

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет комп'ютерних наук, фізики та математики
Кафедра інформатики, програмної інженерії та економічної
кібернетики**

**СТВОРЕННЯ 3D - БІБЛІОТЕКИ ЛАБОРАТОРНИХ РОБІТ ДЛЯ
РОЗДІЛУ ФІЗИКИ «РУХ І ВЗАЄМОДІЯ. ЗАКОНИ ЗБЕРЕЖЕННЯ
В МЕХАНІЦІ» ДЛЯ ШКІЛЬНОГО КУРСУ ФІЗИКИ**

Кваліфікаційна робота (проект)

на здобуття ступеня вищої освіти « бакалавр»

Виконав: студент 4 курсу
Спеціальності 121 Інженерія
програмного забезпечення
Освітньо-професійної програми
«Інженерія програмного
забезпечення» першого
(бакалаврського) рівня освіти
Соценко Олександр Андрійович
Керівники старший викладач
Черненко Ірина Євгенівна
кандидат фізико-математичних наук,
доцент Єрмолаєв Вадим
Анатолійович
Рецензент Таточенко В.І., кандидат
педагогічних наук, доцент

Херсон – 2021

ЗМІСТ

ВСТУП	3
СПИСОК СКОРОЧЕНЬ	5
РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	6
1.1 Поняття віртуальної лабораторії.....	6
1.2 Огляд існуючих віртуальних лабораторій	7
1.3 Огляд існуючих додатків для створення 3D об'єктів	10
1.4 Методи моделювання 3D-об'єктів.....	12
1.5 Вибір засобів реалізації віртуальної лабораторії	13
РОЗДІЛ 2 ПРОЄКТУВАННЯ ТА РОЗРОБКА 3D-БІБЛІОТЕКИ ЛАБОРАТОРНИХ РОБІТ ДЛЯ ВІРТУАЛЬНОЇ ФІЗИЧНОЇ ЛАБОРАТОРІЇ	20
2.1 Архітектура проєкту	20
2.2 Створення 3D-об'єктів.....	22
2.3 Створення віртуальної лабораторії	24
ВИСНОВКИ	29
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	31
ДОДАТКИ	33
ДОДАТОК А	33

ВСТУП

Актуальність. У сучасному діджиталізованому світі, де комп'ютерні технології зустрічаються майже в усіх сферах сучасного життя, все частіше освітні установи приділяють увагу сучасним технологіям навчання. У процесі навчання використовують різні комп'ютерні технології навчання.

Застосування сучасних мультимедійних засобів в процесі навчання дозволяє підвищити інтерес учнів, їхню успішність по даній дисципліні, а також вирішити інші проблеми (наприклад відсутність матеріального забезпечення для наглядної демонстрацію якогось процесу). Комп'ютерні програми навчального характеру надають можливість учням (студентам) брати активну участь у навчальному процесі, допомагають навчатися віддалено, незалежно від матеріального забезпечення.

У зв'язку з потребою сьогодення у збільшенні використання віддалених форм навчання особливо актуальним стає вивчення природничих дисциплін за допомогою спеціальних програмних продуктів - віртуальних лабораторій. Завдання, які ставляться при створенні таких програмних продуктів – це побудувати 3D-моделі реальних об'єктів, створити середовище, у якому можна виконати симуляцію навчальних лабораторних робіт, виконати моделювання реальних процесів, і надати можливість учасникам навчального процесу взаємодіяти з віртуальними об'єктами, як з реальними.

Для розробки даного програмного забезпечення було обрано ігровий рушій Unity 3D та пакет для створення тривимірної графіки Blender. Unity 3D дозволить реалізувати необхідні програмні рішення, а Blender дозволить створити необхідні моделі.

Метою даної роботи є спроектувати 3D-бібліотеку лабораторних робіт шкільного курсу фізики з розділу «Рух і взаємодія. Закони збереження в механіці».

Об'єкт дослідження – методи та технології створення віртуальних лабораторій.

Предмет дослідження – проектування віртуальної лабораторії навчального призначення.

Досягнення зазначеної мети здійснюється шляхом вирішення таких **основних завдань**:

1. Дослідити технології моделювання 3D-об'єктів та створення віртуальних лабораторій.

2. Дослідити та обрати середовище для розробки віртуальної лабораторії.

3. Розробити бібліотеку 3D об'єктів.

4. Спроектувати програмне забезпечення віртуальної лабораторії для шкільного курсу фізики з розділу «Рух і взаємодія. Закони збереження в механіці».

Структура роботи складається з вступу, двох розділів, висновку і списку використаних джерел.

СПИСОК СКОРОЧЕНЬ

ВЛ – віртуальна лабораторія.

ПЗ – програмне забезпечення.

САП – система автоматизованого проектування.

NURBS – non-uniform rational b-spline.

БД – база даних.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Поняття віртуальної лабораторії

«Під поняттям "лабораторія" розуміють місце, спеціально організоване для проведення лабораторних занять, експериментів і місце для пошуку рішень у галузі фундаментальних наук, або для вирішення завдань у певній прикладній галузі знань. Віртуальна лабораторія – це віртуальне програмне середовище, в якому організована можливість дослідження поведень моделей об'єктів, їх сукупностей і похідних, заданих з певною часткою деталізації щодо реальних об'єктів, в рамках певної галузі знань»[1]

ВЛ надає можливість не лише спостерігати за певним явищем, а й брати у ньому участь, а це дає змогу користувачу краще засвоїти отримані під час експерименту знання.

Переваги віртуальних лабораторій

- Відсутність необхідності придбання дорогого устаткування і реактивів.
- Можливість моделювання процесів, протікання яких принципово неможливе в лабораторних умовах.
- Можливість проникнення в тонкощі процесів і спостереження того, що відбувається в іншому масштабі часу, що актуально для процесів, що протікають за долі секунди або, навпаки, тривають протягом декількох років.
- У зв'язку з тим, що управлінням віртуального процесу займається комп'ютер, з'являється можливість швидкого проведення серії дослідів з різними значеннями вхідних

параметрів, що часто необхідно для визначення залежностей вихідних параметрів від вхідних.

- Економія часу та ресурсів для введення результатів в електронний формат. Деякі роботи вимагають подальшої обробки досить великих масивів отриманих цифрових даних, які виконуються на комп'ютері після проведення серії експериментів
- Можливість використання віртуальної лабораторії в дистанційному навчанні, коли в принципі відсутня можливість роботи в лабораторіях навчальних закладів.

1.2 Огляд існуючих віртуальних лабораторій

STAR (Software Tools for Academics and Researchers) – це програма Массачусетського технологічного інституту (MIT) з розробки віртуальних лабораторій для досліджень і навчання. Суть цієї програми полягає в розробці навчальних та дослідницьких додатків із загальної біології, біохімії, генетики, гідрології, в області розподілених обчислень.

- StarBiochem – 3D-визуалізатор молекул білків. Має гнучку і детальну настройку.
- StarGenetics – дозволяє моделювати процеси схрещування, вивчати закономірності успадкування моногенних ознак (так званий закони Менделя).
- StarORF – дозволяє навчитися ідентифікувати так звані відкриті рамки зчитування (ORF - Open Reading Frame) - одиниці в складі ланцюга ДНК або РНК, здатні кодувати білок.
- StarMolSim – це серія інструментів, що моделює процеси молекулярної динаміки. Кожен з інструментів має широкий набір вхідних значень і, аналогічно, широкий набір вихідних значень для аналізу і дослідження.

- StarBiogene – набір інструментів з генетики.
- StarCluster – набір інструментів для створення, налаштування і управління кластерами віртуальних машин.

Дати оцінку цим віртуальним лабораторіям можуть тільки відповідні фахівці, однак можна з певною часткою впевненості стверджувати, що вони відрізняються фундаментальністю, охоплюють широке коло завдань у певній сфері знань, та володіють багатим інструментарієм.

Віртуальні лабораторії VirtuLab

VirtualLab - проект з розробки віртуальних лабораторних робіт для учнів з фізики, хімії, біології, екології. Віртуальні лабораторні роботи реалізовані за допомогою технології Flash. Відрізняються вузькою спеціалізацією, в більшості випадків лінійністю досвіду (вся послідовність дій і результати досвіду задані заздалегідь).

Продукти VirtualLab мають пізнавальну цінність і вирішують задачу проведення лабораторних робіт при відсутності необхідного обладнання.

Приклади лабораторних робіт:

- Вивчення взаємодії часток і ядерних реакцій.
- Порівняння молярних теплоємностей металів.
- Вивчення закону Ома для повного кола.
- Знайомство із зразками металів і сплавів.
- Ідентифікація неорганічних сполук.
- Вивчення зовнішньої будови та різноманіття членистоногих.
Зовнішня будова річкового раку. Зовнішня будова комахи.
- Дослідження змін в екосистемах на біологічних моделях (акваріум).
- Модель хижак-жертва Лотки-Вольтерри.

Algodoo – програма, призначена для фізичних 2D симуляцій. Має дуже багатий інструментарій для створення різних об'єктів, механізмів і

систем з метою моделювання їх фізичного взаємодії і властивостей. Наприклад, можна створити модель працюючого годинника, модель планетохода або пневматичної гвинтівки. Програма здатна симулювати і механічні процеси, і оптичні, а можливість програмування за допомогою скриптової мови Thyme дозволяє створювати об'єкти з різними фізичними властивостями, різними функціями, а також дозволяє створювати різні ефекти і явища. В програмі є можливість завантажувати малюнки: малюнок стає об'єктом симуляції і йому можна задати будь-які фізичні властивості. Є сховище `albox`, де користувачі можуть обмінюватися своїми моделями.

PhET – проєкт розроблений Університетом Колорадо. Проєкт включає безліч віртуальних лабораторій, які демонструють різні явища з областей фізики, біології, хімії, математики, наук про Землю. Досліди мають високу пізнавальну цінність і при цьому дуже цікаві.

Wolfram Demonstrations Project – наочна демонстрація концепцій сучасної науки і техніки. Wolfram претендує на роль єдиної платформи, що дозволяє створити об'єднаний каталог онлайн-інтерактивних лабораторій. Це, на думку його розробників, дозволить користувачам уникнути проблем, пов'язаних із застосуванням різнорідних навчальних ресурсів і платформ розробки. Для перегляду демонстрацій знадобиться завантажити і встановити спеціальний Wolfram CDF Player. На поточний момент (квітень 2021) Wolfram Demonstrations Project володіє значним каталогом - приблизно 8900 інтерактивних демонстрацій. Каталог проєкту складається з 11 основних розділів, що відносяться до різних галузей знань і людської діяльності. Тут є великі фізичні, хімічні та математичні розділи, а також розділи, присвячені техніці, інженерній справі, соціальним наукам. Приклади:

- RadialEngine.
- 3D Skeletal Anatomy of the Arm.
- Epidemic Spread and Transmission Network Dynamics.

- Optimizing the Counterweight Trebuchet.
- A Special Case of the Sum of Two Cosines.
- Keynesian Cross Diagram.

Late Nite Labs – це серія віртуальних лабораторій з хімії, біології та мікробіології. Віртуальні лабораторії оформлені у вигляді 3D світу.

1.3 Огляд існуючих додатків для створення 3D об'єктів

Оскільки для реалізації 3D-бібліотеки необхідно створити самі 3D-об'єкти, необхідно розглянути існуюче ПЗ, за допомогою якого можна буде створювати необхідні об'єкти. Існує багато програм, які створені для цих цілей. Ось деякі з них:

- 3Ds MAX
- Cinema 4D
- AutoCAD
- Autodesk 123D
- ARCHICAD
- SketchUP
- Paint 3D
- Wings 3D
- Blender

3Ds Max – це тривимірний графічний редактор, повнофункційний професійний додаток, був розроблений компанією Autodesk. Програма широко використовується у візуалізації інтер'єрів та екстер'єрів.

Cinema 4D або скорочено **C4D** – це програмний пакет, який був створений компанією Maxon. Програма створена для моделювання тривимірної графіки та анімації, широко використовується у кінематографі та проектуванні дизайнів готових продуктів.

AutoCAD – дво- та тривимірна система автоматизованого проектування (САП) і креслення. Програма широко використовується в

архітектурі, будівництві, машинобудуванні та інших суміжних галузях промисловості. Програма направлена аби створювати високо точні моделі для подальшого їхнього використання у промисловості або 3D-друці. Програма була розроблена компанією Autodesk.

SketchUp – це програма для моделювання відносно простих тривимірних об'єктів. Програма не підходить для реалізації теми дипломної роботи, через дуже бідний функціонал.

Autodesk 123D – це набір засобів САП та 3D-моделювання для ентузіастів, створений компанією Autodesk. Крім простіших можливостей малювання і моделювання, він також підтримує збірку, обмеження і експорт в формат .stl. Для програми доступна також бібліотека готових блоків і об'єктів. Програма занадто проста і не підтримує необхідні функції для реалізації теми дипломної роботи. Програма більше не підтримується компанією Autodesk.

ArchiCAD – це графічний додаток САП BIM (Building Information Modeling) для архітекторів, розроблений угорською компанією Graphisoft. Призначений для інженерії, проектування архітектурно-будівельних конструкцій і рішень, а також для створення меблів та елементів ландшафту.

Paint 3D – додаток, розроблений та представлений в операційній системі Windows 10, представляє собою переосмислення Microsoft Paint для тривимірного моделювання та друку. Сумісний з View 3D, Windows Mixed Reality, Holograms і 3D Builder.

Blender – це програмний додаток для створення тривимірної комп'ютерної графіки, який містить засоби моделювання, анімації, рендерінгу, після-обробки відео. Особливостями даного програмного продукту є малий розмір, висока швидкість рендерінгу, сумісність з багатьма різними операційними системами – FreeBSD, GNU/Linux, Mac OS X, SGI Irix 6.5, Sun Solaris 2.8, Microsoft Windows, SkyOS, MorphOS та Pocket PC. Програма має функції симулювання твердих та м'яких тіл,

симулювання рідин та тканин, велику кількість доступних та функціональних розширень. Програма є вільним програмним забезпеченням та розповсюджується під ліцензією GNU GPL.

Порівняємо характеристики деяких найбільш потужних та популярних графічних 3D редакторів, а саме 3Ds max, Blender, Cinema 4d, ArchiCAD.

Таблиця 1

Критерії	Blender	Cinema 4d	3Ds max	ArchiCAD
Кросплатформність	Так	Так	Ні	Ні
Підтримка експорту декількох форматів	Так	Так	Так	Ні
Підтримується розробниками	Так	Так	Так	Так
Вимоги до системи	Низькі	Середні	Середні	Середні

В якості програми для створення моделей для ВЛ було обрано графічний 3D-редактор Blender, оскільки він володіє всіма необхідними властивостями та характеристиками, наданий момент підтримується зі сторони розробників, має можливість експорту моделей у різні формати та сильно не навантажує систему.

1.4 Методи моделювання 3D-об'єктів

На сьогоднішній день існує всього 4 методи моделювання 3D-об'єктів:

- NURBS моделювання.

- Полігональне моделювання.
- Скульптурне моделювання.
- Комбінований метод.

NURBS моделювання або технологія non-uniform rational b-spline – це технологія неоднорідних раціональних B-сплайнів, створення плавних форм і моделей, у яких немає гострих країв, як у полігональних моделей. Саме через цю відмінну рису технологію NURBS засовують в основному для побудови органічних моделей і об'єктів (наприклад: рослин, тварин, людей).

Полігональне моделювання – це вид 3d-моделювання, вся суть якого зводиться до моделювання об'єктів шляхом апроксимації їх поверхні за допомогою полігонів. Основними об'єктами за допомогою яких моделюють об'єкти є:

- вершина (точка в тривимірному просторі),
- ребро (дві вершини, з'єднані між собою),
- полігон (ребра, з'єднані між собою).

Вони, у свою чергу, формують об'єкт.

Скульптурне моделювання - це 3D-моделювання, яке імітує процес «ліпки» 3d-моделі, тобто деформація поверхні моделі здійснюється спеціальними інструментами «пензлями».

Комбінований метод - це підхід, який базується на всіх вище зазначених методах, які можуть бути скомбіновані на різних етапах моделювання.

1.5 Вибір засобів реалізації віртуальної лабораторії

Процес розробки ВЛ дуже схожий на розробку звичайного ПЗ, але є деякі особливості у створенні ВЛ. Ці особливості відносяться до програмних інструментів за допомогою яких буде створено ВЛ та деяких обмежень. В першу чергу ПЗ, за допомогою якого буде створено

ВЛ, повинно включати фізичний рушій, який буде відповідати за коректну роботу фізичних процесів та властивостей в середині ВЛ.

Основні критерії у виборі технології для створення ВЛ:

- Інтуїтивність, зручність інтерфейсу IDE.
- Фізичний двигун.
- Мова програмування.
- Кросплатформеність.
- Інструменти для реалізації мережевої взаємодії.
- Додаткові сервіси.

На сьогодні існує безліч різноманітних засобів програмного забезпечення, що допомагають в розробці різних видів проектів, у тому числі ігрового характеру.

ПЗ яке спрямоване на створення ігрових додатків, називають ігровим рушієм.

Розглянемо деякі серед найпоширеніших ігрових рушіїв, та мови програмування, які вони використовують для реалізації проектів.

Unity3D – ПЗ за допомогою якого розробляють кросплатформенні додатки. Дозволяє створювати як 3D так і 2D-додатки, має підтримку різних графічних бібліотек, серед них DirectX та OpenGL.

Unity дає можливість обрати одну з двох мов програмування для створення і реалізації власного проекту: JavaScript та C#.

- JavaScript – це об'єктно-орієнтована мова програмування, що є досить популярною серед розробників веб-додатків або браузерних ігор.
- C# – це об'єктно-орієнтована мова програмування, яка була спроектована для платформи .NET. Ця мова програмування має широкий спектр стандартних бібліотек, що досить радикально спрощує час та процес розробки.

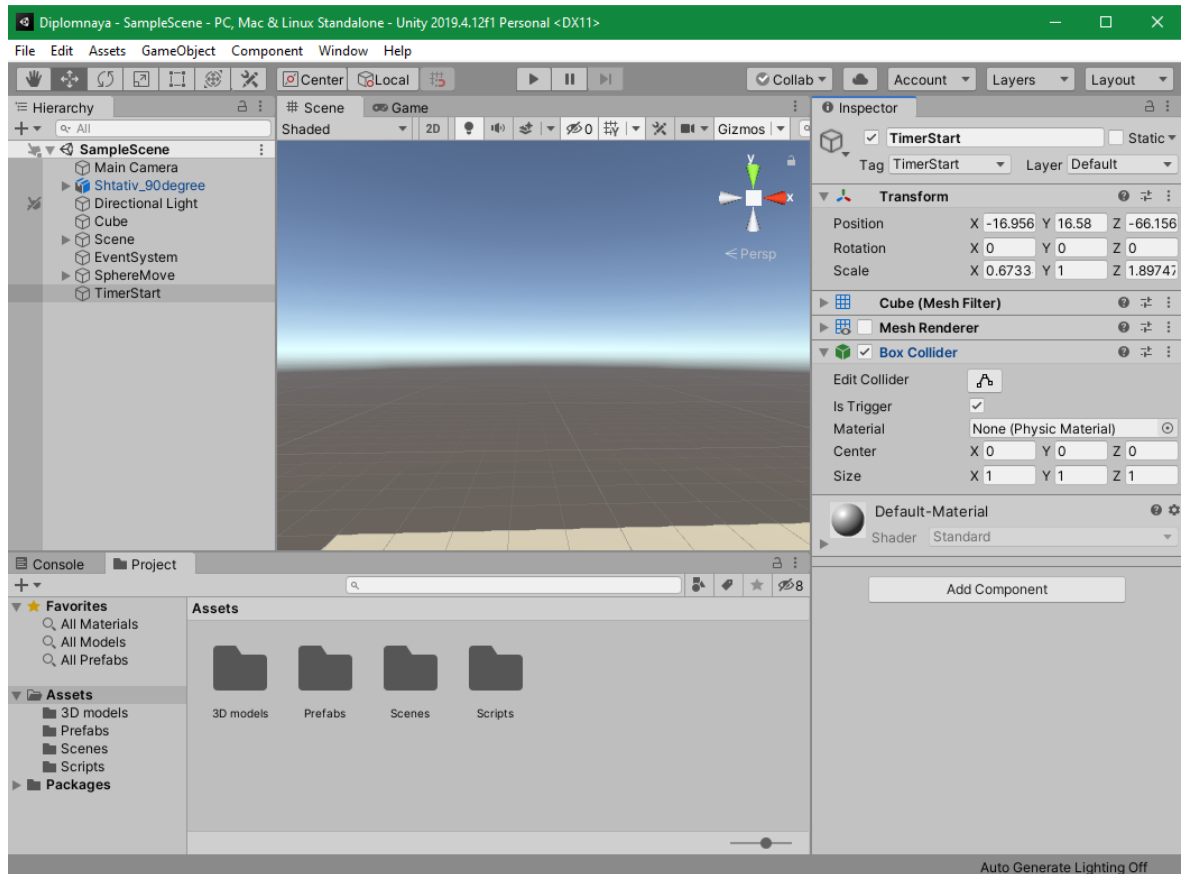


Рисунок 1.1 Інтерфейс Unity 3D

Серед переваг Unity слід зазначити, що з його допомогою можна створювати досить невеликі (за об'ємом займаної пам'яті) проекти у порівнянні з іншими ігровими рушіями, оскільки він включає в себе невеликий набір інструментів для розроблення, завдяки чому має інтуїтивно зрозумілий інтерфейс (див. рисунок 1.1). Також серед переваг Unity можна зазначити те, що програма має стабільну версію яка підтримується командою розробників, які регулярно випускають нові версії програми.

CryEngine – досить потужний ігровий рушій, але вузько направлений на створення ігрових проектів в жанрі «шутер». З допомогою цієї програми можна створювати фотореалістичну графіку. Має підтримку графічної бібліотеки DirectX 12, також підтримує програмування шейдерів, які необхідні для коректної роботи 3-х вимірної графіки, реалізації фотореалістичної графіки та графіки

загалом. Стандартною мовою програмування в середовищі CryEngine є мова C++.

C++ це одна з основних мов програмування, яка використовується в індустрії створення відеоігор, бо ця мова дозволяє забезпечити повний контроль над усіма важливими ресурсами та процесами в середині самого проєкту. Наприклад, керування пам'яттю. CryEngine має безліч існуючих бібліотек та є досить потужним ПЗ, яке володіє досить високою швидкістю роботи та багатоплатформеністю.

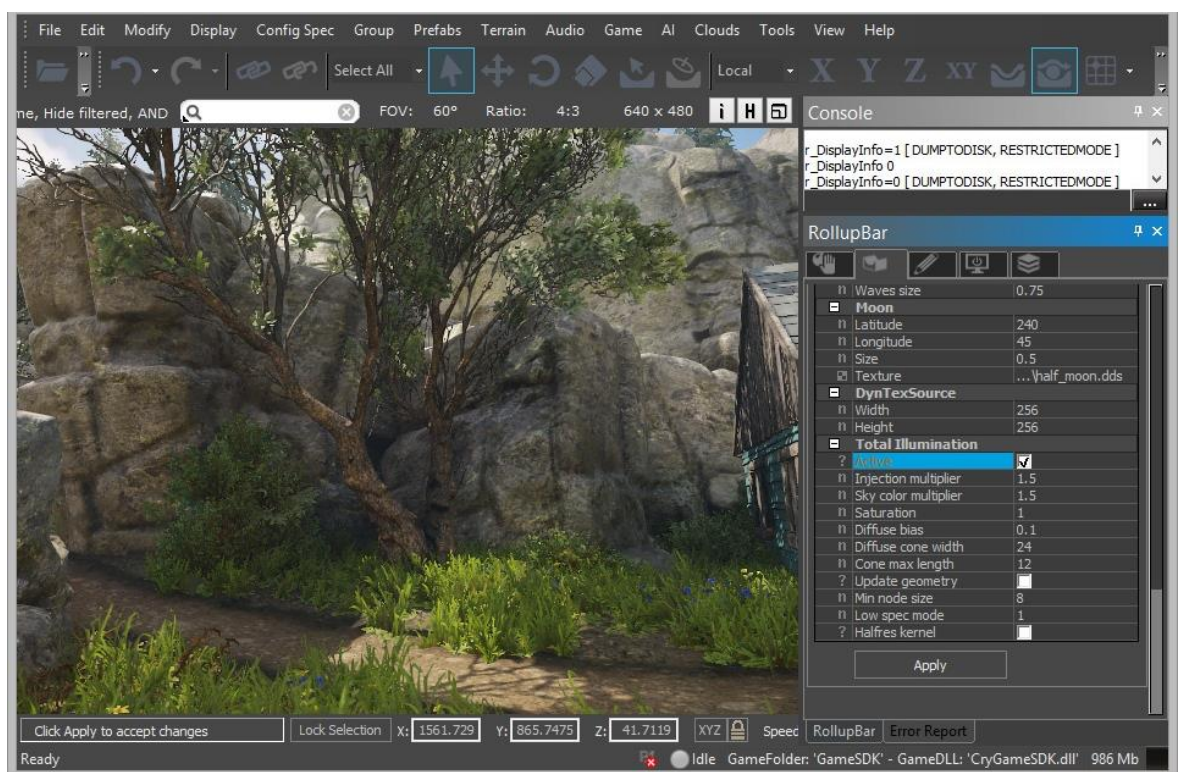


Рисунок 1.2 Інтерфейс CryEngine

Серед недоліків даного ПЗ можна вказати: досить важке засвоєння через необхідність відстежувати всі ресурси програми самостійно для забезпечення її надійної роботи. Ще одним досить вагомим недоліком є те, що ця програма потребує велику кількість ресурсів, аби забезпечити кінцевий продукт «реалізмом». Тому створити легку програму з його допомогою досить складно.

Інтерфейс робочого вікна рушія наведений на рис. 2.3.

Unreal Engine – це ще один популярний серед розробників ігровий рушій з великим сектором інструментарію для створення 2D- та 3D-проектів. Програма використовує мову програмування C++, яка була вже описана вище. Так само, як і CryEngine та Unity3D цей рушій є кросплатформним, та має підтримку багатьох графічних бібліотек, серед них DirectX, OpenGL та Vulkan. Програма постійно оновлюється та має підтримку з боку розробників.

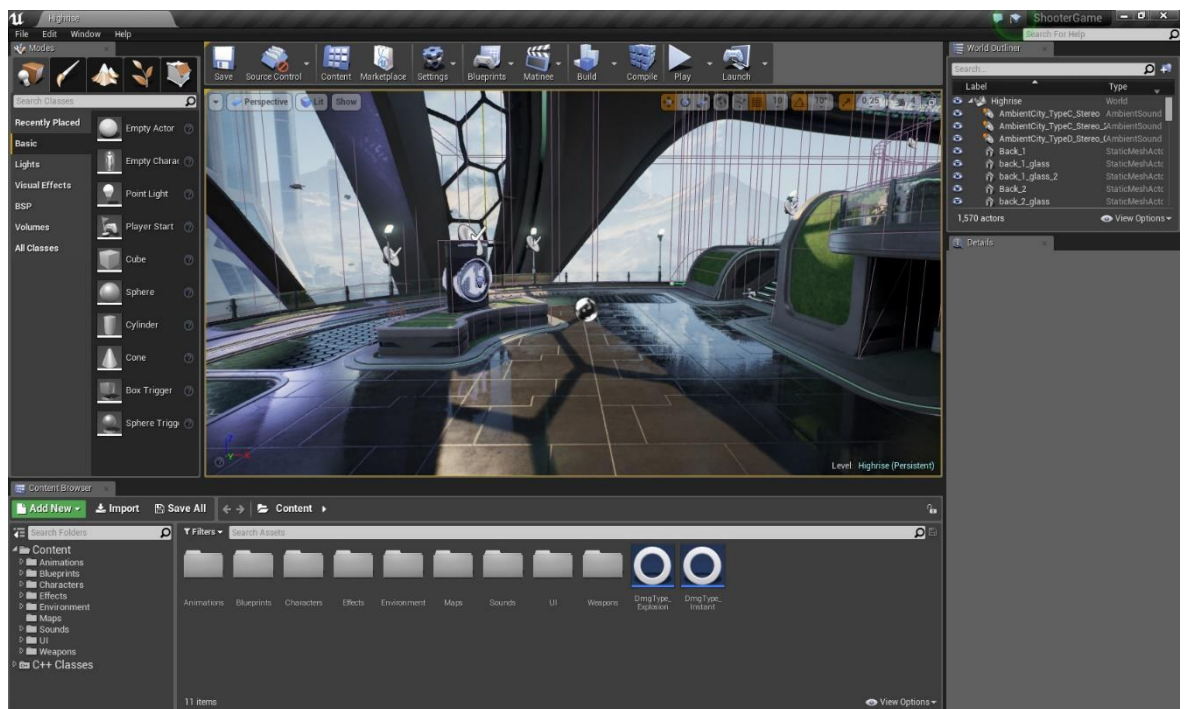


Рисунок 1.3 Інтерфейс Unreal Engine

Завдяки великій кількості бібліотек та існуючих інструментів, можна реалізувати досить якісну графіку, але кінцевий продукт буде не таким легким у порівнянні, якщо реалізувати його за допомогою Unity3D.

Інтерфейс робочого вікна рушія наведений на рис. 2.4.

Cocos2D – програмний продукт з відкритим вихідним кодом, який не містить вбудований редактор ігрової сцени, що робить розробку

проектів складнішою. Має підтримку двох мов програмування, таких як: C++ та JavaScript. Є міжплатформеним, але список підтримуваних операційних систем коротший у порівнянні з вище згаданими рушіями.

Проведемо невелике порівняння вище розглянутих ігрових рушіїв, будемо брати до уваги такі характеристики як:

- кросплатформеність;
- фізичний рушій, який використовується;
- підтримувані мови програмування;

Таблиця 2

Ігровий рушій	Фізичний рушій	Підтримувані мови програмування	Кросплатформеність
Unity 3D	PhysX, Box2D, DOTS, Havok	C# та JavaScript	Так (Android, HTC Vive, iOS, Linux, macOS, Microsoft HoloLens, Nintendo Switch, Oculus Rift, PlayStation 4, Samsung Smart TV, tvOS, WebGL, Wii U, Windows, Windows Phone, Xbox One.)
CryEngine	CryPhysics	C++	Так (Windows, Linux, PlayStation 4, Xbox One, Oculus Rift, OS VR, PS VR, HTC Vive)
Unreal Engine	Havok 2.0 та PhysX 3.3	C++	Так (Microsoft Windows, Linux, Mac OS, Xbox, PlayStation, PSP, PSVita, Wii, Dreamcast, IOS)
Cocos2D	Box2D та Chipmunk	C++, JavaScript та Lua	Так (iOS, Android, Windows Phone 8, Windows 7, Linux, Mac OS X, Windows)

Отже, виходячи із вище зазначеної інформації можна зробити висновок, що найкращим для створення необхідного програмного забезпечення є Unity 3D, оскільки, він є досить простим та гнучким у використанні. Має підтримку багатьох різних платформ, а також програми, створені за його допомогою, займають незначне місце у пам'яті. Цей рушій створений за допомогою мови C++, що робить його швидким у роботі. Рушій задовольняє ряду вимог, необхідних для реалізації ВЛ, а саме:

- результуючий продукт є високорівневою абстракцією;
- є можливість легко вбудувати кінцевий продукт в середину будь-якого web-сервісу;
- дасть змогу легко створити необхідний графічний супровід для представлення інформації в середині ВЛ;
- Unity дозволяє працювати з різноманітними 3D форматами (.fbx, .flt, .dae);
- Unity підтримує високорівневі мови програмування такі як JavaScript та C#;
- Unity дає можливість створювати некомерційні проекти безкоштовно;
- підтримується підключення сторонніх бібліотек, якщо це необхідно.

Для розробки програмної системи було обрано наступні технології:

Мова програмування C#, ігровий рушій Unity 3D, 3-х вимірний графічний редактор Blender.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ТА РОЗРОБКА 3D-БІБЛІОТЕКИ ЛАБОРАТОРНИХ РОБІТ ДЛЯ ВІРТУАЛЬНОЇ ФІЗИЧНОЇ ЛАБОРАТОРІЇ

2.1 Архітектура проєкту

Загальний вигляд клієнт-серверної архітектури ВЛ показано на рис. 2.1. Клієнт та сервер будуть обмінюються даними за допомогою REST запитів. Так як БД буде знаходитися на сервері за обробку даних буде відповідати сервер і у разі необхідності надсилати необхідні дані користувачу в залежності від його дій.

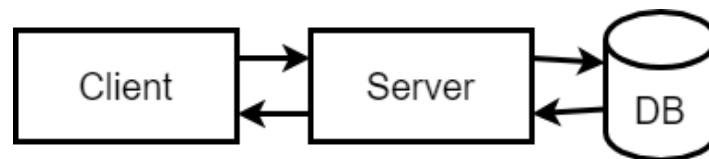


Рисунок 2.1 Клієнт-серверна архітектура

ВЛ містить в собі значний обсяг інформації, що безумовно буде займати зайві ресурси при кожному запуску програми, тому доцільніше реалізувати проєкт у вигляді веб-додатка. Оскільки користувачу всі модулі ВЛ зразу будуть не потрібні, краще буде реалізувати механізм завантаження спочатку основних функцій необхідних користувачу, а вже далі користувач, виходячи із свої цілей, буде додатково обирати необхідний модуль ВЛ. В такому випадку ми знизимо навантаження на трафік ресурсів і підвищимо ефективність їх розподілення. На сьогоднішній день реалізація ВЛ у вигляді веб-додатка є найбільш доцільною. Цьому сприяють безліч різних факторів, такі як імплементація дистанційного навчання, все більший розвиток та розповсюдження швидкісного підключення до мережі інтернет,

діджиталізація та інші. Найбільш очевидною є реалізація ВЛ у вигляді контейнера для сервера.

Етапи розробки віртуальної лабораторії виглядають наступним чином:

- створення 3-х вимірної моделі інструментів необхідних у ході виконання лабораторних робіт;
- перенесення створених моделей в середовище Unity;
- іменування перенесених об'єктів, за допомогою чого користувач буде звертатися до об'єктів;
- ініціалізація параметрів, необхідних для фізичної взаємодії об'єктів, щоб брати участь в експерименті;
- визначення додаткових змінних або констант (прискорення вільного падіння, нормальна або поточна температура середовища та інше);
- опис реалізованого процесу математичною моделлю.

У процесі створення ВЛ ми додаємо об'єкт на сцену. Далі, якщо це необхідно, є можливість додати до об'єкта необхідні компоненти. Додавши необхідний компонент ми можемо редагувати його параметри у вікні Інспектора (на рисунку він виділений червоним прямокутником). Також є можливість створювати власні компоненти, які допоможуть вирішити необхідні завдання проекту.

Кожна ВЛ містить фізичну модель досліджуваного об'єкта, кожен фізичний об'єкт має математичну модель, яка описує необхідні взаємодії.

Одним з основних компонентів для симуляції процесів у фізиці є компонент Collider та компонент Rigidbody.

Collider використовується для того аби фізичний рушій міг коректно визначати перетин з іншими тілами та коректно симулювати фізику взаємодії між ними.

Rigidbody відповідає за контроль положення об'єкта за допомогою фізичного моделювання. Додавання компонента Rigidbody до об'єкта поставить його рух під контроль фізичного механізму Unity. Навіть без додавання будь-якого коду, об'єкт із компонентом Rigidbody буде тягнути вниз під силою дії гравітацією і реагуватиме на зіткнення з вхідними об'єктами, якщо також присутній компонент Collider. Rigidbody також має сценарій API, який дозволяє застосовувати сили до об'єкта та управляти ним фізично. Наприклад, поведінку автомобіля можна визначити з точки зору зусиль, що діють на колеса. Враховуючи цю інформацію, фізичний двигун може обробляти більшість інших аспектів руху автомобіля, тому він буде реалістично прискорюватися і правильно реагувати на зіткнення.

2.2 Створення 3D-об'єктів

Для початку створимо 3d об'єкти які будуть необхідні нам в процесі розробки віртуальних лабораторних робіт. Розглянемо створені 3D об'єкти на прикладі деяких лабораторних робіт (дослідження руху тіла кинутого горизонтально, вивчення закону збереження механічної енергії та дослідження руху тіла по колу).

Для лабораторної роботи, дослідження руху тіла кинутого горизонтально нам знадобляться такі об'єкти як штатив з муфтою та лапкою, кулька, жолоб (бігова доріжка для кульки) з насадкою (рис. 2.2). Для створення цих моделей можна обрати метод полігонального моделювання оскільки ці моделі досить прості і не потребують якихось складних підходів для їх створення.

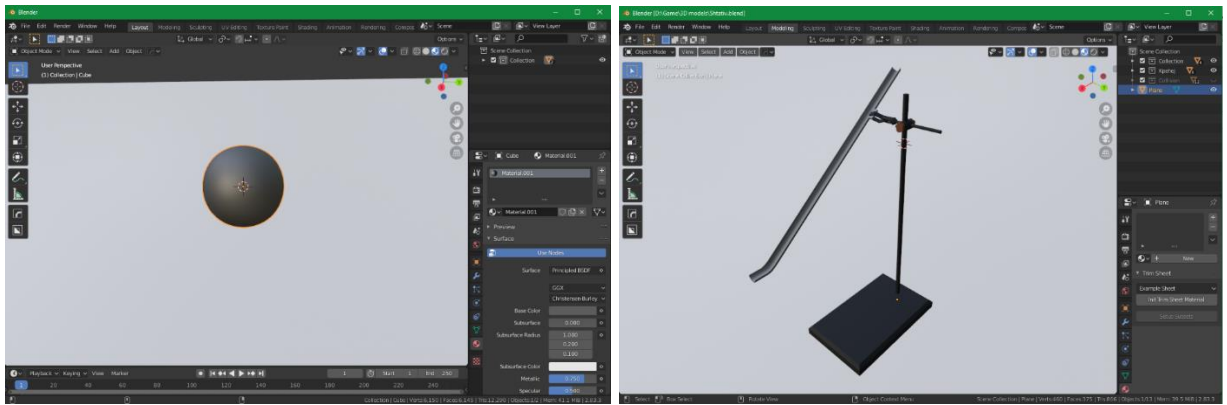


Рисунок 2.2 Модель кульки та штатива з муфтою і закріпленням жолобом у вікні програми Blender

Для лабораторної роботи, вивчення закону збереження механічної енергії, нам знадобляться наступні об'єкти: муфта з штативом, динамометр, тягарець, лінійка (рис. 2.3).

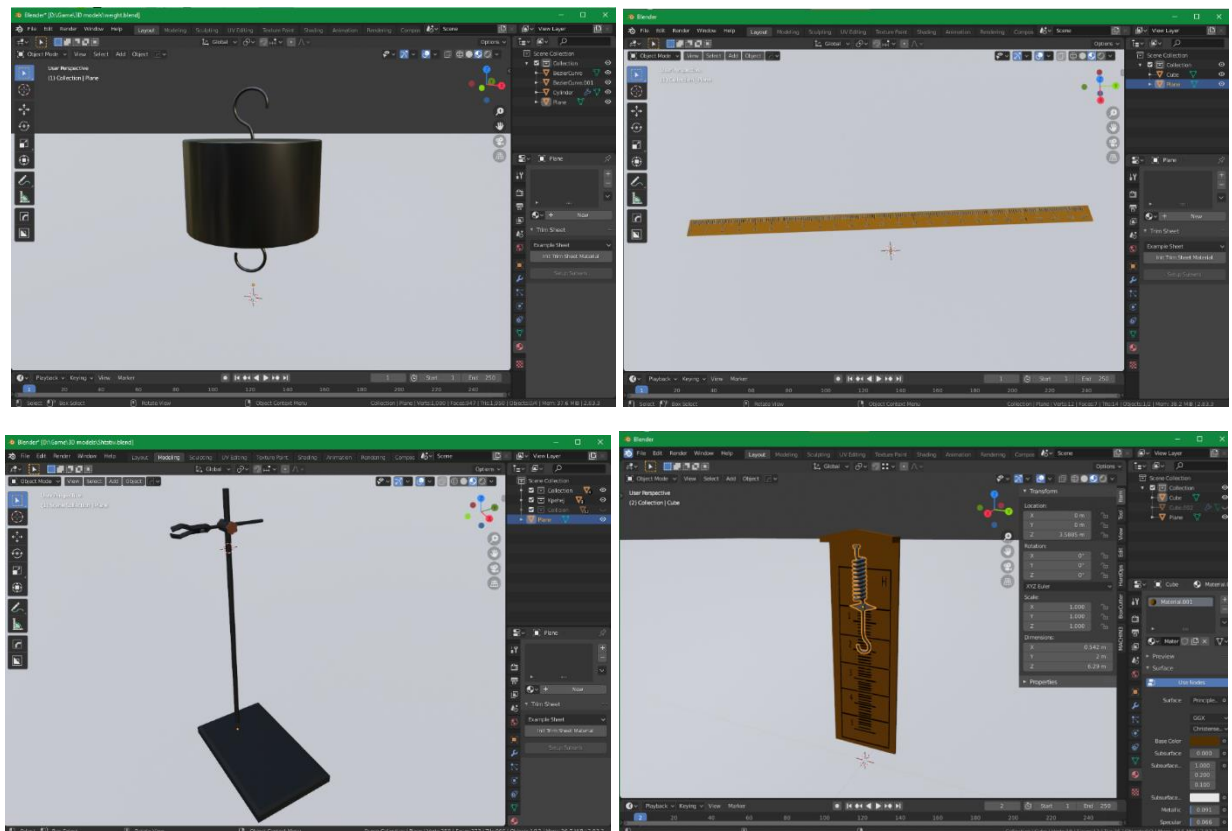


Рисунок 2.3 Модель тягарця, лінійки, динамометра та штатива з муфтою у вікні програми Blender

Після того як необхідні моделі будуть створені їх необхідно додати в Unity 3D і продовжити роботу там.

2.3 Створення віртуальної лабораторії

Для зручної взаємодії користувача з ВЛ необхідно розробити модуль головного меню, у якому користувач зможе обирати лабораторну роботу, при необхідності змінювати деякі налаштування та мати змогу завершити програму (рис. 2.4).

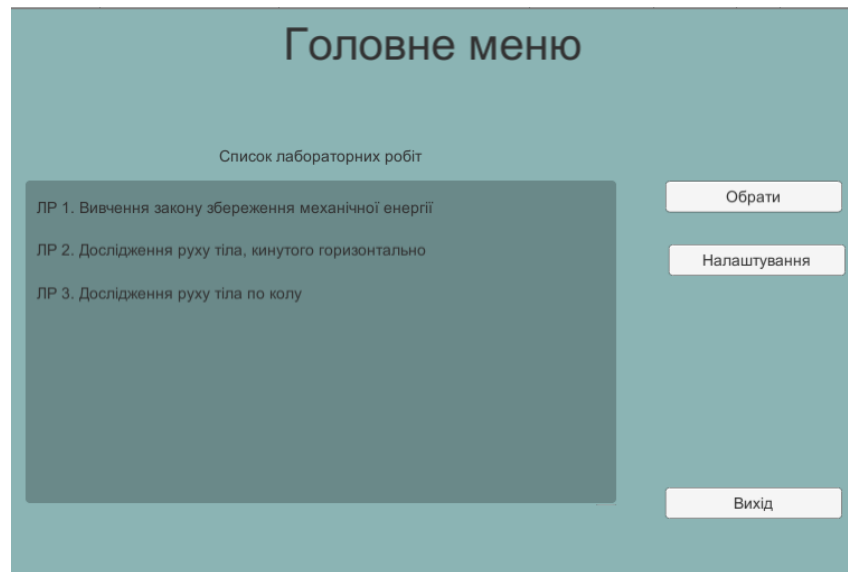


Рисунок 2.4 Головне меню ВЛ

На рисунку 2.5 представлено загальну схему ВЛ.



Рисунок 2.5 Загальна схема ВЛ

З процесом взаємодії ВЛ з лабораторною роботою можна ознайомитись за допомогою діаграми станів (рис. 2.6).

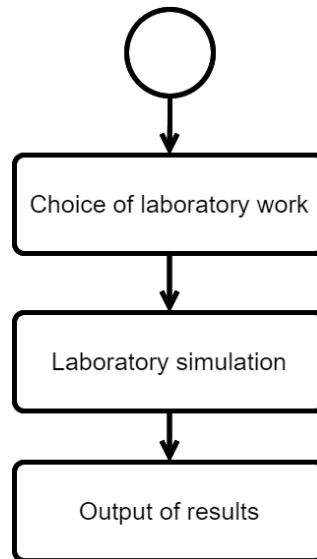


Рисунок 2.6 Діаграма станів ВЛ

Наступним кроком розробки є додавання створених 3D-об'єктів у середовище Unity.

Після додавання всіх об'єктів необхідно задати їх властивості.

Процес додавання властивостей для об'єкту розглянемо на прикладі кульки з лабораторної роботи «Дослідження руху тіла, кинутого горизонтально». Необхідно задати масу кульки, її фізичну поверхню (аби кулька знала як їй реагувати на інші об'єкти) та інші властивості.

Описати властивості об'єктів у програмі Unity можна за допомогою: `rigidbody` та `collider`. На рисунку 2.7 представлено вікна цих компонент з необхідними властивостями для кульки.

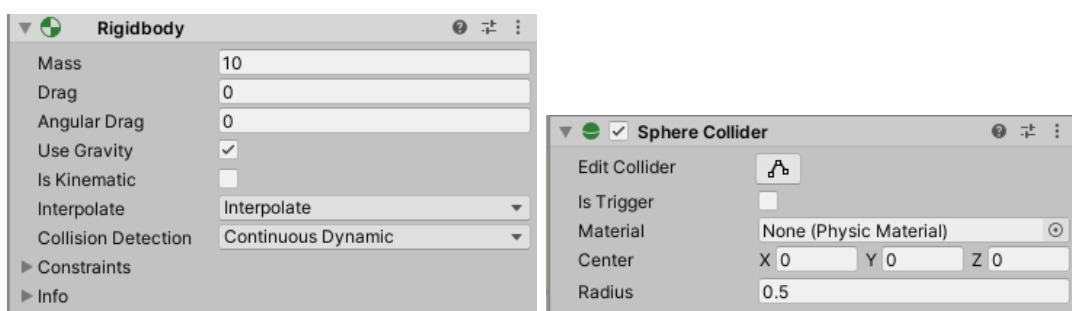
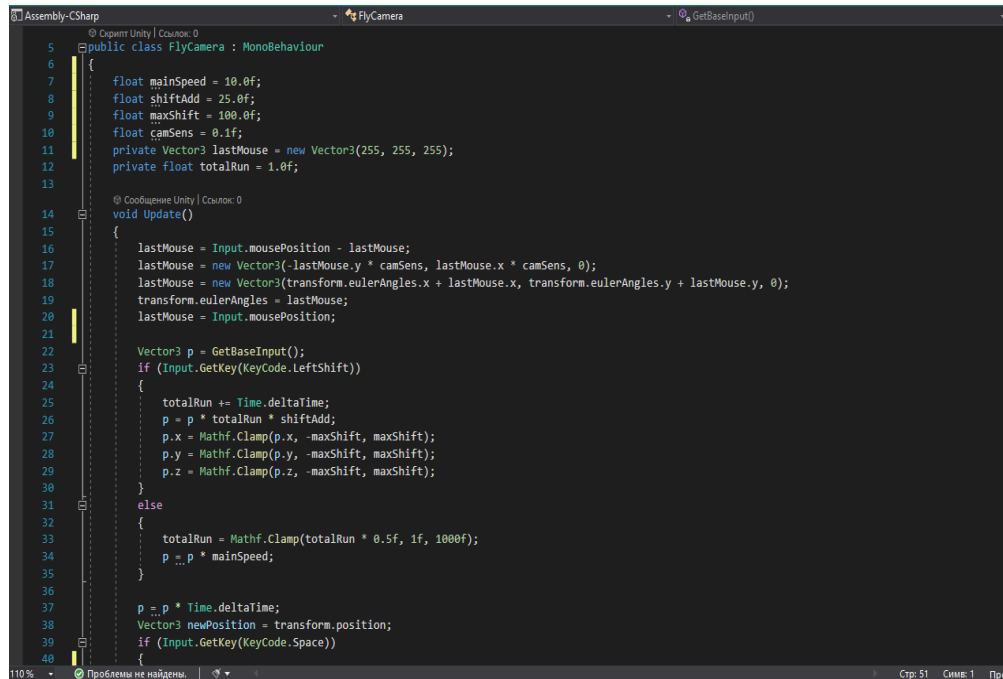


Рисунок 2.7 Компоненти Rigidbody та Collider

Для комфортної роботи в ВЛ напишемо код, який дозволить переміщуватися всередині ВЛ (рис. 2.8).



```

5 public class FlyCamera : MonoBehaviour
6 {
7     float mainSpeed = 10.0f;
8     float shiftAdd = 25.0f;
9     float maxShift = 100.0f;
10    float camSens = 0.1f;
11    private Vector3 lastMouse = new Vector3(255, 255, 255);
12    private float totalRun = 1.0f;
13
14    void Update()
15    {
16        lastMouse = Input.mousePosition - lastMouse;
17        lastMouse = new Vector3(-lastMouse.y * camSens, lastMouse.x * camSens, 0);
18        lastMouse = new Vector3(transform.eulerAngles.x + lastMouse.x, transform.eulerAngles.y + lastMouse.y, 0);
19        transform.eulerAngles = lastMouse;
20        lastMouse = Input.mousePosition;
21
22        Vector3 p = GetBaseInput();
23        if (Input.GetKey(KeyCode.LeftShift))
24        {
25            totalRun += Time.deltaTime;
26            p = p * totalRun * shiftAdd;
27            p.x = Mathf.Clamp(p.x, -maxShift, maxShift);
28            p.y = Mathf.Clamp(p.y, -maxShift, maxShift);
29            p.z = Mathf.Clamp(p.z, -maxShift, maxShift);
30        }
31        else
32        {
33            totalRun = Mathf.Clamp(totalRun * 0.5f, 1f, 1000f);
34            p = p * mainSpeed;
35        }
36
37        p = p * Time.deltaTime;
38        Vector3 newPosition = transform.position;
39        if (Input.GetKey(KeyCode.Space))
40    {

```

Рисунок 2.8 Реалізація переміщення в ВЛ

Після того як користувач вибере і запустить лабораторну роботу, він отримає інформацію, необхідну для виконання лабораторної роботи. До цієї інформації належать: мета, перелік обладнання, хід роботи.

Після запуску лабораторної роботи користувач матиме доступ до довідкової інформації, який буде містити хід виконання лабораторної роботи та доступ до обладнання, яке буде використовуватися у ході лабораторної роботи.

Розглянемо лабораторну роботу «Дослідження руху тіла, кинутого горизонтально». Мета даної лабораторної роботи полягає у тому, щоб виміряти початкову швидкість, надану тілу в горизонтальному напрямку, під час його руху під дією сили тяжіння.

Наведемо фрагмент коду програми з описом математичної моделі кульки (рис. 2.9).

```

Sphere sphere;
Rigidbody sphere_rig;
SphereScript sphere_Script;

Сообщение Unity | Ссылки: 0
void Start()
{
    sphere = FindObjectOfType<Sphere>();
    sphere_rig = sphere.GetComponent<Rigidbody>();
    sphere_Script = sphere.GetComponent<SphereScript>();
}

Сообщение Unity | Ссылки: 0
private void FixedUpdate()
{
    Angle();
    Debug.Log(sphere_Script.collison);
}

ссылка: 1
void Angle()
{
    if (sphere_Script.collison == true)
    {
        AngleVector();
    }
    else
    {
        NormalFall();
    }
    Move();
}

ссылка: 1
private float Starting_speed(float lenght, float height, float g_acceleration)
{
    float starting_speed = lenght * Mathf.Sqrt(g_acceleration / 2 * height);
    return starting_speed;
}

ссылка: 1
private float End_time(float height, float g_acceleration)
{
    float end_time = Mathf.Sqrt(2 * height / g_acceleration);
    return end_time;
}

ссылка: 1
private float Height(float g_acceleration, float time)
{
    float height = (g_acceleration * (time * time)) / 2;
    return height;
}

```

Рисунок 2.9 Опис математичної моделі в коді програми

Під час руху кульки для кращого розуміння її руху додамо стрілку, яка буде вказувати вектор руху тіла.

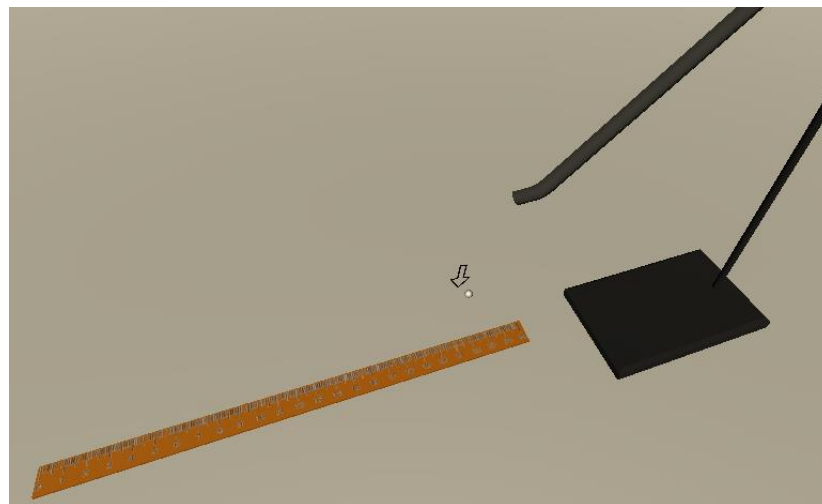


Рисунок 2.10 Вікно лабораторної роботи

Отримані результати в ході виконання експериментів будуть вноситися у таблицю, в кінці якої будуть подані всі виконані розрахунки.

Для лабораторної роботи «Дослідження руху тіла, кинутого горизонтально» таблиця матиме наступний вигляд (рис. 2.11).

Рисунок 2.11 Результати експериментів

Номер досліджу	Дальність польоту L , м	Середня дальність польоту l_p , м	Висота h , м	u_{oc} , м/с
1				
2				
3				
4				
5				

Далі користувачу залишається ввести відповідні дані до формул та знайти необхідні значення.

ВИСНОВКИ

Під час написання кваліфікаційної роботи мета, якої полягала у проектуванні віртуальної лабораторії для шкільного курсу фізики з розділу «рух і взаємодія. Закони збереження в механіці», було досягнуто наступних результатів:

Під час дослідження технологій моделювання 3D-об'єктів, було досліджені такі поняття як моделювання тривимірних об'єктів у комп'ютерній графіці. В ході дослідження було розглянуто найбільш поширене програмне забезпечення для створення 3D моделей. В якості основного програмного забезпечення для створення 3D об'єктів було обрано Blender, так-як він має гнучкий налаштовуваний та потужний функціонал, який дозволяє легко та швидко створювати 3D об'єкти. Було створено бібліотеку 3D об'єктів необхідних для створення віртуальної лабораторії. У процесі створення 3D об'єктів використовувався метод полігонального моделювання.

В якості середовища для створення віртуальної лабораторії було обрано рушій Unity3D оскільки в процесі його дослідження це середовище показало себе як досить гнучке, потужне та легке у налаштуванні. Unity3D володіє широким функціоналом, який дозволить легко спроектувати віртуальну лабораторію.

Під час створення віртуальної лабораторії було створено необхідні 3D об'єкти які були додані до середовища розробки. Всередині середовища було створено та додано скрипти до цих об'єктів, які описують їх поведінку та властивості. Було спроектовано віртуальну лабораторію, яка дає можливість виконувати лабораторні роботи з фізики з розділу «рух і взаємодія. Закони збереження в механіці» Також було розроблено інтерфейс користувача який надає доступ необхідної інформації користувачу в ході виконання лабораторної роботи.

В результаті було: досліджено технології створення 3D об'єктів, розроблено бібліотеку 3D об'єктів необхідних для створення віртуальної лабораторії, обрано та досліджено середовище для створення віртуальної лабораторії, спроектовано віртуальну лабораторію для шкільного курсу фізики з теми «рух і взаємодія. Закони збереження в механіці».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Кравцов Г.М., Баєв А.С., Лемещук О.І., Орлов В.В.
Мультимедійний редактор віртуальної фізичної лабораторії в системі дистанційного навчання «херсонський віртуальний університет» Херсон, 2017. 17 с.
2. Офіційна сторінка Unity 3D. – URL: <http://unity3d.com>.
3. Офіційна сторінка Unreal Engine. – URL: <https://www.unrealengine.com>.
4. Офіційна сторінка CryEngine. – URL: <https://www.cryengine.com>.
5. Офіційна сторінка Cocos2D. – URL: <https://www.cocos.com/en/>.
6. Robert Nystrom. Game Programming Patterns 1st Edition, 2011.
7. Roger Engelbert. Cocos2d-X by Example Beginner's Guide Paperback, 2013.
8. Calling RESTful APIs in Unity3D. – URL: <https://www.red-gate.com/simple-talk/dotnet/c-programming/calling-restful-apis-unity3d/>.
9. Justin Plowman. 3D Game Design with Unreal Engine 4 and Blender, 2016.
10. Steve Santello, Alan R Stagner. Building an RPG with Unreal 4.X, 2016.
11. Katax Emperore, Devin Sherry. Unreal Engine Physics Essentials Paperback, 2015
12. Sascha Gundlach, Michelle K. Martin. Mastering CryENGINE Paperback, 2014
13. Richard Gerard, Marcoux III, Chris Goodswen , Riham Toulan, Sam Howels. CRYENGINE Game Development Blueprints Paperback, 2015

14. Filip Lundgren, Ruan Pearce-Authers. CryENGINE Game Programming with C++, C#, and Lua Paperback, 2013.
15. Jeremy Sumner, Wycherley Gibson. Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game - With Unity and C# (Game Design and Development), 2014
16. Unity 5.x Cookbook: More than 100 solutions to build amazing 2D and 3D games with Unity(Englisch) Taschenbuch, 2015
17. Joe Hocking, Unity in Action: Multiplatform Game Development in C# with Unity 5 1st Edition, 2015.
18. Рекомендації геймдизайнеру від програміста. – URL: <https://habr.com/ru/post/134128/>.
19. Virtual Laboratory – виртуальна лабораторія URL: https://kai.ru/documents/683568/1374382/V_VLab.pdf/668f71d4-f14c-42f5-aa77-c59ce0319fd.
20. Виртуальные лаборатория. – URL: <https://kpfu.ru/docs/F324157708/Virtualnye.laboratorii.pdf>.
21. Офіційна сторінка Blender. – URL: <https://www.blender.org>
22. Офіційна сторінка AutoCAD. – URL: <https://www.autodesk.ru/products/autocad/overview>
23. Офіційна сторінка ArchiCAD. – URL: <https://graphisoft.com/ru/solutions/products/archicad>.
24. Офіційна сторінка Maya. – URL: <https://www.autodesk.ru/products/maya/overview>.
25. Офіційна сторінка Sketchup. – URL: <https://www.sketchup.com/ru>.

ДОДАТКИ

Додаток А

КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНЬСЬКОГО
ДЕРЖАВНОГО УНІВЕРСИТЕТУ

Я, Соценко Олександр Андрійович, учасник(ця) освітнього процесу Херсонського державного університету, **УСВІДОМЛЮЮ**, що академічна доброчесність – це фундаментальна етична цінність усієї академічної спільноти світу.

ЗАЯВЛЯЮ, що у своїй освітній і науковій діяльності **ЗОБОВ'ЯЗУЮСЯ**:

– дотримуватися:

- вимог законодавства України та внутрішніх нормативних документів університету, зокрема Статуту Університету;
- принципів та правил академічної доброчесності;
- нульової толерантності до академічного плагіату;
- моральних норм та правил етичної поведінки;
- толерантного ставлення до інших;
- дотримуватися високого рівня культури спілкування;

– надавати згоду на:

- безпосередню перевірку курсових, кваліфікаційних робіт тощо на ознаки наявності академічного плагіату за допомогою спеціалізованих програмних продуктів;
- оброблення, збереження й розміщення кваліфікаційних робіт у відкритому доступі в інституційному репозитарії;
- використання робіт для перевірки на ознаки наявності академічного плагіату в інших роботах виключно з метою виявлення можливих ознак академічного плагіату;

– самостійно виконувати навчальні завдання, завдання поточного й підсумкового контролю результатів навчання;

– надавати достовірну інформацію щодо результатів власної навчальної (наукової, творчої) діяльності, використаних методик досліджень та джерел інформації;

– не використовувати результати досліджень інших авторів без використання покликань на їхню роботу;

– своєю діяльністю сприяти збереженню та примноженню традицій університету, формуванню його позитивного іміджу;

– не чинити правопорушень і не сприяти їхньому скоєнню іншими особами;

– підтримувати атмосферу довіри, взаємної відповідальності та співпраці в освітньому середовищі;

– поважати честь, гідність та особисту недоторканність особи, незважаючи на її стать, вік, матеріальний стан, соціальне становище, расову належність, релігійні й політичні переконання;

– не дискримінувати людей на підставі академічного статусу, а також за національною, расовою, статевою чи іншою належністю;

– відповідально ставитися до своїх обов'язків, вчасно та сумлінно виконувати необхідні навчальні та науково-дослідницькі завдання;

– запобігати виникненню у своїй діяльності конфлікту інтересів, зокрема не використовувати службових і родинних зв'язків з метою отримання нечесної переваги в навчальній, науковій і трудовій діяльності;

– не брати участі в будь-якій діяльності, пов'язаній із обманом, нечесністю, списуванням, фабрикацією;

– не підроблювати документи;

– не поширювати неправдиву та компрометуючу інформацію про інших здобувачів вищої освіти, викладачів і співробітників;

– не отримувати і не пропонувати винагород за несправедливе отримання будь-яких переваг або здійснення впливу на зміну отриманої академічної оцінки;

– не залякувати й не проявляти агресії та насильства проти інших, сексуальні домагання;

– не завдавати шкоди матеріальним цінностям, матеріально-технічній базі університету та особистій власності інших студентів та/або працівників;

– не використовувати без дозволу ректорату (деканату) символіки університету в заходах, не пов'язаних з діяльністю університету;

– не здійснювати і не заохочувати будь-яких спроб, спрямованих на те, щоб за допомогою нечесних і негідних методів досягати власних корисних цілей;

– не завдавати загрози власному здоров'ю або безпеці іншим студентам та/або працівникам.

УСВІДОМЛЮЮ, що відповідно до чинного законодавства у разі недотримання Кодексу академічної доброчесності буду нести академічну та/або інші види відповідальності й до мене можуть бути застосовані заходи дисциплінарного характеру за порушення принципів академічної доброчесності.

18.04.2021

(дата)

(підпис)

Олександр Соценко

(ім'я, прізвище)