

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО"**

Факультет електроніки

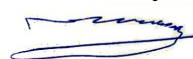
(повна назва інституту/факультету)

Акустичних та мультимедійних електронних систем

(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри



С.А. Найда

(ініціали, прізвище)

“14 ” грудня 2022 р.

Магістерська дисертація

зі спеціальності

171 Електроніка

(код і назва)

на тему:

" Алгоритми та підходи використання інструментів
персонажної анімації при створенні сцен кіно "

Виконав:

студент II курсу, групи ДВ-11мп

(шифр групи)

Войналович Олександр Олександрович

(прізвище, ім'я, по батькові)



(підпис)

Керівник

к.т.н., доцент Трапезон К.О.

(посада, вчене звання, науковий ступінь, прізвище, ініціали)



(підпис)

Рецензент

доц. кафедри ЕПС доц., к.т.н. Михайлов С.Р.

(посада, науковий ступінь, вчене звання, прізвище, ініціали)



(підпис)

Консультант

(назва розділу) (посада, вчене звання, науковий ступінь,

прізвище, ініціали)

(підпис)

Засвідчую, що у цій
магістерській дисертації немає
запозичень з праць інших авторів
без відповідних посилань.

Студент



(підпис)

Київ – 2022 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інститут (факультет) _____ Факультет електроніки
(повна назва)

Кафедра _____ акустичних та мультимедійних електронних систем
(повна назва)

Рівень вищої освіти – другий (магістерський) за освітньо-професійною програмою

Спеціальність (освітня програма) 171 Електроніка

(Електронні системи мультимедіа та засоби Інтернету речей)

(код і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри



Сергій НАЙДА

(підпис)

(ініціали, прізвище)

«14» грудня 2022 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Войналовичу Олександрі Олександровичу

(прізвище, ім'я, по батькові)

1. Тема дисертації «Алгоритми та підходи використання інструментів персонажної анімації при створенні сцен кіно».

Науковий керівник дисертації к.т.н., доц. Трапезон Кирило Олександрович.
(науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

затверджені наказом по університету від « 08 » листопада 2022 р. №4092-с

2. Строк подання студентом дисертації 01.12.2022 р.

3. Об'єкт дослідження: програмні інструменти створення персонажної анімації об'єктів

4. Предмет дослідження (Вхідні дані – для магістерської дисертації за освітньо-професійною програмою): шаблонні чи тестові об'єкти, які використовуються для перевірки методів і алгоритмів персонажної анімації. Спеціальні прийоми з використання сегментної анімації.

5. Перелік завдань, які потрібно розробити: проаналізувати основні принципи та підходи щодо створення анімації персонажів; дослідити методи та алгоритми персонажної анімації; проаналізувати особливості налаштування сегментної анімації об'єктів.

6. Перелік графічного (ілюстративного) матеріалу: 15 слайдів презентації, основними назвами плакатів якої є сформульовані завдання, мета, постановка проблеми, особливості проведення процедур на основі інструментів та правил в спеціалізованій програмі.

7. Орієнтовний перелік публікацій: 1 публікація у фаховому виданні категорії Б. Назва статті: “Особливості використання технології photo wake-up для систем машинного зору”

8. Дата видачі завдання 1. 09. 2022 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Проаналізувати принципи та підходи до створення анімації персонажів. Визначення особливостей ретаргейтінгу та підходів до застосування анімованих текстур.	1.09.2022 – 2.10.2022	Виконано
2	Розглянути методи та алгоритми персонажної анімації. Проаналізувати підходи до створення сегментної анімації.	1.09.2020 – 26.10.2022	Виконано
3	Провести перевірку щодо можливостей використання програмних інструментів персонажної анімації об'єктів при створенні сцен кіно.	27.10.2022 – 15.11.2022	Виконано
4	Підготовка матеріалів до друку та оформлення пояснювальної записки	16.11.2022 – 05.12.2022	Виконано
5	Підготовка та оформлення презентації для доповіді	06.12.2022 – 12.12.2022	Виконано

Студент



(підпис)

Олександр ВОЙНАЛОВИЧ

(ініціали, прізвище)

Науковий керівник



(підпис)

Кирило ТРАПЕЗОН

(ініціали, прізвище)

УДК 004.915

РЕФЕРАТ

Войналович О.О. Алгоритми та підходи використання інструментів персонажної анімації при створенні сцен кіно: магістерська дис. : 171 Електроніка / Войналович Олександр Олександрович. - Київ , 2022. - 80с.

Магістерська дисертація: 80 с. , 39 рис. , 40 джерел.

Ключові слова: візуалізація, візуалізація даних, моделювання.

Актуальність дослідження. В наш час 3D всесвіт надзвичайно популярний, фільми, серіали, мультфільми, реклама, різний відеоконтент, та відеоігри майже неможливо уявити без анімованої 3D графіки, проте 3D об'єкти дещо відрізняються від реального світу, адже аби умовний персонаж мультфільму почав рухатись, необхідно аби художник-аніматор проробив велику роботу, від створення скелету персонажа (ригінгу), до анімації, наприклад за допомогою ключових кадрів. Складність цієї роботи спонукає до автоматизації процесів. Саме тому зараз стає популярним створення вже готових скелетів та анімації для використання в майбутніх проєктах.

Мета дослідження : полягає в дослідженні створення анімації, ригінгу персонажа, та створення навчального продукту на основі отриманих знань.

Завдання для досягнення мети : дослідити усі різновиди створення анімації, обрання методу який найбільш відповідає поставленим задачам, на основі обраного методу створити вихідний продукт.

Об'єкт дослідження : анімація персонажів, програмне забезпечення Blender, та Unity.

Предмет дослідження : алгоритм ригінгу, взаємодія з мешем, алгоритми анімації.

Методи дослідження : робота в програмному забезпеченні Blender, застосування методів та алгоритмів ригінгу та анімації.

Наукова новизна отриманих результатів: розроблено новий продукт під необхідну задачу, створено скелет для подальшої анімації.

Практичне значення одержаних результатів : отриманий результат може бути застосований в майбутніх проєктах, від відеоматеріалів до комп'ютерних ігор.

ABSTRACT

Master's thesis: 80 p. , 39 figs. , 40 sources.

Keywords: visualization, data visualization, modeling.

Relevance of research. Nowadays, the 3D universe is extremely popular, movies, series, cartoons, advertisements, various video content, and video games are almost impossible to imagine without animated 3D graphics, but 3D objects are somewhat different from the real world, because in order for a conventional cartoon character to start moving, it is necessary to the animator did a lot of work, from creating the character's skeleton (rigging) to animating, for example, using keyframes. The complexity of this work encourages the automation of processes. That is why it is now becoming popular to create ready-made skeletons and animations for use in future projects.

The purpose of the research: is to investigate the creation of animation, character rigging, and the creation of an educational product based on the knowledge obtained.

The task to achieve the goal: to explore all types of animation creation, to choose the method that best meets the set tasks, to create the original product based on the chosen method.

Research object: character animation, Blender software, and Unity.

Research subject: rigging algorithm, mesh interaction, animation algorithms.

Research methods: work in Blender software, application of rigging and animation methods and algorithms.

The scientific novelty of the obtained results: a new product was developed for an urgent task, a skeleton was created for further animation.

Practical significance of the obtained results: the obtained result can be applied in future projects, from video materials to computer games.

ЗМІСТ

1 Персонажна анімація та її застосування в кіно.....	10
1.1 Поняття моделі персонажу.....	10
1.2 Принципи створення анімації персонажів.....	11
1.3 Створення персонажів в програмах 3D-моделювання.....	13
1.4 Спеціальні прийоми в персонажній анімації.....	17
1.4.1 Ретаргейтинг.....	20
1.4.2 Інверсна кінематика.....	21
1.4.3 Анімовані текстури.....	23
Висновки до 1 розділу.....	24
2 Методи та особливості персонажної анімації.....	25
2.1 Особливості врахування міміки персонажу.....	25
2.2 Методи скелетної анімації.....	26
2.2.1 Сегментна анімація.....	27
2.2.2 Анімація без вагів.....	28
2.2.3 Еталонні форми в персонажній анімації.....	29
2.3 Спеціальні алгоритми персонажної анімації.....	30
2.3.1 Pose Space Deformation.....	30
2.3.2 Алгоритм П. Слоана.....	32
2.3.3 Алгоритм, заснований на скануванні.....	32
2.3.4 Метод П. Края.....	33
Висновки до 2 розділу.....	33
3 Ригінг персонажа, аналіз та прогнозування анімації.....	34
3.1 Ригінг за допомогою інверсної кінематики.....	34
3.2 Інверсна кінематика.....	39
3.2.1 Якщо нема рішень.....	39

3.2.2 Одне рішення.....	40
3.2.3 Багато рішень.....	40
3.3 Інверсна кінематика у 2D.....	41
3.4 Інверсна кінематика у 3D.....	43
3.4.1 Вісь шарніра перпендикулярна кістці.....	43
3.4.2 Вісь шарніра спрямована вздовж кістки.....	45
3.5 Зручне положення.....	46
3.6 Анімація.....	47
3.6.1 Основні прийоми.....	47
3.6.2 Генерація руху.....	49
3.6.2.1 Генерація параметрів.....	50
3.6.2.2 Методи на основі обмежень.....	53
3.6.2.3 Контроль руху.....	55
3.6.3 Посилення руху.....	58
3.6.3.1 Шум.....	58
3.6.3.2 Рівень деталізації.....	59
Висновки до 3 розділу.....	60
4 Розроблення стартап-проєкту.....	61
4.1 Опис проєкту, призначення.....	61
4.1.1 Опис існуючих рішень та їх обмеження. Конкуренти...	61
4.1.2 Розв'язок проблеми.....	62
4.2 Ринок продукту.....	62
4.2.1 Бізнес модель “Канвас”. Можливі прибутки.....	62
4.3 Маркетинг. Сильні та слабкі сторони продукту.....	63
4.4 Прогнозування розвитку проєкту.....	64
4.5 Ціннісна пропозиція.....	65
4.6 План реалізації стартапу.....	65

Висновки до 4 розділу.....	65
Висновки.....	67
Перелік посилань.....	68

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

МГК – метод головних компонентів.

Меш – у 3D-комп'ютерній графіці та твердотільному моделюванні полігональний меш – це набір вершин, ребер і граней, що визначає форму багатогранного об'єкта. Грані зазвичай складаються з трикутників (трикутної сітки), чотирикутників (квад) або інших простих опуклих багатокутників (n-кутників), оскільки це спрощує візуалізацію, але також можуть складатися з увігнутих багатокутників або навіть багатокутників з отворами.

Ригінг – це техніка в комп'ютерній анімації, у якій персонаж (або інший артикульований об'єкт) представлений у двох частинах: поверхнева репрезентація, що використовується для малювання персонажа (яка називається мешем або шкірою), та ієрархічний набір взаємопов'язаних частин (так звані кістки, разом утворюючи скелет або установку), віртуальну арматуру, яка використовується для анімації (поза та ключовий кадр) сітки.

FBX – Filmbox format.

LBS – Linear blend skinning.

PSD – Pose Space Deformation.

РБФ – радіальні базисних функції.

SE – Shape by Example.

K.L.A.W. – Keyframe-Less Animation of Walking

ВСТУП

В наш час комп'ютерна графіка та 3D стала невід'ємною частиною усього відео-контенту, який споживає людство. Починаючи відеорекламою побутової техніки і закінчуючи високо бюджетним блокбастером чи дорогою відеоігрою. Ринок 3D розвивається з кожним днем, все більше людей залучаються в цю сферу та стають спеціалістами. Важливою складовою анімації є створення скелета для персонажа, що буде анімовуватись. Без наявного скелета, персонаж банально не зможе рухатись. Тому саме цей напрямок під назвою ригінг, стає дедалі популярним напрямком для навчання. Адже фахівці в цій сфері дуже цінуються в усіх компаніях з виготовлення відео-контенту.

З появою віртуальної та доповненої реальності ринок анімації ще більше розширюється. З'являється Meta-всесвіт, в якому кожна людина, має власного аватара, якого потрібно анімувати, аби він виглядав реалістично, неначе живий.

Ігрові студії що виготовляють відеоігри також потребують в своєму продукті анімованих персонажів, адже в багатьох відеоіграх саме ігрові персонажі є головними дієвими особами. Ігрові персонажі немов «ляльки», які рухаються завдяки рухам «ляльководи». В 3D індустрії роль «ляльководи» розділена між декількома спеціалістами, а саме: майстер ригу, аніматор та продюсер.

Вже зараз можна помітити наскільки велика кількість людей цікавляться саме напрямком анімації, та ригінгу персонажів. Цей напрямок дуже перспективний, адже покращувати свої навички можна до нескінченності. З кожним днем плавність анімації буде ставати дедалі кращою.

1 ПЕРСОНАЖНА АНІМАЦІЯ ТА ЇЇ ЗАСТОСУВАННЯ В КІНО

1.1 Поняття моделі персонажу

Одним з основних етапів при розробці 3D-відеоігор та мультиплікаційних фільмів є власне анімація тривимірних моделей. Тривимірна модель зазвичай є полігональною тривимірною поверхнею, анімація якої в кінцевому етапі зводиться до трансформацій (переміщень) вершин, що утворюють цю поверхню. Анімація моделей може задаватися різними способами, включаючи методи Motion Capture, моделювання фізичної поведінки об'єктів, анатомії реальної людини тощо. Найбільшого поширення, однак, набули методи скелетної анімації.

Під моделлю персонажа розуміється тривимірна полігональна сітка (polygon mesh) або оболонка, яка являє собою сукупність вершин, ребер і граней, визначальних форму полігонального об'єкта, що відображається. З вершиною пов'язана інформація про її координатах, векторах нормалі і координатах текстур, що накладаються – растрових зображень, що використовуються для задання кольору, ілюзії рельєфу та деталізації моделі.

Термін «базова форма» використовується для опису недеформованої моделі персонажа. Базове становище вершини – становище вершини у основі моделі. Під ключовою формою мається на увазі форма, яка надається моделі у ключовому кадрі анімації. Під анімацією розуміється трансформація оболонки моделі з початкового положення на кінцеве (з однієї ключової форми в іншу). Форма моделі в кожному проміжному кадрі інтерполюється за ключовими формами. При цьому може застосовуватися дискретна, лінійна або нелінійна інтерполяція (наприклад, криві Безьє).

1.2 Принципи створення анімації персонажів

Існують методи анімації персонажів або об'єктів, що дозволяють зробити їх рухи точними і досконало природними. Можливість застосування одного або декількох подібних методів слід розглядати перед тим, як приступати до анімації по кадрах або методу заповнення проміжків рухом або формою.

Один з найважливіших методів полягає в тому, щоб показати зовнішній вплив оточення на об'єкт, включаючи силу тяжкості, інші об'єкти і т.д. Так, якщо бита ударяє по м'ячу, він трохи сплющується там і в той момент, де і коли він стикається з битою.

Для анімації виразів обличчя корисний наступний метод. Якщо, наприклад, анімація персонажів, що розмовляють, створюється з синхронізацією руху губ, в такому випадку доречно показати дію сили тяжіння і розтягування осіб персонажів.

Коли ж м'яч летить повітрям, він трохи розтягується, як показано на (рис. 1.1). У анімації персонажів такий ефект нерідко перебільшується. А завдяки розмиванню досягається ефект руху об'єкта (зокрема за допомогою фільтрів розмитості руху, які часто використовуються у Photoshop і Fireworks). Чим сильніше об'єкт розмивається, тим більш прискореним мається на увазі його рух.



Рисунок 1.1 – Політ м'яча

При створенні анімації слід спробувати уявити собі рух, який потрібно показати. Основну увагу слід, звичайно, приділяти основним рухам або діям в анімації, щоб відсунути на другий план другорядні рухи. Адже якщо основна дія (так звана основна анімація) явно виражена, анімація від цього лише виграє.

Коли анімація екземпляра здійснюється методом заповнення проміжків, він рухається прямою лінією з початкової точки (у першому ключовому кадрі) в кінцеву точку (у другому ключовому кадрі). Але якщо рух проходить по дузі (рис. 1.2), у цьому випадку можна досягти більшого реалізму, ніж при русі по прямій лінії.

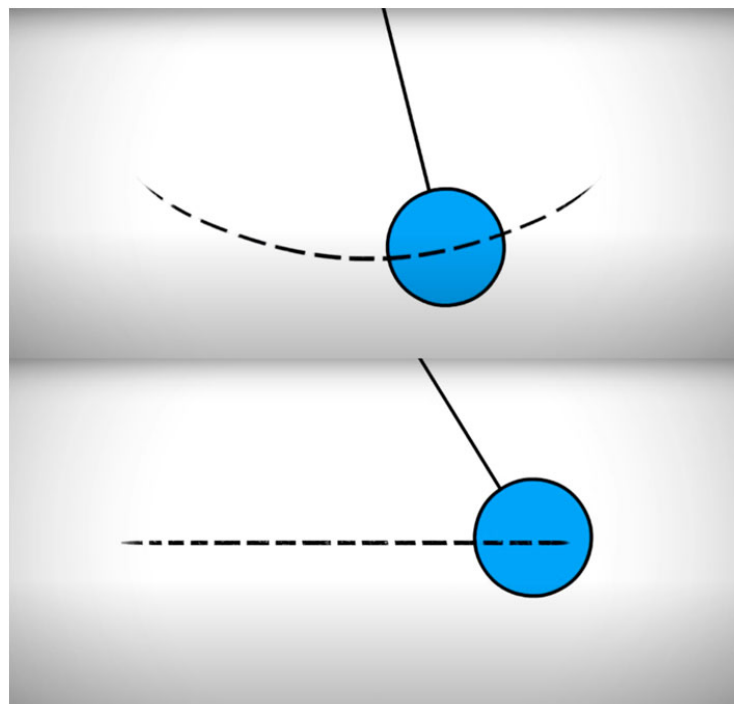


Рисунок 1.2 – Рух по дузі

Наприкінці руху іноді може знадобитися так званий залишковий рух. Під залишковим рухом розуміються дрібні рухи, що здійснюються об'єктом після його зупинки для переходу в стан повного спокою. Інерційність руху це ще один спосіб досягнення більш реалістичного і правдоподібного руху шляхом поступового уповільнення або прискорення об'єкта в проміжку анімації, що заповнюється [34]. Об'єкти можна прискорювати та уповільнювати, використовуючи властивість

інерційності руху в інспекторі Property. А для ще більшого художнього ефекту інерційність руху можна зімітувати вручну, використовуючи цілий ряд проміжків, що заповнюються рухом, для поступово уповільнення рухомого персонажа (рис. 1.3).

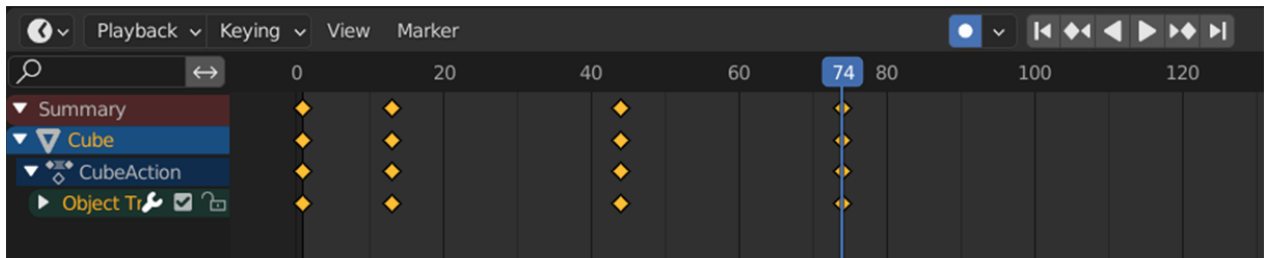


Рисунок 1.3 – Часова доріжка з ключовими кадрами

Ще одним важливим принципом анімації персонажів, що змінюється до того, як відбудеться головна подія, є так зване попередження. Це означає, що перш ніж станеться певна подія, здійснюється анімація персонажа, яка дає натяк на те, що має статися, щоб глядачі могли про це здогадатися. Якщо, наприклад, здійснюється анімація бити, що б'є по м'ячу, то можна показати, як персонаж змахує битою, перш ніж ударити по м'ячу, а якщо персонаж збирається щось сказати, показати вираз його обличчя, що вказує на таку зміну дії. Завдяки запобіганню підвищується реалістичний характер анімації.

Існують й інші принципи створення анімації. До них, зокрема, відноситься розкадрування у вигляді невеликих нарисів, попереднє узгодження анімації в часі, композиція кадру по відношенню до персонажа, зйомка персонажа крупним, середнім або дальнім планом, вибір кута зйомки і порядку переміщення персонажа в кадр. Все це допомагає донести ідею анімаційного проекту до глядача.

1.3 Створення персонажів в програмах 3D-моделювання

Для правильної анімації в Unity персонажі повинні бути попередньо оснащені у програмі 3D-моделювання, як-от 3ds Max, Maya або Blender. Оснащення означає, що художником-дизайнером до моделі доданий внутрішній скелет або людиноподібна структура кісток та кожна кістка (наприклад, кістка руки, кістка ноги та інші) спеціальним чином привішена (примагнічена) до вершин меша. Призначення кісток [35] – завдання того, яким чином вершини меша персонажа повинні рухатися і деформуватися, продовжуючи відповідати скелету. Це спрощує анімацію персонажів, тому що, замість анімувати всі вершини моделі персонажа з високою деталізацією, аніматор може анімувати скелет (кістки) і отримувати деформації персонажа автоматично відповідно до скелета (рис.1.4).

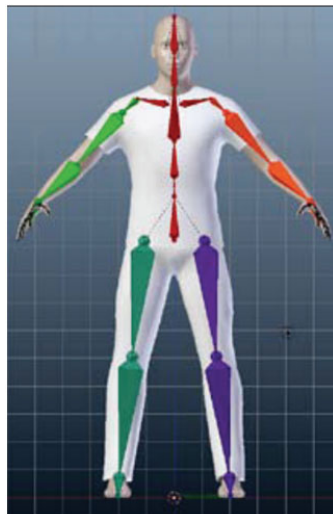


Рисунок 1.4 – Оснащений персонаж

Імпорт оснащених персонажів у Unity є простим перетягуванням. Треба перетягнути файл персонажа FBX з провідника Windows (або Mac Finder) в панель проекту, і Unity завантажить меш та його оснащення автоматично. При цьому меш повинен відобразитися на панелі попереднього перегляду. Однак можна

імпортувати лише оснащення та дані анімації без меша. Для прикладу, використано створений в MakeHuman оснащений персонаж (рис. 1.5).

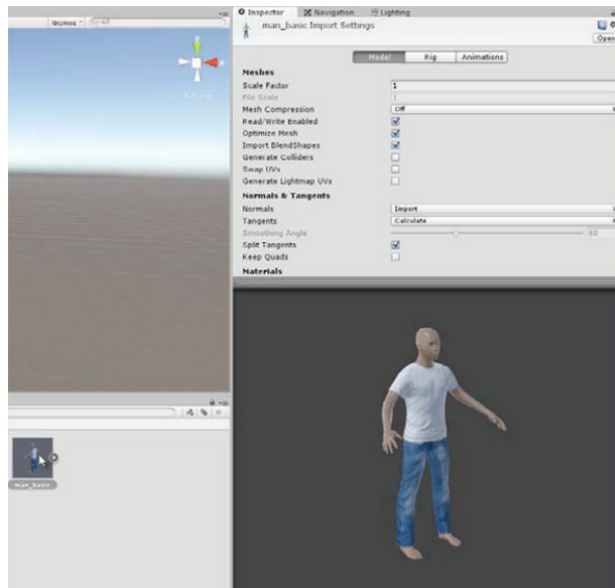


Рисунок 1.5 – Імпорт оснащеного персонажа до Unity

Насамперед після імпорту оснащеного персонажа треба встановити його масштаб, оскільки згодом зміна коефіцієнта масштабування персонажа може призвести до пошкодження або невідповідності іншим параметрам меша [36]. Щоб зробити це, треба перетягнути меш персонажа в сцену і перевірити пропорційність його розмірів по відношенню до решти сцени. В ідеалі бажано дотримуватись реальних одиниць виміру; тобто використовувати метр як одиниця виміру, так як основна одиниця виміру Unity відповідає 1 м. Потрібно змінити масштабний коефіцієнт так, якщо це необхідно, щоб меш мав реальні для сцени розміри (рис. 1.6). Це буде важливо для фізичних розрахунків, і для розрахунків освітлення.

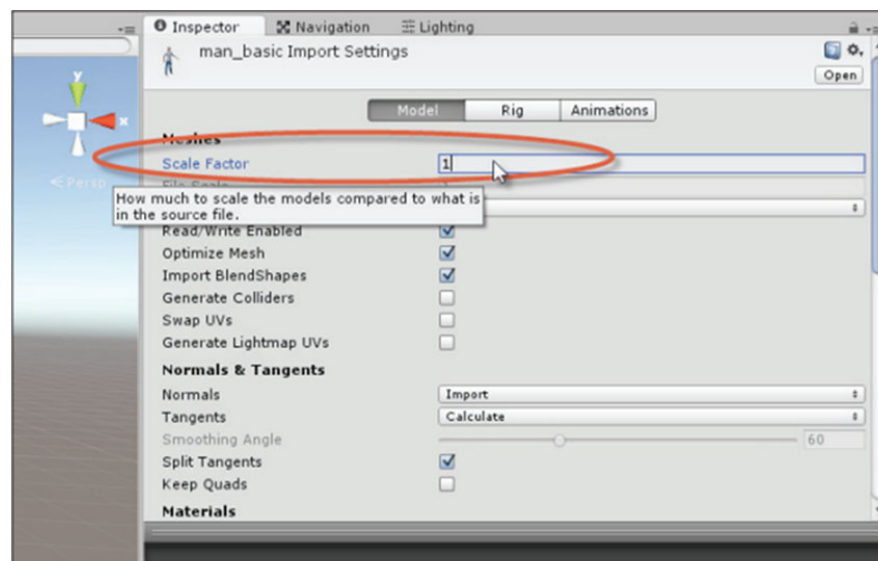


Рисунок 1.6 – Налаштування масштабу для персонажа

Потім треба перейти на вкладку оснащення Rig в інспекторі об'єктів і в якості типу анімації в списку вибрати тип анімації Humanoid, якщо це ще не зроблено. За замовчанням цим значенням, ймовірно, буде тип анімації Generic, якщо меш містить оснащення, і тип анімації None, якщо меш не містить оснащення. Тип анімації Generic використовується для оснащення мешей, що не належать до гуманоїдних персонажів, наприклад для кранів, триголових чудовиськ, змій, дерев та ін. Тип анімації Humanoid є рекомендованим вибором для людиноподібних персонажів, тому що дає доступ до всіх функцій анімації в системі Mecanim. Після вибору типу анімації Humanoid натиснути кнопку Apply (рис. 1.7).

При натисканні на кнопку Apply для підтвердження вибору оснастки з типом анімації Humanoid галочка або хрест з'явиться поряд з кнопкою Configure в об'єкті інспектора. Це індикатор того, чи успішно система Mecanim ідентифікувала оснащення гуманоїдного персонажа всередині меша.

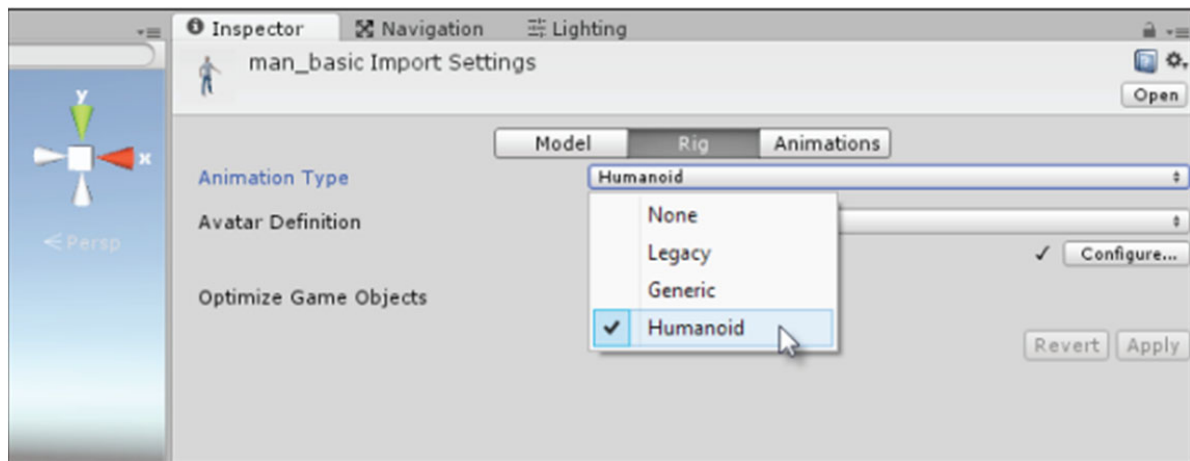


Рисунок 1.7 – Опція Humanoid з меню Animation Type

1.4 Спеціальні прийоми в персонажній анімації

Вибір типу анімації оснастки Humanoid для імпортованого меша персонажа означає, що Unity автоматично намагатиметься налаштувати меш спеціальним чином під вимоги системи Mecanim. Додаток Unity робить це, використовуючи структуру даних, що називається аватар (Avatar) (рис.1.8). Процес автоконфігурування може завершитися успішно, або невдачею. Якщо процес автоконфігурування завершився невдачею, потрібно буде налаштувати аватар меша вручну. Для цього треба натиснути налаштування Configure, щоб відкрити редактор аватара. Додаток Unity при цьому перейде в інший режим і, можливо, запропонує зберегти поточну сцену. Навіть якщо процес автоматичної генерації аватара завершився успішно, за необхідності, можна вручну підлаштувати чи змінити аватар персонажа. Тому не варто пропускати цю секцію, навіть якщо з мешом персонажа не виникло жодних питань



Рисунок 1.8 – Редактор “аватар” для налаштування персонажа

Редактор аватара охоплює відразу три інтерфейсні панелі: панель ієрархії, панель сцени та панель інспектора об'єктів (як це показано на рис 1.8). Разом вони містять усі дані редактора аватара. У панелі ієрархії перераховані всі кістки меша та відображена їхня структура. Панель сцени містить зображення меша персонажа разом із зображеннями кісток. Панель інспектора за замовчуванням об'єктів відображає схему людського тіла. Завдання редактора аватара полягає в тому, щоб забезпечити відповідність кісток меша та їх ієрархії карті людського тіла в панелі інспектора об'єктів. Виділені зеленим кольором кістки на панелі були автоматично правильно зіставлені Unity з аватаром. Не виділені кольором кістки були проігноровані. Кістки, виділені червоним, викликали проблему: або вони неправильно пов'язані з картою тіла, або вони не зорієнтовані і не вирівняні належним чином.

Щоб перевірити відповідність кісток меша карті, треба клацнути на будь-який з круглих значків на карті тіла. При цьому відбудеться вибір об'єкта відповідної кістки у панелях сцени та ієрархії. За допомогою клацання по будь-якому кружечку зможете перевірити правильність зіставлення кістки вибраному положенню на карті тіла (рис 1.9).

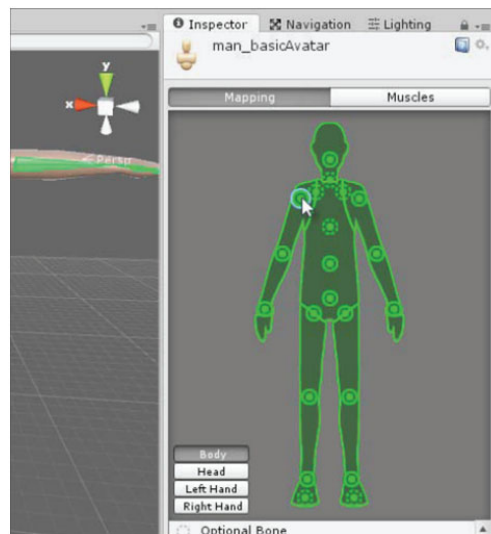


Рисунок 1.9 – Карта кісток для тіла

Якщо персонаж не деформується в екстремальних позах належним чином, можна розкрити групу регуляторів Per-Muscle Settings і задати максимальні та мінімальні межі деформацій (рис.1.10). Інакше кажучи, регулятори контролюють діапазон, у яких скелет впливає на меш, що дозволяє зменшити чи збільшити межі деформацій.

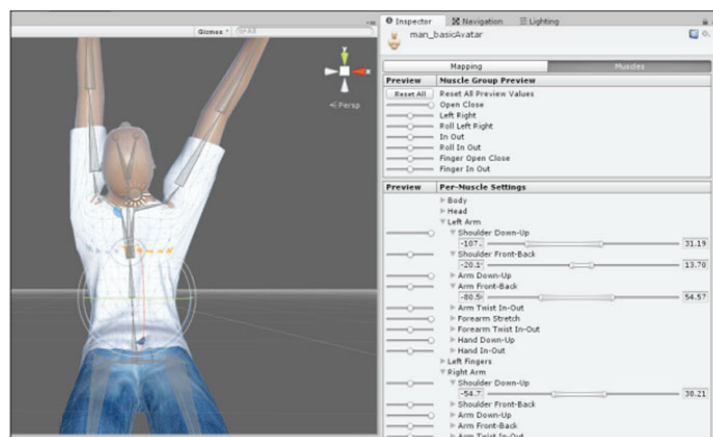


Рисунок 1.10 – Налаштування екстремальних поз за допомогою закладки Muscles

Коли аватар працює та деформації персонажа відповідають нормі, можна натиснути кнопку Apply. При цьому конфігурація аватара буде застосована до файлу меша.

1.4.1 Ретаргейтинг

Правильне настроювання аватара важливе для системи Mecanim, тому що це дозволить виконати ретаргейтинг. Це означає, що будь-який персонаж з правильно налаштованим аватаром може бути анімований за допомогою оснастки та анімації будь-якого іншого персонажа з правильно налаштованим аватаром. Всі налаштування та анімації повністю взаємозамінні між усіма мешами з налаштуваннями аватара гуманоїда, звідси і назва ретаргейтинг (Retargeting) – перенацілювання, тому анімація від одного меша може бути перенацілена для роботи з іншим мешем. Це дозволить повторно використовувати усі анімації персонажів. Наприклад, можна зробити тільки один цикл ходьби, а потім поділитися ним з усіма персонажами, що ходять. Для доступу до даних анімації треба розкрити файл меша і вибрати анімаційний кліп, як це показано на (рис. 1.11).

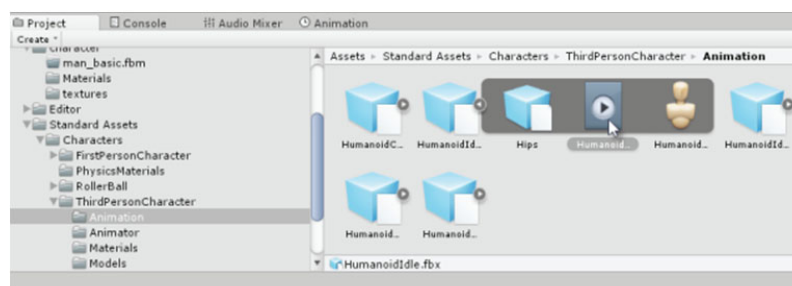


Рисунок 1.11 – Вибір анімаційного кліпу для ретаргейтингу

Більшість анімацій можуть бути переглянуті в інспекторі об'єктів за допомогою моделі Unity за замовчуванням. Можна переглянути ретаргейтинг у дії на прикладі проби анімації для імпортованого персонажа. Змінити модель персонажа для попереднього перегляду можна, натиснувши на іконку аватара в

правому нижньому кутку, і вибрати модель для попереднього перегляду (рис. 1.12).

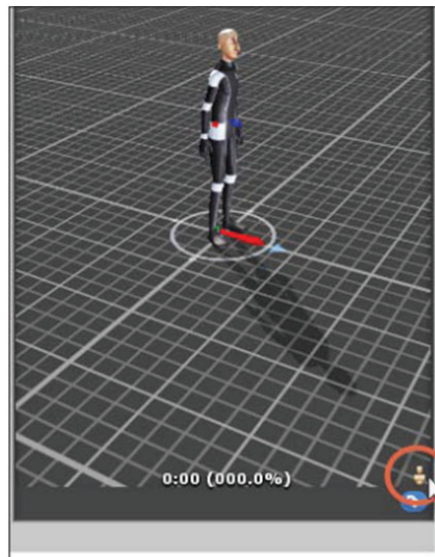


Рисунок 1.12 – Вибір аватара

Після натискання на іконку аватара потрібно вибрати опцію Other і потім вибрати модель персонажа з панелі проекту. Панель попереднього перегляду оновиться в інспекторі об'єктів, щоб показати анімацію, застосовану до персонажа. Якщо вона виглядає добре, то аватар успішно налаштований для ретаргейтингу.

1.4.2 Інверсна кінематика

Інверсна кінематика, що часто скорочується як ІК, дуже важлива для досягнення невимушеної та реалістичної анімації персонажа.

Практично інверсна кінематика дозволяє спозиціонувати кістку руки або ноги в оснащенні, а решта суглобів і кісток буде розміщена автоматично для досягнення реалістичного вигляду. Іншими словами, при створенні анімації ходьби або жестів, де ноги чи руки вступають у контакт із поверхнею, інверсна

кінематика сильно полегшує роботу аніматорів, замість анімації всіх кісток їм достатньо визначити положення стопи та стегна чи руки та плеча (рис. 1.13).

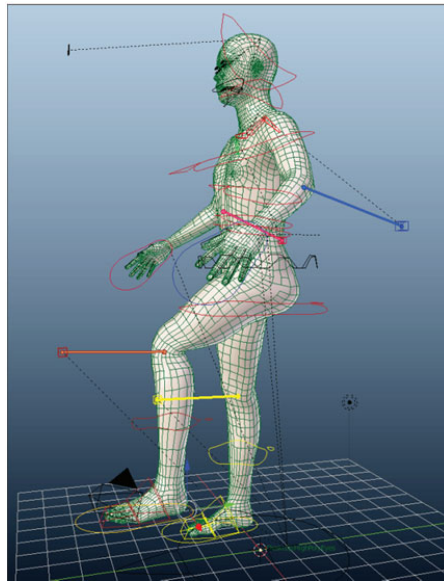


Рисунок 1.13 – Налаштування інверсної кінематики в Blender Вибір

Більшість програм 3D-моделювання, таких як Maya, 3ds Max та Blender, надають можливості та функції для створення інверсної кінематики. Мета цих особливостей – у тому, щоб допомогти художникам створювати анімації легше та швидше. Однак, різні програми зберігають дані інверсної кінематики по-різному, і ці дані зазвичай не можуть бути імпортовані в Unity навіть через файли формату FBX. Це означає, що моделі персонажів майже завжди будуть імпортовані без даних інверсної кінематики. Але це не означає анімації інверсної кінематики, створене в програмах 3D-моделювання виглядатиме інакше після імпорту в Unity. Це лише означає, що всі анімації інверсної кінематики будуть імпортовані до Unity як анімації прямої кінематики (ПК). Вони будуть виглядати так само, але не дадуть доступу до функцій інверсної кінематики, не дозволяючи легко позиціонувати та переміщувати скелет динамічно за допомогою інверсної кінематики. Щоб виправити це, треба буде вручну налаштувати інверсну кінематику для кісток у Unity, якщо потрібно отримати доступ до інверсної кінематики для динамічного переміщення рук та ніг персонажів. Щоб отримати

правильно працюючу інверсну кінематику, спочатку треба імпортувати оснащену модель персонажа в Unity. Потім вибрати модель в панелі проекту та включити оснащення Humanoid, перейшовши на вкладку Rig в інспекторі об'єктів та вибравши тип анімації Humanoid. Це не підключить інверсну кінематику як таку, але налаштує модель для використання системи Аватар, яка потрібна для інверсної кінематики. Якщо застосувати інверсну кінематику до рук персонажа, вона забезпечить, що за розміщення рук всі інші кістки руки автоматично розмістяться так, щоб відповідати положенню рук. Щоб зробити це, треба створити два нові порожні ігрові об'єкти у сцені. Вони являтимуть собою кістки ігрового персонажа. Ці кістки будуть використовуватися для позиціонування руки персонажа. За замовчанням кістки не видно у вікні, якщо вони не вибрані на панелі ієрархії. Щоб кістки стали видимими та інтерактивними об'єктами, треба вибрати іконку куба в інспекторі об'єктів і призначити уявлення 2D Gizmo до об'єкта. Після такого вибору кістки стануть видимими та можуть бути обрані у вікні перегляду.

1.4.3 Анімовані текстури

Текстури – це зображення, які відображаються на геометричних 3D-тілах. Більшість текстур є двовимірними статичними зображеннями, такими як файли JPEG та PNG. Тим не менш, Unity також підтримує і анімовані текстури у вигляді відео- або кіно-текстур. Це дозволить імпортувати в проект відеофайли та використовувати їх як текстури. Фільми можуть відтворюватися зі звуком та відображатися у грі. Таким чином, вони особливо корисні для створення заставок, списків, що взяли участь у розробці, спогадів, реклами та інших спеціальних ефектів. Проте анімовані текстури – ресурс великоваговий, та його застосування сильно знижує продуктивність гри. З цієї причини потрібно використовувати їх

розумно. Unity підтримує різні формати відео, але рекомендованим форматом є OGV (Ogg Theora Video).

Щоб імпортувати анімовану текстуру, треба перетягнути файл формату OGV на панель проекту Unity. Після того як він імпортується, можна переглянути відео, натиснувши кнопку Play на панелі попереднього перегляду інспектора об'єктів (рис.1.14).

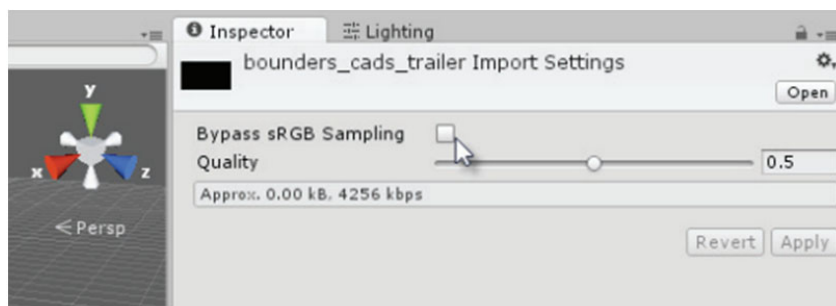


Рисунок 1.14 – Імпорт анімованої текстури з файлу OGV

ВИСНОВКИ ДО 1 РОЗДІЛУ

Згідно з першим розділом магістерської дисертації розглянуто питання персонажної анімації для кіно, а саме поняття моделі персонажу, принципи створення та анімації персонажів. Було розглянуто створення персонажів в програмах 3D-моделювання та підходи до оснащення персонажів на прикладі Unity та Blender. Також розглянуті спеціальні прийоми в персонажній анімації такі як ретаргейтінг, інверсна кінематика та анімовані текстури.

2 МЕТОДИ ТА ОСОБЛИВОСТІ ПЕРСОНАЖНОЇ АНІМАЦІЇ

2.1 Особливості врахування міміки персонажу

Якщо створюється анімація обличчя, така як зміна виразу обличчя або синхронний рух губ, напевно, потрібно буде використовувати змішування форм або анімацію перетворення.

Цей вид анімації зазвичай створюється художником-аніматором у програмі 3D-моделювання, а потім імпортується в Unity. Її створення – це двоступінчастий процес. По-перше, аніматор визначає всі можливі та граничні гримаси меша обличчя, тобто положення та порядок всіх вершин меша обличчя в екстремумах, і створює ключ форми (Shape Key) [37] або змішування форм (Blend Shape) для запису стану меша в цій позі. За допомогою запису серії різних гримас аніматор створює лицьові анімації, в яких генеруються проміжні положення як зважена суміш між розбіжностями в різних гримасах.

На наступних рисунках 2.1 та 2.2, меш голови мавпи в Blender був встановлений в різні екстремальні гримаси. Кожна гримаса записана як ключ форми або змішування форм. У наступному скріншоті відображено початкову або основну гримасу. Іноді вона називається гримасою спокою.

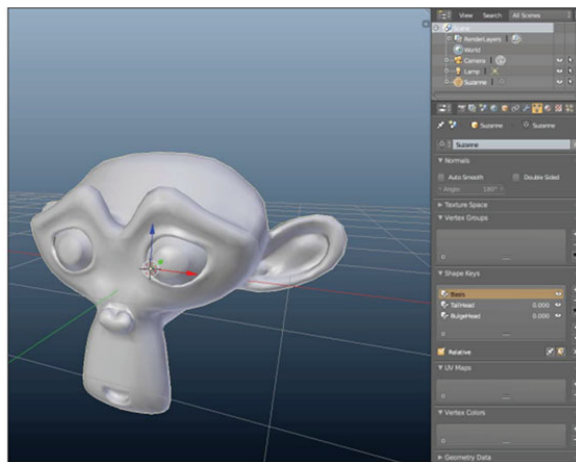


Рисунок 2.1 – Основна гримаса для голови мавпи

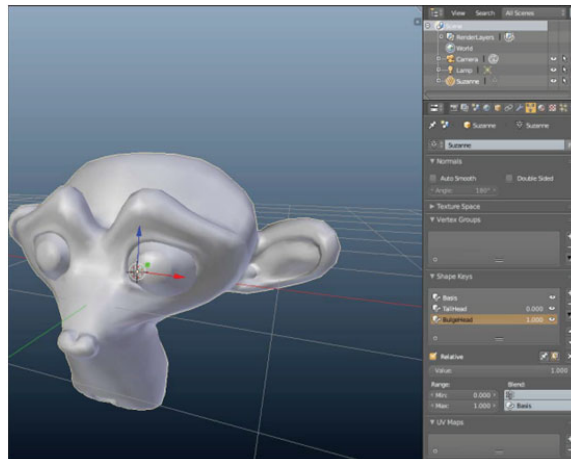


Рисунок 2.2 – Фінальна гримаса для голови мавпи

Після створення набору механізмів вершин та гримас можна експортувати меш із програми 3D-моделювання у файл формату FBX, як і звичайні меші. Кроки експорту відрізняються для різних пакетів.

2.2 Методи скелетної анімації

Методи скелетної анімації засновані на використанні скелета, що є, як правило, ієрархічною структурою кісток. З кожним вузлом структури пов'язані своя локальна система координат (див. рис. 2.8) та тривимірне перетворення. Тривимірне перетворення може бути описане матрицею 4×4 , різні компоненти якої відповідають операціям обертання, переміщення та масштабування відповідно. Кожна кістка займає своє місце в ієрархії скелета і піддається впливу інших кісток: кожна дочірня кістка успадковує трансформації батьківської кістки. Розташування вузлів скелета, що відповідає базовій формі моделі, називається позою прив'язки (binding pose) [38].

Для пози прив'язки матриця перетворення B_i i -ого суглоба скелета буде розраховуватися за такою формулою:

$$B_i = \prod_{j=0}^i T_j R_j,$$

де T_j – матриця зміщення j -ого суглоба; R_j – матриця обертання j -ого суглоба; суглоб з індексом $j=0$ є кореневим, а суглоб з індексом j є батьківським по відношенню до суглобу з індексом $j+1$, $j = \overline{0, i-1}$.

Для будь-якої пози, відмінної від пози прив'язки, матриця перетворення суглоба буде обчислюватися аналогічним чином. Описаний спосіб перетворення положення вузлів скелета називається прямою кінематикою. У разі використання зворотної кінематики аніматор в інтерактивному режимі задає для руки або ноги моделі потрібне положення, а програма автоматично обчислює для кожного з суглобів відповідні кути з урахуванням таких обмежень, наприклад, як фіксовані розміри.

Кожній вершині моделі ставиться у відповідність певна кістка кістяка. Цей процес називається прив'язкою (binding). Прив'язка вершини може виконуватися автоматично до найближчої кістки або здійснюватися тонше – шляхом завдання зв'язків вручну. Таким чином, при трансформації положення окремої кістки змінюється положення пов'язаних із нею вершин. Завдяки цьому робота аніматора значно спрощується: відпадає необхідність анімувати кожну вершину окремо, тому що вершина успадковує трансформації кістки, з якою вона зв'язана.

Первинний варіант алгоритму скелетної анімації було запропоновано Е. Катмулом у 1972 році. Описуючи проблеми автоматизованої анімації у пізнішій статті, Е. Катмул згадує також про алгоритм двомірної скелетної анімації Н. Буртника та М. Вейна, запропонованого 1976 року.

2.2.1 Сегментна анімація

Одним із уявлень моделі персонажа в тривимірній графіці є ієрархія сегментів, які є недеформованими поверхнями; це означає, що всі вершини сегмента під час анімації трансформуються однаково. Ієрархія сегментів була прообразом для ієрархії скелета у наступних методах. Для анімації сегментних моделей достатньо зберігати для кожного сегмента інформацію про положення вершин сегмента, центру сегмента та матрицю перетворення. Через простоту цей метод активно застосовувався в ранніх відеоіграх. Головним недоліком методу є те, що результати його роботи не виглядають реалістично. Це викликано, в тому числі, і неминучими розривами в місцях зчленувань тіл ієрархії, що не деформуються, при анімації.

2.2.2 Анімація без вагів

Алгоритм анімації з прив'язкою вершини до однієї кістки (stitching, rigid skinning, basic skeletal animation) докладно описаний і використовує модель персонажа, представлену єдиною оболонкою, а не ієрархією окремих сегментів. Даний метод отримав назву stitching («зшивання»), тому що до кістки прив'язується не поверхня або сегмент цілком, а кожна з вершин.

Трансформоване положення окремої вершини \vec{v}' розраховується за такою формулою:

$$\vec{v}' = WB^{-1}\vec{v},$$

де \vec{v} – базове положення вершини; B – матриця суглоба, трансформації якого успадковує розглянута вершина, у позі прив'язки; W – поточна матриця цього суглоба, що відповідає трансформованій позі скелета. Усі компоненти формули

задані у глобальних координатах. Цю формулу можна дати геометричну інтерпретацію (рис. 2.3).

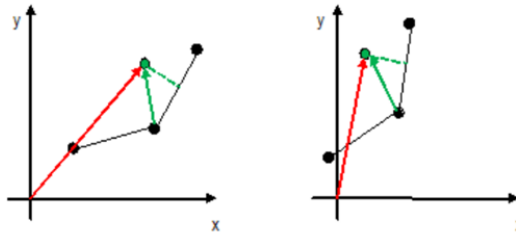


Рисунок 2.3 – Інваріантність положення вершини щодо системи координат, пов'язаної із суглобом.

Спочатку здійснюється переведення координат вершини \vec{v} з глобальної системи координат у локальну систему координат, яка пов'язана з розглянутим суглобом. При переміщенні суглоба положення вершини в системі координат суглоба $\vec{v}_B = B^{-1}\vec{v}$ залишається незмінним. Потім проводиться розрахунок трансформованого положення суглоба – обчислюється матриця W .

Після цього здійснюється зворотний перехід до глобальної системи координат: вершина успадковує трансформації суглоба, і підсумкове положення вершини обчислюється як $\vec{v}' = W\vec{v}_B$. При такій прив'язці неминучі розтягування та самоперетинання моделі, на згинах модель виглядає неприродньо (рис. 2.4).

2.2.3 Еталонні форми в персонажній анімації

Алгоритми аналізованої групи використовують набір еталонних форм-прикладів моделей і відповідні конфігурації скелета [39]. Дані еталонні форми можуть бути отримані з низки джерел: вони можуть бути зроблені вручну фахівцем з 3D моделювання, бути результатом 3D-сканування (технологія Motion Capture), а також можуть бути згенеровані шляхом застосування алгоритмів, що симулюють анатомію людини. Далі результати трансформації, отримані за допомогою алгоритму LBS, порівнюються із заданими еталонними формами та

коригуються відповідно до них. Основний недолік цієї групи алгоритмів полягає в необхідності отримання та використання дуже великої кількості еталонних форм для здобуття реалістичних результатів. Для коректної роботи методів необхідно надавати такий набір еталонних форм, який буде повно описувати всі можливі рухи персонажа, тобто буде репрезентативним. Проводячи корекцію роботи алгоритму LBS, дані методи поступаються йому швидкодії.



Рисунок 2.4 – Скелетна анімація без вагів

2.3 Спеціальні алгоритми персонажної анімації

2.3.1 Pose Space Deformation

Алгоритм анімації в просторі поз PSD є базовим алгоритмом групи, що розглядається, і поєднує в собі алгоритми морфінгу і скелетної анімації LBS. Даний метод передбачає використання пар <еталонна форма моделі, конфігурація скелета> для припасування результатів застосування LBS до очікуваної кінцевої форми.

Для синтезу анімації між ключовими кадрами використовується лінійна інтерполяція із використанням радіальних базисних функцій (РБФ). Під РБФ маються на увазі такі речові функції, які залежать від відстані поточної точки x до центру РБФ c : $\varphi(x, c) = \varphi(\|x - c\|)$.

Вершина з вихідного положення \bar{v}_0 переміщається за допомогою алгоритму скелетної анімації LBS у положення \bar{v}_i , $i = 0, 1, \dots, N$. Потім отримана форма компенсується зсувами \bar{d}_i , $i = 0, 1, \dots, N$, які відповідають еталонними формами моделі \bar{x}_i , $i = 0, 1, \dots, N$. Таким чином, вершина \bar{v} в результаті переміщається у положення $\bar{v}'_i = \bar{v}_i + \bar{d}_i = LBS_i(\bar{v}_0) + \bar{d}_i$. Далі, щоб отримати проміжні положення вершини, проводиться інтерполяція по \bar{d}_i : $\bar{d}_i : \bar{d}_i = \sum_{i=0}^{N-1} \omega_i \varphi(\|x_i - x_j\|)$, де x – проміжна форма моделі, тобто форма відмінна від еталонних форм моделі; $\|x_i - x_j\|$ – евклідова відстань між формами моделі x_i і x_j ; $\varphi(x)$ – гаусова РБФ; ω_i – ваги відповідних еталонних форм.

Через лінійність інтерполяції деформації моделі у проміжних кадрах не відповідають повною мірою відповідним рухам реальної людини. Тому інтерполяцію доводиться проводити у великій кількості досить близьких еталонних форм.

Метод анімації в просторі поз в даний час широко застосовується, хоча аніматори і відзначають певну складність роботи з PSD: застосування цієї технології при моделюванні головного персонажа повнометражного анімаційного фільму "Bolt" зайняло приблизно 3 тижні. Щоб спростити і, відповідно, прискорити роботу з PSD, працівники Walt Disney Animation Studios були змушені розробити велику кількість допоміжних інструментів. У 2006 році було реалізовано апаратну візуалізацію на основі алгоритму PSD.

2.3.2 Алгоритм П. Слоана

Алгоритм SE на відміну від PSD вимагає вирішити по одній лінійній системі на кожну еталонну форму, а не на кожен компонент вершини (тобто $3n$ лінійних систем, де n – кількість вершин моделі). Таке формулювання завдання, як правило, дозволяє досягти роботи зі значно меншим об'ємом даних.

Розглянутий алгоритм також використовує РБФ при інтерполяції, але в комбінації з лінійною гіперплощиною, що дозволяє проводити як гладку інтерполяцію, а й екстраполяцію.

2.3.3 Алгоритм, заснований на скануванні

На відміну від PSD і SE метод, запропонований Б. Алленом у роботі, використовує як еталонні форми дані, отримані в результаті сканування. Це дозволяє отримати досить реалістичні результати, але ставить нові завдання: доповнення сканограм для отримання цілісної форми, визначення конфігурації скелета кожної еталонної форми, установка взаємооднозначної відповідності між еталонними формами [40]. Для розрахунку трансформації кісток

використовуються кватерніони. При вирішенні задачі інтерполяції між еталонними формами при побудові зображення моделі в довільній конфігурації автор наголошує на недоліках використання РБФ: радіальні базисні функції працюють ефективно при обертанні кістки лише по одній осі. Крім того, РБФ інтерпретують компоненти кватерніону як лінійно-незалежні та допускають негативні значення вагових коефіцієнтів. У зв'язку з цим для вирішення задачі інтерполяції аналізований алгоритм застосовує метод k найближчих сусідів: вибираються точки k найближчих еталонних форм і їм приписуються ваги з урахуванням їх віддаленості від форми, що відповідає поточній конфігурації скелета. Цікаво, що інтерполяція проводиться окремо для кожної частини тіла, а потім здійснюється плавне сполучення частин.

2.3.4 Метод П. Края

Алгоритм EigenSkin передбачає наявність еталонних форм, отриманих у результаті роботи алгоритмів, що симулюють анатомію людини. EigenSkin не вимагає зберігання зсувів для всіх еталонних форм, як PSD, а застосовує метод головних компонентів (МГК) для стиснення вихідних даних шляхом побудови базису еталонних форм. Однак МГК застосовується не до поля зсувів по всій моделі, оскільки це призвело б до великої кількості базисних функцій. Замість цього модель розкладається на складові, які визначаються за впливом окремих кісток на полі зсувів. У результаті алгоритм отримує набір базисів, який і застосовується при візуалізації. Таким чином EigenSkin реалізує лінійну версію складних нелінійних алгоритмів.

ВИСНОВКИ ДО 2 РОЗДІЛУ

Згідно з другим розділом магістерської дисертації розглянуто таке питання як врахування міміки персонажу, а саме всі можливі та граничні гримаси меша обличчя. Були розглянуті методи скелетної анімації, сегментної анімації, анімації без вагів та еталонні форми в персонажній анімації.

3 РИГІНГ ПЕРСОНАЖА, АНАЛІЗ ТА ПРОГНОЗУВАННЯ АНІМАЦІЇ

3.1 Ригінг за допомогою інверсної кінематики

Для початку процесу ригінгу в програмному забезпеченні Blender, потрібно додати до сцени “кістку”, за допомогою сполучення клавіш “Shift + A”, і обравши вкладку “Armature” (рис. 3.1).

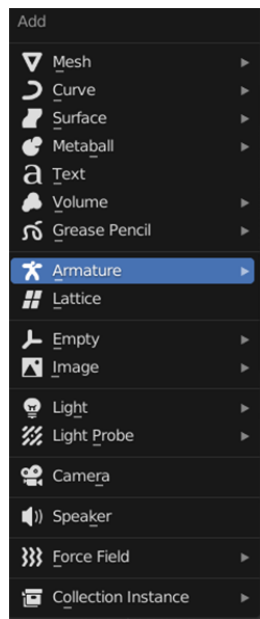


Рисунок 3.1 – Додавання “кістки” в сцену

Після цього в сцені з’являється кістка (рис. 3.2 а), за допомогою неї починають будувати скелет персонажа за допомогою інструмента екструдування (рис 3.2 б).

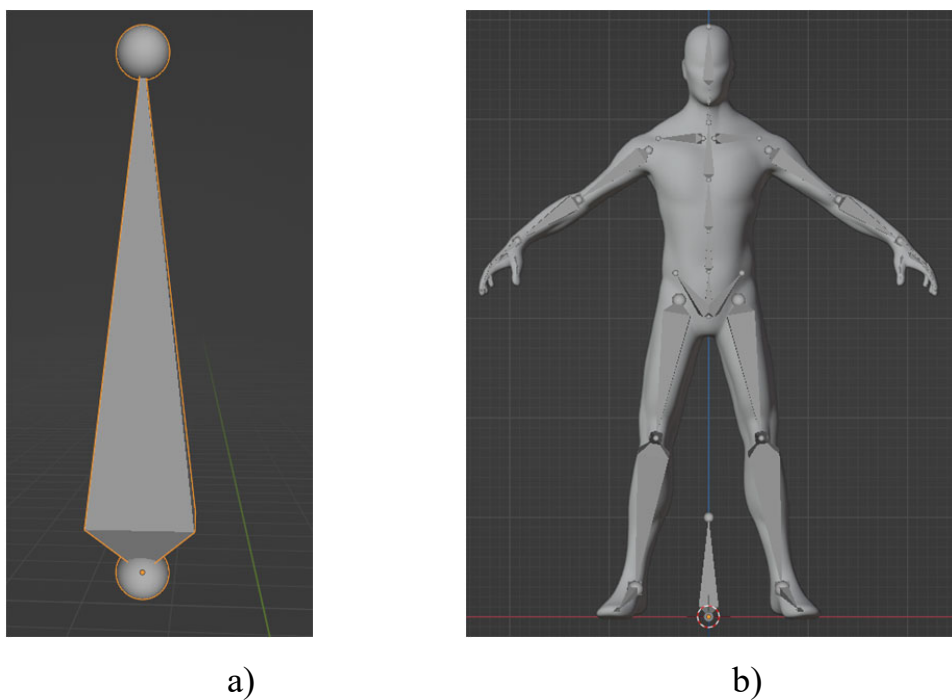


Рисунок 3.2 – а) кістка b) скелет

Далі переходять до інверсної кінематики, за допомогою інструмента “Extrude” в області колін та ахіллового сухожилля створюють кістки керування що напрямлені в інший бік (рис. 3.3).



Рисунок 3.3 – Кістки для інверсної кінематики

За допомогою сполучення клавіш “Alt +P”, відкривають меню в якому натискають “Clear Parent”, аби позбутись зв’язків (рис 3.4).

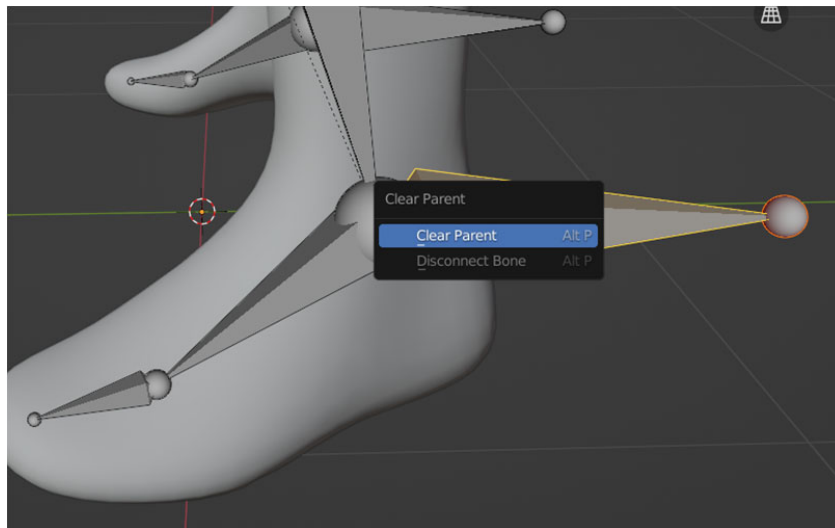


Рисунок 3.4 – Функція Clear Parent

Перейменовують ці кістки для зручності, потім переходять в режим позування, натискають на потрібну кістку і у вкладці “Bone Constraint Properties” обирають “Inverse Kinematics” (рис. 3.5).

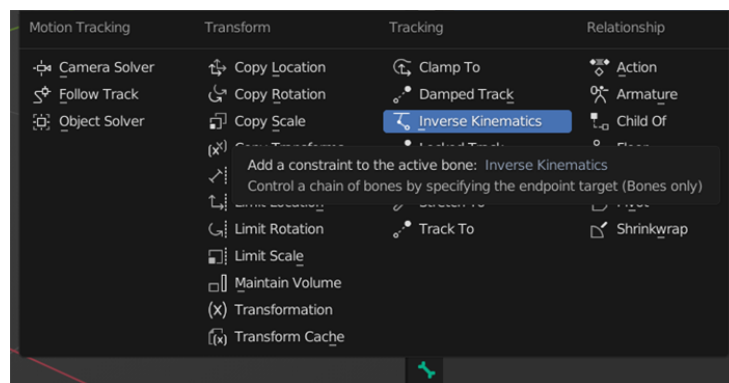


Рисунок 3.5 – Додавання режиму інверсної кінематики на кістку

Обирають потрібні з’єднання і потрібні кістки пофарбуються в зелений колір (рис. 3.6).

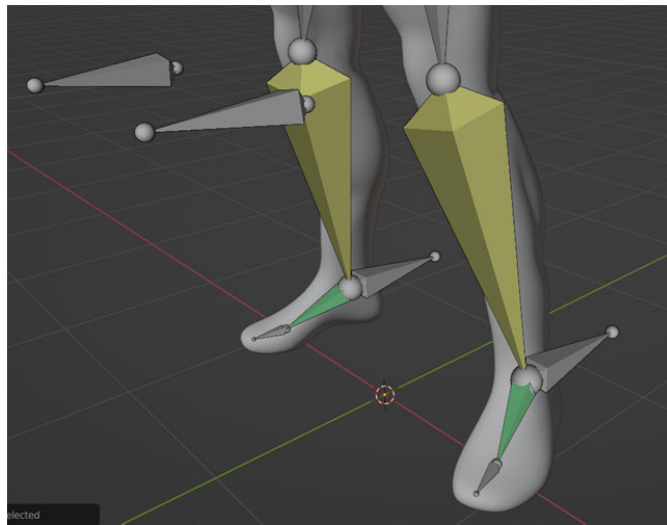


Рисунок 3.6 – Налаштована інверсна кінематика ніг

Ті самі кроки використовують для кісток рук (рис. 3.7).

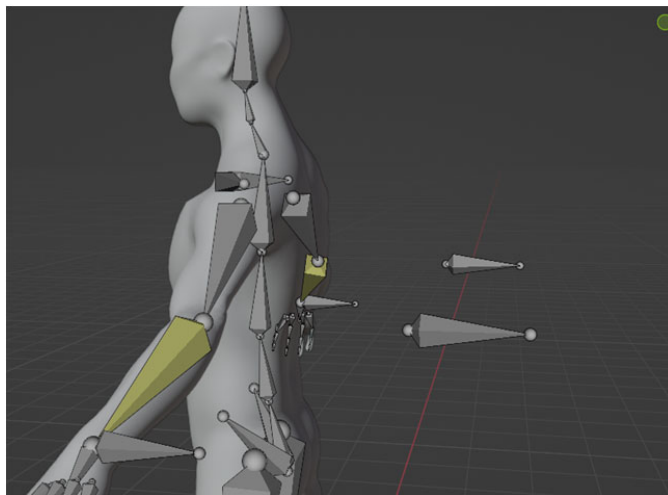


Рисунок 3.7 – Налаштована інверсна кінематика рук

Для повного контролю ступнями потрібно в Edit Mode зв'язати кістку керування та кістку ступні за допомогою сполучення клавіш “CTRL + P” і обрати “Keep Offset”. Після цього в “Pose Mode” потрібно виділити кістку ступні та в меню “Bone Constraint Properties” додати “Copy Location” і обрати кістку до якої вона прив’язана рис. 3.8), ті самі кроки роблять для іншої ступні.

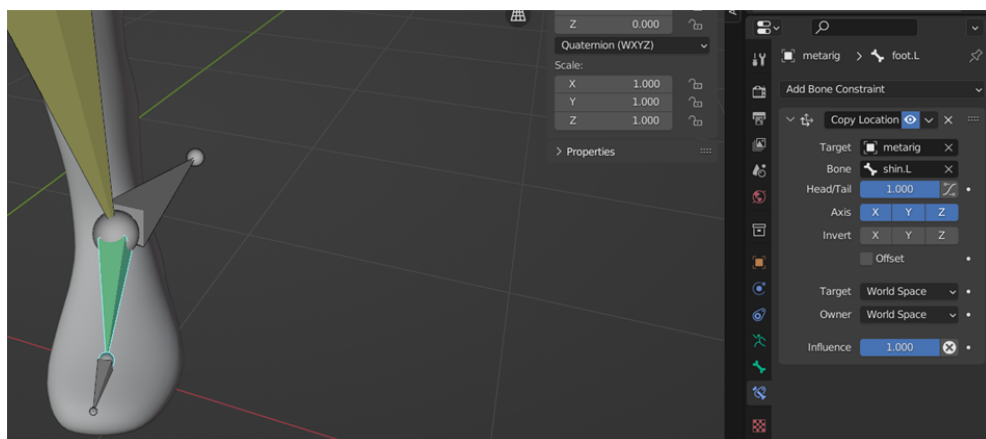


Рисунок 3.8 – Прив’язана кістка ступні до кістки для інверсної кінематики

Після цих кроків ригінг за допомогою інверсної кінематики можна вважати закінченим, в Pose Mode, можна налаштовувати позу персонажа під будь які потреби (рис 3.9).

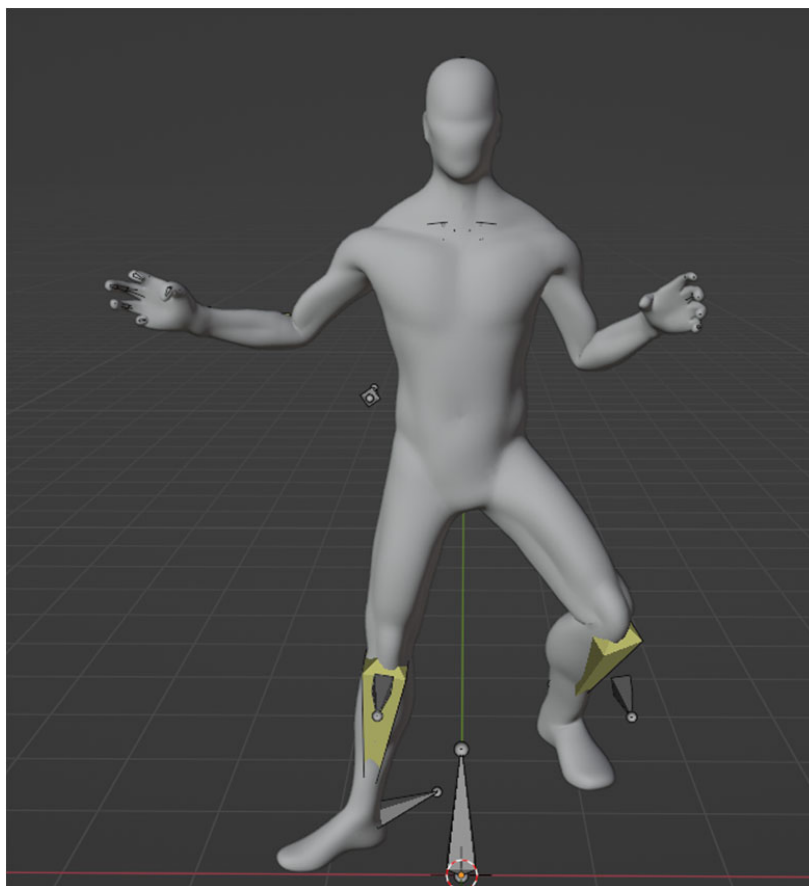


Рисунок 3.9 – Закінчений ригінг та налаштована поза персонажа

3.2 Інверсна кінематика

Кінематика це процес обчислення позиції в просторі для кінця взаємопов'язаної структури при заданих кутах повороту всіх шарнірних точок (Joints). Це легко здійснюється, якщо є лише одне рішення. Інверсна кінематика робить протилежне. Для цієї кінцевої точки обчислюються необхідні кути повороту шарнірів таким чином, щоб вдалося досягти цієї кінцевої точки. При цьому можуть виникнути труднощі, якщо є велика або нескінченна кількість рішень.

Цей процес надзвичайно корисний у робототехніці. Наприклад: рука робота потяглася та взяла об'єкт. Якщо програма знає місце розташування об'єкта щодо плечового суглоба, їй достатньо розрахувати кути повороту шарнірів щоб досягти об'єкта. Також інверсна кінематика корисна у 3D іграх. Наприклад дракон з дуже довгою шиєю повинен реалістично вигнути шию і з'їсти гравця, що стоїть на підлозі. Або гравець захотів підняти з підлоги об'єкт або натиснути на кнопку. Користувач побачить на екрані як гравець потягнеться і торкнеться об'єкта, замість махнути рукою десь поблизу об'єкта.

3.2.1 Якщо нема рішень

Існує багато випадків, коли пов'язана структура не може досягти мети. Наприклад, не можна торкнутися ліктем однієї руки зап'ястя тієї ж руки. Або дістати із землі до верхівки високого дерева (рис.3.10).

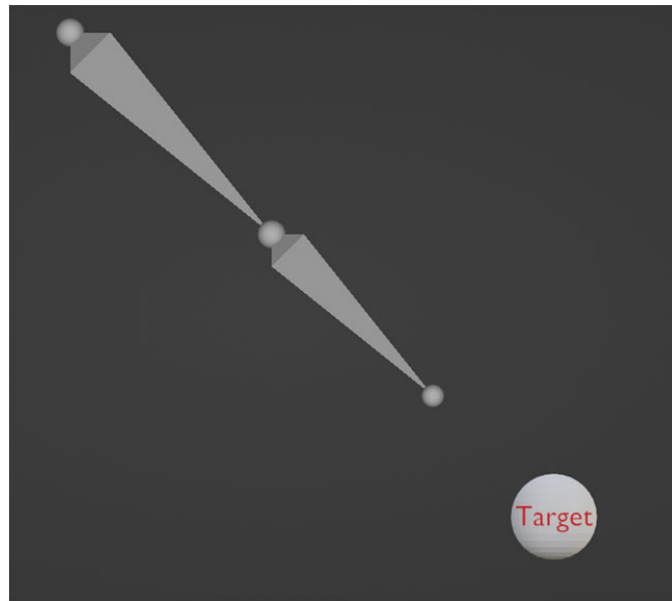


Рисунок 3.10 – структура не може досягти цілі

3.2.2 Одне рішення

Якщо є два рішення, то повинна існувати техніка вибору, яке рішення більше відповідає поточному стану структури (рис.3.11-3.12).

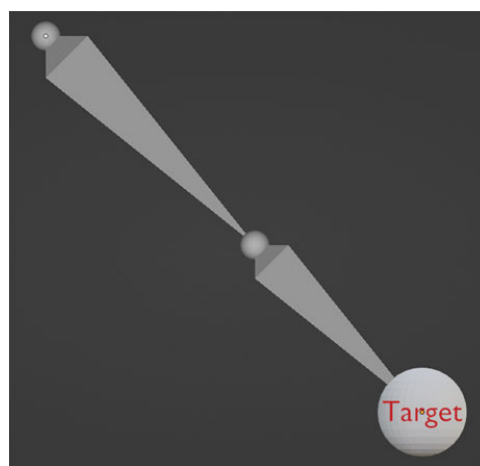


Рисунок 3.11 – одне рішення для структури

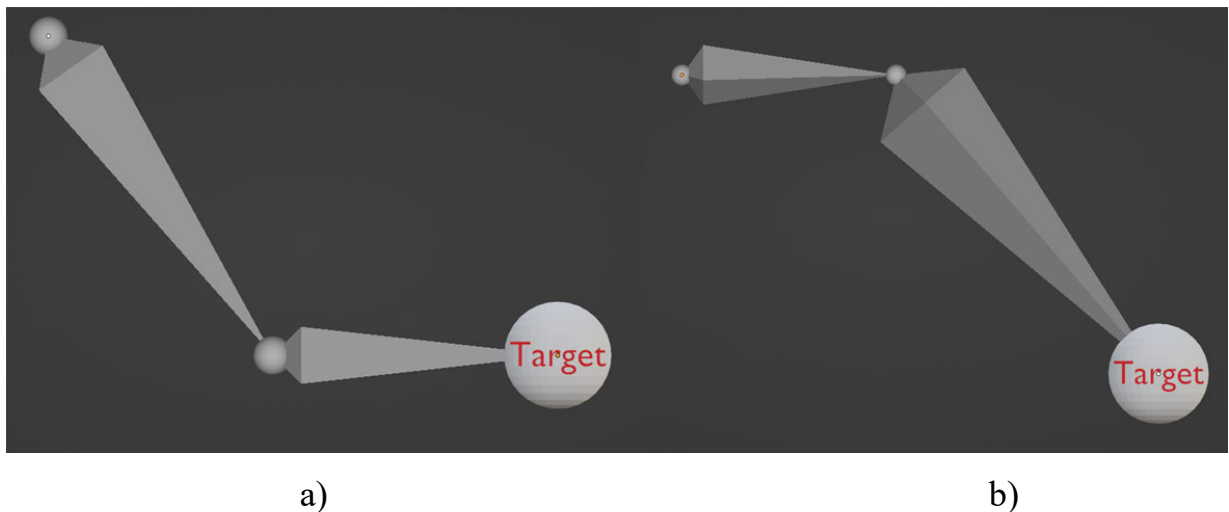


Рисунок 3.12 – а) одне рішення і b) друге рішення для структури

3.2.3 Багато рішень

Коли використовується більше двох шарнірів, часто трапляється, що існує багато розв'язків задачі. Тим не менш, деякі рішення є кращими за інші. Якщо структура представляє, наприклад, руку, деякі рішення більш зручні, інші дуже неприродні. Найчастіше є оптимальне рішення.

Алгоритм не одразу знаходить рішення. Це не проста функція де ви задаєте місце призначення та отримуєте кути. Це ітеративний алгоритм, що робить алгоритм більш реалістичним у програмах, особливо в іграх, де потрібно, щоб об'єкти рухалися згладжено.

Рішення знаходиться за рахунок віртуального застосування сили до кінця ланцюжка та перетягування кінця ланцюжка до кінцевої позиції. Сила діє на кожне зчленування і рухає структуру трохи ближче до кінцевої точки. Так що навіть якщо рішення немає, ланцюжок наблизиться до кінцевої точки так близько, наскільки це можливо. Так як шарніри розраховуються окремо, можна задати їм різні властивості, такі як жорсткість, еластичність і т.д. Кожен із шарнірів також

може мати таку оптимальну позицію, яку алгоритм намагатиметься досягти настільки близько, як тільки це буде можливо.

3.3 Інверсна кінематика у 2D

Починаючи з одного шарніру, у цьому випадку він може обертатися навколо лише однієї осі (напрямок проти годинникової стрілки обирають як позитивний).

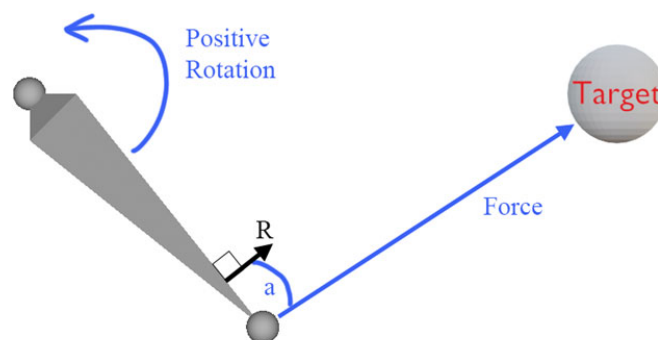


Рисунок 3.13 – Один шарнір

Вектор R діє під прямим кутом кістки. Вектор сили F діє від кінцевої точки у напрямку мети. Кут між R та F є a . Швидкість, з якою потрібно повертати шарнір, пропорційна скалярному добутку (dot product) векторів R і F . Дотримуючись точки зору, що скалярний добуток двох векторів позитивне число, і, звичайно, шарнір повинен обертатися в позитивному напрямку, щоб максимально наблизитися до мети. Якщо кут між R і F прямий, шарнір повернувся так, щоб кістка була максимально близько до мети і тому шарнір не повинен більше повертатися. У цій точці скалярний добуток двох векторів дасть нуль.

$$\text{torque} = (R \cdot F) \times k \quad (3.1)$$

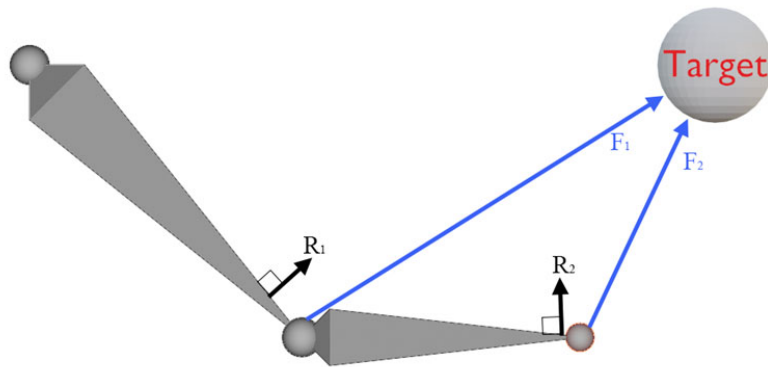


Рисунок 3.14 – Дві кістки

Надалі все буде різновидом цього випадку. Коли є більше однієї кістки, потрібно повторити алгоритм для кожного шарніра по черзі, що був описаний раніше.

Починаючи з шарніру найближчого до мети:

1. Обчислюють вектор сили (від кінцевої точки до мети)
2. Обчислюють скалярний добуток вектора сили та вектора, перпендикулярного до кістки.
3. Помножують скалярний добуток на невелике число (наприклад, 0.01)
4. Додають його до кута шарніра.

Цей процес застосовують для всіх шарнірів послідовно.

3.4 Інверсна кінематика у 3D

У тривимірному просторі обчислення будуть не набагато складнішими. Є три можливі осі, навколо яких шарнір може обертатися. Вищеописаний метод у тривимірному просторі, має не найкращі результати. Проте ця математика, безумовно, працює. Є щось, що вимагає великої кількості змін установок, щоб усе працювало, безперечно, правильно.

3.4.1 Вісь шарніра перпендикулярна кістці

На прикладі, що наведений вище, але цього разу на 3D. Це система з двома шарнірами, і спершу обирають перший шарнір.

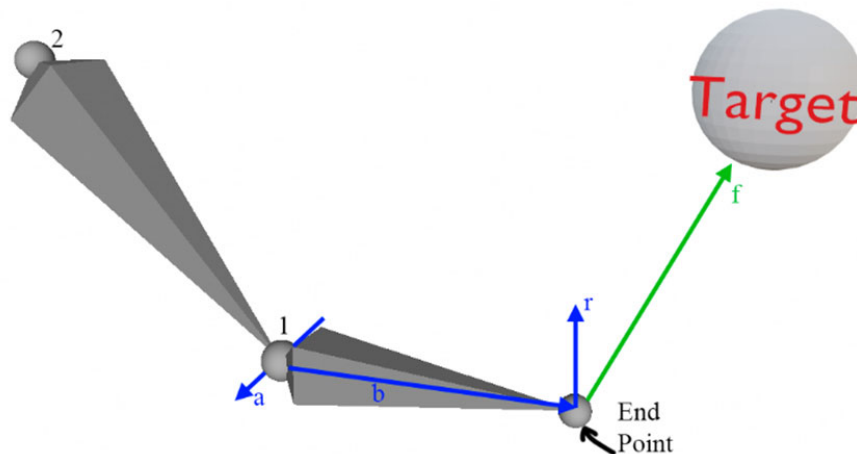


Рисунок 3.15 – перший шарнір структури в 3D

Цей процес застосовують для всіх шарнірів послідовно.

a – вектор уздовж осі шарніра.

b – вектор уздовж кістки.

r – вектор перпендикулярний a і b .

f – вектор сили (діє від кінцевої точки до мети).

Якщо вектор сили буде паралельний a , шарнір не обертатиметься. Якщо він паралельний b , то потрібно потягнути кістку, шарнір не обертатиметься. виходячи з цього, момент, що обертає, повинен бути пропорційний синусу кута між a і f і синусу кута між b і f .

Якщо кінцева точка прийшла в точку мети, рух структури не потрібен, але якщо кінцева точка далеко від мети, потрібно, щоб структура рухалася швидко. Отже, момент, що обертає, повинен бути пропорційний довжині вектора сили.

Є тільки одна річ, яку не можна зрозуміти з вищеописаного – напрямок у якому має обертатися шарнір. Нехай шарнір повинен обертатися в позитивному

напрямку, якщо вектор сили спрямований уздовж r , і в протилежному напрямку, якщо вектор сили спрямований протилежно r .

Склавши все разом отримують:

$$torque = Mag(f) \times SinVect(a, f) \times SinVect(b, f) \times sign(CosVect(r, f)) \times Sensitivity \quad (3.2)$$

У цій формулі $SinVect$ та $CosVect$ повертають відповідно синус та косинус кута між векторами. Mag повертає довжину вектора, і $sign$ повертає знак числа. $Sensitivity$ – це невелика скалярна константа.

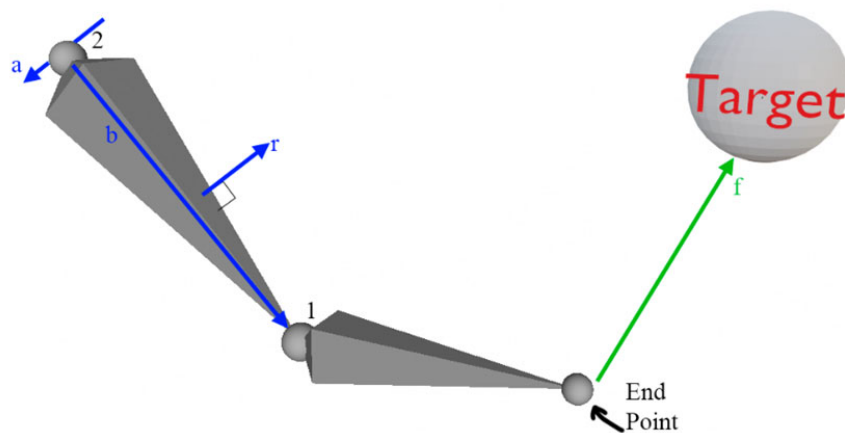


Рисунок 3.16 – другий шарнір в 3D

Тепер можна рухатися до наступного шарніру у структурі. Обчислюють нові значення векторів a , b , f , та рахують обертаючий момент для шарніра 2. Обчислюють обертаючий момент для всіх шарнірів у структурі і додають до кута кожного шарніра за обертаючим моментом. Кінцева точка має тепер бути ближчою до мети. Безперервно повторюючи процес, можна наблизити кінцеву точку структури до мети максимально близько або навіть торкнутися мети.

3.4.2 Вісь шарніра спрямована вздовж кістки

У більш важкому випадку видно пов'язану структуру, що складається з чотирьох шарнірів. Шарнір номер два, проте, може обертатися довкола осі своєї кістки. Це схоже на вигин зап'ястя.

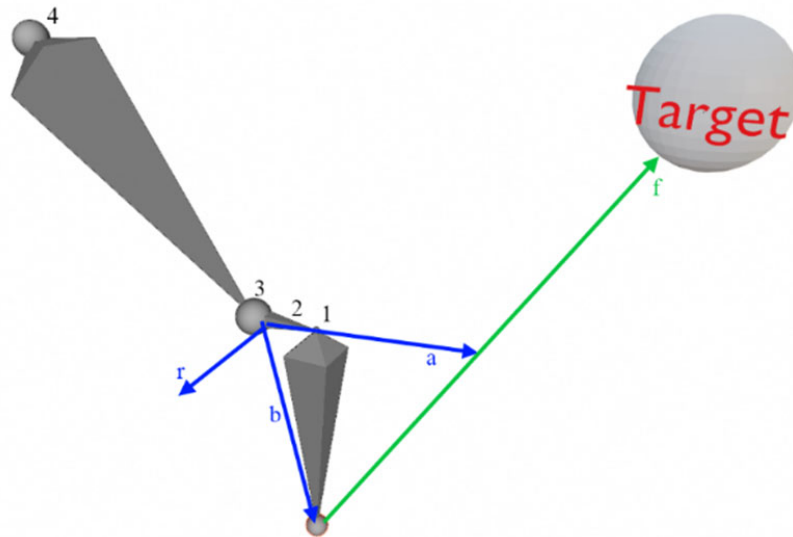


Рисунок 3.17 – 4 шарніри в 3D

І звичайно цей шарнір найбільш важливий. Як і в попередньому випадку, вектор a проходить вздовж осі шарніра, але в цьому випадку це ще й уздовж кістки. Вектор b йде від шарніра до кінцевої точки. Вектор r як і раніше перпендикулярний обом.

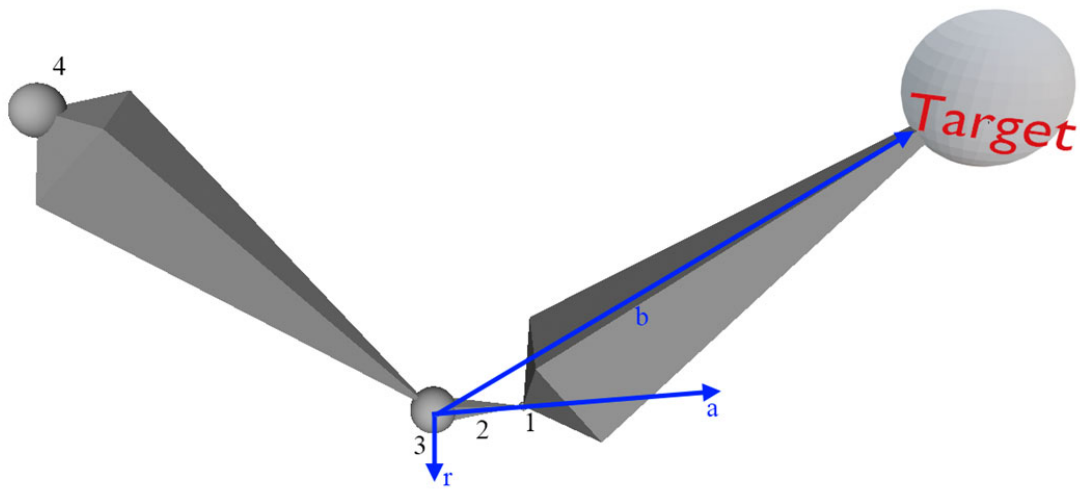


Рисунок 3.18 – повернуті до цілі шарніри в 3D

Щоб досягти мети, шарнір 2 повинен повернутися на 90 градусів (див. праву картинку).

Вектор b різний для шарнірів чия вісь обертання паралельна кістці.

Також є завдання регулювання системи, щоб задати її поведінку в різних ситуаціях. Є багато різних рішень, деякі кращі, а деякі гірші. Розрізняються тим, як вони виглядають, величиною витрачених рухів або зручністю.

3.5 Зручне положення

Кінцівкам краще знаходиться в зручному положенні. Це може бути виконано новим додаванням еластичності шарнірів. Проте в цьому випадку може бути найкращим мати певний ряд положень для кожного шарніра в якому немає опору. Після того як шарнір залишає зручне положення, він опиратиметься все сильніше і сильніше поки зовсім не відмовиться обертатися.

Також бажано тримати кінцівку максимально низько. Зазвичай люди не довго тримають свої руки у висячому положенні більше, ніж потрібно. Тому

положення вибирається виходячи з мінімальної енергії, яка потрібна для утримання положення.

3.6 Анімація

Традиційно мистецтво мультиплікації займало багато часу з ручкою й олівцем, що дуже залежало від майстерності залучених аніматорів.

Мистецтво комп'ютерної анімації зробило величезний крок вперед від цих традиційних технік. Можливо, першою роботою, яка продемонструвала силу використання комп'ютерів для анімації, був міні-шедевр Ріхард 1986 року «Luxo Jr».

Luxo Jr. — це комп'ютерна анімація, у якій зображена лампа, яка грає з м'ячем. У той час анімація робилася за допомогою тривіального методу ключових кадрів.

З тих пір комп'ютерна анімація розвивалася семимильними кроками. Реалістичність анімованих фігур більше не залежить від навичок аніматора, а нові методи пропонують способи анімації фігур із дуже незначним втручанням аніматора.

3.6.1 Основні прийоми

Кілька базових прийомів стали помітними в останні роки та широко використовуються в багатьох різних областях анімації персонажів. Наступне можна вважати частинами «базового інструментарію», який можна використовувати для створення потужних систем анімації.

Більш традиційний метод анімації артикульованої фігури в комп'ютерній графіці передбачає визначення кожної конкретної частини в певних ключових

місцях у просторі, а потім використання певної техніки інтерполяції для анімації між кадрами для відповідного руху, [1]. Цей метод створення руху відомий як ключове кадрування, і ручне введення поз є, мабуть, найпростішим методом визначення руху. Аніматор повинен створювати реальний рух світу вручну. Це не тільки виснажливе та трудомістке завдання [2], але й покладається на майстерність аніматора для створення реалістичних, переконливих ефектів.

В [3] пропонують техніку редагування захопленої анімації або анімації з ключовими кадрами на основі викривлення кривих параметрів руху. Аніматор визначає набір ключових кадрів, які діють як обмеження, щоб отримати плавну деформацію, яка зберігає структуру вихідного руху. Багато різних реалістичних рухів можна отримати з однієї захопленої послідовності рухів, використовуючи лише кілька ключових кадрів для визначення деформації руху. Цей метод добре вписується в існуючу парадигму ключових кадрів. На жаль, він також має деякі обмеження ключових кадрів (дотримання геометричних обмежень для ключів є складним). Метод не базується на якомусь особливо глибокому розумінні структури рухів. Через це екстремальні деформації можуть виглядати спотвореними та неприродними.

Кінематичні методи визначають рух незалежно від основних сил, які викликали рух. Для анімації персонажів корисно розглядати кінематику вперед як процес явного визначення всіх спільних рухів, щоб визначити положення вільного кінця ланцюга (його називають кінцевим ефектором). Навпаки, інверсна кінематика визначає положення та орієнтацію всіх з'єднань в ієрархії з урахуванням стану кінцевого ефектора. Використовуючи зворотну кінематику, геометричні дані потрібно вказати лише для однієї або кількох конкретних гілок у межах артикульованої фігури, а інші в ієрархії слідуватимуть набору [4]. Хоча такий метод надає аніматору корисний інструмент, створення реалістичного руху за допомогою зворотної кінематики є проблематичним, оскільки рух ланок у будь-

якій шарнірній фігурі визначається за допомогою конкретних атрибутів руху, а не за допомогою сил і прискорень. Однак це дозволяє використовувати більш динамічний підхід до анімації персонажів, оскільки конкретні рухи не потрібно визначати заздалегідь. Ще одна дуже популярна техніка – захоплення руху. [5] Обговорює, як створюється реалістичний рух шляхом фіксації руху реального актора за допомогою оптичного відстеження спеціальних датчиків, прикріплених до ключових точок на тілі актора, або відстеження їх магнітним шляхом. Геометричні дані, необхідні для створення реалістичного руху, отримують із даних, зібраних із цих датчиків. Протягом багатьох років ця техніка була дуже популярною в індустрії кіно та комп'ютерних ігор. Хоча він має потенціал створювати дуже реалістичні рухи, ці рухи також є дуже специфічними і, поодинокі, не піддаються впливу факторів динамічного середовища, таких як гравітація та інерція. Деяка цікава робота [6] зосереджена на поєднанні захоплення руху з динамічними факторами.

Хоча кінематика працює незалежно від основних сил, динаміка моделює закони фізики Ньютона, щоб отримати фізично правдоподібну анімацію.

Пряма динаміка керує об'єктами, маніпулюючи силами та крутними моментами, щоб отримати траєкторії та швидкості. З іншого боку, інверсна динаміка визначає сили та крутні моменти, необхідні для створення рухів у системі. Системи, засновані на динаміці, часто називають фізично заснованими, щоб відрізнити їх від чисто геометричних підходів, таких як кінематика. Два підходи зазвичай вимагають розгляду при використанні фізичних методів; пряме (на основі кроків) моделювання та методи, засновані на обмеженнях, таких як просторово-часові обмеження, які будуть обговорюватися найближчим часом.

3.6.2 Генерація руху

Анімація естетично привабливих, реалістичних рухів для людиноподібних персонажів була складною проблемою в галузі комп'ютерної анімації. Той факт, що люди дуже гостро розпізнають правдоподібний рух, робить це завдання досить складним.

Щоб змусити персонажа рухатися реалістично, необхідно використовувати відповідну техніку генерації руху. Обраний метод значною мірою залежить від поставленого завдання та того, наскільки динамічним буде середовище. У багатьох випадках може бути корисним використання кількох різних технік. Наприклад, деякі системи можуть вибрати анімацію рухів ходьби персонажа за допомогою техніки захоплення руху, надаючи персонажу здатність захоплення без сценарію за допомогою інверсної кінематики. Використані основні прийоми визначають загальну реалістичність, швидкість обробки та динамічні здібності персонажа. У свою чергу, динамічна здатність визначає обсяг втручання аніматора, необхідного для анімації персонажа в різних ситуаціях.

Джерело [7] надає хороший огляд місцевості в огляді різних технік, задіяних у анімації людини, що йде.

Згадуються кінематичні та динамічні підходи для методів анімації артикульованої фігури, на додаток до анімації на основі даних руху. Для підходу, заснованого на динаміці, торкаються просторово-часових обмежень, а також розглядається викривлення руху.

3.6.2.1 Генерація параметрів

Для особливих ситуацій руху було розроблено ряд методів параметричного генерування руху.

Система Джека, описана [8], використовує динамічні та кінематичні моделі людей на основі біомеханічних даних із вбудованою поведінкою, такою як ходьба та хапання, для імітації руху людини. Ці вбудовані моделі поведінки узагальнено з даних захоплення руху, щоб забезпечити реалістичний рух.

В [9] використовують спрощену динамічну модель і алгоритм керування, щоб генерувати рухи людини, що йде в K.L.A.W.. Модель телескопічної ноги використовується з 2 ступенями свободи для фази «стійки», тоді як складна модель маятника використовується для фази «гойдання». Гібридна система цілеспрямованого динамічного руху використовується для керування рухом при ходьбі. Верхня частина тіла та руки додаються кінематично та рухаються за коливальним шаблоном, який нагадує людський.

Джерело [10] описує високорівневу моду визначення руху ходьби. За допомогою змінних у часі обмежень, таких як «тримати тулуб у вертикальному положенні», «підтримувати кліренс від землі під час маха ногами» та «підняти й опустити книгу», програмується рух ходьби для моделі людини.

В [11] фокусується на моделюванні рухів рук для підйому на основі моделей м'язів людини. Під час моделювання враховуються такі фактори, як рівень комфорту, сприйняте навантаження та сила, [12] застосовує планування маніпуляцій до комп'ютерної анімації, розглядаючи планувальник шляху, який обчислює траєкторії без зіткнень для кількох взаємодіючих рук для маніпулювання рухомим об'єктом між двома конфігураціями. Алгоритм зворотної кінематики використовується планувальником для генерації реалістичних рухів рук людини, коли вони маніпулюють об'єктами. Алгоритм зворотної кінематики є специфічним для людських рук і базується на нейрофізіологічних дослідженнях. Динаміка і м'язові моделі не розглядаються.

В [13] представляють систему захоплення, яка дозволяє як автоматичні захоплення, так і захоплення за вибором аніматора. Завдання хапання

розбивається на три окремі фази: фаза ініціалізації завдання, де цільовий об'єкт класифікується як єдиний примітив для визначення стратегії хапання; фаза наближення до цілі, на якій визначається остаточне захоплення; фаза виконання захоплення, коли рука змикається навколо предмета. Об'єкти апроксимуються простими примітивами.

Елемент [14] використовує розроблені вручну алгоритми керування для виконання бажаних маневрів, таких як біг і їзда на велосипеді.

Фізична інтуїція щодо поведінки, спостереження за людьми, які виконують визначені завдання, і біомеханічні дані використовуються для встановлення цих алгоритмів керування. Вторинний рух додається до анімації за допомогою симуляції пружини маси тканини, що рухається твердим тілом імітованих людей. Динамічне моделювання використовується для обчислення рухів поведінки. Обмеження також використовуються для моделювання точок контакту між ногами та землею, а також ногами та колесами велосипеда.

Ходжінс та ін. зауважує, що використовуються ручні контролери, оскільки автоматизовані методи ще не доступні для пошуку рішень для систем з кількістю контрольованих ступенів свободи, яка була б необхідною для імітації правдоподібного людського тіла. Крім того, такі методи зазвичай не дають результатів, які виглядають природними в сенсі схожості на рух біологічної системи.

Незважаючи на високоякісні та реалістичні результати, які часто дають ці методи, вони не дають загальних рішень щодо руху. Реалізація нової поведінки зазвичай вимагає додаткової роботи з боку аніматора. Було розроблено ряд інших методів, щоб спробувати автоматично створити рух для нової поведінки.

У [15] описують три основні фази планування завдань, як моделювання світу, специфікація завдання та генерація коду. Обговорюються проблеми захоплення, зосереджуючись на «робочих просторах». Це об'єми простору, в які може досягти

кінцевий виконавець актора. В роботі [16] обговорюють повністю автоматичну процедуру руху рукою, включаючи використання двох рук, коли це необхідно. Для управління всіма артикуляціями руки використовується як пряма, так і зворотна кінематика.

3.6.2.2 Методи на основі обмежень

Автори в [17] і [18] вперше ввели просторово-часові обмеження. Аніматор визначає рух персонажів, спосіб виконання руху, структуру персонажів і фізичні ресурси, доступні для виконання руху. Обмеження на весь рух розглядаються одночасно, щоб знайти оптимальні рухи. Це, по суті, розглядає проблему автоматичної генерації руху як задачу оптимізації траєкторії. Хоча просторово-часові обмеження продемонстрували деякі цікаві результати, основною проблемою є те, що наразі вони обмежені простими системами через розмір і складність необхідних обчислень. Крім того, оптимальні фізично правильні рухи не завжди потрібні для реалістичного руху персонажа [19].

В [20] певним чином покращує корисність просторово-часових обмежень, згаданих [17]. Вони описують використання техніки ієрархічних просторово-часових обмежень для створення фізично заснованого та цілеспрямованого руху шарнірних фігур. Метод ієрархічного просторово-часового обмеження — це переформулювання функції через час узагальнених ступенів свободи в ієрархічному вейвлет-представленні. Проблеми, які виникають під час використання обмеження простору-часу через надмірну складність артикульованої фігури, вирішуються за допомогою ідеї створення просторово-часових вікон, як показано на (рис. 3.19).

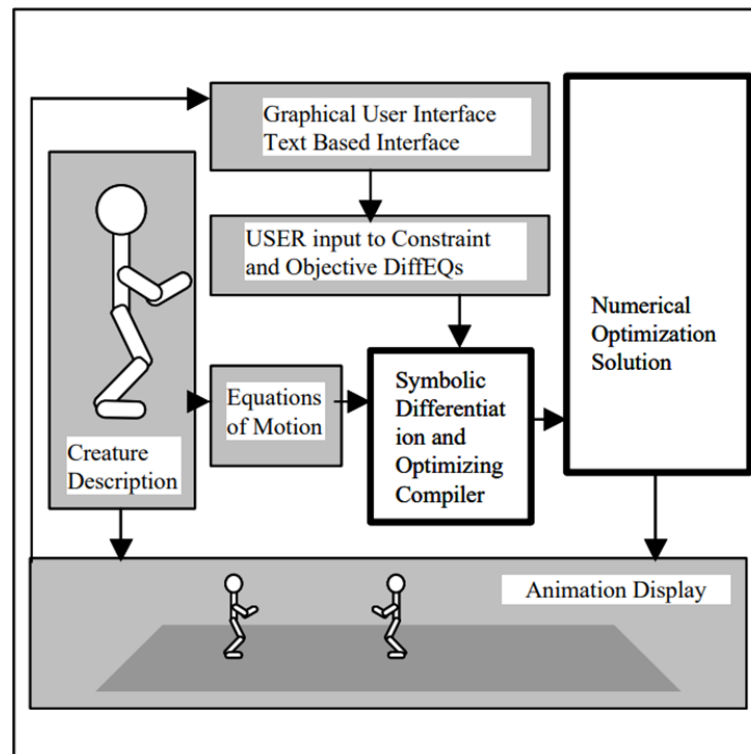


Рисунок 3.19 – Ієрархічна система просторово-часових обмежень, запропонована [20]

В [21] використовує комбінацію просторово-часових обмежень і зворотної кінематики для створення плавних і динамічних переходів між сегментами руху людського тіла. Цей метод дозволяє користувачам розбивати дані про рух на більш дрібні частини та збирати їх у нові, більш складні рухи. Це працює шляхом створення високоякісних рухів (відомих як «основні рухи») за допомогою техніки захоплення руху та створення переходів між цими рухами, щоб їх можна було об'єднати в наступні анімації. Поєднання цих обговорених методів і [20] може зробити просторово-часові обмеження життєздатним варіантом для створення рухів для обмежених частин складної шарнірної конструкції.

Розробки [22] представляють систему моделювання, засновану на динамічних обмеженнях. Моделі будуються з колекцій примітивних елементів із заздалегідь визначеними геометричними обмеженнями; сили обмежень притягують компоненти моделі до належної конфігурації, а елементи рухаються,

щоб задовольнити обмеження. Рух кожного твердого тіла обчислюється за ефектами інерції, сил і моментів, що діють на тіло. Інверсна динаміка використовується для визначення сил, які призведуть до певної поведінки, зберігаючи існуючі обмеження.

3.6.2.3 Контроль руху

Деякі альтернативні підходи до тих, що обговорювалися раніше, базуються на спробах знайти алгоритми керування для моделей, а не на бажаних траєкторіях руху.

В [23] пропонують алгоритм глобального пошуку, який може з нуля генерувати декілька траєкторій для просторово-часових обмежень для 2D артикульованих фігур. Ці стратегії пошуку спираються на метод кодування траєкторій як поведінки, а потім на генетичний алгоритм пошуку для вибору параметрів поведінки.

В [24] детально описують метод, за допомогою якого користувач визначає конфігурацію механічної системи, яка була доповнена датчиками та приводами. Мережа, керована датчиками (S.A.N.), складається із зважених з'єднань між датчиками та виконавчими механізмами. Відповідні S.A.N. вибираються та вдосконалюються за допомогою стохастичної процедури. Цей процес забезпечує автоматичний засіб виявлення багатьох можливих способів пересування для певного об'єкта. S.A.N.s особливо вправні в контролі над істотами, які гарантують нелінійні та негладкі системи.

В [25] і [26] також обговорюють підхід, який додає керування замкнутим циклом до рухів відкритого циклу для керування імітацією ходьби (ілюстровано на рис. 3.20).

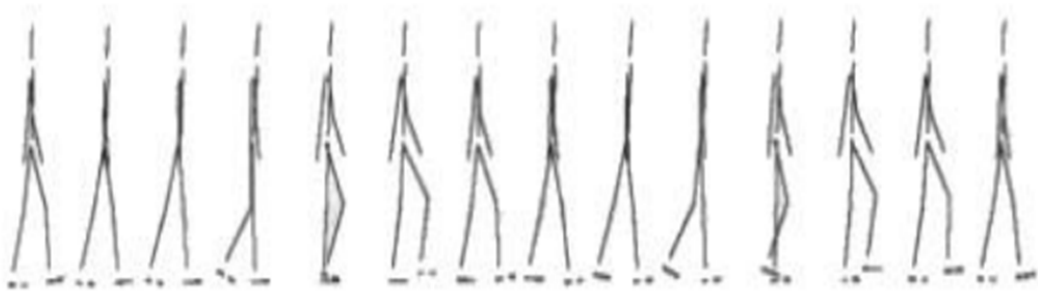


Рисунок 3.20 – Ілюстрація ходьби з використанням контролю обмеження циклу [25]

Аніматор забезпечує керування відкритим циклом, розглядаючи ходьбу як повторюваний рух заводної іграшки. Контроль граничного циклу потім використовується для автоматизації розрахунку коригувань керування, необхідних для підтримки балансу. Напрямок і швидкість ходьби тримаються під контролем аніматора. У документі відзначаються два спрощення: лінійна прогностична модель, яка може бути застосована до таких рухів, як ходьба, незважаючи на їх нелінійність, і введення скороченого вектора стану, який відображає зміни в русі більшої частини маси тіла. Контролери ходу містять низку параметрів, які потрібно налаштувати вручну.

В роботі [27] використовують генетичні програми, щоб дозволити аніматору/користувачу знайти/розвинути відповідну програму контролера для досягнення агентами своїх цілей. Ця робота пов'язана з роботою Witkin і Kass щодо просторово-часових обмежень і особливо з роботою Karl Sims [28], який використав подібний підхід для розробки процедурних текстур.

В джерелі [6] поєднують захоплення руху з динамічним моделюванням з метою створення реалістичного руху, схожого на людину. Система використовує дані захоплення руху як вхідні дані для контролера відстеження для динамічного моделювання. Ця система використовує пряму динаміку та алгоритми керування,

а не інверсну динаміку та мінімізацію крутного моменту. На рисунку 3.21 наведено системне розташування запропонованої системи.

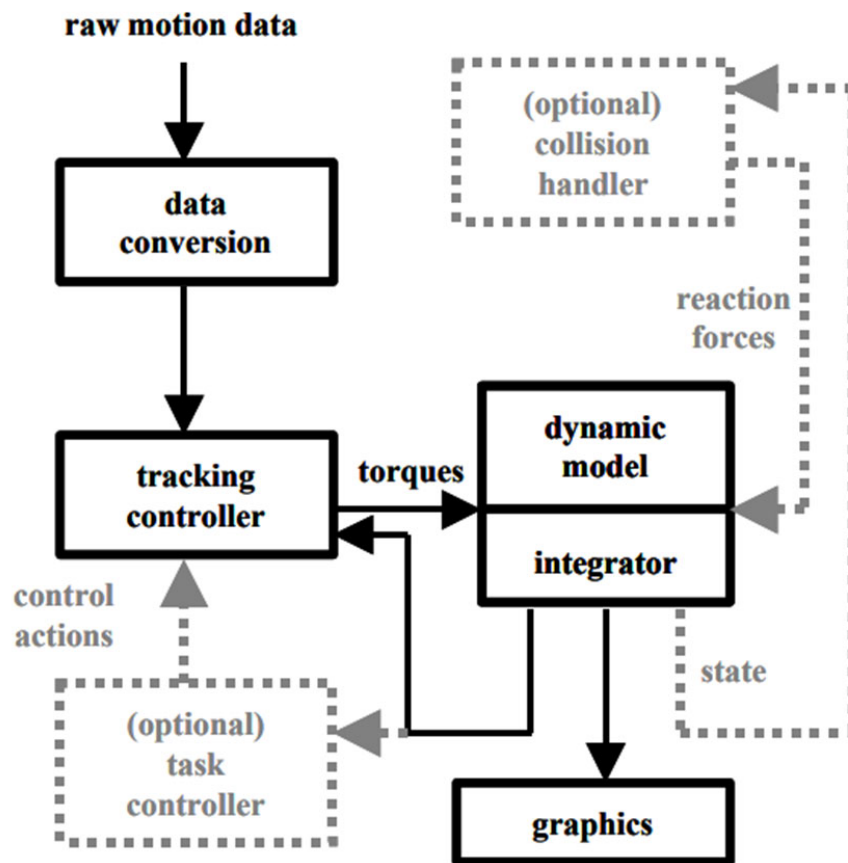


Рисунок 3.21 – Схема системи відстеження для системи, запропонована [6]

Необроблені дані руху перетворюються на безперервні кути з'єднань і використовуються контролером відстеження. Контролер стеження використовує ці дані для обчислення крутних моментів, які потім інтегруються для генерування руху. Для досягнення більш складної поведінки до цієї системи можна додати додатковий контролер завдань і обробник зіткнень.

3.6.3 Посилення руху

Хоча існує багато способів генерувати рух персонажа, більшість методів мають дві спільні проблеми. По-перше, через математичний характер

використовуваних методів часто отримують шаблонні рухи; циклічні рухи виглядають робототехнічними за своєю природою і не містять людських «недоліків», необхідних для передачі реалістичного руху будь-якому глядачеві. Для середньостатистичної людини немає двох абсолютно однакових кроків. Система зору людини є, мабуть, найбільш критичним спостерігачем людського руху, оскільки вона добре обізнана у спостереженні за нормальними повсякденними рухами людини.

Друга проблема – це швидкість. Для генерації динамічного руху необхідно виконати складні та трудомісткі обчислення.

Швидкість стає особливо критичною проблемою під час імітації натовпу віртуальних персонажів.

3.6.3.1 Шум

Джерело [29] використовує принципи процесуального синтезу текстур для створення тонких рухів людини, таких як перенесення ваги та метушня під час стояння. Ритмічні та стохастичні шумові функції визначають змінні в часі параметри, які керують персонажами.

Обчислення динаміки та вирішувачів обмежень уникають, оскільки потрібно передати лише «текстуру» руху. Вирази налаштовані на вміст псевдовипадкового шуму. Основний рух кожного суглоба є синусоїдальним із додатковим шумом, який використовується для запобігання повторюваності руху.

В [30] вводять природну мінливість у циклічну анімацію людського руху. Використовуються шумові функції, які базуються на біомеханічних міркуваннях. Вони застосовуються до базового бігу, створеного за допомогою даних захоплення руху або динамічного моделювання, і вводять природні збурення. Тести, проведені на симуляції чоловічого бігу, погано узагальнили симуляцію

жіночого бігу. Було зроблено висновок, що хоча деякий шум є кращим в анімації, досягнення найбільш бажаного рівня шуму для конкретної анімації не є тривіальним питанням.

3.6.3.2 Рівень деталізації

Оскільки методи динамічного моделювання за своєю суттю дорогі, було досліджено низку методів для зменшення обчислювальних витрат, пов'язаних з анімацією в реальному часі. Як згадувалося, ці прийоми є критичними під час симуляції великої кількості персонажів в одній сцені.

В [31] досліджують методи зменшення обчислювальних витрат на симуляцію шляхом використання менш точного моделювання для персонажів, коли вони менш важливі для глядача або для дії у віртуальному світі. Динаміка твердого тіла була обрана як найвищий рівень деталізації, потім йшло моделювання точкової маси з кінематичними з'єднаннями і, нарешті, найнижчим рівнем деталізації було моделювання точкової маси без кінематичного руху ніг.

Використовуючи ці методи, можна досягти кращої продуктивності в реальному часі для більшої групи істот, ніж це було б можливо, якщо б кожен істоту моделювали динамічно. [31] окреслюють два критерії рівня деталей моделювання: по-перше, результат моделювання має бути по суті незмінним. По-друге, сприйняття руху глядачем має бути однаковим. Зазначається, що основною проблемою під час випробувань було виконання першого критерію.

Робота [32] пропонує рівні деталізації анімації відповідно до частоти обчислень. Кожен рівень деталізації анімації має певний інтервал частоти кадрів. Високочастотна модель є «класичною» механічною моделлю, тоді як нейронна мережа використовується для низьких частот кадрів.

Стаття [33] вивчає конвеєр підмоделей. Кожна підмодель виконує задане завдання в процесі анімації: наприклад, анімація тіла або ніг. Кілька методів інкапсульовано для певного завдання з різними витратами на обчислення. Найбільш відповідний метод анімації вибирається за двома критеріями: складність середовища та відстань до камери. Процес полегшує переходи між одним рівнем та іншим, одночасно забезпечуючи узгодженість між рухами, які фіксуються різними рівнями. Перемикання з одного рівня на інший складається з додавання модуля або отримання модуля з цього конвеєра. Як зауважили [31], складною частиною рівнів деталізації анімації є забезпечення того, щоб різні моделі створювали узгоджені результати.

ВИСНОВКИ ДО 3 РОЗДІЛУ

Згідно з третім розділом магістерської дисертації розглянуто таке питання як ригінг персонажу за допомогою інверсної кінематики, математичний опис роботи інверсної кінематики в 2D та 3D просторі. Також були описані всі процеси анімації, такі як генерація руху, генерація параметрів, методи на основі обмежень, контроль руху, посилення руху, шум та рівень деталізації.

4 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

4.1 Опис проекту, призначення

В якості ідеї для створення стартап-проєкту пропонується створення автоматичного ригінгу персонажів спільно з інструкторами з різних напрямків, від математики до гімнастики. Такий підхід дозволить дітям отримувати нові навички та знання від цікавих 3D персонажів. Робоча назва проєкту пропонується “Дізнавайко”. Передбачається що буде робитись автоматичний ригінг моделі, яка потім буде накладатись на 3D персонажа та створювати навчальний продукт.

4.1.1 Опис існуючих рішень та їх обмеження. Конкуренти

Існуючі програмні рішення дозволяють реалізувати задум проєкту, створення ригінгу моделі займатиме лише половину усієї роботи, також потрібно буде знаходити викладачів та інструкторів, і вже під них налаштовувати анімацію персонажа. Рух губ персонажа, його міміка та інше.

Потенційний ринок для даного стартапу є відеохостинг YouTube, також можливе розповсюдження контенту на платформу TikTok, Facebook, Twitter та Netflix. Для повноцінного заробітку на курсі, будуть створюватись приватні курси для власного сайту FindOuter.com.

Потенційні конкуренти представлені на YouTube каналі під назвою savannahXYZ, але це навчальний контент тільки стосовно моделювання.

Аналіз потенційних ринків та конкурентів в даному сегменті послуг визначається наступними програмами: Blender, Autodesk Maya та 3ds Max.

4.1.2 Розв’язок проблеми

Навчальний проєкт “Дізнавайко” або “FindOuter” визначається наступними особливостями:

- Унікальний канал на відеохостингу YouTube;
- Створений власний сайт за адресою FindOuter.com;
- Постійне створення нового контенту;
- Пошук нових інструкторів та матеріалу;
- Укладання угод з компаніями на використання відомих персонажів для залучення більшої аудиторії;
- Поглиблений платний курс на власному сайті;
- Технічна підтримка з боку розробників;

Таким чином, на основі існуючих аналогів запропонований проєкт має потенціал до розвитку, є частково безкоштовним та гнучким до розширення контенту.

4.2 Ринок продукту

Проєкт в першу чергу орієнтований на дитячу та шкільну аудиторію. Сектор потенціальних споживачів – населення, переважно віком від 4 до 15 років, проте надалі планується розширення та поглиблення матеріалів на більш дорослу аудиторію. Можливий ринок продукту – платформи для відеоконтенту.

4.2.1 Бізнес модель “Канвас”. Можливі прибутки

1. Навчальний проєкт “Дізнавайко” або “FindOuter” спрямований на сегмент, який стосується побутових споживачів.

2. Ключовими цінностями запропонованого навчального проєкту є надання, необхідних в подальшому житті людини знань, для молодого покоління, та покращення наявних знань та навичок в дорослого покоління.
3. Каналами розповсюдження навчального проєкту можна вважати Інтернет-площадки, тобто платформи, для розповсюдження відеоконтенту такі як YouTube, Facebook та TikTok.
4. Ідея розвитку навчального проєкту передбачає технічну підтримку з боку розробників, розширення та покращення контенту що буде надаватись споживачу.
5. Можливими прибутками до запропонованого навчального проєкту є реклама та придбання поглиблених курсів від потенційних користувачів.
6. Першочерговими заходами з реалізації та розповсюдження проєкту можна визначити такі: створення каналу “FindOuter” на YouTube та на інших Інтернет-платформах. Пошук інструкторів, створення ригу та анімації 3D персонажа, створення відеокурсу з обраного напрямку.
7. Ключовими партнерами програмного комплексу можуть стати платформи що надають спеціалістів з різних напрямків навчання.
8. За структурою витрат з боку розробників можна вважати на початковому етапі – оплата роботи інструкторів, та ліцензія програмного забезпечення. Крім цього, можливі витрати можуть бути на рекламу та популяризацію навчального проєкту на ринку відео-послуг.

4.3 Маркетинг. Сильні та слабкі сторони продукту

Передумовою до появи та розповсюдження навчального проєкту можна вважати наступні заходи: реєстрація на відео-платформах з метою популяризації

продукту. Можливості до розвитку комплексу – пошук висококваліфікованих викладачів та покращення навичок в створенні та зйомці відеоматеріалів.

Разом з тим, загрозами навчального проєкту можна вважати незацікавленість молодого покоління до покращення власних знань.

Сильні сторони навчального проєкту: спеціалісти в сфері моделювання, ригінгу та анімації, висококваліфіковані викладачі та інструктори, цікава та незвична програма з подачі матеріалу.

Слабкі сторони навчального проєкту: відсутність на початку масової реклами проєкту, низька зацікавленість молодого покоління до навчання.

Перелічені слабкі сторони визначають низьку ймовірність загрози та закриття проєкту через те, що на ринку подібного продукту майже не існує.

4.4 Прогнозування розвитку проєкту

Старт проєкту заплановано на січень 2023 року. Час на просування проєкту на ринку визначено на рівні року, перші півроку заплановано пошук викладачів та створення навчальної програми. Клієнтська база користувачів – на початку україномовний сегмент Інтернету, потім розширення на англомовний сегмент та розширення вікових рамок проєкту. Збільшення бази підписників на рівні 10000 в перший рік розвитку проєкту. Команда проєкту складається з 4 осіб, де є розробник та автор проєкту, програміст, маркетолог та рекрутер.

Лояльність клієнтів запланована на створенні цікавого навчального контенту, наявності зворотного зв'язку, та живого спілкування з розробниками шляхом засобів Інтернет-комунікацій.

4.5 Ціннісна пропозиція

На даний момент навчальний проєкт є на стадії планування. Для створення продукту необхідно 50 тис. грн., для підтримки проєкту необхідно 20 тис. грн. У цю суму входить робота співробітників, а також наймання викладачів та створення програми. На початковому рівні кошти на утримання продукту передбачаються з рекламних надходжень, та приватного курсу.

4.6 План реалізації стартапу

Місією стартапу є надання необхідних знань молодому поколінню. При цьому на перший план виходить проста та цікава подача, умовна безкоштовність на відеоплатформах.

План реалізації стартапу:

- створення завершеного продукту у вигляді першого відеокурсу на платформі YouTube;
- створення реклами навчального проєкту, реєстрація на спеціальних площадках, таких як Instagram, TikTok, Twitter та Facebook;
- розвиток власного сайту, для додавання туди платного контенту;

ВИСНОВКИ ДО 4 РОЗДІЛУ

Запропоновано в якості стартапу навчальний проєкт “ Дізнавайко” або “FindOuter” який надає необхідні знання поточному користувачеві в незвичний та цікавий спосіб. Визначено основні переваги проєкту у порівнянні з конкурентами та надано положення, які дозволяють оцінити сильні та слабкі сторони проєкту. Додатково сформульовано бізнес модель та окреслено шляхи розвитку проєкту.

Представлена стратегія дозволяє вже в 2023 році випустити проєкт на ринок споживання України. Перевагою запропонованого рішення є умовна безкоштовність, проста, незвична та цікава подача матеріалу.

ВИСНОВКИ

В результаті виконання магістерської дисертації, в першому розділі було розглянуто питання персонажної анімації та її застосування в кіно, а саме було досліджене поняття моделі персонажу, також були розглянуті принципи створення анімації персонажів, було розглянуто програми 3D-моделювання та створення в них персонажів. Було наведено спеціальні прийоми в персонажній анімації, а саме: ретаргейтінг, інверсна кінематика та анімовані текстури.

В другому розділі розглядалось питання методів та особливостей персонажної анімації, розглядались особливості врахування міміки персонажу, методи скелетної анімації та її різновиди: сегментна анімація, анімація без вагів та еталонні форми в персонажній анімації. Були розглянуті спеціальні алгоритми персонажної анімації, такі як: Pose Space Deformation, Алгоритм П. Слоана, Алгоритм, заснований на скануванні та Метод П. Края.

В третьому розділі розглядався процес ригінгу персонажа, регулювання кінематики руху та подальше прогнозування анімації персонажа. В четвертому розділі представляється розроблений стартап-проект, який базується на ідеї створення навчального продукту для дітей з застосуванням технології автоматичного ригінгу та анімації 3D персонажів.

ПЕРЕЛІК ПОСИЛАНЬ

1. J.K. Hodgins, J.F. O'Brien "Computer Animation", Encyclopaedia of Computer Science, 1998.
2. J. Lasseter, "Principles of Traditional Animation Applied to 3D Computer Animation", Computer Graphics, 24(4). Proc. SIGGRAPH 87, 1987.
3. A. Witkin, P. Zoran, "Motion Warping", SIGGRAPH'95, pp. 105-108, 1995.
4. J. Zhao, N. Badler, "Inverse Kinematics Positioning Using Nonlinear Programming for Highly Articulated Figures", ACM Transactions on Graphics, 13(4), October 1994.
5. B. Delaney, "The Mystery of Motion Capture", IEEE CGA, 14-19, September, 1998.
6. V.B. Zordan, J.K. Hodgins, "Tracking and Modifying Upper-body Human Motion Data with Dynamic Simulation", 1999.
7. F. Multon, L. France, M-Paule Cani Gascuel, Gilles Debunne, "Computer Animation of Human Walking: A Survey", June 1998.
8. N.I. Badler, C.B. Philips, B.L. Webber, "Simulating Humans", Oxford University Press, 1993.
9. . Bruderlin, T.W. Calvert, "GoalDirected, Dynamic Animation of Human Walking", Proc. SIGGRAPH'89, pp233-242.
10. D. Pai, "Programming Anthropoid Walking: Control and Simulation", Cornell Computer Science Tech Report TR 90-1178.
11. P. Lee, S. Wei, J. Zhao and N.I. Badler, "Strength Guided Motion", Computer Graphics, Vol. 24, 1990.
12. Y. Koga, K. Kondo, J. Kuffner, J.C. Latombe, "Planning Motions with Intentions", Proc. SIGGRAPH'94, 1994.

13. H. Rijpkema, M. Girard, "Computer Animation of Knowledge-based Grasping", Proc. SIGGRAPH'91, Computer Graphics, Vol. 25, No.4, 1991, pp339-348.
14. J.K. Hodgins, W.L. Wooten, D.C. Brogan, J.F. O' Brien, "Animating Human Athletics", Proc. SIGGRAPH'95, 1995.
15. D. Thalmann, N.M. Thalmann, "Finite Elements in Task-Level Animation", FEM95-J, Switzerland, 1995.
16. R. Mas, D. Thalmann, "A Hand Control and Automatic Grasping System for Synthetic Actors", Proc. Eurographics'94, Oslo, Norway, 1994.
17. A. Witkin, M. Kass, "Spacetime Constraints", Proc. SIGGRAPH'88, 159-168, 1988.
18. LS. Brotman, A. Netravali, "Motion Interpolation by Optimal Control", Computer Graphics, 22 (4), Proc. SIGGRAPH '88, 1988.
19. M. Gleicher, P. Litwinowicz, "Constraint-Based Motion Adaption", Advanced Technology Group, Apple TR 96-153, June 1996.
20. Z. Liu, S.J. Gortler, M.F. Cohen, "Hierarchical Spacetime Control", Proc. SIGGRAPH'94, 1994.
21. C. Rose, B. Guenter, B. Bodenheimer, M.F. Cohen, "Efficient Generation of Motion Transitions using Space-time Constraints", Proc. SIGGRAPH'96, 1996.
22. R. Barzel, A.H. Barr, "A Modelling System Based on Dynamic Constraints", Computer Graphics, Vol. 22, No. 4, 1988, pp179 – 188.
23. J.T. Ngo, J. Marks, "Spacetime Constraints Revisited", Proc. SIGGRAPH'93, 1993.
24. M.V.D. Panne, E. Fiume, "Sensor Actuator Networks", Proc. SIGGRAPH'93, 1993.
25. J. Laszlo, M.V.D. Panne, E. Fiume, "Limit Cycle Control And Its Application To The Animation Of Balancing And Walking", 1996.

26. J.F. Laszlo, M.V.D. Panne, E. Fiume, "Control of Physically-based Simulated Walking", Proc. IMAGINA '97 Monaco, February 19-21, 1997, Monaco, 231-241.
27. L. Gritz, J.K. Hahn, "Genetic Programming for Articulated Figure Motion", 1995.
28. Karl Simms, "Artificial Evolution for Computer Graphics", Computer Graphics, 24(4), 405-13, Proc. SIGGRAPH 1991.
29. K. Perlin, "Real-time Responsive Animation with Personality", IEEE Transactions on Visualization and Computer Graphics, Vol. 1, No. 1, March 1995.
30. B. Bodenheimer, A.V. Shleyfman, J.K. Hodgins, "Effects of Noise on the Perception of Animated Human Running", EG CAS 99, 1999.
31. D.A. Carlson, J.K. Hodgins, "Simulation Levels of Detail for Real-time Animation", Proceedings of Graphics Interface '97, pp 1-8, 1997.
32. S. Chenney, J. Ichnowski and D. Forsith, "Dynamics Modelling and Culling", IEEE Computer Graphics and Applications, pp. 79-87, March 1999.
33. R. Cozot, F. Multon, B. Valton, B. Arnaldi, "Animation Levels of Detail Design for Real-time Virtual Humans", EG CAS 99, 1999.
34. P. Bergeron Controlling facial expressions and body movements in the computer-generated animated short 'Tony De Peltrie' : SIGGRAPH '85: ACM SIGGRAPH Tutorial Notes, 1985.
35. Lewis, J. P. Pose space deformation: a unified approach to shape interpolation and skeleton driven deformation // Proceedings of the 27th annual conference on Computer graphics and interactive techniques, 2000. P. 165-172.
36. Catmull, E. E. A system for computer generated movies // Proc. ACM Annual Conf. 1972. P. 422-431.
37. Burtnyk, N. Interactive skeleton techniques for enhancing motion dynamics in key frame animation // Communications of the ACM, 1972. V. 19. P. 564-569.

38. R. Woodland Filling the Gaps — Advanced Animation Using Stitching and Skinning : Course Technology Cengage Learning, 2008. 549 p.
39. Magnenat-Thalmann, N. Joint-dependent local deformations for hand animation and object grasping // Proceedings on Graphics interface, 1988. P. 26-33.
40. Lander, J. Skin them bones: Game programming for the web generation // Game Developer Magazine, 1998. P. 11-16.

ДОДАТОК А

SUMMARY

One of the main stages in the development of 3D video games and animated films is the actual animation of three-dimensional models. A three-dimensional model is usually a polygonal three-dimensional surface, the animation of which in the final stage is reduced to transformations (movements) of the vertices that form this surface. Model animation can be specified in a variety of ways, including Motion Capture methods, modeling the physical behavior of objects, real human anatomy, and more. However, skeletal animation methods have become the most widespread.

A character model is a three-dimensional polygon mesh or shell, which is a collection of vertices, edges and faces that determine the shape of the displayed polygonal object. Information about its coordinates, normal vectors and coordinates of superimposed textures - bitmap images used to specify color, relief illusion and model detailing is associated with the vertex.

The term "base form" is used to describe an undeformed character model. The base position of the vertex is the position of the vertex at the base of the model. Key shape refers to the shape given to the model at the key frame of the animation. Animation means the transformation of the model shell from the initial position to the final position (from one key shape to another). The shape of the model in each intermediate frame is interpolated by key shapes. At the same time, discrete, linear or non-linear interpolation (for example, Bezier curves) can be used.

There are methods of animating characters or objects that make their movements precise and perfectly natural. The possibility of using one or more of these methods should be considered before proceeding with frame-by-frame animation or the method of filling gaps with movement or shape.

One of the most important methods is to show the external influence of the environment on the object, including the force of gravity, other objects, etc. So, if the bat hits the ball, it flattens a little there and at the moment where and when it hits the bat.

The following method is useful for animating facial expressions. If, for example, the animation of talking characters is created with lip sync, then it is appropriate to show the effect of gravity and the stretching of the faces of the characters.

When the ball flies through the air, it stretches a little, as shown in Fig. In character animation, this effect is often exaggerated. And thanks to blurring, the effect of object movement is achieved (in particular, with the help of motion blur filters, which are often used in Photoshop and Fireworks). The stronger the object is blurred, the more accelerated is meant its movement.

When creating an animation, you should try to imagine the movement you want to show. The main focus should, of course, be on the main movements or actions in the animation, to push secondary movements to the background. After all, if the main action (the so-called main animation) is clearly expressed, the animation will only benefit from this.

When an instance is animated with gap fill, it moves in a straight line from the start point (at the first keyframe) to the end point (at the second keyframe). But if the movement takes place in an arc, in this case it is possible to achieve greater realism than when moving in a straight line.

If you're creating facial animation, such as changing facial expressions or lip sync, you'll probably want to use shape blending or transform animation.

This kind of animation is usually created by an animator in a 3D modeling program and then imported into Unity. Its creation is a two-step process. First, the animator determines all possible and extreme grimaces of the face mesh, that is, the position and order of all vertices of the face mesh in extremes, and creates a shape key (Shape Key) or blend shape (Blend Shape) to record the state of the mesh in this pose. By recording a series of different grimaces, the animator creates facial animations in which intermediate positions are generated as a weighted mixture between the differences in different grimaces.

The monkey head mesh in Blender was set to various extreme grimaces. Each grimace is recorded as a shape key or a blend of shapes. The following screenshot shows the initial or basic grimace. Sometimes it is called a grimace of peace.

One representation of a character model in 3D graphics is a hierarchy of segments, which are undeformed surfaces; this means that all vertices of the segment are transformed in the same way during the animation. The segment hierarchy was the prototype for the skeleton hierarchy in the following methods. For the animation of segment models, it is enough to store for each segment information about the position of the segment vertices, the segment center, and the transformation matrix. Because of its simplicity, this method was heavily used in early video games. The main drawback of the method is that the results of its work do not look realistic. This is caused, among other things, by inevitable gaps in the places of articulation of the bodies of the hierarchy, which are not deformed, during animation.

The algorithms of the analyzed group use a set of reference forms-examples of models and corresponding configurations of the skeleton. These reference shapes can be obtained from a number of sources: they can be made by hand by a 3D modeling specialist, be the result of a 3D scan (Motion Capture technology), and can also be generated by applying algorithms that simulate human anatomy. Next, the transformation results obtained using the LBS algorithm are compared with the specified reference shapes and adjusted accordingly. The main disadvantage of this group of algorithms is the need to obtain and use a very large number of reference forms to obtain realistic results. For the correct operation of the methods, it is necessary to provide such a set of reference forms that will fully describe all possible movements of the character, that is, will be representative. By correcting the operation of the LBS algorithm, these methods are inferior to it in terms of speed.

Kinematics is the process of calculating the position in space for the end of an interconnected structure at given angles of rotation of all hinge points (Joints). This is

easily done if there is only one solution. Inverse kinematics does the opposite. For this end point, the required hinge angles are calculated in such a way that this end point can be reached. Difficulties may arise if there is a large or infinite number of solutions.

This process is extremely useful in robotics. For example: a robot arm reached out and picked up an object. If the program knows the location of the object in relation to the shoulder joint, it is enough to calculate the angles of rotation of the hinges to reach the object. Inverse kinematics is also useful in 3D games. For example, a dragon with a very long neck should realistically bend its neck and eat a player standing on the floor. Or the player wanted to pick up an object from the floor or press a button. The user will see on the screen how the player reaches out and touches the object, instead of waving his hand somewhere near the object.

When more than two hinges are used, it often happens that there are many solutions to the problem. However, some solutions are better than others. If the structure represents, for example, a hand, some solutions are more convenient, others are very unnatural. Often there is an optimal solution.

The algorithm does not immediately find a solution. This is not a simple function where you set a destination and get angles. It is an iterative algorithm, which makes the algorithm more realistic in applications, especially in games where you need objects to move smoothly.

The solution is to virtually apply force to the end of the chain and drag the end of the chain to its final position. A force acts on each joint and moves the structure slightly closer to the end point. So even if there is no solution, the chain will approach the endpoint as close as possible. Since the hinges are calculated separately, you can set different properties for them, such as stiffness, elasticity, etc. Each of the hinges can also have an optimal position that the algorithm will try to get as close as possible.

Traditionally, the art of animation took a lot of time with a pen and pencil, which was very dependent on the skill of the animators involved.

The art of computer animation has taken a huge step forward from these traditional techniques. Perhaps the first work to demonstrate the power of using computers for animation was Pixar's 1986 mini-masterpiece, Luxo Jr.

Luxo Jr. is a computer animation of a lamp playing with a ball. At the time, animation was done using the trivial keyframe method.

Since then, computer animation has developed by leaps and bounds. The realism of animated shapes no longer depends on the skill of the animator, and new techniques offer ways to animate shapes with very little animator intervention.

A few basic techniques have become prominent in recent years and are widely used in many different areas of character animation. The following can be considered parts of the "basic toolkit" that can be used to create powerful animation systems.

A more traditional method of animating an articulated figure in computer graphics involves defining each specific part at certain key locations in space, and then using some interpolation technique to animate between frames for the corresponding movement. This method of creating motion is known as keyframing, and manually entering poses is probably the easiest method to define motion. An animator must create real world movement by hand. Not only is this a tedious and time-consuming task [2], but it also relies on the skill of the animator to create realistic, convincing effects.