ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до магістерської роботи
на ступінь вищої освіти магістр
на тему: **«Розробка системи симуляції для навчання на основі VR технологій»**

Виконав: студент 6 курсу, групи ПДМ-61
спеціальності
121 Інженерія програмного забезпечення
(шифр і назва спеціальності/спеціалізації)

Кисельов В.О.
(прізвище та ініціали)

Керівник Негоденко О.В.
(прізвище та ініціали)

Рецензент _____
(прізвище та ініціали)

Київ –2022

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення
Ступінь вищої освіти - «Магістр»
Спеціальність – 121 Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ
Завідувач кафедри
інженерії програмного забезпечення

Негоденко О.В.

"_"_____2023 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Кисельов В'ячеслав Олександрович
(прізвище, ім'я, по батькові)

1. Тема роботи: «**Розробка системи симуляції для навчання на основі VR технологій**»
   Керівник роботи к.т.н., доцент Негоденко Олена Василівна
   (прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
   затверджені наказом вищого навчального закладу від 12.10 2023 року №122____.
2. Строк подання студентом роботи 31.12.2022
3. Вихідні дані до роботи:
   Науково-технічна література з питань, пов'язаних з програмним забезпеченням, програмуванням на мові C#, та роботою із технологією VR;
   Офіційна документація мови програмування C#.

4.Зміст розрахунково-пояснювальної записки (перелік питань, які потрібнорозробити)
4.1 Методи навчання з використанням VR-технологій та обробки фізики руху користувача.
4.2 Вимоги та опис системи.
4.3 Опис проектування системи.
4.4 Опис використаних технологій.

5. Перелік графічного матеріалу
1. Титульний слайд

2. Мета, об’єкт дослідження, предмет дослідження
3. Аналіз існуючих аналогів
4. Методи обробки фізики руху користувача
5. Засоби програмної реалізації
6. Опис проектування системи
7. Висновки

6. Дата видачі завдання 14.10.2022

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської роботи	Строк виконання етапів роботи до	Примітка
1	Підбір науково-технічної літератури	14.10-16.10	Виконано
2	Формування ідеї дослідження	16.10-20.10	Виконано
3	Аналіз існуючих моделей фізики руху в віртуальній технології	20.10-25.10	Виконано
4	Модифікувати модель фізики руху на основі двигуна Unity	25.10-15.11	Виконано
5	Вступ, висновки, реферат	15.11-20.12	Виконано
6	Попередній захист роботи	20.12-22.12	Виконано
7	Здача роботи	31.12	Виконано

Студент _____ Кисельов В.О.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Негоденко О.В.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи с. 65, рис. 24, джерел 20.

Об'єкт дослідження – процес та механізми функціювання технології VR у галузі навчання.

Предмет дослідження – технології віртуальної реальності.

Мета дослідження – підвищити ефективність практичної складової навчання за допомогою віртуальної реальності.

У роботі проведено аналіз існуючих рішень програмного забезпечення, які дозволяють проводити роботу використовуючи VR-технології. Розглянуто використання програмних засобів, аналіз їх переваг та недоліків. Було вибрано найкращі програмні інструменти для створення додатку.

Проведено описання використаних програмних засобів та середи розробки.

Ключові слова: C#, .Net, VR, Unity SourceTree, Об’єктно орієнтоване програмування.

ЗМІСТ

ВСТУП

Об'єкт дослідження процес та механізми функціювання технології VR у галузі навчання.

Предмет дослідження – технології віртуальної реальності.

Мета дослідження – підвищити ефективність практичної складової навчання за допомогою віртуальної реальності.

Новизна проекту – удосконалення моделі фізики руху існуючих додатків, розробка додатку та інтерфейсу, який зрозумілий та очевидний для кінцевого користувача.

У дипломному проекті було проведено аналіз між існуючими аналогами VR додатків та проведено аналіз недоліків.

У навчальному процесі ВНЗ технологічні системи, які базуються на комп'ютерних технологіях, відіграють величезну роль, оскільки повне занурення до навчального процесу за допомогою спостереження за максимально реалістичною картинкою підвищує мотивацію та успіхи в отриманні інформації й знань. Як один з перспективних освітніх методів сучасні інформаційні технології пропонують нове освітнє середовище – Віртуальна реальність (VR), яка моделюється комп'ютером і розглядається як особливе інформаційне середовище, є одним з перспективних методів висвітлення в сучасних інформаційних технологіях.

1 АНАЛІЗ ПРОЕКТНОГО РІШЕННЯ

1.1 Характеристика процесу діяльності

Навчальна симуляція - це штучне цифрове середовище, яке дозволяє слухачам імітувати сценарії на робочому місці для навчання та підвищення кваліфікації. Сьогодні технологія VR дозволяє відтворювати різні сценарії реального життя в навчальному середовищі, наприклад, готовність до стихійних лих або реалістичні тренування на висоті.

Середовища віртуальної реальності дозволяють слухачам використовувати реалістичне навчальне середовище, яке відображає будь-який світ. Використовуючи VR-додатки, контролери або датчики, підключені до девайсів, вони можуть взаємодіяти з об'єктами та іншими людьми. Рішення для навчання у сфері VR надають великі можливість навчати свій персонал у 3D-середовищі без ризиків, які існують у реальному світу.

Однак безпека – одна з багатьох переваг, яку пропонує VR:

- візуал (використовуючи 3D-графіку, можна деталізовано показати всі процеси аж до найнижчого рівня. Також нічого не забороняє заглибитися ще далі і показати, як всередині самого атома відбувається поділ ядра перед ядерним вибухом. Віртуальна реальність здатна не тільки дати інформацію про саме явище, а й продемонструвати його з до найдрібнішої деталізації);
- залучення (VRдозволяє редагувати сцени, впливати на прогрес і вирішувати завдання в ігровій формі. Сеанси VR дозволяють досліджувати світи минулого очима історичних особистостей).
- VR-заняття (Вигляд від першої особи та ваша присутність у зображуваному світі – одна з основних особливостей віртуальної реальності, що дозволяє проводити заняття повністю у віртуальній реальності).
- скорочення часу та витрат на навчання (у більшості випадків процес навчання може бути дуже затратним. Наприклад, комплексна та практична підготовка медичних працівників та пожежників потребує значного здоров'я і, може, коштувати

вам набагато дорожче, у томучислі навіть життя)

- гнучкість у навчанні (сценарії легко додаються або змінюються, середовища та об'єкти можуть бути змінені, але це необов'язково відноситься до налаштувань фізичних елементів плюс необхідне програмне, апаратне та потужне програмне забезпечення VR можна використовувати для віртуального навчання у будь-якому місці, у будь-який час. Ця гнучкість дозволяє отримати доступ до навчання для співробітників ,що працюють за різними графіками)

Основні кейси для використання навчальної симуляції:

Будівництво

Будівельна галузь відома своїми небезпеками. Освітні симуляції віртуальної реальності також можуть модернізувати методи будівництва та створювати безпечне та безризикове робоче середовище.

3D-симуляція та технологія віртуальної реальності у будівництві можуть використовуватися для відтворення небезпечних ситуацій, підготовки персоналу до проблем на будівельному майданчику, надзвичайних ситуацій, таких як падіння та пожежі, а також для імітації роботи у несприятливих погодних умовах.

Охорона здоров'я

Навчання у віртуальній реальності має великий потенціал для підвищення кваліфікації лікарів та галузі охорони здоров'я в цілому. Завдяки повному зануренню, навчання у віртуальній реальності ідеально відтворює реальну роботу, дозволяючи фахівцям у галузі охорони здоров'я вдосконалювати свої навички без жодних ризиків

Енергетика

За допомогою VR-симуляції ми можемо забезпечити електробезпеку. Це те, що зробив технологічний гігант Intel. За допомогою VIVE's Enterprise корпорація Intel запустила курс повторної сертифікації електробезпеки. Цей навчальний курс з ефектом занурення у віртуальну реальність та реалістичнми сценаріями управління ризиками. В результаті компанія змогла скоротити витрати на навчання та підвищити ефективність роботи співробітників

Воєнна промисловість

На землі: поле бою та бойова підготовка

Армії в усьому світі використовують VR для підготовки на полі бою і бойової підготовки. Віртуальна реальність може забезпечити реалістичну цифрову реплікацію складного середовища, надаючи солдатам безпечний спосіб навчитися взаємодіяти з ворогом, одночасно справляючись із зовнішніми факторами, такими як присутність цивільного населення, а також різноманітні умови, в тому числі різний час доби і погодні виклики. Проходячи тренування з віртуальної реальності, солдати можуть навчитися справлятися з високим рівнем стресу і вдосконалити низку навичок, від ефективної комунікації до критично важливих бойових прийомів. Це також може допомогти їм навчитись боротись з ворожим оточенням, таким як терористи-смертники або снайперські атаки.

Протягом 2020 року Йоркширський полк британської армії випробовував програму підготовки за допомогою віртуальної реальності, надаючи солдатам можливість відпрацьовувати дії на полі бою, що допомагає їм підготуватись до оперативного розгортання.

Тренування з обладнанням

Від тактильних VR-гармат до бронетехніки, підводних човнів, які запускають віртуальні торпеди, і не тільки, VR застосовується у програмах військової підготовки, які зосереджуються на здатності слухачів використовувати конкретне обладнання у своїй роботі.

Програма Pilot Training Next висвітлює, як віртуальна реальність може допомогти в підготовці особового складу в масштабах, подолавши розрив між навчанням в класі і дорогими (і обмеженими) традиційними симуляціями. Вона спростила підготовку пілотів новинних служб ВПС США за допомогою пристроїв HTC VIVE разом з додатковим обладнанням для відтворення віртуальної кабіни пілота. У поєднанні з передовою біометрією і штучним інтелектом ця програма дає слухачам можливість відпрацьовувати техніки, які можна ефективно контролювати і оцінювати.

Віртуальні військові кораблі

Військово-морські установи по всьому світу запровадили навчання з використанням віртуальної реальності для того, щоб забезпечити ознайомлення корабельного персоналу зі складним обладнанням, з яким вони стикаються, від інженерів з технічного обслуговування до пілотів гелікоптерів і офіцерів на містках - і не тільки.

Віртуальна реальність також використовується як інструмент орієнтаційної підготовки у Королівських ВМС. Персонал, який призначається на авіаносці класу "Квін Елізабет" (QEC), може ознайомитися з планом корабля, в тому числі з розташуванням ключових елементів обладнання безпеки, а також шляхами евакуації ще до того, як ступить на борт корабля. Програма підготовки також перевіряє реакцію користувача на надзвичайні сценарії.

Таблиця 1.1 – Опис функцій застосування

Функція	Опис Функції
1. Регестрація\Аутентифікація	Створення облікового запису користувача, використовуючи пошту та персонального паролю
2. Вибір навчальної симуляції	Користувач обирає напрям використання навчальної симуляції
3. Завантаження необхідних бібліотек для роботи	Завантажується всі необхідні частини для роботи обраної симуляції
4. Вибір ситуації для симулювання з наданого списку	В залежності від обраного напряму користувач може обрати ситуацію, яку буде симулювати додаток

1.2 Огляд існуючих аналогів

На сьогоднішній день технології віртуальної реальності досягають високого рівня розвитку, а отже існують вже багато додатків, які можна використовувати, як

симулятори для навчання в багатьох галузях.

Construction VR\MR - симулятори віртуальної реальності дають змогу будівельникам моделювати складні та небезпечні сценарії реального життя та безпечно поводитися з небезпечним обладнанням.

Завдяки технології віртуальної реальності робітники знають, як підготуватися та ефективно діяти в надзвичайних ситуаціях, таких як падіння, пожежі та робота в погану погоду.

Як приклад, наша робота, віртуальна симуляція будівельного майданчика хмарочоса. Основна ідея цієї навчальної VR-симуляції полягає в тому, щоб дозволити робітникам повторити свою роботу на будівельному майданчику, відпрацювати свої навички та покращити свої знання в галузі безпеки.

Рисунок 1.1 – Construction VR\MR

Microsoft Flight Simulator – це третій великий випуск Microsoft Flight Simulator та останній, розроблений Aces Game Studio.

Сюди входять оновлення графічного двигуна та зворотна сумісність з DirectX 10 та Windows Vista. Реліз відбувсяу Північній Америці 17 жовтня 2006 року. Є дві версії гри, обидві на двох DVD. Deluxe Edition включає три кабіни, додаткові літаки

та нову систему авіоніки Garmin G1000, інтегровану в місії.

Можливість управління вежами, які розраховані на багато користувачів в онлайн режимі, більш деталізовані ландшафти аеропортів.

Та комплект розробки програмного забезпечення (SDK) для розробки. Основні покращення пов'язані з графікою. Microsoft також випустила демо-версію Flight Simulator X, що містить три літаки, два аеропорти та дві місії. Сумісність з операційними системами Windows XP SP2 та Windows Vista.

Рисунок 1.2 - Microsoft Flight Simulator

AHTS Ship Simulator – симуляція керуванням 3000-тонним судном і управління ним у відкритому морі. Данна симуляція призначена для вдосконалення професійних навичків керування кораблем, за допомогою розважальної але реалістичного додатку.

Додаток використовую реалістичний корабель з двигунами потужністю 4x2 кВт та носові і кормові бічні рушії, щоб вивести судно для обробки якоря та буксирування до нафтової вишки та утримання його протягом 10 секунд, щоб екіпаж міг стати на

якір.

Також реалізовані критичні ситуації, по типу – пожежа, нестабільні погодні умови та критичні несправності.

Також, як розважальна частина, є можливість увімкнути бойові зіткнення з піратами, з можливістю використання пускових установок або пушок.

Рисунок 1.3 – AHTS Ship Simulator

Human Anatomy VR - надання студентам, медичним школам, школам медсестер, університетам, системам охорони здоров'я, медичним асоціаціям та практикуючим лікарям найцікавішого та інтуїтивно зрозумілого досвіду анатомії людини, доступної на ринку. Багатий контент доставляється в глибокому зануренні з чудовою графікою, інноваційною презентацією та візуальною виставою.

Програмне забезпечення пропонує 15 систем тіла із більш ніж 10 000 реалістичних анатомічних структур, розроблених медичними експертами. Картування кісток з 5000 елементами кісток, організованими у частині, поверхні, межі та орієнтири. 21 мікроанатомічна модель, понад 500 анімацій руху та багато іншого.

Реалізована можливість вивчення всього людського організму без будь-якого медичного втручання.

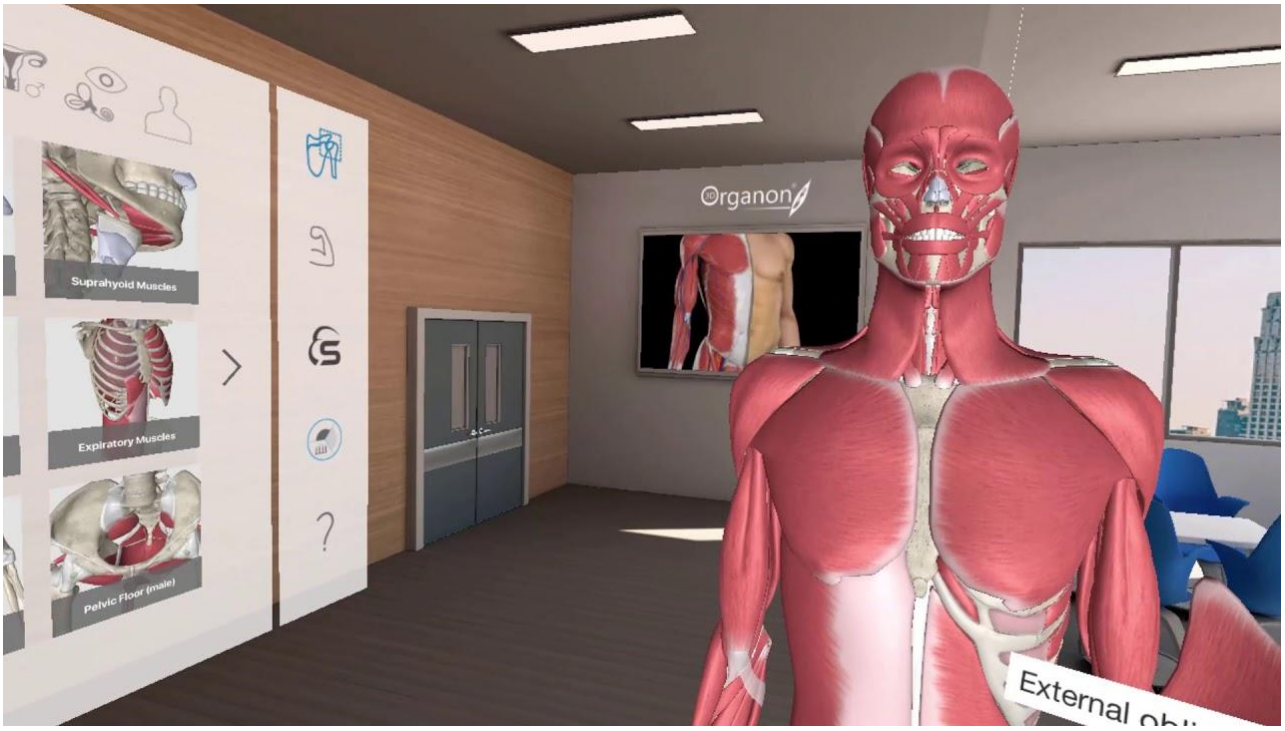


Рисунок 1.4 - Human Anatomy VR

Holofit App - перетворює ваш домашній спортзал або звичайне стаціонарне кардіо на пригоду з дослідження віртуального світу. У поєднанні з Google Street View Holofit перетворює понад 10 мільйонів кілометрів доріг на тривимірний світ, який оточує вас, поки ви потієте. Прокотіться Тур де Франс на велотренажері у своїй вітальні або прокатіться вулицями Сан-Франциско зі свого гаража.

Візуальні ефекти неймовірно стимулюють і час летить непомітно. Якщо вам не подобається реалістичне дорожнє покриття, ви можете використовувати педальний привід, щоб їздити трасою.

Є цікаві завдання і щось для всіх. Ще однією перевагою Holofit є те, що вам не потрібно мати еліптичний тренажер, бігову доріжку чи велотренажер. Режим фрістайлу дозволяє кататися на дошці Exaboard. Це віртуальний тренажер, який дозволяє вам задавати свої рухи, виконуючи рухи всього тіла, створені професійними тренерами. Це неймовірно складне та відмінне кардіо.

Рисунок 1.5 - **Holofit App**

Construction VR\MR та Humar Anatomy VR – використовують модель Asynchronous reprojection, яка лягає в основу їх двигуна.

Asynchronous reprojection – це модель, яка спрямована на забезпечення реакцій приладів відтворення віртуальної реальності на рух користувача, навіть коли графічний процесор не має можливості максимальну частоту кадрів, що призводить до постійної втрати кадрів і погіршення досвіду використання VR технології.

Репроекція полягає в тому, що драйвер бере кілка раніше відрендерних кадрів і використовує інформацію про рух користувача, отриману з приладів і датчиків гарнітури для екстраполяції(або відтворення) минулого кадру, використовуючи його, як прогноз того, що буде відображено на звичайному кадрі.

Також, ця модель підтримує асинхронне виконання, що означає, що цей процес безперервно виконується паралельно з рендерингом, що дозволє відображати отримані кадри без затримок, якщо звичайний кадр не був відтворений своєчасно.

AHTS Ship Simulator та Holofit – використовують модель Asynchronous Timewarp, яка лягає в основу їх двигуна .

Timewarp, також відоме, як репроекція – це технологія у віртуальній реальності, яка викривлє відтворений образ перед відправкою його на дисплей користувача, щоб виправити рух, що ставс після рендерингу. Timewarp може зменшити затримку або збільшити, або підтримувати статчину частоту кадрів. Крім того, він може зменшити тремтіння, спричинене пропуском кадрів.

Цей процес бере вже оброблене зображення, додає на нього свіжу інформацію про положення, отриманної від датчиків HMD, а потім відтворюють його на дісплеї.

Timewarp – є функцію, яку вперше представили OCULUS SDK. Потім оновлена версія була представленя в OCULUS PC SDK.

Asynchronous Timewarp – це модель, коли Timewarp відбувається в іншому потоці (паралельно або асинхронно) з пендерингом зображення. Перед кожною синхронізацією потіс ATW відтворює нові кадри, із зміщенням часу з останього кадру, завершеного потоком рендерингу.

AWS заповнює пропщуенні кадри і зменшує візуальне тремтіння на дисплеї користувача.

1.3 Постановка задачі

Домен - це частина реальної системи, яка є предметом цього дослідження. Під час проектування автоматизованих інформаційних систем предметна область представлена багаторівневими моделями даних. Кількість рівнів залежить від складності розв'язуваних завдань, але кожен охоплює концептуальний і логічний рівень.

Предметом цієї роботи є симуляція для використання в навчанні, завданням якої є створення безпечного середовища в багатьох галузях, як зазначалося раніше.

Провівши аналіз аналогів та визначивши їх недоліки, було вирішено створювати додаток на мові c# використовуючи двигун Unity. Це одна з найкращих мов програмування і двигунів для створення подібних рішень.

Також, обираючи середовище розробки, було вирішено використовувати

Microsoft Visual studio 2022, яка напряму пов’язана з двигуном Unity.

Також, користувачі, після реєстрація можуть обрати напрям, в якому вони будуть використовувати симуляцію: Воєнна промисловість, спорт, медицина...

Створення начальної симуляції, основним завданням якої є отримання необхідних навичок, на основі обраного напряму симуляції, виключаючи всі небезпечні аспекти навчання.

Основні цілі розробки:

- прибрати всі небезпечні фактори в навчанні;
- полегшити процес навчання;
- забезпечити можливість навчання з будь-якої точки світу;

Для досягнення поставлених цілей необхідно вирішити наступні задачі:

- розробити оболочку для двигнуа Unity;
- розробити модель руху користувача;
- розробити базу даних, для зберігання всієї потрібної інформації;
- реєстарція користувачів у системі;
- розробка рівнів складності для кожної симуляції;
- поповнення баз знань інформацією про нові дослідження в різних галузях науки;

1.4 Вибір інформаційного забезпечення

Вхідні дані.

- персональні дані учнів при реєстрації: ФІО, логін, пароль.
- персональні дані вчителів при реєстраці: ФІО, логін, пароль, ключ
- інформація про напрям використання симуляції.
- інформація про рівень обізнаності в обраному напрямі.
- інформація про кількість напрямів в симуляціях.
- інформація про оцінки і рівень обізнаності учня.

Вихідні дані.

- список обраних симуляцій, що виводяться у відсотрованому порядку з відповідним значенням рівня обізнаності в кожному з напрямів.
- інформація про відвідування занять. Яка доступна й учням, й вчителям.
- база знань по кожному з напрямів.
- Сторінка статистики для вчителів по кожному з учнів.
- Сторінка статистики по кожному з напрямів для учнів, з оцінками.

Таблиця 1.2. – Структура та призначення сутностей бази даних

Назва	**Поле**	**Призначення**
Teachers – містить інформацію про вчителя	[Id] INT	Унікальний ідентифікатор
	[Name] NVARCHAR	Ім’я
	[SURNAME] NVARCHAR	Фамілія

Продовження Таблиці 1.2. – Структура та призначення сутностей бази даних

	[SEX] NVARCHAR	Стать
	[SPECIALITYNUMBR]INT	Номер спеціальності
Specialty – містить детальну інформацію про спеціальність	[Id] INT	Унікальний ідентифікатор
	[DESCRIPTION]NVARCHAR	Опис спеціальності
Student - містить інформацію про учня	[Id] INT	Унікальний ідентифікатор
	[Name] NVARCHAR	Ім’я
	[SURNAME] NVARCHAR	Фамілія
	[SEX] NVARCHAR	Стать
	[SPECIALITYNUMBR]INT	Номер спеціальності

Auth User – містить інформацію про користувачів	[Id] INT	Унікальний ідентифікатор
	[PASSWORD] VARCHAR(128)	Пароль
	[LAST_LOGIN] DATETIME(6)	Дата останнього входу в систему
	[IS_TEACHER] TINYINT(1)	Чи має права адміністратора (для редагування)
	[USERNAME] VARCHAR(150)	Логін

Продовження Таблиці 1.2. – Структура та призначення сутностей бази даних

	[FIRST_NAME] VARCHAR(30)	Ім’я
	[LAST_NAME] VARCHAR(150)	Прізвище
	[EMAIL] VARCHAR(254)	Електронна адреса користувача
	[IS_ACTIVE] TINYINT(1)	Чи зараз у системі
	[JOIN_DATE] DATETIME(6)	Дата реєстрації
Registration key – список ключів для реєстрації вчителів	[Id] INT	Унікальний ідентифікатор
	[KEY]NVARCHAR(100)	Ключ реєстрації

SystemAnalysis - глибинна інформація системи користувача	[Id] INT	Унікальний ідентифікатор
	[Id_user]INT	Унікальний ідентифікатор користувача
	[GPU]NVARCHAR	Інформація про відеокарту
	[CPU]NVARCHAR	Інформація про процесор
	[RAM]NVARCHAR	Інформація про оперативну пам’ять

2 АНАЛІЗ МАТЕМАТИЧНИХ МОДЕЛЕЙ ТА МЕТОДІВ РОЗРОБКИ VR

2.1 Основні методи розробки VR

Програмне забезпечення VR, як правило, оптимізовано для роботи з оновленням частоти HMD, хоча і ізольовано. Програма може пропускати кадри, якщо працює Oculus Dash, що в свою чергу зменшує ефективність CPU або GPU для проміжного бюджету продуктивності.

Основним методом розробки буде - покращення методу AWS, а саме доробка його фізики руху, з використанням основи VR двигуна запровадженного Unity.

Основа розробляємого методу полягає в техніці зглажування частоти кадрів, та покращення фізики руху таким чином, щоб зменшити навантаження та час роботи відеокарти і процесору, необхідний для отримання статичних результатів з однакового контенту.

Даний метод має працювати таким чином, щоб при розробці VR додатку, розробнику не потрібно було вмикати або вимикати його, оскільки він асинхронний, він має постійно працювати в фоні, без будь-яких додаткових зусиль з боку

розробників.

Також, даний метод буде підтримувати, навіть при зниженій частоті кадрів не менше 45 Гц, що буде значно покращувати вигляд у VR, оскільки відвторення зображення бу на основі “Depth” аналізу, який працює більш точно при низькій чатсоті кадрів, ніж екстраполяція ASW. Якщо казати просто, то ми будемо отримувати більш плавне зображення, плавну фізику руху та відсутність втрати якості зображення, навіть, якщо частота кадрів падає до половини значення за замовчуванням.

Також, даний метод вирішує проблему шаруватості або помилок відтворення зображення на дисплеї користувача, використовуючи дані про глибину (або метод Depth) для розділення всіх об’єктів перед екстраполяцією.

Що таке метод “Depth”? Він буде вимагати від розробників розкривти глибинну інформацію про зображення, що буде давати можливість підбирати налаштування під кожну систему незалежно.

Данний метод буде відігравати важливу роль у рендерингу зображень в реальному часі. По суті - це двовимірний масив значень, який буде генеруватись графічним процесором під час растеризації нашої віртуальної сцени, де кожн елемент буде зберігати значення глибини, для відповідного елемента нашого створеного буфера. Як тільки даний буфер буде згенерований, під час рендерингу VR-симуляції, оскільки це аснихронно, очікується, що VR-додаток прдеставить його вміст в автоматично згенерований клас, який, в свою чергу, можна буде передати в базу даних, для подальшого використання, з можливістю зібрати цю інформацію один раз, що дає прискорення роботи в майбутньому.

Також, буфер глбини буде зберігати значення у форматі з плаваючуою комою або цілого значення, однак - це будуть, спочатку, необроблені значення відстані від і до певного пікселя. Отже, краще, обробляти його під час растрування зобржаення, для чого буде використовуватись проекційна матриця, яка в свою чергу буде перетворювати кожну вершину і піксель в потрібне нам значення глибини, яке буде зберігатись в пам’яті.

Цей спосіб обробки зображення, буде одним з основних в данній роботі, оскільки проекційна матриця перетворює лінійну відстань у динамічне значення, готове до зберігання в буфері глибини і це, можна вважати, як спсіп ефективного відображення значень відстаней, що дасть нам вищу точність для елементів, розташованих ближче до глядача.

Все ця модель буде працювати для ідеальних умов але різний контент, може вимагати різних схем мапингу інформації. Наприклад, у разнні часи низькоточних форматів глибини на основі цілих чисел, зображення було найменш точним, що призводило до втрати якості зображення, та втрати кадрів.

З огляду на це, також, вирішенно, створити допоміжні функції, які можуть використовуватись при створенні матриці проекції, а також, для того щоб, перераховути значення для створення загальних способів створення матриць. В загальному, це буде тип enum, який буде вміщувати 4 значення, для кожної з сторін перерахункує.

Нас цікавить відстань всіх пікселів до камери рендереингу, яка буде відобаражати віртуальні “очі” гравця, в одиницях відстеження простору. Для цього, як було вказано раніше, буде збиратись не тільки дані глибин а й додаткові метадані, яка буде дозволяти робити перерахунок початкової лінійної відстанні між пікселями. Сама матриця відтворення буде основною частиною для відтворення зображення, але не вся матриця, а лише ті частини, які мають відношення до елементів візуалізації, що дає можливість динамічного відтворення зображення, що ідеально підходить для данного додатку, оскілкьи не потрібно ренедерити всю карту, яку буде використовувати гравець, в один час, що дасть можливість покращити досвід використання додатку.

Також, інша частина даних, яку потрібно буде врахувати - це масштаб світу і виду, який буде виокристовувати наш додаток, для блоків рендерингу. Це частина, яку візьме на себе реалізований функціонал двигуна Unity.

З цього випливає ще одна проблема VR-додату – позиційне тремтіння, коли користувач знаходиться в нерухомості, але зображення тремтить і псує досвід

використання. Для усунення цього дифекту – ми застосуємо повне позиційне викривлення, яке застосовує, як переведення, так і виправлення орієнтації, до вихідного зображення. Ми повинні враховувати глибину кадру (що ми робимо в Depth analaysis), зміщуючи частину зображення ні різну велечину. Але, таке зміщення відтворює дефекти роз'єднання на краях згенерованних об'єктів, де є зони без даних вихідного кадру.

Також, це дорогий метод, в плані навантаження на відеокарту, того йому важко впоратись з прозорістю і має проблеми з згладжуванням зображення.

Ще одна важлива частина розробки – анімовані або рухомі об'єкти, які будуть викликати більше дефектів зображення, оскільки, нові зображення (оскілкьи кожен об'єкт – це сітка трикутників) будуть створюватись шляхом викривлення оригінального зображення без знання шляху руху об'єктів, для всіх кадрів, які буде створювати метод, а отже, вони будуть застигати в часі, що буде призводити до втрати кадрів і замилення зображення. А, отже, сцени з рухомими об'єкатми будуть викликати псування зображення.

Цей дефект напряму залежить від кількості, площі проекції та швидкості руху згенерованого об'єкту. Все просто, чим більша кілкість об'єктів – тим більша втрата кадрів. Також, якщо кількість рухомих об'єктів заповнють більшу частину дисплею – це може викликати занепокоєння.

Використовуючи глибинний аналіз та оновлену фізичну модель руху об'єктів ми позбуваємось цих дефектів. Крім того, співвідношення частоти кадрів між рендерингом, буде впливати на сприйняття якості та ефект тремтіння руху. А, отже, ми уникаємо фіксованності частоти кадрів. Наприклад, при частоті оновлення 90 Гц ми досягаємо 90 Гц, якщо система не відповидає вимогам додатку – він автоматично буде брати половину частоти – 45 Гц, що призведе до подвоєння зображення, але, оскільки, ми будемо розташовувати подвійні зображення взаємно – на сітківці воно буде стабільним.

Отже, новий метод має наступні переваги над існуючими:

- Мінімізація втрати кадрів.

- Покращення зображення, за рахунок зниження навантажень на відеокарту і процесор.
- Знження мінімальних вимог до системи користувача.
- Покращенний досвід використання VR-технології.
- Асинхронність рендерингу світу, що дозволяє обробляти інформацію про об'єкти динамічно.

2.2 Математична постановка задачі

Перш за все нам потрібен віртуальний світ, в якому будуть міститися геометричні моделі.

Для наших цілей достатньо мати тривимірний евклідів простір з декартовими координатами. Тому нехай він позначає віртуальний світ, в якому кожна точка представляється у вигляді трійки дійсних координат: (x,y,z). Координатні осі нашого віртуального світу показані на рисунку 3.1. Ми будемо послідовно використовувати правосторонні системи координат, оскільки вони є переважним вибором у фізиці та інженерії, найвідомішою з яких є бібліотека графічного рендерингу DirectX компанії Microsoft.

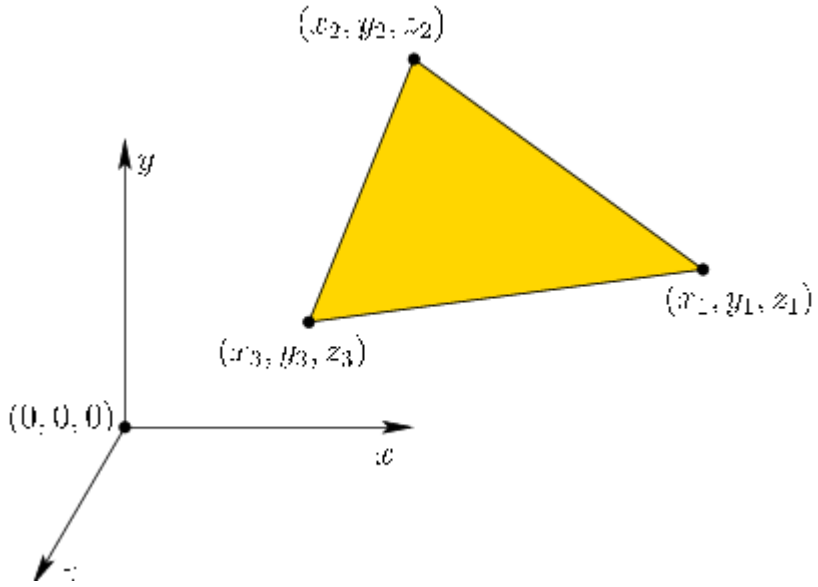


Рисунок 2.1 - Модель об’єкту у віртуальному світі

Точкам віртуального світу задаються координати в правосторонній системі координат, в якій вісь Y спрямована вгору. Початок координат (0,0,0) лежить в точці перетину осей. Також зображено тривимірний трикутник, який задано трьома вершинами, кожна з яких є точкою на просторі

Основна мета - створити основу для графічного рендерингу, який додає ефекти завдяки освітленню та властивостям матеріалів.Зрештою, результат з'являється на фізичному дисплеї. Вони також пояснюють, як камери формують зображення, принаймні ідеалізовану математику цього процесу. Уявіть собі, віртуальну камеру, яка розміщена у віртуальному світі. Як має виглядати віртуальна картинка, знята цією камерою? Для того, щоб віртуальна реальність працювала правильно, "камера" повинна фактично бути одним з двох віртуальних людських очей, поміщених у віртуальний світ. Отже, що повинно бачити віртуальне око, виходячи з його положення і орієнтації у віртуальному світі? Ми просто розрахуємо, де вершини моделі з'являться на плоскому, прямокутному екрані у віртуальному світі.

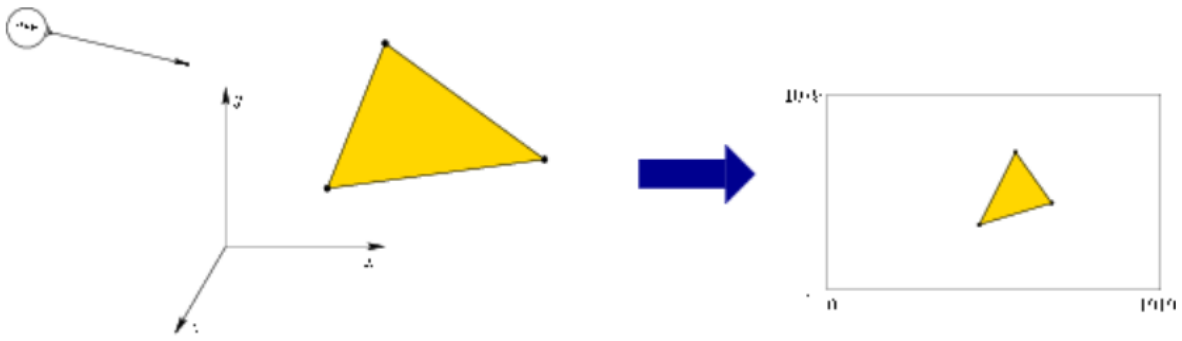


Рисунок 2.2 - Розташування камери у VR

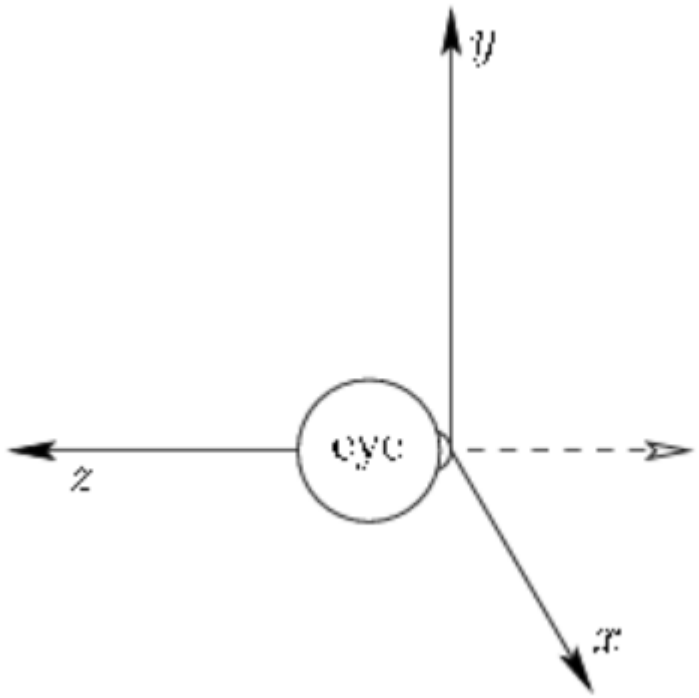


Рисунок 2.3 - Модель віртуального ока

Розглянемо око, яке дивиться вниз по осі Z в негативному напрямку. Початком моделі є точка, в якій світло потрапляє в око.

На рис. 3.3 зображено віртуальне око, яке дивиться вниз по від'ємній осі Z. Воно розміщене таким чином, що з точки зору ока X збільшується вправо, а Y збільшується вгору.

Припустимо, що око - це об'єктна модель, яку ми хочемо помістити у віртуальний світ в деяке положення E=(e1, e2, e3) і напрям заданий настуною матрицею.

$$R = \begin{matrix} x1 & y1 & z1 \\ x2 & y2 & z2 \\ x3 & y3 & z3 \end{matrix} \quad (1)$$

Замість того, щоб переміщати око у віртуальному світі, нам потрібно

перемістити всі моделі у віртуальному світі до системи відліку ока. Це означає, що нам потрібно застосувати зворотне перетворення що призводить до відповідної матриці:

$$T = \begin{matrix} x1 & x2 & x3 & 0 \\ y1 & y2 & y3 & 0 \\ z1 & z2 & z3 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \rightarrow \begin{matrix} 1 & 0 & 0 & -e1 \\ 0 & 1 & 0 & -e2 \\ 0 & 0 & 1 & -e3 \\ 0 & 0 & 0 & 1 \end{matrix} \tag{2}$$

Як зазначалося, це може відповідати переміщенню всіх моделей віртуального світу. Більш прийнятною інтерпретацією в даній ситуації є те, що система координат віртуального світу переміщується таким чином, щоб вона відповідала системі координат ока з рис. 3.3.

Нехай рухома модель задана у вигляді сітки трикутників. Для її переміщення ми застосовуємо одне перетворення до кожної вершини кожного трикутника. Спочатку розглянемо простий випадок переміщення, а потім значно складніший випадок обертання. Комбінуючи переміщення і обертання, модель можна розмістити в будь-якому місці і при будь-якій орієнтації у віртуальному світі.

Розглянемо наступний 3D трикутник якому координати його вершин виражені через узагальнені константи. Нехай x_t, y_t і z_t - величини, на які ми хочемо змінити положення трикутника вздовж осей x, y і z відповідно. Маємо наступну операцію зміни положення:

$$\begin{aligned} (x1, y1, z1) &\longrightarrow (x1 + xt, y1 + yt, z1 + zt) \\ (x2, y2, z2) &\longrightarrow (x2 + xt, y2 + yt, z2 + zt) \\ (x3, y3, z3) &\longrightarrow (x3 + xt, y3 + yt, z3 + zt) \end{aligned} \tag{3}$$

А оскільки кожен об’єкт це сітка трикутників, то достатньо застосувати перетворення тільки до вершин. Всі трикутники збережуть свої розміри і форму.

Тепер обертання об’єкту - одним з найпростіших способів параметризації тривимірних обертань є побудова їх з ``2D-подібних" перетворень використовуючи метод “Yaw, Pitch, Roll” (Обертання навколо осі спереду назад називається Roll. Обертання навколо осі з боку в бік називається Pitch. Обертання навколо вертикальної осі називається Yaw).

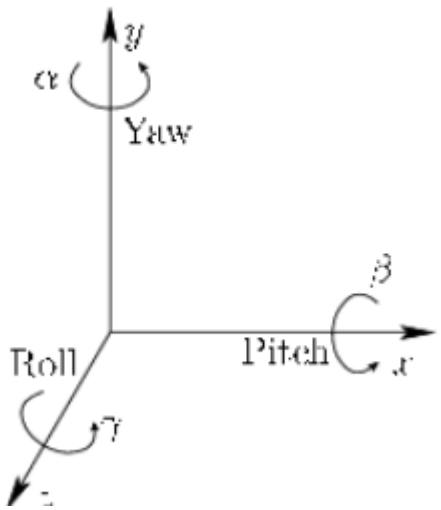


Рисунок 2.4 – Схема “Yaw, Pitch, Roll”

Спочатку розглянемо обертання навколо осі Z. Нехай “Roll” називається обертання гамми проти годинникової стрілки навколо осі Z. Матриця обертання задається наступною формулою –

$$Rz(\gamma) = \begin{matrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{matrix} \quad (4)$$

Так само, нехай “Pitch” це обертання Бети проти годинникової стрілки навколо

осі X. Маємо наступну формулу –

$$Rx(\beta) = \begin{matrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{matrix} \quad (5)$$

При цьому точки повертаються відносно координат Y та Z, а координата X залишається незмінною.

І нарешті “Yaw” це обертання Альфи проти годинникової стрілки навколо осі Y. Маємо наступну формулу –

$$Ry(\alpha) = \begin{matrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{matrix} \quad (6)$$

При цьому обертання відбувається відносно X і Z при незмінності Y. К

$$R(\alpha, \beta, \gamma) = Ry(\alpha)Rx(\beta)Rz(\gamma) \quad (7)$$

3 ЗАСОБИ РОЗРОБКИ ТА ТЕХНОЛОГІЇ

3.1 Вимоги до розробки

Розроблена система має відповідати наступним вимогам:

- Надавання користувацького інтерфейсу для кращого досвіду взаємодії користувача з віртуальною реальністю.
- Надати можливість реаєстрації користувачів.
- Збір та зберігання даних пройдених симуляцій користувачем .
- Можливість вибору варіанту переміщення у головному меню.
- Можливість робити відеозапис проходження симуляції.
- Розробити базу даних для зберігання всєї інформаці.
- Надати можливість використання веб адміністративної панелі для керування аккаунтом.
- Можливість інтеграції додатку в Steam.
- Надавати можливість для тренування пам’яті користувачу.

Вимоги до апаратної частини:

- ОС – Windows 10.
- Окуляри віртуальної реальності (Oculus Rift або сумісні моделі).
- Гарнітура.
- Дві базові VR станції.
- Два контролера.
- Адаптер живлення.
- Кабель DisplayPort 1.2-USB 3.0

3.2 Використанні технології

3.2.1 Мова програмування C#

C# - це сучасна об'єктно-орієнтована мова програмування з типо безпекою.

C# належить до сімейства мов Сі може бути знайомий всім, хто працював з C, C++, Java або JavaScript.

C#-це об'єктно-орієнтована мова, та також вона підтримує компонентно-орієнтоване програмування. При розробці сучасних програм все більшого значення набуває створення програмних компонентів у вигляді автономних пакетів, що допомагають реалізувати окремі функції. Ключовою особливістю таких компонентів і те, що вони представляють модель програмування з погляду властивостей, методів, і подій. Вони мають атрибути, що надають декларативну інформацію про компонент. Вони містять власну документацію. C# надає мовні конструкції, що безпосередньо підтримують концепцію цієї роботи. Завдяки цьому C# добре підходить для створення та розгортання програмних компонентів. C# активно підтримується Microsoft. Помилки, покращення синтаксису та нові функції оновлюються швидше, ніж в інших мовах програмування.

Ця мова є однією з найпопулярніших мов програмування. Це важливо для розробників. Популярність мови прямо пропорційна кількості онлайн-матеріалів, доступних цією мовою. Більшість часу всіз вертаються до Google або Stack Overflow для вирішення своїх проблем розробки, але в багатьох випадках ви можете знайти багато відповідей на C#. Це значно економить час новачків під час вирішення різних завдань розробки.

Також свою популярність C# набув за рахунок гнучкості, якщо порівнюватиз деякими іншими мовами програмування. Є велика кількість різних додатків, які можуть бути написані за допомогою C#, .Net і Visual Studio:

- Додатки для Windows.
- Мобільні додатки.
- Веб-додатки.
- Ігри.

Додатки для Android і ios, які розробляються за допомогою додаткових

фреймворків, таких як Xamarin або Mono.

Звичайно, всі ці речі можливо виконувати і за допомогою інших мов програмування, але зазвичай, в таких випадках, використовуються сторонні інструменти інших розробників. Осно

3.2.2 Технологія віртуальної реальності

Що таке віртуальна реальність? Для того, щоб підійти до терміну "віртуальна реальність", слід спочатку розглянути значення слів "віртуальна" і "реальність" окремо. "Віртуальний" пояснюється як "несправжній" або "удаваний". У технологічному середовищі цей термін вже давно використовується для опису чогось, що не є фізичним, але присутнє за своєю функціональністю або ефектом, наприклад, віртуальний диск. Це функціональна копія CD-приводу, включаючи носій даних, який не існує як апаратне забезпечення, але з точки зору користувача функціонує точно так само, як фізичний привід.

Віртуальна реальність (VR) стала повсюдною темою, яка з ентузіазмом сприймається в різних галузях і обіцяє не що інше, як революцію в споживанні цифрового контенту. Технологію відзначають не тільки ранні послідовники та ентузіасти технологій, але все більше і більше великих компаній, таких як IKEA або Audi, роблять свою продукцію доступною для потенційних клієнтів за допомогою віртуальної реальності. І поза сферами комунікації та розваг, існує також великий потенціал в освіті, дослідженнях, медицині та терапії.

Але що таке віртуальна реальність, і як вона змінить нашу поведінку у використанні медіа в майбутньому?

Перш за все, під VR розуміється штучна реальність, створена за допомогою спеціального обладнання та програмного забезпечення. Це може бути реалістичний археологічний візит до минулих культур або імітація польоту на космічному кораблі, а також екскурсія на будівельний об'єкт у реальному масштабі. Все це складно порівняти з існуючими 3D технологіями - і ще складніше описати, не випробувавши VR на собі.

Ядром сучасного VR-обладнання є VR-окуляри (або гарнітура, наголовний дисплей) з двома дисплеями високої роздільної здатності для показу штучно згенерованих зображень і сполученою з ними системою датчиків для визначення положення і орієнтації голови. Якщо між системою датчиків і виведенням зображення на екран проходить менше 11 мілісекунд - так звана затримка "датчик-фотон" - створюється враження "присутності" у віртуальній реальності ("присутності").

Наразі на ринку представлено два класи пристроїв віртуальної реальності:

- Гарнітури високого класу, такі як Oculus Rift або HTC Vive, розроблені спільно з розробником ігор Valve, які підключаються до звичайного ПК. Перевагаю даних девайсів є підтримка SDK Oculus, розроблена компанією Oculus. Даний пакет бібліотек значно покращує досвід використання VR рисунок 3.1.

Рисунок 3.1 Набір HTC Vive

- Мобільні гарнітури, такі як Google Daydream, Samsung GearVR або картонна Google Cardboard, які значно дешевші і працюють зі смартфонами.

Рисунок 3.2 Набір Google Daydream

3.2.3 Angular

Що таке AngularJS?

AngularJS - це фреймворк JavaScript, розроблений компанією Google. Він спрямований на розробку веб-додатків і приділяє велику увагу структурі та якості. Це був перший фреймворк, який також підходив для великих корпоративних додатків завдяки своїй орієнтації на архітектуру, тестування та ізольовані компоненти в області JavaScript. Завдяки таким методам, як ін'єкція залежностей та розвиненому інструментарію, він забезпечує ефективну та підтримувану розробку програмного забезпечення на основі JavaScript. Також у є підручник з AngularJS для початківців з прикладами та поясненнями.

Коротко про AngularJS можна виділити кілька ключових моментів:

- JavaScript фреймворк для динамічних веб-додатків.
- Призначений для розробки односторінкових додатків .
- Розроблено компанією Google і продуктивно використовується там.
- Проект з відкритим вихідним кодом (з 2009 року) MVC/MVVM фреймворк, що підтримує двонаправлене зв'язування даних.
- Розроблено для хорошої тестованості

Angular - це наступна версія фреймворку з відкритим вихідним кодом AngularJS.

Кодову базу було повністю переписано і тепер вона використовує в якості основи мову TypeScript. Однак, основна ідея та концепції фреймворку залишилися незмінними, що дозволяє здійснювати міграцію або навіть гібридне використання версій. Фокус проекту розширився від розробки фреймворку до розробки цілої платформи для веб-додатків. Наразі над проектом постійно працює понад 30 осіб, яких підтримують сотні розробників зі спільноти відкритого програмного забезпечення. З впровадженням таких інструментів, як інструмент командного рядка Angular-CLI, проекти тепер можна створювати і запускати у виробництво ще швидше. Чергова версія фреймворку AngularJS.

- Повністю переробляє свого попередника.
- Перейняв багато концепцій з AngularJS.
- На основі мови TypeScript.
- Платформа для розробки веб-, десктопних та мобільних додатків.
- Розроблено командою з 30 осіб в Google.
- Спрямована на професійну веб-розробку для корпоративних додаткі.

AngularJS був першою версією фреймворку, яка була використана мільйони разів і викликала революцію в сфері розробки веб-додатків. З переходом на версію 2 мова фреймворку змінилася з JavaScript на TypeScript. Крім того, мета Angular тепер полягає в тому, щоб бути платформою для веб-розробки, тому JS був вилучений з назви, і проект отримав назву Angular 2. Навесні 2017 року команда вирішила структурувати версії за допомогою системи SEMVER (Semantic Versioning) та визначити фіксований ритм для релізів. Таким чином, наразі існує лише проект Angular, який включає в себе нові можливості та проривні зміни у чітко визначений період (кожні 6 місяців), що потім призводить до стрибка версій. Це має багато переваг для команд розробників, які можуть набагато краще планувати свою розробку та оновлення програмного забезпечення.

3.2.4 GIT

Контроль версій - це система, яка записує зміни у файлі або наборі файлів з

часом, щоб конкретну версію можна було відкликат ипізніше. Якщо ви розробник, який бажає зберегти всі версії своєї програми, дуже розумно використовувати систему контролю версій. Це дозволяє вам відкочувати вибрані файли, відкочувати цілі проекти, порівнювати зміни з плином часу та бачити, хто останнім змінив те, що може викликат ипроблему.

Система контролю версій дозволяє кільком розробникам одночасно працювати над одним проектом, не завдаючи шкоди один одному.Існує три типи SLE: локальні, централізовані та розподілені. Тепер давайте розглянемо розподілені системи контролю версій.

Розподілена система контрол юверсій – у цій системі користувачі копіюють цілі репозиторії. Це означає, що кожен розробник має копію всього вихідного коду та змін,внесених під час розробки проекту. Це означає, якщо один із серверів вийде з ладу, будь-який інший клієнтський репозиторій може бути скопійований на інший сервер для продовження роботи, без втрати даних. Схема розподільної системи контролю версій зображено на рисунку 3.3

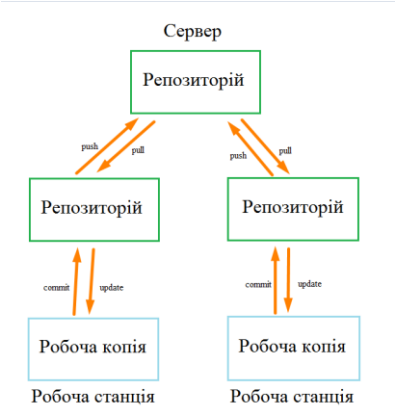


Рисунок 3.3 – Схема розподільної системи контролю версій

Git – це розподілена система контролю версій. Ця система дозволяє розробникам відстежувати зміни файлів та працювати спільно з іншими розробниками. Найбільш відомими особливостями Git є його простий дизайн, швидкість, підтримка нелінійної розробки, повна децентралізація та можливість працювати надвеликими проектами за

необхідності.

GitHub – сервіс онлайн-хостинга репозиторіїв, котрий вміщує в собі всі ті функції, що підтримує Git - функції розподільного контролю версій і функціональність управління початковим кодом. Використання GitHub разом з Git дає розробникам можливість збереження коду онлайн та зручно взаємодіяти з іншими розробниками. Важлива та зручна в розробці функція – це історія комітів, завдяки чому просто відстежувати які саме зміни було внесено. В історії комітів виділено файли та частини коду, в котрі було внесено зміни (рис. 3.4). Можливість порівняти частини коду спрощує пошук можливих помилок та відладку коду.

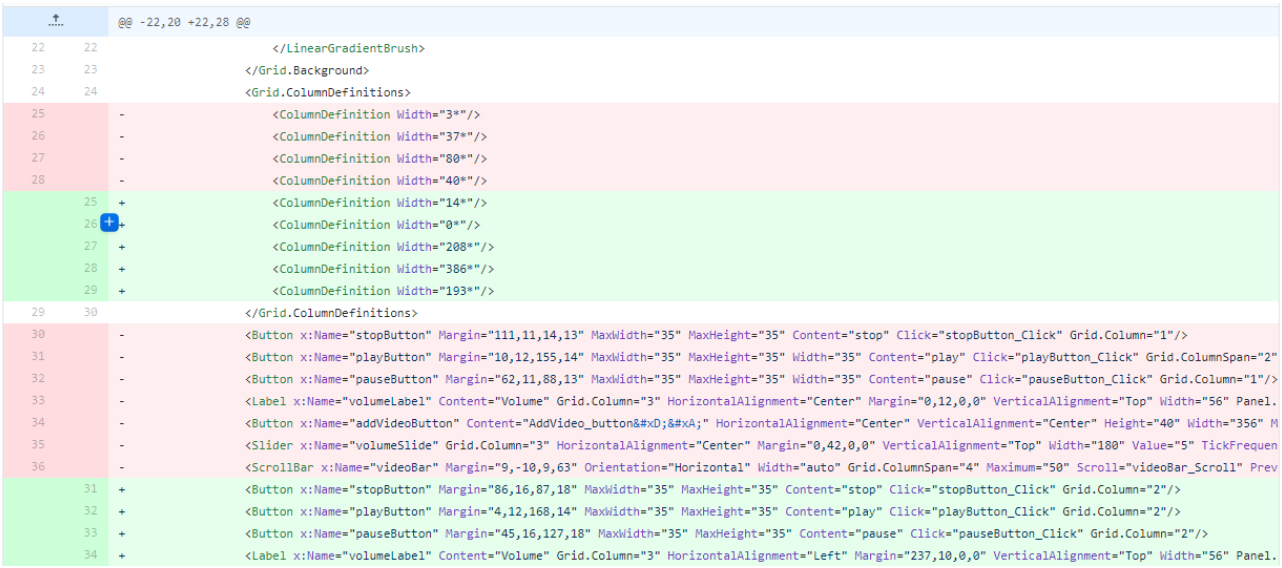


Рисунок 3.4 – Вигляд коміту на сервісі GitHub

Отже,що таке Git двома словами?

Основна відмінність Git від інших систем контролю версій полягає в тому,як Git обробляє свої дані.Концептуально більшість інших систем зберігають інформацію як списку змін файла. Ці іншісистеми (CVS, Subversion, Perforce, Bazaari т. д.) бачать інформацію ,що зберігається у вигляді послідовності файлів, та зміни, що вносяться в кожен файл з часом (це зазвичай називається керуванням версіями в базовій дельті).

Git не думає та не зберігає дані таким чином. Натомість Git думає про свої дані як про серію знімків мініатюрної файлової системи. Щоразу, коли ви фіксуєте або

зберігаєте стан свого проекту, Git по суті створює моментальний знімок того, як виглядали всі ваші файли в той момент часу, і зберігає посилання на цей знімок. Щоб бути більше фективним, Git не зберігає файл знову, якщо файл не змінився, він зберігає лише посилання на той же збережений файл.Git вважає ці дані потоком знімків. Це важлива відмінність Git від будь-якої іншої системи контролю версій. Це робить Git більш схожим на міні-файлову систему з дуже потужними інструментами, які створені поверх неї, а не просто на систему контролю версій. У Gitє контроль на сумаперед фіксацією, і мипосилаємося на цю контрольну суму. Це означає, що вміст файлу або каталогу не може бути змінено, якщо git не знає про це.

Ця функціональність вбудована на найнижчому рівні та є невід'ємною частиною. Ви не втратите інформацію та не зіпсуєте файли. Механізм ,який Git використовує для цієї контрольної суми, називається SHA-1.

Це рядок із 40 символів, що складається із шістнадцяткових символів(0-9ia-f), розрахований на основі вмісту файлу Git або структури каталогів. Слідкуйте за тим, щоб ці файли не були втрачені або пошкоджені. Все це робить гіт найкращим вибором для супроводження даного додатку.

3.2.5 SourceTree

Хоча гіт також дуже корисний з погляду підтримки проектів, але має свої недоліки. Одна з них-це постійна дюжина консольних команд, які забирають багато часу і не працюють коректно. Куди йти це досить складне завдання.

SourceTree може допомогти вирішити ці проблеми. SourceTree був створений, щоб зробити Git доступним для всіх розробників, особливо для новачків у Git.

Кожна команда Git може бути виконана одним натискання миші з використанням інтерфейсу SourceTree. Ця програма спрощує такі функції:

- Створення та клонування репозиторіїв з будь-якого місця.
- Commit, push, pull та merge в декілька кліків
- Розвинена система пошуку помилок та конфліктів.
- Відстежування всього життєвогу циклу проекту.

Як вже зазначалось, супроводження реалізації додатку буде проводитись за допомогою GitHub. Використання SourceTree спрощує роботу з Гіт до декількох кліків рис 3.5.

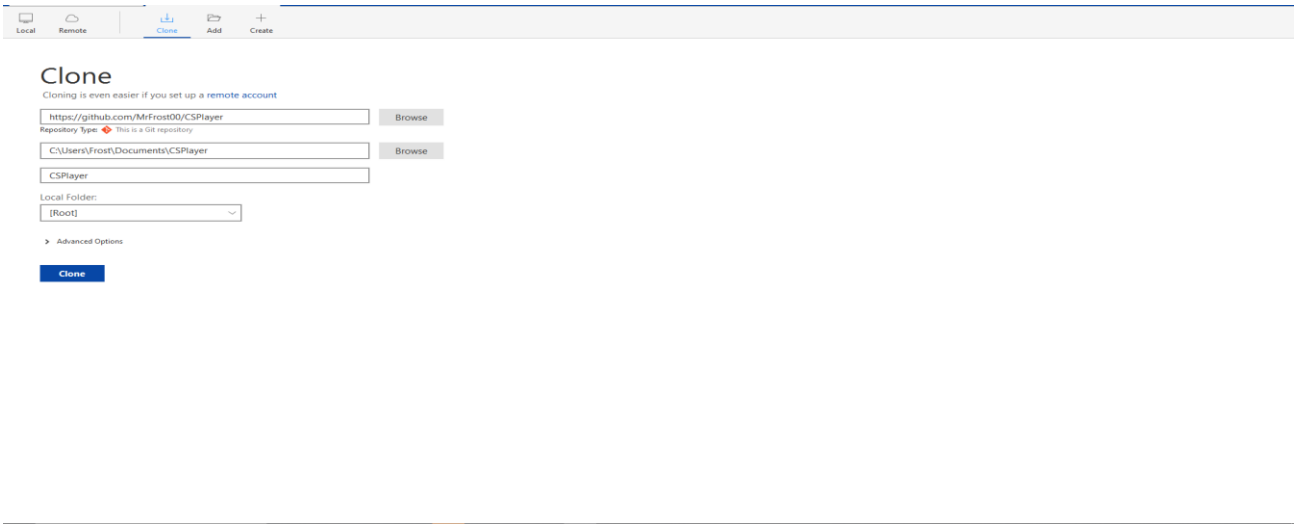


Рисунок 3.5 – Клонування\створення репозиторію

Одним з основого функціоналу гіта є – комміт внесених у проект змін, при звичайному використанні прийшлось би писати декілька консольних команд, у SourceTree все зводиться до 2 кліків рис.3.6.

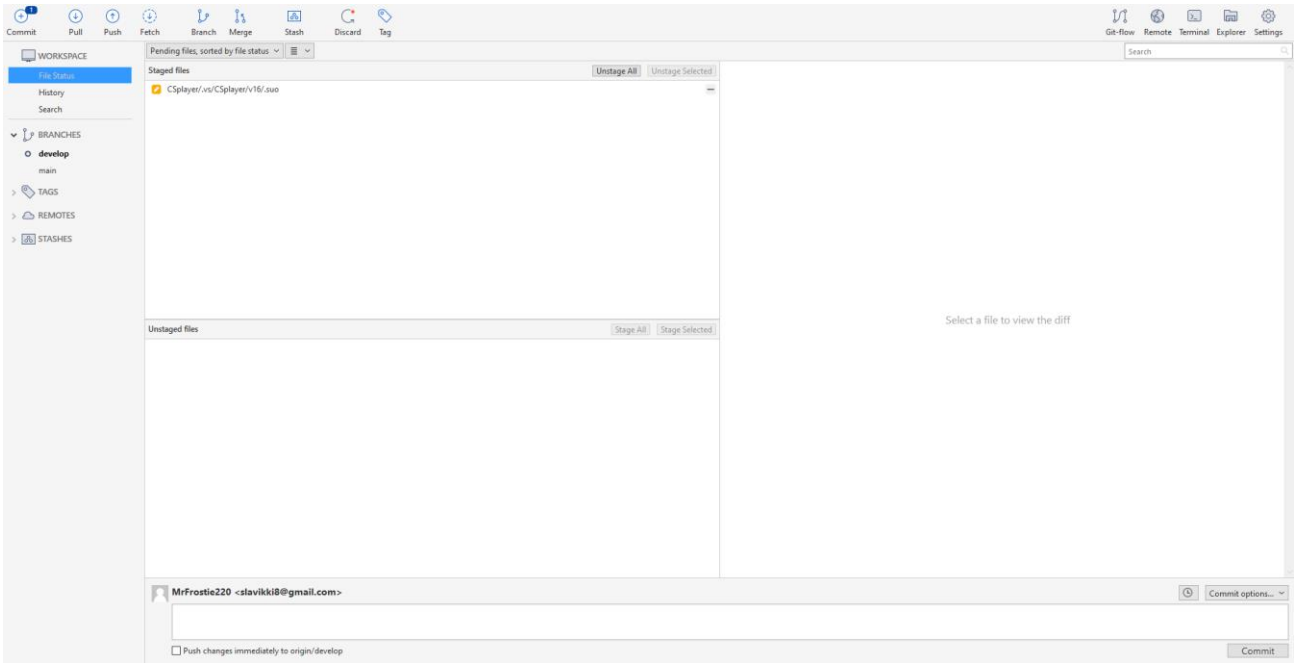


Рисунок 3.6 – Комміт внесених змін

Також даний додаток спрощує систему гілок гіта, оскільки, зазвичай, використовують 2 гілки – master та develop, виникає необхідність їх зливання одна з одною. При опису цього процесу консольними команда – зазвичай виникають труднощі. SourceTree – автоматизував цей процес рис.3.7.

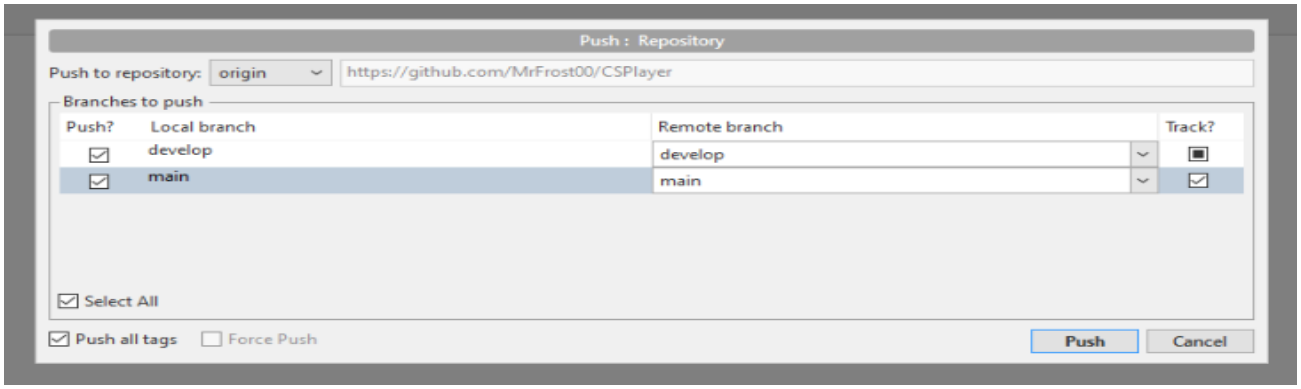


Рисунок 3.7 – Пуш та автоматичне зливання гілок

Також реалізована зручна візуалізація всього життєвого циклу процесу, яку можна відстежувати крок за кроком. В один клік можна побачити всі зміни внесені на потрібному пуші, та в один клік можна відкатити проект до потрібної версії без втрати змін, та лишніх проблем рис.3.8.

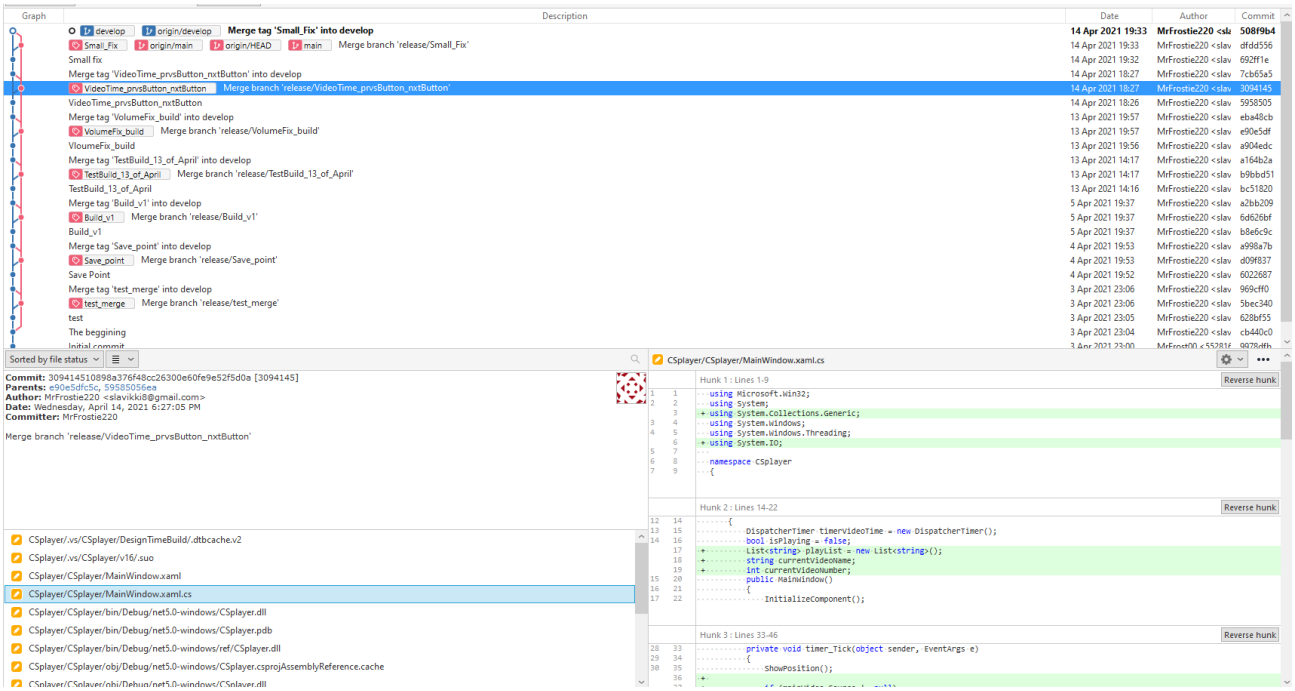


Рисунок 3.8 – Візуалізація життєвого циклу проекту

3.2.6 Abdobe Photoshop

Adobe Photoshop - це програма для редагування зображень та ретушування фотографій для використання на комп'ютерах з ОС Windows або MacOS.

Photoshop пропонує користувачам можливість створювати, вдосконалювати чи іншим чином редагувати зображення та ілюстрації. Зміна фонів, імітація картини з реального життя або створення альтернативного уявлення про Всесвіт - все це можливо в Adobe Photoshop. Це найбільш широко використовуваний програмний інструмент для редагування фотографій, маніпуляцій із зображеннями та ретуші для численних форматів файлів зображень та відео.

Інструменти в Photoshop дають можливість редагувати як окремі зображення, так і великі партії фотографій. Існує кілька версій Photoshop, включаючи Photoshop CC, Photoshop Elements, Photoshop Lightroom і Photoshop Express, версію Photoshop для iOS зі зменшеними функціями.

Це найкращий вибір для створення сучасного дизайну окремих елементів додатку.

Adobe Photoshop CC - це версія Creative Cloud Photoshop, доступна за

передплатою. Вважається версією продуктів професійного рівня сімейства Photoshop. Photoshop CC доступний разом із Photoshop Lightroom або як частина більшої передплати на Creative Cloud.

Photoshop CC - це вдосконалене програмне забезпечення для обробки зображень, яке використовується дизайнерами, веб-професіоналами, відеоредакторами та фотографами для зміни чи маніпулювання цифровими зображеннями. Photoshop в основному використовується для редагування 2D-зображень, хоча він пропонує деякі функції редагування 3D-зображень. Photoshop включає функцію аналізу зображень і може бути використаний для підготовки зображень до використання в Інтернеті або в друкованому вигляді.

Adobe Photoshop Elements - версія споживчого рівня сімейства продуктів Photoshop. Photoshop Elements містить багато професійних можливостей, які можна знайти в Adobe Photoshop CC, однак вони мають більш спрощені опції, розроблені з урахуванням користувача початкового рівня. Більш конкретно, він призначений для фотографів-любителів та любителів цифрової фотографії. Photoshop Elements побудований з використанням тієї самої основної технології цифрових зображень, що і Photoshop CC. До загальновживаних можливостей Photoshop Elements належать:

- Маніпулювання кольором зображення.
- Обрізання зображень.
- Виправлення недоліків, таких як пил на об'єктиві або червоні очі.
- Малювання на зображенні пером або олівцем.
- Додавання тексту до зображень.
- Видалення людей або предметів із зображення.
- Впорядкування фотографій для швидкого доступу.
- Публікація зображень в Інтернеті або надсилання електронною поштою.

3.2.7 Unity

Що таке Unity?

Ігровий рушій та IDE Unity - це 3D/2D ігровий рушій та потужна

кросплатформенна IDE для розробників.

Як ігровий рушій, Unity здатний забезпечити багато з найважливіших вбудованих функцій, які змушують гру працювати. Це означає такі речі, як фізика, 3D-рендеринг і виявлення зіткнень. З точки зору розробника, це означає, що немає необхідності винаходити велосипед.

Замість того, щоб починати новий проект, створюючи новий фізичний движок з нуля - розраховуючи кожен рух кожного матеріалу або те, як світло повинно відбиватися від різних поверхонь.

Що робить Unity ще більш потужним, так це те, що він також включає в себе процвітаючий "Магазин активів". Це, по суті, місце, де розробники можуть завантажувати свої творіння і робити їх доступними для спільноти.

Хочете гарний ефект вогню, але не маєте часу створювати його з нуля? Перевірте магазин ресурсів, і ви напевно щось знайдете. Хочете додати у свою гру елементи керування нахилом, не вдаючись до трудомісткого процесу точного налаштування чутливості? Напевно, для цього теж знайдеться ресурс! Все це означає, що розробник гри може зосередитися на тому, що має значення: на створенні унікального і цікавого ігрового процесу, кодуючи тільки ті функції, які є унікальними для цього бачення.

Що таке Unity IDE? Як і ігровий рушій, Unity - це IDE. IDE означає "інтегроване середовище розробки", що описує інтерфейс, який надає доступ до всіх інструментів, необхідних для розробки, в одному місці. Програмне забезпечення Unity має візуальний редактор, який дозволяє творцям просто перетягувати елементи в сцени, а потім маніпулювати їх властивостями.

Програмне забезпечення Unity також надає безліч інших корисних функцій та інструментів: наприклад, можливість переміщатися по папках в проекті або створювати анімацію за допомогою інструменту часової шкали.

Коли справа доходить до кодування, Unity перемикається на альтернативний редактор за вашим вибором. Найпоширенішим варіантом є Visual Studio від Microsoft, який здебільшого інтегрується без проблем. Яку мову використовує Unity? Рушій використовує C# для роботи з кодом і логікою, з цілою купою класів і API для Unity,

які вам потрібно буде вивчити. Хороша новина полягає в тому, що в Unity можна зробити дуже багато без необхідності працювати з великою кількістю коду. Тим не менш, розуміння того, як програмувати, створить набагато більше варіантів того, чого ви можете досягти, і Unity дає вам гнучкість, щоб змінити майже все.

На щастя, C# також є однією з найбільш дружніх до початківців мов програмування. І її варто вивчити, оскільки вона широко використовується в індустрії, а також має багато спільного з іншими популярними мовами, такими як C та Java.

Unity проти інших ігрових рушіїв

Звичайно, є й інші великі ігрові рушії, доступні для розробки. Ігровий рушій Unity стикається з жорсткою конкуренцією з боку таких рушіїв, як Unreal Engine та Cryengine. Саме тут Unity вступає в свої права як інструмент розробки. Хоча раніше це програмне забезпечення було відоме як "Unity 3D", воно стало настільки ж ефективним, як і інструмент 2D розробки. Мало того, спосіб роботи з графікою дозволяє дуже легко переносити досвід на більш низьке апаратне забезпечення.

Саме з цих причин Unity підтримує переважну більшість ігор в Google Play Store. Оскільки Unity є крос-платформним, це означає, що на ньому так само легко створювати ігри для iOS, ПК або навіть ігрових консолей. Unity також пропонує відмінну підтримку VR для тих розробників, які зацікавлені в розробці для Oculus Rift або HTC Vive, що є найважливішою частиною в моїй роботі. Отже, в чому Unity не такий хороший? Ну, в порівнянні з Unreal або Cryengine, Unity не настільки здатний створювати неймовірну графіку високого класу. Тим не менш, останні оновлення допомагають йому надолужити згаяне!

3.2.8 AWS

Amazon Web Services (AWS) - це комплексна платформа хмарних обчислень, що надається компанією Amazon. Перші пропозиції AWS були запущені в 2006 році для надання онлайн-послуг для веб-сайтів і клієнтських додатків. Наразі послуга доступна у 190 країнах світу.

Для мінімізації впливу простоїв та забезпечення стабільності системи, AWS

організовано за регіонами з різними центрами обробки даних. Ці регіони мають центральні хаби в Північній Америці (регіон США: Схід і Захід), Південній Америці (регіон Сан-Паулу, Бразилія), Європі/Близькому Сході/Африці (регіон ЄС: Ірландія, Франкфурт) та Азійсько-Тихоокеанському регіоні (Азіатсько-Тихоокеанський регіон і Китай).

У портфоліо AWS входить понад три десятки різноманітних веб-сервісів, у тому числі:

- Amazon Cloud Drive дозволяє користувачам завантажувати та отримувати доступ до музики, відео, документів та фотографій з пристроїв, підключених до Інтернету. Сервіс дозволяє користувачам транслювати музику та відео на своїх пристроях.
- Amazon CloudSearch - це масштабований пошуковий сервіс, який зазвичай використовується для інтеграції користувацьких можливостей пошуку в додатки.
- Amazon Dynamo Database (також DynamoDB або DDB) - це повністю керований сервіс баз даних NoSQL, який пропонує низьку затримку та високу масштабованість.
- Amazon Elastic Compute Cloud 2 (EC2) - це веб-сервіс, який дозволяє бізнес-клієнтам запускати додатки на масштабованих обчислювальних потужностях. EC2 слугує практично необмеженим набором віртуальних машин (ВМ).
- Amazon ElastiCache надає повністю керовану послугу кешування, яка сумісна по протоколу з Memcached, системою кешування з відкритим вихідним кодом для прискорення роботи динамічних додатків за рахунок зменшення навантаження на базу даних. Крім того, ElastiCache пропонує відкриту базу даних в пам'яті Redis, яка підтримує відсортовані набори даних і списки.
- Amazon Mechanical Turk - це сервіс для розробників для доступу до ресурсу "людського інтелекту" при створенні своїх додатків. API дозволяють інтегрувати людський інтелект у так звані віддалені виклики процедур (RPC), використовуючи мережу людей для виконання завдань, для яких комп'ютери не пристосовані.

- Amazon Redshift - це сервіс сховища даних петабайтного масштабу, призначений для аналітичних робочих навантажень, що взаємодіють зі стандартними SQL-клієнтами та інструментами бізнес-аналітики (BI).
- Amazon Simple Storage Service (S3) - це масштабоване сховище, яке може зберігати та отримувати будь-які обсяги даних. Послуга використовується для резервного копіювання та архівування даних і додатків.

3.2.9 SQL Databases

Абревіатура SQL розшифровується як Structured Query Language (мова структурованих запитів) і описує мову спілкування з реляційними базами даних. Команди SQL дозволяють відносно легко вставляти, змінювати або видаляти дані. Основні можливості мови SQL

Мова баз даних SQL характеризується простим синтаксисом. Вона базується на реляційній алгебрі і семантично складається в основному з елементів англійської мови. Мова SQL є стандартизованою і може використовуватися на платформах з багатьма системами баз даних. Існують різні діалекти SQL, які перешкоджають стовідсотковій сумісності.

Більшість існуючих систем управління базами даних мають SQL-інтерфейси і дозволяють отримати доступ до збережених даних за допомогою універсальної мови баз даних SQL. Фактичні SQL запити можуть бути виконані з невеликими зусиллями, оскільки для більшості операторів потрібно знати лише кілька різних команд.

Різні категорії команд SQL Крім команд для створення, редагування і видалення даних, SQL має елементи для визначення структур даних і для запитів до наборів даних.

В основному, можна розрізняти три різні категорії команд SQL. Це такі категорії:

- DML-команди (Data Manipulation Language - мова маніпулювання даними)
- DDL-команди (Data Definition Language - мова визначення даних)
- DCL-команди (Data Control Language - мова управління даними)

У той час як DML-команди призначені для редагування, вставки або видалення даних або для доступу до бази даних на читання, DDL-команди можуть бути використані для визначення схеми бази даних. Нарешті, команди DCL використовуються для управління індивідуальними правами або для контролю транзакцій.

Запити даних з бази даних вводяться командою "SELECT", за якою слідують подальші команди типу "FROM" або "WHERE". У контексті запиту, SQL дозволяє виводити результат у вигляді таблиці, і ці дані можуть бути повторно використані або відредаговані як нова таблиця.

Приклади команд SQL в різних категоріях Нижче наведено кілька прикладів команд з різних категорій SQL. Важливі команди DML

- SELECT FROM… для зчитування даних з таблиць.
- DELETE FROM… для видалення даних з таблиць.
- INSERT INTO… для вставки даних в таблиці.
- CREATE TABLE… для створення таблиць.
- ALTER TABLE… для зміни таблиць.

Мова баз даних SQL не є повноцінною мовою програмування. Тому за допомогою самих команд SQL не можна створювати повноцінні програми. Однак SQL можна дуже добре поєднувати з іншими мовами програмування або вбудовувати в програми. Для цього використовуються різні методи, такі як вбудований SQL, програмні інтерфейси або фреймворки.

У командах SQL розрізняють динамічний і статичний SQL. У той час як статичні оператори SQL фіксуються під час компіляції програми, динамічні оператори SQL відомі тільки безпосередньо під час виконання програми. Це означає, що оператори SQL повинні інтерпретуватися безпосередньо системою баз даних під час виконання програми.

3.2.10 OpenVR

Комплект для розробки програмного забезпечення та інтерфейс прикладного

програмування від Valve

OpenVR - це комплект розробки програмного забезпечення (SDK) та інтерфейс програмування, розроблений компанією Valve для підтримки SteamVR (HTC Vive) та інших девайсів віртуальної реальності (гарнітури віртуальної реальності). Платформа SteamVR використовує його як стандартний API (Application Programming Interface) і середовище виконання, що дозволить нам інтегрувати додаток в Steam, оскільки він служить інтерфейсом між апаратним і програмним забезпеченням VR і реалізований SteamVR.

Хоча OpenVR є SDK за замовчуванням для HTC Vive, він розроблений для підтримки декількох постачальників. Наприклад, розробник може створювати функції тригерних кнопок на базі OpenVR для контролерів Oculus Rift або Windows MR, оскільки обидві системи підтримуються SDK.

Компанія Valve оголосила, що буде співпрацювати з проектом Open Source Virtual Reality (OSVR), хоча масштаби співпраці поки неясні.

Ще один чудовий приклад ідеї «записуй один раз, грай будь-де» в контексті VR — це WebVR.

WebVR — це базований на веб-стандартах API для створення можливостей WebVR, відомих в Інтернеті, які можна розповсюджувати та запускати з використанням URL - адреси. Чи використовуєте вивисокоякісну гарнітуру віртуальної реальності, таку як Oculus, Vive, одну з гарнітур Microsoft Mixed Reality, мобільну гарнітуру віртуальної реальності, таку як Gear VR або Daydream Viewer, або одну з автономних гарнітур, такі як Oculus Go та Lenovo Mirage Solo, або якщо вони містять браузер із підтримкою WebVR, наприклад, смартфон або планшет. Ви також можете розміщувати програми AR, як частину того ж API завдяки розширенню під назвою WebXR.

3.2.11 Microsoft Visual Studio

Інтегроване середовище розробки Microsoft Visual Studio підтримує множину

передових мов програмування. До них відносяться Visual Basic, C, C++, C#, SQL Server, PHP та Python. Visual Studio також чудово підходить для розробки з використанням Javascript, HTML та CSS. Крім власних програм Win32/Win64, Visual Studio може розробляти програми Windows, динамічні веб-служби або веб-сайти, а також програми для .NET Framework. Розробка та впровадження мобільнихпрограмдля Android, iOS та Windows Phone також можуть виконуватися за допомогою інструментів, що надаються Xamarin, дочірньою компанією Microsoft. Можливості Visual Studio Середовище розробки Visual Studio має багато корисних функцій. Наприклад, редактор має інтерактивну довідку, яка реагує на поточну позицію курсора.

Крім того, ключові слова виділяються кольором у вихідному тексті. На додаток доавтоматичної перевіркисинтаксисуредактор Visual Studio також має інструмент IntelliSense, який автоматично додає методи та функціїумірувведення вихідного тексту.

Крім того, середовище розробки має графічні інтерфейси для інтеграції бібліотек .NET та ActiveX, а також веб-сервісів.

Крім того, так званий "Server Explorer" забезпечує доступ до зовнішніх джерел даних, таких як Microsoft SQLServer. Також інтегровані різні редактори WYS та WYG, які дозволяють,наприклад,розроблятиінтерфейсидлявеб-додатків або додатків Windows.

Кількість мов,які підтримуються Visual Studio, за останні роки значно зросла. Насамперед це пов'язано з власною сертифікаційною програмою Microsoft, яка дозволяє стороннім постачальникам додавати до Visual Studio інші мови програмування. Прикладами цього єEiffel, Delphi, Prolog чи Visual COBOL. Щоб ви могли розробляти програми ASP.NET, версії Visual Studio 2005 тановіші версії включають вбудований веб-сервер, тому вам не потрібно окремо встановлювати служби Microsoft Internet Information Services (IIS).

Для розробки програм середовище IDE надає користувачам різні редактори. До нього входять традиційні текстові редактори для мов програмування, що

підтримуються Visual Studio,а також редактори для розробки графічних інтерфейсів. Крім того, для веб-сторінок XML Schema(XSD) та HTML доступні як текстові, так і графічні редактори. Те саме стосується бінарних діаграм або діаграм класів і послідовностей. Завершує пропозицію редактора для розробки інсталяторів додатків.

3.2.12 NodeJS

Мова програмування Javascript дозволяє гнучко проектувати свій сайт і динамічно адаптувати ваш контент.

Типовим прикладом цього є повзунок або код, який підганяє контейнер під кінцевий пристрій клієнта, щоб забезпечити оптимальне відображення. Крім того, ви можете перевірити введення користувача за певною схемою. Це гарантує, що поля електронної пошти надсилаються лише з правильними даними.

Як і в будь-якій мові програмування, у вас є засоби для керування сайтом за допомогою логіки та умов. Але Javascript може набагато більше. Ви можете запропонуват исвоїм клієнтам не тільки прості програми, такі як живі лінії, що біжать, але і дуже складні програми і невеликі 2D/3D ігри.

Карти Google – чудовий прикладтого, наскільки потужним може бути Javascript. Майже всі веб-розробники використовують JavaScript для надання різних стилів CSS елементам DOM (об'єктна модель документа) та для вставлення додаткового контенту до списків.

DOM – це інтерфейс між вашим сценарієм та елементами вашого сайту. Всі зображення, текст, таблиці та їх стилі доступні та можуть бути змінені через DOM. Наприклад, натискання кнопки може змінити колірну схему зі світлою на темну або динамічно адаптувати меню до кінцевого пристрою.

Javascript зазвичай говорить про виконання коду на стороні клієнта (також відомий як зовнішній інтерфейс), оскільки він виконується браузером на вашому комп'ютері або смартфоні після завантаження HTML та CSS.

Node.js – це середовище виконання Javascript, незалежно від хост-програм, таких як веб-браузери.

Це означає, що Javascript також може працювати на стороні сервера (так званим бекендом) і може використовуватися для розробки сценаріїв, інструментів та веб-застосунків на стороні сервера. Це також робить Javascript привабливим для бекенд-розробників.

В іншому випадку такі програми вимагають знання інших мов програмування, таких як PHP, Python, Ruby, ASP.NET. За допомогою Node.js ви отримуєте можливість уніфікувати свою веб-розробку та використовувати одну мову програмування як для інтерфейсу, так і сервера рис 3.8.

Це має багато переваг. Javascript — одна з найпростіших мов для вивчення Ви можете використовувати ті самі угоди про імена для зовнішнього інтерфейсу та внутрішнього інтерфейсу, щоб створити чисту та узгоджену кодову базу.

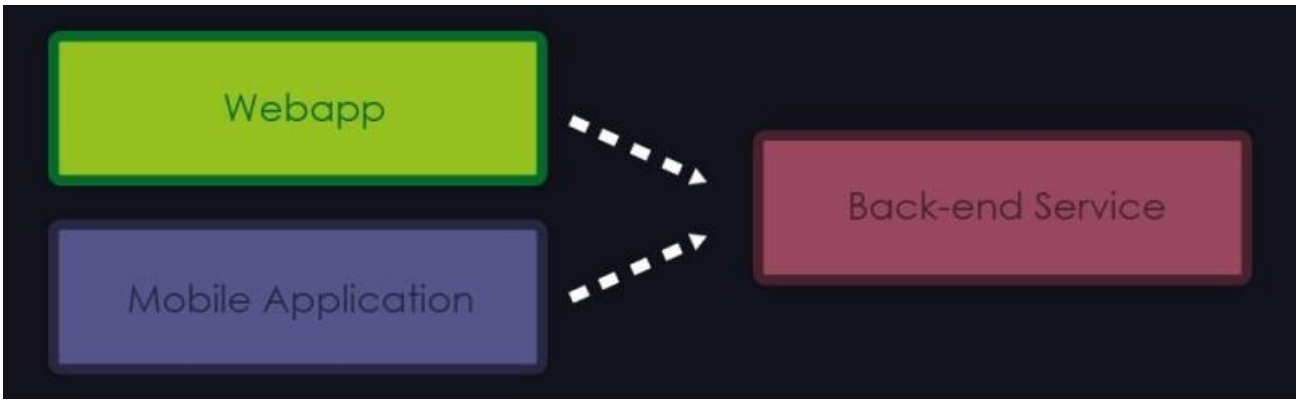


Рисунок 3.8 – Схема роботи NodeJS

Так щоробить Node.js таким особливим? З одного боку, Javascript - одна з найпростіших мов для вивчення, з низьким порогом входу. Крім того, Node.js має високу масштабованість і досить швидку обробку даних. Це з тим, що Node.js може працювати як асинхронно, і паралельно. Ключовим словом є неблокуючий введення - висновок, тому що функція, що виконується, неповинна чекати завершення дії.

Звичайний Javascript насправді є однопоточним і обробляє дані послідовно. Це означає, що одночасно може виконуватися лише одна дія. Поки виконується ця дія, інші компоненти очікують на завершення. Натомість він надає асинхронну систему подій, що дозволяє виконувати паралельну роботу без використання складних

багатопотокових архітектур. Node.js використовує функції зворотного дзвінка, щоб повідомити функцію про їїзавершення. При цьому паралельно можуть вирішуватись інші завдання.

Крім продуктивності, він поставляється з менеджером пакетів (npm - Node Package Manager), який пропонує величезну (якщо не найбільше) кількість пакетів з відкритим вихідним кодом. Він доступний безкоштовно, тому вам не доведеться починати заново.

Таким чином, ви можете розпочати безпосередньо з основної логіки вашогосервісу. За допомогою npm ви можете знаходитимодулі та всі їх незалежності, встановлювати їх, підтримувати та знову видаляти.

3.3 Кроки розробки

Першим етапом буде створення навігаційної площини, для тестування розробленної моделі. Можна використати кубічні примітиві і розігнати їх під потрібні розміри, щоб були моделі сходів\площин\переходів.

Далі, йде створення перших скриптів для обробки поведінки руху і додати його в об’єкт VR Rig. Також, буде використовуватись стандартні бібліотеки Unity, рис 3.9

```
using UnityEngine;
using UnityEngine.XR;
using UnityEngine.XR.Interaction.Toolkit;
```

Рисунок 3.9 – Бібліотеки Unity для VR

Далі, потрібно створити загальнодоступну змінну, яка буде зберігати всі вхідні данні від користувача, за допомогою контролеру. Також, можно використовувати клас Device Characteristics, але, оскільки, нам потрібня своя модель руху, ми її використовувати не будемо. Присвоємо їй значення LeftHand, для тестів рис 3.10

Рисунок 3.10 – Зчитувач руху з контролеру

Далі, треба створити метод типу Update, який буде обробляти всі рухи контролеру від користувача рис 3.11

```
void Update()
{
    InputDevice device = InputDevices.GetDeviceAtXRNode(inputSource);
    device.TryGetFeatureValue(CommonUsages.primary2DAxis, out inputAxis);
}
```

Рисунок 3.11 – Метод апдейт для контролеру

Далі, потрібно оброблювати метод Start і в тому ж самому класі створити метод FixedUpdate, для управління рухом. Цей метод буде використовуватись як метод наших фізичних розрахунків. Також, буде додаватись серіалізовані змінні швидкості рис. 3.12

```
private CharacterController character;
[SerializeField]
private float speed = 1f;

Unity Message | 0 references
void Start()
{
    character = GetComponent<CharacterController>();
}
Unity Message | 0 references
private void FixedUpdate()
{
    Vector3 direction = new Vector3(inputAxis.x, y: 0, inputAxis.y);
    character.Move(direction * speed * Time.deltaTime);
}
```

Рисунок 3.12 – Обробка руху користувача

Щоб була можливість тестувати, потрібно додати компонент Character Controller. Це дасть можливість завантажити тести і ми маємо базовий функціонал руху.

Далі, є можливість підставити клас розробленої моделі, в замість базового класу руху, який є в Unity і ми будемо мати можливість тестувати розробленний метод в тестовій середі.

Далі, потрібно реалізувати гравітацію, адже це буде важлива частина симулції. Для реалізації гравітації потрібно створити 2 змінні – сила тяжіння та швидкість падіння, а далі, використовуємо звичайну форулу розрахунку прискорення сили тяжіння. В Unity, ми можемо використати за допомогою SphereCast, який є кращим варіантом реалізації фізики гравітації. Також, потрібно створити об’єкт типу “Sphere” і використовувати його для зберігання наших “заземлених” об'єктів, щоб SphereCast міг розрахувати на предмет того, чи ми заземлені і на якому об’єкті ми знаходимось рис 3.13.

```
bool CheckIfGrounded()
{
    //indicates we are on the ground
    Vector3 rayStart = transform.TransformPoint(character.center);
    float rayLength = character.center.y + 0.01f;
    bool hasHit = Physics.SphereCast(rayStart, character.radius, Vector3.down, out RaycastHit hitInfo, rayLength, groundLayer
    return hasHit;
}
```

Рисунок 3.13 – Розрахунок заземлення

```
private void FixedUpdate()
{
    Quaternion headYaw = Quaternion.Euler(x: 0, rig.cameraGameObject.transform.eulerAngles.y, z: 0);
    Vector3 direction = headYaw * new Vector3(inputAxis.x, y: 0, inputAxis.y);
    character.Move(direction * speed * Time.deltaTime);

    bool isGrounded = CheckIfGrounded();
    if (isGrounded)
    {
        fallingSpeed = 0;
    }
    else
    {
        fallingSpeed += gravity * Time.fixedDeltaTime;
    }

    character.Move(Vector3.up * fallingSpeed * Time.fixedDeltaTime);
}
```

Рисунок 3.14 – Оновленний клас заземлення

Далі, потрібно додати 6 ступенів свободи управління, прив’язавши всі рухи у фізичному прості до рухів персонажу. Для цього, потрібно вказати контролеру персонажу відповідати і слідувати за його рухами. Для цього потрібно створити нову функцію “CapsuleFollowHeadset”, щоб зв’язати їх. Там буде розраховуватись простір між рівнем очей і вершиною голови персонажа. Тобто, ми використаємо формулу, яку описували в другому розділі, про розташування віртуальної камери рис 3.15

```
void CapsuleFollowHeadset()
{
    character.height = rig.cameraInRigSpaceHeight + heightOffset;
    Vector3 capsuleCenter = transform.InverseTransformPoint(rig.cameraGameObject.transform.position);
    character.center = new Vector3(capsuleCenter.x, character.height / 2 + character.skinWidth, capsuleCenter.z)
}
```

Рисунок 3.15 – Реалізація CapsuleFollowHeadset

Далі, треба вдосконалити поведінку фізики руху, відповідно до можливостей VR-гарнітур, щоб запобігти ефекту “тремтіння”, як було описано в другому розділі.

Тепер, у нас буде можливість використовувати три методи для переміщення по тестовій сцені, а саме:

- Тригер телепорту.
- Переміщення в реальному просторі.
- І вісь тачпада.

Це все можно використовувати одночасно, для тестів.

Останнє, що нам потрібно додати, для покращення якості життя тестової і дев сцен – це додавання можливості для гравця обертати положення персонажа, не рухаючи голову в реальному просторі. Це потрібно, для можливості, використовувати симуляцію сидячі за столом.

В Unity є вже реалізованний компонент, який дозволяє увімкнути цю функцію - Snap Turn Provider (Device-Based).

Не просто комбінуйте готові складні елементи, подібні до тих, які в досталь

доступні в UnityAssetStore.

Ці потужні елементи забезпечують безліч візуальних ефектів, але проста їх вставка та вибір рідко мають сенс, оскільки не буде зрозуміла їх складна структура для подальшого розуміння і розробки своїх елементів. Це також полегшує розуміння запрограмованого ігрового процесу. Крім того, багато багатогранних ігрових персонажів можуть бути створені тільки за допомогою зовнішніх програм, а потім повинні бути імпортовані в Unity.

Рекомендується, спочатку, розробити двовимірну гру (2D-гру). При цьому він вже працює з багатьма елементами Unity, які будуть потрібні для розробки тривимірних ігор (3D-ігор). У 2D-ігри грають виключно на плоскій поверхні екрану. Ясніше та простіше для розуміння, ніж 3D-ігри.

Таким чином, ви можете зосередитися на розробці ігор, а не займатися одночасно орієнтацією, обертанням і перспективою в 3D.

4 ПРОЕКТУВАННЯ

Всі внутрішні компоненти програми діляться на блоки, відповідно до призначення і типу рис.4.1

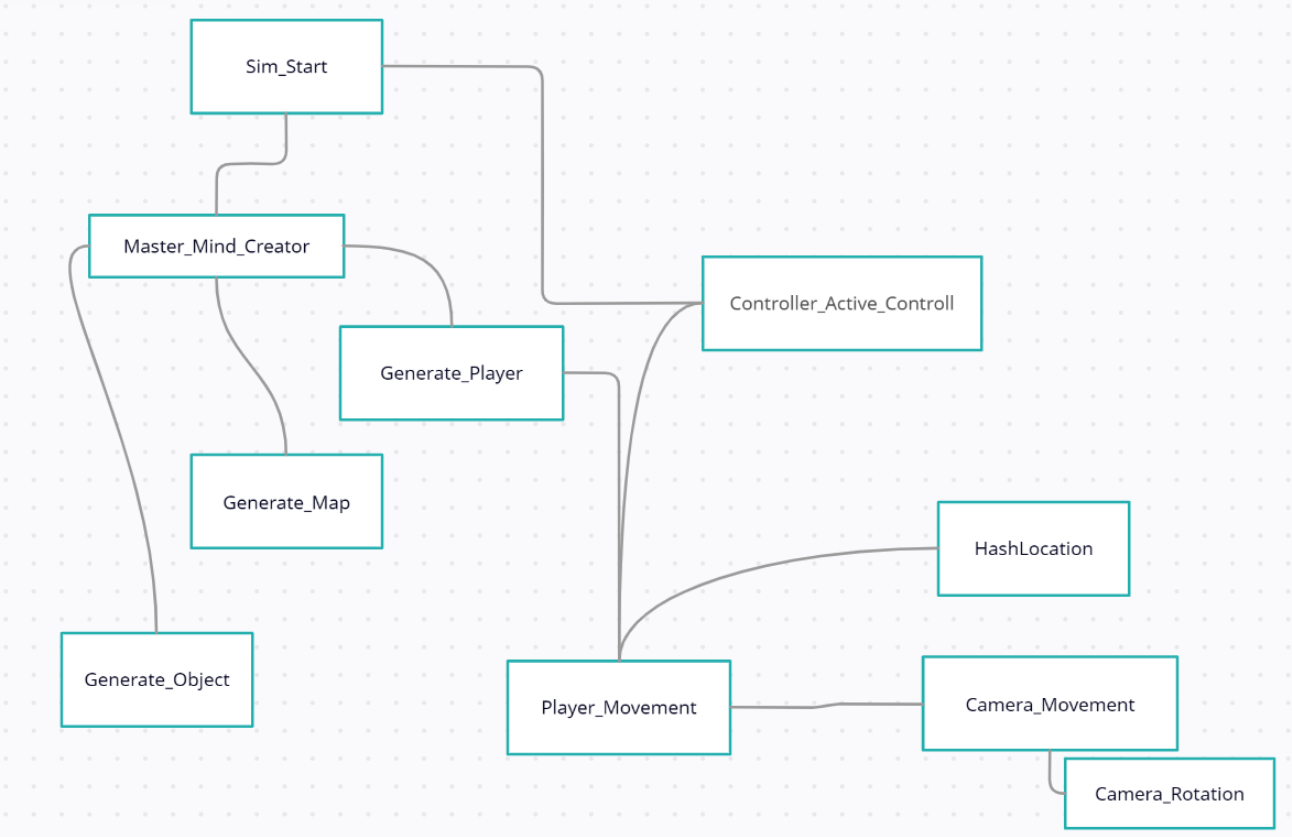


Рисунок 4.1 – Схема внутрішніх компонентів додатку

В даному додатку реалізовані основні патерни проектувння, для покращення якості роботи.

1. Single Responsibility Principle – кожен метод відповідає за 1 функціонал.
2. Template Method – шаблонний метод, який дозволяє наслідникам перевизначати кроки алгоритму, не змінюючи його.
3. Mediator – посередник, який дозволяє об’єктам взаємодіяти через певний об’єкт – посередник.

Інтерфейс додатку розроблений за принципом KISS (keep it short and simple), тобто, для кінцевого користувача він зрозумілий та очевидний.

Принцип KISS можна описати наступними пунктами:

- Не використовуйте складну логіку, якщо це можна зробити просто.
- Не бійтеся викидати код. Під час рефакторингу коду завжди видаляйте непотрібний код.
- Вирішіть проблему, а потім закодуйте її. Перш за все витратьте час, щоб зрозуміти проблему та способи її вирішення. Тільки після цього стрибка вирішити це. Розбийте свої завдання на підзадачі та вирішуйте одне за іншим, роблячи їх простими.
- Не використовуйте техгології без необхідності.

Даний додаток можна представити у вигляді UML-діаграми рис.4.2.

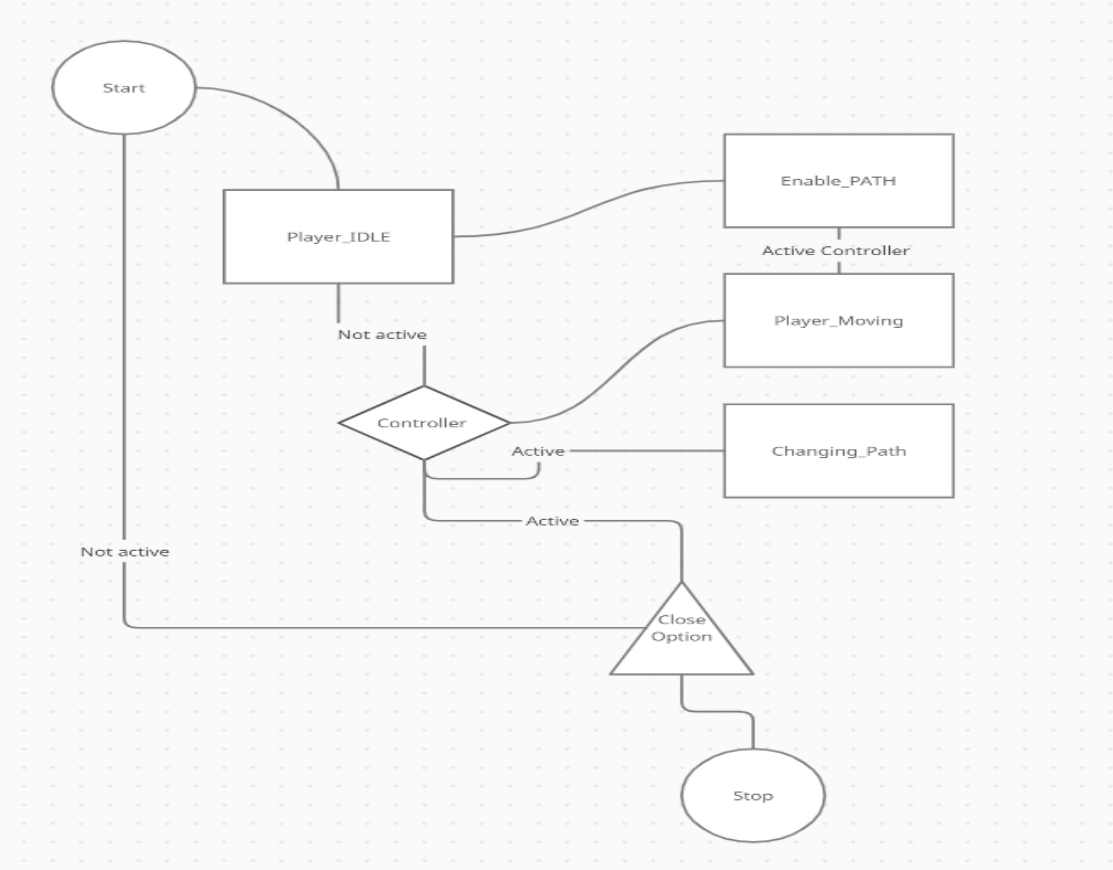


Рисунок 4.2 – UML-діаграма

5. ТЕСТУВАННЯ

Створений застосунок має працювати чітко та без збоїв, з метою забезпечення

належної якості, для тестування застосунку було створено набір фундаментальних тестів.

Тест-кейс – це артефакт, який описує сукупність кроків, умов та параметрів, необхідних для перевірки реалізації функції, що тестується або її частини.

Тест №1.

Взаємодія користувача з VR-інтерфейсом.

Вхідні дані: користувач, за допомогою, контролера намагається рухатись по тестовій середі.

Очікуваний результат: користувач має можливість рухатись по тестовій середі.

Отриманий результат: користувач має можливість рухатись по тестовій середі.

Тест пройдено успішно.

Тест №2

Стресс тест з використанням нової моделі.

Вхідні дані: користувач запускає скрипт, для запуску стрес тесту при мінімальних навантаженнях.

Очікуваний результат: навантаження на відеокарту та процесор не більше 40%.

Отриманий результат: рис 5.1

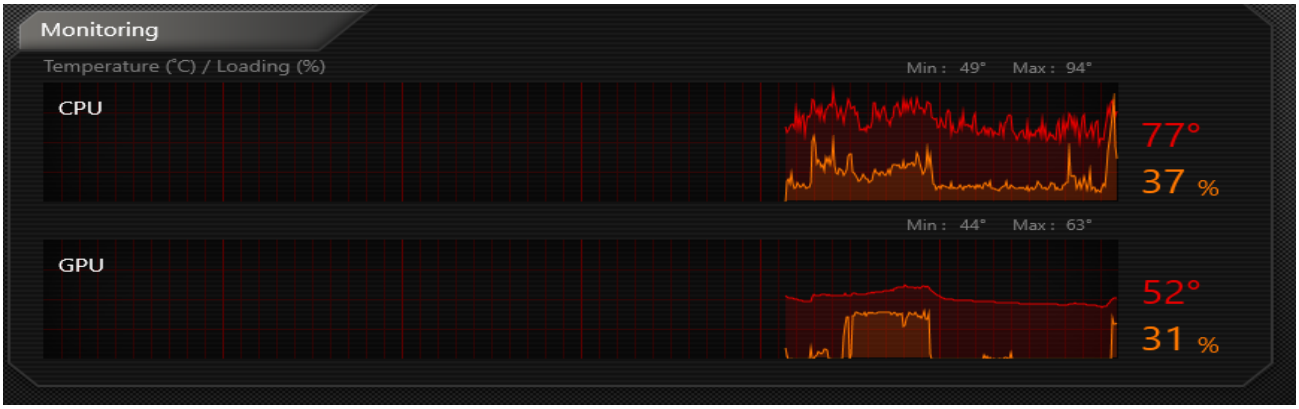


Рисунок 5.1 – Результати стрес тести при мінімальних навантаженнях

Тест пройдено успішно.

Тест №3

Стресс тест з використанням нової моделі.

Вхідні дані: користувач запускає скрипт, для запуску стрес тесту при максимальних навантаженнях.

Очікуваний результат: навантаження на відеокарту та процесор не більше 75%.

Отриманий результат: рис 5.2

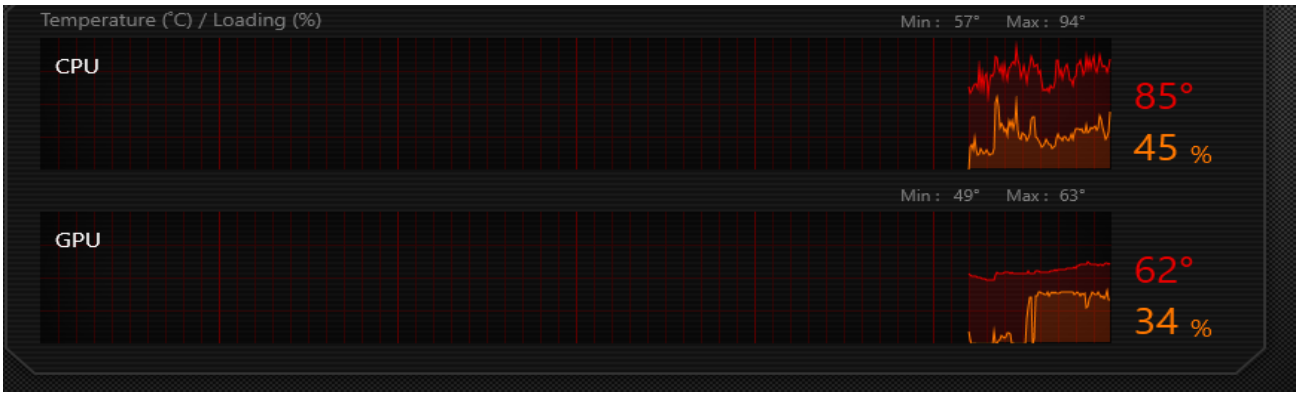


Рисунок 5.2 – Результати стрес тесту при максимальних навантаженнях

Тест пройдено успішно.

ВИСНОВКИ

- Проаналізовано існуючі аналоги і проблеми, які виникають під час розробки і використанні додатків, які підтримують режим віртуальної реальності.
- Розроблено модель руху використовуючи VR технології.
- Проведено тестування програмного забезпечення.
- Проведено моделювання руху актора в симуляції VR.
- Встановлено, що при використанні розробленного методу,за результатами кількох тестів, навантаження на відеокарту зменьшуеється від 3% до 10%, а на процесор від 5 до 15%, ніж при використанні аналогів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Mark J. Price C# 7 and .NET Core: Modern Cross-Platform Development - Second Edition/- Packt Publishing– с. 96 – 120.
2. Ben Albahari, Joseph Albahari C♯ 6.0: Pocket Reference /- O'Reilly Media; 1st edition - с. 117 – 123.
3. ANDREW TROELSEN, Philip Japikse, C# 6.0 and the .NET 4.6 Framework /- Apress, 2015 – с. 1083.
4. Joseph Albahari, Ben Albahari, C# 7.0 in a Nutshell: The Definitive Reference /- "O'Reilly Media, Inc.", 2017 – с. 120.
5. Griffiths I. Programming C# 8.0: Build Cloud, Web, and Desktop Applications / – "O'Reilly Media Inc.", 2019 – с. 140.
6. Коноваленко І.В. Програмування мовою C# 6.0 / – Тернопіль, ТНТУ, 2016. с. 227.
7. Dialog boxes overview (WPF .NET). [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://docs.microsoft.com/en-us/dotnet/desktop/wpf/windows/dialog-boxes-overview?view=netdesktop-5.0 Дата звернення: 07.04.2021.
8. Git Branching. [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://git-scm.com/book/en/v2/Git-Branching-Branches-in-a-Nutshell Дата звернення: 07.04.2021.
9. VR-ТЕХНОЛОГІЇ ЯК МЕТОД І ЗАСІБ НАВЧАННЯ. // Юлія Трач // Освітологічний дискурс, 2017, № 3-4 (18-19). 2017.
10. ВИКОРИСТАННЯ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ В ОСВІТНЬОМУ ПРОЦЕСІ // Олексюк Василь Петрович, Ковальчук Олена Юріївна// 2022.
11. Карабін О. Й. Використання доповненої реальності у підготовці майбутніх вчителів інформатики в умовах дистанційного навчання Вісник Запорізького національного університету : збірник праць. Педагогічні науки. Запорізький національний університет, 2020. № 3(36), ч. II. С. 68–72.

12. Віртуальна та доповнена реальність: як нові технології надихають вчитися – освітній блог | «Освіторія». Освіторія. URL: https://osvitoria.media/opinions/virtualna-ta-dopovnena-realnist-yakoyumozhe-buty-suchasna-osvita/.
13. ПСИХОЛОГО-ПЕДАГОГІЧНІ АСПЕКТИ ПОБУДОВИ ОСВІТНЬОГО ПРОЦЕСУ В УМОВАХ ЦИФРОВІЗАЦІЇ ОСВІТИ // Винничук Олег Теофілович, Садовник Владислав Олегович // 2022.
14. N. Morse, O. Glazunova, «Models of effective use of information-communication and distance learning technologies at higher educational institutions», Information technologies and teaching aids. vol. 2, issue 6. p. 134-148. 2008.
15. O. Didenko, D. Kuprienko, «Electronic journal to record students' progress (cadets, students) as a means of rationalization of the educational process», Information Technologies and Learning Tools, vol. 47, issue 3, p. 110-123. 2015.
16. Oleksiuk V., Oleksiuk O. Exploring the potential of augmented reality for teaching school computer science. Proceedings of the 3rd international workshop on augmented reality in education (AREdu 2020). URL: http://ceur-ws.org/Vol-2731/paper04.pdf.
17. VR-Technology in Teaching: Opportunities and Challenges. Caroline Graeske, Sofia Aspling Sjöberg, 2021.
18. I. Bloshchynskyi, «Usage of Anki specialised program application during future Border Guard officers' independent foreign lanquage professional training for passing state examination, Information Technologies and Learning Tools, vol. 58, issue 2, p. 49-58, 2017. [Online]. Available: https://journal.iitta.gov.ua/index.php/itlt/article/view/1605. July 3, 2018.
19. V. Bykov, Models of Organizational Systems of Open Education. Kyiv: Atika, 2008.
20. V. Shtilveld, «Distance education – from theory to practice», in topical network seminar [Online]. Available: http://www.osvita.org.ua/distance/articles/15/. March 10, 2018.

ДОДАТОК

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра інженерії програмного забезпечення

“Розробка симуляції для навчання на основі VR технології”

Виконав: Студент групи ПДМ-61 Кисельов В.О.

Керівник: к.т.н., доцент Негоденко О.В.

Київ - 2022

МЕТА, ОБ’ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: підвищити ефективність практичної складової навчання за допомогою віртуальної реальності

Об’єкт дослідження: процес та механізми функціювання технології VR у галузі навчання.

Предмет дослідження: технології віртуальної реальності

АНАЛІЗ ІСНУЮЧИХ ІТ РІШЕНЬ ТА ЇХ МОДЕЛЕЙ

Модель	Додаток
Оболонка моделі Asynchronous reprojection на основі двигуна Unreal Engine 3.0	Microsoft Flight Simulator
Оболонка моделі Asynchronous Timewarp на основі двигуна CryEngine	AHTS Ship Simulator
Оболонка моделі Asynchronous Timewarp 2.0 на основі двигуна Unreal Engine 4.0	Human Anatomy VR

Математична модель розміщення камери гравця у віртуальному світі

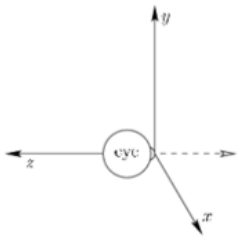
Для того, щоб віртуальна реальність працювала правильно, "камера" повинна, фактично бути одним з двох віртуальних людських очей, поміщених у віртуальний світ.

Припустимо, що око - це об'єктна модель R, яку ми хочемо помістити у віртуальний світ в деяке положення $E = (e1, e2, e3)$ і напрям заданий наступною матрицею (2):

$$R = \begin{matrix} x1 & y1 & z1 \\ x2 & y2 & z2 \\ x3 & y3 & z3 \end{matrix} \quad (2)$$

Замість того, щоб переміщати око у віртуальному світі, нам потрібно перемістити всі моделі у віртуальному світі до системи відліку ока. Це означає, що нам потрібно застосувати зворотне перетворення що призводить до відповідної матриці (3):

$$T = \begin{matrix} x1 & x2 & x3 & 0 \\ y1 & y2 & y3 & 0 \\ z1 & z2 & z3 & 0 \\ 0 & 0 & 0 & 1 \end{matrix} \rightarrow \begin{matrix} 1 & 0 & 0 & -e1 \\ 0 & 1 & 0 & -e2 \\ 0 & 0 & 1 & -e3 \\ 0 & 0 & 0 & 1 \end{matrix} \quad (3)$$



Модель віртуального ока

Математична модель руху об'єктів в VR

4

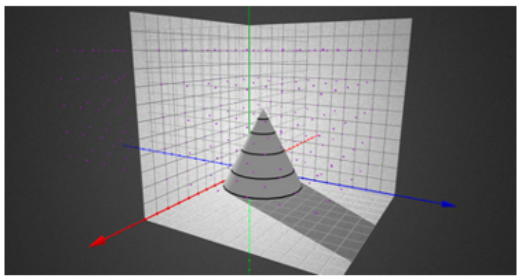
Розглянемо 3D трикутник, в якому координати його вершин виражені через задані узагальнені константи (1):

$$((x1, y1, z1), (x2, y2, z2), (x3, y3, z3)) \quad (1)$$

Нехай x_t, y_t і z_t - величини, на які ми хочемо змінити положення трикутника вздовж осей x, y і z відповідно (2):

$$\begin{aligned} (x1, y1, z1) &\longrightarrow (x1 + xt, y1 + yt, z1 + zt) \\ (x2, y2, z2) &\longrightarrow (x2 + xt, y2 + yt, z2 + zt) \\ (x3, y3, z3) &\longrightarrow (x3 + xt, y3 + yt, z3 + zt) \end{aligned} \quad (2)$$

А оскільки кожен об'єкт це сітка трикутників, то достатньо застосувати перетворення тільки до вершин. Всі трикутники збережуть свої розміри і форму



Кожен об'єкт в 3Д просторі - це сітка трикутників

Математична модель руху об'єктів в VR

4

Тепер обертання об'єкту - одним з найпростіших способів параметризації тривимірних обертань є побудова їх з ``2D-подібних" перетворень використовуючи метод "Yaw, Pitch, Roll"

"**Roll**" називається обертання γ проти годинникової стрілки навколо осі Z

$$Rz(\gamma) = \begin{matrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{matrix} \quad (1)$$

"**Pitch**" це обертання **β** проти годинникової стрілки навколо осі X.

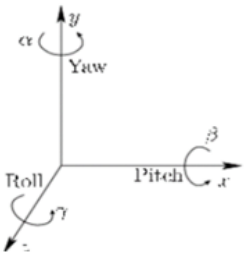
$$Rx(\beta) = \begin{matrix} 1 & 0 & 0 \\ 0 & \cos\beta & -\sin\beta \\ 0 & \sin\beta & \cos\beta \end{matrix} \quad (2)$$

"**Yaw**" це обертання α проти годинникової стрілки навколо осі Y

$$Ry(\alpha) = \begin{matrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{matrix} \quad (3)$$

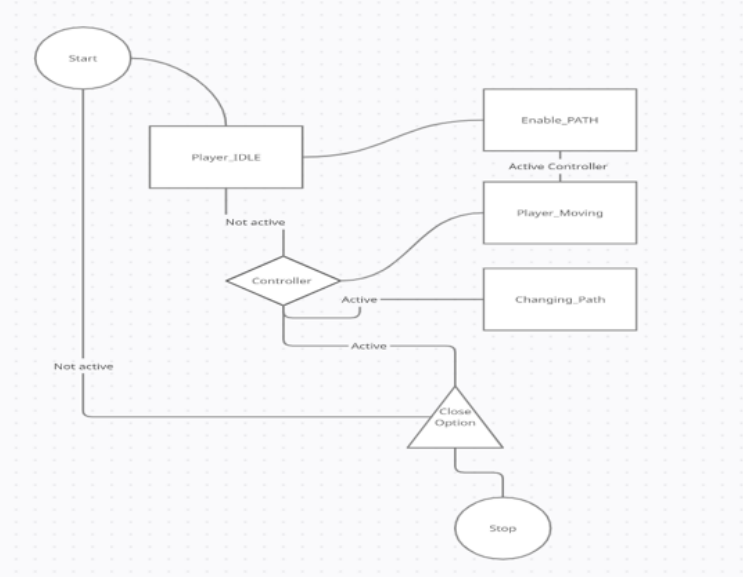
Обертання "Roll", "Pitch" і "Yaw" можна послідовно комбінувати для отримання будь-якого можливого 3D-обертання, маючи наступну формулу (4):

$$R(\alpha, \beta, \gamma) = Ry(\alpha)Rx(\beta)Rz(\gamma) \quad (4)$$

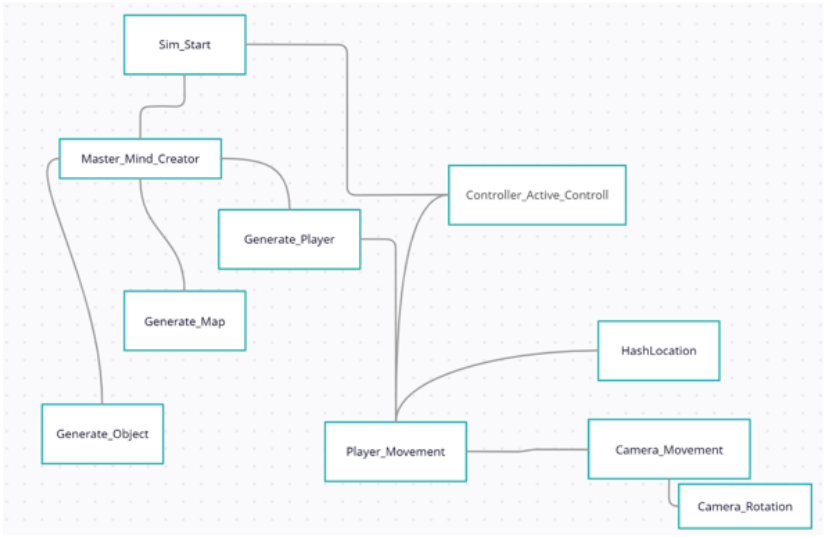


Обертання об'єкту

Узагальнений алгоритм руху в VR в вигляді UML діаграми



Діаграма класів



Характеристика розробленої системи в порівнянні з аналогами

АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Тези доповідей на конференціях:

1. Негоденко О.В. Кисельов В.О. Актуальність розробки симуляції для навчання на основі VR технологій / XV Науково-технічна конференція «Сучасні інфокомунікаційно технології» – Київ: ДУТ, 2022.

ВИСНОВКИ

1. Проаналізовано існуючі аналоги і проблеми, які виникають під час розробки і використанні додатків, які підтримують режим віртуальної реальності.
2. Розроблено модель руху використовуючи VR технології.
3. Проведено тестування програмного забезпечення.
4. Проведено моделювання руху актора в симуляції VR.
5. Встановлено, що при використанні розробленного методу,за результатами кількох тестів, навантаження на відеокарту зменьшуеється від 3% до 10%, а на процесор від 5 до 15%, ніж при використанні аналогів.

ДЯКУЮ ЗА УВАГУ!