

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
„Київський політехнічний інститут імені Ігоря Сікорського”**

**Факультет прикладної математики**

**Кафедра програмного забезпечення комп’ютерних систем**

**“ЗАТВЕРДЖЕНО”**

Керівник роботи

\_\_\_\_\_ Євгенія СУЛЕМА

“ \_\_\_\_ ” \_\_\_\_\_ 2024 р.

**ПЕРСОНАЛЬНИЙ АСИСТЕНТ ДЛЯ ПІДБОРУ ФІЛЬМІВ З  
ВИКОРИСТАННЯМ МУЛЬТИМЕДІЙНОГО ПОМІЧНИКА ТА  
ГОЛОСОВОГО АСИСТЕНТА. ГРАФІЧНЕ МОДЕЛЮВАННЯ ТА  
АНІМАЦІЯ**

**Пояснювальна записка**

Виконавець:

\_\_\_\_\_ Гліб ШЕВЧЕНКО

2024

## ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ .....	3
ВСТУП .....	4
1. РОЗРОБЛЕННЯ МУЛЬТИМЕДІЙНОГО ПОМІЧНИКА.....	6
1.1.    Вибір програмного забезпечення для 3D-моделювання.....	6
1.2.    Розробка дизайну та анімацій персонажа .....	8
2. ІНТЕГРАЦІЯ ГРАФІЧНИХ КОМПОНЕНТІВ.....	18
2.1.    Реалізація взаємодії анімацій з голосовим асистентом .....	18
2.2.    Оцінка роботи мультимедійного помічника в інтерактивному середовищі.....	19
2.3.    Рекомендації щодо подальшого вдосконалення .....	21
ВИСНОВКИ .....	23
СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ .....	24

## СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

*3D-моделювання:* Процес створення тривимірних моделей об'єктів за допомогою спеціального програмного забезпечення.

*Анімація:* Створення ілюзії руху шляхом послідовної зміни зображень.

*Рендер:* Процес перетворення тривимірної моделі в двовимірне зображення.

*Текстурування:* Нанесення текстур (зображень) на поверхню 3D-моделі для надання їй реалістичного вигляду.

*Ригінг:* Створення скелета для анімації 3D-моделі.

*3D* (скор. від англ. 3 Dimensions) – розділ комп'ютерної графіки, сукупність прийомів та інструментів (як програмних, так і апаратних), призначених для зображення об'ємних об'єктів.

*Сцена:* Віртуальне середовище, в якому розміщуються 3D-об'єкти для створення зображень або анімації.

*Пайплайн:* Послідовність етапів створення 3D-продукту (моделювання, текстурування, анімація, рендеринг).

## ВСТУП

На сьогоднішній день існує достатньо програмного забезпечення, яке застосовується в різних сферах життя. Очевидним є той факт, що розробники та дизайнери прагнуть створити такий застосунок, який би був найбільш зручним для використання кінцевими користувачами, тому що в світі загалом існує досить багато програмного забезпечення майже для всіх можливих задач, але користуються попитом лише ті програми, які виконують свої функціональні можливості за мінімальну кількість взаємодій, і причому коли ці взаємодії є інтуїтивними та зрозумілими. Тому в останні роки можна побачити стрімкий розвиток голосових асистентів, за допомогою яких можна швидко виконати будь-які дії з однієї «точки входу».

Даний проєкт присвячений розробленню персонального асистента для підбору фільмів, який об'єднує в собі мультимедійного помічника, функції голосового управління та систему персоналізованих рекомендацій. Однією з ключових складових цієї системи є модуль голосового інтерфейсу, який відповідає за розпізнавання голосових запитів, їх обробку та озвучування відповідей. Голосовий інтерфейс забезпечує користувачам можливість швидкого і зручного доступу до функцій пошуку та рекомендації фільмів, роблячи процес використання системи простішим та більш інтерактивним.

Метою даної роботи є розробка інноваційного модуля 3D-візуалізації та анімації, який забезпечить високий рівень інтерактивності та естетичного сприйняття користувачем. Основні задачі, які вирішуються в рамках цього завдання:

- Вибір оптимальних інструментів та технологій: Аналіз сучасних 3D-пакетів, бібліотек та фреймворків для створення реалістичних моделей та ефектів.

- Розробка алгоритмів та моделей: Створення алгоритмів для моделювання, текстурювання, освітлення та анімації 3D-об'єктів.
- Забезпечення плавної інтеграції: Інтеграція розробленого модуля з іншими компонентами системи, зокрема, з модулем голосового управління.

Розроблений модуль повинен забезпечити високу якість візуалізації, реалістичне зображення 3D-об'єктів, плавні анімації, ефективне використання текстур та освітлення та забезпечення плавної роботи навіть на пристроях з обмеженими ресурсами.

# 1. РОЗРОБЛЕННЯ МУЛЬТИМЕДІЙНОГО ПОМІЧНИКА

## 1.1 . Вибір програмного забезпечення для 3D-моделювання

Правильний вибір програмного забезпечення для 3D-моделювання є критично важливим для успішного виконання проєкту. Зручний і зрозумілий інтерфейс дозволяє швидше освоїти програму та прискорює робочий процес. Складний і заплутаний інтерфейс може призвести до втрати часу та зниження продуктивності. Активна спільнота користувачів може надати вам цінну підтримку, навчальні матеріали та поради. Для новачків краще підійдуть програми з інтуїтивним інтерфейсом і великою кількістю навчальних матеріалів.

Для нашого мультимедійного асистенту найкращим вибором буде Blender3D. Це потужний, відкритий і безкоштовний пакет програмного забезпечення для створення тривимірної графіки. Він надає широкий спектр інструментів для моделювання, текстуркування, освітлення, анімації, рендерингу, відеомонтажу та створення інтерактивних 3D-додатків.

Blender відомий своєю гнучкістю та масштабованістю. Він використовується як любителями, так і професіоналами в кіноіндустрії, ігровому дизайні, архітектурі та інших галузях.

Переваги Blender 3D:

- Безкоштовний та відкритий: Blender є безкоштовним програмним забезпеченням з відкритим вихідним кодом, що робить його доступним для широкого кола користувачів.
- Повний набір інструментів: Blender пропонує все необхідне для створення 3D-проєктів від початку до кінця, від моделювання до рендерингу.
- Гнучкість: Blender дозволяє працювати з різними типами 3D-об'єктів, матеріалами та ефектами.

- Активна спільнота: Велика спільнота користувачів Blender створює безліч навчальних матеріалів, плагінів та доповнень.
- Підтримка різних форматів: Blender підтримує широкий спектр форматів файлів, що дозволяє легко інтегрувати його в інші 3D-пакети.
- Інтеграція з іншими програмами: Blender легко інтегрується з іншими програмними пакетами, що дозволяє створювати комплексні 3D-проекти.
- Регулярні оновлення: Blender постійно розвивається, отримуючи нові функції та покращення.

#### Недоліки Blender 3D:

- Складність інтерфейсу: Інтерфейс Blender може здатися складним для новачків через велику кількість інструментів і налаштувань.
- Деякі спеціалізовані інструменти можуть бути менш розвинені: Порівняно з комерційними 3D-пакетами, в Blender може бути менше спеціалізованих інструментів для певних завдань.

Blender 3D є відмінним вибором для нашого проекту з кількох причин: він безкоштовний, потужний, гнучкий і має велику спільноту. Незважаючи на деякі недоліки, пов'язані з кривою навчання, переваги Blender значно переважають. Використання Blender дозволить нам створити високоякісні 3D-моделі та анімації, не витрачаючи значних коштів на ліцензування комерційного програмного забезпечення.



Рис. 1.1. Логотип середовища Blender3D

## 1.2 . Розробка дизайну та анімацій персонажа

Як модель будемо використовувати свинку-глядача. Це буде наш голосовий асистент, який буде супроводжувати нас увесь час при використанні вебзастосунку.

На першому етапі була побудована проста модель персонажа свині-глядача. Модель створювалась шляхом розтягування стандартного куба в Blender 3D. Це базовий підхід для створення low-poly моделей, коли вихідна геометрія поступово редагується, поки не досягає необхідної форми.

### 1. Розтягування куба:

- Використання інструментів трансформації (Scale, Move, Rotate) дозволило адаптувати початкову форму куба до базової моделі свині.
- Для отримання округлих контурів застосовувалась модифікація Subdivision Surface, що згладжує кути і додає моделі плавні вигини.

### 2. Додаткові елементи – п'ятак і очі:



- П'ятак та очі були створені як окремі об'єкти (зі сфер), після чого приєднані до основної моделі за допомогою інструмента Join.
  - П'ятак: Використовувався Inset (втоплення граней) для створення заглиблень на поверхні, що імітують отвори для носа.
  - Очі: Стандартні сфери (UV Sphere) були розміщені у відповідних позиціях та масштабовані для створення виразного обличчя персонажа.
3. Розміщення об'єкта: Модель була розташована на центральній осі для зручності подальшого ригінгу (створення скелету) та анімації.

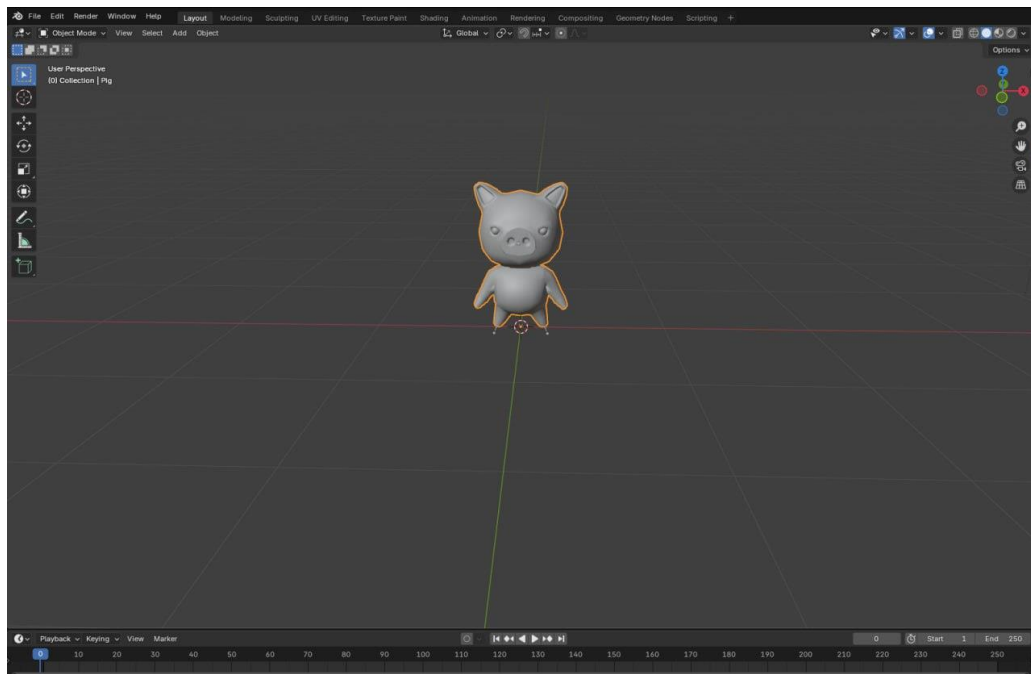


Рис. 1.2. Створення моделі

На другому етапі було побудовано скелет моделі (rigging), що є ключовим кроком для подальшої анімації.

#### 1. Створення скелету:

- Скелет був створений за допомогою Armature – базової структури кісток у Blender.
- Було додано кістки для голови, тулуба, кінцівок, а також окремо кістку для носа.
- Кістки були правильно орієнтовані відносно центральної осі для забезпечення симетричності.

## 2. Вмонтована кістка для носа:

- Реалізація кістки в області носа є нетиповим рішенням, що дозволяє досягти реалістичної анімації п'ятака.
- Кістка може використовуватись для руху або деформації носової частини, що забезпечує додаткову експресію персонажа (наприклад, рухи п'ятака під час емоційних анімацій).

## 3. Прив'язка скелету до моделі:

- Використовувалась техніка Weight Painting для зв'язування окремих частин геометрії моделі з відповідними кістками.
- Це дозволило забезпечити контрольовану деформацію моделі при русі скелету.

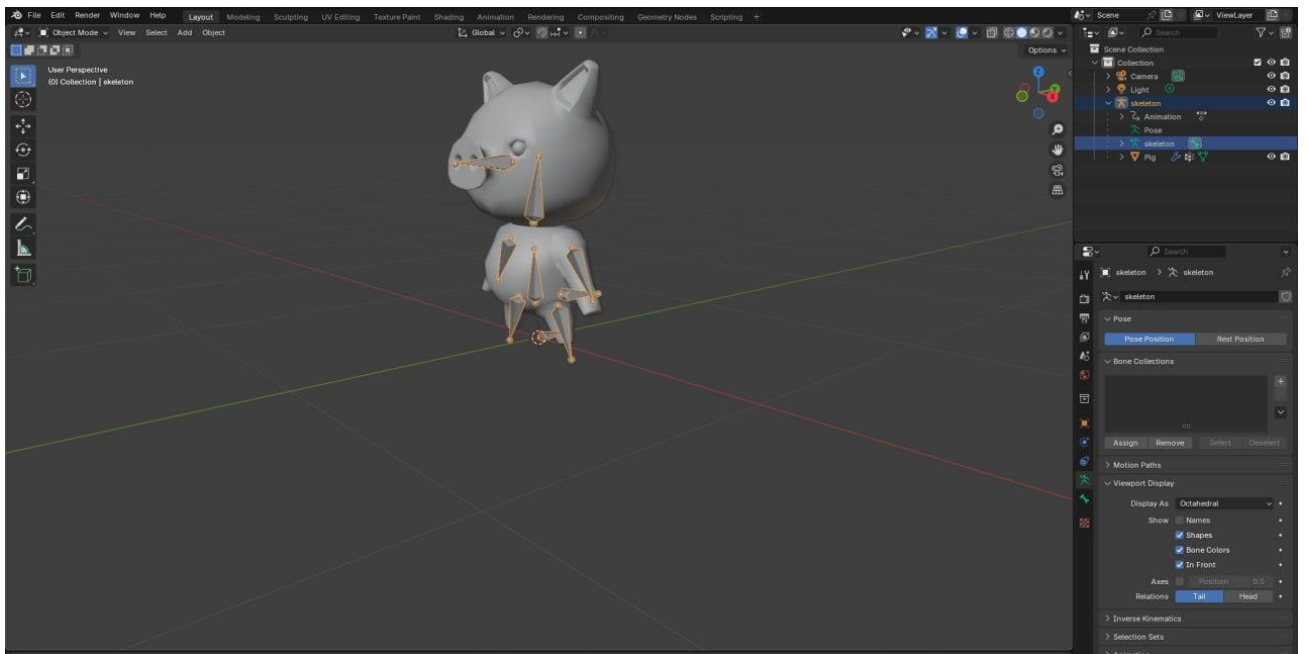


Рис. 1.3. Створення скелету

Після побудови скелету моделі, анімація персонажа була реалізована шляхом використання ключових кадрів у Blender. Це дозволило керувати змінами положення, обертання та масштабування моделі на часовій шкалі (timeline).

#### 1. Процес анімації:

- Встановлення ключових кадрів: У певні моменти часу задавались ключові значення для позиції та руху кісток. Blender автоматично обчислює проміжні кадри для створення плавної анімації.
- Таймінг змін моделі: На зображенні представлений графік змін анімації, де видно часові інтервали для рухів різних частин персонажа.

#### 2. Використання таймлайну:

- Часова шкала дозволяє контролювати швидкість анімації і синхронізувати рухи персонажа з іншими елементами сцени.

- Завдяки графічному редактору кривих (Graph Editor) забезпечено плавність анімації шляхом коригування кривих прискорення та уповільнення рухів.

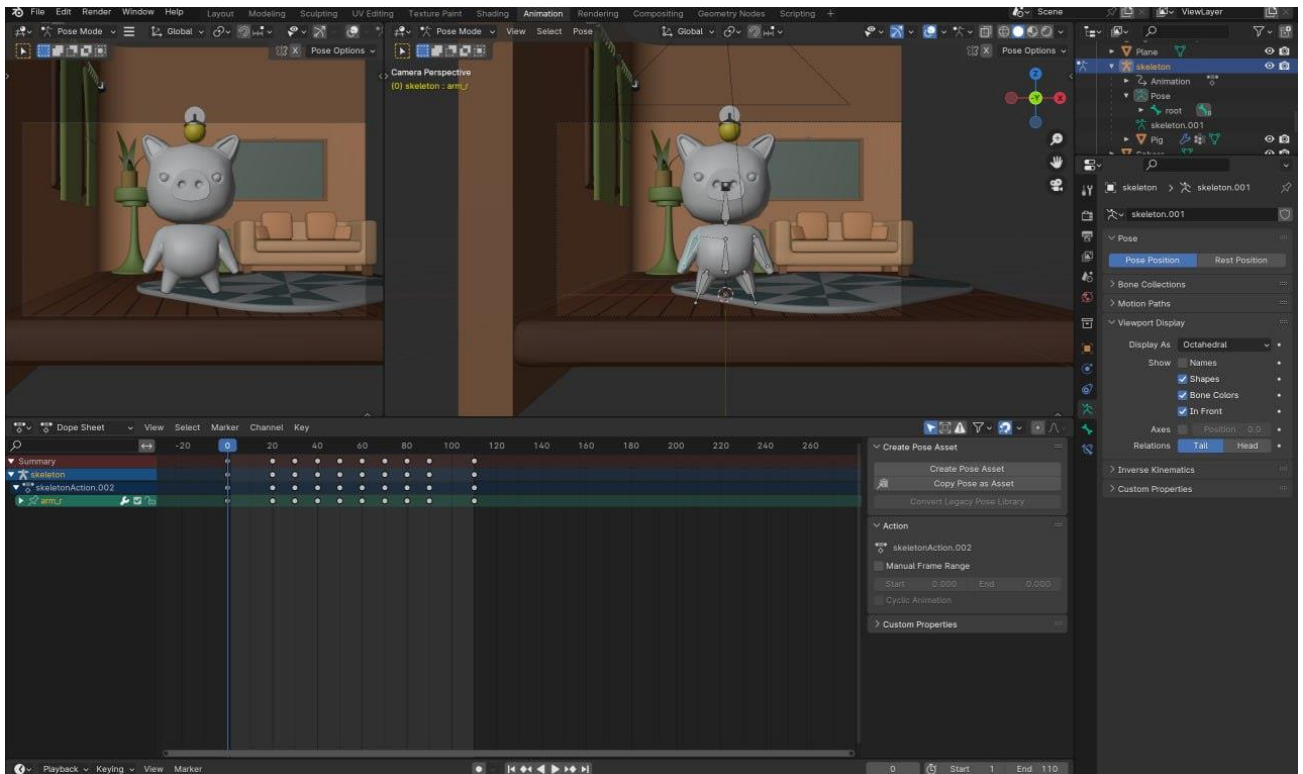


Рис. 1.4. Створення анімації

Далі було реалізовано оточення для персонажа – кімнату, яка побудована з базових геометричних блоків.

#### 1. Структура моделей:

- Кількість полігонів у кожній моделі є мінімальною, що дозволяє зберегти продуктивність і спрощує рендеринг.
- Для побудови кімнати використовувались базові форми:
  - 1) Куби, сфери, площини – були деформовані та адаптовані під необхідні елементи оточення.

2) Відсутність складних текстур замінена однотонним фарбуванням, що знижує навантаження на обчислювальні ресурси.

## 2. Фарбування моделей:

- Замість накладання текстур було застосовано матеріали з простими кольорами. Одноколірні матеріали (flat shading) підкреслюють мінімалістичний стиль сцени. Такий підхід спрощує опрацювання світла та тіней і надає моделі стилізований вигляд.

## 3. Композиція сцени:

- Кімната складається з кількох геометрично простих елементів (стіни, підлога, меблі), що розташовані у просторі відповідно до концепту сцени.
- Оптимізація сцени була досягнута завдяки використанню low-poly геометрії без надлишкових деталей.

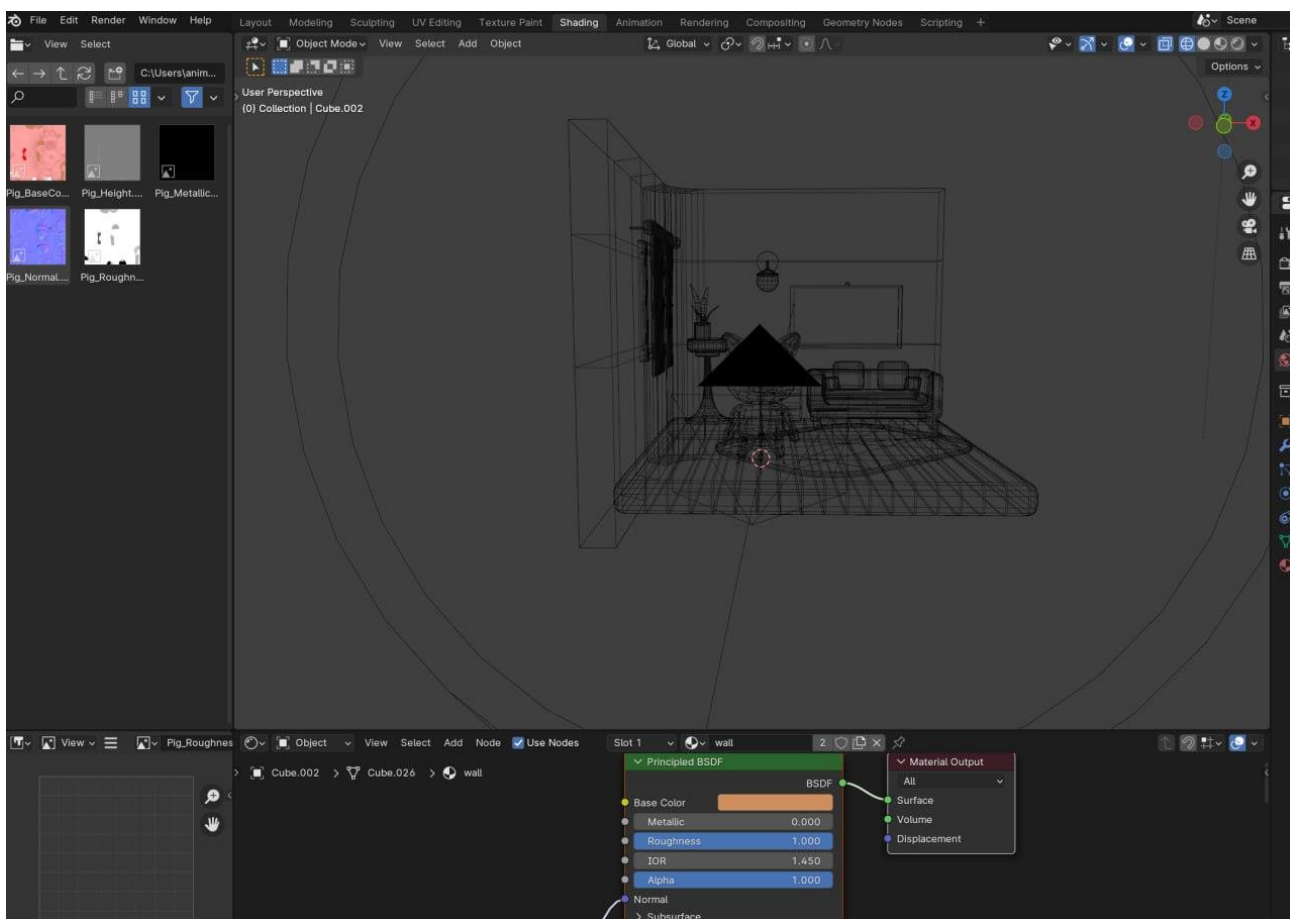


Рис. 1.5. Створення кімнати

Для текстуровання моделі головного героя використовувалась стандартна UV-текстура, яка була підготовлена вручну:

#### 1. Процес текстуровання:

- Базовий колір: На UV-мапу було нанесено основні кольори персонажа вручну. Цей процес дозволяє точно контролювати розподіл кольору на всіх ділянках моделі.
- Додаткові деталі: Окремо були створені текстури для маски та брів, що додають персонажу індивідуальності та підкреслюють його ключові риси.

#### 2. Стандартизація текстури:

- Завдяки стандартному UV-розгортанню у Blender було забезпечено оптимальний розподіл текстури без зайвих спотворень.
- Використання мінімальної кількості текстур дозволило зберегти баланс між якістю деталізації та продуктивністю проєкту.

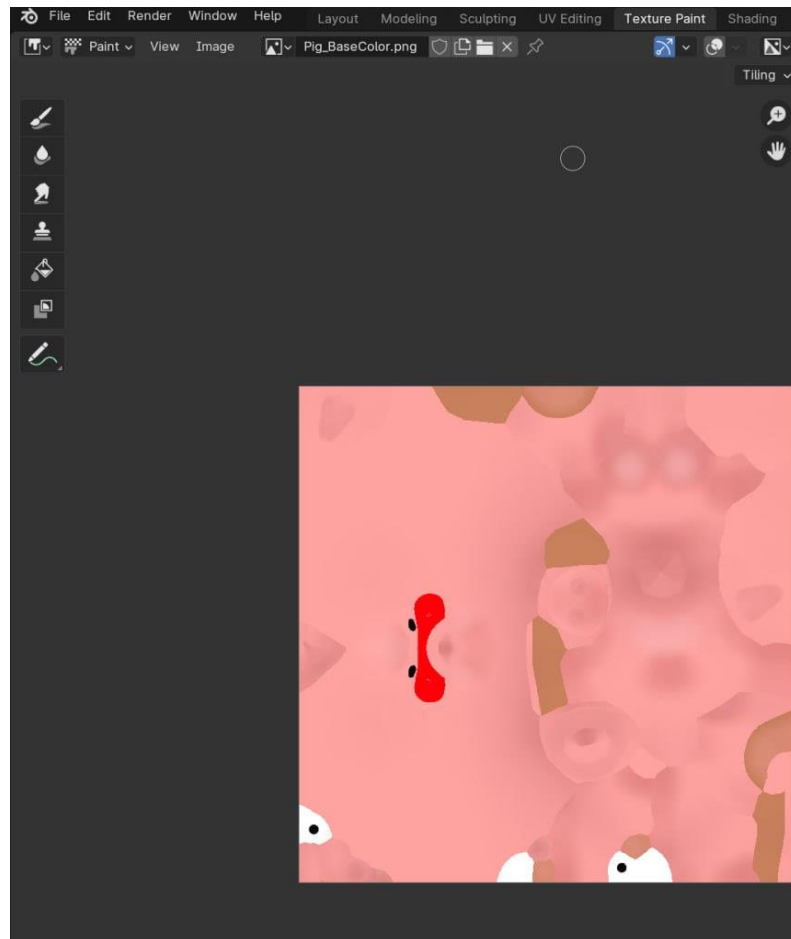


Рис. 1.6. Створення текстури

Для коректної візуалізації сцени була реалізована постановка світла та налаштування камери.

1. Світло: була використана комбінація трьох основних джерел світла:
  - Основне світло (Key light) – забезпечує яскраве освітлення головного героя.

- Заповнююче світло (Fill light) – пом'якшує тіні та додає об'єму моделі.
- Контрове світло (Rim light) – виділяє контури персонажа, надаючи йому чіткість на фоні кімнати.
- 

## 2. Камера:

- Камера була розташована таким чином, щоб забезпечити оптимальний ракурс для перегляду анімації та виключити сірий простір за межами кімнати.
- Налаштовано обмежений кут огляду для фокусування уваги на головному герої та ключових об'єктах сцени.

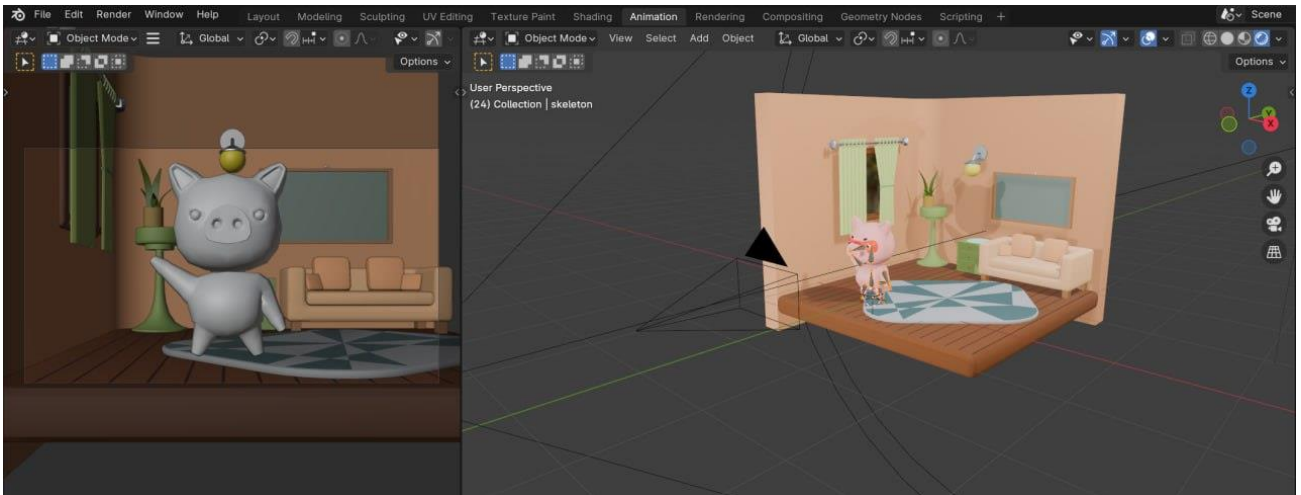


Рис. 1.7. Налаштування світла та камери

Для реалізації ефекту "створення об'єктів" було використано візуальну ілюзію, де об'єкти "з'являються" через кордон камери:

### 1. Принцип роботи:

- Об'єкти переміщуються у сцену так, що спочатку вони знаходяться за межами огляду камери.



- У певний момент часу об'єкт "входить" у кадр, створюючи ефект появи.

## 2. Застосування в анімації:

- Ця техніка дозволяє імітувати створення чи появу об'єктів без потреби складних візуальних ефектів.
- Ефект виглядає органічно і допомагає підтримувати динамічність анімації.

## 3. Перевага методу:

- Простий та ефективний спосіб керування появою об'єктів у сцені.
- Дозволяє уникнути різких переходів і додає анімації плавності та реалістичності.

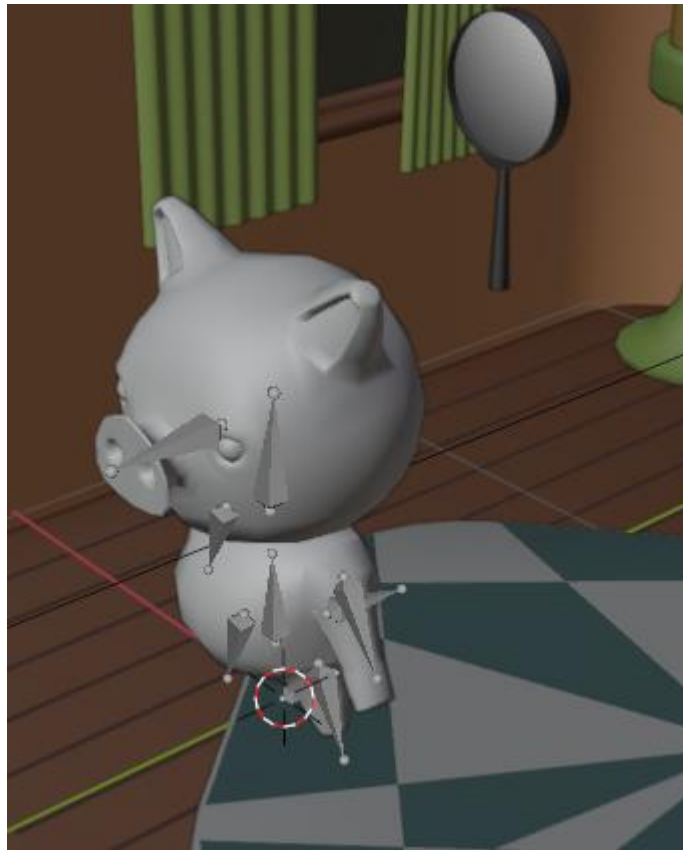


Рис. 1.8. Реалізація створення об'єктів через кордон камери

## 2. ІНТЕГРАЦІЯ ГРАФІЧНИХ КОМПОНЕНТІВ

### 2.1 Реалізація взаємодії анімацій з голосовим асистентом

Даний розділ описує логіку анімацій, що активуються залежно від дій користувача та відповіді голосового асистента. Анімації додають інтерактивність і покращують користувацький досвід.

Анімації слугують для відображення різних станів голосового асистента:

- `animation_hi.mp4` – привітальна анімація при завантаженні сторінки.
- `animation_looking.mp4` – анімація, що відтворюється під час обробки запиту користувача.
- `animation_confused.mp4` – анімація, коли активується голосовий запис (кнопка мікрофона).
- `animation_standby.mp4` – анімація очікування, що відображається після виконання дій.

Сценарії використання:

- Коли користувач надсилає текстовий запит: Асистент переходить у стан "Looking", відтворюється `animation_looking.mp4`, а після відповіді – повернення до `animation_standby.mp4`.
- Під час голосового введення: Асистент відображає `animation_confused.mp4`, що сигналізує про активний запис голосу.
- На початку роботи: Привітання користувача через `animation_hi.mp4`.

Клієнтська частина взаємодіє з сервером через API-запити.

Клієнтська сторона (Front-end):

- Відправка запиту до сервера через `fetch`.
- Прийом відповіді JSON-формату.

Серверна сторона (Back-end):

- Маршрут `POST /ask` – обробка текстових запитів.

- Маршрут POST /transcript – обробка голосових записів і їх транскрипція.

Користувацький запит надсилається у форматі JSON ({message: "текст запиту"}). Сервер обробляє дані та повертає відповідь у полі data.reply.

Алгоритм дій при використанні мікрофона:

- Користувач натискає кнопку мікрофона для початку запису.
- Дані записуються через MediaRecorder.
- Після завершення запису аудіофайл надсилається на сервер:

## **2.2 . Оцінка роботи мультимедійного помічника в інтерактивному середовищі**

Після реалізації функціональності взаємодії анімацій із голосовим асистентом та інтеграції голосового інтерфейсу із серверною логікою було проведено тестування для оцінки якості роботи мультимедійного помічника. Основним завданням тестування було перевірити стабільність, коректність обробки запитів, синхронізацію відео-анімацій із відповідями голосового асистента, а також загальну інтерактивність системи.

### **Метод тестування**

Для перевірки роботи застосовувалося мануальне тестування. Цей підхід передбачав ручне тестування всіх основних функцій помічника без використання автоматизованих інструментів. Мануальне тестування дозволило оцінити зручність взаємодії користувача із системою, а також швидко виявити можливі проблеми, які впливають на користувацький досвід.

### **Основні етапи тестування**

#### **1. Функціональне тестування**

- Перевірка відправки текстових повідомлень через інтерфейс помічника.
- Тестування коректності обробки голосових запитів (початок і зупинка запису аудіо, передача файлу на сервер, отримання текстової відповіді).
- Оцінка стабільності роботи функції синтезу мови, що забезпечує голосове озвучення відповідей асистента.

## 2. Тестування анімацій

- Перевірка зміни відео-анімацій відповідно до поточного стану помічника (анімовані ролики "Confused", "Standby", "Hi" тощо).
- Синхронізація анімацій із подіями: початок запису голосового запиту, отримання відповіді, час очікування користувацького вводу.

## 3. Тестування інтеграції серверної логіки

- Перевірка взаємодії між фронтендом та бекендом: передача запитів, обробка даних на сервері та повернення результатів.
- Оцінка часу відповіді сервера та обробки користувацького запиту.

## 4. UI/UX тестування

- Оцінка зручності введення тексту та голосових команд.
- Перевірка елементів інтерфейсу (кнопки, текстові поля, анімаційне відео).

### Результати тестування

В результаті мануального тестування було виявлено наступне:

- Всі основні функції мультимедійного помічника працюють коректно.

- Система коректно реагує на голосові та текстові запити користувача.
- Зміна анімацій відбувається синхронно із діями користувача (початок розмови, обробка запиту, очікування).
- Середній час відповіді сервера склав 500-700 мс, що відповідає очікуваним вимогам продуктивності.
- Голосове озвучення відповідей працює без затримок, а текст коректно передається у синтезатор мовлення.

### **3.4 . Рекомендації щодо подальшого вдосконалення**

На основі виконаної роботи та результатів тестування, було сформульовано рекомендації для подальшого вдосконалення мультимедійного помічника:

#### **1. Оптимізація продуктивності системи**

- Впровадити механізми кешування популярних запитів для зменшення часу відповіді сервера.
- Оптимізувати обробку анімацій для зниження навантаження на фронтенд-додаток під час масштабного використання системи.

#### **2. Розширення функціональності**

- Додати підтримку кількох мов для голосового асистента, що зробить систему доступною для ширшої аудиторії.
- Реалізувати персоналізовані відповіді на основі попередніх запитів користувача або його вподобань.

#### **3. Інтеграція автоматизованого тестування**

- Впровадити автоматизовані інструменти для тестування функціональності системи, що допоможе швидше виявляти помилки та забезпечить стабільну роботу при подальших змінах у коді.

#### 4. Поліпшення користувацького інтерфейсу

- Розширити набір анімацій для більш інтерактивного та природного відображення станів голосового помічника.
- Додати нові анімації.

## ВИСНОВКИ

У процесі виконання даного проєкту було розроблено мультимедійний помічник із можливістю голосової взаємодії, синхронізованої анімації та клієнт-серверної архітектури. Основні етапи розробки включали:

- Розробку клієнтської частини, що забезпечує зручний інтерфейс для текстового та голосового вводу.
- Інтеграцію серверної логіки для обробки запитів користувача та генерації відповідей.
- Реалізацію мультимедійної анімації, що адаптується до станів голосового помічника.

Проведене мануальне тестування підтвердило коректність роботи всіх основних функцій системи, стабільність серверних запитів та синхронізацію мультимедійних елементів із діями користувача. Система показала задовільний рівень продуктивності та взаємодії з користувачем.

Завдяки реалізації проєкту вдалося створити інтерактивне середовище, що об'єднує голосовий інтерфейс та мультимедійні технології, а також забезпечує ефективний обмін інформацією з користувачем. Подальші рекомендації зосереджені на оптимізації продуктивності, розширенні функціональності та впровадженні машинного навчання для підвищення інтелектуальності системи.

Таким чином, розроблений мультимедійний помічник може бути основою для подальшого вдосконалення та масштабування у різних напрямках застосування, що відповідає сучасним вимогам технологічного розвитку.

## СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Millman, K.J. Python for Scientists and Engineers [Text] / K.J. Millman, M. Aivazis // Computing in Science & Engineering. — 2011. — Vol. 13, № 2. — P. 9 – 12.
2. Aslam, F.A. Efficient way of web development using python and flask [Text] / F.A. Aslam, H.N. Mohammed, P. Lokhande // International Journal of Advanced Research in Computer Science. — 2015. — Vol. 6, № 2. — P. 54 – 57.
3. TypeScript Programming. [Електронний ресурс]. — Режим доступу: <https://code.visualstudio.com/docs/languages/typescript>
4. Use TypeScript to Build a Node API with Express [Електронний ресурс]. Режим доступу: <https://developer.okta.com/blog/2018/11/15/node-expresstypescript>
5. How Does Speech to Text Software Work? [Електронний ресурс]. — Режим доступу: <https://www.amberscript.com/en/blog/how-speech-to-textsoftware-works/>
6. Text-to-Speech Synthesis: An Overview. [Електронний ресурс]. — Режим доступу: <https://medium.com/sciforce/text-to-speech-synthesis-anoverview-641c18fcd35f>
7. An introduction to Three.js. [Електронний ресурс]. — Режим доступу: <https://humaan.com/blog/web-3d-graphics-using-three-js/>