ДИПЛОМНА РОБОТА

БАКАЛАВРА

ДРБ. ІП - 11.00.00.000 ПЗ

Група ІП-19-1К

Соколовський Максим

2022

Міністерство освіти і науки України
Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

УДК 004.4

ДИПЛОМНА РОБОТА БАКАЛАВРА

Тема: «Розробка бізнес-моделей програмного забезпечення симуляцій на основі агентних сутностей»

(назва згідно з наказом ректора)

Спеціальність – 121 “Інженерія програмного забезпечення”

ПОЯСНЮВАЛЬНА ЗАПИСКА

ДРБ.ІП - 11.00.00.000 ПЗ

(позначення)

Студент

ІП-19-1К /М.В.Соколовський/
(шифр групи) (підпис) (дата) (розшифровка підпису)

Керівник роботи

доц., к.т.н. /М.М.Яцишин/
(посада) (підпис) (дата) (розшифровка підпису)

Консультанти:

з питань нормоконтролю

доц., к.т.н /В.Я. Піх/
(посада) (підпис) (дата) (розшифровка підпису)

Допускається до захисту
Завідувач кафедри

проф., д.т.н. /В.І.Шекета/
(посада) (підпис) (дата) (розшифровка підпису)

2022
(рік)

Івано-Франківський національний технічний університет нафти і газу

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**
проф. **В.І. Шекета**
“ ____ ” _____2022 р.

ЗАВДАННЯ

НА ДИПЛОМНУ РОБОТУ БАКАЛАВРА СТУДЕНТОВІ

Соколовський Максим Валерійович

(прізвище, ім’я, по-батькові)

1. Тема проекту (роботи) «Розробка бізнес-моделей програмного забезпечення симуляцій на основі агентних сутностей»

керівник проекту (роботи) Яцишин Микола Миколайович, к.т.н., доцент

затверджені наказом закладу вищої освіти від **“** 25 ” травня 2022 р. № _____

2. Строк подання студентом проекту (роботи) 12 червня 2022р.

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

4. Зміст розрахунково - пояснювальної записки (перелік питань, які потрібно розробити)

1.Вступ до проблеми

2. Огляд існуючих концепцій, рішень та сервісів в даній області

3. Побудова моделі або алгоритму власного рішення

4. Документування реалізації власного оригінального рішення вибраними засобами

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1. Плюси та мінуси різних мов програмування в Maya

2. Просторові відносини, які підтримуються різними системами

3. Просторові відносини, які підтримує наша система

4. Список інструментів, які використовуються для реалізації запропонованої нами системи

5. Порівняння між різними системами генерації 3D-сцени

6.Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання Видав	Завдання прийняв

7. Дата видачі завдання _____

Керівник _____
(підпис)

Завдання прийняв до виконання_____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Визначення теми дипломної роботи	01.02.2022	виконано
2	Огляд існуючих концепцій, рішень та сервісів в даній області	01.03.2022	виконано
3	Побудова моделі або алгоритму власного рішення	01.04.2022	виконано
4	Документування реалізації власного оригінального рішення вибраними засобами	01.06.2022	виконано
5	Тестування розробленого програмного продукту	08.06.2022	виконано
6	Оформлення пояснювальної записки дипломної роботи завідувачем кафедри	10.06.2022	виконано

Студент _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

В даній дипломній роботі запропоновано схему вилучення об’єктів, що цікавлять, та їх просторових відносин із текстового опису, наданого користувачем, для створення тривимірної динамічної сцени та анімації, щоб зробити її більш реалістичною. Створення 3D-сцени вручну – процес тривалий і вимагає досвіду із середовищем створення контенту. Автоматичне створення сцени з використанням текстового опису та бібліотеки розроблених компонентів пропонує швидку та просту альтернативу для ручного представлення сцени та підтвердження ідей концепції.

КЛЮЧОВІ СЛОВА: ГЕНЕРАЦІЯ ТЕКСТУ В СЦЕНУ, 3D-АНІМАЦІЯ, МУЛЬТИМЕДІА, КОМП'ЮТЕРНА ГРАФІКА, ГРАФІЧНИЙ КОНТЕНТ, ПРОСТОРОВІ ПРЕДСТАВЛЕННЯ.

ANNOTATION

This thesis proposes a scheme for extracting objects of interest and their spatial relationships from a user-provided textual description to create a 3D dynamic scene and animate it to make it more realistic. Manually creating a 3D scene is a lengthy process and requires experience with a content creation environment. Automatic scene creation using a textual description and a library of designed components offers a quick and easy alternative to manual scene rendering and proof-of-concept ideas.

KEY WORDS: GENERATION OF TEXT IN THE SCENE, 3D ANIMATION, MULTIMEDIA, COMPUTER GRAPHICS, GRAPHIC CONTENT, SPATIAL REPRESENTATIONS.

Зміст

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ 7
ВСТУП 8

1 ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ ТА СЕРВІСІВ В ПРЕДМЕТНІЙ ОБЛАСТІ 12
1.1 Архітектура 3D анімації 12
1.3 Постановка задачі дослідження 16
1.4 Структуризація очікуваних результатів 16
1.5 Конвеєр 3D-анімації 17
1.6 Етап підготовки 3D проекту 18
1.7 Опис етапу виробництва 23
1.8 Етап постпродакшн для 3D проекту 28

2 АЛГОРИТМІЧНІ ТА МАТЕМАТИЧНІ РІШЕННЯ ДЛЯ ЗАДАЧ СТВОРЕННЯ 3D ПРОЕКТІВ 30
2.1 Текст до сцени 30
2.2 Текст до анімації 38
2.3 Архітектура на основі Maya API 44
2.4 Опис мови розробки 47

3 РОЗРОБКА МОДЕЛЕЙ ТА АЛГОРИТМІВ 3D СИМУЛЯЦІЙ НА ОСНОВІ АГЕНТНИХ СУТНОСТЕЙ 49
3.1 Обмежувальна рамка 3D об’єкта 49
3.2 Просторові відношення об’єктів 50
3.3 Розробка покрокового алгоритму та генерація схеми 52
3.4 Тестування розробленого продукту 59

3.5 Представлення результатів генерації 3D сцен 63

ВИСНОВКИ 66

СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА 67

БІБЛІОГРАФІЧНА ДОВІДКА

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ТЗ – Технічне завдання

БД – База даних

СУБД - Система управління базами даних

ПЗ - Програмне забезпечення

ВСТУП

Реальні взаємодії відбуваються у всіх трьох вимірах простору. Об’єкти в реальному світі є тривимірними, оскільки вони мають висоту, ширину та глибину, тоді як рисунок на папері часто є двовимірним і має лише висоту та ширину. 2D-зображення описують сцену з певної точки зору, тоді як одна 3D-сцена описує реальний світ з різних точок зору. 3D моделі більш реалістичні, ніж 2D. 3D-сцени можна використовувати для створення 2D-зображень або малюнків безпосередньо в різних точках огляду. 3D-моделювання використовується в різних галузях, таких як кіно, відеоігри, реклама, архітектура та віртуальна реальність.

Щоб пояснити історію чи ситуацію, ми використовуємо природну мову як засіб спілкування. Людина, яка слухає цю історію, візуалізує об’єкти та оточення в розповіді та розуміє зміст. Наприклад, 3D-сцена може бути використана як наочний посібник для навчання дітей, що може зацікавити їх темою та уявити фактичну історію, яку викладає вчитель. Створення такої тривимірної сцени, яка представляє історію, є трудомістким і складним процесом, який включає створення об’єктів, вибір матеріалів для кожного об’єкта та їх акуратне розміщення в сцені. Таким чином, існує вимога до автоматичних методів генерації тексту до сцени, які могли б приймати текст як вхідні дані та відповідно генерувати 3D-сцену.

Однак за останні роки ця технологія значно покращилася в таких областях, як графіка, комп’ютерний зір та обробка природної мови. Це технологічне зростання дозволило нам точніше отримувати інформацію з введеного тексту та створювати більш реалістичні 3D-сцени та анімації.

Метою цієї дипломної роботи є створення системи, яка здатна генерувати тривимірну сцену з введеного тексту, описуючи об’єкти, їх розміщення, положення та їх взаємозв’язки. Щоб створити 3D-сцени з тексту, ми витягуємо назви об’єктів та їх просторові відносини з вхідного тексту, який описує реальні об’єкти світу; потім об'єкти імпортуються з локальної бібліотеки

об'єктів. Бібліотека об’єктів складається з окремих 3D-об’єктів, більшість з яких створені автором, а деякі завантажені з мультимедійних веб-сайтів, які продають 3D-об’єкти. Ці компанії продають 3D-об’єкти на своїх умовах, де будь-який вид перерозподілу заборонений. На основі об’єктів, згаданих у вхідному тексті, тривимірні об’єкти імпортуються в кінцеву сцену. Після імпортування об’єктів на основі просторових відносин об’єкти перегруповуються та організуються. Нарешті, рух до об’єктів та ефекти додаються до сцени відповідно до інформації, наданої у введеному тексті.

Згенеровані 3D-сцени можна використовувати багатьма способами (наприклад, відеоігри, реклама, архітектура, право) в індустрії анімації, яку можна класифікувати на розважальну, наукову та інші галузі. Кожна з цих галузей має кілька підкатегорій.

Основною метою індустрії розваг є надання розваг аудиторії у вигляді 3D-анімаційних фільмів, телевізійних шоу, відеоігор та рекламних роликів.

Повністю анімаційний фільм і фільм з візуальними ефектами - це два типи фільмів, які створюються в індустрії 3D-анімації.

На відміну від повністю анімаційних повнометражних фільмів, фільми з візуальними ефектами знімає звичайна знімальна група. Зроблені кадри надсилаються в студії візуальних ефектів, щоб додати необхідні ефекти до знімків.

Створення 3D-анімаційного телевізійного шоу є дорогим і трудомістким. Ці візуалізації використовуються в освітніх шоу, щоб допомогти аудиторії зрозуміти конкретні теми.

У індустрії відеоігор художники використовують програмне забезпечення 3D-графіки для створення віртуальних світів і персонажів, які використовуються в движку відеоігор . Ця галузь є надзвичайно успішною і принаймні такою ж прибутковою, як і кіноіндустрія. В індустрії відеоігор є дві основні галузі: 3D-анімація в грі, яка створює фактичне ігрове середовище, в яке беруть участь гравці під час гри, і ігровий кінематограф, який створюється в

кінематографі та використовується як сюжетні сцени, щоб допомогти керувати. історія вперед між рівнями гри. Ігровий кінематограф, як і фільми, сьогодні обмежений лише бюджетом і часом, необхідним для розробки сцен 3D-анімації та відтворення останніх сцен у відеогрі. Художники ігрового кінематографа виконують подібну роботу, як і 3D-аніматори, але зазвичай дотримуючись набагато швидшого терміну. Кінематографічні сцени та трейлери кінематографічних ігор мають високу якість, що може конкурувати в кіноіндустрії.

Ці короткі анімації повинні бути в змозі представити велику кількість інформації за короткий проміжок часу. Для реклами продукту художник створює 3D-модель, яка служить прототипом фактичного продукту, щоб викликати інтерес споживачів до цього продукту.

Реклама має дуже високий рівень якості про продукти і створюється за короткий проміжок часу. Студії рекламної анімації середніх розмірів і дотримуються надійного робочого процесу, щоб забезпечити швидший вихід для цього типу анімації.

Наукова індустрія в першу чергу використовує 3D-анімацію в галузі медицини та архітектури.

У медичній галузі тривимірна анімація використовується багатьма способами, від створення візуалізації певної медичної події до опису біологічної реакції. Найпопулярнішим медичним використанням типу 3D-анімації є медична візуалізація для освіти або маркетингу. Анімація використовується для навчання громадськості та медичного персоналу новим методикам чи препаратам. 3D-анімація може створити багатий візуальний посібник для людських і біологічних систем і може надати важливу інформацію за короткий проміжок часу.

Тривимірна анімація в медичній індустрії – це ринок, який значно розвивається, який може бути прибутковим для окремого художника або невеликої студії професіоналів. Величезним недоліком цієї галузі є те, що

більшість людей, які сьогодні навчаються 3D-анімації, воліли б працювати в індустрії відеоігор або кіноіндустрії, а не в фармацевтичній компанії чи університетському дослідницькому проекті. ігрові компанії більш креативні та цікаві, тоді як робота в фармацевтичній компанії полягає у створенні анімації, яка нагадує продукт.

1 ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ ТА СЕРВІСІВ В ПРЕДМЕТНІЙ ОБЛАСТІ

1.1 Архітектура 3D анімації

Архітектори використовують 3D-анімацію для створення кращих і стабільніших комп'ютерних проектів. Сьогодні архітектори використовують програми CAD не тільки для створення моделей, а й для тестування та візуалізації цих моделей, щоб побачити, як будуть виглядати конструкції, перш ніж вони будуть створені. Таке програмне забезпечення, як Autodesk AutoCAD [6] і Autodesk Revit, дозволяє архітекторам перевіряти довговічність конструкцій за певних обставин, щоб побачити, чи можуть вони витримати певний тип стихійного лиха. Файли CAD можна конвертувати, а потім відтворювати за допомогою програмного забезпечення для 3D-анімації, такого як Autodesk 3ds Max і Autodesk Maya, щоб клієнти могли побачити, як може виглядати структура зсередини і зовні. Цей тип робіт стає все більш популярним і може бути дуже рентабельним способом перевірити зовнішній вигляд будівлі з конкретних матеріалів перед фактичним її будівництвом. Приклад інтер'єрної та зовнішньої архітектурної обробки показано на рисунку 1.1.



Рисунок 1.1 Внутрішнє та зовнішньое оформлення архітектури [3]

У законі 3D-анімація може бути використана для криміналістики та реальної реконструкції та моделювання місця ДТП. Цей вид анімації створений,

щоб довести, спростувати або розробити докази в судовій справі, щоб допомогти або захисту, або обвинуваченню. Він може мати суто комп'ютерне моделювання або лише ключову анімацію місця злочину, щоб допомогти судді або присяжним вивчити місце злочину, якщо потрібно. Його можна використовувати як приклад, щоб довести, що злочинець міг або не міг застрелити когось із певного місця або описати автомобільну аварію. Ці типи анімації не використовуються як очевидні докази, але можуть бути використані для опису сцени, в якій стався інцидент.

3D-анімація також може бути використана для створення місця злочину шляхом його сканування за допомогою 3D-лазерного сканера . Це 3D-лазерне сканування може ідеально відтворити місце злочину, щоб у разі потреби його можна було використовувати як довідник. Це 3D-сканування може мати точність до міліметрів, тому воно може мати вирішальне значення для судової справи або розслідування. Ці 3D-сканери занадто дорогі для окремої людини, яка хоче ними користуватися.



Рисунок 1.2 Анімація пострілів з певного місця [3]

Технології, які рухають індустрію 3D-анімації, змінюються щороку. Нові поля 3D-анімації – це доповнена реальність і проекція.

У доповненій реальності (AR) користувач дивиться на реальний світ і може додавати до нього тривимірні елементи.

Як правило, ми дивимося через камеру і можемо розміщувати тривимірні об’єкти через камеру, як показано на рисунку 1.3. Навіть об’єкти реального світу за допомогою AR можна виміряти. Інші оглядові пристрої кріпляться на голові з прозорим козирком, який додає тривимірні елементи до реального світу.

Проекційне відображення — це новий метод, який може перетворити будь-яку поверхню на відеодисплей, як правило, великих будівель. Ця техніка використовує проектори для проектування 3D-анімації на будівлю. Він демонструє нові та захоплюючі ефекти, такі як руйнування будівель або пожежа на поверхні. Ця техніка використовується для створення багатьох захоплюючих ефектів, і в майбутньому вона може стати одним з основних застосувань у 3D-анімації..



Рисунок 1.3 Приклад доповненої реальності

1.2 Мотивація

Схильність людей розуміти відео або зображення передбачає більше, ніж розуміння текстового вмісту. Анімація створюється шляхом переміщення нерухомих зображень з певною кількістю кадрів в секунду, щоб створити ілюзію того, що об’єкт рухається. Ми бачили різні програми, де можна використовувати 3D-анімацію. Його також можна використовувати як наочний посібник для навчання дітей, щоб вони могли уявити реальну історію, яку викладає вчитель. У медицині посттравматичний стресовий розлад (ПТСР) — це психічне захворювання, яке спричиняється жахливою подією, як-от війна або зіткнення на дорозі. Після безпосереднього зіткнення з такими умовами у людини можуть виникнути такі симптоми, як втрата інтересу до спілкування з іншими, погані сни, уникнення людей, труднощі з концентрацією тощо. Щоб лікувати таких пацієнтів, лікарі можуть відтворити подію у віртуальному середовищі та проаналізувати параметри здоров'я пацієнта при його впливі. Таким чином, програмне забезпечення, яке допомагає перетворити текст або промову в реалістичну віртуальну сцену, може бути дуже корисним для медичних працівників при лікуванні людей, які страждають на посттравматичний стресовий стрес.

Мета створення тривимірних сцен із тексту має багато складних завдань, пов’язаних із цим. Система повинна бути в змозі ідентифікувати об’єкти та їх просторові відносини в тексті, а також визначити, як об’єкти пов’язані один з одним у сцені відповідно до їх просторових відносин. Система також повинна бути в змозі знати розміри об'єктів, щоб розмістити їх у сцені з урахуванням їх просторових відносин. Ці завдання вказують на те, що генерація тексту до 3D-сцени є широким і складним завданням.

У цій дипломній роботі ми описуємо, як мова може бути пов’язана для представлення геометричних уявлень об’єктів за допомогою просторових відносин між ними, які є ключовими елементами в описі реального світу.

1.3 Постановка задачі дослідження

3D-сцена потрібна багатьом галузям у різних галузях. Створення 3D-сцени займає багато часу і вимагає багато зусиль для вивчення складного програмного забезпечення для 3D-анімації. Щоб зменшити зусилля людини у навчанні програмного забезпечення та створенні 3D-сцени, ми пропонуємо структуру через Autodesk Maya для створення 3D-сцен із введеного тексту для опису сцени в реальному світі.

Існує кілька ключових завдань у створенні 3D-сцени з текстових описів.

• Визначення ключових об'єктів, які будуть присутні в сцені.

• Визначення просторових відносин, що з'єднують об'єкти.

• Розміщення об'єктів у сцені шляхом знаходження розмірів об'єктів.

• Додавання руху об'єктам.

• Додавання ефектів до окремих об'єктів і до всієї сцени.

У наступних розділах ми пояснюємо кожне з цих завдань, які ведуть нас до успішного створення 3D-сцени з текстових описів.

1.4 Структуризація очікуваних результатів

У цій дипломній роботі ми робимо наступні внески.

• Запропонована система генерує 3D-сцени з введеного тексту. Ми використовували Набір інструментів природної мови (NLTK), щоб витягти імена об’єктів із введеного тексту, щоб визначити об’єкти, які мають бути присутніми в сцені.

• Ми витягуємо різні просторові відносини, пов'язані з об'єктами, з вхідного тексту, щоб визначити їхнє розташування в сцені.

• Ми надаємо структуру позиціонування об'єктів, яка допомагає розташовувати об'єкти в сцені, обчислюючи значення рамки об'єктів та враховуючи просторові відносини між ними.

• Ми також додаємо рух і ефекти до статичних сцен.

1.5 Конвеєр 3D-анімації

Конвеєр 3D-анімації складається з комбінації людей, апаратного та програмного забезпечення для розробки продукту або активу 3D-анімації, який працює в певному порядку, де кінцевим продуктом може бути повнометражний фільм, короткометражний фільм, телевізійне шоу або відеогра. Залежно від типу проекту, конкретне програмне та апаратне забезпечення може змінюватися, але основні етапи 3D-анімації схожі. Нижче наведено три основних етапи конвеєра 3D-анімації:

• Підготовка

• Виробництво

• Пост-продакшн

На стадії виробництва 3D-художники розробляють продукт, а під час постпродакшну художники доставляють проект у його остаточному вигляді. Увесь конвеєр виробництва 3D-анімації показаний на рисунку 1.4.

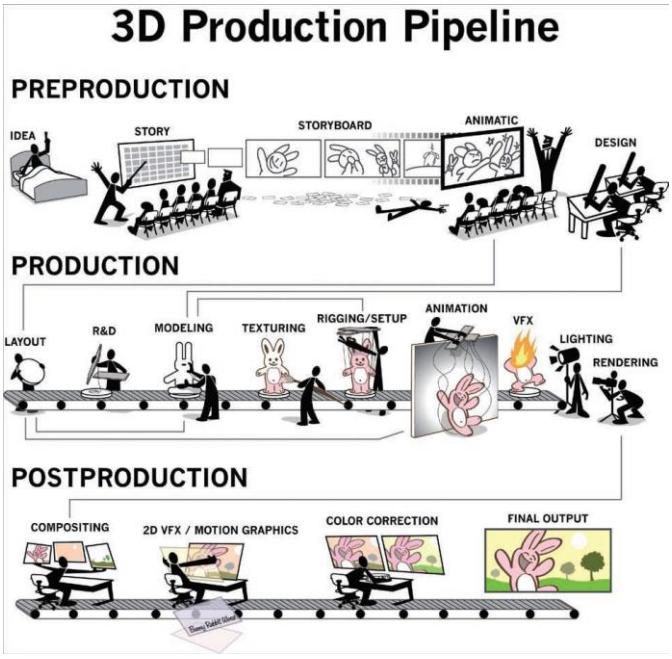


Рисунок 1.4: конвеєр 3D-анімації

1.6 Етап підготовки 3D проекту

Підготовка — це етап планування, проектування та дослідження повного 3D-проекту. Це важливий етап, оскільки саме тут генеруються ідеї та розробляються виробничі плани для управління проектом. Чудова ідея з надійним планом виробництва має набагато кращі шанси бути завершеною, ніж чудова ідея без плану.

Команда підготовки до виробництва розділена на дві групи: художники/оповідання та менеджмент. Художники/сценаристи створюють ідеї, історії та дизайни, а група управління створює план виробництва. Крім того, група творчого управління працює безпосередньо з художниками/оповідачами. Вони також відповідають за продаж проекту інвесторам.

Усі компоненти різноманітної індустрії розваг обробляють етап підготовки до виробництва приблизно однаково. Художники використовують цей етап, щоб отримати хороші ідеї, записуючи ці ідеї в цікаву історію, а потім вирішуючи, як розповісти цю історію візуально. Ці галузі можуть і більше використовувати приблизно половину всього терміну виробництва на стадії підготовки до виробництва.

Після перегляду фільму, адаптованого з книги-бестселера, можна було почути або навіть сказати: «Книга була набагато кращою, ніж фільм». Це тому, що існує суттєва різниця між оповіданням написаної історії, оскільки перше призначене для читання, тоді як останнє — це розповідь у візуальній історії, яку призначено для перегляду. Щоб історія добре відображалася на екрані, вся виробнича команда повинна розглянути різні аспекти: донести до аудиторії, що відбудеться з різними ракурсами; вибір кольору в проекті, щоб керувати настроєм і почуттями аудиторії. Щоб ще більше вплинути на настрій аудиторії, музику, звук або тишу слід використовувати разом із розповіддю.

В архітектурній індустрії 3D-художники проводять більшу частину свого виробничого часу на стадії підготовки. Дизайн і функціональність – головні

цілі цих художників. Якщо проект архітектури повністю затверджений, кінцевим продуктом є об’єкт, який буде створено в реальному світі. Художники, які займаються візуалізацією продуктів, витрачають багато часу на цій стадії підготовки, щоб переконатися, що їх дизайн продукту саме такий, як вони хочуть. Продукт можна побачити у фізичному світі, перш ніж запустити його у виробництво, використовуючи швидкий прототип для кінцевої моделі. Швидке створення прототипів – це створення віртуального об’єкта в реальному світі за допомогою технології адитивної конструкції, як-от 3D-принтер, який має інтенсивний лазер, який сплавляє невеликі частинки пластику разом, щоб створити об’єкт.

У медико-анімаційній індустрії художники витрачають свій час на дослідження та розробки типу анімації, який їм потрібно створити. Художники повинні чітко визначити тип медичної техніки або органічної системи, яку вони повинні візуально описати у своєму проекті. Протягом цього часу художники займаються широкими дослідженнями та використовують експертів-консультантів, щоб перевірити фактичну правильність своїх візуалізацій. Як і в інших галузях індустрії розваг, ці артисти також повинні розуміти, як найкраще відображати те, що вони намагаються пояснити чи описати. Вони повинні враховувати різні питання: від правильного розміщення камер до загальної анімації та кінцевого продукту.

На відміну від інших, юридична галузь може не витрачати багато часу на стадії підготовки. Команди з цих галузей використовують 3D комп’ютерне моделювання, щоб відтворити будь-який злочинний інцидент, щоб перевірити можливості того, як злочин може статися. Команда криміналістів може використовувати 3D-сканер, щоб отримати дані з місця злочину, щоб слідчий міг повернутися на місце злочину «практично» набагато пізніше після злочину. Художники в цій галузі повинні досліджувати, як краще вони можуть зобразити сцену віртуально, а не зосереджуватися на контурах об’єкта. Чим більш базові та загальні конструкції об’єктів, тим краще в цій галузі, оскільки неточні деталі

можуть ввести дослідника в оману в процесі прийняття рішення. Підготовчий етап складається з п'яти компонентів:

• Ідея

• Оповідання

• Розкадровка

• Анімація

• Дизайн

Ідея

Ідея для проекту може виникнути звідусіль і практично з будь-якого місця. Він може бути викликаний словом, реченням, запахом, кольором, звуком, мимохідною розмовою з незнайомцем або підслуховуванням чужої розмови. Цієї іскри достатньо, щоб розпалити діалог всередині вас або з іншими, які хочуть реалізувати цю ідею. Після появи ідеї необхідно докласти мінімум зусиль, щоб перевірити ідею, поділившись з іншими та самим, щоб ми могли знати труднощі, які можуть виникнути в майбутньому. Маючи прихильність до ідеї, ми повинні правильно працювати, щоб втілити ідею в реальність. Отримавши гарну ідею, ми повинні з’ясувати, які персонажі будуть частиною цього проекту, які потребують людських і обчислювальних ресурсів для цього проекту.

Історія

Сценарій або сценарій є формальною письмовою формою фінального оповідання. Він написаний основними рухами персонажів, середовищем, часом, діями та діалогами. Цей сценарій призначений для команди підготовки та виробництва, щоб створити візуальне уявлення про всю історію. Багато людей з цих команд швидко збирають інформацію.

Сценарій є основою всього проекту. Побачивши це, режисер або будь-які члени команди можуть знати, що їм робити далі, і навіть можуть переконатися, що робота, виконана до цього моменту, відповідає історії.

Розкадровка

Розкадровка — це візуальна форма всієї історії. Він містить ранні ідеї постановки камери та ранні проекти потенційних візуальних ефектів. Кожне зображення в розкадровці візуально описує важливу сцену в історії зі сценарію. На рисунку показано приклад розкадрування. Розкадровка виявляє прогалини та діри безперервності в історії, які потрібно виправити.

Розкадровка використовується всією виробничою командою, щоб візуально зрозуміти, що відбувається в проекті. Розкадровка — це як план анімаційного проекту, який описує, як мають виглядати сцени перед початком проекту. Такі галузі, як візуалізація продуктів та архітектура, використовують розкадровки, щоб показати, як продукт повинен найкраще описувати, або найкращий спосіб показати інфраструктуру будівлі.

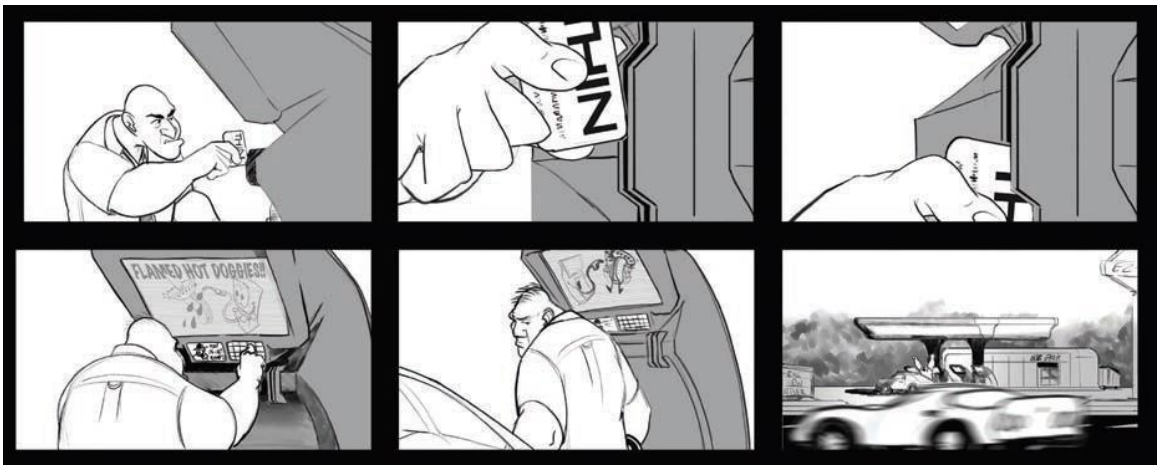


Рисунок 1.5 - Приклад розкадрування

Аніматик — це коротка анімація всієї історії. Він створюється шляхом перегляду зображень розкадровки з короткими діалогами та простими звуковими ефектами, щоб показати хід проекту. Художники попереднього виробництва ходять взад і вперед серед розкадрування та анімації, щоб створити остаточне редагування всього проекту.

Аніматика дозволяє режисеру та редактору створити стиль редагування та темп для всього проекту. Може здатися, що проект повністю завершений, коли

ми бачимо окремі кадри, але коли ми їх складаємо за сценарієм, вони можуть не працювати візуально. Animatic використовується для попередньої візуалізації візуальних ефектів у фільмах живої дії. Продюсери цих проектів люблять знати, як будуть виглядати всі кадри, перш ніж вони почнуться зйомки. Це допомагає режисеру та оператору знати, як сцена має бути знята, щоб відповідати візуальним ефектам. Це також допомагає акторам зрозуміти, як вони будуть взаємодіяти з віртуальним персонажем у фільмі, якого насправді немає в реальному світі.

Дизайн

У цьому компоненті вирішується остаточний вигляд персонажів, дизайн реквізиту, одягу та дизайну середовища. Художник може намалювати багато варіантів дизайну персонажів за день, а потім доопрацювати дизайн; професійному 3D-художнику потрібно щонайменше тиждень, щоб створити остаточну модель. Ми дізнаємося, як буде виглядати персонаж, і можемо внести зміни, якщо потрібно, перш ніж створювати його в програмному забезпеченні 3D-анімації.

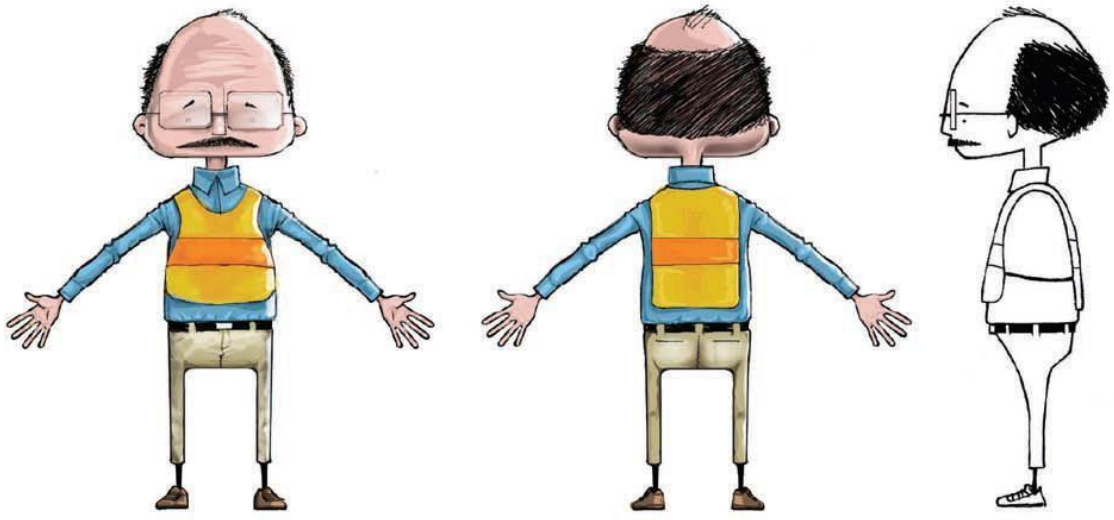


Рисунок 1.6 - Просте мистецтво персонажів у різних видах

1.7 Опис етапу виробництва

На стадії виробництва всі сценарії, розкадровки та дизайни передаються відповідним художникам для створення анімації. Стадія виробництва стає набагато легшою для створення проекту, якщо етап підготовки до виробництва виконано добре. На етапі підготовки до виробництва необхідно вирішити якомога більше проблем і прийняти дизайнерські рішення, щоб художникам-постановникам не доводилося здогадуватися, що вони мають створювати. Етап виробництва включає в себе наступні компоненти:

- Макет
- Дослідження та розробки (R&D)
- Моделювання
- Текстурування
- Такелаж/Налаштування
- Анімація
- Візуальні ефекти (VFX)
- Освітлення та візуалізація

Макет

На цьому етапі визначається форма і розмір об’єкта. Починаючи з етапу підготовки до виробництва, верстальщик бере необхідну інформацію, таку як форма та розмір об’єкта, і починає просту анімацію об’єктів. На цьому етапі моделі не розраховані на такі деталі, як обличчя чи пальці. Потрібна основна інформація про трансформацію анімації, яка показує, як персонаж рухається з однієї точки в іншу або напрямок, у якому персонаж звернений до камери. Основна анімація дозволяє режисеру починати проектувати композицію кожного кадру з точки зору камери. 3D-макет також буде розширюватися разом із рештою виробництва в міру просування виробництва. Етап 3D-макетування дозволяє іншим компонентам етапу виробництва почати працювати раніше, ніж зазвичай.

Дослідження та розробка

Компонент досліджень і розробок проходить через весь конвеєр 3D-анімації. Художники з різних компонентів сидять разом із технічними директорами, щоб обговорити майбутні технічні проблеми в проекті. Наприклад, у фільмі Pixar у пошуках Немо команда досліджень і розробок мала створити вигляд води, включаючи об'єкти, які присутні у воді, як-от рослини та риби. На момент виробництва цього фільму жоден 3D-анімаційний фільм не показував воду, оскільки це було складно контролювати та ефективно відображати. Тим не менш, у Finding Nemo майже на кожному кадрі є чудові анімаційні кадри у воді. У фільмі «Монстри проти прибульців» від DreamWorks Animation дослідницькій команді довелося створити персонажа B.O.B., синього драглистого персонажа, і керувати ним. Дуже складно створити персонажа, який не мав справжньої форми, але його потрібно сформувати, коли це необхідно. Команди дослідницьких і розробників працюють над наступним кращим і створюють ефекти, яких ми ще не бачили.

Моделювання

Модель — це геометрична поверхня, яка представляє об'єкт, який можна створювати, повертати та переглядати за допомогою програмного забезпечення для 3D-анімації, такого як Maya від Autodesk. Модель також можна створити шляхом сканування реального об'єкта за допомогою сканера та створення 3D-подання цього об'єкта в програмному забезпеченні для 3D-анімації. Існують різні типи програмного забезпечення для моделювання в індустрії 3D-анімації, і кожен з них використовується для певних цілей. Індустрія розваг використовує Maya, 3ds Max або Blender від Autodesk. Це програмне забезпечення може виконувати весь етап виробництва: моделювання, анімацію, фізичне динамічне моделювання, текстурування, освітлення та візуалізацію. Індустрія архітектури та візуалізації продуктів використовує Autodesk AutoCAD, продукти Robert McNeel & Associates Rhinoceros і Dassault Systemes SolidWorks, а також Google SketchUp Pro для створення об'єктів, характерних для цієї галузі. Це програмне

забезпечення не може виконувати анімацію, динаміку та рендеринг, але чудово справляється з моделюванням об’єктів у точному масштабі. Юридичні та медичні галузі можуть використовувати будь-яке з вищезазначених програм, залежно від їхніх вимог до конкретного проекту. Після того, як моделі повністю розроблені, моделі відправляються художнику по текстурі для фарбування та розтушовування.

Текстурування

Художники текстур застосовують колір і характеристики поверхні до геометричних моделей у компоненті текстурування. 3D-моделі зазвичай надходять до художника текстури в кольорі за замовчуванням програмного забезпечення для 3D-анімації. Робота художника текстури полягає в тому, щоб зробити екстер’єр моделі схожим на її реальний об’єкт. Наприклад, якщо модель являє собою дерев’яний стіл, художник переконається, що стіл при візуалізації виглядає як дерево, з якого він буде виготовлений.

Художники текстур використовують різні програми та методи для завершення своєї роботи. Художник може намалювати ці текстури вручну або використовувати зображення, щоб додати текстуру до об’єкта. На рисунку показано об’єкт до текстурування у сірому кольорі за замовчуванням у програмному забезпеченні 3D, і той самий об’єкт після фарбування текстур.



Рисунок 1.7 - Остаточна модель перед текстурами (ліворуч) і після застосування текстур (праворуч)

Такелаж/налаштування

У такелажі контрольна установка розміщується в об’єкті, щоб аніматори могли переміщати об’єкт.

Завдання монтажника — створити систему контролю, яка допоможе аніматорам працювати якомога швидше та ефективно. Кожен об’єкт, який рухається в проекті 3D-анімації, матиме якусь систему керування для керування ним.

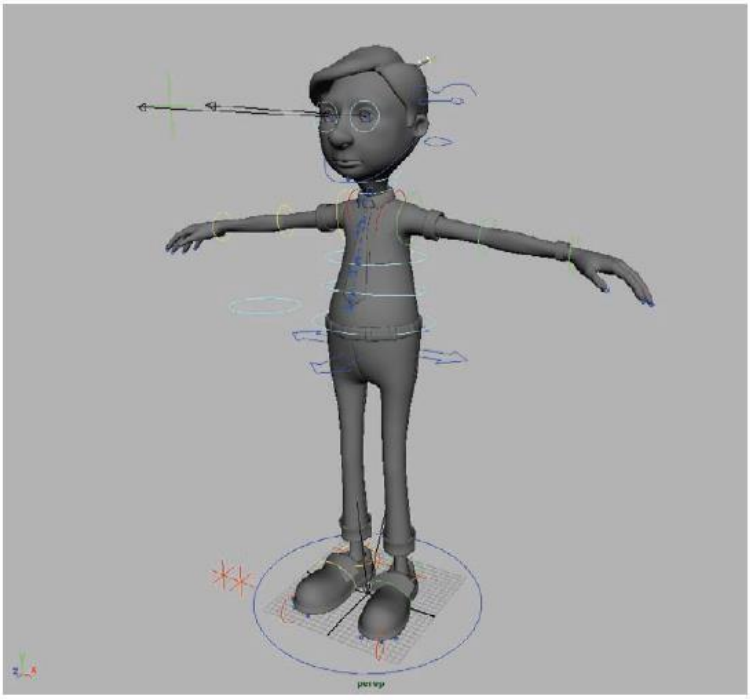


Рисунок 1.8 - Обладнаний символ

Ця система керування може керувати простою ієрархією батьків і дітей, суглобами, контролерами, шкірою, м’язовою системою та плаваючим GUI (графічним інтерфейсом користувача), щоб допомогти аніматору вибирати та створювати ключові кадри. Цей компонент дуже технічний і вимагає від художника з хорошими навичками вирішення проблем. На рисунку показаний кінцевий об'єкт персонажа з контрольною установкою. Лінії навколо об’єкта використовуються аніматорами і допомагають їм переміщати об’єкт.

Анімація

У анімації створюється рух предметів або персонажів. Аніматори персонажів змушують глядачів повірити, що те, що вони бачать на екрані, є справжнім і живим. Аніматори персонажів повинні розуміти фізичні властивості персонажа і впроваджувати те саме в проект. Типами анімації є анімація з ключовими кадрами (аніматор встановлює кожну позу і визначає їх відповідно до часу), анімація шляху (аніматор встановлює криву як шлях для об'єкта), захоплення руху (аніматор фіксує рух з реального світу від актора до контрольної установки), а також процедурну анімацію (програміст розробляє набір правил, і об’єкт рухається відповідно до цих правил). Компонент анімації є найбільш трудомістким компонентом етапу виробництва.

Візуальні ефекти (VFX)

Художник 3D-візуальних ефектів (VFX) анімує властивості персонажа та навколишнє середовище, наприклад, хутро, волосся, тканину, вогонь, воду та пил. Більшість візуальних ефектів засновані на динамічному фізичному механізмі всередині програмного забезпечення 3D, яке має природні властивості, такі як повітря, гравітація та опір, щоб керувати цими ефектами. Отже, базові знання з фізики та математики є абсолютно необхідними. Художник 3D VFX працює над різними типами ефектів, щоб покращити сцену, в якій вони перебувають, а не домінувати над ними.

Освітлення та візуалізація

Художники з освітлення розглядають колірні дизайни на стадії підготовки до виробництва і вставляють освітлення, щоб створити настрій для сцени або послідовності. Освітлення в програмі 3D-анімації подібне до реального освітлення фільму або фотографії. Художники 3D мають доступ до кількох типів світла, які імітують світло в реальному світі, наприклад, навколишнє світло, прожектор, місцеве освітлення та сонячне світло. Після налаштування всіх світильників художник розбиває весь проект на фрагменти візуалізації, які є окремими частинами процесу візуалізації. Ці окремі проходи знову з’єднуються разом у складовому компоненті на етапі постпродакшну.

1.8 Етап постпродакшн для 3D проекту

Постпродакшн є останнім етапом проекту 3D-анімації. На цьому етапі проект виглядає витонченим і професійним. Внесення змін у постпродакшн є більш економічно ефективним, ніж витрачати години часу на рендеринг, щоб виправити знімок. Постпродакшн може бути швидким і простим процесом, якщо підготовчі та виробничі групи запланували вирішення проблем, які можуть виникнути під час постпродакшн. Постпродакшн етап складається з таких компонентів:

- Композитинг
- 2D візуальні ефекти
- Корекція кольору
- Кінцевий результат

Складання

Під час композитингу художники шарують усі створені та зняті зображення, щоб створити остаточне вихідне зображення. Розшарування може бути простим завданням з кількома шарами, або це може бути складним завданням із сотнями поєднаних між собою шарів. Остаточні зображення можуть бути створеними 3D зображеннями або змішаними 3D та 2D зображеннями.

2D візуальні ефекти та анімаційна графіка

2D-візуальні ефекти та анімаційна графіка зазвичай поєднуються на етапі композитингу. І композитор, і художники 2D візуальних ефектів можуть бути однаковими. Художник додасть такі ефекти, як переходи між сценами, додавання титульних карток на початку та в кінці проекту. Завдання художника-моушн-графіки — створити елементи графічного дизайну кадру, якщо це необхідно. Цей тип роботи можна переглянути в заголовках фільмів.

Корекція кольору

Корекція кольору, також відома як синхронізація кольору або градація кольору, гарантує, що всі кольори сцени є однаковими. Усі проекти 3D-анімації створюються як окремі кадри, а потім створюються у вигляді послідовностей і, нарешті, як цілий проект. Щоразу, коли камера змінює, створюється знімок, і в кінцевому фільмі є тисячі знімків. Кожен кадр потрібно коригувати, щоб отримати вигляд фільму.

Остаточний результат

Кінцевий вихід 3D-анімації може бути в різних типах: фільм, відео, швидке прототипування, 3D стереоскопічна плівка та друковані медіа. Кожен із цих типів виводу має різні робочі процеси та технічні обмеження. Але найпоширенішим типом виводу є відео, яке можна відтворювати на комп’ютері.

2 АЛГОРИТМІЧНІ ТА МАТЕМАТИЧНІ РІШЕННЯ ДЛЯ ЗАДАЧ СТВОРЕННЯ 3D ПРОЕКТІВ

Попередня робота над створенням 3D-сцени вимагає або попередньо відформатованого введення, або використання анотованих об’єктів і сцен. 3D-моделі, що зображують предмети реального світу, також мають бути анотовані різними характеристиками, щоб можна було використовувати пріоритети для їх взаємовідносин один з одним. Просторові відносини відіграють вирішальну роль у розміщенні об’єктів у сцені, яка показує, як об’єкти пов’язані. Попередні дослідження припускали, що це знання світу надається за допомогою ручного анотування, або воно має обмежений дискурс системи «текст-сцена» та виробничі можливості, щоб зменшити необхідне знання світу.

Замість того, щоб мати описи, які описують розташування об’єктів для окремих сцен, використовуючи анотації вручну або вводячи в систему в іншому форматі, ця робота зосереджується на тому, щоб показати, що природна мова може використовуватися для створення 3D-сцен шляхом вилучення необхідної інформації з тексту. .

На основі типу вихідних даних, які генеруються системою генерації 3D-сцени, дослідницьку роботу в цій галузі можна класифікувати як текст-зображення, текст-сцена, текст-відео, текст-анімація та їх зворотний процес. також. У своїй роботі ми в основному зосереджуємось на тексті-сцені та тексту-анімації; Після кількох століть розвитку комп’ютерного обладнання та програмного забезпечення для 3D-анімації ми вважаємо, що дуже своєчасно провести дослідження в цих областях.

2.1 Текст до сцени

Words eye є піонером серед усіх систем, які генерують 3D-сцени з текстових описів. WordsEye базується на конвеєрі, який перетворює текст у

семантичне уявлення, яке можна відобразити, вибираючи й упорядковуючи моделі з бази даних моделей, щоб сформувати тривимірну сцену. Автори WordsEye продовжили вдосконалювати систему, з наступними документами, щоб надати більше інформації про те, як система працює.

WordsEye є загальнодоступним онлайн за адресою https://www.wordseye.com/.

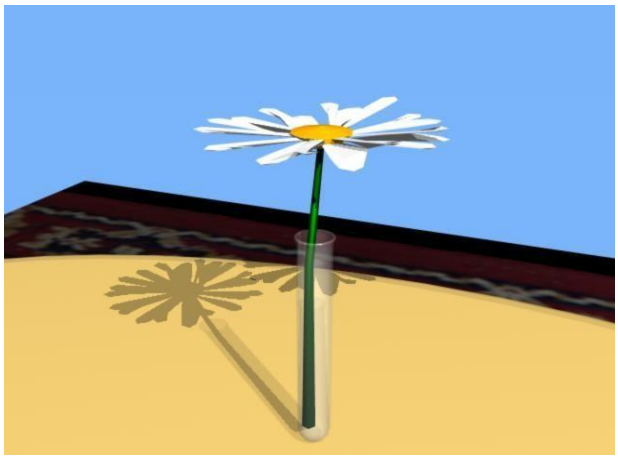


Рисунок 2.1 - Ромашка в пробірці



Рисунок 2.2 - Птах у клітці. Клітка для птахів стоїть на стільці

База даних моделі з 2000 тривимірних об’єктів використовується системою WordsEye 2001 року. Найважливіше те, що автори WordsEye визнали важливі ключові аспекти семантики та атрибутів графічного представлення, які необхідні для цього завдання. Основний внесок WordsEye полягає в тому, щоб

надати велику колекцію найбільш значущих семантик об'єктів, необхідних для створення тексту сцени: розміри, орієнтація, частини (розмір яких об'єктів потрібно змінити або змінити їх орієнтацію під час розміщення їх у сцені), просторові відносини. (які об’єкти розміщуються в сцені для стосунків «включено» чи «під»), пози скелета та функціональні теги. Ці атрибути пов’язані з семантичними зображеннями вхідного тексту, розробленими за допомогою сутностей, дій, атрибутів і відносин.

WordsEye розроблено в LISP (LISP означає обробку списків). WordsEye витягує інформацію з введеного тексту, позначаючи та аналізуючи. Потім він створює структуру залежності, яка перетворюється на семантичне представлення. Попередньо визначені правила зображення використовуються для розміщення тривимірних об’єктів у сцені із заданими атрибутами кольору та просторовими відносинами з набору низькорівневих зображувачів, які перетворюються із семантичного представлення.

Система PUT є основною ідеєю для всіх систем генерації тексту в 3D сцени. Через обмеження в обробці природної мови та графіки до розробки PUT аніматори зіткнулися з великими труднощами при використанні природної мови для геометричного опису. PUT пропонує інтерактивне середовище моделювання та інтерфейс програмування, який поєднує природні просторові відносини з прямими маніпуляціями для розміщення жорстких об’єктів у 3D-сценах.

Система PUT використовує моделі глядача, орієнтованого на глядача, як дві маніпуляційні метафори. Модель глядача «Боже око» використовує координати на основі світу, а модель, орієнтована на глядача, використовує систему координат глядача. У світових координатах координати всіх об’єктів пов’язані зі світом, тоді як у системі координат глядача, яку також можна назвати точкою зору камери, координати світу перетворюються на глядачів, на основі яких переглядаються користувача. Параметри розміщення об’єктів отримують з динамічних, неописових структур, які називаються схемами

зображень. Схеми зображень — це структури, засновані на моделях взаємодії людини з фізичним світом, таких як рухи тіла, взаємодії об’єктів та просторові орієнтації. Просторові відносини та рух є найбільш фундаментальними схемами зображення. Для пояснення структури та застосування просторових понять використовуються схема зображення та зображення-перетворення. Прості схеми можна комбінувати, щоб утворити складну схему шляхом перетворення зображень. Таким чином, для структурування групи складних понять потрібна відносно невелика кількість схем зображень.

PUT — це синтаксичний аналізатор C++, який використовує ці схеми зображень для просторового розміщення об’єктів. Опис регіону щодо еталонного об’єкта є більш детальним у визначенні основних просторових відносин. Еталонним об'єктом може бути вже існуючий об'єкт або довільна точка в двох або трьох вимірах. У системі PUT використовується наступна термінологія: еталонний об'єкт називається орієнтиром (LM), а об'єкт, який розміщується, називається траєктором (TR). Просторові відносини пов'язують TR з LM. Щоб указати розташування TR, використовуються дієслова розміщення, такі як «put». Нижче наведено деякі команди розміщення об’єктів, які використовуються в цій системі.

Поставте «box1» на «box2» нижче «box3». повісьте «box1» на «box2» на (0, 10). помістіть «box1» над «box2» та «box3» представили систему генерації тексту для 3D-сцени, яка використовує XML як вхідні дані для Autodesk Maya для генерації 3D-сцени. Ця система складається з трьох модулів: мовного двигуна, об’єктної бази даних і графічного движка.

У мовному механізмі семантична інформація про об’єкти та просторові відносини витягується з вхідного тексту. Лі та ін. виконував ручну анотацію текстових описів для створення XML, який можна використовувати як синтаксичний аналізатор, який можна легко інтегрувати в інші програми. На рисунку наведено приклад XML-файлу для вхідного тексту.

```
<?xml version="1.0" ?>
- <kitchen>
    <table attribute="middle" />
  - <chair01>
      <glorelpos relatedTo="table">directly-north-of</glorelpos>
      <reldis relatedTo="table">adjacent-to</reldis>
      <relori relatedTo="table">directly-facing</relori>
    </chair01>
  - <chair02>
      <glorelpos relatedTo="table">directly-west-of</glorelpos>
      <reldis relatedTo="table">adjacent-to</reldis>
      <relori relatedTo="table">directly-facing</relori>
    </chair02>
  - <fridge>
      <glorelpos relatedTo="table">northeast-of</glorelpos>
      <reldis relatedTo="table">near</reldis>
      <relori relatedTo="table">facing</relori>
    </fridge>
  - <cabinet>
      <glorelpos relatedTo="table">northwest-of</glorelpos>
      <reldis relatedTo="table">near</reldis>
      <absori relatedTo="air">facing-direct-south</absori>
    </cabinet>
  </kitchen>
```

Рисунок 2.3 - Приклад файлу XML як вхідні дані [20]

«Стіл знаходиться посередині кухні. Один стілець розташований на північ від столу, поруч зі столом і прямо проти нього. Інший стілець розташований на захід від столу, поруч зі столом і прямо проти нього. Холодильник розташований на північний схід від столу, біля столу та навпроти нього. Шафа розташована на північний захід від столу і біля столу, і прямо на південь».

Цей XML-документ надається як вхідні дані в систему для створення 3D-сцени. Сутності представляють об'єкти, а атрибути вказують на просторові зв'язки. Потім інформація в документі XML перетворюється на семантичну веб-структуру, яка показує, як об’єкти пов’язані один з одним із просторовими відносинами в сцені. Для кожного об’єкта в семантичній мережі виділіть просторове відношення, пов’язане з цим об’єктом, і розмістіть об’єкт у сцені

відповідно до нього. Остаточна сцена кухні з введеного тексту показана на рисунку.

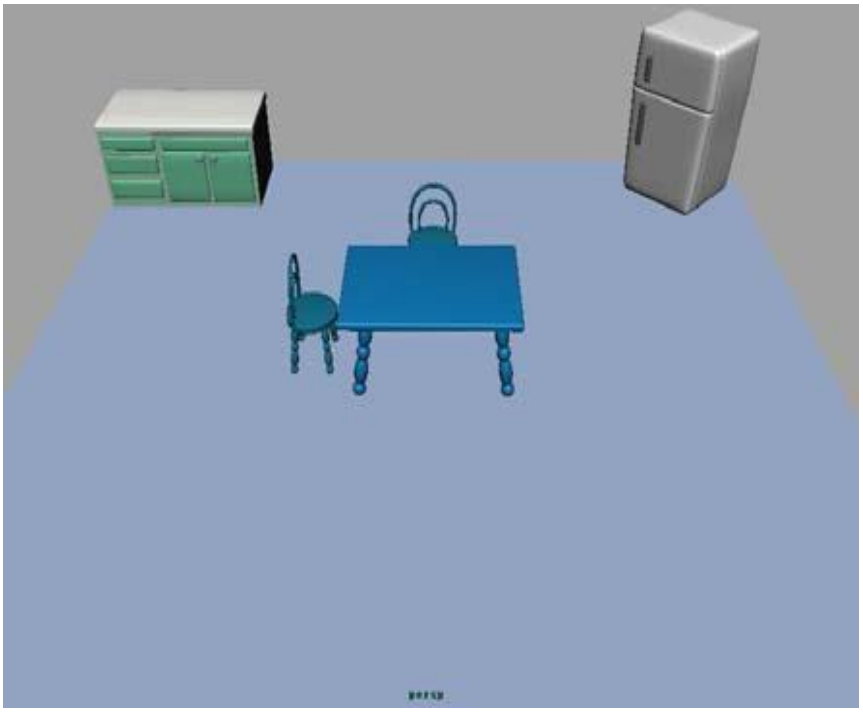


Рисунок 2.4 - Згенерована 3D-сцена

Нова платформа для автоматичної побудови тривимірної сцени з текстового опису, яка є розширенням роботи, де вони використовували обмежувальну рамку для розміщення об'єктів у сцені, що допомагає зменшити зіткнення між об'єктами у відносно великій сцені.

Чанг та ін. запропонував інтерактивне навчання просторових знань для системи генерації тексту в 3D сцени. Він використовує 3D-моделі внутрішніх сцен, які мають текст, пов’язаний з ними у вигляді імен і тегів. Вони встановлюють ймовірність появи батьківського об’єкта, коли дана категорія дочірнього об’єкта присутня у введеному тексті. Наприклад, спостерігаючи за виделками та тарілками на столі більшу частину часу, ми можемо визначити, що стіл потрібно імпортувати, коли є тарілки та вилки. Батьківські об’єкти сегментуються на плоскі поверхні, щоб визначити, які поверхні батьківських об’єктів підтримують дочірні об’єкти. Поверхні класифікуються на вгору, вниз, ліворуч, праворуч, спереду, на, ззаду за напрямком їх нормалі. Просторові

зв'язки між об'єктами виділяються у вигляді on(A, B) або left(A, B), де A і B — об'єкти, a on і left — просторові відносини. Система використовує конвеєр Stanford CoreNLP для обробки введеного тексту. Зіставляючи слова в тексті зі списком відомих типів сцен із набору даних сцени, визначається тип сцени. Щоб ідентифікувати об’єкти, система шукає фрази іменників і використовує заголовне слово як категорію. Стенфордська система відліку використовується для визначення того, коли посилається на той самий об’єкт. Щоб ідентифікувати властивості об’єкта, прикметники та іменники виділяються у фразі іменника, а також узгоджуються з шаблонами залежностей, щоб отримати більше атрибутів та ключових слів. Беручи до уваги категорію та атрибути об’єкта, а також інші слова у фразі іменника, що згадує об’єкт, ми визначаємо набір пов’язаних ключових слів, які будуть використані пізніше для запиту бази даних 3D-моделі. Для виділення просторових відносин між об’єктами використовуються шаблони залежностей.

На додаток до оригінального введеного тексту, наданого користувачем для створення сцени, користувач може давати текстові команди для вибору та зміни об’єктів для фільтрації сцени. Наприклад, користувач може давати команди «зняти стілець» або «поставити горщик на стіл».

У продовження попередньої роботи, Chang et al. запропонував Навчання просторових знань для генерації тексту в 3D-сцени. У цій роботі він створив систему для вивчення пріоритетів появи об’єктів, які допомагають визначити тип сцени. Наприклад, вхід є «на столі є ніж», тоді система створює сцену кухні з іншими пов’язаними об’єктами. Чанг та ін. моделювали взаємне розташування об'єктів на основі їх категорій об'єктів і типу сцени.

Чанг та ін. запропонував метод лексичного обґрунтування для отримання ваги лексичних ознак шляхом навчання класифікатора відповідної задачі заземлення для створення сцени. Потім вони об’єднали цей метод лексичного обґрунтування з попереднім методом (модель на основі правил) для покращеної системи генерації сцени. Кожна іменникова фраза витягується за допомогою

методу, заснованого на правилах; Чанг та ін. витягнув ознаки та обчислив оцінку категорії. Якщо категорія перевищує 0,5, вони вибирають найкращу категорію для представлення фрази-іменника. В іншому випадку вони використовують заголовок фрази-іменника для категорії об’єкта. Вони порівнюють результати з попередніми моделями в п’яти категоріях: випадкові, вивчені, правило, комбіновані, людські. У випадкових умовах вони вибирають деякі випадкові об’єкти для розміщення в сцені. Вивчена умова вибирає чотири найбільш ймовірні об’єкти, а умова правила використовує підхід, заснований на правилах, для створення сцени. Умова комбінації використовує як лексичні, так і засновані на правилах методи для створення сцени. Люди створюють свою власну сцену в людському стані. Порівняння між 3D-сценами, створеними різними моделями, можна побачити на рисунку 2.5.

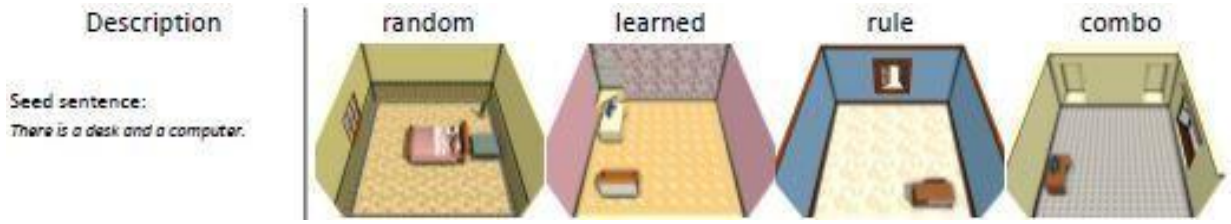


Рисунок 2.5 - Сцена, створена чотирма різними методами [25]

У 2017 році Chang et al. запропонував метод SCEENSEER , який може модифікувати або змінювати об'єкти в створених сценах. Враховуючи сцену S і операцію O, ми хочемо знайти сцену S', подібну до S з операцією O.

Запропонована система підтримує операції Select, Delete, Replace, Move, Scale і LookAt: Select використовується для вибору різних об’єкти, Delete використовується для видалення об’єктів зі сцени, Replace використовується для заміни одного об’єкта іншим, Move використовується для зміни розташування об’єкта, Scale використовується для зміни розміру об’єктів і LookAt використовується для створення фокусування камери на вибраних об’єктах.

2.2 Текст до анімації

На відміну від генерації статичної 3D-сцени, генерація 3D-анімації є більш складною. Окрім проблеми розбору природної мови для семантичного представлення, існують додаткові проблеми, як-от послідовність подій в анімації. Введений текст необхідно проаналізувати, щоб визначити об’єкти та дії, пов’язані з ними. Найскладніша частина – це спланувати послідовність подій, щоб об’єкти не зіткнулися або не змінили свій шлях без потреби.

Текст до анімації значно складніший. Деякі з цих структур намагаються не лише забезпечити візуальну анімацію, а й створити музику та розповідь. Окрім проблеми аналізу природної мови до семантичного представлення, існують додаткові проблеми в оповіді, такі як вибір точок огляду камери, освітлення тощо. Крім того, планування для тимчасово розширених послідовностей дій вводить ще більш складні параметри для вирішення неясностей та обмеження.

Багато проблем генерації тексту до сцени також можна знайти в анімації (також відомій як художня література до анімації). Вони включають: перетворення та обґрунтування текстового вмісту природної мови до змістовних уявлень, визначення неявних обмежень, відображення слів до графічних примітивів та візуалізацію примітивів у контексті генерації анімації.

Ще одне завдання, яке ускладнюється, — це тлумачення дієслів і дій. Є більше неявних тверджень і незгаданих знань здорового глузду, що стосуються наслідків і обмежень повсякденних дій. Наприклад, навіть для створення статичного зображення «Олексій чистить зуби» потрібно розуміти, що потрібно стояти перед дзеркалом і раковиною з зубною щіткою в руці. Сучасні стратегії НЛП ще не є адекватними для комп’ютеризованого анотування практичних сценаріїв, що описують історії, які потрібно анімувати.

З боку графіки автоматичне позування персонажів і моделювання реалістичних дій людини набагато складніше, ніж комп’ютеризоване проектування статичних сцен. Отже, позування та анімація персонажів в

індустрії розваг переважно виконуються руками досвідчених 3D-художників за величезну ціну. Автоматизованим системам майже ніколи не довіряють проект зображення переконливих людських почуттів у віртуальних персонажах без людської допомоги чи перевірки. З'єднання колективних послідовностей анімацій та планування того, як вони будуть реалізовані в конкретному геометричному середовищі, також є складним завданням. Навіть просте твердження на кшталт «Чоловік хапає книгу на столі» важко, якщо чоловік сидить на дивані в іншій частині кімнати. Важливо намалювати послідовність рухів, яка змушує чоловіка встати, підійти до книги, реалістично переміщаючись по меблях, а потім взяти книгу в позу. Більше того, якби ми хотіли відобразити деяку частину широкого діапазону людських емоцій, нам було б важко зробити це за допомогою сучасних графічних систем, оскільки вимагалося б значного ручного покращення анімації персонажів.

Перетворення тексту в анімацію – це наступний крок, який включає створення статичних сцен. Додавання анімації до статичних сцен робить її складною та завершеною.

Система CARSIM створює анімацію фактичного місця ДТП, яка підсумовує звіти про автомобільну аварію, написані французькою мовою. Він також реалізує лінгвістичний аналіз, тобто витягує корисну інформацію з тексту, витягує з бази даних об'єкти, які мають бути присутніми в сцені, і створює 3D-анімацію відповідно до опису сцени.

У системі CARSIM сцена складається з трьох категорій об'єктів: нерухомі об'єкти (СТАТИЧНІ), рухомі об'єкти (ДИНАМІЧНІ) та зіткнення (АВАРІЯ). СТАТИЧНІ та ДИНАМІЧНІ об'єкти описують загальне середовище, в якому відбувається аварія. Кожне зіткнення представлено зв'язком між двома об'єктами, які знаходяться або в ДИНАМІЧНОМУ та/або СТАТИЧНОМУ стані. Для статичного об'єкта можна визначити два параметри: один з характером об'єкта, а інший з його розташуванням. Статичним об'єктам, які беруть участь у аварії, присвоюється параметр ідентифікації (ID), тоді як

об’єкти ROAD мають параметр KIND, який залежить від характеру дороги. Можливі значення KIND для цієї системи: перехрестя, пряма дорога, turn_left і turn_right. Інші можливі статичні об’єкти – дерева, світлофори, знаки зупинки та пішохідні переходи. Параметр COORD використовується для визначення їх розташування. Оскільки дерева та світлофори також можуть брати участь у зіткненнях, вони також мають посвідчення особи. Світлофори мають параметр COLOR, щоб позначати колір світла (червоний, оранжевий, зелений або неактивний).

У разі динамічних об’єктів вони не можуть бути лише за своєю природою та положенням. Крім характеру та положення, необхідно визначити їх рух. Кожен динамічний об’єкт представлений параметром VEHICLE, параметром KIND (автомобіль або вантажівка) та унікальним ідентифікатором ID. Рух динамічного об'єкта визначається двома параметрами: ПОЧАТКОВИЙ НАПРЯМ і ПОДІЯ. ІНІЦІАЛЬНИЙ НАПРЯК визначає напрямок об’єкта, який прямує до, а параметр EVENTS – це впорядкований список переміщень атомів. CARSIM має такі події: рух вперед, зупинка, поворот ліворуч, поворот направо, зміна смуги наліво, зміна смуги праворуч, обгін. Аварія описується за допомогою впорядкованого списку зіткнень, які відбуваються в симуляції аварії. Зіткнення описується шляхом надання назв двох об’єктів, які беруть участь у зіткненні, і деяких додаткових атрибутів. Цими атрибутами є місце зіткнення та частини транспортних засобів, які брали участь у зіткненні. Транспортний засіб, який зіткнувся, є актором, а транспортний засіб, який вдарився, називається жертвою. Між актором і жертвою відбувається зіткнення. Актор є динамічним об'єктом, а жертва може бути як статичним, так і динамічним об'єктом. З усіма цими параметрами система CARSIM формує місце ДТП із звітів. Наприклад, «Я їхав на перехресті з повільною швидкістю, приблизно 40 км/год. Автомобіль В приїхав зліва, проігнорував пріоритет справа та зіткнувся з моїм транспортним засобом. При першому ударі моє заднє крило з лівого боку вдарилося, і через слизьку дорогу я не впорався з

керуванням і врізався в металевий захист дерева, отже, відбулося друге лобове зіткнення»– це перетворений текст з Французькі звіти про ДТП. Цей текст записується як офіційний опис, а його відповідні результати показані на рисунку.

```
// Static objects
STATIC [
        ROAD [
                KIND = crossroads;
        ]
        TREE [
                ID = tree1; COORD = ( 5.0, -5.0 );
        ]
]
// Dynamic objects
DYNAMIC [
        VEHICLE [
                ID = vehicleB; KIND = car;
                INITDIRECTION = east;
                CHAIN [
                        EVENT [
                                KIND = driving_forward;
                        ]
                ]
        ]
        VEHICLE [
                ID = vehicleA; KIND = car;
                INITDIRECTION = north;
                CHAIN [
                        EVENT [
                                KIND = driving_forward;
                        ]
                ]
        ]
]
// Collision objects
ACCIDENT [
        COLLISION [
                ACTOR = vehicleB, front;
                VICTIM = vehicleA, leftside;
                COORD = ( 1.0, 1.0);
        ]
        COLLISION [
                ACTOR = vehicleA, front;
                VICTIM = tree1, unknown;
        ]
]
```

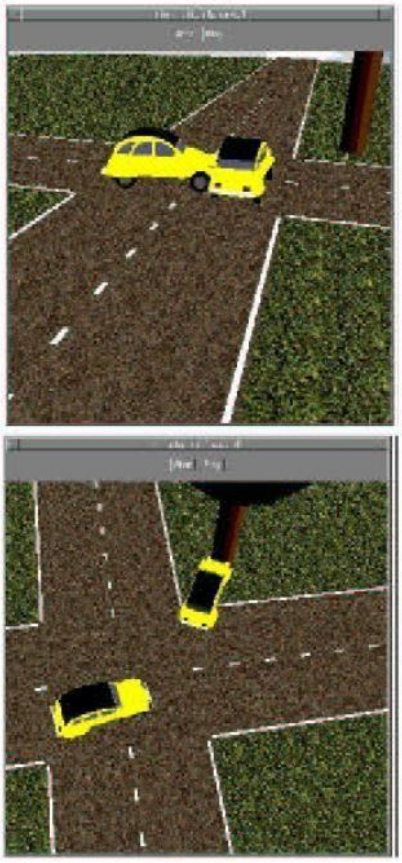


Рисунок 2.6 - Вхід системи CARSIM (ліворуч) і вихід (праворуч)

Свон запропонував систему, засновану на правилах, для створення анімації з обмеженої форми китайської природної мови. Ця система використовує п'ять рівнів уявлень, кожен з яких має відповідну мову з чітко визначеним синтаксисом. Система приймає вхідні дані як обмежену форму китайської природної мови під назвою Moon Light, яка трансформується в мову опису семантики на основі кадрів під назвою Golden Forest. Ця мова включає в себе середовище, символи, об’єкти та послідовність дій, виділених з тексту. Використовуючи базовані на знаннях і прагматичні міркування, система

створює якісний сценарій під назвою Rainbow, що складається з послідовності впорядкованих у часі сцен. Також заплановано рух камери та освітлення для кожної сцени. Високорівневий якісний сценарій потім перетворюється на кількісний сценарій під назвою Evergreen, який визначає серію статичних кадрів зображення, які потрібно створити, і включає конкретні характеристики об'єктів, рухів, камери, освітлення та дій. Нарешті, кількісний сценарій перетворюється на низькорівневу мову сценаріїв анімації під назвою SCRIPT, яку можна використовувати для відтворення остаточної анімації. Розробка цієї системи зажадала багато зусиль вручну; понад 30 аспірантів зробили внесок у проект протягом майже 10 років.

Glass представив систему, яка перетворює анотований художній текст у тривимірну анімацію. Анотований текст містить такі обмеження, як макет, зовнішній вигляд і рух, пов'язані з об'єктами в сцені.

На рисунку 2.7 наведено приклад анотованого художнього тексту. Щоб сегментувати введений текст на різні сцени, використовуються анотації.

They had it on the top of a hill, in a sloping field that looked down into a sunny **<setting>valley</setting>**. **<avatar>Anne</avatar>** didn't very much like a big brown **<object>cow</object>** who **<transition type='ARRIVE' subject='cow'>came</transition>** up **<relation type='near' subject='cow' object='her'>close<relation>** and stared at her, but it **<transition type='DEPART' subject='it'>went</transition>** away when **<avatar>Daddy</avatar>** told it to.

Рисунок 2.7: Анотований художній текст [30]

Анотації аватар і об'єктів використовуються для вибору списку об'єктів, які мають бути присутніми в кожній сцені. Анотації відносин і переходів пізніше використовуються для створення списку абстрактних обмежень, які підсумовують макет сцени. Дескриптор сутності створюється для кожного аватара та анотації об'єкта, які з'являються в сцені. Дескриптори сутності призначають геометричну модель, отриману з бібліотеки попередньо створених моделей, кожному аватару або об'єкту. Об'єкти фіксуються у невизначеному положенні, а аватарам призначаються траєкторії, які забезпечують переміщення

об’єктів. Співпосилання вирішуються шляхом збереження стану, в якому є останні згадані чоловічі або жіночі об’єкти. Абстрактні обмеження містять об’єкт, їхні відносини, такі як near, inside, no_collide, behind, in_front_of, to_left_of і to_right_of і часовий інтервал, який визначає, яке обмеження має виконуватися під час анімації. Приклад абстрактних обмежень показано на рисунку 16. Абстрактні обмеження потім перетворюються на математичні обмеження для створення сцени. Обмеження зі статичними об’єктами вирішуються спочатку, а потім – динамічними.

```
CONSTRAINT 1:       CONSTRAINT 2:       CONSTRAINT 3:       CONSTRAINT 4:
 Subject: MAN        Subject: MAN        Subject: MAN        Subject: MAN
 Relation: OUTSIDE   Relation: INSIDE    Relation: NEAR      Relation: NEAR
 Object: ROOM        Object: ROOM        Object: TABLE       Object: CHAIR
 Start-time: 0       Start-time: 5       Start-time: 9       Start-time: 14
 End-time: 5         End-time: 30        End-time: 14        End-time: 30
```

Рисунок 2.8: Приклад абстрактних обмежень

Базу даних кліпів для різних персонажів. Система витягує корисну інформацію та тимчасові обмеження з введеного тексту та шукає відповідний ролик у базі даних. Заплановано, що кліпи руху створять цілу анімацію відповідно до тимчасових обмежень. Спочатку над вхідним текстом виконується синтаксис і семантичний аналіз. Аналіз синтаксису дає деревовидну структуру з тегами фраз і залежностями, тоді як семантичний аналіз витягує інформацію про рух із тексту. Для кожного руху в тексті створюється один кадр запиту. Часові обмеження створюються системою, яка містить інформацію про час виконання між рухами. Планування руху визначає виконання кожного кліпу руху на основі тимчасового обмеження. Для кожного кадру запиту система шукає відповідний ролик у базі даних про рух. Створюється розклад руху, який містить кліпи рухів та час їх виконання. Цей розклад направляється на синтез руху, який створює остаточну плавну анімацію з необхідними об’єктами. Приклад системи Oshita та її результат показаний на рисунку 2.9.

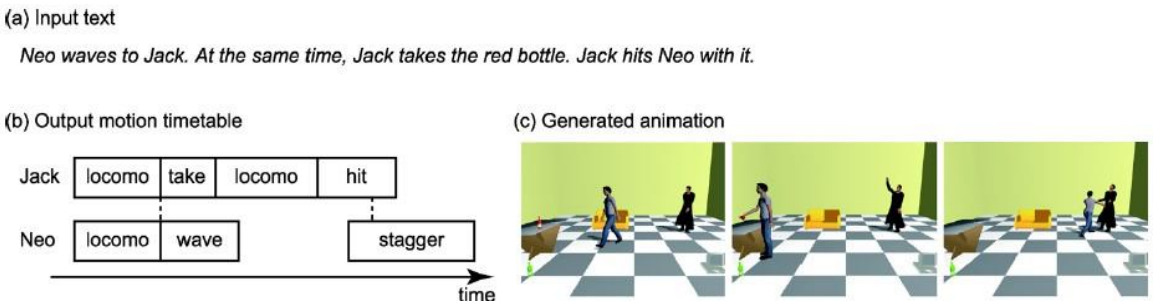


Рисунок 2.9: Приклад системи Ошіти

Autodesk Maya — це програмне забезпечення для 3D-анімації, яке використовується в більшості анімаційних індустрій. Спочатку Маya була розроблена системою Alias Software в 1998 році, а пізніше в 2005 році вона була придбана Autodesk і перейменована в Autodesk Maya. З моменту свого початкового випуску Маya стала широко використовуватися в кіно та ігровій індустрії. Він використовується для створення 3D-об'єктів, які зображують реальні об'єкти. Він також може створювати візуальні ефекти, які можна використовувати у фільмах та іграх. Autodesk Maya займається всіма етапами виробничого процесу від проектування моделей до їх візуалізації.

2.3 Архітектура на основі Маya API

Маya побудована на C++ API (Інтерфейс програмного забезпечення). Ми можемо додавати нові функції або змінювати наявні за допомогою команд Маya та інтерфейсів Маya API. Інтерфейс команд Маya може бути розроблений за допомогою MEL (вбудована мова Маya) або Python як мови сценаріїв для написання сценаріїв. Інтерфейс API Маya можна розробити за допомогою Python або C++ для написання плагінів для Маya.

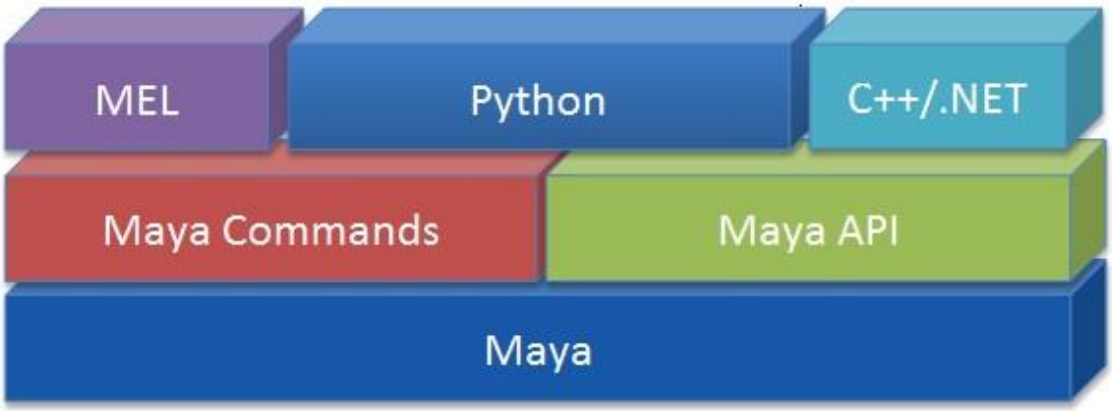


Рисунок 2.10: Архітектура Майя

Зараз ми розглянемо всі мови програмування та API всередині Maya. Maya має дві бібліотеки API. Перший - це бібліотека команд або скорочено cmds. Він складається з команд, які можна використовувати для створення більших додатків та інструментів. Інша бібліотека всередині Maya називається OpenMaya API. Це дає технічним розробникам глибокий доступ до внутрішньої архітектури Maya, але це пов'язано зі складністю, і художникам вона не дуже підходить для швидкого ознайомлення.

Таблиця 2.1 - Плюси та мінуси різних мов програмування в Maya

MEL	Python	C++ and C#
Специфічна Майя	Можна використовувати за межами Maya	Можна використовувати за межами Maya
Інтерактивний	Не можна використовувати інтерактивно	Не можна використовувати інтерактивно
Використовується лише для скриптів	Використовується для скриптів і плагінів	Використовується лише для плагінів
	Повільніше, ніж C++ і C#	Потрібно скласти
	Легко навчатися	Набагато краща продуктивність

Maya має чотири мови програмування. Перший – MEL. Це специфічна мова Maya, яка призначена для простого ознайомлення з художниками, і вона може використовувати лише бібліотеку команд. MEL — це інтерактивна мова,

де ми можемо вводити команди та бачити результати в режимі реального часу в редакторі сценаріїв у Maya.

Це протилежно іншій основній мові Maya C++, яка була використана для створення самої Maya. C++ є більш складною мовою, і вона може отримати доступ лише до OpenMaya API. Крім того, C++ має бути скомпільований завчасно, а це означає, що його можна розповсюджувати лише як плагіни і не можна розробляти в режимі реального часу. Як і C++, Maya має іншу мову під назвою C-Sharp. Підтримка C# у Maya обмежена Windows, і вона має ті самі застереження, що й C++. Його потрібно зібрати, і він може отримати доступ лише до OpenMaya API. Однак C# набагато легший у використанні, ніж C++, і він досить популярний в ігровій спільноті.

Четвертою мовою програмування в Maya є Python. Python займає унікальне положення, оскільки може отримати доступ як до відкритого API Maya, так і до бібліотеки команд. Це єдина мова всередині Maya, яка може це зробити. На відміну від MEL, Python також використовується програмістами, які не використовують Maya. Python досить популярний серед програмістів.

Google обрав Python для своїх технологій машинного навчання, Dropbox широко використовує його, а вчені використовують його для наукових досліджень. Python популярний з кількох причин. Це інтуїтивно зрозуміла мова з багатьма розширеними функціями, як C++, так і C-Sharp. Це відбувається за рахунок продуктивності. Python набагато легше використовувати, ніж C++ і C#, але він може бути набагато повільнішим. Зазвичай це компроміс, який варто зробити, оскільки код Python досить легко писати і часто не потребує швидкості, яку пропонують C++ або C#.

У цій дипломній роботі ми використовували мову Python для написання сценаріїв у Maya, щоб розширити її функціональні можливості. Оскільки не всі функції, надані Maya, можна отримати за допомогою Python, виникла необхідність використовувати MEL між ними. Команди MEL також можна використовувати в сценаріях Python, викликаючи функцію mel.eval().

Python 2.7 — остання версія серії Python 2, а Python 3.7 — остання версія серії Python 3. Python 2 не матиме додаткової активної розробки, за винятком деяких простих виправлень безпеки. Оскільки між версіями багато змін, вони несумісні одна з одною. Autodesk спочатку запустив Python 2.4 у Maya 8.5. Вони оновили версії Python до 2.7 до Maya 2014, і з цього моменту вони використовують ту саму версію Python.

Індустрія візуальних ефектів вирішила залишитися на Python 2.7 дотепер через значні інвестиції в екосистему Python 2, які було б важко перенести на Python 3. Основною причиною того, що індустрія візуальних ефектів все ще використовує Python 2.7, є мінімізувати несумісність між різними програмними пакетами. Тому, щоб розширити функціональні можливості Maya, ми використовували Python 2.7 як нашу мову програмування.

2.4 Опис мови розробки

Оскільки багато програм для анімації, включаючи Maya, все ще використовують Python 2.7, більшість розширених бібліотек не підтримуються. Компілятори всередині цих програм розроблені для виконання завдань, призначених для анімації. Пакети Python 2 для Windows компілюються за допомогою Microsoft Visual Studio 2008. Однак інтерпретатор Python від Maya компілюється в будь-яку версію Visual Studio Maya, в яку було скомпільовано. На рисунку показано різні версії Visual Studio, які використовуються Python в системі та в Maya. . Зверніть увагу, що MSC v.1500 і MSC v.1900 є різними компіляторами, які використовуються для цього інтерпретатора. Різні компілятори компілюють несумісний код. Це означає, що залежності більшості цих бібліотек аварійно завершуються, коли ми намагаємося запустити їх через Maya. Отже, щоб мати спеціальні бібліотеки, які не входять до складу Maya, ми повинні розробити власні бібліотеки та створити багато їхніх залежностей із

джерела, використовуючи відповідну версію Visual Studio, що є трудомістким і трудомістким процесом.

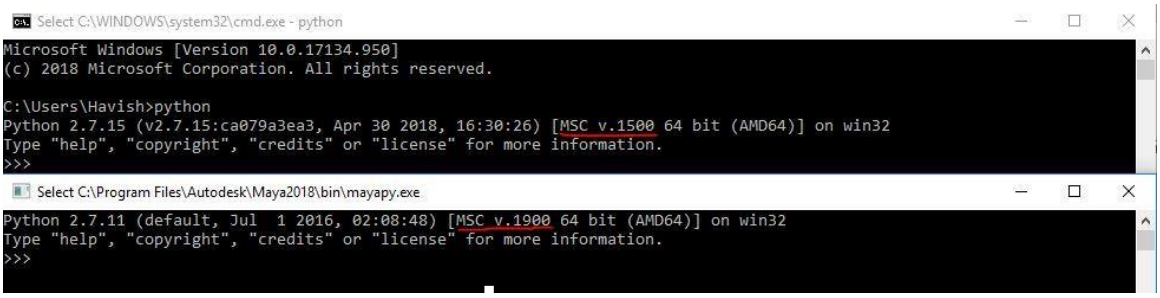


Рисунок 2.11: Різні версії Visual Studio, встановлені всередині Maya і системи

Надзвичайно складна робота полягає в тому, щоб обґрунтувати природну мову детальним зображенням реальних сутностей. Для вирішення цієї проблеми можна використовувати візуально розташовану структуру. У цьому контексті ми покращуємо розуміння системи, надаючи більше інформації про реальний світ. Ми використовували візуально розміщений фреймворк для створення 3D-сцен, тому що не вистачає мови здорового глузду, яка пов’язує мову з повсякденним середовищем. Наприклад, просторове відношення «включено» має бути візуалізовано залежно від контексту. Розміщення об’єкта відрізняється як у ситуаціях «на стіні», так і «на підлозі».

.

3 РОЗРОБКА МОДЕЛЕЙ ТА АЛГОРИТМІВ 3D СИМУЛЯЦІЙ НА ОСНОВІ АГЕНТНИХ СУТНОСТЕЙ

У цьому розділі наведено огляд основних концепцій обмежувальної рамки та просторових відносин. Далі в цьому розділі ми пояснюємо структуру нашої системи, її алгоритм, створення сцени та результати.

3.1 Обмежувальна рамка 3D об’єкта

Обмежувальна рамка — це невидимий кубоїд навколо кожного 3D-об’єкта в програмному забезпеченні 3D-анімації, який дає нам розміри по відношенню до осі X, Y і Z. Для об’єкта в тривимірному просторі обмежувальна рамка дає значення Xmin, Ymin, Zmin, Xmax, Ymax і Zmax, за якими ми можемо знайти довжину об’єкта за допомогою Xmin і Xmax, висоту об’єкта за допомогою Ymin і Ymax, ширину об’єкта за допомогою Zmin і Zmax. Знання цих розмірів допомагає нам запобігти накладанню, коли ми розміщуємо об’єкт. На рисунку 20 є таблиця об’єктів у тривимірному просторі. Щоб дізнатися розміри таблиці, ми повинні знайти boundingBox('Table'), який дає Xmin, Ymin, Zmin, Xmax, Ymax і Zmax. Загалом, o (Xmax - Xmin) дає довжину, o (Ymax - Ymin) дає висоту, а o (Zmax - Zmin) дає ширину об'єкта, де o означає об'єкт.

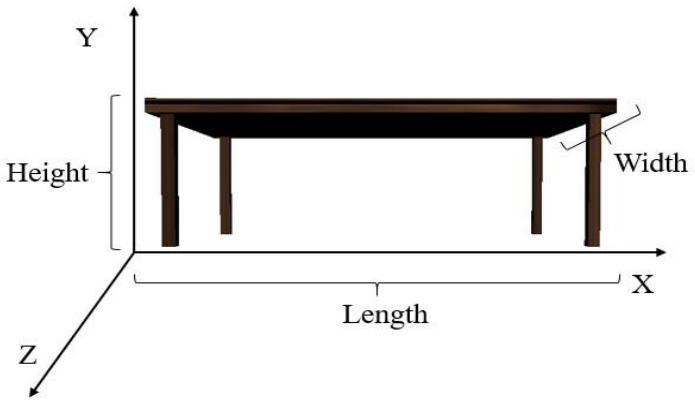


Рисунок 3.1 – Стіл в тривимірному просторі відносно осі X, Y, Z

Таблиця (Xmax - Xmin) = Довжина

Таблиця (Ymax - Ymin) = висота

Таблиця (Zmax - Zmin) = Ширина

3.2 Просторові відношення об’єктів

Просторові відносини - це слова, які описують, як об'єкти пов'язані один з одним у реальному світі. Вони відіграють вирішальну роль у системах генерації тексту в сцену, оскільки допомагають розміщувати об’єкти в 3D-сцені. Як правило, просторові відношення — це прийменники на зразок на, перед, під тощо. Таблиця 3.1 показує різні просторові відношення, які використовувалися в раніше запропонованих системах.

Таблиця 3.1: Просторові відносини, які підтримуються різними системами

Системи генерації 3D-сцени	Просторові відносини
Покласти	у, на, у, вгорі/внизу, спереду/назад, ліворуч/праворуч, поруч, ліворуч, праворуч, проти
WordsEye	на, під, за, поруч, у, перед, праворуч, ліворуч
Автоматичне генерування 3D-сцени на основі Майя	передній, прямий, сусідній, поблизу, північ, захід, північний схід, північний захід, прямий південь, прямо-північ, прямо-захід-від, обернений, прямо-зубий, лицем-прямий-південь
Інтерактивне навчання простору Знання для тексту в 3D-сцени	ліворуч, праворуч, вище, нижче, спереду, ззаду, зверху, поруч, біля, всередині, зовні

Ми розділили просторові відносини на дві категорії: розташування та напрямок. Різні місця навколо об’єкта можна визначити як: на, перед, позаду, всередині, ліворуч, праворуч, зверху та знизу. Напрямки включають північ, південь, схід, захід, північний схід, північний захід, південний схід і південний захід. Ця категоризація призначена для розташування об’єктів на певній

відстані від інших об’єктів. Коли між двома об’єктами існує просторове відношення розташування, принаймні одна грань об’єкта контактує з обличчям іншого об’єкта. Коли між двома об’єктами існує напрямний просторовий зв’язок, один об’єкт розміщується на певній відстані від іншого. Враховуючи просторові відношення між об’єктами, обчислюються їх значення обмежувальної рамки та переміщуються об’єкти в сцені.

Таблиця 3.2: Просторові відносини, які підтримує наша система

Розташування просторових відносин	Напрямок просторових відносин
на, під, перед, ліворуч, праворуч, всередині, ззаду, вище, нижче, поруч	схід, захід, північ, південь, південний схід, південний захід, північний схід, північний захід

Наприклад, у введеному тексті «Зліва від книги є ваза», ліворуч – це просторове відношення між книгою та вазою. Щоб розмістити вазу ліворуч від книги, ми враховуємо довжину предметів і переставляємо вазу.

Як інший приклад, у вхідному тексті: «На північний схід від книги є лампа», північний схід – це просторове відношення між книгою та лампою. На відміну від попереднього прикладу, тут ми повинні знати як довжину, так і ширину книги, щоб розмістити лампу по діагоналі навпроти неї.

Запропонована нами структура показана на рисунку. Вона розділена на дві підзадачі обробка тексту та позиціонування об’єктів. Під час обробки тексту система витягує корисну інформацію, як-от назви об’єктів та їх просторові відношення в тексті, використовуючи набір інструментів природної мови.

Під час позиціонування об’єктів система витягує з бази даних усі необхідні об’єкти, визначені на попередньому кроці, та імпортує їх у сцену. Після імпорту об’єктів система знаходить значення рамки для кожного об’єкта в сцені, щоб знайти висоту, ширину та довжину об’єктів.

Ці розміри та просторові відносини, які витягуються з тексту на першому кроці, використовуються для розміщення об’єкта в сцені.

Після розміщення об’єкта, якщо є якийсь рух, пов’язаний з об’єктом, він додається шляхом додавання ключових кадрів щодо часу. На відміну від попередніх моделей, у нас є окремі функції руху, які додають ключові кадри до об'єкта щодо часу, що вказує на те, що об'єкт знаходиться в русі.

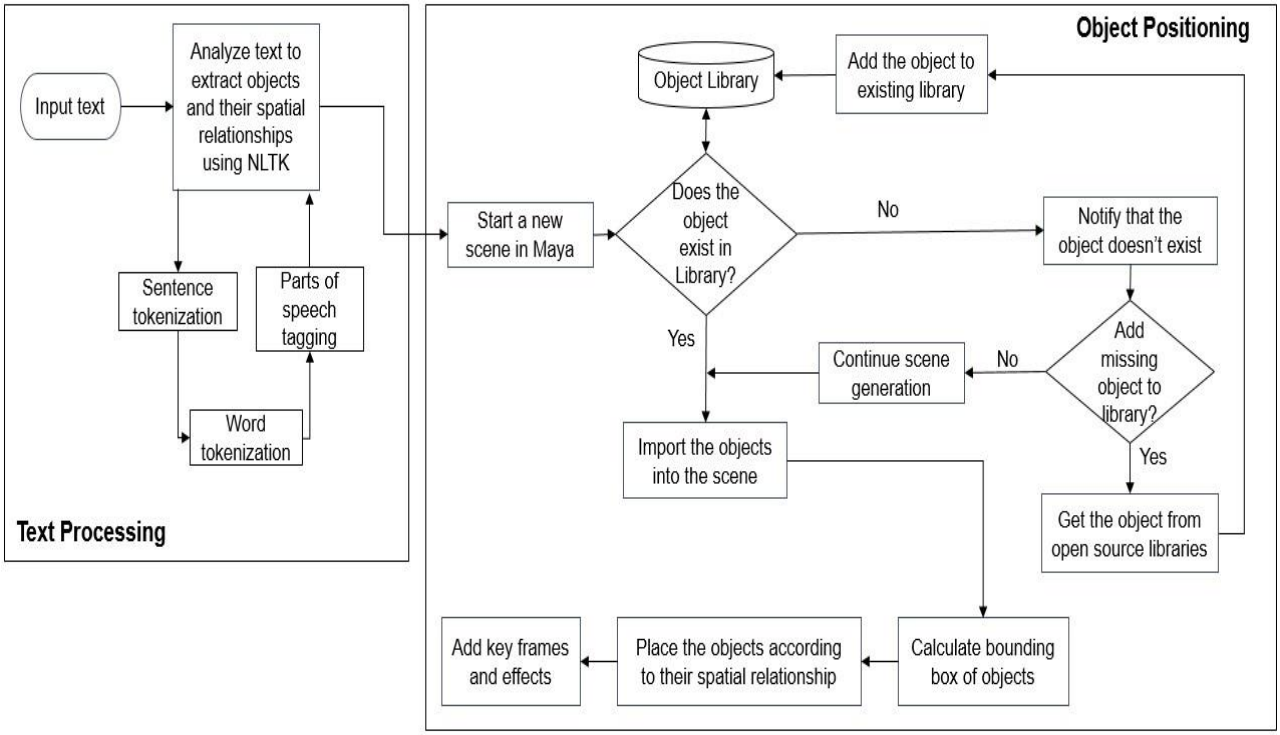


Рисунок 3.2: Структура системи

3.3 Розробка покрокового алгоритму та генерація схеми

Покроковий алгоритм запропонованої нами системи генерації 3D-сцени з тексту обговорюється нижче.

На кроці 5 обчислення тривимірної точки, до якої має рухатися об’єкт, повністю залежить від просторового відношення та розташування існуючого об’єкта. Детальний опис розміщення об’єктів пояснюється в наступному підрозділі.

Algorithm 1: Dynamic scene creation from text

Input: Text of a scene description
Output: Generated 3D scene

```
Extract object names and spatial relations from the input text
if object exists in the database then
    Import the objects into the scene
    Calculate the bounding box values of the objects to determine length,
     height and width of them.
    Find the 3D point p(x, y, z) at which the object has to be placed by
     calculating the distance that has to be there in between the objects
     (o1, o2) depending on the spatial relation R.
        p(x, y, z)=Δ(o1, o2, R)
    Move the object to ,the calculated 3D point.
        move(p, o1)
    Add motion and effects to the scene as described in the input text
else
    Notify the user that object doesn't exist in the database
    if the user wants to add the missing object to library then
        Get the object from open source libraries and add it to the
         existing library
    else
        Continue with scene generation
End
```

Алгоритм 3.3: Динамічне створення сцени з тексту

Як обговорювалося в рамках, наша генерація сцени поділяється на обробку тексту та позиціонування об’єктів. Спочатку ми обговорюємо структуру для обробки тексту, а потім фреймворк для позиціонування об’єктів.

Обробка тексту

При обробці тексту система витягує з введеного тексту імена об’єктів і просторові відносини. Зазвичай імена об’єктів є іменниками, і вони витягуються за допомогою Набору інструментів природної мови (NLTK). NLTK написана на Python і поширюється з ліцензією з відкритим кодом. Він має десятки алгоритмів, які обслуговують різні функціональні можливості і використовуються як основа багатьох дослідницьких проектів. NLTK підтримує великий і структурований набір текстів, який називається корпусом. Алгоритми навчаються на цих корпусах (множина корпусів). У цій дипломній роботі ми використовували модулі nltk.tokenize та nltk.pos_tag. Модуль Tokenize використовується для поділу даного тексту на речення та речення на

слова. Модуль тегів використовується для позначення слів відповідними частинами мови (POS). Нижче наведено приклад токенізації та POS-тегування для введеного тексту.

text= "Сфера на кубі і конус на сфері" sentence_tokenized = nltk.sentence_tokenize (текст) //розбиття введеного тексту на речення

 [«Сфера на кубі», «конус на кулі»]

word_tokenized=nltk.word_tokenize(sentence_tokenized) //розбиваємо кожне речення на слова

 [['Sphere', 'is', 'on', 'cube'], ['cone', 'is', 'on', 'sphere']] Parts_of_speech_tagging=nltk.pos_tag(word_tokenized) //позначення частин мови до кожного слова [(‘sphere’, ‘NN’), (‘is’, ‘VBN’), (‘on’, ‘IN’), (‘cube’, ‘NN’)]

[(‘cone’, ‘NN’), (‘is’, ‘VBN’), (‘on’, ‘IN’), (‘sphere’, ‘NN’)]

//where NN stands for noun, VBN stands for verb and

IN stands for preposition

Якщо будь-який з об’єктів недоступний, користувач отримає повідомлення про те, що конкретний об’єкт не знайдено. Просторові відносини заздалегідь визначені. Просторові відносини, які підтримувалися нашою системою, показані в таблиці 3.2. Наприклад, «На столі стоїть ваза» є вхідним текстом. Тут ваза і стіл – це два об’єкти, а «включено» – просторове відношення між ними. На рисунку наведено приклад сцени кімнати, яка показує, як об’єкти пов’язані просторовими відносинами з вхідним текстом «Перед столом стоїть стілець. На столі – книга. Ліворуч від книги – ваза. У вазі стоїть квітка, на північний схід від книги — лампа».

Якщо з об’єктом пов’язаний будь-який рух чи ефект, ми витягуємо тип руху чи ефекту, і він також застосовується до об’єкта. Ми підтримуємо набір ефектів, які можна включити в сцену, як-от сніг, хмари, небо та вогонь. Наприклад, «Будівля горить» є вхідним реченням; ми створили систему, щоб створити сцену, де горить будівля, замість того, щоб підпалювати будівлю. Тут

вогонь - це ефект, який стосується будівлі об'єкта. Вся ця інформація, яка витягується з тексту, використовується для розміщення об’єктів у сцені.

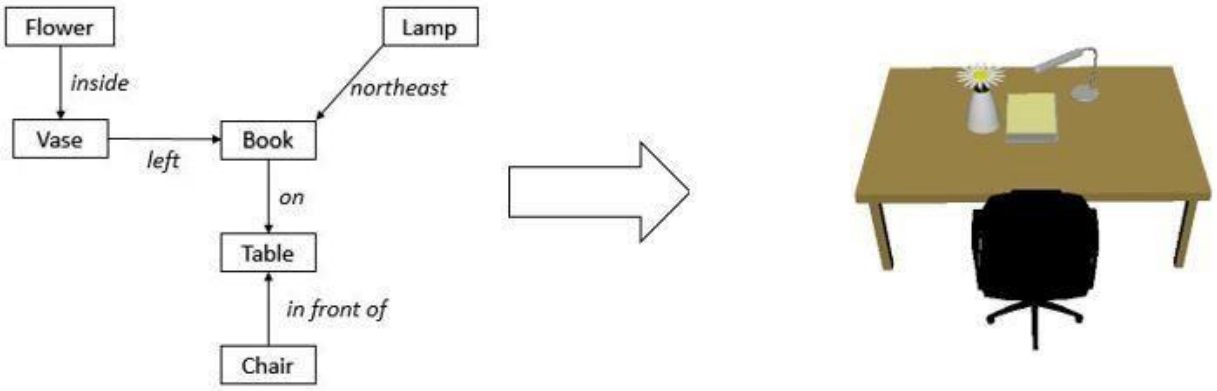


Рисунок 3.3: Відображення того, як об'єкти пов'язані з просторовими відносинами та відповідною кінцевою сценою

Позиціонування об'єкта

Наступне завдання після вилучення назв об’єктів і просторових відносин – створення 3D-сцени. Ми ведемо базу даних окремих об’єктів у локальній системі. Ці об’єкти можна змінювати, змінюючи їхні атрибути розміру та кольору перед створенням кінцевої сцени або після створення кінцевої сцени за допомогою різних інструментів, доступних у Маya. Ми також можемо додати нові об’єкти до цієї бази даних або замінити існуючі об’єкти новими. Багато художників в індустрії анімації створюють об’єкти для свого використання та роблять їх доступними в Інтернеті; ці об'єкти також можна використовувати в нашій системі. Деякі з 3D-моделей, які ми використовували, належать TurboSquid, Highend3D, cgtrader та clara.io. Усі ці компанії безкоштовно надають тисячі об’єктів, якими можуть користуватися мільйони користувачів у різних програмах. Ці компанії продають 3D-об’єкти на своїх умовах, де будь-який вид перерозподілу заборонений.

Як ми бачили в Алгоритмі 1, позиціонування об’єктів засноване на іменах об’єктів і просторових відносинах, витягнутих під час обробки тексту, система імпортує 3D-об’єкти в сцену. Спочатку всі об’єкти будуть розміщені в положеннях (0, 0, 0) у тривимірному просторі. Після їх імпорту система витягує розміри (Xmin, Ymin, Zmin, Xmax, Ymax і Zmax) об’єктів за допомогою функції обмежувальної рамки. Використовуючи ці значення, ми знаходимо висоту, ширину та глибину об’єктів. Залежно від просторового відношення між об’єктами ми обчислюємо один або два виміри між висотою, довжиною і шириною, оскільки не кожне просторове відношення вимагає всіх трьох. Наприклад, просторове відношення «включено» вимагає висоти об’єктів для їх розміщення. Просторове відношення «ліворуч» або «праворуч» вимагає довжини об’єктів для їх розміщення, а для просторового відношення «північно-східний» потрібні як довжина, так і глибина об’єктів, щоб розмістити їх у сцені.

Щоб знайти тривимірну точку (X, Y, Z), в якій потрібно розмістити об’єкт, ми повинні знати відстань між об’єктами. Щоб знайти відстань (Δ), ми повинні розглянути значення обмежувальної рамки об’єктів та їх просторове співвідношення. Наприклад, «Рослина на столі» є вхідним текстом для нашої системи. Тут рослина і стіл є об'єктами, а "на" - їх просторове відношення, яке з'єднує. Як ми вже обговорювали, система імпортує об’єкти в сцену в (0, 0, 0) і знаходить boundingBox(‘plant’) і boundingBox(‘table’) для розмірів. Розглянемо значення таблиці boundingBox як (-1unit, 0unit, -0.5unit, 1unit, 1unit, 0.5unit), висота розраховується шляхом додавання Ymin і Ymax. Отже, висота предмета 1 см. Це означає, що ми повинні розмістити рослину на 1 см над столом. У цьому сценарії тривимірною точкою є (0, 1, 0), до якої потрібно перемістити рослину. На рисунку показано різне розміщення об’єктів з урахуванням просторових відносин зліва, справа, зверху, знизу, спереду та ззаду.

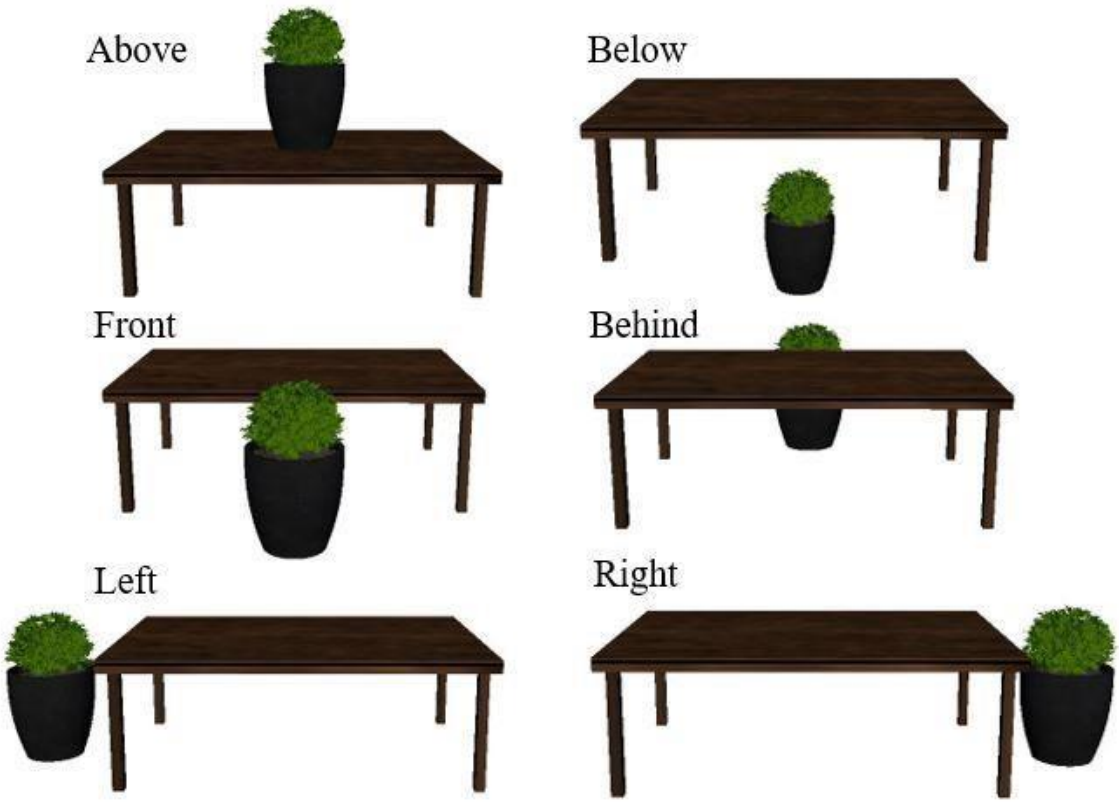


Рисунок 3.4: Просторові відносини щодо таблиці

Ми бачили приклад того, як ми можемо розмістити об’єкт, якщо його потрібно перемістити лише в одному напрямку. Що, якщо це просторове відношення на кшталт північного сходу чи північного заходу, південного сходу чи південного заходу, де ми повинні розглянути два виміри об’єкта, щоб розмістити його по діагоналі? Скажімо, "рослина знаходиться на північний схід до столу" - це вхідний текст. Тепер нам потрібно розмістити рослину по діагоналі до столу. Отже, ми повинні враховувати як довжину, так і ширину об’єкта. Оскільки напрямок північно-східний, ми додамо Xmax і Zmax як для таблиці, так і для рослини.

Досі ми бачили приклади, які мають єдині просторові відносини. Скажімо, «На столі лежить книга, праворуч від книги є рослина» — це вхідний текст. Тут книга і стіл пов’язані з просторовим відношенням «на», а рослина і книга – з просторовим відношенням «право». Як пояснювалося раніше, система знаходить висоту столу та кладе книгу на стіл. Щоб розмістити рослину

праворуч від книги, ми повинні враховувати тривимірну точку (X, Y, Z) книги, оскільки вона знаходиться на столі, і рослина також має бути на столі. Отже, ми переміщаємо рослину від початку (0, 0, 0) до розташування книги, а потім переміщаємо її праворуч від книги, знайшовши довжину книги та рослини.

Конкретне просторове відношення «на» можна використовувати двома різними способами, наприклад «килим на підлозі» і «годинник на стіні». У цих двох вхідних інструкціях просторове відношення «включено» повинно розташовувати об’єкти в різних положеннях системою, оскільки вона не повинна розташовувати годинник над стіною. Отже, ми розглянули це як два різних сценарії та впровадили їх у нашу систему.

Після позиціонування об'єкта наступним кроком є додавання анімації та ефектів до сцени. У Autodesk Maya існують різні прийоми для додавання руху об’єкту. Ми використали анімацію ключових кадрів, яка є однією з них. У цій техніці місцезнаходження об’єкта зберігається відносно часу. Замість того, щоб мати однаковий рух для кожного об’єкта, ми маємо різні рухи, які можна призначити будь-якому об’єкту в сцені, щоб об’єкти подібного типу мали подібний рух.

Розгляньте сцену, зображену на рисунку, де чоловік іде, а автомобіль рухається.

Зазвичай автомобіль рухається швидше, ніж людина, тому обом об’єктам не можна надати однаковий рух. Ми показали різницю в русі, приділивши більше часу чоловікові і менше часу машині.

Нарешті, додайте ефекти до сцени. Система перевіряє будь-які ефекти, пов’язані з об’єктами, і додає їх. Наприклад, введеним текстом є «будівля горить». Тут вогонь – це ефект, який потрібно додати до будівлі, тому система додає ефект, як показано на рисунку, для вхідного тексту «Будівля горить». Ми вважаємо, що це також унікальна умова, коли система додає об’єкту ефект вогню замість того, щоб помістити об’єкт на вогонь.



Рисунок 3.5: Згенерована 3D-сцена з ефектом вогню

3.4 Тестування розробленого продукту

Згенеровані 3D-сцени

Запропонована система була протестована на невеликому сховищі 3D-об’єктів. Кожна зі згенерованих 3D-сцен показана нижче з введеним текстом.

Таблиця 3.3: Список інструментів, які використовуються для реалізації запропонованої нами системи

Пункт	Деталі
OS	Windows
Мови	Python 2.7 & MEL
IDE	PyCharm
Обробка тексту	Набір інструментів природної мови (NLTK) Бібліотека Python
програмне забезпечення 3D моделі	Autodesk Maya 2018 Створено мною та деякими з загальнодоступних репозиторіїв, таких як TurboSquid, clara.io і cgtrader

Введіть текст 1:

«Перед столом стоїть стілець. На столі книжка. Зліва від книги є ваза. Усередині вази знаходиться квітка. На північний схід від книги є лампа»

Згенерована 3D-сцена 1:



Рисунок 3.6: Згенерована тривимірна сцена для введення тексту 1

Введіть текст 2:

«На дорозі є машина, а машина рухається. Ліворуч від дороги є пішохідна доріжка і стежкою йде чоловік. Будівля залишилася до пішохідної доріжки. Будівля виходить на пішохідну доріжку. Будівля горить».

Згенерована 3D-сцена 2:



Рисунок 3.7: Згенерована тривимірна сцена для введення тексту 2

Введіть текст 3

«Під час сходу сонця небо вкрите хмарами. Над горами літають птахи».

Згенерована 3D сцена 3:

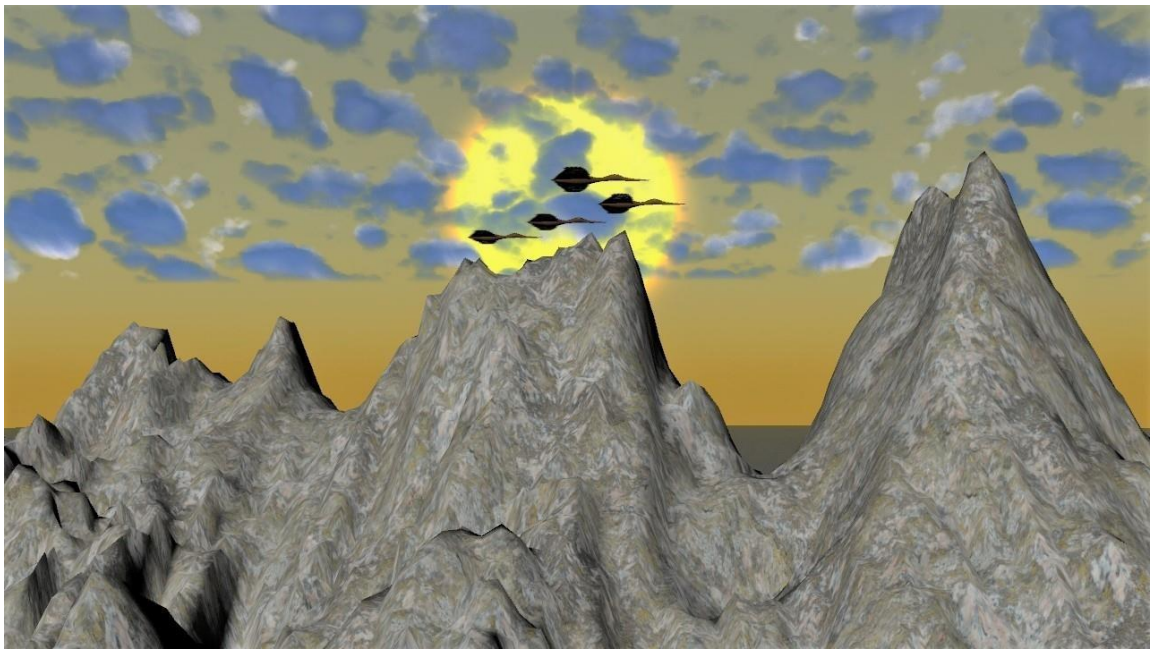


Рисунок 3.8: Згенерована тривимірна сцена для введення тексту 3

«У кімнаті є стіл, а перед столом — диван. Табурет праворуч від дивана і лампа на табуреті. На столі — ваза, а всередині — квітка. На північний схід від столу є рослина. На північний захід від столу є книжкова шафа. На підлозі лежить килим.

Годинник на передній_стіні, а телевізор під годинником. Фоторамка на лівій_стіні».

Згенерована 3D-сцена 4:

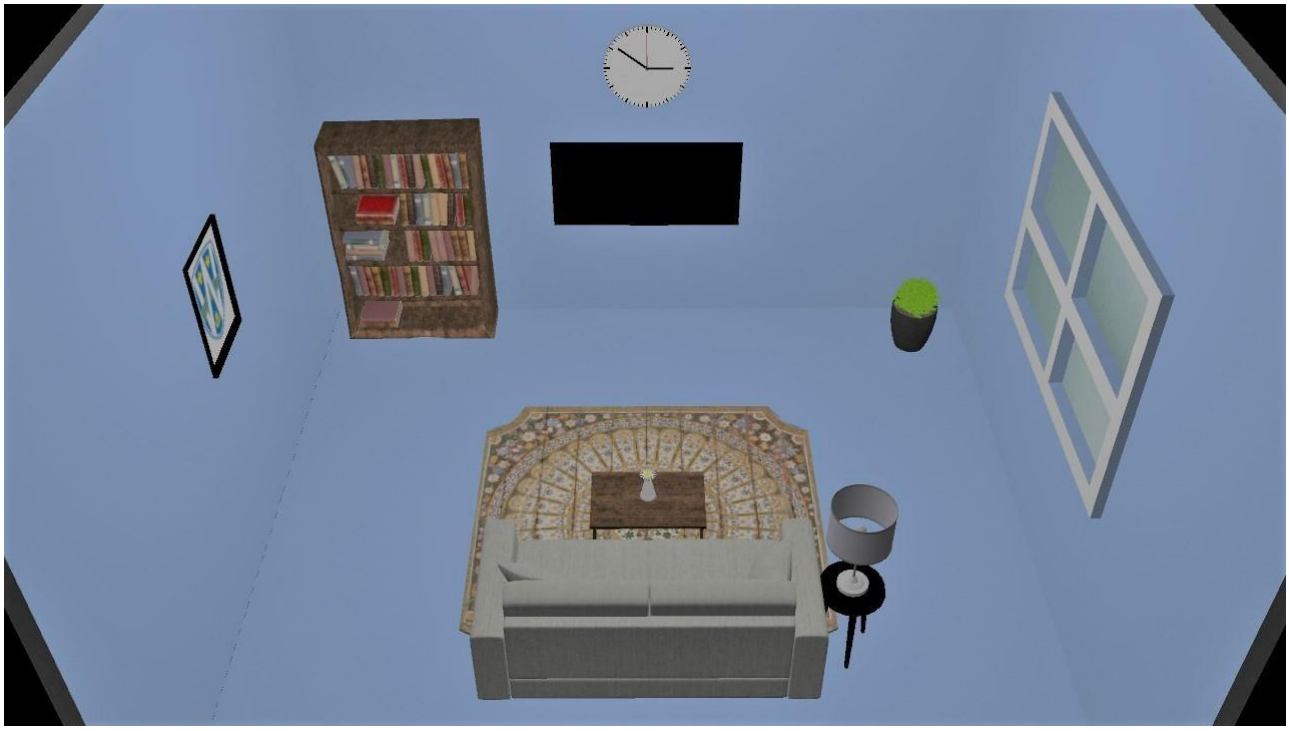


Рисунок 3.9: Згенерована тривимірна сцена для введення тексту 4

Введіть текст 5

«Будинок Одетти розташовано праворуч від Сансет-авеню, а Epi-хол — ліворуч від Сансет

проспект. Поруч із залом Epi знаходиться Lambton Tower, а Chrysler Hall знаходиться на північ від Lambton

Вежа. На північ від будівлі Одетти знаходиться будівля Толдо. Студенти ходять по

Sunset Avenue”

Згенерована 3D-сцена 5:

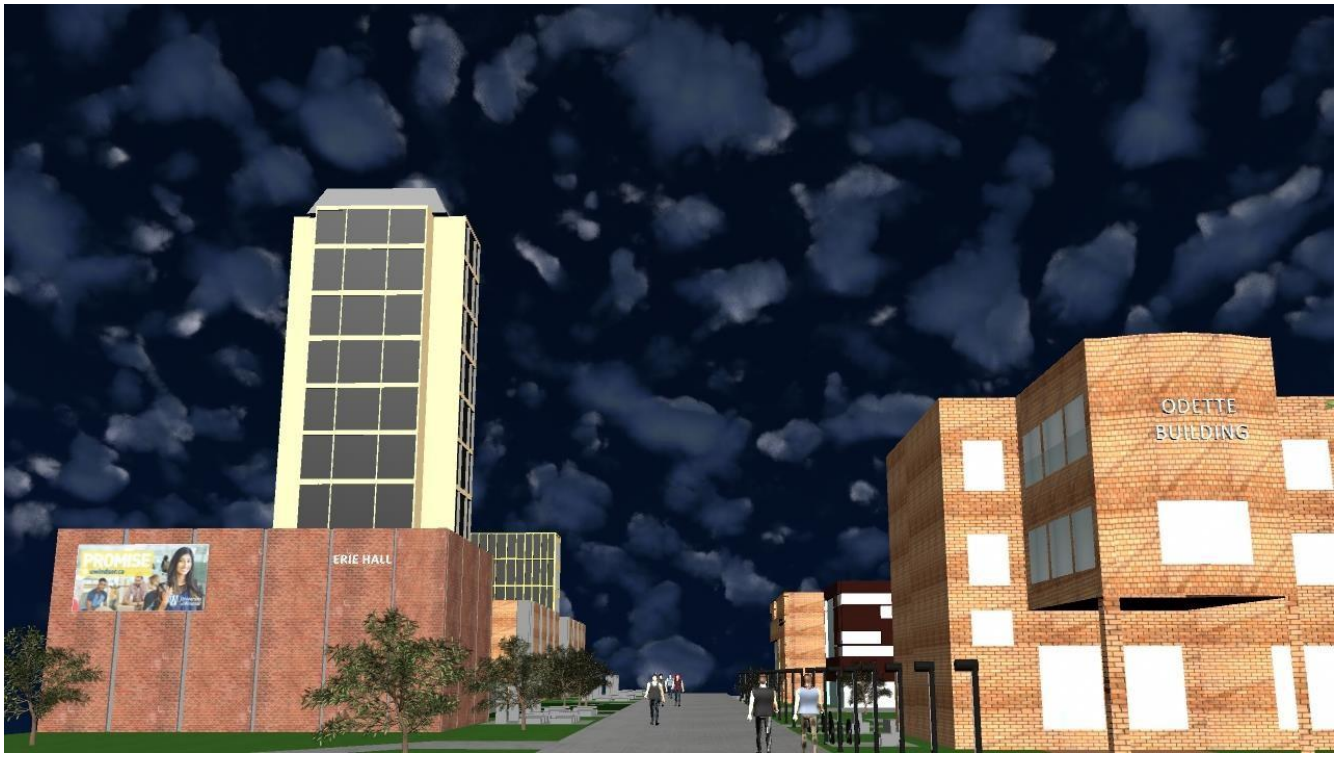


Рисунок 3.10: Згенерована тривимірна сцена для введення тексту 5

3.5 Представлення результатів генерації 3D сцен

Оскільки всі системи генерації тексту до сцени використовують попередньо розроблені 3D-об’єкти, ми не збираємося порівнювати на основі цієї функції, а порівнюватимемо на основі попередньо відформатованого введення, позиціонування об’єктів, анімації та ефектів.

Зауважте, що попередньо відформатований вхід є недоліком системи, оскільки користувач повинен розробити сценарій із введеного тексту, і цей вхід надається системі для створення 3D-сцени. У таблиці нижче Y означає Так, що вказує, що функція включена в систему, тоді як N означає Ні, що вказує, що функція не включена.

Таблиця 3.4: Порівняння між різними системами генерації 3D-сцени

	Попередньо відформатований вхід	Позиціонування об’єкта	Анімація	Ефекти
Покласти [1]	Y	Y	N	N
WordsEye [1]	N	Y	N	N
Автоматичне створення 3D-сцени на основі Maya	Y	Y	N	N
Нова рамка для Автоматична 3D сцена Будівництво	Y	Y	N	N
Вивчення просторових знань для тексту в 3D-сцені Покоління	N	Y	N	N
SCENESEER	N	Y	N	N
CARSIM	Y	Y	Y	N
Автоматизація створення 3D-анімація від Annoted Художній текст	Y	Y	Y	N
Створення анімації з Тексти природної мови і Семантичний аналіз руху Пошук і планування	N	N	Y	N
Автоматична генерація комп'ютерної анімації	Y	N	Y	N
Наша система	N	Y	Y	Y

Для більшості систем генерації 3D-сцени основною метою є позиціонування об’єкта за допомогою просторових відносин. Оскільки включення НЛП в індустрію анімації є досить складним і вимагає великих людських зусиль, лише кілька систем генерації 3D-сцени використовують його. Системи, які не використовують НЛП, записують введений текст у попередньо відформатований сценарій, призначений для певних систем. Жодна із систем генерації 3D-сцени не має ефектів, доданих до об’єктів. Ефекти, які підтримує

наша система, — це сніг, дощ, схід сонця, хмарне небо та додавання вогню до окремого об’єкта.

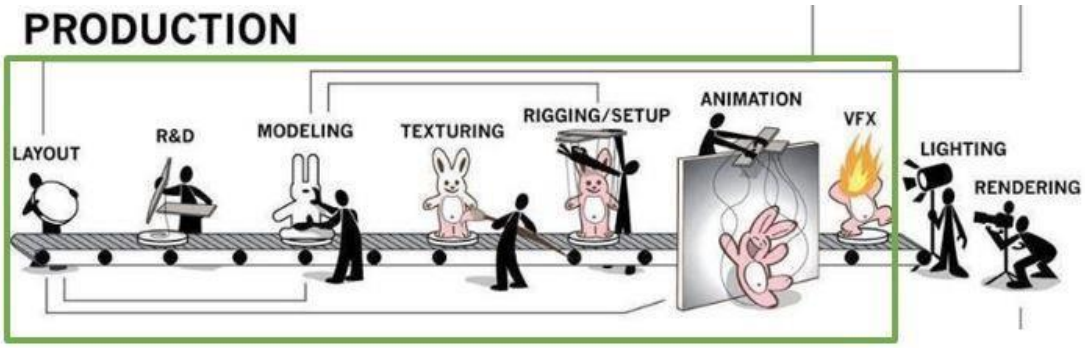


Рисунок 3.11: Завдання, досягнуті на етапі виробництва за допомогою пропонованого фреймворка

Лі та ін. також розробили подібний фреймворк, який генерує 3D-сцену в Maya. Але головна відмінність полягає в тому, що користувач повинен знати, як написати файл XML. Цей файл XML складається з імен об’єктів у вигляді тегів і пари атрибутів ім’я-значення для просторових відносин. Щоб створити 3D-сцену за допомогою цього підходу, користувач повинен перетворити звичайний текст у файл XML, як показано на рисунку, а потім надати його як вхідні дані. У нашій системі ми можемо безпосередньо вводити текст допомогою нашого фреймворка.

ВИСНОВКИ

В дипломній роботі бакалавра запропоновано фреймворк для генерації тексту для 3D сцен. Вирішено проблему генерації тексту в 3D-сцені, витягуючи з тексту назви об’єктів і просторові відносини, а потім розміщуючи об’єкти в сценах відповідно до їх просторових відносин. Після розміщення об’єктів система також додає анімацію та ефекти до об’єктів, щоб зробити сцену більш реалістичною та повною. Спроектовано обмежувальну рамку, просторові відносини, структуру, алгоритм і систему генерації сцени.

Продемонстровано, як обробляти введений текст шляхом вилучення назв об'єктів і просторових відносин.

В роботі було показано, як значення обмежувальної рамки використовуються для обчислення довжини, висоти та ширини, і використав ці значення для розміщення об’єктів у сцені з урахуванням просторових відносин.

Також в дипломній роботі продемонстровано, як можна додати анімацію та ефекти до об'єктів.

Розглянуто просторове відношення «на» для різних сценаріїв розміщення 3D об'єктів.

СПИСОК ПОСИЛАНЬ НА ДЖЕРЕЛА

[1] P.E. Agre, D. Chapman, Pengi: an implementation of a theory of activity, American Association for Artificial Intelligence (1987) 268 –272.

[2] D. Ballard, Animate vision, Artificial Intelligence 48 (1991) 57– 86.

[3] A. Baumberg, Hierarchical shape fitting using an iterated linear filter. British Machine Vision Conference, 1996, Edinburgh, Scotland, p. 313 –323.

[4] A. Baumberg, D.C. Hogg, Learning flexible models from image sequences, European Conference on Computer Vision, 1994, Stockholm, Sweden, p. 299–308.

[5] A. Baumberg, D.C. Hogg, An adaptive eigenshape model, British Machine Vision Conference, Birmingham, UK, 1995, p. 87–96.

[6] A. Baumberg, D.C. Hogg, Generating spatiotemporal models from training examples, Image and Vision Computing 14 (1996) 525 –532.

[7] D. Beymer, P. McLauchlan, B. Coifman, J. Malik, A real-time computer vision system for measuring traffic parameters, IEEE Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, 1997. 134 H. Buxton / Image and Vision Computing 21 (2003) 125–136

[8] M.J. Black, D.J. Fleet, Probabilistic detection and tracking of motion boundaries, International Journal of Computer Vision 38 (2000) 231–245.

[9] A. Blake, M. Isard, Active Contours, Springer, Berlin, 1998.

[10] A. Blake, M. Isard, D. Reynard, Learning to track the visual motion of contours, Artificial Intelligence 78 (1995) 179–212.

[11] A. Bobick, S. Intille, J. Davies, F. Baird, C. Pinhanez, L. Campbell, Y. Ivanov, A. Schutte, A. Wilson, The Kidsroom: a perceptually-based interactive and immersive story environment, Presence: Teleoperators and Virtual Environments 8 (1999) 367–1391.

[12] A. Bobick, A. Wilson, A state-based technique for the summarization and recognition of gesture. In IEEE International Conference on Computer Vision, Cambridge, MA, 1995, p. 382–388.

[13] M. Brand, N. Oliver, A. Pentland, Coupled hidden Markov models for complex action recognition, IEEE Conference on Computer Vision and Pattern Recognition, San Juan, Puerto Rico, 1997.

[14] H. Buxton, S. Gong, Visual surveillance in a dynamic and uncertain world, Artificial Intelligence 78 1995) 431–459.

[15] H. Buxton, R. Howarth, Watching behaviour: the role of context and learning. In International Conference on Image Processing, Lausanne, Switzerland, 1996, p. II: 797–800.

[16] H. Buxton, R. Howarth, Spatial and temporal reasoning in the generation of dynamic scene descriptions, in: P. Olivier, K.-P. Gapp (Eds.), Representation and Processing of Spatial Expressions, Lawrence Erlbaum Associates, Hillsade, NJ, 1997.

[17] E. Charniak, Bayesian networks without tears, AI Magazine 12 (4) (1991) 50–63.

[18] VIEWS consortium, The VIEWS project and wide-area surveillance, ESPRIT Workshop at ECCV, Genoa, Italy, 1992.

[19] T.F. Cootes, D. Cooper, C.J. Taylor, J. Graham, Active shape models—their training and application, Computer Vision and Image Understanding 61 (1995) 38–59.

[20] T.F. Cootes, C.J. Taylor, D. Cooper, J. Graham, Training models of shape from sets of examples, British Machine Vision Conference, Springer, Berlin, 1992, p. 9– 18.

[21] T. Dean, K. Kanazawa, Probabilistic temporal reasoning, American Association for Artificial Intelligence, St Paul, Minnesota, 1988, p. 524–528.

[22] G.J. Edwards, C.J. Taylor, T.F. Cootes, Learning to identify and track faces in image sequences, British Machine Vision Conference, Colchester, UK, 1997, p. 130– 139.

[23] J.O. Eklundh, P. Nordlund, T. Uhlin, Issues in active vision: attention and cue integration/selection, British Machine Vision Conference, Cambridge, UK, 1996, p. 1–12.

[24] J. Forbes, T. Huang, K. Kanazawa, S. Russell, The BATmobile: towards a Bayesian automated taxi, International Joint Conference on Artificial Intelligence, Montreal, Canada, 1995, p. 1878–1885.

[25] B.J. Frey and N. Jojic, Flexible models: a powerful alternative to exemplars and explicit models, CVPR Workshop on Models versus Exemplars, Kauai, Hawaii, 2001.

[26] N. Friedman, Learning belief networks in the presence of missing values and hidden variables, International Conference on Machine Learning, 1997, p. 125–133.

[27] A. Galata, N. Johnson, D.C. Hogg, Learning variable length Markov models of behaviour, Computer Vision and Image Understanding 81 (2001) 398–413.

[28] S. Gong, Visual behaviour: modelling hidden purposes in motion, International Conference on Neural and Stochastic Methods in Image and Signal Processing, San Diego, CA, 1992, p. 45–57.

[29] S. Gong, Visual observation as reactive learning, International Conference on Adaptive and Learning Systems, Orlando, FL, 1992, p. 175–187.

[30] S. Gong, H. Buxton, Bayesian nets for mapping contextual knowledge to computational constraints in motion segmentation and tracking, British Machine Vision Conference, Guildford, UK, 1993, p. 229– 238.

[31] D. Heckerman, A tutorial on learning with Bayesian networks, Learning in Graphical Models, NATO Science Series, 1998, p. 301 – 354.

[32] A. Hill, A. Thornham, C.J. Taylor, Model-based interpretation of 3D medical images, British Machine Vision Conference, Guildford, UK, 1993, p. 339–349.

[33] D.C. Hogg, Model-based vision: a program to see a walking person, Image and Vision Computing 1 (1983) 5–20.

[34] S. Hongeng, F. Bremond, R. Nevatia, Bayesian framework for video surveillance applications, International Conference on Pattern Recognition, Barcelona, Spain, 2000.

[35] S. Hongeng, R. Nevatia, Multi-agent event recognition, IEEE International Conference on Computer Vision, Vancouver, Canada, 2001.

[36] R. Howarth, Interpreting a dynamic and uncertain world: task-based control, Artificial Intelligence 100 (1998) 5–85.

[37] R. Howarth, H. Buxton, Selective attention in dynamic vision, International Joint Conference on Artificial Intelligence, Chambery, France, 1993, p. 1579–1585.

[38] R. Howarth, H. Buxton, Visual surveillance monitoring and watching, European Conference on Computer Vision, Cambridge, UK, 1996, p. 321–334.

[39] R. Howarth, H. Buxton, Attentional control for visual surveillance, in: S. Maybank, T. Tan (Eds.), ICCV Workshop on Visual Surveillance, IEEE Press, London, 1997.

[40] R. Howarth, H. Buxton, Conceptual descriptions from monitoring and watching image sequences, Image and Vision Computing 18 (2000) 105–135.

БІБЛІОГРАФІЧНА ДОВІДКА

Тема дипломної роботи бакалавра: “Розробка бізнес-моделей програмного забезпечення симуляцій на основі агентних сутностей”

Обсяг пояснювальної записки: 70 аркушів

Дата закінчення дипломної роботи 13 червня 2022р.

Підпис студента _____