

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК, ФІЗИКИ ТА
МАТЕМАТИКИ
КАФЕДРА ІНФОРМАТИКИ, ПРОГРАМНОЇ ІНЖЕНЕРІЇ ТА
ЕКОНОМІЧНОЇ КІБЕРНЕТИКИ**

**МЕТОДИ ТА ТЕХНОЛОГІЇ СТВОРЕННЯ 3D ОБ'ЄКТІВ
НАВЧАЛЬНОГО ПРИЗНАЧЕННЯ ДЛЯ МОБІЛЬНИХ
ПРИСТРОЇВ**

Кваліфікаційна робота (проект)

на здобуття ступеня вищої освіти «бакалавр»

Виконав: студент 4 курсу 441 групи

Спеціальності: 121 Інженерія програмного
забезпечення

Освітньо-професійної програми:

Інженерія програмного забезпечення

Гордовенко Олександр Вікторович

Керівники: кандидат фізико-математичних
наук, доцент

Кравцов Геннадій Михайлович

кандидат фізико-математичних
наук, доцент

Єрмолаєв Вадим Анатолійович

Рецензентка: кандидатка фізико-
математичних наук, доцентка

Бистрянцева Анастасія Миколаївна

ЗМІСТ

ВСТУП	3
РОЗДІЛ 1. Аналіз предметної області	4
1.1. Основні поняття та терміни мультимедійних 3D об'єктів	4
1.2. Технології та етапи розробки мультимедійних 3D об'єктів	7
1.3. Програмне забезпечення для моделювання 3D об'єктів	16
1.4. Аналіз програмного забезпечення Unity 3D	19
РОЗДІЛ 2. Розробка та використання програмних модулів 3D об'єктів ..	24
2.1. Аналіз вимог та зовнішні специфікації	24
2.2. Розробка програмного забезпечення для віртуальних 3D об'єктів	27
2.3. Реалізація графічного інтерфейсу користувача	30
ВИСНОВКИ	33
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	34
ДОДАТКИ	
Додаток А	36

Вступ

Актуальність дослідження. У наш час неможливо уявити наше життя без цифрових технологій. Вся необхідна інформація знаходиться завжди у нас під рукою. Все це завдяки інтернету, який об'єднує в собі усі всесвітні джерела інформації. Тому користувач може без будь-яких зусиль може отримати інформацію з іншого кінця світу. Для користування цією інформацією потрібно мати відповідні пристрої такі як: комп'ютери, ноутбуки, смартфони, планшети тощо. Саме вони набирають все більшу популярність серед користувачів інтернету.

Саме для таких пристрої створюються різні навчальні додатки такі як віртуальні роботи і т.п. За допомогою віртуальних лабораторій користувачі можуть провести різні експерименти прямо у себе вдома перед комп'ютером або смартфоном. Переваг такого способу напевно більше ніж недоліків. Наприклад, можна не перейматися безпекою віртуальних лабораторій. Всі вибухи та реакції будуть відбуватися тільки у віртуальному середовищі і ви можете не боятися зробити щось не так при цьому можливо вдатися до будь-якого експерименту. Окрім безпеки і комфорту слід зазначити фінанси які не потрібно витратити на купівлю реагентів та обладнання, які коштують чималих коштів. Крім усього іншого, віртуальна лабораторна може бути використана на інтерактивних уроках в школах чи університетах.

Предмет та об'єкт дослідження: Методи та технології розробки 3D об'єктів навчального призначення для мобільних пристроїв.

Мета дослідження: спроектувати 3D об'єкти навчального призначення для мобільних пристроїв. Досягнення поставленої мети потребує вирішення таких **завдань:**

- Розробити бібліотеку об'єктів для мобільного додатку.
- Реалізувати мобільний додаток.
- Протестувати мобільний додаток.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1. Основні поняття та терміни мультимедійних 3D об'єктів

Сьогодні, як ніколи, еволюція спрямована у віртуальний світ завдяки процвітаючому розвитку інноваційного програмного забезпечення та технологій та підготовці кваліфікованого персоналу, здатного використовувати свій потенціал. Все це змінює суспільство, надаючи йому нове обличчя та нові засоби для задоволення його потреб. Яскравим прикладом є 3D-моделювання, віртуальна техніка проектування, яка знаходить застосування у найрізноманітніших галузях і розвивається щодня завдяки роботі кваліфікованих фахівців: експертів з 3D-моделювання.

Що таке 3D моделювання?

За визначенням це процес моделювання тривимірної фігури (3D-моделі) шляхом вставки її у віртуальний простір, що генерується комп'ютером за допомогою спеціальних програм, що називається програмним забезпеченням 3D. Операцій та модифікацій, які це програмне забезпечення може застосувати до моделей, незліченна кількість, починаючи від первинного моделювання, яке розвивається під час будівництва, шляхом таких дій, як Наплавлення (визначення поверхневих матеріалів), картографування, нанесення текстури, визначення постановки, освітлення та відповідне затінення, все для того, щоб оптимізувати остаточний рендерінг самої моделі.

Історія комп'ютерної графіки 3D, звісно, зовсім недавня, той самий термін комп'ютерна графіка народився лише в 1960 році.

Одним із перших тривимірних зображень на комп'ютері було зображення відомого «першої людини» або «Боінга», зробленого

Вільямом Феттером; набір рядків, що описують віртуальний силует пілота літака.

Починаючи з 1959 року, General Motors у співпраці з IBM розробила систему "ЦАП", одну з перших САПР; за допомогою пера та сенсорного екрану оператори малювали математичні криві у віртуальному просторі, за допомогою яких вони розмежовували профілі, секції та поверхні автомобілів. Система "Adage", яку багато хто вважає першою незалежною робочою станцією САПР, також з першої половини шістдесятих.

Отже, зародження 3D-моделювання відбулося у промисловій галузі, насамперед як підтримка дизайну. З тих пір сфери використання тривимірного моделювання та тривимірної графіки надзвичайно розширились, значною мірою залишивши технічну сферу.

Види 3D моделювання

З типологічної точки зору, все 3D-моделювання ділиться на дві великі родини, кожна з яких стосується конкретного типу моделі:

Органічне моделювання - це типове моделювання, яке використовується для створення людей чи істот, тварин чи гуманоїдів. Він використовується для всіх "природних" предметів, таких як скелі, рослини, дерева та для території загалом, в цих випадках моделі тим успішніші, чим більше вони багаті на деталі. Багато об'єктів промислового дизайну, що мають м'які та округлі форми, також можуть використовувати органічне моделювання.

Геометричне моделювання - це найдавніший вид моделювання, він використовується для створення технічних або механічних об'єктів, або в будь-якому випадку для будь-чого, що має штучну природу, і не підпадає під попередню категорію. Як правило, складність моделей, виготовлених за допомогою цього виду моделювання, набагато нижча,

якщо ми розглядаємо зовнішній вигляд окремих форм, але не якщо враховуємо аспекти, що стосуються точності та відповідності деталей.

Звичайно, один і той же об'єкт може містити як органічне, так і геометричне моделювання, або він може бути утворений набором частин, що містять як органічні, так і геометричні моделі.

Сфери використання 3D моделювання

Системи моделювання використовуються у всіх сферах комп'ютерної 3D-графіки, настільки, що в деяких випадках 3D-моделювання та 3D-графіка є синонімами.

- Наукові або технічні програми
- Математичні, фізичні та природничі науки (біологія, фізика, математика, астрономія тощо)
- Територіальне вивчення (геологія, сейсмологія, метеорологія тощо)
- Історичні науки (археологія, палеонтологія, палеоантропологія тощо)
- Прикладні науки
- Медицина (судово-медичні, реконструктивні, діагностичні дослідження тощо)
- Цивільна інженерія
- Промислове будівництво
- Архітектура
- Індустріальний дизайн
- Конструкція механічних деталей
- Художні додатки
- Кіно і телебачення
- Відеоігри та додатки для відеоігор
- Рекламна графіка
- Редакційні публікації

- Веб дизайн
- Мультимедійні програми
- Художнє виробництво

1.2. Технології та етапи розробки мультимедійних 3D об'єктів

Техніки 3D моделювання

Їх можна розділити на три основні категорії:

Процедурне моделювання (автоматичне та напіваавтоматичне)

Ручне моделювання

З даних реальних моделей (тривимірне сканування)

Що, в свою чергу, можна розділити на три різні жанри моделювання:

Моделювання твердих тіл - де отриманий об'єкт вважається суцільним об'ємом.

Об'ємне моделювання - визначає сутності, що генерують неявну поверхню.

Поверхнєве моделювання - об'єкт у цьому випадку визначається зовнішніми його поверхнями.

У деяких моделях об'єкт вважається утвореним поверхнями, поки вони відкриті, тоді як він розпізнається як твердий, коли всі поверхні зварені між собою і утворюють закрите тіло.

У наведеному нижче списку розглядаються різні методи ручного моделювання. Деякі з описаних методів (наприклад, поверхні латок), будучи досить застарілими, є застарілими та застарілими у порівнянні з новітніми та вдосконаленими методами. Незважаючи на це, деякі 3D-модельєри зберігають деякі з цих інструментів усередині як аксесуари чи додаткові засоби.

Основні конструкції (тверді тіла та поверхні)

Примітиви - Як правило, кожен пакет 3D, який займається не лише рендерингом, містить у собі більш-менш великий набір примітивів,

тобто заздалегідь визначених об'єктів (твердих тіл або поверхонь), безпосередньо використаних; зазвичай стандартними примітивами, які є завжди, є: площина, куб / паралелепіпед, сфера, циліндр, конус / піраміда, бик і часто чайник.

Створення за допомогою екструзії - це простий спосіб виготовлення екструдованих фігур, починаючи з основного 2d-креслення або плоского багатокутника, і присвоюючи йому певну висоту та напрямок екструзії.

Створення за допомогою обертання - більш складна, ніж попередня конструкція, оберт можна розглядати як видавлювання навколо осі, він завжди починається з профілю або базового багатокутника, а замість глибини призначається кут обертання.

Патч-поверхні. Моделювання патчів - одна з найдавніших систем, що застосовуються в 3D-графіці, і з часом вона значно розвинулася завдяки розробці ряду нових методів. У найпростішій формі він визначає параметричні поверхні, породжені чотирма або більше сусідніми кривими, що утворюють замкнутий периметр.

Поверхня Кунса – визначає інтерпольований патч між чотирма кривими кривих, маючи спільні вершини, перший алгоритм цього класу поверхонь був розроблений Стівеном А. Кунсом в 1967 році.

Пластина Безьє – це параметрична поверхня, якою можна управляти локально через сітку контрольних точок, об'єднуючи більше патчів Безьє, ви отримуєте більш складні поверхні, звані сплайн-поверхнями, в цьому випадку контрольні точки розташовані на перетині між різними плямами.

Моделювання сплайну (поверхні). Моделювання сплайну використовує описану раніше техніку виправлення та сплайн-криві. В основному модель, виготовлена за допомогою цієї системи, утворена клітиною сплайн-кривих, що перетинаються і з'єднані між собою. Потім

проміжки між трьома або чотирма сплайн-кривими, з'єднаними в їх контрольних точках, заповнюються плямами. Ця система піддається насамперед створенню органічних моделей.

Досконалі конструкції (тверді речовини та поверхні)

Вдосконалені конструкції використовують ту саму концепцію екструзії та простого обертання, але додають набагато складніші засоби управління.

Лофт Форма об'єкта надається поверхнею з'єднання ряду профілів, розташованих уздовж певного шляху, з можливістю управління поверхнею через напрямні криві, в яких профілі складають ділянки отриманого об'єкта.

Гібридна система між оборотом та розгорткою, ми завжди починаємо з профілю та осі обертання, але ми також додаємо базовий шлях (також званий рейкою), якого профіль повинен слідувати під час Революції.

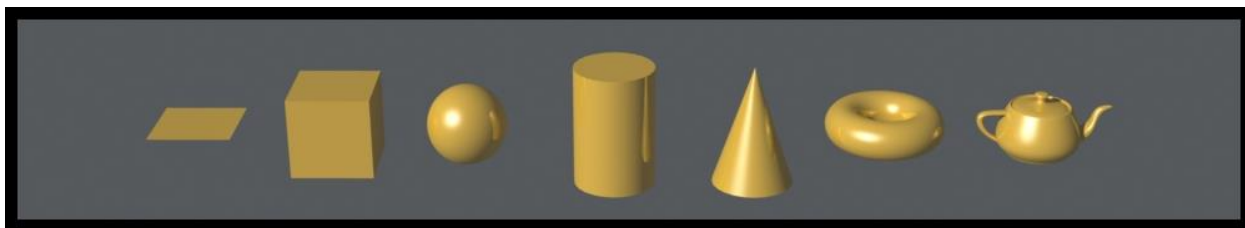


Рисунок 1.1 - Прості 3D об'єкти

Протягом багатьох років подібні конструкції насправді були оснащені багатьма типами засобів управління, були додані додаткові вказівки, елементи контролю типу скручування, визначення дотичних та ін.

Ці системи моделювання, завдяки високій точності, в основному використовуються для визначення об'єктів технічного або промислового дизайну.

Полігональне моделювання. Це основні прийоми у галузі тривимірної графіки. Полігональне моделювання працює на поверхнях,

організованих у більш-менш деталізовані сітки полігональних граней. Ці поверхні можуть наблизити кінцевий об'єкт лише в тому випадку, якщо ми знаходимось у полігонах низького рівня (у цьому випадку об'єкт називається Low Poly). В інших випадках полігональна модель - коли моделювання завершено - також може бути сформована дуже великою кількістю граней.

Наступні системи переходять від найпростіших до найдосконаліших:

За допомогою рухомих елементів - полігональна модель складається з 3 основних елементів: граней, сторін і вершин; довільне переміщення окремого елемента або їх груп визначає модифікацію вихідної сітки. Вибір компонента сітки та переміщення її (перетягування, обертання, зміна розміру тощо) у просторі є найосновнішим методом полігонального моделювання.

З базових примітивів - один із найпростіших і найпряміших способів почати моделювання полігонального об'єкта - це почати з базового полігонального примітиву і почати його модифікувати, рухаючи, обертаючи, масштабуючи його компоненти, поки не отримаєте потрібну форму. Цей прийом дуже простий, але загалом дозволяє отримати моделі, які не дуже складні, тобто обмежені складністю (також з точки зору полігональної щільності сітки) вихідного примітиву.

Метод площинної сітки - на додаток до модифікації існуючих полігонів сітки (наприклад, примітивів), існує можливість створити кожен полігон об'єкта окремо та побудувати полігони в найбільш зручному положенні для створення кінцевої моделі. Одну із систем прямого малювання багатокутників називають методом площинної сітки. По суті, мова йде про створення сітки багатокутників, розташованих на площині, що мають структуру, профіль та загальну конформацію кінцевого об'єкта. Після розташування багатокутників на площині ми переходимо до визначення їх тривимірності: або

переміщенням точок сітки по глибині моделі, або за допомогою екструзійних систем.

Метод павутини - це різновид попередньої техніки. У цьому випадку всі основні багатокутники моделі не будуються і не розташовуються, але ми починаємо з однієї з її (центральної) областей, і починаємо створювати та моделювати окремі багатокутники за допомогою системи "павутинної павутини", тобто зсередини і поступово рухаючись до зовнішніх ділянок моделі. Це складна і трудомістка система, що використовується в основному за високий ступінь точності.

Прогресивна обробка - це найдосконаліша система, її можна вважати однією з парадигм 3D-моделювання. Застосовуючи будь-який аналізований раніше метод, ви починаєте визначати форму дуже схематично, здебільшого шляхом апроксимації її морфології та подбаючи про те, щоб початкова кількість полігонів була надзвичайно низькою. Маючи можливість керувати кількома багатокутниками, можна дуже легко змінити пропорції та загальний об'єм фігури.

Тільки коли вас влаштовує необроблений зовнішній вигляд моделі, ви можете почати - застосовуючи конкретні інструменти кожного програмного пакета - до подальшого визначення форми. Важливо, щоб на кожному завершальному етапі спочатку визначали більші обсяги моделі, а потім визначали все менші та менші площі, визначення та кількість деталей, які можуть бути надані, на розсуд тривимірного графіку. Принциповий принцип, про який слід пам'ятати, полягає в тому, що: чим менша кількість багатокутників, присутніх у моделі, тим більша можливість модифікації загальної морфології - чим більша кількість багатокутників, тим менше буде можливо змінити вже задану форму раніше.

На практиці кожен крок незворотний, чим більше деталей об'єкта визначено, тим менше буде можливо змінити (або виправити) його

загальний вигляд. Цю проблему можна усунути, поступово зберігаючи модель, щоб мати доступні всі проміжні етапи моделювання. У разі помилки ви можете перезапустити попередню модель менш детально, якщо використовуване програмне забезпечення використовує шари, Ви можете зберігати різні версії окремими шарами.

Твердотільне моделювання

Твердотільне моделювання, або CSG, є різновидом геометричного моделювання, що використовується в основному в технічній галузі та області САПР. До 1981 р., а потім Граніт РТС з виходом Pro / ENGINEER в 1987 р., Який представив концепцію параметричного 3D, і Parasolid, Unigraphics, у 1988 р.

Тверде моделювання використовує такі основні інструменти:

Основні примітиви - це ті самі примітиви, проаналізовані вище.

Конструкції методом екструзії та обертання, прості та вдосконалені - також у цьому випадку твердотільні моделі використовують ті ж методи, описані вище.

Булеві операції - похідний від булевої алгебри, цей метод замість цього є ексклюзивним для твердого моделювання. Це дозволяє отримувати складні фігури, починаючи з твердих примітивів, складаючи їх разом за допомогою трьох операцій: об'єднання, віднімання та перетину.

Фаски та філе - це автоматичні функції, які працюють уздовж країв твердих тіл, дозволяючи їх філетувати певним радіусом або скошувати під певним кутом.

Твердотільне параметричне моделювання

Різновидом класичного моделювання твердих тіл, який дав великий поштовх для розробки САПР, стало введення концепції параметричного твердого тіла на основі ознак, яке відрізняється від концепції традиційного моделювання твердого тіла тим, що тверде тіло

створюється на основі порівнянних операцій з тими, які потрібно зробити, щоб зробити справжній шматок.

Крім того, при параметричному моделюванні тверді речовини регулюються та пов'язані між собою фізичними, математичними або геометричними параметрами.

Впровадження цієї системи концепцій тривимірного моделювання зазвичай пояснюється розробками доктора Семюеля П. Гейсберга, які призвели до випуску системи Pro / ENGINEER в 1987 році.

У поточній панорамі промислових пропозицій САПР більшість найпопулярніших програм застосовують цю парадигму, хоча існують відмінності в підході до моделювання параметрів, управління подіями та математичних та комп'ютерних технологій для їх реалізації.

Меташар. Форма об'єкта, змодельована за допомогою сферичних метакульок

Меташари (або "краплі") - це особливий тип примітивів, що використовуються для створення органічних моделей, дизайну або моделювання рідин. Вони є об'ємними сутностями, такими як вокселі, мають центральне ядро, яке відображається як неявна поверхня і зовнішня сила або поле "впливу". Коли дві метаболи об'єднані, вони реагують між собою через зовнішнє силове поле, яке їх приваблює (якщо воно позитивне) і визначає їх злиття, або відкидає (якщо воно негативне), і викликає віднімання об'єму.

Як тільки модель побудована як неявна поверхня, як правило, можна перетворити її на справжню багатокутну сітку, викликаючи такі параметри, як кінцева щільність сітки. Було розроблено кілька форм Metaball:

Сферичні метакульки - це метакульки у своїй природній формі, прив'язані до цієї геометрії, вони мають недолік, у випадку складних

органічних моделей, великої кількості сутностей, які потрібно розташувати, особливо за наявності витягнутих та еластичних форм.

Меташари з іншою геометрією - використовуючи ті самі принципи, що і сферичні метакульки, ці сутності можуть приймати форму інших типів примітивів і мають ту перевагу, що дозволяють наблизити кінцеву форму за допомогою набагато меншої кількості сутностей.

Метамузики - ці геометричні сутності представляють одну з основних еволюцій метакульок. Вперше вони були представлені REM Infogràfica як плагін для 3DS Max, що називається MetaReyes у редакції 3.1. Метам'язи - це деформуються метакульки на сплайн-трактах; модифікуючи контрольні точки сплайну та параметри метаболів, отримують видовжені форми, що наближають м'яз, взаємодія різних метам'язів, згідно типових методів метаболів, створює остаточну форму. Цінність і межа цих примітивів полягає в їх спеціалізації у визначенні майже виключно анатомічних форм.

Тверде параметричне моделювання. Це свого роду моделювання, що використовується в області САПР.

Параметричне моделювання твердих тіл у САПР заповнило деякі недоліки простого моделювання твердого тіла. Це дозволяє генерувати тверді речовини, вводячи числові параметри, наприклад. вимірювання висоти, довжини, глибини, радіусів та кутів, а також мати можливість втручатися в ці параметри навіть після виготовлення моделі, змінювати та оновлювати геометрію без необхідності її перебудови. На додаток до моделювання з використанням параметрів, було також введено поняття "ознака" та дерево побудови: на практиці всі механічні обробки, застосовані до твердотільної моделі, реєструються (як ознака) у своєрідному хронологічному дереві, яке працює відповідно до схема залежності батько син;

У будь-який час під час моделювання можна повернутися в дерево побудови, вибрати об'єкт, відредагувати та змінити його параметри та оновити всю модель новими параметрами. Параметричне моделювання твердих тіл зараз називають «гібридним», оскільки багато моделей САПР додали вдосконалені функції гібридного твердотільного та поверхневого моделювання для створення більш складних моделей.

3D скульптура. Термін "фарбування переміщенням" також використовується для позначення цієї техніки, оскільки вона походить від спільності методів карти переміщення та техніки 3D-малювання.

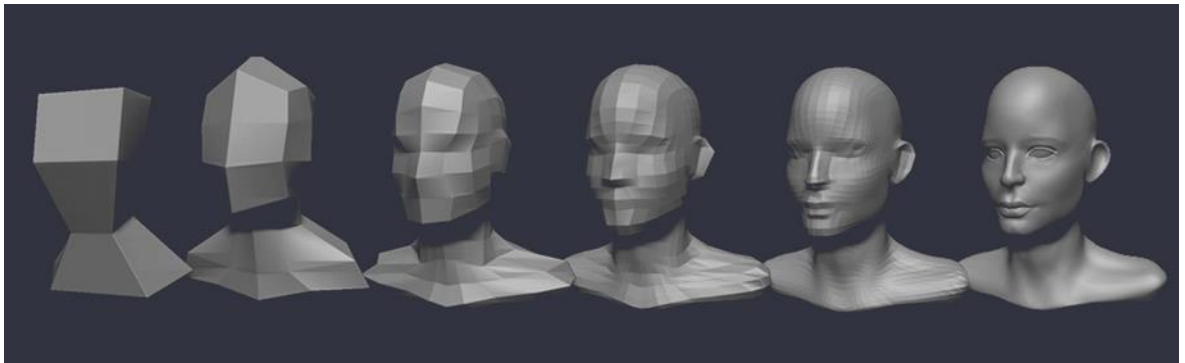


Рисунок 1.1 - Етапи 3D скульптури

Це система, дуже схожа на традиційні техніки скульптури, вона працює за допомогою віртуальних пензлів, змінних розмірів та функцій, які, проходячи по поверхні моделі, змінюють її геометрію в режимі реального часу, викликаючи виступи, западини, подряпини та надрізи, як якщо ви працювали над шматком глини.

Попередниками цієї техніки були програми прямого фарбування на сітці, які, однак, не працювали на каналі переміщення. Першим прикладом цього типу скульптури був модуль "ремісник", який використовувала Майя, але справжнім родоначальником моделей, заснованих на цій технології, безсумнівно, є ZBrush від Pixologic, за яким слідує низка інших комерційних пакетів.

Пряме ліплення з сітки в основному використовується для обробки та деталізації простих моделей високої чіткості, виготовлених іншими методами, але також може використовуватися для визначення моделі з нуля, починаючи з простих примітивів, таких як коробки або сфери. Він значною мірою використовується в органічному моделюванні, особливо в моделюванні та визначеннях характеру.

З огляду на надзвичайну складність моделей, отриманих за допомогою цієї техніки (які можуть бути сформовані багатьма мільйонами багатокутників), майже завжди необхідно переносити дані тривимірної сітки на карти переміщень або звичайні карти, які можна використовувати набагато легших моделі

1.3. Програмне забезпечення для моделювання 3D об'єктів

На сьогоднішній час існує широкий спектр програмного забезпечення для моделювання 3D об'єктів. Незалежно від проекту який хочете створити, потрібно підібрати програмне забезпечення, яке допоможе, як найкраще, його реалізувати. Був підібраний список кращих програм, на які звернемо увагу. Кожен з цих програм володіє високо ефективними функціями та інструментами для 3D моделювання.

3DS Max. 3D Studio Max або 3DSMAX - це програмне забезпечення для 3D-моделювання, анімації та рендерингу, яке є найкращим у світі. Він почав використовувати комп'ютерну ігрову анімацію, і більшість із них придумали виробництво спецефектів для фільмів та телешоу, таких як "Люди Ікс" та "Останній самурай".

Maya. Maya - найкраще програмне забезпечення для 3D-анімації у світі для професійних рекламних роликів у кіно та ТБ, анімації персонажів, кіноакробатики тощо. Maya - це виробниче програмне забезпечення високого рівня на кінематографічному рівні, багате на

функції, гнучке, просте у вивченні та використанні, високоефективне, що забезпечує сильне відчуття реальності.

Blender. Blender - це мультиплатформене і повне програмне забезпечення для 3D-анімації з відкритим кодом, яке пропонує цілий ряд анімаційних рішень, починаючи від моделювання, анімації, матеріалів, рендерингу, закінчуючи обробкою звуку, редагуванням відео тощо. Ігровий двигун у режимі реального часу дозволяє створювати 3D-інтерактивний контент для незалежного відтворення.

За допомогою Blender гравці, які люблять тривимірну графіку, можуть створювати свої улюблені 3D-моделі, не витрачаючи грошей. Він не тільки підтримує всі типи полігонального моделювання, але й може оживляти!

SketchUp. SketchUp - це інструмент для малювання, який безпосередньо керує процесом створення резюме малюнка. Це чудовий інструмент для створення тривимірних архітектурних креслень, який не тільки виражає ідеї дизайнера, але й дозволяє дизайнеру безпосередньо спілкуватися з клієнтом.

SketchUp - надзвичайно популярне та просте у використанні програмне забезпечення для 3D-дизайну, яке офіційному веб-сайту подобається "олівець" електронного дизайну. Його основний момент продажу полягає в тому, що він простий у використанні, і кожен може швидко розпочати роботу. І користувачі можуть використовувати SketchUp для створення 3D-моделей безпосередньо для GoogleEarth, справді круто!

ZBrush. ZBrush - це програмне забезпечення для цифрової скульптури та живопису, яке відоме своїми потужними функціями та інтуїтивно зрозумілим робочим процесом. Він має простий інтерфейс, плавне функціонування та поєднання функцій, розроблених з

практичними ідеями, які надихають на творчість і дають художникам свободу творити без обмежень.

Це повністю скасовує те, як працювали традиційні інструменти тривимірного дизайну в минулому, звільняючи руки та думки художників, прощаючись із минулим способом покладатися на мишу та параметри для незручного створення, і повністю поважаючи творче натхнення та звички дизайнерів традиційні умови праці .

Microsoft Visual Studio - це інтегроване середовище розробки (IDE) від Microsoft. Застосовується для розробки комп'ютерних програм, а також веб-сайтів, веб-програм, веб-сервісів та мобільних додатків. Visual Studio використовує платформи розробки програмного забезпечення Microsoft, такі як Windows API, Windows Forms, Windows Presentation Foundation, Windows Store та Microsoft Silverlight. Він може створювати як власний код, так і керований код.

Visual Studio включає редактор коду, що підтримує IntelliSense (компонент завершення коду), а також рефакторинг коду. Інтегрований налагоджувач працює як налагоджувач вихідного рівня, так і налагоджувач машинного рівня. Інші вбудовані інструменти включають програміст коду, дизайнер для побудови графічних інтерфейсів, веб-дизайнер, дизайнер класів та конструктор схем баз даних. Він приймає плагіни, які розширюють функціональність майже на кожному рівні, включаючи додавання підтримки для систем керування джерелами (таких як Subversion та Git) та додавання нових наборів інструментів, таких як редактори та візуальні дизайнери для мов, що належать до домену, або наборів інструментів для інших аспектів життєвий цикл розробки програмного забезпечення (як клієнт Azure DevOps: Team Explorer).

Visual Studio підтримує 36 різних мов програмування та дозволяє редактору коду та налагоджувачу підтримувати (різною мірою) майже

будь-яку мову програмування, за умови, що існує спеціальна послуга. Вбудовані мови включають C, C++, C++ / CLI, Visual Basic .NET, C#, F#, JavaScript, TypeScript, XML, XSLT, HTML та CSS. Підтримка інших мов, таких як Python, Ruby, Node.js та M, серед інших, доступна через плагіни. Раніше підтримувались Java (і J#).

Найпростіший випуск Visual Studio, випуск Community, доступний безкоштовно. Гасло видання Visual Studio Community - "Безкоштовна повнофункціональна IDE для студентів, розробників з відкритим кодом та окремих розробників".

Станом на березень 2021 року поточною готовою до виробництва версією Visual Studio був 2019 рік, зі старими версіями, такими як 2012 та 2013 на розширеній підтримці, та 2015 та 2017 на основній підтримці.

1.4. Аналіз програмного забезпечення Unity 3D

Unity - це середовище розробки для відеоігор, створений компанією Unity Technologies. Який доступний як платформа для скачування для Microsoft Windows, Mac OS, Linux. Платформа платформи підтримує збірку на різних типах платформ (лінзовий сегмент платформ лінз). Щоб запустити свою версію 5.4.0 і не завантажувати вміст для перегляду у свій веб-плагін, ви можете використовувати WebGL. Він має дві версії: Unity Professional (pro) та Unity Personal.

Редактор Unity має простий Drag & Drop інтерфейс, який легко налаштовується, складається з різних вікон, завдяки чому можна зробити відладку гри прямо в редакторі. Двигун використовує для написання сценаріїв C#.

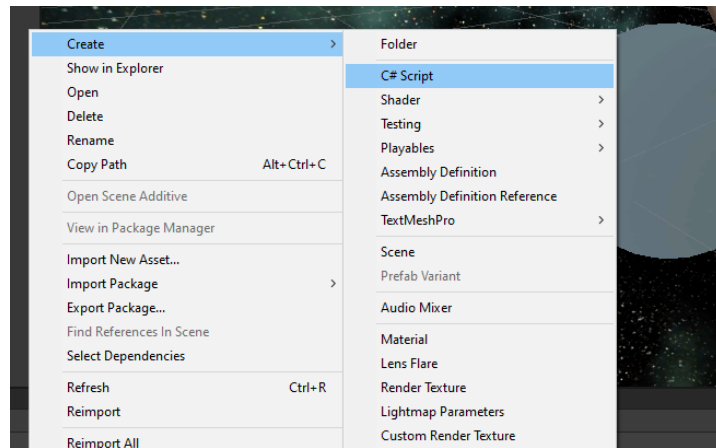


Рисунок 1.1 - Створення C# сценарію

Раніше підтримував також Boo (діалект Python, підтримку прибрали в 5-й версії) та модифікацію JavaScript, відома як UnityScript 2017.1) Розрахунки фізики виконує фізичний модуль PhysX від NVIDIA. Графічний API - DirectX (на даний момент DX 11, підтримується DX 12).

Проект в Unity розділяється на сцени (рівні) - окремі файли, які містять свої ігрові світи зі своїми наборами об'єктів, сценаріїв та налаштувань. Сцени можуть містити в собі як власні об'єкти (моделі), так і пусті ігрові об'єкти - об'єкти, які не мають моделей. Також на об'єктах є назва (в Unity допускається наявність двох та більше об'єктів з однаковими назвами) яка може бути тегом (міткою) і шаром, в якому він повинний відображатись.

Будь-який об'єкт, що присутній на сцені, обов'язково має компонент Transform, що зберігає в собі дані об'єкта, такі як: розміри об'єкта, повороту, та координати точок місця розташування на всіх трьох осях. Також об'єкти, які мають видимої геометрію, присутній за замовчування компонент Mesh Render, який робить їх видимими.

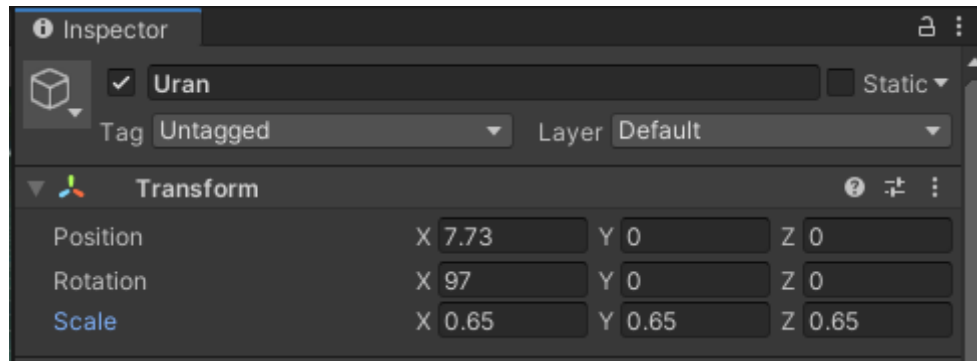


Рисунок 1.2 - Застосування компонента Transform

До об'єктів можна застосовувати колайдери (в Unity т. Н. Колайдери - collider), яких існує декілька типів.

Середовище Unity має підтримку фізики твердих тіл і тканин, а також фізики типу Ragdoll (ганчірна лялька). У редакторі присутня система успадкування об'єктів; дочірні об'єкти будуть наслідувати всі зміни позиції, повороту і масштабу батьківського об'єкта. Сценарії в редакторі закріплюються до об'єктів у вигляді окремих компонентів.

При імпорті текстури в Unity можна згенерувати alpha-канал, mip-рівні, normal-map, light-map, карту відображень, проте безпосередньо на модель текстуру прикріпити не можна - буде згенеровано матеріал, до якого буде призначений шейдер, і потім матеріал прикріплюється до моделі. Редактор Unity підтримує створення і редагування шейдерів. Редактор Unity має інструмент для створення анімацій, але водночас анімацію можна створити безпосередньо в 3D-редакторі та імпортувати разом з моделлю, а потім розділити на файли.

Unity 3D включає в себе систему Level Of Detail (скор. LOD), суть якої полягає в тому, що при віддаленні від гравця, високодеталізовані моделі замінюються на менш деталізовані, і навпаки, а також систему Occlusion culling, суть якої в тому, що об'єкти, які не потрапляють в поле зору камери не візуалізується геометрія і колізія, що зменшує навантаження на центральний процесор і дає змогу оптимізувати проект.

Під час компіляції проекту створюється виконуваний файл (.exe) програми (для Windows), а в окремій папці - дані програми (включаючи всі ігрові рівні і спільні бібліотеки).

Двигун підтримує майже всі популярні формати. Моделі, звуки, текстури, матеріали, можна передавати іншим розробникам, або викладати у вільний доступ, попередньо запакувавши у формат .unitypackage. Цей формат також використовується у внутрішньому магазині Unity Asset Store, в якому розробники можуть безкоштовно і за гроші викладати в загальний доступ різні елементи, потрібні при створенні ігор. Щоб користуватися Unity Asset Store, необхідно мати обліковий запис розробника Unity. Unity має всі необхідні компоненти для створення мультиплеєра. UNet був видалений, починаючи з версії 2018.4; рішення «з коробки» для мультиплеєра відсутня. Також, користувачеві, можна використовувати, зручний для нього спосіб контролю версій. Наприклад, Tortoise SVN, Git або Source Gear.

Також в екосистему Unity входить інструментарій для командної розробки, під назвою Unity Asset Server, що є доповненням, яке додає контроль версій і ряд інших серверних рішень.

Зазвичай, ігровий двигун дозволяє задіяти безліч функціональних можливостей в різних проектах, в які входять моделювання фізичних середовищ, процесів та явищ, динамічні тіні, карти нормалей і багато іншого. На противагу від інших багатьох ігрових движків, Unity має дві вагомні переваги: наявність візуального середовища розробки та підтримка міжплатформенності. Крім інструментарію візуального моделювання, перший фактор включає в себе, ще й інтегроване середовище, що сприяє підвищенню продуктивності розробників, зокрема, етапів тестування та створення прототипів. Під міжплатформенною підтримкою мається на увазі не тільки місце розгортання (установка на персональному комп'ютері, на мобільному

девайсі, консолі і т. Д.), але і наявність інструментарію розробки (інтегроване середовище може використовуватися під Windows і Mac OS).

Модульна система компонентів Unity – являється третьою перевагою, за допомогою якої відбувається конструювання ігрових об'єктів, коли останні є комбінованими пакетами функціональних елементів. На відміну від механізмів успадкування, об'єкти в Unity створюються за допомогою об'єднання функціональних блоків, а не переміщення в вузли дерева успадкування. Такий підхід актуальний при розробці ігор, тому що полегшує створення прототипів.

Недоліком цієї платформи є обмеження візуального редактора при роботі з багатокомпонентними схемами, коли в складних сценах візуальна робота ускладнюється.

Другий недолік - це відсутність підтримки Unity посилань на зовнішні бібліотеки, роботу з якими програмістам доводиться налаштовувати самостійно, і це також ускладнює командну роботу.

Останній недолік пов'язаний з використанням шаблонів примірників (англ. Prefabs). З одного боку, ця концепція Unity пропонує гнучкий підхід візуального редагування об'єктів, але з іншого боку, редагування таких шаблонів є складним. Також, WebGL-версія двигуна, в силу специфіки своєї архітектури (трансляція коду з C# в C++ і далі в JavaScript), має ряд невирішених проблем з продуктивністю, споживанням пам'яті і працездатністю на мобільних пристроях.

Таблиця 1.2

Опис кнопок панелі інструментів Unity

Найменування	Опис
Create menu	Відображає список Асетів та інших вкладених папок, які можна додати в обрану в даний момент папку.
Search bar	За допомогою рядка пошуку можна шукати файл в рамках проекту. Можна здійснювати пошук у всьому проекті, в папках верхнього рівня проекту (перераховані окремо), в папці, яку ви вибрали в даний момент, або в Сховище активів.
Search by Type	Вибір цієї властивості обмежить пошук відповідно до певних типів, наприклад Mesh, Prefab, Scene.
Search by Label	Вибір цієї властивості потрібен для підбору тега з метою його пошуку.
Hidden packages count	Ця властивість використовується для перемикання видимості пакетів у вікні проекту.

РОЗДІЛ 2 РОЗРОБКА ТА ВИКОРИСТАННЯ ПРОГРАМНИХ МОДУЛІВ 3D ОБ'ЄКТІВ

2.1. Аналіз вимог та зовнішні специфікації

При розробці програмного забезпечення для реалізації 3D моделей

Сонячної системи, потрібно створити об'єкти які будуть містити фізичні властивості (компонент Rigidbody). Всі об'єкти будуть створюватися у програмному середовищі Unity 3D. Перед створенням проекту необхідно проаналізувати його структуру, яка має фізичні моделі. Давайте побудуємо структуру 3D моделі з фізичними властивостями.

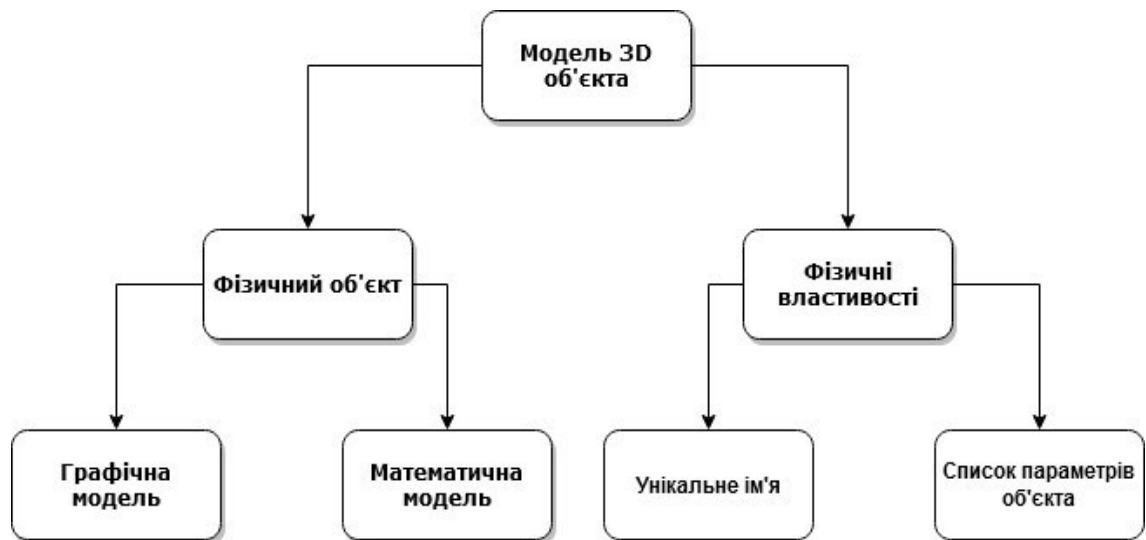


Рисунок 2.1 - Структура фізичного об'єкта проекту

Проект складається з окремих об'єктів які містять фізичні властивості такі як: маса, радіус, сила тяжіння і т.д. Всі модель мають своб математичну модель. Потрібно врахувати те що користувач повинен мати можливість змінювати математичну модель самостійно за допомогою інтерфейсу користувача. Під зміною математичної мається на увазі зміна фізичних властивостей об'єкта та окремих його параметрів. Тому головним сенсом віртуальної лабораторної роботи – це можливість змінювати фізичні властивості і параметри.

Таким чином користувач матиме можливість гнучко проводити дослідження в лабораторних роботах змінюючи потрібні вхідні параметри і змінні. В процесі розробки до кожного об'єкту буде

додаватися його унікальне ім'я, яке дозволить користувачам звертатися до тих самих об'єктів, які будуть використовуватися віртуально в роботі. І так, що нам необхідно зробити:

Для початку, у віртуальну сцену Unity нам необхідно додати 3D об'єкт, котрий ми створили за допомогою спеціального програмного забезпечення, в режимі 3D. Надати унікальне ім'я об'єкту

Для того, щоб була можливість використовувати ці об'єкти, під час виконання віртуальних лабораторних робіт, необхідно задати необхідні нам параметри.

Додамо додатковий масив констант та змінних, такі як: маса тіла, сила тяжіння, швидкість руху, сила тертя, ітак далі.

Щоб уникнути не состиковок і помилок під час виконання завдань, нам необхідно додати обмеження для користувачів, на зміну параметрів математичних моделей 3D об'єкта

В графічний інтерфейс самого програмного забезпечення, можна також включити спеціальні датчики або лічильники які будуть точно вказувати результати вичислення сили, відстані тощо.

Так само необхідно прописати спеціальні формули для роботи програмного забезпечення для віртуальної сонячної системи. Це дозволить побудувати сценарій фізичних процесів.

Для створення віртуальної сонячної системи потрібно ввести систему координат. Нехай наша система складається із двох тіл:

- 1) Масивної зірки масою M і центром (x_0, y_0) .
- 2) Легкої планети масою m з центром у точці (x, y) , швидкістю (v_x, v_y) і прискорення $a=(a_x, a_y)$.

Після нескладних маніпуляцій з другим законом Ньютона, закону всесвітнього тяжіння і подібними трикутниками, було знайдено що:

$$a_x = G * M * (x_0 - x) / r^3$$

$$a_y = G * M * (y_0 - y) / r^3$$

Це дозволяє скласти алгоритм переміщення планети в полі гравітації зірки:

1. Перед початком задаємо початкове положення планети (x, y) і початкову швидкість (v_x, v_y) .
2. На кожному кроці обчислюємо нове прискорення за формулою вище, після цього перераховуємо швидкість і координати:

$$v_x := v_x + T * a_x$$

$$v_y := v_y + T * a_y$$

$$x := x + T * v_x$$

$$y := y + T * v_y.$$

2.2. Розробка програмного забезпечення для віртуальних 3D об'єктів

Розробка проводилася на мові програмування C # в середовищі розробки Visual Studio. Додаток складається з двох сцен: меню та основної сцени. Ієрархія об'єктів ігрового поля представлена на Рисунок 2.2.

Об'єкти умовно можна розділити на три категорії: відповідальні за генерацію і знищення об'єктів (GeneratorEnemy, GeneratorAmmo, DestroyObject), графічний інтерфейс (Canvas і вкладені об'єкти) та (Background, Middleground, Foreground і вкладені об'єкти) та ігрове поле.

Вкладеність ігрових об'єктів визначає точку відліку їх координат: об'єкт верхнього рівня за точку відліку бере центр ігровий сцени, а вкладений об'єкт - координати об'єкта більш високого рівня.

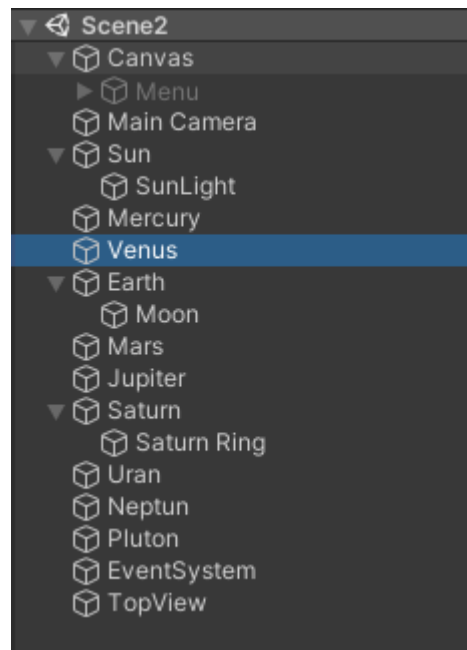


Рисунок 2.2 - Ієрархія сцени

Розробка гравітації. Гравітація є фундаментальною силою фізики яка залежить від маси де всі речі - від зірок, планет і галактик до світла та субатомних частинок - притягуються один до одного. Залежно від розміру, маси та щільності об'єкта гравітаційна сила, яку він чинить, змінюється. А коли йдеться про планети нашої Сонячної системи, які відрізняються за розмірами та масою, сила гравітації на їх поверхнях значно варіюється. Наприклад, гравітація Землі, як уже зазначалося, еквівалентна $9,80665 \text{ м / с}^2$ (або $32,174 \text{ футів / с}^2$). Це означає, що об'єкт, якщо його утримувати над землею і відпустити, буде прискорюватися до поверхні зі швидкістю близько $9,8$ метрів за кожну секунду вільного падіння. Це стандарт для вимірювання сили тяжіння на інших планетах, який також виражається як одиничний g .

В нашому проєкті гравітація розроблена за допомогою мови програмування C#. Відповідно до закону всесвітнього тяжіння Ісаака Ньютона, гравітаційне тяжіння між двома тілами можна виразити математично як $F = G (m_1 * m_2 / r^2)$ - де F - сила, m_1 і m_2 - маси об'єктів,

що взаємодіють, r - відстань між центрами мас і G - гравітаційна стала ($6,674 \times 10^{-11}$ Н м² / кг²). Гравітація на мові програмування C# зображена на рисунку 2.3.

```
private void FixedUpdate()
{
    foreach (Rigidbody body in affectedBodies)
    {
        Vector3 forceDirection = (transform.position - body.position).normalized;
        float distanceSqr = (transform.position - body.position).sqrMagnitude;
        float strength = 10 * componentRigidbody.mass * body.mass / distanceSqr;

        body.AddForce(forceDirection * strength);
    }
}
```

Рисунок 2.3. Закон всесвітнього тяжіння на C#

Для роботи цього C# сценарію необхідно додати до об'єкта компонент "Rigid body", і прибрати властивість "Is kinematik". Щоб змусити планету обертатись навколо орбіти треба додати до об'єкта C# сценарій "AddForce".

Ігровий об'єкт Sun. Об'єкт Sun є головним об'єктом проекту, він притягує всі інші об'єкти. До нього підключені компоненти Sprite Renderer, Box Collider

2D, Rigidbody, Gravity(Script), Change Look (Script).

Компонент Sprite Renderer відповідає за двомірне графічне представлення ігрового об'єкта. Компонент Box Collider 2D відповідає за обробку таблиці з іншими об'єктами. Компонент Rigidbody представляє ігровий об'єкт фізичних властивостей, таких як маса, швидкість і прискорення.

Компонент Gravity (Script) відповідає за силу притягання планет один до одного та інших об'єктів. Компонент Change Look (Script) відповідає за наведення камери на об'єкт.

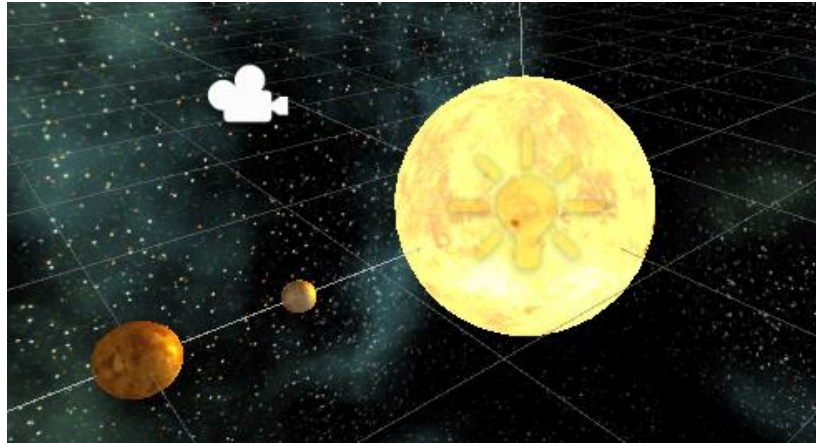


Рисунок 2.3 - Зображення об'єкта Sun

2.3. Реалізація графічного інтерфейсу користувача

Створимо інтерфейс користувача де можливо буде ввести вхідні дані. Також це необхідно для відображення ігрового меню.

Інтерфейси користувача - це основний функціонал практично в кожному сучасному програмному додатку. Ігри не є винятком з цього правила. У переважній більшості ігор для відображення інформації про стан здоров'я, навичках, картах, боєприпасах до зброї і т.д. використовується той чи інший користувацький інтерфейс. Unity 3D пропонує різноманітні елементи інтерфейсу користувача, які ми можемо використовувати у своєму додатку.

Для створення інтерфейсу користувача спочатку потрібно перейти в 2D-режим та створити Canvas (полотно). Полотно - це область де всі елементи користувацького інтерфейсу повинні будуть всередині. При створення будь-якого елемента, буде створено автоматично полотно.

Область полотна відображається у вигляді прямокутника в режимі перегляду сцени. Це дозволяє легко розміщувати елементи інтерфейсу користувача без необхідності постійно бачити Game View. Будь-який елемент інтерфейсу, які ми створимо, повинен бути дочірнім по відношенню до полотна. Коли ми створюємо полотно, ми також можемо

звернути увагу на те, що в нашій ієрархії буде створений новий елемент під назвою "EventSystem". Цей об'єкт використовується для визначення таких речей, як введення миші, що дуже важливо для елементів інтерфейсу користувача, таких як кнопки.

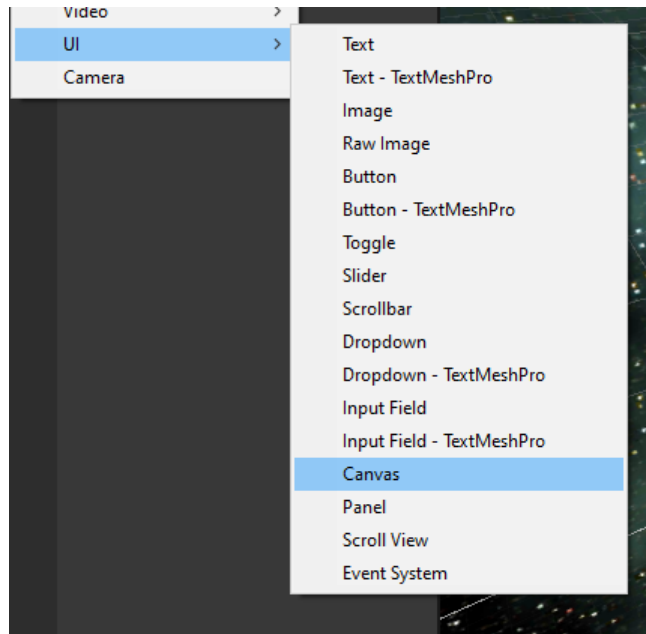


Рисунок 2.4 - Створення елемента Canvas

Canvas ми обов'язково прикріплюємо до головної камери. Це потрібно для того щоб при переміщенні самої камери, всі кнопки та інтерактивні елементи не втрачалися з виду. Тепер ми хочемо додати панель на полотно. Панель - це, по суті, область, в яку можна помістити елементи користувацького інтерфейсу. Перевагою панелі, є те, що ми можемо легко вмикати і вимикати їх за допомогою сценаріїв, не зачіпаючи полотно в цілому.

Це дозволяє нам створювати більш динамічний, дружлюбний користувацький інтерфейс в залежності від стану нашої програми. Наприклад, у нас в програмі є параметри зі змінними їх потрібно змінити. Коли користувач змінить параметри до об'єкта, ми хочемо, щоб в інтерфейсі відобразалося меню, в якому будуть відображатися вхідні

дані та результати обчислень. Також розробимо сценарій для переключання ігрових сцени та можливість поставити гру на паузу.

Для роботи сценарію необхідно перетягнути його на кнопку і вписати назву сцени на яку треба переключити при натисканні на

```
public class MenuPause : MonoBehaviour
{
    public static bool GameIsPaused = false;
    public GameObject pauseMenuUI;

    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Escape))
        {
            if (GameIsPaused)
            {
                Resume();
            }
            else
            {
                Pause();
            }
        }
    }
    public void Resume()
    {
        pauseMenuUI.SetActive(false);
        Time.timeScale = 1f;
        GameIsPaused = false;
    }
    void Pause()
    {
        pauseMenuUI.SetActive(true);
        Time.timeScale = 0f;
        GameIsPaused = true;
    }
    public void LoadMenu()
    {
        Debug.Log("Load");
        Time.timeScale = 1f;
        SceneManager.LoadScene("Setting");
    }

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


кнопку.

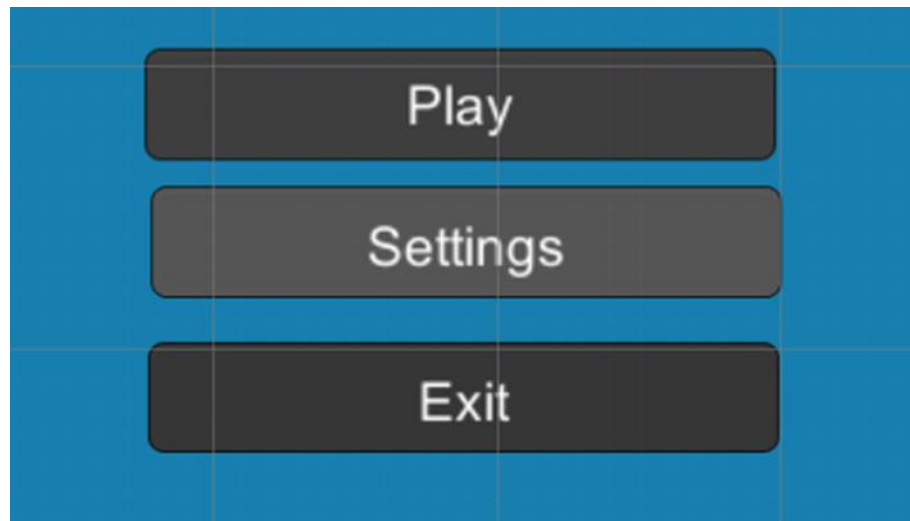


Рисунок 2.5 - Вигляд внутрішньо ігрового меню

Отже, даним додатком можуть користуватися, як і вдома так і будь-якому навчальному закладі, наприклад університет, школа, і так далі. Проект має модульну структуру, в яку можна легко додати нові сценарії та об'єкти. Навчальні мобільні додатки мають величезний пріоритет над традиційним навчанням в закладах освіти, так як його можна використовувати абсолютно в будь-якому місці і самостійно для підняття рівня своїх знань. Напевно одною з головних переваг є те, що не потрібно мати спеціальні прилади та буди у лабораторії, для виконання віртуальних робіт, а це суттєво економить час та гроші.

ВИСНОВКИ

У ході виконання дипломного проекту було виконано поставлене завдання: розробити 3D об'єкти навчального призначення для мобільних пристроїв.

Під час створення мобільного додатку були досліджені такі поняття як створення віртуальних 3D об'єктів у комп'ютерній графіці. В ході моделювання об'єктів проаналізували різні методи та технології розробки віртуальних об'єктів, такі як полігонне моделювання, скульптинг, патч-поверхні та ін. Було переглянуто декілька найпопулярніших програм для 3D моделювання, і зупинили на середовищі розробки Unity, так як він містить найбільшу кількість найнеобхідніших нам інструментів і дозволяє зібрати проект для будь-якого пристрою, будь мобільні та стаціонарні. Також є можливість прикріплювати сценарії та задавати фізичні властивості об'єктів.

Було проведено аналіз вимог, де ми визначили структуру проекту та формули для C# сценаріїв

Загалом в ході виконання дипломної роботи були створені сценарії на мові програмування C#, а також були задані фізичні властивості для кожної моделі.

Розроблено інтерфейс користувача, до якого прив'язана бібліотека об'єктів, та блок меню де вводяться вхідні дані. Та показують значення змінних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. 3D моделювання URL: https://it.wikipedia.org/wiki/Modellazione_3D
(дата звернення: 15.10.2020)
2. 3D-моделювання: визначення, технічні додатки та кар'єрні можливості URL:
<https://www.pedago.it/blog/modellazione-3d-definizione-applicazione-lavoro.htm#:~:text=Cos'%C3%A8%20la%20Modellazione%203D,programmi%20particolari%20chiamati%20software%203D>
(дата звернення: 15.10.2020).
3. 3D-моделювання: 20 найкращих програм для 3D-друку URL:
<https://www.printit3d.com.br/post/modelagem-3d-20-melhores-softwares-para-impress%C3%A3o-3d> (дата звернення: 15.10.2020).
4. Unity ігровий двигун URL:
[https://ru.wikipedia.org/wiki/Unity_\(%D0%B8%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA\)](https://ru.wikipedia.org/wiki/Unity_(%D0%B8%D0%B3%D1%80%D0%BE%D0%B2%D0%BE%D0%B9_%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA)) (дата звернення: 15.10.2020).
5. 3D Modeling URL:
https://es.wikipedia.org/wiki/Modelado_3D#Proceso_de_modelado
(дата звернення: 15.10.2020).
6. What is 3D Modeling & How Is 3D Modeling Used URL:
<https://homesthetics.net/what-is-3dmodeling>
(дата звернення: 15.10.2020).
7. 3D Modeling: An Overview on Various Techniques URL:
<https://professional3dservices.com/blog/3d-modeling-techniques.html>
(дата звернення: 15.10.2020).
8. 3D modeling URL:<https://whatis.techtarget.com/definition/3Dmodeling>
(дата звернення: 15.10.2020).
9. Двигун Unity – особливості, переваги та недоліки URL: <https://cubiq.ru/dvizhokunity>

(дата звернення: 15.10.2020).

10. Второе дыхание Unity 3D URL: <https://anadea.info/ru/blog/unity-3d-gets-a-second-wind> (дата звернення: 15.10.2020).
11. Перемещение объекта по нажатию кнопки в Unity3d на C# URL: <https://habr.com/ru/post/437898/> (дата звернення: 15.10.2020).
12. Движение объекта к точке в Unity3D URL: <https://habr.com/ru/post/347904/> (дата звернення: 15.10.2020).
13. Learning C# and coding in Unity for beginners - Unity URL: <https://unity3d.com/ru/learning-c-sharp-in-unity-for-beginners> (дата звернення: 15.10.2020).
14. C Sharp URL: https://ru.wikipedia.org/wiki/C_Sharp (дата звернення: 15.10.2020).
15. C# Tutorial URL: <https://www.w3schools.com/cs/> (дата звернення: 15.10.2020).
16. What is Unity 3D & What is it Used For? URL: <https://conceptartempire.com/what-is-unity/> (дата звернення: 15.10.2020).
17. Движок Unity – особенности, преимущества и недостатки URL: <https://cubiq.ru/dvizhokunity> (дата звернення: 15.10.2020).

ДОДАТКИ

Додаток А

КОДЕКС АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ ХЕРСОНЬСЬКОГО ДЕРЖАВНОГО УНІВЕРСИТЕТУ

Я, Гордовенко Олександр Вікторович, 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.

(дата)

(підпис)

(ім'я, прізвище)

