максимальних налаштувань по якості

Аналіз результатів тестування процесорів було проведено на 15-ти моделях базових конфігурацій, які вважаються актуальними і на сьогоднішній день. Тестування було проведено при мінімальних значеннях навантаження відео карти при грі, тобто коли навантаження складало менше ніж 99%, цього разу при розширенні 1920x1080 при максимальних налаштуваннях якості графіки зображених на рисунку 2.5.

Результати тестування процесорів Intel та AMD при максимальних налаштуваннях якості з розширенням 1920x1080

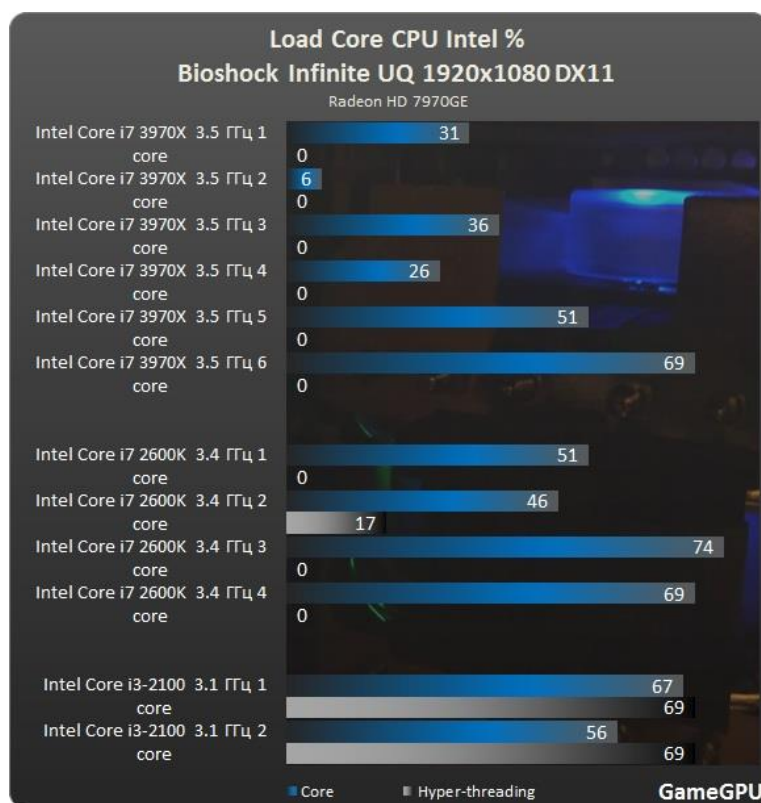


Рисунок 2.5 – Результат тестового навантаження процесорних ядер Intel при максимальних налаштуваннях якості з розширенням 1920x1080



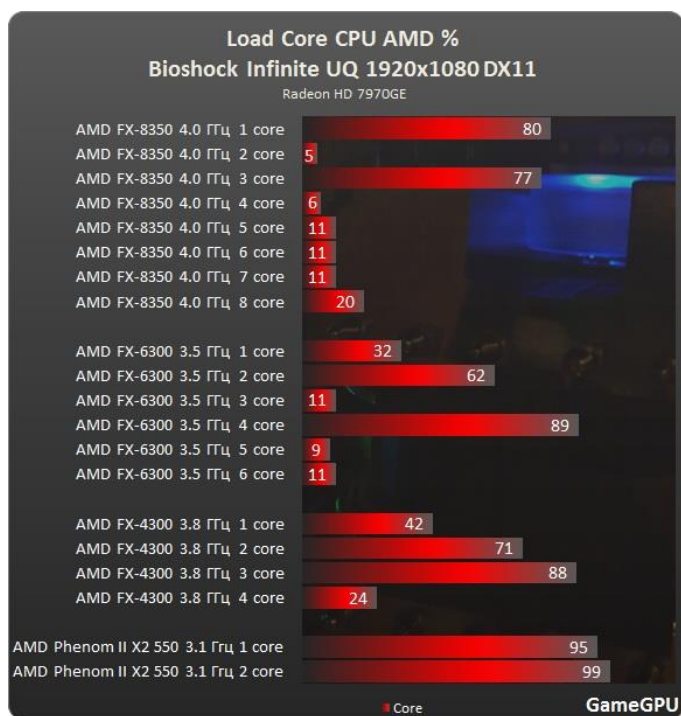


Рисунок 2.6 – Результат тестового навантаження процесорних ядер AMD при максимальних налаштуваннях якості з розширенням 1920x1080

При здатності підтримки до 4-ох обчислювальних потоків, це ніяким чином не повпливало на продуктивність.

Усі відео карти тестувалися за максимальних показників графічної якості при використанні програми MSI Afterburner. Метою проведення цього тестування було проаналізувати як себе поведуть відео карти різних виробників за однакових умов. За показник продуктивності було взято середнє і мінімальне FPS. Дозвіл технологічної демонстрації прописується вручну шляхом редагування файлу GameUserSettings.ini, який викликається за допомогою edit settings.bat. Результати тестування за максимальних налаштувань якості memoryGPU представлено на рисунку 2.7 на якому наведено результати тестування при максимальних налаштуваннях якості 1920x1080.

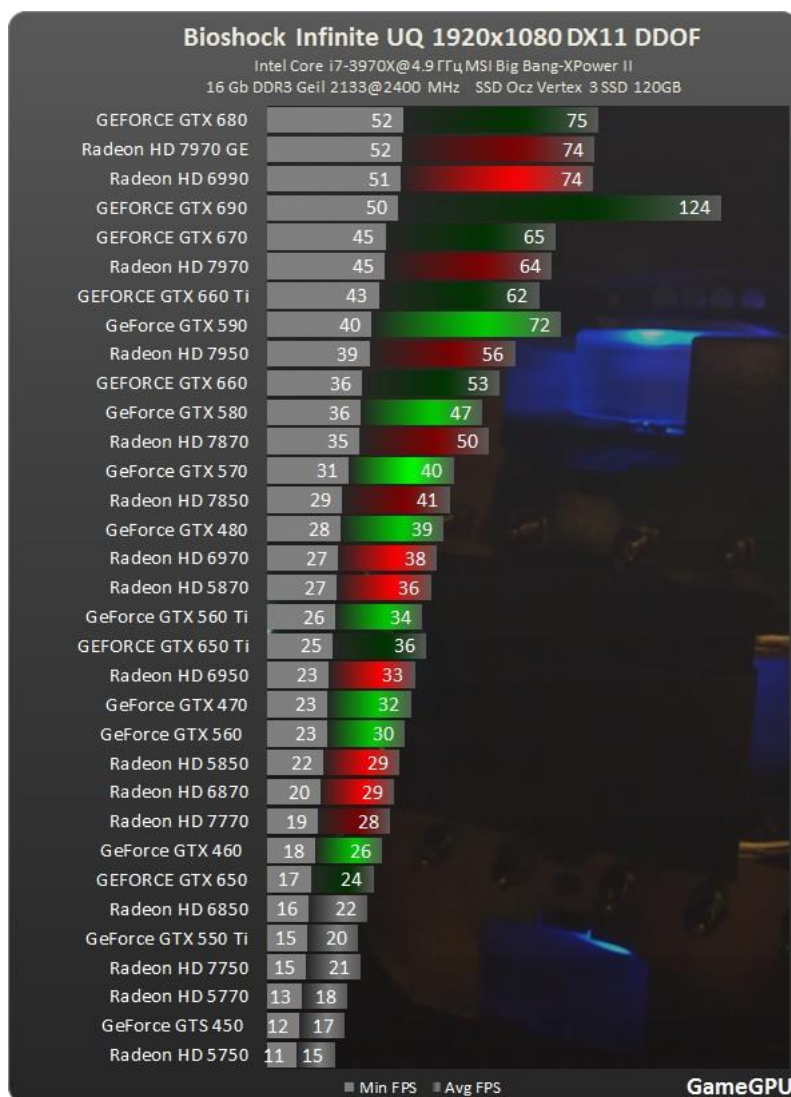


Рисунок 2.7 – Показник FPS при розширенні 1920x1080

При даному налаштуванні прийнятний показник FPS показали відео карти рівня Radeon HD7770 або GeForce GTX650. Оптимальним вибором будуть відео карти RadeonHD7970GE або GeForce GTX680 та вище.

Тестування на спожиту грою відеопам'ять проводилося при використанні програми MSI Afterburner.

За показник я взяв до уваги вільнодоступні результати від AMD та NVIDIA з розширеннями 1920x1080 та 2560x1600 та відповідно, а також з різними налаштуваннями згладжування, як представлено на рисунку 2.8.

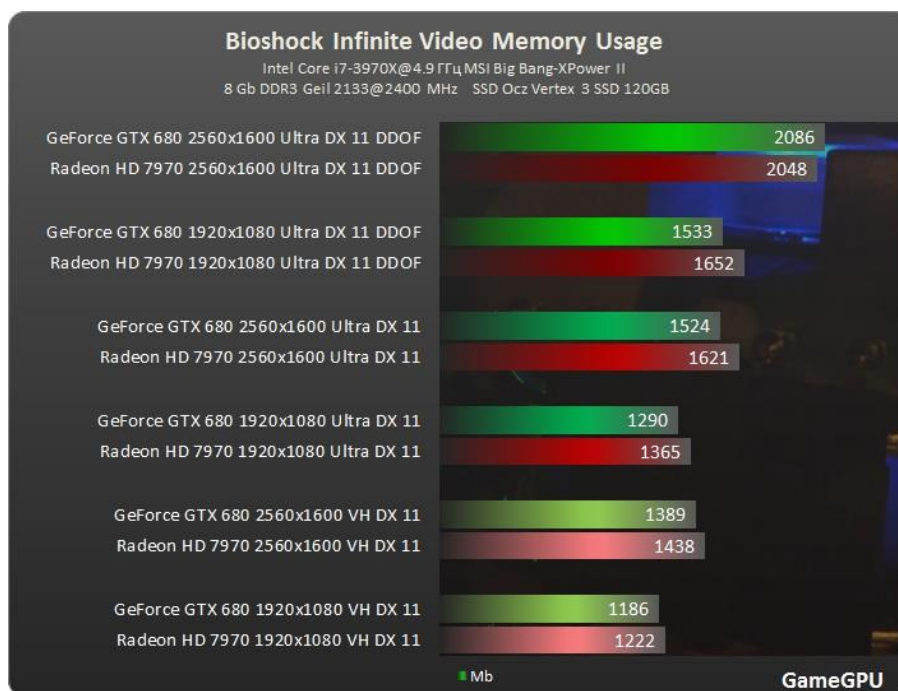


Рисунок 2.8 – Споживання грою відеопам'яті при максимальних налаштуваннях

Таким чином, можна зробити висновки про те, що рекомендованим обсягом використання відео пам'яті для розширення 1920x1080 буде 1365Мб відео пам'яті, для розширення 2560 x 1600 стане 2048Мб відео пам'яті, а для розширення яке складає 3840 x 2160 близько 2048Мб відео пам'яті.

Демонстраційна версія може підтримувати до 5-ти обчислювальних потоків, проте, при досить низькій продуктивності процесу це не є хорошим результатом.

Отже, тепер я можу зробити певні висновки стосовно ігрового рушія типу Unreal Engine версії 4.

Перш за все, таке рішення однозначно піднімає планку якості графіки на абсолютно новий рівень і значно полегшує роботу розробника при створенні ігрового проєкту. При цьому, істотно зростуть і системні вимоги до ігор ь.

Це в свою чергу може означає, що для гри в Full HD буде необхідним допустимий мінімум із RadeonHD7770 чи GeForceGTX650. Мінімальне значення FPS при 25 кадрах зможуть забезпечити вирішення не нижче RadeonRadeonR9 280X або GeForceGTX770. Для більш високих розширень буде необхідна наявність підтримки іграми мульти процесорних SLIта CF.

## 3 БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

### 3.1 Безпека в надзвичайних ситуаціях

#### 3.1.1 Запобігання виникненню надзвичайних ситуацій [34-36]

Запобігання виникненню надзвичайних ситуацій – це підготовка та реалізація комплексу правових, соціально-економічних, політичних, організаційно-технічних, санітарно-гігієнічних та інших заходів, спрямованих на регулювання безпеки, проведення оцінки рівнів ризику, завчасне реагування на загрозу виникнення надзвичайної ситуації на основі даних моніторингу (спостережень), експертизи, досліджень та прогнозів щодо можливого перебігу подій з метою недопущення їх переростання у надзвичайну ситуацію або пом'якшення її можливих наслідків.

Зазначені функції запобігання надзвичайним ситуаціям техногенного та природного характеру в нашій країні виконує Єдина державна система запобігання надзвичайним ситуаціям техногенного і природного характеру і реагування на них, затверджена Постановою Кабінету Міністрів України.

Єдина державна система запобігання надзвичайним ситуаціям техногенного і природного характеру і реагування на них (ЄДСЗР) включає в себе центральні та місцеві органи виконавчої влади, виконавчі органи рад, державні підприємства, установи та організації з відповідними силами і засобами, які здійснюють нагляд за забезпеченням техногенної та природної безпеки, організують проведення роботи із запобігання надзвичайним ситуаціям техногенного та природного походження і реагування у разі їх виникнення з метою захисту населення і довкілля, зменшення матеріальних втрат.

Основною метою створення ЄДСЗР є забезпечення реалізації державної політики у сфері запобігання і реагування на надзвичайні ситуації, забезпечення цивільного захисту населення.

ЄДСЗР складається з постійно діючих функціональних і територіальних підсистем і має чотири рівні: загальнодержавний, регіональний, місцевий та об'єктовий.

Функціональні підсистеми створюються міністерствами та іншими центральними органами виконавчої влади для організації роботи, пов'язаної із запобіганням надзвичайним ситуаціям та захистом населення і територій від їх наслідків.

Кожний рівень ЄДСЗР має координуючі та постійні органи управління щодо розв'язання завдань у сфері запобігання надзвичайним ситуаціям, захисту населення і територій від їх наслідків, систему повсякденного управління, сили і засоби, резерви матеріальних та фінансових ресурсів, системи зв'язку та інформаційного забезпечення.

Координуючі органи ЄДСЗР:

- державна комісія з питань техногенно-екологічної безпеки та надзвичайних ситуацій;
- національна рада з питань безпечної життєдіяльності населення;
- комісії Ради міністрів Автономної Республіки Крим, обласних, Київської та Севастопольської міських державних адміністрацій з питань техногенно-екологічної безпеки та надзвичайних ситуацій;
- комісії районних державних адміністрацій і виконавчих органів рад з питань техногенно-екологічної безпеки та надзвичайних ситуацій;
- комісії з питань надзвичайних ситуацій об'єкта.

До складу ЄДСЗР входять відповідні сили і засоби функціональних і територіальних підсистем, а також недержавні (добровільні) рятувальні формування, які залучаються для виконання відповідних робіт.

У виняткових випадках, коли стихійне лихо, епідемія, епізоотія, аварія чи катастрофа ставить під загрозу життя і здоров'я населення і потребує термінового проведення великих обсягів аварійно-рятувальних і відновлювальних робіт, Президент України може залучати до виконання цих робіт у порядку,

визначеному Законом України «Про надзвичайний стан», спеціально підготовлені сили і засоби Міноборони.

На базі існуючих спеціалізованих служб і підрозділів (будівельних, медичних, хімічних, ремонтних та інших) в областях, районах, населених пунктах, підприємствах, установах та організаціях утворюються позаштатні спеціалізовані формування, призначені для проведення конкретних видів невідкладних робіт у процесі реагування на надзвичайні ситуації. Ці формування проходять спеціальне навчання, періодично залучаються до участі у практичному відпрацюванні дій з ліквідації надзвичайних ситуацій.

У виконанні робіт, пов'язаних із запобіганням надзвичайним ситуаціям і реагуванням на них, можуть брати участь також добровільні громадські об'єднання за наявності у представників цих об'єднань відповідного рівня підготовки, підтвердженого в атестаційному порядку. Свої дії вони повинні узгоджувати з територіальними органами та уповноваженими з питань надзвичайних ситуацій та цивільного захисту населення, а роботи виконувати під їх керівництвом.

Надзвичайний стан по всій території України або в окремих її місцевостях вводиться постановою Верховної Ради України з негайним повідомленням Президента України або Указом Президента України, який підлягає затвердженню Верховною Радою України.

Під час надзвичайного стану держава може вживати заходів, передбачених Законом «Про надзвичайний стан», відступаючи від своїх зобов'язань за Конституцією лише настільки, наскільки це вимагається гостротою стану, за умови, що такі заходи не є несумісними з іншими зобов'язаннями за міжнародним правом і не тягнуть за собою дискримінації на основі національності, мови, статі, релігії чи соціального походження.

З метою ліквідації наслідків надзвичайної ситуації у мирний час може проводитись цільова мобілізація. У виняткових випадках допускається залучення працездатного населення і транспортних засобів громадян для виконання

невідкладних аварійно-рятувальних робіт за умови обов'язкового забезпечення безпеки праці. При Цьому забороняється залучення неповнолітніх, а також вагітних жінок до робіт, які можуть негативно вплинути на стан їхнього здоров'я.

### 3.1.2 Аварії з викидом шкідливих речовин [36]

Аварії з викидом сильнодіючих отруйних речовин (СДОР) і зараженням навколишнього середовища виникають на підприємствах хімічної, нафтопереробної, целюлозно-паперової, м'ясо-молочної та харчової промисловості (що мають холодильні установки і які використовують у них в якості холодоагенту речовини типу аміак), водопровідних і очисних спорудах (що використовують хлор), а також при транспортуванні СДОР по залізниці і автомобільних дорогах.

Безпосередніми причинами викиду СДОР є порушення правил зберігання і транспортування, недотримання техніки безпеки, вихід з ладу агрегатів, механізмів, трубопроводів, ушкодження ємностей тощо. При цьому можливе ураження робітників, службовець і інших категорій населення, що знаходяться в районах викиду СДОР.

Сильнодіючими отруйними речовинами називаються хімічні сполуки, які в певних кількостях, що перевищують ГДК, шкідливо впливають на людей, сільськогосподарських тварин, рослини, викликаючи у них ураження різного ступеня.

СДОР можуть бути елементами технологічного процесу (аміак, хлор, сірчана й азотна кислоти, фтористий водень) і можуть утворюватись при пожежах на об'єктах господарства (оксид вуглецю, оксид азоту, хлористий водень, сірчистий газ).

У народному господарстві великого поширення набуло використання таких СДОР, як хлор (для знезараження води; при виробництві целюлози на

виробництво 1 т целюлози потрібно 40 кг хлору), аміак (при виробництві добрив; як холодоагент у холодильних установках), сірководень, сірковуглець та ін.

Об'єкти, які виробляють СДОР, використовують їх у процесі виробництва, здійснюють їх зберігання, поділяють на 3 ступені хімічної небезпеки. Ступінь хімічної небезпеки визначається видом СДОР та його сумарною кількістю.

Ступінь хімічної небезпеки дає змогу оцінити його з точки зору хімічної небезпеки для населення, сільськогосподарських тварин і навколишнього середовища та розробити засоби їх захисту. На території України у 140 містах та 46 населених пунктах розташовано 877 хімічно небезпечних об'єктів та 287 000 об'єктів використовують у своєму виробництві СДОР чи їх похідні.

Шкідлива дія СДОР на організм людини може проявлятися як в результаті потрапляння таких речовин у краплиннорідкому вигляді на шкіру, так і в результаті вдихання їх парів. За токсичним властивостям СДОР в основному відносяться до групи речовин загальноотруйної та задушливої дії. Симптомами отруєння ними в більшості випадків є головний біль, запаморочення, потемніння в очах, шум у вухах, наростаюча слабкість, задишка, нудота, блювота, а при сильних отруєннях - втрата свідомості, судоми і навіть смерть.

Внаслідок аварій із СДОР утворюється зона хімічного — зараження та осередок хімічного зараження.

Зона хімічного зараження СДОР включає територію, на яку поширюється хмара СДОР. Площі хімічного зараження СДОР визначаються напрямком і швидкістю вітру та іншими параметрами.

Осередок хімічного ураження включає територію, на якій відбулися масові ураження людей, тварин та рослин.

У населених пунктах, лісах, виробничих приміщеннях, підвалах і комунікаційних тунелях стійкість зараження СДОР буде вище, ніж на відкритій місцевості, оскільки вплив вітру проявляється у меншій мірі.

При цьому характер токсичної дії деяких СДОР може змінюватися, наприклад, пари хлору в концентрації 0,1-0,2 мг / л смерть людини можуть



викликати при вдиханні їх протягом не менше як 1 години, а пари в концентрації 10-15 мг / л викличуть рефлекторну зупинку дихання при 1-2 вдихах. Деякі з СДОР при високих концентраціях їх парів здатні викликати шкірні ураження людини (з утворенням пухирів).

В першу чергу необхідно захистити органи дихання від подальшої дії сильнодіючих отруйних речовин. На потерпілого необхідно надіти протигаз або ватяну марлеву пов'язку, попередньо змочивши її при отруєнні хлором водою або 5 % розчином питної соди, а при отруєнні аміаком – водою або 2% розчином лимонної кислоти, і винести (вивести) його із зони ураження. У випадках отруєння сильнодіючими отруйними речовинами потерпілому забезпечити спокій і тепло. При виході із зони зараження вимийте очі і відкриті ділянки тіла і зверніться за допомогою до медичного працівника або до медичного закладу.

При отруєнні аміаком винесіть потерпілого з зони зараження, шкіряні покриви, очі, ніс, рот вимийте водою. В очі закапайте дві три краплі 30% розчину альбуциду, в ніс оливкове масло.

При отруєнні хлором винесіть потерпілого із зони зараження. При зупинці дихання зробіть штучне дихання. Шкіряні покриви, рот, ніс вимийте 5 % розчином питної соди або водою.

При отруєнні метаном винесіть потерпілого із зони зараження. При зупинці дихання зробіть потерпілому штучне дихання.

При отруєнні чадним газом винесіть потерпілого із зони зараження, розстебніть комірець одягу. При необхідності зробіть штучне дихання. При необхідності зверніться за допомогою до медичного працівника або відправте потерпілого в медичний заклад.

При отруєнні СДОР дійте у відповідності з отриманими розпорядженнями управління (відділу) з питань надзвичайних ситуацій та цивільного захисту населення області (міста обласного підпорядкування, району).

У системі цивільної оборони розроблена «Методика прогнозування масштабів зараження СДОР при аваріях». Методика дає змогу розрахувати можливу площу хімічного зараження та оцінити можливі втрати людей.

### 3.2 Основи охорони праці [37,38]

Темою кваліфікаційної роботи є Розробка програми для створення та редагування ігрових моделей з використанням Autodesk 3D Max, Blue. При використанні ПЗ, яке є результатом даної розробки, як і при використанні будь-якого іншого ПЗ, необхідно дотримуватися рекомендацій та вимог з охорони праці. В цьому розділі пропоную розглянути основні нормативні документи, в яких зазначені вимоги до робочих місць та приміщень при використанні ПК.

При розробці системи потрібно запобігти негативному впливу виробничих факторів, а саме дотримуватися правил та норм щодо приміщень де знаходиться робоче місце, відповідно до «Правил охорони праці під час експлуатації електронно-обчислювальних машин» та ДСанПіН «Гігієнічні вимоги до організації роботи з візуальними дисплейними терміналами електронно-обчислювальних машин».

Відповідно до ДСанПіН робоче місце працівника не можна розміщувати у підвалах та цокольних поверхах. Площа робочого місця для виконання розробки повинна становити не менше 6 м<sup>2</sup>, а об'єм 20 м<sup>3</sup>. Приміщення, в якому працюють на ПК повинно бути забезпечене природнім та штучним освітленням згідно з нормами ДБН. Природне освітлення здійснюється крізь світлові отвори, направлені на північний схід і вони забезпечують коефіцієнт природньої освітленості (КПО) не менший ніж 1,5%. Розраховується КПО за методикою, що міститься у ДБН. В приміщеннях де проводиться розробка та міститься ПК, штучне освітлення здійснюється системою загального рівномірного освітлення. Джерелом штучного освітлення слугують люмінесцентні лампи. Віконні отвори

приміщення обладнані вертикальними жалюзьями та зовнішніми козирками. У даному приміщенні функціонує система опалення. Відповідно до санітарних норм та правил у даному приміщенні знаходиться аптечка першої медичної допомоги. Також, у даному приміщенні щоденно проводиться вологе прибирання. Згідно правил НПАОП обладнання та організація робочого місця працюючого з ПК має відповідати ергономічним вимогам, а саме:

- оптимальна робоча поза користувача ПК забезпечується конструкцією робочого місця;
- робоче місце розташоване так відносно світлових отворів, що природне світло падає збоку зліва;
- відповідно вимогам, робочий стілець є підйомно-поворотним з регульованою висотою;
- будова робочого стола відповідає вимогам ергономіки та дозволяє оптимально розміщувати на робочій поверхні ПК, допоміжне обладнання та документи.

Нормовані параметри мікроклімату, іонного складу повітря, вмісту шкідливих речовин мають відповідати вимогам Санітарних норм;

Мікрокліматичні умови приміщення повинні відповідати нормальним значенням таких показників:

- температура повітря;
- відносна вологість повітря;
- швидкість руху повітря;
- інтенсивність теплового (інфрачервоного) опромінення.

За ступенем впливу на тепловий стан людини мікрокліматичної умови поділяють на оптимальні та допустимі.

Для робочої зони приміщення встановлюються оптимальні та допустимі мікрокліматичні умови з урахуванням складності виконуваної роботи та періоду року. При виконанні роботи на ПК, що пов'язано з нервово-емоційним напруженням, у приміщенні потрібно підтримувати температура повітря  $+22^{\circ}\text{C}$  –

+24<sup>0</sup>С , відносну вологість 60-40%. Дані вимоги до параметрів мікроклімату містяться у санітарних нормах ДСН.

Оскільки розробка системи комплексної обробки подій здійснюються за ПК, необхідно проводити перерви по 15 хвилин через кожен годину. Якщо, виробничі обставини не дозволяють здійснювати часті перерви, тривалість роботи з ПК не повинна бути більшою чотирьох годин.

Відповідно до основних правил електробезпеки під час використання ПК потрібно дотримуватися таких вимог:

- при використанні ліній електромережі необхідно запобігти виникненню електричного джерела займання внаслідок короткого замикання чи перевантаження мережі;

- до електромережі ПК підключається тільки з а допомогою штепсельних з'єднань і електророзеток;

- спеціальні контакти для підключення нульового захисного провідника повинні міститися у штепсельних з'єднаннях та електророзетках. При цьому, під'єднання нульового захисного провідника відбувається швидше, ніж під'єднання фазового та нульового робочого провідника;

- підключення живлення ПК до двопровідної електромережі є недопустимим, навіть з використанням перехідних пристроїв. Тому здійснюється підключення по трьохпровідній мережі.

Згідно основних вимог до пожежної безпеки приміщення, у яких знаходиться ПК, мають бути оснащені переносними вуглекислотними або аерозольно-водопінними вогнегасниками. Підходи до засобів пожежогасіння повинні бути вільними. Також, згідно вимог НАПБ «Перелік однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації», у приміщенні де здійснюється робота з ПК, робочі місця повинні бути обладнанні системою автоматичної пожежної сигналізації з димовим пожежним сповіщувачем.

Таким чином, в даному підрозділі виконано огляд основних законодавчих актів та нормативів при роботі з ПК. Так як, дипломна робота магістра спрямована на розробку системи комплексної обробки подій та її реалізація для сфери алгоритмічної торгівлі, то було наведено основні стандарти і правила щодо влаштування робочих місць де використовують комп'ютерну техніку. Також було наведено вимоги, які стосуються приміщення де знаходиться робоче місце працівника. Крім того, розглянуті санітарні норми та правила щодо мікроклімату у даному приміщенні. Наведенні правила електробезпеки під час роботи з ПК. Також поданні основні вимоги до пожежної безпеки приміщення. При дотриманні даних правил гарантуються безпечні умови праці на ПК та запобігання шкідливих виробничих факторів.

## ВИСНОВКИ

К результати представлено завершену роботу на здобуття першого першого (бакалавра) рівню вищої освіти метою якої було розробити програмний продукт призначенням якого є створення та редагування ігрових моделей, анімацій та логіки з використанням Autodesk 3D Max, Blueprint.

Для початку, що вважаю за потрібне зазначити що перші проблеми які в мене виникали в процесі роботи, були пов'язані із недооцінкою етапу планування та відповідно огляду існуючих рішень. Оскільки тема дослідження мене цікавила дуже, я заглибився в проведення глибокого аналізу ринку у цій галузі, існуючих рішень та підходів. Звичайно результати я використав у роботі, бо, як з'ясувалося, продуктів, які вирішують дану проблему є чимало і всі вони володіють певним часто добротним функціоналом. Проте, важливим є те, що деякі із них орієнтовані суто конкретно під вирішення визначених і завдань, інші, ж навпаки, охоплюють широку область. Попри це, така велика кількість програмних продуктів створює достатньо жорстку конкуренцію, однак це мене лише підбадьорило, бо задача таки чином виявилась ще цікавішою для мене і я вважав та й вважаю, що простір для розвитку існує завжди. Також, аналізуючи таку кількість інформації я розглянув багато підходів, і зміг обрати, краще на мою думку, застосувати, і зробити достойний уваги продукт.

Також важливо розуміти, що перш ніж писати код, потрібно дуже детально і, головне, спокійно обдумати ігрову механіку, її зрозумілість для обраної цільової аудиторії, передбачити зручність та інтерес до неї. Оскільки часу на аналіз пішло багато ,тут мені довелось дещо «економити», тобто, моя порада для себе ж в майбутньому правильно планувати час і жорстко його дотримуватись як би цікаво не було.

Також було зроблено робоча версія гри так звана версія «для самого себе». По якій мені стало зрозуміло на що потрібно детальніше звернути увагу, зробити

зрозумілішою логіку, дещо відкорегувати інтерфейс на більш зручніший для сприйняття.

Також з зробив висновки про те, що ще на етапі планування гри потрібно ретельно продумувати технічні нюанси реалізації, враховуючи «можливості» недобросовісних гравців. Врахувати безперебійність роботи мого програмного продукту-гри. Для тестування я використав різноманітні програми, такі як: Fraps, MSI Afterburner, AIDA.

Отже, вже пройшовши усі етапи написання моєї випускної роботи ,я зрозумів наскільки це цікавий, креативний та міждисциплінарний процес. Не дивлячись на те що розробка зайняла у мене супер багато часу, в результаті я вважаю що отримав неоціненний досвід практичної роботи та відчутний багаж нових знань, який мені точно придасться в майбутньому.

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. М.Р. Петрик, Д.М. Михалик, О.Ю. Петрик, Г.Б. Цуприк. Методичні вказівки до виконання атестаційної роботи магістра за спеціальністю 121 – “Інженерія програмного забезпечення” для усіх форм навчання [Текст] – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя – 2020 – 27 с.

2. Вебдизайн: підруч. для студ. вищ. навч. закл. / О. В. Пасічник, В. В. Пасічник ; за заг. ред. В. В. Пасічника ; М-во освіти і науки України. — Л. : Магнолія 2006, 2010. — 519 с. : іл. — (Серія «Комп'ютинг»). — ISBN 978-966-2025-33-0

3. Плєскач В.Л. Інформаційні системи і технології на підприємствах: підручник /В.Л. Плєскач, Т.Г. Затонацька. —К. :Знання, 2011. —718с

4. Нікольський Ю.В. Дискретна математика / Ю.В.Нікольський, В.В.Пасічник, Ю.М.Щербина – Львів: Магнолія Плюс, 2007. – 608 с.

5. Інформаційні технології видобутку даних (Data mining, високопродуктивні обчислення у складних системах): навчальний посібник ІВ Бойко, МР Петрик, Г Цуприк - 2020

6. Дискретні структури (Алгебраїчні та числові системи, комбінаторний аналіз) : навчально-методичний посібник для студентів спеціальності 121 «Інженерія програмного забезпечення», аспірантів та викладачів вищих навчальних закладів / Укладач : Бойко І.В., Петрик М.Р., Цуприк Г.Б. – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2017 – 64 с.

7. Моделювання та видобуток даних (висопродуктивні обчислення у великих алгебраїчних та числових системах, комбінаторному аналізі): навчальний посібник. Тернопіль: ТНТУ 2019 – 62 с.

8. Wolf M. J. P. Encyclopedia of Video Games: The Culture, Technology, and Art of Gaming / Ed. by Mark J. P. Wolf. Santa Barbara, CA: Greenwood, 2012. 740 p.



9. Wolf, Mark J. P. The Video Game Explosion: A History from PONG to Playstation and Beyond [Архівовано 18 квітня 2016 у Wayback Machine.]. ABC-CLIO, 2008. 380 p.
10. Donovan Tristan. Replay: The History of Video Games. Yellow Ant, 2010. 516 p.
11. [Електронний ресурс] – Режим доступу: URL: <https://www.guinnessworldrecords.com/world-records/most-successful-game-engine>
12. «The Imagination Engine: Чому відеоігри нового покоління вразять ваш світ» [Електронний ресурс] – Режим доступу: URL: <https://www.wired.com/2012/05/ff-unreal4/>
13. Бйорн Страуструп . Мова програмування C++. Короткий курс - 2019. - 320 с. - ISBN 978-5-907144-12-5 .
14. Бйорн Страуструп . Програмування: принципи та практика з використанням C++. - 2016. - 1328 с. - ISBN 978-5-8459-1949-6 .
15. [Електронний ресурс] – Режим доступу: URL: [https://uk.wikipedia.org/wiki/CryEngine\\_3](https://uk.wikipedia.org/wiki/CryEngine_3)
16. [Електронний ресурс] – Режим доступу: URL: [https://uk.wikipedia.org/wiki/C\\_Sharp](https://uk.wikipedia.org/wiki/C_Sharp)
17. [Електронний ресурс] – Режим доступу: URL: <https://store.epicgames.com/en-US/>
18. [Електронний ресурс] – Режим доступу: URL: How Do Video Games Work?(англ.) Capstone. ISBN 9781625210388.
19. [Електронний ресурс] – Режим доступу: URL: <https://uk.wikipedia.org/wiki/Steam>
20. [Електронний ресурс] – Режим доступу: URL: <https://www.activision.com/>
21. [Електронний ресурс] – Режим доступу: URL: <https://www.ea.com/ea-studios/codemasters>
22. [Електронний ресурс] – Режим доступу: URL: <https://www.eidosmontreal.com/>

23. [Електронний ресурс] – Режим доступу: URL: <https://www.ubisoft.com/en-us/>

24. Платформа цифрової дистрибуції [Електронний ресурс] – Режим доступу: URL: <https://uk.wikipedia.org/wiki/Origin>

25. [Електронний ресурс] – Режим доступу: URL: <https://web.archive.org/web/20120226235109/http://www.programsfactory.univ.kiev.ua/content/books/2/53>

26. Системи штучного інтелекту: навч. посіб. / Н. Б. Шаховська, Р. М. Камінський, О. Б. Вовк. — Львів: Вид-во Львівської політехніки, 2018. — 392 с. — ISBN 966-941-197-6.

27. Системи штучного інтелекту: навч. посіб. / Ю. В. Нікольський, В. В. Пасічник, Ю. М. Щербина; за наук. ред. В. В. Пасічника; М-во освіти і науки, молоді та спорту України. — 2-ге вид. — Львів: Магнолія-2006, 2013. — 279 с.: іл.— ISBN 978-617-57-40-11-4

28. Розробка автоматизованої інформаційної системи обробки даних / Казмірчук В.М., Цуприк Г.Б./ Матеріали VIII Науково-технічної конференції «Інформаційні моделі, системи та технології» – Тернопіль 9-10 грудня 2020. — Т. : ТНТУ, 2020. — С. 148.

29. Software Requirements 3 Karl Wieggers , Joy Beatty – Microsoft Press. 1999. - 672p.

30. Agile Software Requirements: Lean Requirements Practices for Teams, Programs, and the Enterprise Dean Leffingwell , Don Widrig - Addison-Wesley Professional, 2010. – 560p.

31. More about Software Requirements: Thorny Issues and Practical Advice: Thorny Issues and Practical Advice Karl Wieggers - Microsoft Press. 2005. - 224p.

4. Software Requirements and Specifications Michael A. Jackson - Addison-Wesley Professional, 1995. – 228p.

32. Managing Software Requirements: A Use Case Approach Dean Leffingwell , Don Widrig - Addison-Wesley, 1999. – 502p.

33. Практичний курс тестування відеоігор (online + offline) [електронний ресурс] – режим доступу: url: <https://qagroup.com.ua/events/praktychnyj-kurs-testuvannia-videoigor-online-offline/>

34. М.Р. Петрик, Д.М. Михалик, О.Ю. Петрик, Г.Б. Цуприк. Методичні вказівки до виконання атестаційної роботи магістра за спеціальністю 121 – “Інженерія програмного забезпечення” для усіх форм навчання [Текст] – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя – 2020 – 27 с.

35. Запобігання виникненню надзвичайних ситуацій [Електронний ресурс] / Wikipedia – URL: [https://uk.wikipedia.org/wiki/Запобігання\\_виникненню\\_надзвичайних\\_ситуацій\\_характеру](https://uk.wikipedia.org/wiki/Запобігання_виникненню_надзвичайних_ситуацій_характеру).

36. Ураження сильнодіючими отруйними речовинами [Електронний ресурс] / URL: <http://www.wikidocs.ru/preview/44941>.

37. Дистанційний курс «Основи охорони праці» сайту дистанційного навчання ТНТУ [Електронний ресурс]. – Режим доступу: URL: <http://dl.tntu.edu.ua/index.php>

38. Охорона праці : Навчальний посібник [Текст] Геврик Є.О. - Ельга: Ніка-Центр. - 2003. – 279 с.