

РІВНЕНСЬКИЙ ДЕРЖАВНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ

Факультет математики та інформатики

(повна назва інституту/факультету)

Кафедра інформатики та прикладної математики

(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_ Батишкіна Ю. В.  
(підпис) (ініціали, прізвище)

«25» Травня \_\_\_\_\_ 2021р.

Протокол №7

**Дипломний проєкт (робота)**

ступеня «Бакалавр»  
(назва ОКР)

з напрямку підготовки (спеціальності) 113 Прикладна математика  
(код та назва напрямку підготовки або спеціальності)

на тему: Розробка програмного забезпечення для віртуальної  
подорожі по університету

Виконав(-ла): студент(-ка) 2 (2 р.н.) курсу, групи ПМ – 41  
(шифр групи)

\_\_\_\_\_ Кулик Віталій Олексійович \_\_\_\_\_  
(прізвище, ім'я, по батькові) (підпис)

Керівник \_\_\_\_\_ к.т.н., доц. Сінчук А.М. \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант \_\_\_\_\_  
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище та ініціали) (підпис)

Рецензент \_\_\_\_\_ к.т.н., доц. Бабич С.М. \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цьому дипломному проєкті  
немає запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Рівне – 2021р.

## Зміст

Вступ.....	3
Розділ 1. ОГЛЯД ІСНУЮЧИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ВІРТУАЛЬНОЇ ПОДОРОЖІ.....	5
1.1. Відомі програмні засоби створення програмного забезпечення для віртуальної подорожі.....	5
1.2. Можливості створення 3D моделей.....	16
1.3. Графічні редактори.....	27
Розділ 2. РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ ВІРТУАЛЬНОЇ ПОДОРОЖІ ПО УНІВЕРСИТЕТУ.....	33
2.1. Опис процесу створення.....	34
2.2. Візуалізація роботи.....	55
2.3. Аналіз труднощів, котрі виникали в ході роботи.....	59
Висновки.....	61
Список літератури.....	6
2.....	

використаної

## ВСТУП

Для того, щоб стати спеціалістом з «Прикладної математики», потрібно докласти чимало зусиль для вивчення не тільки математичних аспектів, а й різного роду програмних засобів, котрі сприяють вирішенню прикладних задач, зокрема, в області моделювання економічних, соціальних, екологічних та інших сферах людської діяльності. Одні з найголовніших навиків програміста є можливість створення, розробки, модифікації та подальшої підтримки програмних засобів, а також алгоритмізації та автоматизації роботи програмних додатків, тощо. Розробка різного роду програмних продуктів дає можливість автоматизувати і тим самим полегшити робочі процеси, створити нові способи та засоби пізнання світу, урізноманітнити види відпочинку та багато іншого. Вже ні для кого не секрет, що програмісти вирішують багато проблемних етапів в роботі різних галузей життя.

**Актуальність.** В сучасних реаліях життя, все більше набувають популярності віртуальні подорожі, екскурсії, візуалізація різноманітних реальних фізичних об'єктів та таке інше. Таким чином виникла ідея розробити відповідний програмний засіб для відображення моделі віртуальної подорожі по університету, зокрема з ціллю демонстрації майбутнім відвідувачам внутрішній вигляд навчального закладу. Тим самим презентувати та зацікавити даний університет потенційним вступникам. Основним завданням даної візуалізації, є представлення можливості власноруч побувати в стінах навчального закладу, задля полегшення орієнтування в ньому, у випадку безпосереднього відвідування.

Головною *метою даної роботи* є змоделювати приміщення одного з корпусів навчального закладу та розробити програмний продукт віртуальної подорожі по ньому, шляхом створення відповідного програмного додатку.

В роботі було поставлено такі завдання дослідження:

1. Проаналізувати вже відомі програмні засоби за для можливості програмування різного роду фізичних об'єктів, або ж явищ (таких як наприклад гравітація, інерція, і т.д.), а також не малу роль відіграє соціальна

доступність та можливість частково або повноцінно не комерційного використання;

2. Підібрати програмний засіб для створення 3D моделей фізичних об'єктів, з урахуванням подальшого експорту в основну програму, паралельно врахувавши функціональні можливості і варіативність вирішення тих чи інших поставлених задач;

3. Вибрати програмний засіб для написання програмних «скриптів» (коду), або ж аналогічний йому текстовий редактор.(при умові його відсутності у вибраному програмному засобі);

4. Проаналізувати та обрати відповідні графічні редактори задля створення ілюзії об'ємності при перегляді на сцені, редагування фото та відеоматеріалів.

#### **Апробація результатів дослідження.**

1. Звітна науково-практична конференція професорсько-викладацького складу Рівненського державного гуманітарного університету, яка відбулася 13-14 травня 2021 року. Рівне.
2. V Всеукраїнська науково–практична інтернет-конференція здобувачів вищої освіти та молодих науковців „ПРИКЛАДНІ АСПЕКТИ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ ТА ОБГРУНТУВАННЯ ТЕХНІЧНИХ І УПРАВЛІНСЬКИХ РІШЕНЬ”, яка відбулася 14 травня 2021 року. Рівне.

**Обсяг роботи.** Дипломна робота складається з вступу, 2-х розділів, висновків, списку використаної літератури. Повний обсяг роботи складає 62 сторінки машинного тексту, в тому числі 51 рисунок.

## Розділ 1.

# ОГЛЯД ІСНУЮЧИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ВІРТУАЛЬНОЇ ПОДОРОЖІ

В даному розділі вивчено та проаналізовано всі можливі методи та засоби програмної реалізації вище представлених завдань, враховуючи функціональні можливості в створенні аналогічних проектів. Проте виключно за рахунок їх не універсальності або вузько направленості, як в області програмного забезпечення, на якій вони можуть функціонувати, так і апаратного забезпечення, яке матиме змогу запустити дані додатки. Як наслідок обрано оптимальний підхід для отримання, в кінцевому результаті, повноцінного програмного засобу, котрий можна запустити на будь якому комп'ютері під управлінням операційної системи Windows.

### **1.1. Відомі програмні засоби створення програмного забезпечення для віртуальної подорожі**

При виборі програмних засобів, котрі можуть бути взаємозамінні, було вибрано основні критерії пошуку, а саме: доступність; зрозумілість; достатність джерел інформації.

Також варто звертати увагу на мову програмування на якій необхідно буде програмувати в даному додатку(так як вивчати досі не знайому мову, за для виконання поставленої задачі, не раціонально), а також перспективність подальших реалізацій, в ході вивчення аспектів роботи даного програмного засобу.

Іншими словами окрім вище наведених критеріїв, необхідно також звертати увагу і на сам факт того що даний засіб можна буде використовувати в майбутньому за для своїх майбутніх цілей.

Або ж скористатись вже раніше відомим програмним засобом який може в повній мірі надати повний спектр можливостей необхідних для реалізації поставлених задач.

В результаті були розглянуті наступні програмні засоби:

- Програмний засіб Unreal Engine.
- Програмний засіб Game Maker Studio.
- Програмний засіб Unity.

Розглянемо програмний засіб Unreal Engine (логотип на рис. 1.1.).



Рис. 1.1. Логотип програмного засобу Unreal Engine

Unreal Engine – міцний ігровий рушій, який використовується багатьма професійними та інді-розробниками ігор. При першому використанні інструменту схожого на Unreal Engine важко розібратися з чого саме почати [1, с.17].

Однією з основних переваг даного рушія є те, що він являється цілком безкоштовним. Користувачі мають доступ до всіх можливостей безкоштовно! Що було причиною компанії Epic Games зробити його таким? Це як безкоштовний піар хід, адже є певна ймовірність що саме на твоєму ігровому рушії буде зроблена якась хороша гра в яку будуть грати багато людей, таким самим чином все більше і більше показуючи можливості твого ігрового рушія. Проте не слід думати що вони такі собі «Меценати», котрі нічого за це не будуть мати. Коли ви випустите гру і почнете заробляти на ній гроші, то будете платити Epic Games 5% ліцензійних відрахувань від загальної суми яку надходжень. У Epic також є магазин, в якому ви можете придбати ігри інших розробників а також завантажити сам ігровий рушій за для подальшого його використання.

Даний рушій настільки простий у використанні, що за першу годину ви цілком встигнете завантажити і встановити Unreal Engine, а також створити свій перший проект і познайомитися з інтерфейсом редактора.

Для початку необхідно створити акант користувача і завантажити та встановити Epic Games Launcher. Після цього ви завантажите UE4. І почнете знайомство з даним гейм редактором[1, с.23].

Цей рушій був використаний при розробці таких ігор, як ARK: Survival Evolved, Tekken 7 та Kingdom Hearts III. Розробка Unreal Engine 4 дуже проста для початківців. За допомогою Blueprints Visual Scripting користувач може створювати готові проекти, не вводячи жодного рядка коду. У поєднанні зі зручним інтерфейсом це дозволяє швидко виготовити робочі прототипи.

Написаний на мові C++, рушій дозволяє створювати ігри для більшості операційних систем і платформ: Microsoft Windows, Linux, Mac OS і Mac OS X, консолей Xbox, Xbox 360, PlayStation 2, PlayStation Portable, PlayStation 3, Wii, Dreamcast та Nintendo GameCube.

У березні 2010 року рушій був продемонстрований на комунікаторі Palm Pre на базі мобільної платформи webOS. Для спрощення перенесення рушія використовує модульну систему залежних компонентів:

- підтримує різні системи візуалізації (Direct3D, OpenGL, Pixomatic (раніше підтримувалась Glide API, S3 Metal, PowerVR SGL),
- відтворення аудіо (EAX, OpenAL, DirectSound3D (раніше підтримувалося A3D),
- субтитри відтворювального голосу,
- розпізнавання мови (лише для PlayStation 3, Nintendo, Xbox360, Wii та Microsoft Windows, також запланованих для Mac та Linux),
- мережеві модулі та підтримка різних пристроїв введення.

Для онлайн-ігор підтримуються технології Windows Live, Xbox Live та GameSpy, що мають до 64 гравців (клієнтів) одночасно.

Тепер розглянемо програмний засіб Game Maker Studio (ЛОГОТИП на рис. 1.2.)



Рис. 1.2. Логотип програмного засобу Game Maker Studio

GameMaker Studio - один з найпопулярніших ігрових рушіїв, що дозволяє розробляти додатки під безліч платформ. GameMaker Studio є серйозним розвитком його попередника - Game Maker і головною відмінністю є додавання платформ, а також іншим істотним доопрацюванням.

Даний рушій став потужним інструментом для професійної розробки. Творець і головний розробник перших шести версій оригінального конструктора Game Maker - Марк Овермарс, наступні версії, включаючи GameMaker: Studio, розробляються компанією YoYo Games.

Безкоштовна версія (Standard) обмежена компіляцією під Windows. У порівнянні з нею, Professional версія має безліч переваг, включаючи управління ресурсами, компіляцію для macOS, Ubuntu і запуск на Android. Також, в професійній версії можна купувати окремі модулі(доповнення), що розширюють функціональність програми. Версія Master Collection містить всі поточні модулі і майбутні доповнення версії 1.x.

До особливостей даного рушія можна віднести:

1) GameMaker Studio дозволяє писати розширення під безліч платформ на відповідних їм мовами. Підтримуються наступні типи файлів розширень: gml всюди, dll-бібліотеки на Windows, Windows Phone, Xbox One, js-скрипти для HTML5, so-бібліотеки на Linux і Tizen, dylib-бібліотеки на Mac, prx на PS4, suprs на PSVita, а також спеціальні placeholder для iOS і Android.

2) можливість створити проксі-розширення для проектів на багатьох платформах, що дозволяє використовувати однакові назви функцій в коді, але звернення відбуватиметься до бібліотек відповідної платформи(будь то Linux, Windows, Android та інші).



3) вбудовані функції для роботи з кодуваннями Base64, JSON, MD5, SHA-1, можливості розпакування ZIP-архівів, читання і записи в файли з розширенням \*.ini, текстових а також двійкових файлів, управління каталогами. Є можливість взаємодіяти з мережею: колективна гра по UDP, TCP, Bluetooth, відправка http-запитів, завантаження будь-яких файлів, взаємодія з Steam API і Facebook.

З версії 1.1.1086 додана підтримка шейдерів - ефективного інструменту управління дальністю відображення графіки, додавання графічних ефектів і перетворень. GameMaker Studio підтримує вершинні і фрагменти шейдери на мовах GLSL ES, HLSL9, HLSL11 і GLSL.3 пристроїв, крім миші і клавіатури, присутні функції для взаємодії з джойстиком і геймпадами, а також для взаємодії з акселерометром смартфона, для обробки торкань і нахилу.

GameMaker Studio містить безліч математичних функцій для роботи зі скалярними і векторними величинами(що являється основою всіх геометричних обчислень в області графічного позиціонування), включаючи тригонометричні обчислення, знаходження ступенів, логарифмів, інтерполяцій, нормалей векторів, скалярних творів. Є вбудований фізичний рушій Box2D, набір функцій для роботи з ним, демонстраційні проекти Angry Cats і Angry Cats Space.

На відміну від GameMaker, Studio використовує компілятор, а не інтерпретатор коду, що прибало можливість виконання чистого коду "на льоту", але істотно збільшило продуктивність ігор. Також є додатковий модуль YoYoCompiler, що транслює GML-код в C++ і оптимізує його. Проекти, зібрані з використанням цього компонента, показують хорошу продуктивність з сотнями ігрових елементів на рівні. Однак цей модуль доступний не для всіх платформ.

Є підтримка багатьох сервісів монетизації (таких як AdMob, Google Analytics, Google Play Licensing) з коду. Також є підтримка систем управління версіями - є приклади інтеграції з SVN, GIT, Mercurial.

В останніх версіях з'явилася безліч опцій налагодження програм в новому вікні The Debug Module, що дозволяють ставити крапки зупинки,

стежити за виконанням коду, значеннями будь-яких змінних, конкретних об'єктів, вмістом буферів, кількістю і часом виконання різних «скриптів» за допомогою (GML).

GML або GameMaker Language - чудовий інструмент для розширення і без того величезного різноманіття інструментів, наданих GameMaker Studio. Сценарії GML дозволяють користувачам писати свій власний код, створюючи організовану кодову базу, яку простіше модифікувати та налагоджувати, ніж GameMaker в котрій вбудована функція перетягування[2, с. 7].

Розглянемо програмний засіб Unity (логотип на рис. 1.3.).



Рис. 1.3. Логотип програмного засобу Unity

Unity - це потужна екосистема розробки ігор для створення 2D та 3D ігор. Завдяки своїй базовій, але все ще потужній безкоштовній версії, Unity подолала бар'єри для розвитку ігрового розвитку. Хоча редактор Unity має візуальний та інтуїтивний характер, вам доведеться навчитися писати сценарії, щоб завершити свою гру. Щоб не боятися - все, що вам потрібно - це комп'ютер, підключення до Інтернету та мотивація.

Сценарії, програмування та кодування терміни - синоніми, що стосуються процесу написання комп'ютерного коду, щоб спрямувати певні положення ігрової поведінки. Хоча це може здатися трохи лякаючим, як і в будь-якій іншій справі, ви маєте на увазі почати спочатку і навчитися одній справі. Ви не можете створити гру, не написавши якийсь код, але створення коду - це лише частина процесу розробки гри. Доволі цікаво навчитися знаходити та використовувати персонажі, анімацію, спец ефекти та інші активи, як і писати код для керування їхньою взаємодією.

Unity Technologies, виробник Unity3D, забезпечує чудову документацію про підтримку, яка найчастіше включає робочий зразок коду.

Можна вивчити основи програмування в контексті використання Unity для створення простої гри, підключаючи вивчене до відповідної інформації, яка міститься в документації, і навіть до використання деякого зразкового коду [3, с. 5].

Unity став одним із найпопулярніших ігрових механізмів для розробників ігор, і Unity Technologies продовжує вносити кардинальні зміни, щоб зробити Unity більш доступним для незалежних розробників. Зараз існує більше платформ, на які можна переносити ігри Unity (це означає, що їх можна використовувати на багатьох пристроях), Asset Store доступний для централізованих ігрових ресурсів, а Unity надає постійну підтримку, яка розширилась, включаючи професійну оцінку та відгуки про вашу гру. Існує навіть цілий новий відділ, який співпрацює з розробниками та публікує ігри.

Створення ігор - це дивовижний досвід і забезпечує навіть більше прав на хвастощі, ніж високий бал чи майже неможливий постріл у голову.

Під «іграми» люди, які стоять за Unity, також усвідомлюють розширення області «серйозних ігор» - симуляції та інші захоплюючі інтерактивні враження, розроблені з використанням Unity3D для зростаючої кількості різних галузей та використання. Починаючи з досліджень Марса НАСА та віртуальних медичних навчальних середовищ CliniSpace, закінчуючи реконструкцією місця анімованих злочинів rossPlatform DeSign та віртуальними промисловими тренерами, серйозні ігри з'являються на нових майданчиках стрімко. Якщо серйозна розробка ігор є напрямком, в якому ви хочете рухатись, я вважаю, що Unity - найкращий інструмент для цього.

Найкращі ігри - це ті, які забезпечують найкращий досвід користувачів. Взаємодія з користувачем походить як від зовнішнього вигляду, так і від відчуття гри, що є ще одним способом сказати графіку та код. Графіка або ілюстрація життєво важлива з очевидної причини: це те, що бачить користувач. Це задає настрій і залучає користувача. Код - це те, що знаходиться під капотом, і не менш важливо. Найкраща графіка у світі не може компенсувати гру, яка працює повільно, реагує несподівано, не протікає в міру просування гри або просто виходить з ладу. Unity - це потужний, популярний

інструмент для розробників ігор, оскільки він дозволяє контролювати та плавно інтегрувати обидва ці важливі аспекти, щоб створити приємний досвід [3, с.1].

Unity дозволяє створювати додатки, що працюють на більш ніж 25 різних платформах, що включають персональні комп'ютери, ігрові консолі, мобільні пристрої, інтернет-додатки та інші .

Випуск Unity відбувся в 2005 році і з того часу йде постійний розвиток. Основними перевагами Unity є наявність візуальної середовища розробки, багатоплатформленна підтримки і модульної системи компонентів. До недоліків відносять появу складнощів при роботі з багатоконпонентними схемами і труднощі при підключенні зовнішніх бібліотек .

Також слід врахувати, що на Unity написані тисячі ігор, додатків, візуалізації математичних моделей, які охоплюють безліч платформ і жанрів. При цьому Unity використовується як великими розробниками, так і незалежними студіями.

До можливостей даного ігрового рушія можна віднести:

1) редактор Unity має простий Drag & Drop інтерфейс, який легко налаштовувати, що складається з різних вікон, завдяки чому можна проводити налагодження гри прямо в редакторі.

2) рушій використовує для написання «скриптів» на мові програмування C#. Раніше підтримувалися також Boo (діалект Python, підтримку прибрав в 5-й версії) і модифікація JavaScript, відома як UnityScript (підтримка припинена в версії 2017.1).

3) розрахунки фізики виробляє фізичний рушій PhysX від NVIDIA. Графічний API - DirectX (на даний момент DX 11, підтримується DX 12).

Проект на Unity ділиться на сцени (рівні) - окремі файли, що містять свої ігрові світи зі своїм набором об'єктів, сценаріїв, і налаштувань. Сцени можуть містити в собі як, власне, об'єкти (моделі), так і порожні ігрові об'єкти - об'єкти, які не мають моделі «пустушки».

Об'єкти, в свою чергу містять набори компонентів, з якими і взаємодіють «скрипти» (написані програмістом коди). Також у об'єктів є назва

(в Unity допускається наявність двох і більше об'єктів з однаковими назвами), може бути тег (мітка) і шар, на якому він повинен відображатися. Так, у будь-якого об'єкта на сцені обов'язково присутній компонент Transform - він зберігає в собі координати місця розташування, повороту і розмірів об'єкта по всіх трьох осях. У об'єктів з видимою геометрією також за замовчуванням присутній компонент Mesh Renderer, що робить модель об'єкта видимою.

До об'єктів можна застосовувати колізії (в Unity т. Н. Колайдери - collider), яких існує декілька типів. Також Unity підтримує фізику твердих тіл що є важливим аспектом при виборі саме даного програмного засобу, і тканини, а також фізику типу Ragdoll (лялька з тряпок).

У редакторі присутня система успадкування об'єктів; дочірні об'єкти будуть повторювати всі зміни позиції, повороту і масштабу батьківського об'єкта. Скрипти в редакторі прикріплюються до об'єктів у вигляді окремих компонентів.

При імпорті текстури в Unity можна згенерувати alpha-канал, mip-рівні, normal-map, light-map, карту відображень, проте безпосередньо на модель текстуру прикріпити не можна - буде створено матеріал, з яким буде призначений шейдер, і потім матеріал прикріпиться до моделі. Редактор Unity підтримує написання і редагування шейдерів.

Редактор Unity має компонент для створення анімації, але також анімацію можна створити попередньо в 3D-редакторі та імпортувати разом з моделлю, а потім розбити на файли. Unity 3D підтримує систему Level Of Detail (скор. LOD), суть якої полягає в тому, що на далекій відстані від гравця високо деталізовані моделі замінюються на менш деталізовані, і навпаки, а також систему Occlusion culling, суть якої в тому, що у об'єктів, що не потрапляють в поле зору камери не візуалізується геометрія і колізія, що знижує навантаження на центральний процесор і дозволяє оптимізувати проект.

При компіляції проекту створюється виконуваний (\*.exe) файл гри (для Windows), а в окремій папці - дані гри (включаючи всі ігрові рівні і спільні бібліотеки).

Рушій підтримує безліч популярних форматів. Моделі, звуки, текстури, матеріали, «скрипти» можна запаковувати в форматі \*.unitypackage і передавати іншим розробникам, або викладати у вільний доступ. Цей же формат використовується у внутрішньому магазині Unity Asset Store, в якому розробники можуть безкоштовно і за гроші викладати в загальний доступ різні елементи, потрібні при створенні ігор.

Щоб використовувати Unity Asset Store, необхідно мати акаунт розробника Unity. Unity має всі потрібні компоненти для створення багатокористувацької гри. UNet був видалений, починаючи з версії 2018.4; рішення «з коробки» для багатокористувацької гри відсутня. Також можна використовувати відповідний користувачеві спосіб контролю версій. Наприклад, Tortoise SVN, Git або Source Gear. У Unity входить Unity Asset Server - інструментарій для спільної розробки на базі Unity, є доповненням, що додає контроль версій і ряд інших серверних рішень.

В свою ж чергу даний рушій має також низку недоліків. Як правило, ігровий рушій надає безліч функціональних можливостей, що дозволяють їх задіяти в різних іграх, в які входять моделювання фізичних середовищ, карти нормалей, динамічні тіні і багато іншого. На відміну від багатьох ігрових рушіїв, у Unity є дві основні переваги: наявність візуальної середовища розробки та міжплатформної підтримки.

Перший фактор включає не тільки інструментарій візуального моделювання, а й інтегровану середу, лінію складання, що направлено на підвищення продуктивності розробників, зокрема, етапів створення прототипів і тестування. Під міжплатформною підтримкою надається не тільки місце розгортання (установка на персональному комп'ютері, на мобільному пристрої, консолі і т. Д.), Але і наявність інструментарію розробки (інтегроване середовище може використовуватися під Windows і Mac OS) .

Третьою ж перевагою називається модульна система компонентів Unity, за допомогою якої відбувається конструювання ігрових об'єктів, коли останні є комбіновані пакети функціональних елементів. На відміну від механізмів успадкування, об'єкти в Unity створюються за допомогою

об'єднання функціональних блоків, а не приміщення в вузли дерева успадкування. Такий підхід полегшує створення прототипів, що актуально при розробці ігор.

Як недоліки наводяться обмеження візуального редактора при роботі з багатокомпонентними схемами, коли в складних сценах візуальна робота стає складнішою.

Другим недоліком називається відсутність підтримки Unity посилань на зовнішні бібліотеки, роботу з якими програмістам доводиться налаштовувати самостійно, і це також ускладнює командну роботу. Ще один недолік пов'язаний з використанням шаблонів примірників (англ. Prefabs). З одного боку, ця концепція Unity пропонує гнучкий підхід візуального редагування об'єктів, але з іншого боку, редагування таких шаблонів є складним.

Також, WebGL-версія рушія, в силу специфіки своєї архітектури (трансляція коду з C# в C++ і далі в JavaScript), має ряд невирішених проблем з продуктивністю, споживанням пам'яті і працездатністю на мобільних пристроях.

Серед всіх перерахованих програмних засобів був обраний Unity. Так як даний засіб, позитивно виділяється на ряду з іншими за рахунок своєї простоти використанні і варіативності можливостей.

Додатковим аспектом вибору останнього було те що мова програмування яку підтримує даний засіб являється мовою C# яка мені доволі сильно імponує. Також чи малий вплив на вибір зіграв той факт що є достатня кількість інформації в різній мірі доступності за для кращого розуміння тих чи інших аспектів програмування. Іншими словами чим більше інформації буде доступною для того щоб вирішити поставлену задачу, тим простіше буде її виконати. І це доволі важливий аспект, як і той що програма відносно проста в візуальному плані і не зважаючи на те що весь інтерфейс виконаний на англійській мові, проте все ж слід зазначити що немає нічого що не можна при достатній мірі стремління вивчити. Також не мало-важливу роль у виборі відіграв фактор безкоштовності даного ігрового рушія.

## 1.2. Можливості створення 3D моделей

Критерії до вибору даного програмного засобу аналогічні як і при виборі засобу в якому будуть відбуватись всі основні маніпуляції, за виключенням лише того що слід обирати такий засіб, при роботі в якому буде змога моделі експортувати в головний проект.

Отже вибір даного засобу повинен бути обґрунтований можливостями взаємодії з іншими а також критеріями які описані і до головного рушія, а саме:

- Зрозумілість;
- Доступність;
- Достатність джерел інформації;

Також слід врахувати операційну систему на якій буде виконуватись всі маніпуляції, а саме Windows 10. Що також в певній мірі слугує певним «фільтром» при виборі програм для моделювання.

Отже в ході аналізу і врахування засобу який буде слугувати програмним рушієм мого Екскурс-додатку. Були проаналізовані такі програмні засоби:

Розглянемо програмний засіб AutoCAD (логотип на рис. 1. 4.).



Рис. 1.4. Логотип програмного засобу AutoCAD

AutoCAD з'явився більше чверті століття тому, у той час, коли більшість користувачів думали, що персональні комп'ютери не здатні виконувати такі промислові завдання, як САПР. (Абревіатура розшифровується як «Комп'ютерне проектування», «Автоматизований дизайн» або те й інше, залежно від того, з ким ви розмовляєте). Що не менш



дивовижно той факт, що через 25 років після свого створення, AutoCAD з великим відривом залишається королем пагорба мікрокомп'ютерів CAD, що робить його однією з старіших, проте досі актуальних і підтримуваних програм для ПК за всю історію. Цілком можливо, що довгострокове майбутнє САПР може належати спеціальному програмному забезпеченню на базі 3D, такому як програми Autodesk Inventor та Revit, але на сьогоднішній день та найближче майбутнє AutoCAD - це місце дії CAD.

Протягом багатьох років AutoCAD покращувався, в зв'язку з чим додавались нові функції і можливості що робило його складнішим в засвоєнні інформації, зокрема, щоб не відставати від зростаючої складності процесів проектування та складання, які призначені для обслуговування AutoCAD. Недостатньо лише просто намальовати приємні на вигляд лінії. Якщо ви хочете пограти в САПР з великими хлопцями та дівчатами, вам потрібно ретельно впорядкувати намальовані вами предмети, їх властивості та файли, в яких вони перебувають.

Адміністратору потрібно узгодити роботу з САПР з іншими користувачами, які будуть працювати над цими самими кресленнями чи використовуватимуть їх. Потрібно бути підкованим щодо доставки креслень через Інтернет. Можливо, доведеться навіть вивчити роботу з AutoCAD у трьох вимірах.

AutoCAD 2011 надає інструменти для виконання всіх цих речей, але не завжди легко зрозуміти, який молоток взяти або яким цвяхом вдарити першим. З цією книгою у вас є чудові шанси створити презентабельний, придатний для друку та друку малюнок із першої чи другої спроби, не розкладаючи T-квадрат на екрані комп'ютера[4, с.1].

AutoCAD існує вже давно - з 1982 року. AutoCAD відкрив перехід від справді дорогих системних САПР(Система автоматизованого проектування і розрахунку) на міні-комп'ютерах, що коштують десятки тисяч доларів, до лише трохи дорогих програм САПР(Система автоматизованого проектування і розрахунку) для мікрокомп'ютерів, які коштують кілька тисяч доларів.

3D-здібності AutoCAD зросли стрибкоподібно за останні кілька випусків, і 3D-моделювання стає звичним способом перевірки дизайну перед їх розробкою. Тим не менше, AutoCAD - це, в першу чергу, програма для створення двовимірних технічних креслень: креслення, в яких виміри та точність важливі, оскільки подібні креслення часто використовуються для побудови чогось. Креслення, створені вами за допомогою AutoCAD, повинні відповідати стандартам, давно встановленим для ручних креслень.

Попередні інвестиції у використання AutoCAD, безсумнівно, дорожчі, ніж інвестиції, необхідні для використання олівця та паперу, і крива навчання набагато крутіша. То навіщо турбуватись? Основними причинами використання AutoCAD, а не олівця та паперу, а :

- точність (створення ліній, кіл та інших фігур з точними розмірами простіше за допомогою AutoCAD, ніж за допомогою олівців);
- модифікація (малюнки набагато легше модифікувати на екрані комп'ютера, ніж на папері. Модифікації САПР набагато чистіші);
- ефективність (створення багатьох видів креслень відбувається швидше за допомогою програми САПР - особливо креслення, що передбачають повторення, наприклад, плани поверхів у багатоповерховій будівлі). Але для ефективності потрібні вміння та практика.

Чому вибрати AutoCAD? AutoCAD - це лише відправна точка цілої галузі програмних продуктів, призначених для роботи з AutoCAD. Autodesk надзвичайно допоміг цьому процесу, розробивши ряд програмних інтерфейсів до AutoCAD. Деякі додаткові продукти стали більш продуманими в певних аспектах, що Autodesk придбав їх та включив у свої власні продукти. Коли ви порівнюєте всі ресурси - включаючи доповнення, розширення, навчальні курси, книги тощо - AutoCAD не має великої конкуренції з САПР для ПК [4, с.12-13].

AutoCAD і спеціалізовані додатки на його основі знайшли широке застосування в машинобудуванні, будівництві, архітектурі та інших галузях промисловості. Програма випускається на 18 мовах.

Рівень локалізації варіює від повної адаптації до перекладу тільки довідкової документації. Російськомовна версія локалізована повністю, включаючи інтерфейс командного рядка і всю документацію, крім керівництва з програмування.

До можливостей даного програмного засобу можна віднести:

1) ранні версії AutoCAD оперували невеликим числом елементарних об'єктів, такими як кола, лінії, дуги і текст, з яких склалися більш складні. На цій посаді AutoCAD заслужив репутацію «електронного кульмана» (мається на увазі інструмент для малювання, креслення проте електронний), яка залишається за ним і понині. Однак на сучасному етапі можливості AutoCAD вельми широкі і набагато перевершують можливості «електронного кульмана».

2) в області двовимірного проектування AutoCAD як і раніше дозволяє використовувати елементарні графічні примітиви для отримання більш складних об'єктів. Крім того, програма надає вельми обширні можливості роботи з шарами і об'єктами призначеними для анотування в просторі моделі (розмірами, текстом, позначеннями). Використання механізму зовнішніх посилань (XRef) дозволяє розбивати креслення на складові файли, за які відповідальні різні розробники, а динамічні блоки розширюють можливості автоматизації 2D-проектування звичайним користувачем без використання програмування. Починаючи з версії 2010, в AutoCAD реалізована підтримка двовимірного параметричного креслення. У версії 2014 з'явилася можливість динамічної зв'язку креслення з реальними картографічними даними (GeoLocation API).

AutoCAD включає в себе повний набір інструментів для комплексного тривимірного моделювання (підтримується твердотільне, поверхневе і полігональне моделювання). AutoCAD дозволяє отримати високоякісну візуалізацію моделей за допомогою системи «рендеринга» mental ray. Також в програмі реалізовано управління тривимірної печаткою (результат моделювання можна відправити на 3D-принтер) і підтримка хмар точок (дозволяє працювати з результатами 3D-сканування). Проте слід зазначити,

що відсутність тривимірної параметризації не дозволяє AutoCAD безпосередньо конкурувати з машинобудівними САПР середнього класу, такими як Inventor, SolidWorks та іншими. До складу AutoCAD 2012 включена програма Inventor Fusion, яка реалізує технологію прямого моделювання.

Розглянемо програмний засіб 3DS Max (логотип на рис. 1.5.)



Рис. 1.5. Логотип програмного засобу 3DS Max

Дана програма, яка займає провідну роль серед аналогічних в даному жанрі програм, найкращий варіант для користувачів, робить перші кроки у світ можливостей 3D [5, с.18].

3Ds Max - це програмне забезпечення для 3D-моделювання, анімації і «рендеринга»(своєрідний спосіб збірки з координат точок, вершин, граней, в суцільний, 3D об'єкт), створене і розроблене для ігор і візуалізації дизайну. Програма включена в колекцію мультимедійних і розважальних програмних продуктів, пропонованих Autodesk. Воно також входить в колекцію Autodesk по архітектурі, проектування та будівництва і є одним з інструментів в галузі дизайну виробів і виробництва.

3Ds Max використовується художниками і професіоналами в області візуальних ефектів в кіно і телеіндустрії, а також розробниками і дизайнерами ігор для створення ігор віртуальної реальності. Програмне забезпечення дуже корисно для проектування будівель, інфраструктури та будівництва, а також для розробки продуктів і планування виробництва.

Крім того, 3DS Max допомагає користувачам створювати масивні ігрові світи, деталізованих персонажів, налаштовувати оточення будівлі, створювати сцени, в яких багато людей, імітувати фізичні властивості рідин, таких як вода, масло і лава. Крім того, в 3ds Max є контролери анімації, які користувачі можуть створювати, змінювати і ділитися ними.

Програмне забезпечення також має функції 3D-рендеринга, такі як можливість імітації реальних налаштувань камери. Крім того, він пропонує бібліотеку ресурсів, яка дозволяє користувачам легко шукати 3D-контент. 3Ds Max також надає функції для 3D моделювання, накладання текстур і ефектів. Завдяки цьому, користувачі зможуть створювати і анімувати геометрію різними способами, а також застосовувати моделювання поверхонь і сіток.

Опис і особливості програми.

Це багатоплатформний додаток для розробки моделей, сцен, анімацій, матеріалів і всього, що пов'язано зі світом 3D. Всі спец ефекти в кіно, презентації нових моделей техніки, одягу, авто - все це заздалегідь розроблені тривимірні моделі, які потім перетворюються в реальні об'єкти.

Професіонали вибирають 3Ds Max з кількох причин, одна з яких - підтримка всіх існуючих знімків екрану і розширень. Для майстра своєї справи найважливіше необмежений потенціал і можливості, а простота і зручність на другому місці.

Операційні системи які підтримують даний програмний засіб: Windows, MAC. Мовна підтримка: English, Dutch, Polish, Turkish, Swedish.

Види підписок: помісячна, річна.

Платні тарифні плани та пакети:

- Малий бізнес;
- Середній бізнес;
- Великий бізнес;
- Для приватних осіб.

Також містяться додаткові можливості як:

- Хмарне сховище;
- Відкритий вихідний код API.

Сьогодні важко уявити розвиток тривимірних сцен, інтерактивних ігор та реалістичних ефектів у кіноіндустрії без використання сучасних технологій комп'ютерного моделювання. Дизайнерська робота, яка повинна не лише створювати реалістичні сцени, а й звертати увагу на її художні якості, стає

незамінною. Ця книга охоплює вивчення інструменту моделювання тривимірних комп'ютерних сцен.

Графічна система 3ds Max розроблена компанією Autodesk (США) і призначена для моделювання тривимірних сцен та анімації. Перші версії цієї програми з'явилися близько двадцяти п'яти років тому, і з тих пір програма стала чи не найпопулярнішим інструментом моделювання як для початківців, так і для професійних дизайнерів.

Остання версія 3ds Max 2018, описана в посібнику, є результатом її еволюційного розвитку. 64-розрядна операційна система Windows 8 Professional x64 і новішої версії з багатоядерним процесором Intel або AMD, щонайменше 4 ГБ оперативної пам'яті, відеокартою з мінімум 512 МБ пам'яті та дисковим простором близько 6 ГБ рекомендується для встановити графічну систему 3ds Max 2018.

В програмі можна реалізовувати від плоских форм і геометричних фігур, до більш складніших за структурою моделей, для подальшого моделювання та призначення матеріалів та встановлення освітлення та реалістичного візуалізації.

Вивчення команд та інструментів можна здійснити наглядним чином всього лише виконавши певні прості базові вправи а саме:змоделювати інтер'єр кімнати, виконавши творчі завдання з моделювання тривимірних об'єктів[6, с.6].

Розглянемо програмний засіб Blender (логотип див. на рис. 1.6.)



Рис. 1.6. Логотип програмного засобу Blender

Основною перевагою даної програми над іншими аналогами являється її повна безкоштовність на ряду з великою функціональністю.

Користувачі зазвичай пов'язують безкоштовне програмне забезпечення з термінами "погано", "з обмеженими можливостями" або просто "демо

версія". Блендер повністю функціональний. Це працює як програма розвитку громад з відкритим кодом, де люди з усього світу сприяють її успіху.

Blender - це безкоштовна програма з відкритим кодом для «рендерингу» \ анімації \ ігор, що підтримується Фондом Blender, і її можна безкоштовно завантажити з веб-сайту [www.blender.org](http://www.blender.org). Ціль фонду може бути узагальнена наступним чином:

“Фонд Blender - це незалежна організація (голландська “ stichting ”), яка діє як некомерційна суспільно-корисна корпорація, з наступними цілями:

1. Заснувати сервіси для активних користувачів та розробників Blender.
2. Підтримувати та вдосконалювати поточний продукт Blender за допомогою загальнодоступної системи вихідних кодів за ліцензією GNU GPL.
3. Встановити механізми фінансування або доходу, які відповідають цілям фонду та покривають витрати фонду.
4. Надати світовому інтернет-співтовариству доступ до 3D-технологій загалом, а Blender - основним.

Blender може бути важкою для сприйняття програмою яка вчасності має в своїй основі необмежені можливості(в своїх областях).

Будь-яка програма «рендерингу» та анімації має жорстку криву навчання, і Blender не є винятком. Через кілька тижнів справа стає легшою.

Є буквально тисячі користувачів Blender у всьому світі, які переглядають форуми, щоб давати та отримувати поради. Можна в будь який момент скористатись їх порадами або у випадку якщо знаєте вирішення проблеми інших людей то можете також і допомагати тим хто цього потребує [7, с.7].

До можливостей даного програмного засобу можна віднести: характерну особливість пакету Blender є його невеликий розмір в порівнянні з іншими популярними пакетами для 3D-моделювання. Документація в поставку не входить, але доступна онлайн.

Демонстраційні сцени можна скачати на офіційному сайті [15] або на сайті відкритих проектів «Blender Cloud». Функції пакета:

- Підтримка різноманітних геометричних примітивів, включаючи полігональні моделі, систему швидкого моделювання в режимі subdivision surface (SubSurf), криві Безьє, поверхні NURBS, metaballs (метасфери), скульптурне моделювання та векторні шрифти.
- Універсальні вбудовані механізми «рендерингу» і інтеграція з зовнішніми «рендерер» YafRay, LuxRender і багатьма іншими.
- Інструменти анімації, серед яких інверсна кінематика, скелетна анімація і сіткова деформація, ключові кадри, нелінійна анімація, редагування вагових коефіцієнтів вершин, обмежувачі.
- Динаміка м'яких тіл (включаючи визначення колізій об'єктів при взаємодії), динаміка твердих тіл на основі фізичного рушія Bullet.
- Система частинок включає в себе систему волосся на основі частинок.
- Модифікатори для застосування неруйнівних ефектів.
- Мова програмування Python використовується як засіб визначення інтерфейсу, створення інструментів і прототипів, системи логіки в іграх, як засіб імпорту / експорту файлів (наприклад, COLLADA), автоматизації завдань.
- Базові функції нелінійного відео та аудіо монтажу.
- Композитинг відео, робота з хромокею(одно кольоровий задній фон, на котрому можна розмістити будь які підложки).
- Трекінг(відслідкування) камери і об'єктів.
- Real-time контроль під час фізичної симуляції і «рендерингу».
- Процедурне і node-based текстурування, а також можливість малювати текстуру прямо на моделі.
- Grease Pencil - інструмент для 2D-анімації в повному 3D-Пайплайн.
- Blender Game Engine - підпроект Blender, що надає інтерактивні функції, такі як визначення колізій, рушій динаміки і програмована логіка. Також він дозволяє створювати окремі real-time-додатки починаючи від архітектурної візуалізації до відео ігор. Вилучений у версії 2.8.



Додатково слід зазначити, що:

- У програмі Blender сутність, що взаємодіє з навколишнім світом, і її дані (форма або функції об'єкта) розділяємості. Ставлення Об'єкт-Дані представляється відношенням 1: n (термін, що відноситься до теорії баз даних, позначає можливість декількох об'єктів використовувати одні і ті ж дані - один до багатьох або сюр'єкція).
- Внутрішня файлова система, що дозволяє зберігати кілька сцен в єдиному файлі (званому .blend-файл).
- Всі «.blend»-файли сумісні як із старішими, так і з більш новими версіями Blender. Також всі вони переносяться з однієї платформи на іншу і можуть використовуватися як засіб перенесення створених раніше робіт.
- Blender робить резервні копії проектів під час всієї роботи програми, що дозволяє зберегти дані при непередбачених обставинах.
- Всі сцени, об'єкти, матеріали, текстури, звуки, зображення, post-production-ефекти можуть бути збережені в єдиний «.blend»-файл.
- Налаштування робочого середовища можуть бути збережені в «.blend»-файл, завдяки чому при завантаженні файлу користувач отримає саме те, що зберіг в нього. Ви можете зберегти файл як «власні за замовчуванням», і кожен раз при запуску Blender буде видаватися необхідний набір об'єктів та підготовлений до роботи інтерфейс.

До методів і потреб застосування можна віднести:

- 1) дана програма необхідні для 3D модельєрів - вони створюють 3D моделі персонажів, будівель, техніки тварин для ігор, і рідше в кіноіндустрії;
- 2) це необхідно для професії візуалізатор. Візуалізація інтер'єрів приміщень, екстер'єрів, виставкових стендів;
- 3) це дизайнери. Для створення зовнішньої реклами, друкованої продукції, а так само дизайну сайтів. Найчастіше набагато простіше і швидше змодельювати об'єкт в потрібному ракурсі, ніж шукати його і підбирати ракурс або малювати, і виходить набагато реалістичніше, так як при візуалізації об'єкта враховується фізичні особливості об'єктів.

А якщо необхідно включити в дизайн об'єкти, які не існують в реальному світі, то їх можна тільки змоделювати або намалювати. Так як програма є векторної, у вас вийдуть дуже якісні картинки в результаті. Навіть на банерах за якістю вони будуть вигравати в порівнянні з фотографіями.

Також Blender буде корисний для Анімації і спец ефектів - використовується в кіноіндустрії для створення повнометражних і короткометражних мультфільмах, в рекламній продукції (реклама по телебаченню), для ефектної презентації, наприклад, при будівництві житлового комплексу або майбутнього ремонту в квартирі. А також при використанні в пост-обробці, створення різних спец ефектів у фільмах і їх монтаж, поєднання 3d і відеоряду, і анімації персонажів у відеоіграх.

Створення інтерактивних ігор використовуючи рушій Блендер. Але найчастіше використовують сторонній ігровий рушій.

Можливості пакету 3D графіки і чим він примітний:

1) він є вільним рушієм 3D моделювання - це означає, що ви можете його використовувати як в особистих, так і в комерційних цілях. А так же у нього є відкритий вихідний код, що дозволяє при навичках програмування змінювати цю програму на свій розсуд.

2) обсяг дистрибутива він становить близько 50 мегабайт.

3) в нього вже інтегровано 6 рушіїв «рендерингу», включено за замовчуванням 2 Cycles Render і Blender Render.

4) можливість підключати сторонні як платні так і безкоштовні рушії «рендерингу» такі як V-Ray.

5) можливість працювати зі скульптурному і відчутти себе професійним скульптором. На озброєнні в blender є велика кількість інструментів.

6) при збереженні проекту можна все текстури і ресурси зберегти в єдиному файлі і вони ніколи не загубляться як це буває в інших 3d редакторах.

7) він має безліч мов як інтерфейсу так і підказок. Так що проблем з мовою виникнути не повинно.

Серед програмних засобів необхідних для створення 3D моделей був вибраний програмний засіб Blender. Слід зауважити що причини дуже схожі

що і в Unity, а саме те що я раніше вже користувався даним програмним засобом, а також фактор безкоштовності. Які в свою чергу відіграли чималу роль.

Проте не слід також відкидати і того аспекту що як і в Unity в Blender-і також є достатньо інформації для вирішення поставлених задач або пошуку рішень на проблеми які можуть виникати.

### 1.3. Графічні редактори

В ході аналізу необхідності виконання тих чи інших задач, було прийнято рішення також звернути увагу і на програми які будуть допоміжними засобами при досягненні поставлених задач. Іншими словами самі по собі програми будуть необхідні лише для візуального покращення відображення, та сприймання інформації користувачем, але в своїй суті не є необхідними.

Оскільки новостворена програма буде містити фото матеріали реальних 3D об'єктів, було б доцільно заручитись допомогою відповідних фото редакторів, а також оскільки дані фото можуть, і будуть накладені на текстури приміщень, тому також доцільно подумати про комфортність та якість накладання останніх.

В зв'язку з чим є необхідність в програмах які зроблять комфортним накладанням текстур, на 3D об'єкти а також стіни.

Розглянемо програмний засіб Crazybump (логотип на рис. 1.7.)



Рис. 1.7. Логотип програмного засобу Crazybump

CrazyBump - повнофункціональний редактор карт і 3D-графіки зі спрощеної робочим середовищем. Необхідно для початку роботи скачати CrazyBump і встановити всього один лаунчер на комп'ютер. Додаток повністю сумісний з усіма версіями ОС Віндовс і готове до роботи без рестарту системи. Основні можливості утиліти полягають не тільки в створенні поверхні текстури і накладення векторів, а й в редагуванні вже готових карт. Робоче середовище розрахована на недосвідченого і професійного користувача; інструменти можна додавати, прибирати і міняти місцями для зручності.

Перед тим як додаток завантажити безкоштовно 32 і 64 біт, необхідно знати розрядність. Запуск здійснюється без прав доступу адміністратора. Робоче середовище повністю російською мовою, що істотно спрощує ознайомлення. Працює редактор з трьома основними режиму графіки; це накладення ефектів нормалей (сюди відносять нерівності на поверхні), рельєф і відблиски або дзеркальне відображення. Кожен ефект детально опрацьований і має свої параметри регулювання для Windows. Кожен крок роботи відображається у вкладці «Історія». Це дозволяє скасувати одне або кілька останніх дій. Задати точне число запам'ятовування кроків є в спеціальному розділі налаштувань.

Працює редактор з найбільш популярними розширеннями: OBJ, ASK, X і ASE. На комп'ютер не потрібно установки додаткового програмного забезпечення, що спрощує роботу. Є можливість попереднього перегляду обраних ефектів. Навантаження на процесор і графічну карту розподіляються по особливому алгоритму роботи, що дозволяє без затримки спостерігати поточні зміни картинки. Деталізація рельєфу вимагає не тільки знань, але ще і хорошою інструментальної бази, яка є в робочому середовищі.

- До головних достоїнств утиліти відносяться:
- зручний інтерфейс;
- повністю настроюється середовище;
- російська мова;
- швидка обробка змін;
- режим попереднього перегляду накладень.

Розглянемо програмний засіб GIMP (логотип на рис. 1.8.)



Рис. 1.8. Логотип програмного засобу GIMP

GIMP (або The GIMP) - пакет для створення і редагування растрових зображень (растровий графічний редактор), що розробляється спільнотою розробників за технологією розробки з відкритим вихідним кодом (Open Source) і розповсюджується на умовах вільної ліцензії GNU (GNU Public License - GPL). GIMP прекрасно підходить для аматорської та напівпрофесійної роботи з зображеннями: обробки фотографій, створення графічних композицій і колажів, створення елементів дизайну web-сторінок. Можливості GIMP дозволяють обійтися без дорогих комерційних пакетів растрової графіки або їх піратських версій, що особливо актуально в зв'язку з посиленням захисту авторських прав.

Проект GIMP був народжений в 1995 році в університеті Берклі (Каліфорнія). Завдяки відкритій системі розробки та актуальності самого завдання створення і розвитку потужного відкритого пакету растрової графіки проект дуже динамічно розвивався і розвивається до сих пір. Спочатку для графічного інтерфейсу пакета використовувалася графічна бібліотека Motif, але потім була розроблена нова бібліотека - GTK (GIMP ToolKit), яка в подальшому стала однією з основних вільно розповсюджуваних бібліотек для графічних інтерфейсів (зараз використовується GTK2+).

Найважливішими відмітними особливостями GIMP є:

- по-перше, вільна модель розробки і поширення (тому проект розвивався більше 10 років і буде розвиватися і надалі, і новітні версії загальнодоступні),
- по-друге, багатоплатформність (GIMP реалізований для всіх варіантів Linux, всіх гілок BSD-систем, для MacOS і для MS Windows),

- по-третє, гнучкість і розширюваність (мається як вбудовану мову для створення розширень, так і можливості розвивати функціональність пакета на мові Python)[8, с.7-8].

Розглянемо програмний засіб Adobe Photoshop (логотип на рис. 1.9.)



Рис. 1.9. Логотип програмного засобу Adobe Photoshop

Adobe Photoshop CS6, еталон якості цифрових зображень, забезпечує високу продуктивність, потужні можливості редагування графіки і має інтуїтивно зрозумілий інтерфейс. Інструментарій Adobe Camera Raw, пропонований разом з додатком Photoshop CS6, забезпечує високу ступінь гнучкості і контролю при обробці RAW-зображень. До того ж тепер ви можете застосовувати ті ж методи до форматів TIFF \*.nJPEG. Програма Photoshop CS6 розширює межі можливостей редагування цифрових зображень і допомагає втілити в життя ваші дизайнерські фантазії [9, с.12].

Для роботи поліграфістів, дизайнерів, фотографів, мультиплікаторів, архітекторів та інших творчих професій важливо мати повний набір інструментів для корекції мультимедійних файлів. Adobe Photoshop - незвичайний редактор, в якому можна не тільки коригувати і ретушувати зображення, а й перетворювати форми, зберігати GIF-анімації, використовувати шари і ефекти. Ці функції дозволяють легко трансформувати, клонувати, покращувати зображення, застосовувати різні фільтри для створення живописної, що продає картинки.

#### Основні можливості Adobe Photoshop

Ліцензована версія Adobe Photoshop оснащена великою кількістю функціональних можливостей для отримання високоякісних знімків:

- поліпшення якості відсканованих, віддрукованих знімків, усунення дефектів кольору під час зйомки;
- створення багатошарового зображення з можливістю редагування кожного окремо для передачі кольору, роботи зі складною графікою;
- монтування фотографій і картинок, окремих елементів в колажі;
- відновлення старих знімків, ретуш фотографій;
- робота з ескізами і кресленнями;
- великий вибір інструментів для обробки текстових елементів (шрифти, ефекти, фільтри);
- обробка 2D і 3D моделей, додавання до них текстур, робота з формами;
- створення цікавих графічних елементів для поліграфії, дизайну, оформлення сайтів і так далі;
- підготовка до друку на принтері або в інтернеті і миттєва публікація файлів;
- робота зі стандартними форматами для растрових і векторних зображень;
- розфарбовування чорно-білих файлів.

#### Переваги Adobe Photoshop для якісної ретуші

Adobe Photoshop - основний інструмент комп'ютерної графіки, який використовується для всіх видів комерційної обробки медіа файлів. Суть Photoshop в тому, щоб скорегувати зображення, поліпшити його сприйняття, надати фото і відео чіткості і гармонійності. Ця програма дозволяє підсилити ефект від картинки, створити складні і красиві конструкції з простих елементів.

#### Чому варто використовувати Adobe Photoshop:

- Підтримує всі основні формати медіа-файлів і інструментарії для їх редагування.
- Має повноцінний функціонал для корекції і створення картинок.
- Підтримує повноцінну роботу графічних планшетів.
- Може розширюватися за рахунок плагінів.

- Існує безкоштовна версія, але для професійного використання рекомендуємо купити ліцензований ключ Adobe Photoshop.

Серед графічних редакторів які можуть знадобитись в ході виконання завдання були вибрані всі перераховані редактори. Так як відповідають вимогам, а також являються простими у використанні з великою низкою можливостей.



## Розділ 2.

### РОЗРОБКА ПРОГРАМНОГО ЗАСОБУ ВІРТУАЛЬНОЇ ПОДОРОЖІ ПО УНІВЕРСИТЕТУ

В ході розробки екскурс-ролику було витрачено певні ресурси, а також знадобились деякі матеріали для реалізації а саме:

- Відео матеріали (зняті відео ролики приміщень, конкретних об'єктів обстановок).
- Фото матеріали (з фотографовані об'єкти обстановки, приміщень, фасаду, постерів, рекламних вивісок, інформаційних матеріалів, тощо).
- Комп'ютерний ресурс (зокрема обчислювальні потужності які можуть або сповільнювати хід розробки додатку або обробки допоміжних матеріалів, так і пришвидшувати завдяки достатньо швидкій обробці інформації).
- Ресурс часу (обмеження в часі на виконання поставлених задач, що в якійсь мірі дає змогу чомусь приділяти більше часу а чомусь приходиться виділяти менше).
- Фінансовий ресурс (багато матеріалів технічного спрямування можна знайти в довільному доступі, проте також слід пам'ятати що для кращого виконання поставлених задач якісь матеріали може бути необхідно придбати, будь то інформаційні(літературу), технічні(гаджети, обладнання), програмні, та інші, за для легшого, комфортнішого, простішого виконання тих чи інших задач).
- Саме середовище розробки(іншими словами програмний засіб за допомогою якого буде реалізована поставлена задача).
- Кастомні шрифти, які могли б знадобитись для більш індивідуалізації роботи в області текстового відображення інформації.
- Різного роду вбудовані в середовище шаблони, які вже присутні в програмному засобі і котрі можна використати в проекті для створення більш приємної картини відображення.

- Теоретична інформація, яка буде використана в ході демонстрацій відео-екскурсійного матеріалу.
- Різноманітні завантаження з вбудованого в середовище магазину безкоштовних 3D моделей або спрайтів або іншого роду матеріалів, котрі можуть знадобитись в ході виконання програми.

## 2.1. Опис процесу створення

Визначившись з середовищем розробки Екскурс-додатку, котрим буде Unity, а також програмним засобом котрий необхідний для створення та подальшого експортування 3D моделей приміщення, яким буде слугувати Blender, а також допоміжними додатками для покращення матеріальної бази, можна приступити до самого процесу створення який я розбив на такі етапи:

1. Збір, та структурування необхідних для створення 3D моделей а також самого додатку, матеріалів:

- 1.1. Фото матеріали плакатів, об'єктів на стінах, на підлозі, предметів обстановки, дверей, вивісок(Рис. 2.1.).

Рис. 2.1. Фото матеріали необхідні для одного поверху

- 1.2. Створення планограми коридорів, приміщень, з детальними відмітками розміщень тих чи інших деталей на них(Рис. 2.2.).



Рис. 2.2. Планограма першого поверху

2. Редагування, покращення, обрізання отриманих фото матеріалів і доведення їх до канонічного вигляду за допомогою програмного засобу Adobe Photoshop(Рис.2.3., Рис.2.4.).

Рис. 2.3. Початковий етап обробки фото матеріалу

Рис. 2.4. Кінцевий етап обробки фото матеріалу

3. Завантаження фотографій які будуть слугувати в ролі фонів стін, стелі, підлоги в програму GIMP, котра і зможе зробити з простих фотографій «Безшовні» текстури, з котрими в подальшому і будемо мати справу(Рис.2.5., Рис.2.6.)

Рис. 2.5. Початковий вигляд «Безшовної текстури»

Рис. 2.6. Кінцевий вигляд «Безшовної текстури»

4. За допомогою програми Crazybump створюємо «Карти нормалій» для вже оброблених в вище зазначених програмних засобах фотографій. Це дасть змогу в подальшому зробити ілюзію об'ємності і допоможе прорахувати траєкторію проходження сонячних променів від джерел світла(Рис.2.7., Рис.2.8.).

Рис. 2.7. Вікно вибору типу об'ємності

Рис. 2.8. Вікно детального налаштування «Карти нормалі»

5. Спираючись на наявні фото матеріали, відзняті відео матеріали(котрі слугують орієнтиром в обстановці приміщення), а також планограми коридорних приміщень, створюємо максимально детальну модель коридорних просторів першого, другого та третього поверхів за допомогою програмного засобу Blender (зберігаючи пропорції, відстані і розміщення відносно одне одного різних об'єктів, як і на фото матеріалах) (Рис.2.9., Рис.2.10., Рис.2.11.).

Рис. 2.9. Вікно вигляду 3D моделі першого поверху

Рис. 2.10. Вікно вигляду 3D моделі третього поверху

Рис. 2.11. Вікно вигляду 3D моделі другого поверху

6. Маючи новостворені 3D моделі приміщень і планограми розміщення об'єктів в цих приміщеннях а також матеріали які стосуються стін, стелі, підлоги, накладаємо раніше створені текстури на дані моделі(Рис.2.12., Рис.2.13., Рис.2.14.).

Рис. 2.12. Вікно вигляду 3D моделі другого поверху після накладання текстур

Рис. 2.13. Вікно вигляду 3D моделі третього поверху після накладання текстур

Рис. 2.14. Вікно вигляду 3D моделі першого поверху після накладання текстур

7. За допомогою «Карт нормалій» які були створені раніше, додаємо ілюзії об'ємності до накладених текстур(Рис.2.15.).

Рис. 2.15. Вікно вигляду налаштування «нодів» до матеріалів

8. Зберігаємо створену модель (в розширені яке необхідне для комфортного імпорту в середовище розробки яким в даному випадку виступає Unity) з накладеною на неї текстурами відносно кожного з поверхів та згідно з планограми та фото матеріалами які були отримані раніше, за для досягнення максимальної ідентичності в імітації відображення(Рис.2.16.).

Рис. 2.16. Вікно вибору налаштувань імпорту 3D моделі

9. Імпортуємо ново створену 3D модель, з накладеними на неї текстурами стін, стелі, та підлоги в середовище розробки Unity для проведення подальших дій що до її коректного відображення(Рис.2.17., Рис.2.18., Рис.2.19.).



Рис. 2.17. Відображення матеріалів імпорту 3D моделі першого поверху



Рис. 2.18. Відображення матеріалів імпорту 3D моделі другого поверху



Рис. 2.19. Відображення матеріалів імпорту 3D моделі третього поверху

10. Генеруємо колайдери до імпортованої текстури, за для того щоб вона мала об'ємність, і з нею можна було взаємодіяти безпосередньо як з «об'єктом» (рис.2.20).

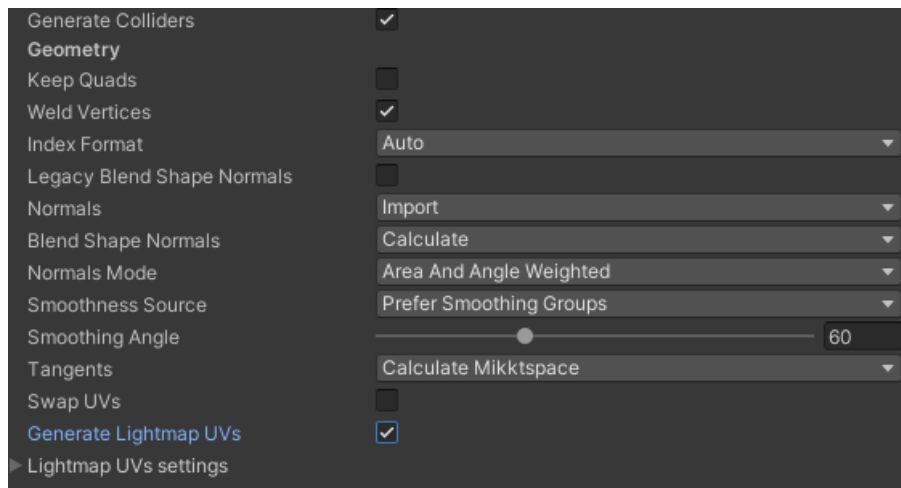


Рис. 2.20. Вікно вибору властивостей 3D моделі

11. Налаштовуємо джерела освітлення, щоб найбільш правдоподібно виглядало до того як є в дійсності(Рис.2.21.).

Рис. 2.21. Відображення джерел освітлення

12. Завантажуємо, імпортуємо, модель «користувача» яким в подальшому і будемо керувати(Рис.2.22.).

Рис. 2.22. Вигляд моделі «користувача»

13. Налаштовуємо відображення камери, відповідно до доданої моделі користувача (яка буде слугувати певним видом відображення в режимі «Від першого лиця») (Рис.2.23.).

Рис. 2.23. Налаштування вигляду відображення «Від першого лиця»

14. За допомогою вбудованих функцій створюємо «скрипт» для пересування користувача а також повороти голови за допомогою мишки.

using UnityEngine;

[AddComponentMenu("Camera-Control/Camera")]

public class Camera : MonoBehaviour{

    public GameObject player;

    private Vector3 offset;

    public enum RotationAxes { MouseXandY = 0, MouseX = 1, MouseY = 2 }; //Випадаюче меню для налаштування осей обертання

    public RotationAxes axes = RotationAxes.MouseXandY; //отримуємо осі з даного елемента

    //Змінні чутливості мишки по осі X та по осі Y

    public float sensivityX = 2f;

    public float sensivityY = 2f;

    //зміні призначені для обмежень по максимальним кутам повороту для осі X

    public float minimumX = -360f;

    public float maximumX = 360f;

    //зміні призначені для обмежень по максимальним кутам повороту для осі Y

    public float minimumY = -40f;

    public float maximumY = 90f;

    //Змінні для визначення нинішнього кута повороту

    public float rotationX = 0f;

    public float rotationY = 0f;

    //Змінна яка містить Основний тип повороту (Quaternion)

```

Quaternion originalRotation;

void Start(){
    offset = transform.position - player.transform.position;//дізнаємося
який відступ від об'єкта до камери
    //заборона обертання нашого твердого тіла якщо камера не буде
рухатись
    if (GetComponent<Rigidbody>()){
        GetComponent<Rigidbody>().freezeRotation = true; }
    originalRotation = transform.localRotation; }

public static float ClampAngle(float angle, float min, float max) {
    //умови дає змогу крутитись безкінечно навколо своєї осі
    if (angle < -360f)
        angle += 360f;
    if (angle > 360f)
        angle -= 360f;
    return Mathf.Clamp(angle, min, max);//функція обраховує мінімум
максимум і саме значення англ(кута і поверне його для подальшого
використання) }

void Update(){
    if (axes == RotationAxes.MouseXandY)//Якщо рухати мишкою по
двом осям X і Y {
        rotationX += Input.GetAxis("Mouse X") * sensivityX;
        rotationY += Input.GetAxis("Mouse y") * sensivityY;
        rotationX = ClampAngle(rotationX, minimumX, maximumX);//по
цій функції буде обраховуватись обертання по осі X
        rotationY = ClampAngle(rotationY, minimumY, maximumY);//по
цій функції буде обраховуватись обертання по осі Y
        Quaternion xQuaternion = Quaternion.AngleAxis(rotationX,
Vector3.up);
        Quaternion yQuaternion = Quaternion.AngleAxis(rotationY, -
Vector3.right);

```



```

        transform.localRotation = originalRotation * xQuaternion *
yQuaternion; }
    else
        if (axes == RotationAxes.MouseX)//Якщо рухати мишкою по осі X
        { rotationX += Input.GetAxis("Mouse X") * sensivityX;
          rotationX = ClampAngle(rotationX, minimumX, maximumX);
          Quaternion xQuaternion = Quaternion.AngleAxis(rotationX,
Vector3.up);
          transform.localRotation = originalRotation * xQuaternion; }
    else
        if (axes == RotationAxes.MouseY)//Якщо рухати мишкою по осі Y
        { rotationY += Input.GetAxis("Mouse Y") * sensivityY;
          rotationY = ClampAngle(rotationY, minimumY, maximumY);
          Quaternion yQuaternion = Quaternion.AngleAxis(rotationY, -
Vector3.right);
          transform.localRotation = originalRotation * yQuaternion; }}
    void LateUpdate()//функція визивається 1 раз в кінці кадру
    { transform.position = player.transform.position + offset; }}

```

15. Маючи «скрипт» пересування, налаштуємо анімацію гравця, яка буде відповідати тим чи іншим діям зі сторони користувача (будь то рух вперед, зупинка) (Рис.2.24.).

Рис. 2.24. Налаштування анімації «користувача»

16. Налаштуємо розташування 3D моделі гравця, а також його пропорції вже в середині імпортованої моделі приміщення (щоб відповідати висоті дверей, дверних проїомів, стелажів, тощо) (Рис.2.25.).

Рис. 2.25. Налаштування позиції та розмірів «користувача»

17. Перенесення всіх фото матеріалів які стосуються плакатів, дверей, вивісок, тощо, в «робочу зону» Unity для проведення подальших маніпуляцій над ними(Рис.2.26.).

Рис. 2.26. Вікно відображення матеріалів «робочої зони»

18. Перетворення звичайних зображень які були перенесені в робочу зону, в новий тип відображення такий як «Спрайт» (Рис.2.27.).

Рис. 2.27. Вікно налаштування властивостей зображень

19. Керуючись планограмою приміщення перенесення всіх «Спрайт» відповідно до їх поточного місця розташування в реальних приміщеннях(Рис.2.28.).

Рис. 2.28. Вікно вигляду першого поверху після розміщення всіх необхідних «Спрайтів»

20. За допомогою вбудованих функцій створюємо скрипт для «переміщення між поверхами».

```
using UnityEngine;
using UnityEngine.SceneManagement;//не забути додати цю бібліотеку щоб працював сцен менеджер
using UnityEngine.UI;
public class LoadLevel : MonoBehaviour{
    public string playerTag;
    public static bool Organization { get; set; }//left=true right=false
    private Text TextErrorr;
    void Start(){
        TextErrorr = GameObject.Find("TextErrorr").GetComponent<Text>();
        TextErrorr.enabled = false; }
    void Update(){ }
    void OnTriggerEnter(Collider other) {
```

```

if (other.tag == "Player"){
    if (gameObject.name == "Triger_Corpus_1_ErrorrLeft" ||
gameObject.name == "Triger_Corpus_1_ErrorrRight"){
        TextErorr.enabled = true;
        Debug.Log($"Tuda Nezia {gameObject.name}");
    }
    if (gameObject.name == "Triger_Corpus_1_2Right"){
        Organization = true;
        SceneManager.LoadScene(2); }
    if (gameObject.name == "Triger_Corpus_1_2Left"){
        Organization = false;
        SceneManager.LoadScene(2); }
    if (gameObject.name == "Triger_Corpus_1_3Right"){
        Organization = true;
        SceneManager.LoadScene(3); }
    if (gameObject.name == "Triger_Corpus_1_3Left"){
        Organization = false;
        SceneManager.LoadScene(3); }
    if (gameObject.name == "Triger_Corpus_1_1Right"){
        Organization = true;
        SceneManager.LoadScene(1); }
    if (gameObject.name == "Triger_Corpus_1_1Left"){
        Organization = false;
        SceneManager.LoadScene(1); }}}
void OnTriggerExit(Collider other2) {
    TextErorr.enabled = false; }}

```

21. Додаємо об'єкти які будуть слугувати точками переходів (своєрідні тригери) між поверхами, та прикріплюємо новостворений «скрипт» «переміщення між поверхами» на них(Рис.2.29.).

Рис. 2.29. Вигляд тригерів переходів між поверхами

22. Додаємо нову сцену, котра буде слугувати меню вибору різних параметрів(Рис.2.30.).

Рис. 2.30. Вигляд головного меню

23. Прописуємо «скрипти» які будуть використовуватись при натисненні на ту чи іншу кнопку меню, та переходів між сценами (у випадку натиснення на кнопку переходу на нову сцену).

```
using UnityEngine;
using UnityEngine.SceneManagement;
public class Button : MonoBehaviour{
    public GameObject buttonStart;
    public GameObject buttonsMenu;
    public GameObject buttonsExit;
    public GameObject buttonsFaq;
    public GameObject buttonsSettings;
    public GameObject buttonsBack;
    public GameObject buttonsTheEnd;
    public GameObject Corpus_1_1;
    public GameObject Corpus_1_2;
    public GameObject Corpus_1_3;
    public void ShowStartMenu(){
        buttonsMenu.SetActive(false);
        buttonsExit.SetActive(false);
        buttonsFaq.SetActive(false);
        buttonsSettings.SetActive(false);
        buttonStart.SetActive(true); }
    public void ShowExitButtons()//Кнопка "Вихід" робить видимими 2
кнопки і текст а не видимим все інше
    { buttonsFaq.SetActive(false);
        buttonsSettings.SetActive(false);
        buttonsMenu.SetActive(false);
```

```

        buttonStart.SetActive(false);
        buttonsExit.SetActive(true); }

public void ShowFaq()//Кнопка "Про програму" робить невидимим всі
вікна окрім налаштувань.
    { buttonsSettings.SetActive(false);
      buttonsMenu.SetActive(false);
      buttonsExit.SetActive(false);
      buttonStart.SetActive(false);
      buttonsFaq.SetActive(true); }

public void ShowSettings()//Кнопка "налаштування", робить
видимими всі настройки і не видимими всі інші
    { buttonsMenu.SetActive(false);
      buttonsExit.SetActive(false);
      buttonsFaq.SetActive(false);
      buttonStart.SetActive(false);
      buttonsSettings.SetActive(true); }

public void ShowMainMenu()//Кнопка повернення на головний екран,
робить не видимими всі параметри окрім мейн
    { buttonsExit.SetActive(false);
      buttonsFaq.SetActive(false);
      buttonsSettings.SetActive(false);
      buttonStart.SetActive(false);
      buttonsMenu.SetActive(true); }

public void ClosedTheProgram()
    { buttonsMenu.SetActive(false);
      buttonsExit.SetActive(false);
      buttonsFaq.SetActive(false);
      buttonsSettings.SetActive(false);
      buttonStart.SetActive(false);
      Application.Quit();}

```

```

public int number_0_0_scene, number_1_1_scene, number_1_2_scene,
number_1_3_scene;

public void Show_Corpus_0_0(){
    SceneManager.LoadScene(number_0_0_scene); }

public void Show_Corpus_1_1(){
    SceneManager.LoadScene(number_1_1_scene); }

public void Show_Corpus_1_2(){
    SceneManager.LoadScene(number_1_2_scene); }

public void Show_Corpus_1_3(){
    SceneManager.LoadScene(number_1_3_scene); }

void Update(){
    if (Input.GetKey(KeyCode.Escape)) {
        SceneManager.LoadScene(number_0_0_scene); } }

```

24. Кожен поверх університету, виконуємо як нову сцену, з відповідним додаванням на неї матеріалів які будуть відноситись до конкретно даної сцени(Рис.2.31.).

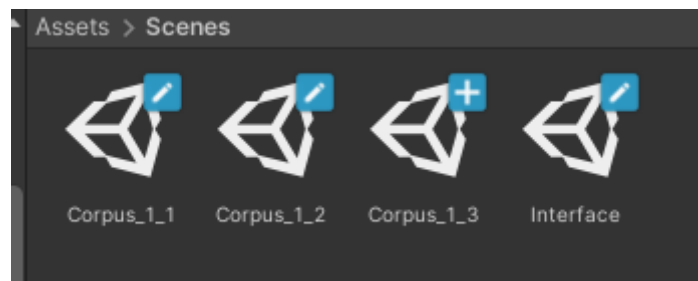
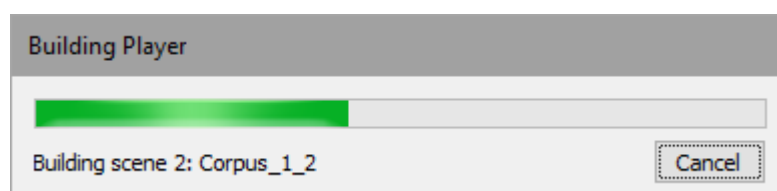


Рис. 2.31. Вікно відображення створених сцен

25. Перетягуємо всі активні сцени в вікно компіляції програми Unity в порядку, в якому буде відбуватись запуск програми(Рис.2.32.).

Рис. 2.32. Вікно налаштування способу компіляції

26. Компілюємо програму для запуску на інших комп'ютерах під управлінням операційної системи Windows(Рис.2.33.).



### Рис. 2.33. Вигляд вікна компіляції проекту

В результаті отримаємо папку в котрій містяться всі необхідні для запуску файли(Рис.2.34.).

### Рис. 2.34. Вигляд файлів в папці проекту

## 2.2. Візуалізація роботи

При запуску програми, нас зустрічає «Головне меню» з можливостями вибору(Рис.2.35.):

### Рис. 2.35. Вигляд «Головного меню»

В разі натиснення кнопки «Вихід», буде запропоновано підтвердити наш вибір(Рис.2.36.):

### Рис. 2.36. Вигляд вікна «Підтвердження виходу»

В разі натиснення кнопки «Про програму», перед користувачем буде висвітлена наступна інформація(Рис.2.37.):

### Рис. 2.37. Вигляд вікна «Про програму»

В разі натиснення кнопки «Вибір корпусу», користувачу буде запропоновано конкретизувати свій вибір наступним чином(Рис.2.38.):

### Рис. 2.38. Вигляд вікна «Вибір корпусу»

В разі вибору будь якого з варіантів, «користувача» буде «переміщено» у відповідне місце вибране ним. (на прикладі 2 корпусу 1 поверху) (Рис.2.39., Рис.2.40., Рис.2.41., Рис.2.42.):

Рис. 2.39. Початкове місце старту при виборі «Корпус №2 поверх 1»

Рис. 2.40. Вигляд збоку «Корпус №2 поверх 1»

Рис. 2.41. Вигляд на «Факультет математики та інформатики» «Корпус №2 поверх 1»

Рис. 2.42. Вигляд правої частини «Корпус №2 поверх 1»

### **2.3. Аналіз труднощів, котрі виникали в ході роботи**

Під можливими труднощами, можна розуміти, ймовірні складнощі при виконанні тих чи інших завдань, задач.

Наприклад навіть на етапі вибору основного середовища розробки, вже можна стикнутись з проблемами, будь то середовище буде підходити по всім критеріям, проте не матиме якоїсь з основних аспектів які необхідні для комфортної роботи, будь то фактор платності програми або ж зрізаність версії в зв'язку з формою розповсюдження на безоплатній основі, або мова програмування для написання «скриптів» буде відрізнитись від раніше вивчених, тому буде необхідність додатково вивчати інші мови, тощо.

На етапі вибору середовища для 3D моделювання, проблеми можуть виникнути наприклад програма не підтримує форматів які необхідні для експорту в головне середовище розробки екскурс додатку, або функціонал програми не достатній доля втілення всіх поставлених задач, і цілей, тощо.

Або ж при виборі програм для редагування зображень, можуть виникнути проблеми з розширеннями файлів, або з нехваткою навиків, або вмінь які необхідні для роботи в них, що в результаті спричинить додаткові затрати в часі на виконання тих чи інших прикладних задач.



Також не слід забувати про труднощі, перешкоди загального характеру, такі як:

- Обмеження в часі, в зв'язку з чим необхідно розподілити всі процеси по мірі необхідності в часі на їх реалізацію, щоб в короткій терміні максимально продуктивно досягнути всі поставлені цілі.
- Недостатня кількість матеріалів належної якості, в зв'язку з чим необхідно намагатись компенсувати даний не достаток або різними програмними засобами для покращення вже існуючих матеріалів.

## ВИСНОВОК

В процесі написання даної роботи було досліджено різного роду засоби та методи реалізації поставленої задачі, з описанням переваг та недоліків кожного, та подальшого вибору одного з них для виконання поставленої задачі.

Відповідно з цим були проаналізовані всі можливі аспекти роботи в даному програмному середовищі та виконані всі дії що до максимальної оптимізації робочого процесу, з завчасним підготуванням необхідних матеріалів, оцінкою не достатків програм-аналогів а також труднощів з якими можна зіткнутись при виконанні поставленої задачі, та виходячи з цього підбір альтернатив для вирішення цих не достатків.

В першому розділі даної роботи описані аналоги програмних засобів за допомогою яких можна було б виконати поставлені задачі, а також описанні конкретні програмні засоби за допомогою яких була виконана робота, враховуючи всі переваги та недоліків даних програмних засобів, та системними вимогами які необхідні за для комфортного користування ними.

В другому розділі описаний сам хід роботи, всі нюанси з якими можна зіткнутись в ході виконання поставленої задачі та засоби їх вирішення. А також візуально продемонстровані, та описані вікна з якими необхідно мати справу в ході розробки екскурсу додатку а також в допоміжних програмних засобах.

Робота виконана з детальним описом кожного з розділів. При програмуванні, моделювання, редагуванні було отримана екскурс-програма, що демонструє візуальний стиль приміщень навчального закладу.

**СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ:**

1. Арам Куксон, Райан Даулингсока Разработка игр на Unreal Engine 4 за 24 часа/ Арам Куксон, Райан Даулингсока – М.: Бомбора, 2019. – 528с.
2. Matthew DeLucas GameMaker Game Programming with GML/ Matthew DeLucas – М.:Packt Publishing, 2014. – 350с.
3. Janine Suvak Learn Unity3D Programming with UnityScript/ Janine Suvak – М.: Apress, 2014. – 443с.
4. David Byrnes AutoCAD 2011 for Dummies/ David Byrnes – М.: John Wiley & Sons, 2010. – 532с.
5. Stewart Jones Sams Teach Yourself 3ds Max in 24 Hours/ Stewart Jones – М.: Sams, 2013. – 392с.
6. Iftikhar B. Abbasov A Fascinating Journey into the World of 3D Graphics with 3ds Max/ Iftikhar B. Abbasov – М.: DMK Press, 2017. – 151с.
7. James Chronister Blender Basics: A Classroom Tutorial Book/ James Chronister. – М.: cdschools.org, 2017 - 266с.
8. Иван Хахаев Графический редактор GIMP первые шаги/ Иван Хахаев – М.: ДМК - пресс, 2009. – 223с.
9. Райтман М.А. Adobe Photoshop CS6 - Официальный учебный курс/ Райтман М.А. – М.: Эксмо, 2013. – 432с.