

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук

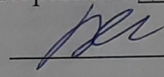
Бакалаврська дипломна робота на тему:
РОЗРОБКА ПРОГРАМНОГО МОДУЛЯ КОМП'ЮТЕРНОЇ ГРИ SPACE
DUST

Виконав: студент 4 курсу, групи 2КН-206
спеціальності 122 – Комп'ютерні науки

(цифр і назва напрямку підготовки, спеціальності)

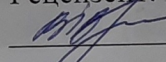
 Севастьянов І.В.
(прізвище та ініціали)

Керівник: д.т.н., доцент. каф. КН

 Денисюк В.О.
(прізвище та ініціали)

« 06 » 07 2024 р.

Рецензент: к.т.н., доцент., зав. каф. САІТ

 Варчук І. В.
(прізвище та ініціали)

« 06 » 07 2024 р.

Допущено до захисту

Зав. кафедри проф., д.т.н. Яровий А.А. 

« 06 » 07 2024 р.

Вінниця ВНТУ - 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних наук
Рівень вищої освіти перший (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 122 Комп'ютерні науки
Освітньо-професійна програма – Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
проф., д.т.н. Яровий А.А.
« 07 » 06 2024р.

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ
Севастьянову Івану Вадимовичу

1. Тема роботи: Програмний модуль комп'ютерної гри "Space Dust"
керівник роботи: Денисюк В.О., доцент
затверджені наказом вищого навчального закладу " н " 05 20 24 року № 80
2. Строк подання студентом роботи 27 05 2024
3. Вихідні дані до роботи :кількість гравців – 1 чол;кількість рівнів – 1;частота кадрів – 60 кадрів за секунду;розширення екрану – 1920x1080.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):аналіз предметної області, постановка задачі, аналіз і вибір програмних засобів для розробки, розробка програмного забезпечення, розробка та тестування гри, оцінка результатів роботи
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
діаграма основних компонентів архітектури програмного модулю відеогри, модель станів персонажа, діаграма процедури генерації для створення ландшафту, діаграма модульного дизайну у розробці ігор, загальний вигляд інтерфейсного вікна типу «головне меню» комп'ютерної гри, загальний вигляд інтерфейсного вікна типу «локація».

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Денисюк В.О., д.т.н., доцент., каф. КН		

7. Дата видачі завдання 27.05.2024

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської дипломної роботи	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз сучасної ігрової індустрії та класифікація комп'ютерних ігор	06.05.24	11.05.24	
2	Розробка архітектури програмного модулю відеогри	12.05.24	17.05.24	
3	Розробка методів створення ігрової механіки	18.05.24	19.05.24	
4	Розробка алгоритму функціонування програмного модулю комп'ютерної гри	20.05.24	21.05.24	
5	Програмна реалізація модулю комп'ютерної гри	22.05.24	25.05.24	
6	Тестування програмного модулю комп'ютерної гри	26.05.24	28.05.24	
7	Розробка інструкції користувача.	29.05.24	01.06.24	
8	Оформлення матеріалів до захисту БДР	02.06.24	06.06.24	

Студент  Севаст'янов І.В.
(підпис) (прізвище та ініціали)Керівник роботи  Денисюк В.О.
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Бакалаврська дипломна робота обсягом 86 сторінок формату А4 включає 25 рисунків і список використаних джерел з 15 найменувань.

Дана робота присвячена розробці та впровадженню програмного модуля для комп'ютерної гри «Space Dust». Основною метою є створення оптимізованої 3D графіки з зручним та інтуїтивно зрозумілим управлінням. У роботі розроблено структурну схему функціонування модуля, алгоритмічну

схему роботи програмного модуля та UML-діаграму алгоритму модуля гри у жанрі 3D раннера. Unity використовується для розробки основних компонентів гри (інтерфейс, анімації, дизайн тощо), а C# — для створення скриптів. Архітектура модуля побудована за підходом, який забезпечує розділення логіки, представлення та обробки даних.

Результатом роботи є функціональний та надійний програмний модуль, що дозволяє гравцю керувати персонажем на 3D платформі та взаємодіяти з навколишніми об'єктами.

Ключові слова: комп'ютерна гра, 3D раннер, комп'ютерна графіка.

ABSTRACT

The bachelor's thesis is 86 pages in size, includes 25 figures and a list of 15 references.

This work is devoted to the development and implementation of a software module for the computer game “Space Dust”. The main goal is to create optimized 3D graphics with convenient and intuitive control. The paper develops a structural diagram of the module's functioning, an algorithmic diagram of the program module, and a UML diagram of the algorithm of the game module in the genre of 3D runner. Unity is used to develop the main components of the game (interface, animations, design, etc.), and C# is used to create scripts. The architecture of the module is based on an approach that ensures the separation of logic, presentation, and data processing.

The result is a functional and reliable software module that allows the player to control the character on a 3D platform and interact with the surrounding objects.

Keywords: computer game, 3D runner, computer graphics

ЗМІСТ

ВСТУП.....	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	9
1.1 Аналіз та загальна характеристика комп'ютерних ігор.	9
1.2 Аналіз проблеми графічного дизайну у відеоіграх у жанрі 3D ранерах.	24
1.3 Аналіз сучасних відеоігор у жанрі 3D ранери	29
1.4 Висновок до розділу 1	34
2 РОЗРОБКА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ ВІДЕОГРИ «SPACE DUST».....	35
2.1 Розробка архітектури програмного модулю відеогри «Space Dust»	35
2.2 Розробка методів створення ігрової механіки	45
2.3 Розробка алгоритму функціонування програмного модулю комп'ютерної гри «Space Dust»	50
2.4 Висновок до розділу 2	52
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЮ КОМП'ЮТЕРНОЇ ГРИ «SPACE DUST»	54
3.1 Обґрунтування вибору рушія для програмної реалізації	54
3.2 Програмна реалізація модулю комп'ютерної гри "Space Dust"	58
3.3 Тестування програмного модулю комп'ютерної гри	64
3.4 Висновок до розділу 3	67
ВИСНОВКИ	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	69
ДОДАТОК А	71
Показники звіту подібності Unichек	71
ДОДАТОК Б	72
Інструкція користувача	72
ДОДАТОК В.....	74
Лістинг програми	74
ДОДАТОК Г	76
Графічний матеріали	76

ВСТУП

Актуальність досліджень. У сучасному світі мобільних ігор, що стрімко розвивається, жанр "ранер" (endless runner) займає особливе місце, надаючи гравцям динамічний та захоплюючий досвід. Його популярність обумовлена простотою освоєння, інтуїтивно зрозумілим керуванням та нескінченним геймплеєм, що спонукає до постійного покращення результатів.

Створення успішної гри в жанрі runner вимагає не лише креативного підходу до дизайну рівнів, персонажів та сюжету, але й глибокого розуміння технічних аспектів розробки. Одним з ключових елементів є програмний модуль, який забезпечує функціональність гри, взаємодію з користувачем та реалізацію ігрових механік. Він є своєрідним "серцем" гри, що відповідає за її логіку, фізику, штучний інтелект та інші важливі аспекти.

Актуальність даної роботи зумовлена постійним зростанням попиту на мобільні ігри-ранери, а також необхідністю збільшення швидкодії якісних та конкурентоспроможних ігрових продуктів. Покращення програмного модуля з використанням Unity є важливим кроком у цьому напрямку, оскільки цей двигун надає широкий спектр інструментів та можливостей для розробки мобільних ігор.

Метою дослідження є підвищення рівня оптимізації комп'ютерної графіки для комп'ютерної гри у жанрі 3D-ранерра, а також покращення зручності керування персонажа для взаємодії з навколишніми об'єктами.

Об'єкт дослідження: є процес оптимізації комп'ютерної 3D графіки.

Предмет дослідження: відеогра в жанрі 3D раннер.

Задачі дослідження:

1. Провести аналіз предметної області, проблеми з комп'ютерною графікою та додатків аналогів.
2. Розробка архітектури та методів комп'ютерної гри.

3. Розробка алгоритму функціонування програмного модуля комп'ютерної гри «Space Dust».
4. Розробка та тестування комп'ютерної гри
5. Оцінка результатів роботи

Практична цінність. Програмний модуль з оптимізацією графіки та керування персонажем у грі 3D-раннер "Space Dust" відповідатиме найвищим стандартам якості. Він забезпечить плавну роботу гри, високу продуктивність та інтуїтивно зрозумілий інтерфейс, що сприятиме залученню та утриманню гравців.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз та загальна характеристика комп'ютерних ігор.

Галузь відеоігор є однією з найдинамічніших і найприбутковіших у світі, з постійним зростанням, що перевищує 200 мільярдів доларів США у 2023 році. Цей ринок включає різні сегменти: консольні ігри (PlayStation, Xbox, Nintendo Switch), ПК-ігри (через платформи як Steam і Epic Games Store), мобільні ігри (App Store і Google Play) та хмарні ігри (Google Stadia, NVIDIA GeForce Now, Xbox Cloud Gaming).

Технологічні інновації значно впливають на розвиток галузі. Віртуальна і доповнена реальність (VR і AR) створюють більш занурюючі ігрові світи, а штучний інтелект (AI) покращує ігровий процес, роблячи його більш реалістичним. Блокчейн та NFT дозволяють створювати унікальні ігрові предмети з реальною вартістю, що змінює економіку ігор.

Відеоігри стають важливою частиною сучасної культури, впливаючи на кіно, літературу, музику та піднімаючи соціально важливі теми. Крім того, кіберспорт набирає популярності, залучаючи великі аудиторії та значні призові фонди. Популярні змагання з ігор, таких як League of Legends і Dota 2, перетворилися на масштабні події з мільйонними призовими фондами [4].

Однак, індустрія стикається з викликами. Серед них — проблема залежності від відеоігор, питання етики та цензури через насильство і контент для дорослих, а також висока конкуренція, де успіх залежить від інновацій та ефективного маркетингу.

Загалом, індустрія відеоігор продовжує стрімко розвиватися, інтегруючи нові технології та розширюючи вплив на світову економіку і культуру, хоча і стикається з певними викликами, які потребують вирішення.

Розвиток індустрії відеоігор значно впливає на різні аспекти суспільства. Вона створює нові робочі місця і сприяє розвитку суміжних

галузей, таких як кіберспорт, медіа, реклама та технології. Крім того, відеоігри все частіше використовуються в освіті та тренуваннях, оскільки вони забезпечують інтерактивний і захоплюючий спосіб навчання.

Інновації в галузі також стимулюють розвиток апаратного забезпечення. Високі вимоги сучасних ігор сприяють прогресу в розробці графічних процесорів (GPU), процесорів (CPU) та інших компонентів комп'ютерів і консолей. Це, в свою чергу, впливає на суміжні технологічні ринки, такі як віртуальна реальність, штучний інтелект і хмарні обчислення.

Соціальні мережі і платформи для стрімінгу, такі як Twitch і YouTube, також відіграють важливу роль у популяризації відеоігор. Геймери можуть транслювати свої ігрові сесії, спілкуватися з глядачами і створювати контент, що сприяє збільшенню аудиторії та впливає на продажі ігор. Інфлюенсери в цій сфері можуть мати значний вплив на смаки та вподобання гравців.

Незважаючи на економічні успіхи, галузь відеоігор стикається з проблемами регулювання і контролю. Уряди різних країн встановлюють правила щодо захисту прав споживачів, захисту неповнолітніх і боротьби з азартними іграми в відеоіграх. Це включає контроль за мікротранзакціями та лутбоксами, які можуть викликати залежність і фінансові проблеми у гравців.

Проблеми з етикою і соціальною відповідальністю також залишаються актуальними. Розробники ігор повинні враховувати вплив свого продукту на психічне здоров'я гравців, уникати надмірного насильства і дискримінаційного контенту. Крім того, вони мають забезпечувати безпеку і конфіденційність даних користувачів.

У майбутньому очікується подальше зростання і трансформація галузі відеоігор. Технології доповненої реальності (AR) та змішаної реальності (MR) відкриють нові можливості для ігрового досвіду. Розвиток 5G мереж сприятиме покращенню якості онлайн-ігор і хмарного геймінгу. Зростаючий інтерес до екологічної відповідальності може спонукати розробників створювати більш енергоефективні рішення.

Таким чином, індустрія відеоігор продовжує еволюціонувати, інтегруючи нові технології і відповідаючи на виклики сучасного світу. Її вплив на економіку, культуру і суспільство залишається значним, роблячи відеоігри важливою частиною нашого життя і майбутнього.

Зростання популярності відеоігор також призводить до розвитку нових форматів взаємодії з аудиторією. Одним із них є інтерактивні ігрові шоу, де глядачі можуть впливати на події в грі через стрімінгові платформи. Це поєднання ігор та медіа створює нові можливості для залучення користувачів і генерації доходів через рекламу та підписки.

Розвиток технологій також стимулює зростання мобільного геймінгу. Мобільні пристрої стають все потужнішими, що дозволяє розробникам створювати більш складні ігри з високою графікою та багатим ігровим процесом. Це, у свою чергу, робить мобільні ігри доступнішими і популярнішими серед широкої аудиторії, включаючи ті категорії гравців, які раніше не цікавилися традиційними іграми.

Крім того, індустрія відеоігор активно інтегрує елементи соціальної відповідальності. Все більше компаній починають звертати увагу на екологічну сталість, прагнучи зменшити вуглецевий слід своєї діяльності. Це включає оптимізацію серверних ферм для хмарного геймінгу та використання відновлюваних джерел енергії для живлення центрів обробки даних.

Відеоігри також стають важливим інструментом для соціальної інтеграції і підтримки спільнот. Вони дозволяють людям з різних куточків світу взаємодіяти і спілкуватися, створюючи глобальні спільноти гравців. Особливо це стало помітним під час пандемії COVID-19, коли відеоігри стали одним з основних способів соціалізації та розваги для багатьох людей [5].

Сучасні відеоігри також використовуються для терапії та реабілітації. Ігрова терапія показує позитивні результати в лікуванні різних психічних і фізичних розладів, включаючи депресію, тривожність та проблеми з руховими функціями. Ігри, що спеціально розроблені для цих цілей, можуть сприяти покращенню настрою і загального благополуччя пацієнтів.

З поглибленням інтеграції штучного інтелекту в ігрову індустрію, очікується, що ігри стануть ще більш адаптивними і персоналізованими. AI зможе аналізувати поведінку гравців і відповідним чином налаштовувати ігровий процес, роблячи його більш захоплюючим і відповідним до індивідуальних уподобань.

Індустрія відеоігор також залучає значні інвестиції, що сприяє її подальшому зростанню та розвитку. Великі корпорації, такі як Google, Amazon та Apple, активно вкладають кошти в розвиток геймінгових платформ і технологій, що підвищує конкуренцію і стимулює інновації.

У перспективі, відеоігри продовжуватимуть відігравати важливу роль у суспільстві, культурі та економіці. Вони розширюють межі можливого, об'єднуючи людей та створюючи нові форми взаємодії і розваг. Галузь відеоігор має всі шанси стати ще більш впливовою та значущою у найближчі роки, продовжуючи дивувати і надихати мільйони гравців по всьому світу.

Деякі з найбільш захоплюючих напрямків відеоігор включають розширену реальність (AR) та віртуальну реальність (VR). Ці технології дозволяють гравцям потрапляти в ігрові світи, які виглядають і відчуються дуже реалістично. Такі ігри не лише забавляють, а й можуть використовуватися для навчання, тренування та навіть лікування фобій.

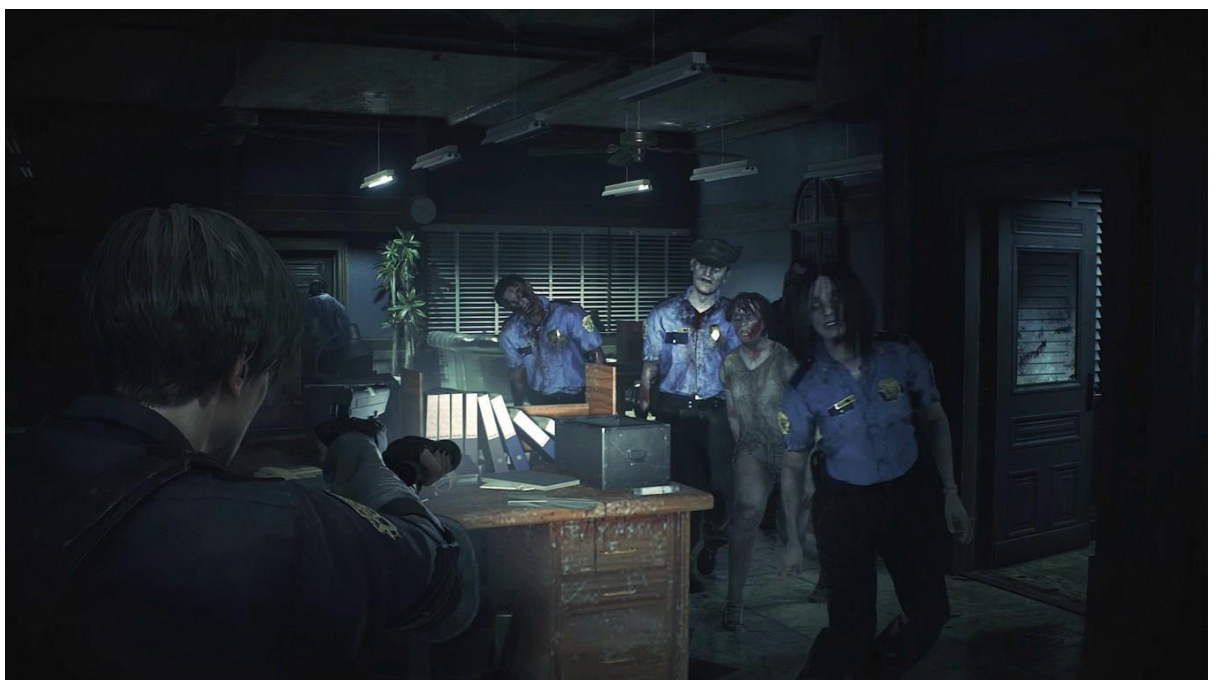
Також спостерігається зростаючий інтерес до ігор з використанням музики та аудіо-візуального мистецтва як головних елементів геймплею. Це відкриває нові можливості для творчості та експериментів у створенні ігрового досвіду, де аудіовізуальні аспекти стають не менш важливими, ніж геймплей та сюжет.

Крім того, зростає популярність ігор з акцентом на соціальну взаємодію та співпрацю. Кооперативні ігри, де гравці повинні працювати разом, щоб досягти спільних цілей, стають все популярнішими. Це може бути як у вигляді онлайн кооперації, так і локальних мультиплеєрних ігор, де гравці можуть грати на одному екрані.

Не можна також забувати про розвиток інді-ігор, які часто відзначаються оригінальними концепціями, незвичайним геймплеєм та глибоким сюжетом. Інді-розробники займаються створенням ігор, які часто виходять за рамки стандартних жанрів та використовують унікальні мистецькі підходи.

Загалом, відеоігри продовжують розвиватися та еволюціонувати, впливаючи на нашу культуру, спосіб розваг та соціальну взаємодію. Перед нами відкриті безліч можливостей у цій галузі, і ми можемо очікувати більше захоплюючих та інноваційних ігор у майбутньому.

1. Хоррор-ігри переносять нас у захопливий світ, де серце б'ється швидше, а кров пронизана адреналіном. Вони ефективно використовують різноманітні засоби, щоб викликати наші найглибші страхи та тривоги. Похмурі місця, хід моторошної музики, реалістичні звукові ефекти та гра світла й тіні створюють атмосферу напруження, яка тримає гравців в напруженому стані. Жахливі створіння, привиди, зомбі та інші вороги переслідують гравця, залишаючи відчуття небезпеки та безпорадності. Заплутані сюжети, таємничі події та несподівані повороти підсилюють емоційне напруження гри. Різкі звуки, скримери (раптові появи страшних



образів) та непередбачувані події спонукають гравців до переляку та викиду адреналіну [3].

Рисунок 1.1 – Зображення відеогри в жанрі Хоррор

Жанр ігор, спрямованих на викликання страху, не просто лякає, а й впливає на нашу психіку та емоції. Вони викликають страх перед невідомим, тривогу, відразу і навіть задоволення від адреналінового сплеску. Обмеженість ресурсів, слабкість персонажа, темрява та обмежена видимість, моторошна музика та реалістичні звукові ефекти посилюють відчуття страху та безпорадності. Розв'язування задач та дослідження оточення додають інтелектуальний виклик і змушують гравців бути уважними до деталей.

Ці ігри часто запозичують ідеї та образи з фільмів жахів, але також мають свої унікальні можливості. Інтерактивність дозволяє гравцям безпосередньо брати участь у подіях, що робить досвід більш особистим та емоційним. Візуальні та звукові ефекти занурюють гравців у світ гри, а нелінійний сюжет дозволяє впливати на його розвиток та кінцівку.

Віртуальна реальність (VR) відкриває нові горизонти для цього жанру, дозволяючи гравцям відчувати себе ще ближче до подій. VR-гарнітури створюють ефект присутності, що робить досвід ще більш реалістичним та страшним.

Якщо ви вирішили зануритися у світ таких ігор, оберіть гру відповідно до своїх уподобань, грайте в компанії друзів, робіть перерви, не грайте вночі та пам'ятайте, що це лише гра. Вони можуть стати захопливим та незабутнім досвідом, якщо підійти до них з розумінням та відповідальністю.

Цей жанр не стоїть на місці, постійно розвиваючись і експериментуючи з новими ідеями. Сучасні розробники прагнуть не лише налякати гравців, але й змусити їх замислитися над важливими темами, такими як людська природа, мораль, соціальні проблеми та екзистенційні питання.

Ігри цього типу можуть бути не лише джерелом розваг, але й інструментом для дослідження темних сторін людської психіки. Вони

дозволяють безпечно зіткнутися зі своїми страхами та тривогами, краще зрозуміти себе та навколишній світ.

Не варто забувати, що такі ігри мають вікові обмеження і не всі вони підходять для дітей та підлітків. Важливо враховувати індивідуальні особливості та психологічну стійкість гравця, щоб уникнути негативних наслідків.

Цей жанр – це унікальний вид розваг, який продовжує розвиватися та дивувати нас новими ідеями та підходами. Якщо ви готові до випробування своїх нервів та занурення у світ страху та напруги, то такі ігри чекають на вас. Але пам'ятайте, що найголовніше – це підходити до гри з розумінням та відповідальністю. – це отримувати задоволення від гри та не забувати про реальне життя.

2. Головоломки – це жанр відеоігор, який захоплює розум та випробовує нашу кмітливість. Вони пропонують гравцям різноманітні задачі, що вимагають логічного мислення, уважності до деталей та творчого підходу. Від простих логічних ігор до складних фізичних головоломок, цей жанр пропонує безліч можливостей для інтелектуального розвитку та розваги [4].



Рисунок 1.2 – Зображення відеогри в жанрі головоломки

Головоломки не лише розважають, але й приносять значну користь. Вони покращують пам'ять, увагу, концентрацію, логічне мислення та просторову уяву. Розв'язування таких задач допомагає зняти стрес і розслабитися, а успішне вирішення складних викликів підвищує самооцінку та приносить задоволення. Ці розваги доступні для людей будь-якого віку та рівня підготовки, від класичних логічних ігор, таких як Sudoku та Nonograms, до інноваційних фізичних випробувань, таких як Portal та The Witness.

Сучасні інтерактивні задачі вражають різноманітністю та оригінальністю, вони можуть переносити нас у фантастичні світи, запрошувати до розслідування таємниць або просто вимагати знайти вихід з лабіриту. Незалежно від тематики, ці ігри завжди пропонують цікавий і корисний досвід. Вони створюють спільноти, де гравці діляться знахідками, обговорюють рішення та допомагають один одному подолати складні випробування. Такі заняття також використовуються в освіті для навчання дітей і дорослих, допомагаючи розвивати логічне та критичне мислення, а також навички розв'язання проблем. Вони можуть бути інтегровані в різні навчальні дисципліни, роблячи процес навчання цікавішим та ефективнішим.

Сьогодні ці ігри доступні на різних платформах, від персональних комп'ютерів та консолей до мобільних пристроїв, що робить їх ще більш доступними та дозволяє насолоджуватися ними будь-де та будь-коли. Незалежно від того, чи ви новачок у світі головоломок, чи досвідчений гравець, завжди є щось нове та цікаве для відкриття. Цей вид розваг продовжує розвиватися, пропонуючи нові виклики та можливості для самовдосконалення.

3. Симулятори – це захопливий жанр відеоігор, який дозволяє нам зануритися у різноманітні професії та життєві ситуації, не виходячи з дому. Відчуйте себе пілотом літака, що злітає в небо, водієм вантажівки, який долає тисячі кілометрів, або фермером, який вирощує власні овочі та фрукти [5].



Рисунок 1.3 – Зображення гри у жанрі симулятор

Ігри цього жанру вражають своєю реалістичністю, відтворюючи фізику, графіку та звуки з високою точністю. Деякі з них вимагають від гравців реальних знань і навичок, таких як розуміння правил дорожнього руху або управління складними механізмами. Інші ж пропонують більш розслаблений геймплей, дозволяючи насолоджуватися процесом і експериментувати без страху помилитися.

Однією з привабливих особливостей багатьох таких ігор є відкритий світ. Він надає гравцям свободу досліджувати віртуальні простори, створювати власні історії та знаходити унікальні способи вирішення завдань. Можливість додавати модифікації дозволяє розширити ігровий досвід і налаштувати гру під свої уподобання.

Цей тип ігор користується великою популярністю серед гравців різного віку та інтересів. Вони не лише розважають, але й навчають, дозволяючи розвивати нові навички та знання в безпечному та контрольованому середовищі. Це не просто розваги, а можливість зануритися у захопливий світ професій та хобі, які завжди цікавили, але здавалися недосяжними. Відчуйте

адреналін від перегонів на високій швидкості, спокій від вирощування рослин чи задоволення від створення власного міста.

Кожна гра цього жанру – це унікальний досвід, який допомагає краще зрозуміти світ навколо нас та відкрити для себе нові захоплення. Вони можуть стати віддушиною від повсякденності, джерелом натхнення і навіть поштовхом до вибору майбутньої професії.

Такі ігри постійно розвиваються, пропонуючи гравцям все більш реалістичні та захопливі враження. Завдяки новим технологіям, таким як віртуальна реальність, ми можемо ще глибше зануритися у віртуальні світи та відчути себе справжніми професіоналами.

4. Екшн-ігри – це справжній вибух адреналіну та емоцій, де гравці поринають у світ динамічних пригод та випробувань. Відчуйте себе героєм, який сміливо кидається в бій, ухиляється від куль та перемагає ворогів за допомогою різноманітної зброї та бойових технік. Випробуйте швидкість та азарт перегонів, де кожна секунда має значення, а кожен поворот може стати вирішальним [6].



Рисунок 1.4 – Зображення відеогри в жанрі Екшн

Ці ігри пропонують безліч можливостей для самовираження та розвитку навичок. Відточуйте свою реакцію, точність та стратегічне мислення, борючись з ворогами у шутерах від першої особи, або демонструйте свою майстерність у рукопашному бою у файтингах. Досліджуйте відкриті світи, виконуйте захопливі місії та розкривайте таємниці, занурюючись у захопливі сюжети та історії.

Незалежно від того, чи ви віддаєте перевагу реалістичним військовим симуляторам, фантастичним пригодам у відкритому космосі або стилізованим боям з демонами, екшн-ігри завжди знайдуть чим вас здивувати та захопити. Відчуйте прилив адреналіну, насолоджуйтесь видовищними спецефектами та станьте частиною незабутніх пригод!

Ігри, сповнені динаміки та адреналіну, пропонують гравцям поринути у світ захопливих пригод і випробувань. Відчуйте себе героєм, який сміливо кидається в бій, ухиляється від куль та перемагає ворогів за допомогою різноманітної зброї та бойових технік. Випробуйте швидкість та азарт перегонів, де кожна секунда має значення, а кожен поворот може стати вирішальним.

Ці ігри пропонують безліч можливостей для самовираження та розвитку навичок. Відточуйте свою реакцію, точність та стратегічне мислення, борючись з ворогами у шутерах від першої особи, або демонструйте свою майстерність у рукопашному бою у файтингах. Досліджуйте відкриті світи, виконуйте захопливі місії та розкривайте таємниці, занурюючись у захопливі сюжети та історії.

Незалежно від того, чи ви віддаєте перевагу реалістичним військовим симуляторам, фантастичним пригодам у відкритому космосі або стилізованим боям з демонами, динамічні ігри завжди знайдуть чим вас здивувати та захопити. Відчуйте прилив адреналіну, насолоджуйтесь видовищними спецефектами та станьте частиною незабутніх пригод!

Цей жанр постійно розширює свої межі, впроваджуючи нові технології та механіки, щоб зробити геймплей ще більш захопливим. Сучасні ігри

вражають реалістичною графікою, інтуїтивно зрозумілим керуванням та глибоким зануренням у ігровий світ. Використання штучного інтелекту дозволяє створювати ворогів, які здатні адаптуватися до вашого стилю гри, роблячи кожне зіткнення унікальним та викликом.

Крім того, багато таких ігор мають багатокористувацькі режими, де можна змагатися з друзями або іншими гравцями з усього світу. Командні битви, кооперативні місії та змагання за першість у рейтингах додають новий рівень інтерактивності та соціальної взаємодії.

Віртуальна реальність (VR) і доповнена реальність (AR) відкривають нові можливості для цього жанру, дозволяючи гравцям буквально занурюватися у гру. VR-гарнітури та контролери руху створюють ефект присутності, що додає відчуття реалістичності у бойові сцени та пригоди.

Сюжетні лінії в динамічних іграх також стають більш складними та інтригуючими, часто включаючи моральні вибори, які впливають на розвиток подій. Це дозволяє гравцям не лише відчувати адреналін від динамічних дій, але й занурюватися у глибокі історії, переживаючи справжні емоції разом із персонажами.

5. Стратегії – це жанр відеоігор, що кидає виклик нашому розуму, вимагаючи ретельного планування, тактичного мислення та здатності

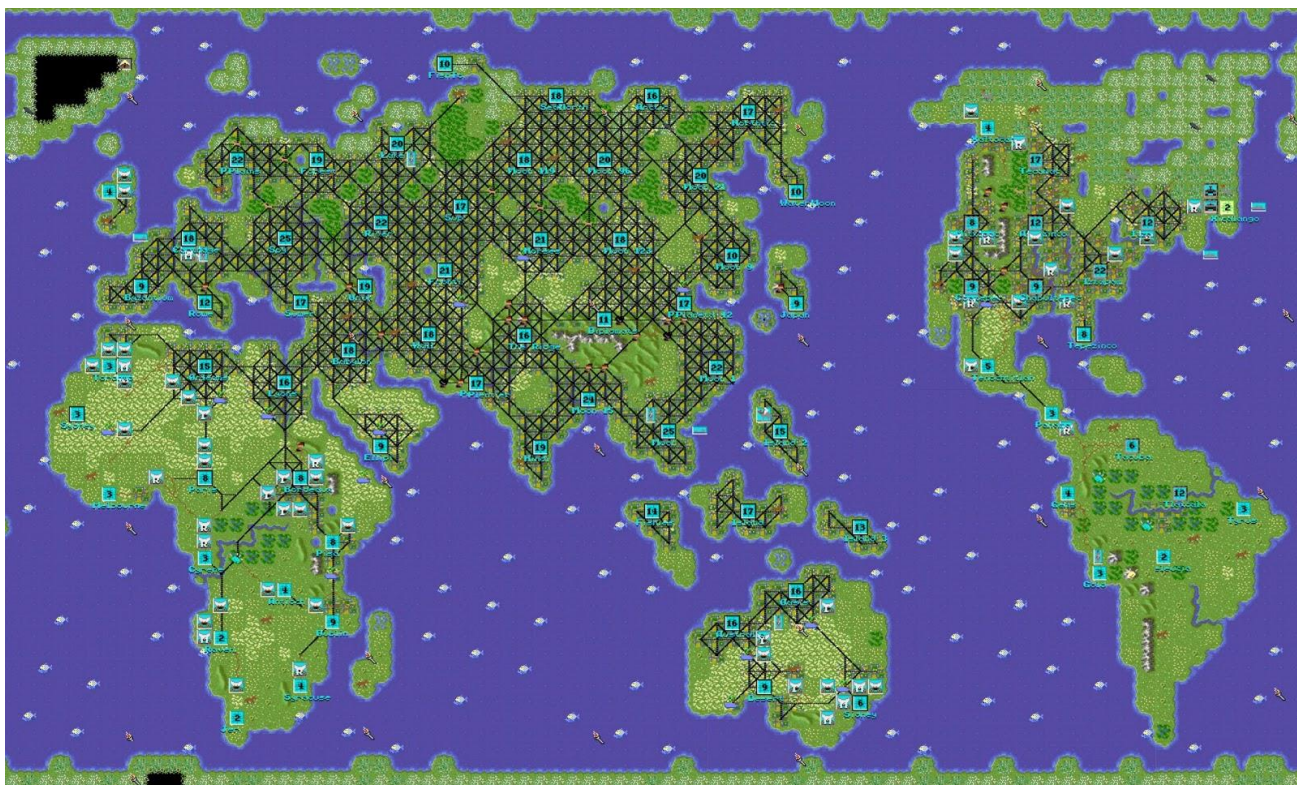


Рисунок 1.5 – Зображення відеогри у жанрі стратегія

приймати важливі рішення. У цьому світі ми стаємо полководцями, які ведуть свої війська до перемоги, правителями, що будують могутні імперії, або менеджерами, які керують складними економічними системами [7].

У стратегічних іграх ми повинні збирати та розподіляти ресурси, розвивати технології, будувати міста та керувати арміями. Потрібно враховувати безліч факторів, таких як особливості місцевості, сильні та слабкі сторони наших військ, дії супротивника та політична ситуація. Наші рішення можуть мати довготривалі наслідки, тому кожен крок слід ретельно обмірковувати.

Ці ігри пропонують різноманітні механіки та сетинги. Від масштабних битв у реальному часі до покрокових дуелей, від історичних епох до

фантастичних світів – кожен може знайти стратегію, що відповідає його інтересам та уподобанням.

Стратегії – це не просто розвага, а можливість розвинути свої інтелектуальні здібності, навчитися приймати важливі рішення та відчувати себе справжнім стратегом. Якщо ви любите виклики та готові до інтелектуальних баталій, то цей жанр чекає на вас!

Графічні стратегії відкривають безмежні можливості для розвитку аналітичного та стратегічного мислення гравців. Вони змушують гравців глибше розуміти взаємозв'язки та наслідки їхніх дій, оскільки кожен крок може мати вирішальне значення для подальшого розвитку ситуації.

У цих іграх вимагається не лише вміння вибирати оптимальні дії в конкретних обставинах, але й здатність уважно аналізувати та прогнозувати можливі наслідки. Гравцеві потрібно бути вкрай обачним та аналітично налаштованим, щоб забезпечити успішний результат.

Ці ігри також сприяють розвитку творчого мислення, оскільки іноді для досягнення мети потрібно шукати нестандартні або незвичні рішення. Це стимулює гравців до винахідливості та гнучкості думки, сприяючи розвитку творчих навичок.

У світі графічних стратегій гравці можуть відчувати справжній вплив на події та розвиток ситуації, що надає відчуття власного значення та контролю над ігровим процесом. Це особливо захоплююче для тих, хто любить брати ініціативу та приймати рішення, що впливають на весь геймплей.

6. Ігри рольового жанру (RPG) – це захоплива можливість відчувати себе в іншому світі, пережити нові пригоди та прийняти нові рішення. Вони перетворюють нас на героїв, які рушають у незабутні відправлення, зіткнення зі злом, дослідження таємничих підземелля і впливають на долі вигаданих світів [8].



Рисунок – 1.6 Зображення відеогри у жанрі RPG

Створення та розвиток персонажів – це ключовий аспект RPG. Ми можемо обирати расу, клас, зовнішність та інші характеристики наших героїв і спостерігати за їхнім ростом та еволюцією протягом гри. Нові здібності, покращення навичок і збирання потужного спорядження допомагають нам стати непереможними воїнами, мудрими магами або хитрими розбійниками.

Сюжет та діалоги відіграють важливу роль у світі RPG. Ми занурюємося у захопливі історії, знайомимося з цікавими персонажами та приймаємо рішення, що впливають на подальший розвиток сюжету. Наші вибори можуть мати далекосяжні наслідки, змінюючи долі персонажів та навіть цілих світів.

Бойова система в RPG може бути різноманітною: від пошагових битв, де ми маємо час на обдумування кожного ходу, до динамічних зіткнень у реальному часі, що вимагають швидкої реакції та вправності. Незалежно від типу бойової системи, ми використовуємо різноманітну зброю, магію та здібності, щоб перемогти ворогів та захистити друзів.

Дослідження світу – це ще один важливий аспект RPG. Ми мандруємо величезними відкритими світами, відвідуємо різноманітні локації, знаходимо скарби, виконуємо побічні завдання та взаємодіємо з навколишнім середовищем. Кожен світ унікальний та наповнений цікавими деталями, які чекають на своїх дослідників.

1.2 Аналіз проблеми графічного дизайну у відеоіграх у жанрі 3D ранерах.

3D-графіка у відеоіграх, зокрема в жанрі 3D-ранерів, є потужним інструментом, який допомагає створювати реалістичні та захоплюючі ігрові світи. Принципи 3D-графіки, такі як моделювання, текстуркування, освітлення та рендеринг, використовуються для створення динамічного геймплею. Моделювання дозволяє створювати деталізовані об'єкти, текстуркування надає їм візуальної глибини, а освітлення створює атмосферу та підкреслює деталі. Однак використання 3D-графіки може також викликати проблеми, такі як низька продуктивність та складність оптимізації. Для подолання цих проблем розробники використовують різні методи оптимізації, такі як зменшення деталізації об'єктів на великій відстані та стиснення текстур. Тестування та оптимізація гри на різних пристроях також відіграють важливу роль у забезпеченні плавної роботи та високої якості графіки.

Для подолання цих проблем розробники також можуть використовувати техніки, які дозволяють динамічно змінювати рівень деталізації об'єктів в залежності від їх відстані до камери. Наприклад, техніка рівня деталізації (LOD) зменшує деталізацію об'єктів, які знаходяться далеко

від камери, щоб зменшити навантаження на систему. Крім того, розробники можуть застосовувати техніки стиснення текстур та оптимізації освітлення, щоб знизити вимоги до обчислювальних ресурсів. Такий підхід дозволяє зберегти якість графіки і забезпечити плавний геймплей навіть на пристроях з обмеженими технічними можливостями.

Додатково, розробники можуть використовувати технології та інструменти для автоматичної оптимізації графіки, такі як системи автоматичного розміщення об'єктів у сцені для оптимізації розташування елементів та мінімізації обсягу обробки. Також розробники можуть використовувати різні техніки управління освітленням, такі як динамічне зміщення джерел світла або використання попередньо обчислених карт освітлення, щоб підвищити продуктивність без значного втручання у візуальну якість. Такі підходи допомагають забезпечити ефективне використання ресурсів та підвищити оптимізацію графічної частини гри.

Крім того, розробники можуть використовувати техніки асинхронної завантаження ресурсів, які дозволяють підгружати необхідні текстури, моделі та інші ресурси на льоту, забезпечуючи оптимальне використання пам'яті та швидкий доступ до даних. Також важливою є оптимізація алгоритмів обробки фізики та колізій, яка допомагає зменшити навантаження на процесор та покращити продуктивність гри без втрати якості графіки. Ці техніки та стратегії сприяють створенню ефективних та захоплюючих ігрових досвідів у жанрі 3D-ранерів.

Важливою є інтеграція з передовими технологіями рендерингу, такими як шейдери та техніки відслідковування променів, які можуть покращити реалістичність освітлення та візуальний рівень деталізації у грі. Використання адаптивних розрядних глибин для рендерингу тіней та оптимізація алгоритмів антиаліасингу також можуть сприяти підвищенню продуктивності та покращенню якості зображення. Ці технології дозволяють досягти більшої реалістичності та іммерсії у візуальному відтворенні гри, що є важливим для створення захоплюючого геймплею у жанрі 3D-ранерів.

Наприклад, для зниження навантаження на процесор та покращення продуктивності гри, розробники можуть використовувати техніку батчингу, яка об'єднує кілька графічних об'єктів у один батч для оптимізації процесу рендерингу. Також важливим є ефективне використання пам'яті, наприклад, шляхом використання компресії текстур та оптимізації їх розміру для зменшення обсягу пам'яті, необхідного для завантаження та зберігання графічних ресурсів. Ці підходи допомагають забезпечити ефективне використання обчислювальних та пам'ятевих ресурсів і гарантують плавну роботу гри навіть на пристроях з обмеженими характеристиками [9].

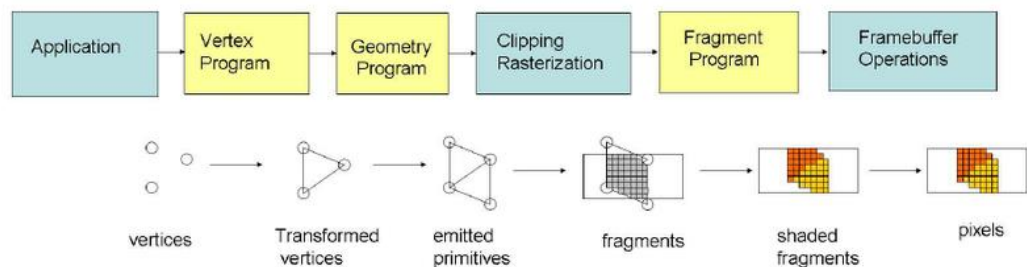


Рисунок 1.7 – Зображення принцип роботи графічного конвеєра

Удосконалення продуктивності:

1. Рівень деталізації (LOD): Зниження рівня докладності віддалених об'єктів.
2. Оптимізація освітлення: Використання спрощених моделей освітлення.
3. Кешування: Запам'ятовування проміжних результатів обчислень для прискорення рендерингу.

Підвищення реалістичності:

1. Трасування променів: Створення живописних відбиттів та тіней (за умови наявності достатніх ресурсів).
2. Фотограметрія: Використання фотографій реальних об'єктів для створення текстур.

Спробувавши використати Urbge 0.3 для створення свого нового проекту - ретро FPS в стилі старої школи. Головна мета була проста: розважитися та насолодитися процесом розробки, адже робота з Urbge була приємною та зручною.

На перший погляд, гра працювала добре. Швидкість складала стабільні 60 кадрів на секунду, і роздільна здатність 1080p надавала чудове візуальне враження.

Проте, коли я додав анімацію холостого циклу для персонажа NPC, раптом все змінилося. Частота кадрів спала до 16 кадрів на секунду, що вкрай розчарувало. Навіть після повторного запуску Urbge та гри проблема залишалася нерозв'язаною [10].

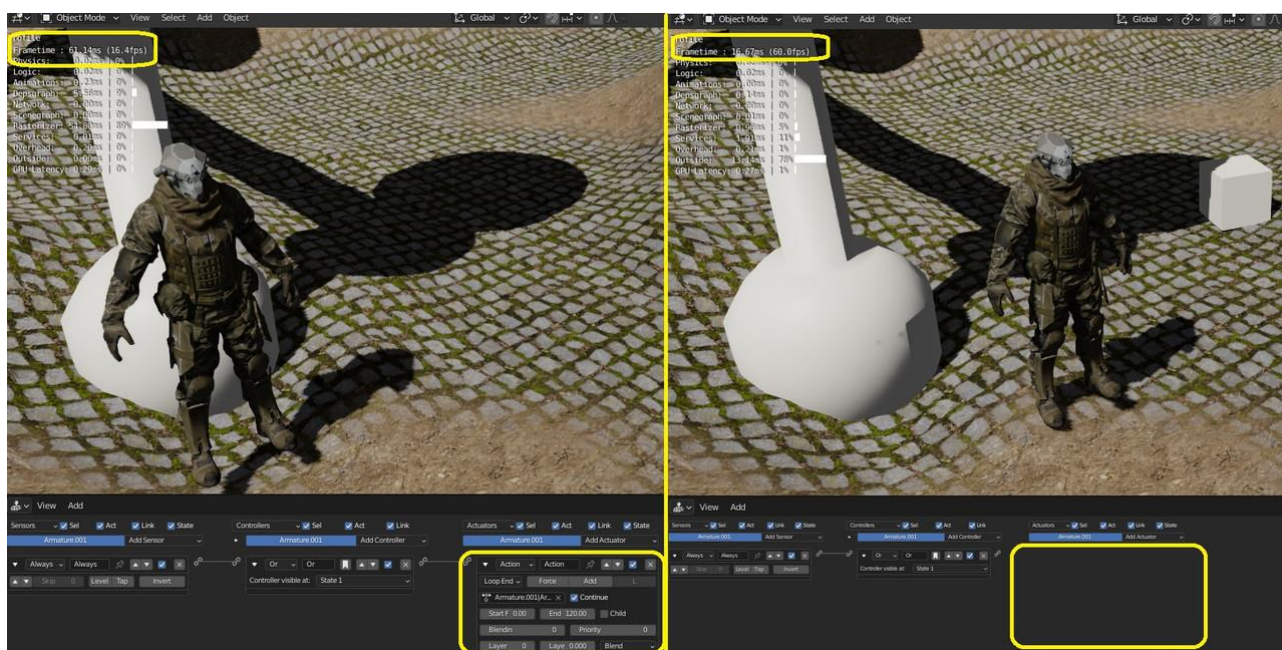


Рисунок 1.8 – Зображення використання Urbge 0.3 та без Urbge 0.3

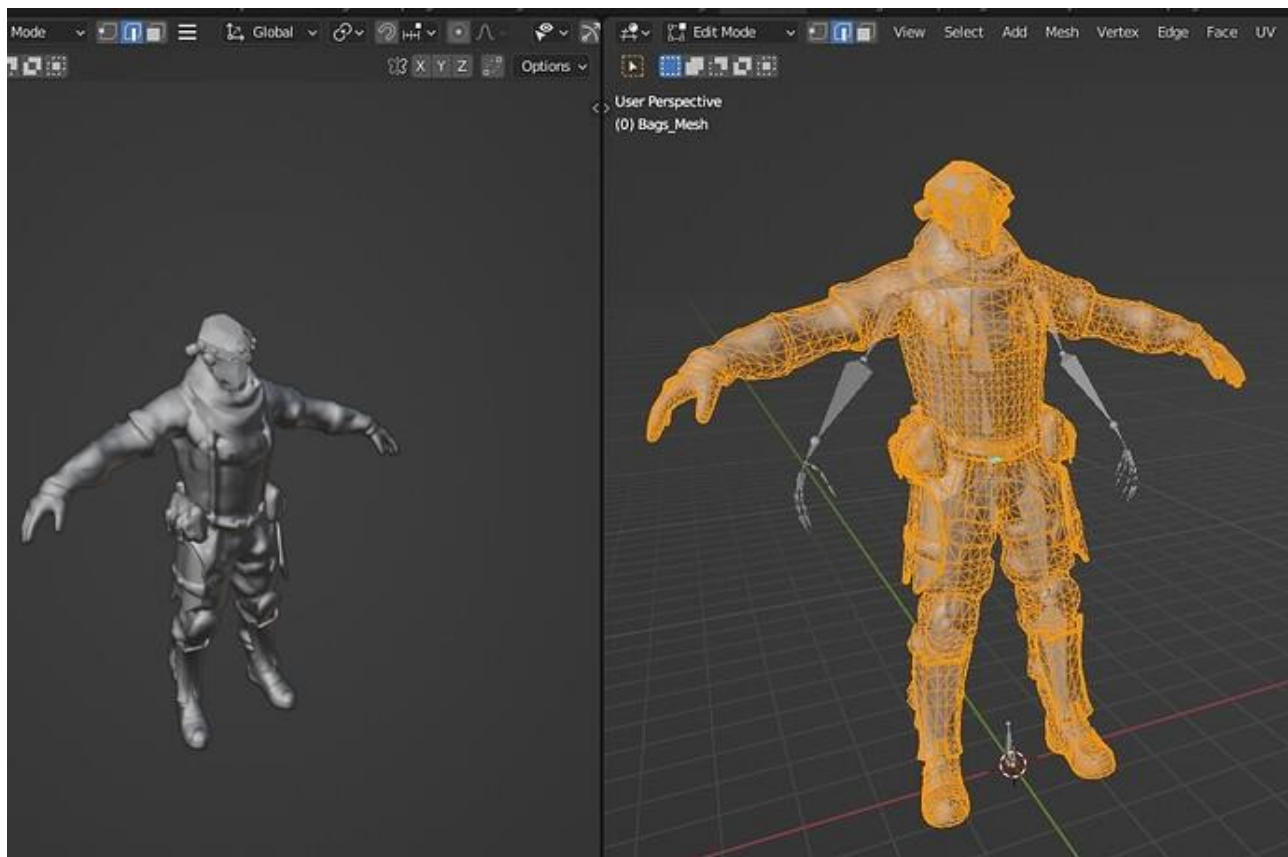


Рисунок 1.9 – Зображення полігони персонажа

Модель NPC має обмежену кількість полігонів, що, в принципі, повинна добре працювати на більшості сучасних ігрових движків.

Приклад алгоритму LOD:

Цей алгоритм визначає рівень деталізації об'єкта в залежності від відстані до камери. Якщо відстань до камери більша за 100, встановлюється найнижчий рівень деталізації. Якщо відстань менше 100, але більше 50, встановлюється середній рівень деталізації. Інакше, якщо відстань менше 50, встановлюється найвищий рівень деталізації.

Відмінне використання алгоритму LOD (рівня деталізації) дозволяє оптимізувати відображення об'єктів в залежності від їх віддаленості, що допомагає підтримувати високу продуктивність гри без втрати якості графіки. Використання таких інструментів і технік є важливим для успішної розробки

захоплюючих 3D-ранерів, де графіка грає вирішальну роль у створенні іммерсивного геймплею.

1.3 Аналіз сучасних відеоігор у жанрі 3D ранери

Сценарій гри в стилі "Ранер" (або "Endless Runner") визначає ключові аспекти ігрового процесу, включаючи механіку, об'єкти, їх поведінку та умови гри. Основна ідея полягає в тому, що гравець керує персонажем, який автоматично біжить вперед, уникаючи перешкод.

Сценарій визначає, які об'єкти є перешкодами, які можна збирати, а також які бонуси та нагороди можна отримати. Він також описує, як гравець керує персонажем, зазвичай за допомогою сенсорного екрана або клавіатури.

Сценарій може включати різні рівні складності та режими гри, такі як "виживання" або проходження рівня на час. Візуальне оформлення та звукові ефекти також є важливими елементами сценарію, що створюють атмосферу гри.

Сценарій визначає умови гри, такі як швидкість персонажа, частота появи перешкод, кількість життів, час проходження рівня та інші параметри, які можуть змінюватися залежно від складності. Ці параметри налаштовуються для створення захопливого ігрового процесу, який мотивує гравця продовжувати грати та покращувати свої результати.

Важливим аспектом сценарію є можливість прогресування у грі, яка зберігає інтерес гравця. Це досягається за рахунок поступового збільшення складності, додавання нових елементів та можливості покращувати свої результати та досягнення.

Сценарій також визначає систему нагород та досягнень, які мотивують гравця досягати нових вершин. Це можуть бути різні види внутрішньоігрової валюти, розблокування нових персонажів, костюмів, рівнів або режимів гри. Важливо, щоб система нагород була збалансованою та стимулювала гравця до подальшого розвитку.

Додатково, сценарій гри в стилі "Ранер" може включати елементи соціальної взаємодії, такі як таблиці лідерів, можливість змагатися з друзями або ділитися своїми досягненнями в соціальних мережах. Це сприяє утриманню гравців та збільшує популярність гри.

Важливим аспектом є також розробка сюжету та сеттінгу гри, які створюють унікальну атмосферу та занурюють гравця у світ гри. Це може бути фантастичний світ, історична епоха, міський пейзаж або будь-який інший сеттінг, який відповідає тематиці гри.

Subway Surfers — це популярна мобільна гра в жанрі "біг без кінця", випущена в 2012 році компаніями Kiloо та SYBO Games. Гравець керує персонажем, який біжить по залізничних коліях, уникаючи перешкод і збираючи монети та бонуси. Гра доступна на платформах iOS, Android, Windows Phone та інших [11].



Рисунок 1.10 – Зображення відеогри Subway Surfer

Гравець починає гру, керуючи персонажем, який втікає від інспектора та його собаки. Потрібно уникати зіткнень з поїздами, бар'єрами та іншими перешкодами, переміщаючись між трьома доріжками. Під час бігу гравець

збирає монети та бонуси, які можна використати для покращення персонажа або відкриття нових.

У грі є багато персонажів, яких можна розблокувати за допомогою монет або спеціальних ключів. Також доступні борди (дошки для серфінгу), які надають тимчасове підвищення швидкості або допомагають уникати перешкод. Гра регулярно отримує оновлення, які включають нові міста або тематичні локації. Часто проводяться сезонні та тематичні події, що додають нові завдання та бонуси.

Гравці можуть змагатися з друзями через таблиці лідерів і підключатися через соціальні мережі, щоб поділитися своїми досягненнями. Subway Surfers була однією з перших ігор у жанрі endless runner, яка здобула глобальну популярність. Вона часто знаходиться в топах завантажень у магазинах додатків. Гра відома своєю барвистою графікою, плавною анімацією та простим, але захоплюючим геймплеем. На момент написання (2024 рік) гра має мільйони активних користувачів щомісяця.

У травні 2018 року Subway Surfers стала першою грою, що досягла одного мільярда завантажень у магазині Google Play. Гра часто співпрацює з різними брендами та проводить крос-промоції. За мотивами гри створено анімаційний серіал, який виходить на YouTube. Subway Surfers продовжує залишатися однією з найпопулярніших мобільних ігор завдяки своїй доступності, регулярним оновленням та захоплюючому геймплею.

Популярність Subway Surfers пояснюється її динамічним геймплеем, постійними оновленнями та доступністю для широкої аудиторії. Кожне оновлення гри додає нові міста з унікальними візуальними елементами, що дозволяє гравцям подорожувати віртуальним світом. Наприклад, гравці можуть бігати по вулицях Токіо, Нью-Йорка, Парижа чи інших відомих міст.

Особлива увага приділяється графіці та дизайну гри. Яскраві кольори, детально промальовані персонажі та анімації роблять гру привабливою для користувачів будь-якого віку. Простий та інтуїтивно зрозумілий інтерфейс

дозволяє новачкам швидко розібратися з керуванням, а вбудовані підказки та навчання допомагають освоїти основні механіки гри.

Соціальні функції гри також сприяють її популярності. Гравці можуть підключатися через Facebook та інші соціальні мережі, змагатися з друзями, ділитися своїми досягненнями та порівнювати результати в таблицях лідерів. Це додає елемент конкуренції і мотивації, що утримує інтерес до гри на довгий час.

Крім основного геймплею, Subway Surfers має безліч додаткових активностей. Гравці можуть збирати спеціальні предмети під час гри, виконувати щоденні завдання та місії, що дозволяють отримувати додаткові бонуси та нагороди. Ці елементи роблять гру більш різноманітною та захоплюючою.

Subway Surfers також відома своїми колабораціями з різними відомими брендами та медіа-франшизами. Це додає грі нові елементи і привертає увагу нових гравців. Наприклад, в грі з'являлися спеціальні персонажі та дошки для серфінгу, створені у співпраці з такими брендами, як Nike або Coca-Cola.

Анімаційний серіал за мотивами Subway Surfers, який виходить на YouTube, також сприяє популярності гри. Серіал дозволяє глядачам зануритися в світ гри, дізнатися більше про персонажів та їхні пригоди. Це створює додатковий зв'язок між гравцями та ігровою всесвітом, розширюючи її вплив за межі мобільного додатку.

В цілому, Subway Surfers продовжує розвиватися і залишатися однією з найулюбленіших мобільних ігор завдяки своїй простоті, динамічному геймплею, яскравій графіці та постійним оновленням. Незважаючи на конкуренцію в жанрі endless runner, гра утримує провідні позиції і продовжує залучати мільйони гравців по всьому світу.

Temple Run – це популярна мобільна гра в жанрі нескінченного бігу, розроблена Imaing Studios. Вперше випущена 4 серпня 2011 року, вона швидко здобула широку популярність. У грі гравець керує персонажем, який вкрав ідол з храму і тікає від демонічних мавп. Завдання – пробігти якомога далі, уникаючи перешкод і збираючи монети [12].



Рисунок 1.11 – Зображення відеогри Temple Run

Персонаж автоматично біжить вперед, а гравець використовує свайпи для стрибків, ковзання і поворотів, а також нахилом пристрою змінює положення на доріжці. Під час бігу необхідно уникати різних перешкод, таких як дерева, скелі і ями. На шляху гравець збирає монети, які можна використовувати для покращення персонажа або розблокування нових героїв, а також бонуси, що тимчасово надають особливі можливості.

У січні 2013 року вийшло продовження – Temple Run 2, яке пропонує покращену графіку, нові перешкоди, додаткові бонуси і нові локації. Гра включає нові механіки, такі як поїздка на вагонетці та спуск по тросу. Також було випущено кілька версій гри у співпраці з іншими брендами, наприклад, Temple Run: Brave і Temple Run

Temple Run стала справжнім культурним феноменом, популяризуючи жанр нескінченних бігунів на мобільних пристроях. Гра завантажена мільйонами користувачів по всьому світу і стала однією з найбільш знакових мобільних ігор свого часу, встановивши нові стандарти в індустрії мобільних розваг.

1.4 Висновок до розділу 1

У першому розділі було проведено огляд індустрії відеоігор, включаючи її історію, жанри, технології та вплив на суспільство. Особливу увагу було приділено жанру 3D-ранерів, який є популярним і перспективним напрямком у мобільних іграх.

Було розглянуто ключові аспекти розробки 3D-ранерів, такі як геймплей, графічний дизайн, технічні особливості та проблеми, з якими стикаються розробники. Зокрема, було проаналізовано проблеми, пов'язані з використанням 3D-графіки, такі як продуктивність та оптимізація.

Також було проведено аналіз двох популярних ігор у цьому жанрі - Subway Surfers та Temple Run. Було виявлено, що ці ігри мають багато спільних рис, таких як основні механіки геймплею, але також відрізняються у візуальному стилі, сеттінгу та додаткових функціях.

Цей аналіз дозволяє зробити висновок, що жанр 3D-ранерів є динамічним та конкурентним. Для успіху в цьому жанрі розробникам необхідно враховувати як технічні аспекти, такі як оптимізація графіки та продуктивність, так і геймплейні елементи, унікальний сеттінг.

У наступному розділі буде розглянуто процес розробки та проектування програмного модуля для 3D-ранеру з використанням ігрового двигуна Unity. Це дозволить глибше зрозуміти технічні аспекти створення таких ігор та запропонувати рішення для подолання проблем, виявлених у цьому розділі.

2 РОЗРОБКА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО МОДУЛЯ ВІДЕОГРИ «SPACE DUST»

2.1 Розробка архітектури програмного модулю відеогри «Space Dust»

Розробка архітектури програмного модуля відеогри "Space Dust" є ключовим етапом у створенні захоплюючого та функціонального ігрового досвіду. Цей розділ присвячений детальному опису архітектури модуля, яка включає в себе ключові компоненти, їх взаємодію та функціональні можливості.

Компонент керування персонажем (Player Controller) є центральним елементом взаємодії гравця з ігровим світом у 3D-раннері "Space Dust". Він відіграє ключову роль у забезпеченні плавного, чуйного та інтуїтивно зрозумілого керування персонажем, що є критично важливим для задоволення та успіху гри.

Основні функції:

1) Обробка вхідних даних: Компонент постійно відстежує вхідні дані від гравця, які можуть надходити з різних джерел, таких як сенсорний екран (нахили, свайпи), клавіатура (натискання клавіш) або геймпад (рух джойстиків). Отримані дані інтерпретуються відповідно до налаштувань керування. Наприклад, нахил пристрою вправо може означати команду "рух вправо", а свайп вгору - "стрибок". Вхідні дані можуть бути шумними або неточними. Компонент фільтрує та згладжує їх, щоб забезпечити плавне та передбачуване керування.

2) Рух персонажа: Компонент відповідає за переміщення персонажа ігровим світом відповідно до команд гравця. Це включає рух вперед (автоматичний біг), рух вліво/вправо та стрибки. Компонент враховує фізичні закони ігрового світу, такі як гравітація та тертя, щоб рух персонажа виглядав реалістично. Також компонент виявляє зіткнення персонажа з перешкодами та

іншими об'єктами на рівні. Це дозволяє реалізувати різні ігрові механіки, такі як ухилення від перешкод, збирання бонусів або отримання пошкоджень.

3) Анімація: Компонент запускає відповідні анімації для кожної дії персонажа. Наприклад, при бігу відтворюється анімація бігу, при стрибку - анімація стрибка тощо. Також компонент забезпечує плавний перехід між різними анімаціями, щоб рухи персонажа виглядали природно та реалістично. Анімації синхронізуються з рухом персонажа, щоб створити враження, що персонаж дійсно виконує ці дії.

Взаємодія з іншими компонентами:

1) Level Generator: Отримує інформацію про розташування перешкод та бонусів на рівні, щоб правильно реалізувати зіткнення та збір бонусів.

2) Obstacle and Bonus Manager: Обробляє зіткнення персонажа з перешкодами та бонусами, ініціюючи відповідні дії (втрата здоров'я, отримання бонусу тощо).

3) UI Manager: Надсилає інформацію про стан персонажа (наприклад, кількість життів, зібрані монети) для відображення на інтерфейсі користувача.

4) Sound Manager: Відтворює звукові ефекти, пов'язані з діями персонажа (кроки, стрибки, зіткнення тощо).

Технічна реалізація: компонент керування персонажем зазвичай реалізується як скрипт на мові C# в Unity. Він використовує фізичний движок Unity для реалізації руху та зіткнень, а також систему анімації Unity для відтворення анімацій. Для обробки вхідних даних використовуються стандартні засоби Unity, такі як Input Manager.

Важливість: компонент керування персонажем є одним з найважливіших компонентів 3D-ранеру. Він визначає, наскільки зручно та приємно грати в гру. Добре реалізоване керування персонажем робить гру більш захоплюючою та дозволяє гравцеві повністю зосередитися на ігровому процесі.

Компонент генерації рівня (Level Generator) є одним з ключових елементів у 3D-раннері "Space Dust", оскільки він відповідає за створення

ігрового середовища, яким рухається персонаж. Цей компонент забезпечує динамічність та різноманітність ігрового процесу, генеруючи рівні на льоту, що робить кожне проходження унікальним.

Основні функції:

1) Процедурна генерація: алгоритмічне створення: Замість того, щоб створювати кожен рівень вручну, компонент використовує алгоритми для автоматичної генерації рівнів. Це дозволяє створювати нескінченну кількість унікальних рівнів, що забезпечує високу реіграбельність гри. Параметри генерації: Алгоритми можуть використовувати різні параметри, такі як довжина рівня, кількість та типи перешкод, частота появи бонусів тощо. Ці параметри можуть змінюватися в залежності від прогресу гравця або інших факторів. Рандомізація: Для забезпечення різноманітності рівнів використовуються елементи випадковості. Наприклад, розташування перешкод, тип бонусів та інші елементи можуть бути визначені випадковим чином.

2) Управління складністю: прогресивна складність: У міру просування гравця по рівнях, їх складність поступово зростає. Це може відбуватися за рахунок збільшення швидкості руху персонажа, збільшення кількості та різноманітності перешкод, зменшення відстані між ними тощо. Адаптивна складність: Компонент може аналізувати дії гравця та адаптувати складність рівня відповідно до його навичок. Наприклад, якщо гравець успішно долає перешкоди, складність може бути збільшена, а якщо гравець часто програє, складність може бути знижена.

3) Різноманітність: типи перешкод: Компонент створює різні типи перешкод, такі як статичні бар'єри, рухомі перешкоди (потяги, автомобілі), пастки тощо. Це робить ігровий процес більш цікавим та вимагає від гравця різних навичок для успішного проходження рівнів. Типи бонусів: Компонент також генерує різні типи бонусів, які можуть давати гравцеві тимчасові переваги (наприклад, невразливість, подвійні очки) або постійні покращення (наприклад, нові персонажі, дошки для серфінгу).

Взаємодія з іншими компонентами:

1) Player Controller: Отримує інформацію про поточну позицію гравця, щоб генерувати рівень перед ним та знищувати елементи рівня, які залишилися позаду.

2) Obstacle and Bonus Manager: Передає інформацію про створені перешкоди та бонуси, щоб вони могли бути розміщені на рівні та взаємодіяти з персонажем.

Технічна реалізація: компонент генерації рівня зазвичай реалізується як скрипт на мові C# в Unity. Він використовує різні алгоритми та структури даних для представлення рівня та його елементів. Для генерації випадкових чисел використовуються вбудовані функції Unity.

Важливість: компонент генерації рівня є критично важливим для успіху 3D-ранеру "Space Dust". Він забезпечує нескінченний ігровий процес, підтримує інтерес гравця та створює унікальний досвід кожного проходження. Добре реалізований компонент генерації рівня робить гру більш захоплюючою та реіграбельною.

Компонент перешкод та бонусів (Obstacle and Bonus Manager) відіграє важливу роль у створенні динамічного та захоплюючого ігрового процесу в 3D-раннері "Space Dust". Він відповідає за створення, розміщення та управління всіма об'єктами на ігровому рівні, крім самого персонажа гравця. Цей компонент забезпечує різноманітність та непередбачуваність ігрового світу, змушуючи гравця постійно адаптуватися та приймати швидкі рішення.

Основні функції:

1) Створення та розміщення об'єктів: генерація: Компонент створює різноманітні перешкоди (потяги, бар'єри, ями тощо) та бонуси (монети, магніти, невразливість тощо) на основі алгоритмів, визначених компонентом генерації рівня. Розміщення: Об'єкти розміщуються на рівні з урахуванням різних факторів, таких як тип об'єкта, складність рівня, прогрес гравця та інші параметри. Пул об'єктів: Для оптимізації продуктивності компонент може використовувати пул об'єктів, тобто створювати обмежену кількість об'єктів

на початку гри та повторно використовувати їх, замість того щоб створювати нові об'єкти щоразу.

2) Управління поведінкою об'єктів: Рух: Компонент визначає, як об'єкти рухаються по рівню. Перешкоди можуть рухатися назустріч гравцеві, бонуси можуть обертатися або рухатися по складній траєкторії. Зіткнення: Компонент визначає, як об'єкти взаємодіють з персонажем гравця та іншими об'єктами на рівні. Наприклад, зіткнення з перешкодою може призвести до втрати здоров'я або завершення гри, а збір бонусу може дати гравцеві тимчасову перевагу. Зникнення: Компонент визначає, коли об'єкти повинні зникати з рівня. Наприклад, перешкоди можуть зникати після того, як гравець їх пройде, а бонуси можуть зникати через певний час.

Взаємодія з іншими компонентами:

1) Level Generator: Отримує від компонента генерації рівня інформацію про те, які об'єкти потрібно створити та де їх розмістити.

2) Player Controller: Обробляє зіткнення об'єктів з персонажем гравця та повідомляє компоненту керування персонажем про наслідки цих зіткнень.

3) UI Manager: Надсилає інформацію про зібрані бонуси та інші події, пов'язані з об'єктами, для відображення на інтерфейсі користувача.

4) Sound Manager: Відтворює звукові ефекти, пов'язані з об'єктами (наприклад, звук зіткнення з перешкодою або звук збору бонусу).

Технічна реалізація: компонент перешкод та бонусів зазвичай реалізується як скрипт на мові C# в Unity. Він використовує різні класи та методи Unity для створення, розміщення та управління об'єктами. Для виявлення зіткнень використовуються фізичні компоненти Unity, такі як Collider та Rigidbody.

Важливість: компонент перешкод та бонусів є важливим елементом геймплею 3D-панеру "Space Dust". Він створює виклики та можливості для гравця, робить ігровий процес більш динамічним та захоплюючим. Добре реалізований компонент перешкод та бонусів може значно підвищити реіграбельність та задоволення від гри.

Компонент інтерфейсу користувача (UI Manager) є важливим елементом 3D-раннера "Space Dust", який забезпечує взаємодію гравця з грою та надає йому необхідну інформацію для успішного проходження рівнів. Він відповідає за всі візуальні елементи, які гравець бачить на екрані, крім самого ігрового світу.

Основні функції:

1) Відображення інформації: Рахунок: Показує поточний рахунок гравця, який може залежати від пройденої відстані, зібраних монет та інших факторів. Монети: Відображає кількість монет, зібраних гравцем під час гри. Відстань: Показує пройдену гравцем відстань. Життя/Здоров'я: Відображає кількість життів або рівень здоров'я персонажа. Інші показники: Може відображати інші важливі показники, такі як час, швидкість, бонуси тощо.

2) Меню та кнопки: головне меню: Надає доступ до різних розділів гри, таких як початок гри, налаштування, таблиця лідерів тощо. Меню паузи: Дозволяє гравцеві призупинити гру, змінити налаштування або вийти з гри. Кнопки дій: Реалізує кнопки для виконання різних дій у грі, таких як стрибок, ухилення, використання бонусів тощо. Інтерактивні елементи: Може включати інші інтерактивні елементи, такі як слайдери для налаштування гучності звуку або вибору персонажа.

3) Візуальні ефекти: Анімації: Використовує анімації для покращення візуального сприйняття інтерфейсу, наприклад, анімацію збільшення рахунку при зборі монет або анімацію переходу між меню. Переходи: Забезпечує плавні переходи між різними екранами та меню, щоб зробити інтерфейс більш зручним та приємним для використання. Інші ефекти: Може включати інші візуальні ефекти, такі як розмиття фону, ефекти частинок тощо.

Взаємодія з іншими компонентами:

1) Player Controller: Отримує інформацію про стан персонажа (рахунок, монети, здоров'я тощо) для відображення на інтерфейсі.

2) Obstacle and Bonus Manager: Отримує інформацію про зібрані бонуси та інші події для оновлення інтерфейсу.

3) Save/Load Manager: Отримує команди від гравця (через кнопки меню) для збереження або завантаження прогресу гри.

4) Sound Manager: Надсилає команди для відтворення звукових ефектів, пов'язаних з діями в інтерфейсі (наприклад, натискання кнопок).

Технічна реалізація: UI Manager зазвичай реалізується за допомогою інструментів Unity для створення інтерфейсу користувача (UI Canvas, UI Elements). Він використовує скрипти на мові C# для керування відображенням інформації, обробки подій та взаємодії з іншими компонентами гри.

Важливість: UI Manager є важливим компонентом, який забезпечує зручність та інформативність гри. Він допомагає гравцеві орієнтуватися в ігровому процесі, розуміти свій прогрес та приймати рішення. Гарно розроблений інтерфейс користувача може значно покращити враження від гри та зробити її більш привабливою для гравців.

Компонент збереження та завантаження (Save/Load Manager) відіграє важливу роль у забезпеченні збереження прогресу гравця у 3D-раннері "Space Dust". Він дозволяє гравцеві продовжити гру з того місця, де він зупинився, зберігаючи досягнуті результати, розблоковані персонажі та інші важливі дані.

Основні функції:

1) Збереження прогресу: збір даних: компонент збирає всі необхідні дані про поточний стан гри, такі як поточний рівень, кількість зібраних монет, розблоковані персонажі, досягнення гравця тощо. Серіалізація: Дані перетворюються у формат, який можна зберегти на пристрої гравця. Це може бути формат JSON, XML або інший зручний для зберігання та читання формат, Запис даних: Серіалізовані дані записуються у файл або інше сховищеданих, наприклад, PlayerPrefs в Unity.

2) Завантаження прогресу: читання даних: Компонент зчитує збережені дані з файлу або сховища. Десеріалізація: Дані перетворюються з формату зберігання назад у структури даних, які використовуються грою. Відновлення стану гри: Компонент відновлює стан гри відповідно до завантажених даних.

Це може включати встановлення поточного рівня, кількості монет, розблокування персонажів тощо.

Взаємодія з іншими компонентами:

1) Player Controller: Отримує дані про стан персонажа (рівень, монети тощо) для збереження та встановлює стан персонажа після завантаження.

2) Level Generator: Отримує дані про поточний рівень для збереження та встановлює рівень після завантаження.

3) Obstacle and Bonus Manager: Отримує дані про розблоковані персонажі та бонуси для збереження та встановлює їх доступність після завантаження.

4) UI Manager: Отримує команди від гравця (через кнопки меню) для збереження або завантаження прогресу гри та може відображати повідомлення про успішне збереження/завантаження.

Технічна реалізація: компонент збереження та завантаження зазвичай реалізується як скрипт на мові C# в Unity. Він використовує різні класи та методи Unity для роботи з файлами та сховищами даних, а також бібліотеки для серіалізації/десеріалізації даних (наприклад, JsonUtility в Unity).

Важливість: компонент збереження та завантаження є важливим елементом для забезпечення комфорту гравця. Він дозволяє гравцеві не втрачати свій прогрес, що підвищує задоволення від гри та мотивацію продовжувати грати. Крім того, цей компонент може бути використаний для реалізації різних ігрових механік, таких як система досягнень або можливість продовжити гру після поразки за певну плату.

Компонент звуку (Sound Manager) відповідає за створення звукового оформлення гри "Space Dust", що відіграє важливу роль у формуванні атмосфери та посиленні емоційного впливу на гравця. Він керує відтворенням фонової музики, звукових ефектів та забезпечує налаштування гучності.

Основні функції:

1) Відтворення звукових ефектів: Різноманітність звуків: Компонент містить бібліотеку звукових ефектів, що відповідають різним подіям у грі:

стрибкам, зіткненням з перешкодами, збору бонусів, досягненням тощо. Тригери: Звукові ефекти відтворюються у відповідь на певні події в грі, такі як дії гравця, зіткнення, зміна стану гри. Позиціонування звуку: Для деяких звуків може бути важливим їх позиціонування у просторі, щоб створити реалістичне відчуття напрямку та відстані до джерела звуку.

2) Відтворення музики: Фонова музика: Компонент відтворює фонову музику, яка створює атмосферу гри та підтримує її темп. Адаптивна музика: Музика може змінюватися в залежності від ситуації в грі, наприклад, ставати більш напруженою під час небезпечних моментів або більш спокійною під час збору бонусів. Переходи між треками: Компонент забезпечує плавні переходи між різними музичними треками, щоб уникнути різких змін та підтримувати музичну цілісність.

3) Управління гучністю: Налаштування гучності: Компонент дозволяє гравцеві налаштовувати гучність звукових ефектів та музики окремо. Зміна гучності в реальному часі: Гучність може динамічно змінюватися під час гри, наприклад, зменшуватися під час діалогів або збільшуватися під час екшн-сцен.

Взаємодія з іншими компонентами:

1) Player Controller: Отримує події від компонента керування персонажем (стрибки, зіткнення тощо) та відтворює відповідні звукові ефекти.

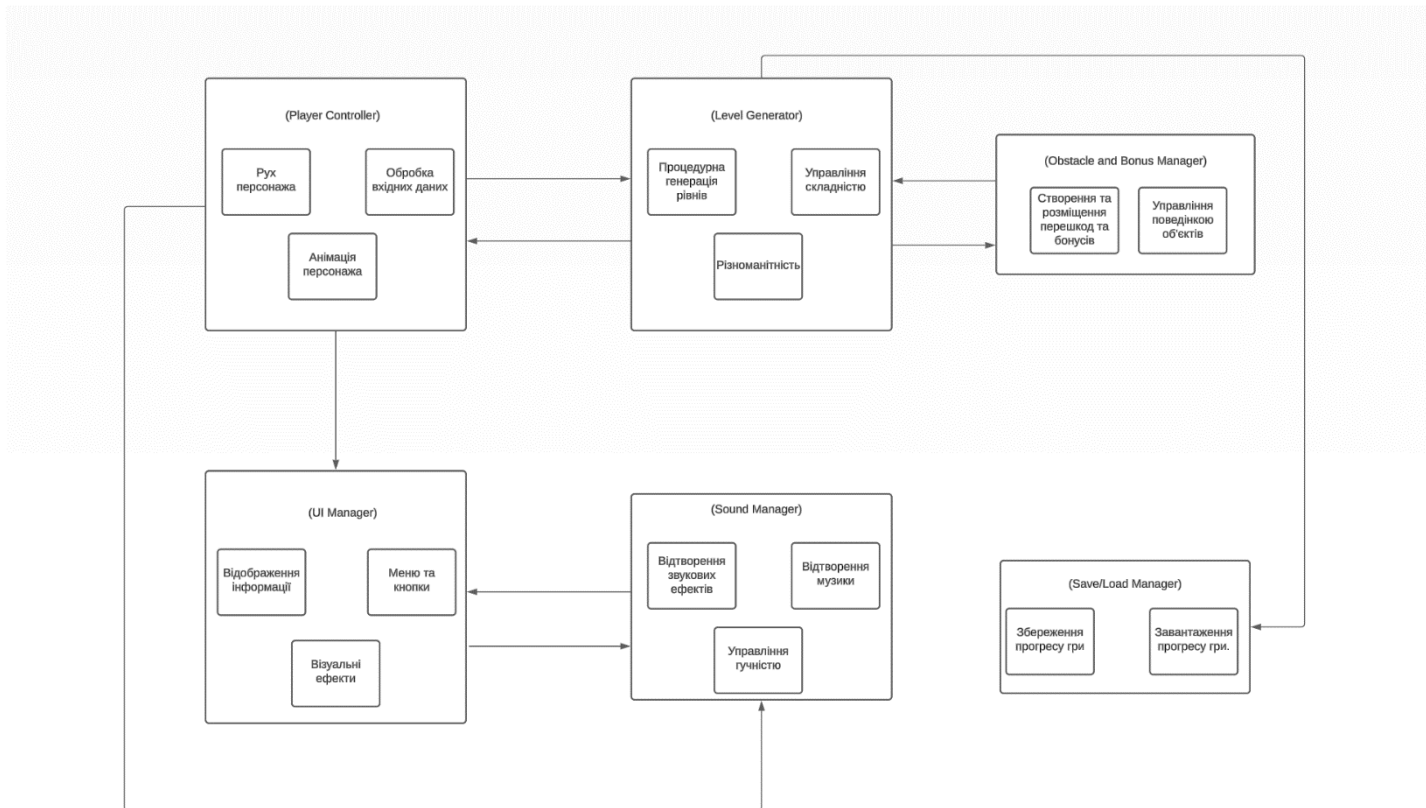
2) Obstacle and Bonus Manager: Отримує події від компонента перешкод та бонусів (зіткнення, збір бонусів) та відтворює відповідні звукові ефекти.

3) UI Manager: Отримує команди від гравця (через інтерфейс користувача) для зміни налаштувань гучності.

Технічна реалізація Sound Manager зазвичай реалізується як скрипт на мові C# в Unity. Він використовує аудіосистему Unity (AudioSource, AudioClip) для відтворення звуків та музики. Для позиціонування звуку використовуються 3D-звукові компоненти Unity.

Важливість: Звукове оформлення є невід'ємною частиною будь-якої гри, і "Space Dust" не є винятком. Компонент Sound Manager допомагає

створити більш захоплюючий та емоційний ігровий досвід, підкреслюючи важливі моменти гри та створюючи відповідну атмосферу. Якісне звукове оформлення може значно підвищити задоволення від гри та зробити її більш привабливою для гравців.



2.1 Конструктивна схема основних компонентів архітектури програмного модулю комп'ютерної гри «Space Dust»

2.2 Розробка методів створення ігрової механіки

Розробка методів створення ігрової механіки є важливим етапом у створенні відеоігор, оскільки саме ігрова механіка визначає, як гравець взаємодіє з грою та які дії він може виконувати. У контексті 3D-раннера "Space Dust" ігрова механіка включає в себе кілька ключових аспектів:

1) **Управління персонажем:** Управління персонажем є основою будь-якої гри. У "Space Dust" воно буде інтуїтивно зрозумілим та простим, щоб гравці могли швидко освоїти його та зосередитися на ігровому процесі. Основними діями персонажа будуть: Рух вліво та вправо: Це дозволить гравцеві уникати перешкод та збирати бонуси, розташовані на різних лініях руху. Для реалізації цього механізму буде використано акселерометр мобільного пристрою або клавіші зі стрілками на клавіатурі.

Стрибки: Стрибки дозволять гравцеві перестрибувати через перешкоди на землі та збирати бонуси, розташовані у повітрі. Для стрибка буде використано жест свайп вгору по екрану або клавіша пробілу [13].

Ковзання: Ковзання під низькими перешкодами додасть різноманітності до геймплею та змусить гравця швидко реагувати на зміну ситуації на рівні. Ковзання буде активовано жестом свайп вниз або натисканням клавіші вниз.

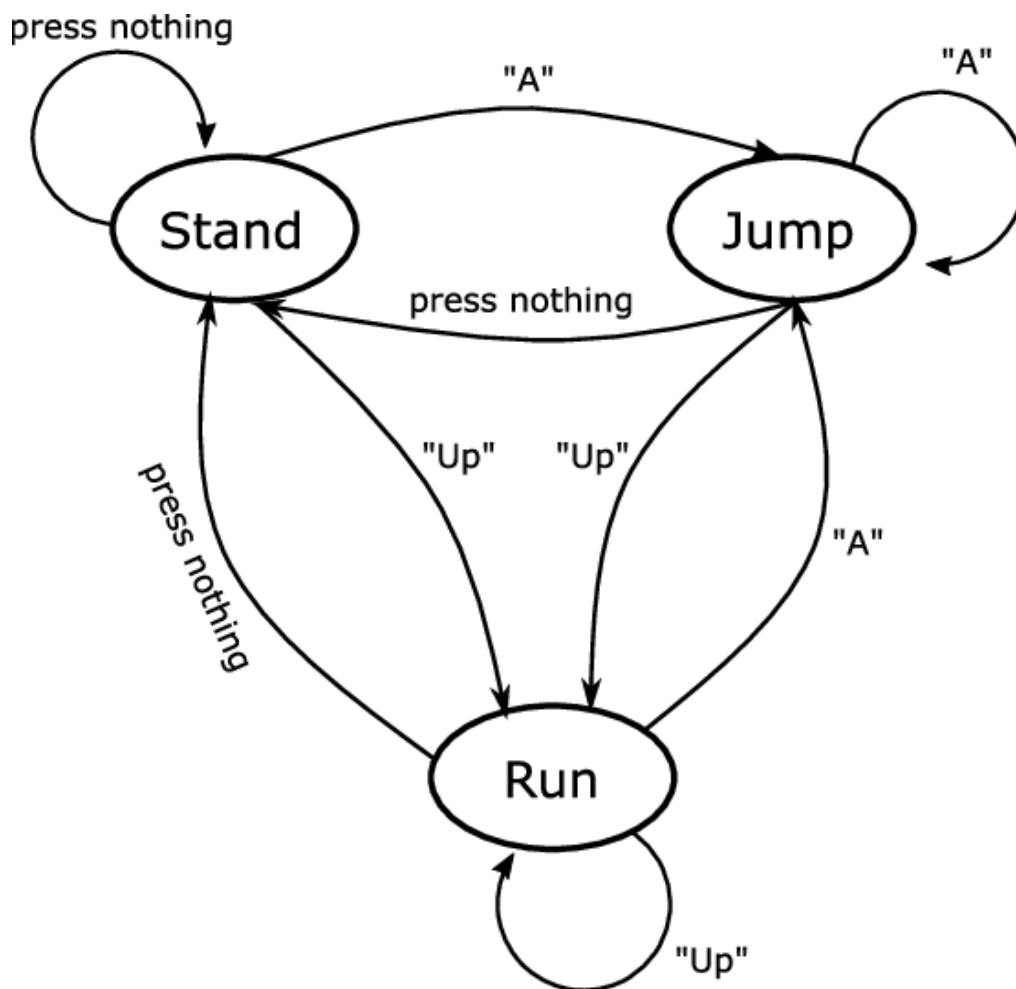


Рисунок 2.2 – Модель станів персонажа

2) Генерація рівня: Оскільки "Space Dust" є раннером, рівні повинні бути нескінченними та різноманітними. Для цього буде використана процедурна генерація, яка дозволить створювати унікальні рівні на льоту під час гри. Алгоритм генерації враховуватиме такі фактори:

Складність рівня: Складність буде поступово збільшуватися в міру проходження гри. Це може виражатися у збільшенні швидкості руху персонажа, збільшенні кількості та складності перешкод, зменшенні відстані між ними тощо. Це буде реалізовано за допомогою системи рівнів, де кожен новий рівень буде складнішим за попередній.

Різнманітність перешкод: Для того, щоб гра не набридала гравцеві, будуть створені різні типи перешкод: статичні (стіни, ящики), рухомі (потяги,

лазери) та пастки (ями, падаючі предмети). Це змусить гравця постійно бути наготові та використовувати різні навички для успішного подолання перешкод. Різні типи перешкод будуть з'являтися на рівні випадковим чином, але з урахуванням складності рівня.

Розташування бонусів: Бонуси будуть розміщуватися на рівнях таким чином, щоб стимулювати гравця до дослідження рівня та ризику. Наприклад, деякі бонуси можуть бути розташовані у важкодоступних місцях, що вимагатиме від гравця додаткових зусиль для їх отримання. Розташування бонусів також буде генеруватися випадковим чином, але з урахуванням того, щоб вони були доступні для гравця.

3) Система перешкод та бонусів: Перешкоди та бонуси є невід'ємною частиною будь-якого раннера. У "Space Dust" будуть реалізовані різні типи перешкод та бонусів:

Статичні перешкоди, такі як стіни, ящики та інші об'єкти, які потрібно обходити або перестрибувати. Вони будуть мати різні розміри та форми, щоб урізноманітнити геймплей.

Рухомі перешкоди, такі як потяги, лазери та інші об'єкти, що рухаються, які вимагають від гравця швидкої реакції та точного розрахунку часу. Швидкість та траєкторія руху цих перешкод будуть змінюватися в залежності від складності рівня.

Пастки, такі як ями або падаючі предмети, які додають елемент несподіванки та ризику. Пастки будуть активуватися при наближенні гравця або через певний проміжок часу.

Бонуси: Монети: Основна ігрова валюта, яку можна використовувати для купівлі нових персонажів, костюмів або покращень. Монети будуть розкидані по всьому рівню, і гравець повинен буде збирати їх, уникаючи перешкод.

Магніти: Притягують монети до гравця, полегшуючи їх збір. Магніти будуть діяти протягом обмеженого часу.

Невразливість: Тимчасово захищає гравця від зіткнень з перешкодами. Це дозволить гравцеві пройти небезпечні ділянки рівня без ризику програти.

Подвійні очки: Збільшують кількість очок, які гравець отримує за певний час. Це стимулюватиме гравця збирати більше монет та бонусів.

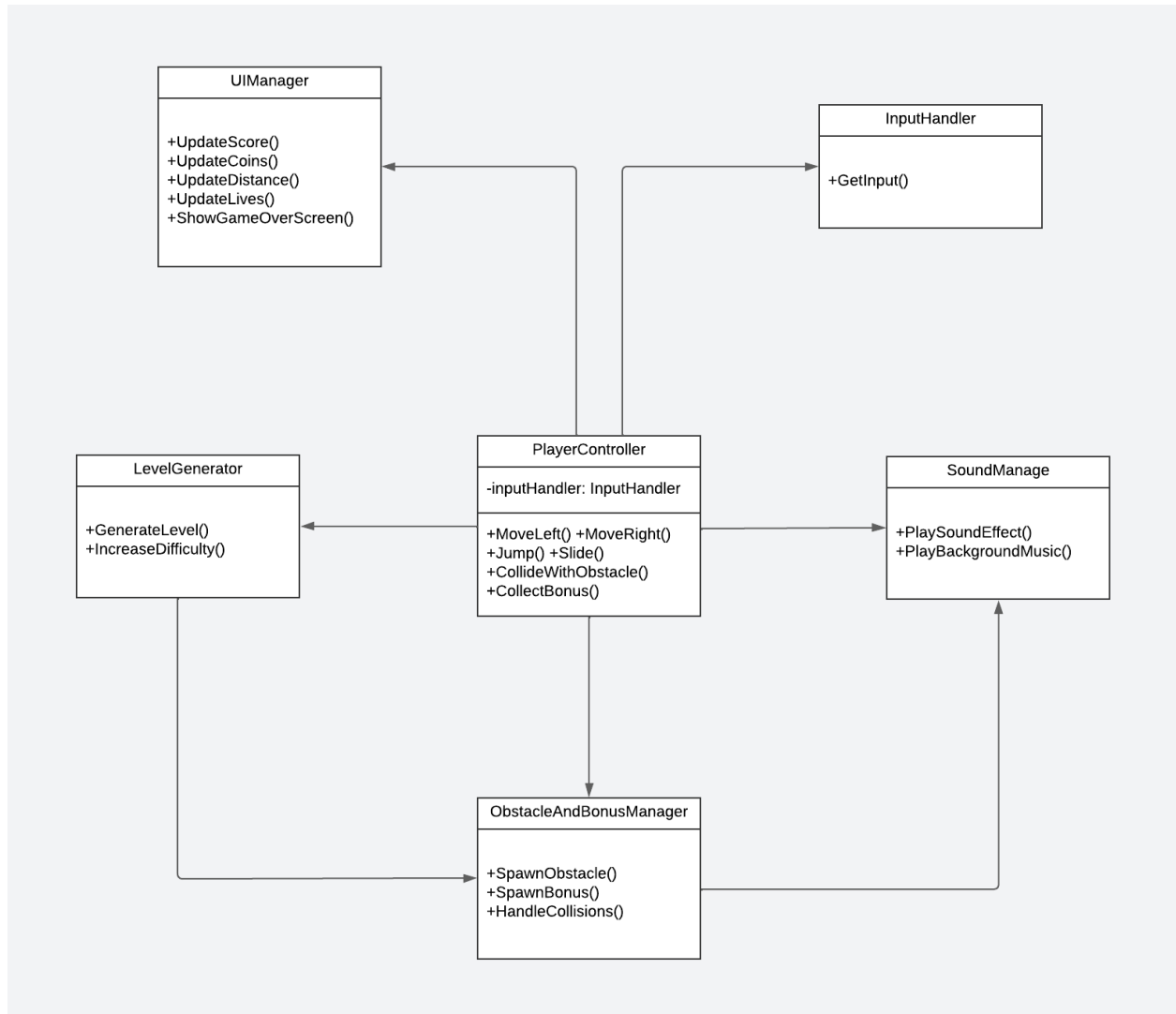


Рисунок 2.3 – UML-діаграма класів модульного дизайну програмного модулю комп'ютерної гри «Space Dust»

4) Система підрахунку очок та прогресу: Система підрахунку очок мотивує гравця досягати кращих результатів. У "Space Dust" очки будуть нараховуватися за пройденої відстань, зібрані монети та бонуси. Система прогресу включатиме: Розблокування нових персонажів та костюмів: Це

можна зробити за досягнення певних цілей (наприклад, пробігти певну відстань) або за монети. Нові персонажі та костюми будуть мати унікальні характеристики та зовнішній вигляд. Покращення характеристик персонажа: За монети гравець зможе покращувати швидкість, магнетизм та інші характеристики свого персонажа. Це дозволить гравцеві краще справлятися з перешкодами та досягати вищих результатів [14].

Досягнення: Виконання певних завдань (наприклад, пробігти певну відстань без зіткнень) буде винагороджуватися досягненнями, що додасть грі ще один рівень зацікавленості. Досягнення будуть відображатися у профілі гравця та можуть бути використані для порівняння результатів з іншими гравцями.

5) Інші ігрові механіки: Крім основних механік, у грі будуть реалізовані й інші елементи, що зроблять геймплей цікавішим та різноманітнішим:

Система життів: Гравець матиме обмежену кількість життів (наприклад, три), які втрачатимуться при зіткненні з перешкодами. Після втрати всіх життів гра закінчиться.

Можливість продовження гри: Після втрати всіх життів, гравець зможе продовжити гру з того ж місця за певну кількість монет. Це дасть гравцеві другий шанс покращити свій результат.

Таблиця лідерів: Таблиця лідерів дозволить гравцям змагатися з друзями та іншими гравцями по всьому світу, що додасть грі соціальний аспект. Гравці зможуть бачити свої результати у порівнянні з іншими та прагнути досягти вершини рейтингу.

Всі ці механіки будуть реалізовані за допомогою ігрового двигуна Unity, який надає широкий спектр інструментів та можливостей для створення 3D-ігор. Це дозволить створити захоплюючий та збалансований ігровий процес, який буде цікавим для широкої аудиторії гравців.

2.3 Розробка алгоритму функціонування програмного модулю комп'ютерної гри «Space Dust»

Алгоритм роботи програмного модулю комп'ютерної гри «Space Dust» представлено на рисунку 2.4.

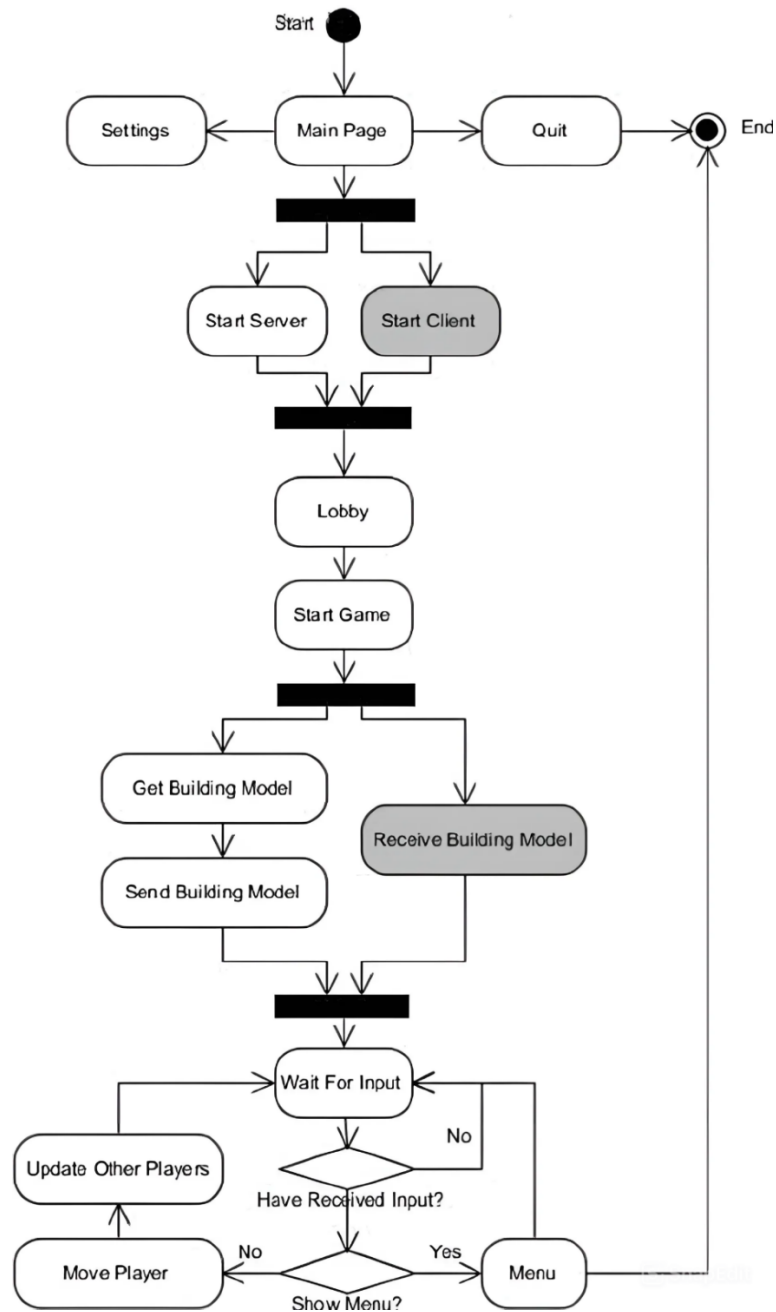


Рисунок 2.4 - Алгоритм роботи програмного модулю комп'ютерної гри «Space Dust»

Спершу завантажуються всі необхідні ресурси для гри: моделі персонажів, елементи оточення, текстури, анімації, звукові ефекти та музичний супровід. Потім створюються основні ігрові об'єкти: головний персонаж, доріжка (яка може складатися з динамічно створюваних сегментів), перешкоди і бонуси.

Основний ігровий цикл включає кілька важливих етапів. По-перше, відбувається оновлення стану гри, тобто позицій персонажа, перешкод та бонусів. Одночасно генеруються нові сегменти доріжки та обробляються зіткнення. По-друге, відтворюється графіка, що включає відображення персонажа, його анімації, оточення та інших об'єктів. По-третє, обробляється ввід від користувача, який включає натискання клавіш або жестів для стрибків, ухилень та інших дій.

Процес генерації контенту включає рандомізацію сегментів доріжки для забезпечення різноманітності та визначення рівня складності сегментів залежно від прогресу гри. Також відбувається випадкове розміщення перешкод і бонусів із врахуванням певних правил, щоб підтримувати баланс у грі.

При зіткненні з перешкодами зменшується кількість очок життя персонажа або гра завершується. При зіткненні з бонусами гравець отримує додаткові очки чи інші переваги, наприклад, тимчасову невразливість.

Анімація персонажа охоплює різні дії, такі як стрибки, ухилення і падіння, з плавними переходами між ними. Візуальні ефекти додають динамічності, наприклад, ефекти при зборі бонусів або зіткненнях.

Звуковий супровід гри включає фонову музику та звуки оточення, а також звукові ефекти при зборі бонусів і зіткненнях. Це додає глибини ігровому процесу і покращує занурення гравця у гру [15].

Після завершення гри відображається екран результатів з досягненнями гравця. Можливі варіанти перезапуску гри або виходу з неї. Результати можуть зберігатися для подальшого аналізу або порівняння.

Оптимізація гри є важливою для ефективного використання ресурсів, особливо в 3D графіці. Якщо гра розробляється для різних платформ, необхідно враховувати їх специфіку. Регулярне тестування допомагає виявляти баги та покращувати ігровий процес.

Взаємодія між цими модулями відбувається через систему подій та повідомлень. Наприклад, коли персонаж збирає бонус, Obstacle and Bonus Manager надсилає повідомлення UI Manager, щоб оновити кількість зібраних монет на екрані. Коли гравець програє, Player Controller надсилає повідомлення UI Manager, щоб відобразити екран завершення гри.

Така модульна архітектура дозволяє розробникам легко додавати нові функції та механіки в гру, а також змінювати існуючі, не впливаючи на інші модулі. Це робить процес розробки більш гнучким та ефективним.

2.4 Висновок до розділу 2

У другому розділі було детально розглянуто процес розробки та проектування програмного модуля для 3D-раннера "Space Dust". Було описано ключові компоненти архітектури модуля, такі як:

Player Controller: Відповідає за керування персонажем, обробку вхідних даних від гравця та взаємодію з іншими компонентами гри.

Level Generator: Забезпечує процедурну генерацію рівнів, враховуючи різні параметри та забезпечуючи різноманітність ігрового процесу.

Obstacle and Bonus Manager: Створює та керує перешкодами та бонусами на рівні, забезпечуючи динамічність та непередбачуваність геймплею.

UI Manager: Відповідає за відображення інтерфейсу користувача, надаючи гравцеві необхідну інформацію та елементи керування.

Save/Load Manager: Забезпечує збереження та завантаження прогресу гравця, дозволяючи продовжити гру з останнього збереження.

Sound Manager: Відповідає за звукове оформлення гри, включаючи відтворення музики та звукових ефектів.

Було розроблено методи створення ігрової механіки, включаючи управління персонажем, генерацію рівня, систему перешкод та бонусів, систему підрахунку очок та прогресу, а також інші ігрові механіки, такі як система життів, можливість продовження гри та таблиця лідерів.

Також було представлено алгоритм функціонування програмного модуля, який описує взаємодію між різними компонентами та основні етапи ігрового циклу.

В цілому, другий розділ надає детальний опис технічних аспектів розробки 3D-раннера "Space Dust" та пропонує конкретні рішення для створення захоплюючого та функціонального ігрового досвіду. Цей розділ є важливим внеском у розуміння процесу розробки ігор та може бути використаний як основа для подальшої реалізації та вдосконалення гри.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОГРАМНОГО МОДУЛЮ КОМП'ЮТЕРНОЇ ГРИ «SPACE DUST»

3.1 Обґрунтування вибору рушія для програмної реалізації

Розробка програмного забезпечення, особливо ігор, вимагає ретельного вибору середовища розробки. Існує безліч ігрових рушіїв, кожен зі своїми перевагами та недоліками. Деякі з них спеціалізуються на 2D-графіці (Construct 2, Corona, Cocos2D-X), інші на 3D (Unreal Engine, Unity 5, CryEngine). Важливим фактором є також вартість використання: деякі рушії безкоштовні, тоді як інші вимагають щомісячної або щорічної підписки.

Для полегшення вибору пропонується розглянути порівняльну таблицю з коротким описом плюсів та мінусів популярних рушіїв CryEngine, Cocos2D-x, Unity 3D та Unreal Engine.

Таблиця 3.1 – Опис переваг та недоліків рушіїв Feature, Unity 3D, Unreal Engine, Godot, GameMaker Studio 2 та Buildbox.

Feature	Unity	Unreal Engine	Godot	GameMaker Studio 2	Buildbox
Мова програмування	C# - широко використовується, має статичну типізацію, що допомагає виявляти помилки на етапі компіляції.	C++ - потужна та гнучка мова, але має складний синтаксис та вимагає більшої уваги до управління пам'яттю.	GDScript - проста у вивченні мова, схожа на Python, з динамічною типізацією, що прискорює розробку, але може	GML - спеціалізована мова для створення ігор, має простий синтаксис, але менш потужна та гнучка, ніж C# або C++.	Візуальний інтерфейс без коду, не вимагає знань програмування.

			призводить и до помилок під час виконання .		
3D-графіка	Потужні інструменти для роботи з 3D-графікою, підтримка шейдерів, освітлення, тіней та інших ефектів.	Високоякісна 3D-графіка з підтримкою передових технологій, таких як трасування променів у реальному часі.	Хороші можливості для 3D-графіки, але менш потужні, ніж в Unity або Unreal Engine.	Обмежені можливості для створення 3D-ігор.	Обмежені можливості для створення 3D-ігор.
Цільові платформи	Підтримує широкий спектр платформ, включаючи мобільні пристрої, ПК та ігрові консолі.	Підтримує широкий спектр платформ, але основний фокус на ПК та консолях.	Підтримує широкий спектр платформ, включаючи мобільні пристрої, ПК та деякі консолі.	Підтримує ПК, мобільні пристрої та деякі консолі.	Підтримує мобільні пристрої та веб.
Вартість	Безкоштовна версія з достатньою функціональністю для створення повноцінних	Безкоштовний для проектів з невеликим бюджетом, але	Повністю безкоштовний з відкритим вихідним кодом.	Безкоштовна версія з обмеженою функціональністю, платні версії пропонують	Безкоштовна версія з обмеженою функціональністю, платні підписки пропонують

	ігор. Платні версії пропонують розширені можливості.	вимагає відрахувань з продажів для комерційних проектів.		більше можливостей.	більше можливостей.
Складність	Легкий у вивченні та використанні, великий вибір навчальних матеріалів та підтримка спільноти.	Більш складний у вивченні, вимагає знань C++, але пропонує більше контролю та гнучкості.	Простий у вивченні завдяки GDScript, але менше навчальних матеріалів та менша спільнота.	Простий у вивченні, але обмежений у можливостях.	Дуже простий у використанні, не вимагає програмування.

Занирюючись у світ ігрових рушіїв, де панують CryEngine та Unreal Engine, кожен з яких має свої виняткові переваги та унікальні можливості.

Відомий завдяки неймовірній графіці в таких ігрових іконах, як Crysis і Far Cry, CryEngine зачаровує розробників, які прагнуть досягти вражаючого рівня візуальної якості у своїх творіннях. Цей рушій постійно розвивається, додаючи інноваційні функції та можливості, які відкривають нові горизонти для візуальних ефектів та реалізму.

Unreal Engine, творіння Epic Games, заслужив свою популярність завдяки здатності адаптуватися до різноманітних ігрових жанрів та масштабів. Підтримка широкого спектру платформ, включаючи консолі, ПК та мобільні пристрої, робить його універсальним інструментом. Відкритий вихідний код і використання мови C++ надають розробникам велику гнучкість і можливість підлаштовуватися під конкретні потреби проекту.

Кожен з цих рушіїв має свої унікальні переваги. Unreal Engine вважається більш універсальним, особливо для командної роботи, в той час як CryEngine приваблює розробників, які ставлять на перше місце візуальну якість. Unity, з іншого боку, ідеально підходить для невеликих команд та мобільних проектів завдяки своїй простоті та великій спільноті користувачів.

Вибір між CryEngine та Unreal Engine значною мірою визначається потребами вашого проекту. Якщо вам важлива візуальна якість, CryEngine - чудовий варіант. Однак, якщо вам потрібна універсальність і можливість працювати на різних платформах, то Unreal Engine може бути кращим рішенням.

Unity 3D відомий своєю універсальністю та доступністю. Завдяки компонентному підходу, інтуїтивно зрозумілому інтерфейсу та потужному візуальному редактору, цей ігровий рушій робить розробку ігор дуже простою навіть для початківців. Однією з головних переваг Unity 3D є велика бібліотека готових активів та плагінів, які можна легко інтегрувати у ваш проект, що значно скорочує час розробки та дозволяє зосередитися на творчих аспектах.

Славиться своїми багатоплатформними можливостями, оскільки підтримує широкий спектр пристроїв та операційних систем, від ПК до мобільних пристроїв, консолей та VR/AR гарнітур. Це дозволяє розробникам легко адаптувати свої ігри під різні платформи та охопити максимальну аудиторію. Рушій пропонує потужні інструменти для роботи з графікою, фізикою, анімацією та іншими важливими аспектами розробки, що дозволяє навіть невеликим командам створювати високоякісні ігри з реалістичною фізикою та візуалізацією.

Unity 3D - чудовий вибір для тих, хто хоче створювати ігри без глибоких технічних знань. Простота використання, різноманітність ресурсів та потужні функції роблять його ідеальним інструментом для розробників

усіх рівнів кваліфікації.

Для забезпечення високої якості, зрозумілості та естетичності програм важливо дотримуватися архітектурних принципів. Одним з таких патернів є MVC (Model-View-Controller), який розділяє програму на три взаємопов'язані компоненти: модель, представлення та контролер. Використання MVC дозволяє розробникам розділити відповідальність між різними частинами програми, зробити код більш структурованим і легшим в обслуговуванні.

Хоча MVC традиційно використовується для розробки графічних інтерфейсів користувача, він також став популярним у веб-розробці, оскільки багато сучасних мов програмування та фреймворків підтримують його.

Unity 3D також підтримує MVC, що допомагає розробникам ігор створювати добре структуровані проекти. Такий підхід допомагає створювати якісні, масштабовані та легко підтримувані ігри.

3.2 Програмна реалізація модулю комп'ютерної гри "Space Dust"

Ігровий рушій Unity 3D обрано для розробки гри "Space Dust" через його простоту використання, швидкість розробки, кросплатформеність, потужний функціонал та широку спільноту підтримки. Unity 3D надає інтуїтивно зрозумілий інтерфейс та широкий спектр інструментів, що дозволяє швидко створювати прототипи та реалізовувати ігрові механіки. Крім того, Unity 3D підтримує різноманітні платформи, включаючи Windows, macOS, Linux, iOS та Android, що забезпечує можливість охопити широку аудиторію гравців.

Для гри "Space Dust" буде використана компонентна архітектура Unity 3D. Цей підхід дозволяє розділити логіку гри на окремі компоненти, що відповідають за різні аспекти геймплею, такі як керування персонажем, генерація рівня, перешкоди, інтерфейс користувача та звук.

Основні компоненти:

1) Player Controller: Цей компонент відповідає за керування діями персонажа, обробляє вхідні дані від гравця (наприклад, нахили, свайпи або натискання клавіш) та реалізує рух, стрибки та інші дії персонажа.

2) Level Generator: Цей компонент генерує ігрові рівні під час гри, використовуючи процедурну генерацію. Це забезпечує нескінченність та різноманітність рівнів, що підвищує реіграбельність гри.

3) Obstacle and Bonus Manager: Компонент відповідає за створення та розміщення перешкодна рівні. Він також обробляє зіткнення персонажа з цими об'єктами, активуючи відповідні ігрові події, такі як отримання бонусів або втрата життів.

4) UI Manager: Цей компонент відображає інтерфейс користувача, включаючи рахунок гравця, кількість зібраних монет, пройденої відстані та інші важливі дані. Він також може містити елементи керування, такі як кнопки для паузи або перезапуску гри.

5) Sound Manager: Цей компонент відповідає за відтворення звукових ефектів та музики в грі. Він може відтворювати різні звуки залежно від дій гравця або подій у грі, створюючи атмосферу та підсилюючи емоційне сприйняття.

Для реалізації ігрової механіки в Unity 3D використовуються скрипти на мові програмування C#. Кожен компонент архітектури гри буде реалізований у вигляді окремого скрипта, який буде містити логіку роботи цього компонента. Наприклад, скрипт `PlayerMovement.cs`, відповідає за рух персонажа, обробляє вхідні дані від гравця та переміщує персонажа в ігровому світі відповідно до цих даних.

```

using UnityEngine;

public class PlayerMovement : MonoBehaviour
{
    public float moveSpeed = 5f;
    public float jumpForce = 10f;
    private CharacterController characterController;

    void Start()
    {
        characterController = GetComponent<CharacterController>();
    }

    void Update()
    {
        float horizontalInput = Input.GetAxis("Horizontal");
        Vector3 moveDirection = new Vector3(horizontalInput, 0f, 0f);
        characterController.Move(moveDirection * moveSpeed * Time.deltaTime);
        if (Input.GetButtonDown("Jump"))
        {
            characterController.Move(new Vector3(0f, jumpForce, 0f));
        }
    }
}

```

Рисунок 3.1 Реалізація скрипта PlayerMovement.cs

Скрипт LevelGeneration.cs відповідає за генерацію рівнів, він буде створювати нові сегменти доріжки, перешкоди та бонуси в міру просування гравця вперед.

```

using UnityEngine;

public class LevelGeneration : MonoBehaviour
{
    public GameObject[] platformPrefabs;
    public float spawnDistance = 10f;
    private Vector3 lastSpawnPosition;

    void Start()
    {
        lastSpawnPosition = transform.position;
        for (int i = 0; i < 5; i++)
        {
            SpawnPlatform();
        }
    }

    void Update()
    {
        if (transform.position.z - lastSpawnPosition.z < spawnDistance)
        {
            SpawnPlatform();
        }
    }

    void SpawnPlatform()
    {
        int randomIndex = Random.Range(0, platformPrefabs.Length);
        GameObject platform = Instantiate(platformPrefabs[randomIndex], lastSpawnPosition + new Vector3(0f, 0f, 10f), Quaternion.identity);
        lastSpawnPosition = platform.transform.position;
    }
}

```

Рисунок 3.2 Реалізація скрипта LevelGeneration.cs

Скрипт ObstacleAndBonus.cs відповідає за поведінку перешкод. Він буде визначати, як вони рухаються, як взаємодіють з персонажем та коли зникають з рівня.

```

using UnityEngine;

public class ObstacleAndBonus : MonoBehaviour
{
    public enum Type { Obstacle, Coin, Magnet, Invincibility };
    public Type type;

    void OnTriggerEnter(Collider other)
    {
        if (other.CompareTag("Player"))
        {
            if (type == Type.Obstacle)
            {
                Debug.Log("Зіткнення з перешкодою!");
            }
            else if (type == Type.Coin)
            {
                Debug.Log("Монета зібрана!");
            }
            Destroy(gameObject);
        }
    }
}

```

Рисунок 3.3 Реалізація скрипта LevelGeneration.cs

Скрипт UIManager.cs відповідає за керування інтерфейсом користувача. Він забезпечує відображення та оновлення інформації про гру на екрані, такої як рахунок, пройдена відстань,

```

using UnityEngine;

using UnityEngine.UI;

public class UIManager : MonoBehaviour
{
    public Text scoreText;

    public Text coinsText;

    public Text distanceText;

    void Update()
    {
        scoreText.text = "Score: " + PlayerController.score;

        coinsText.text = "Coins: " + PlayerController.coins;

        distanceText.text = "Distance: " + PlayerController.distance;
    }
}

```

3.4 Реалізація скрипта UIManager.cs

Скрипт SoundManager.cs відповідає за відтворення звукових ефектів та музики в грі. Він містить аудіо кліпи для різних звуків та музичних треків, а також методи для їх відтворення та керування гучністю.

```

using UnityEngine;

public class SoundManager : MonoBehaviour
{
    public AudioClip jumpSound;
    public AudioClip coinSound;
    public AudioClip crashSound;
    public AudioClip backgroundMusic;
    private AudioSource audioSource;

    void Start()
    {
        audioSource = GetComponent<AudioSource>();
        PlayBackgroundMusic();
    }

    public void PlaySound(AudioClip clip)
    {
        audioSource.PlayOneShot(clip);
    }

    public void PlayBackgroundMusic()
    {
        audioSource.clip = backgroundMusic;
        audioSource.loop = true;
        audioSource.Play();
    }
}

```

Рисунок 3.5 Реалізація скрипта SoundManager.cs

3.3 Тестування програмного модулю комп'ютерної гри

Під час тестування програмного модуля комп'ютерної гри важливо перевірити реалізацію всіх ігрових механік, які були спроектовані та розроблені. Також слід оцінити рівень оптимізації комп'ютерної графіки та графічний інтерфейс відеогри. Процес тестування починається з відкриття Unity та запуску розробленого програмного модуля. Після відкриття проекту

відображається головне меню комп'ютерної гри «Space Dust», яке можна побачити на рисунку 3.6.

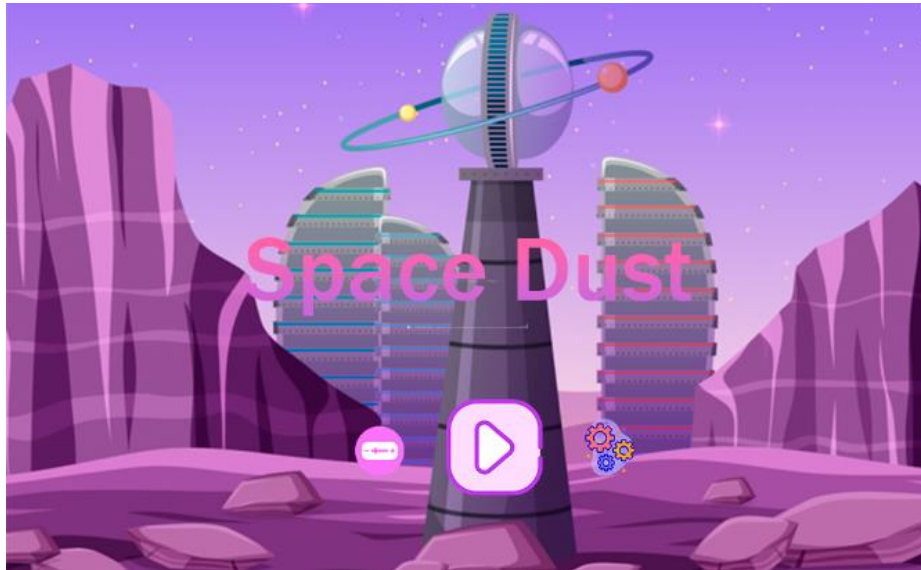


Рисунок 3.6 – Загальний вигляд інтерфейсного вікна "Головне меню" в комп'ютерній грі "Space Dust".

Після натискання на середню кнопку, що означає «Грати», відкривається перша локація комп'ютерної гри (рис. 3.5)

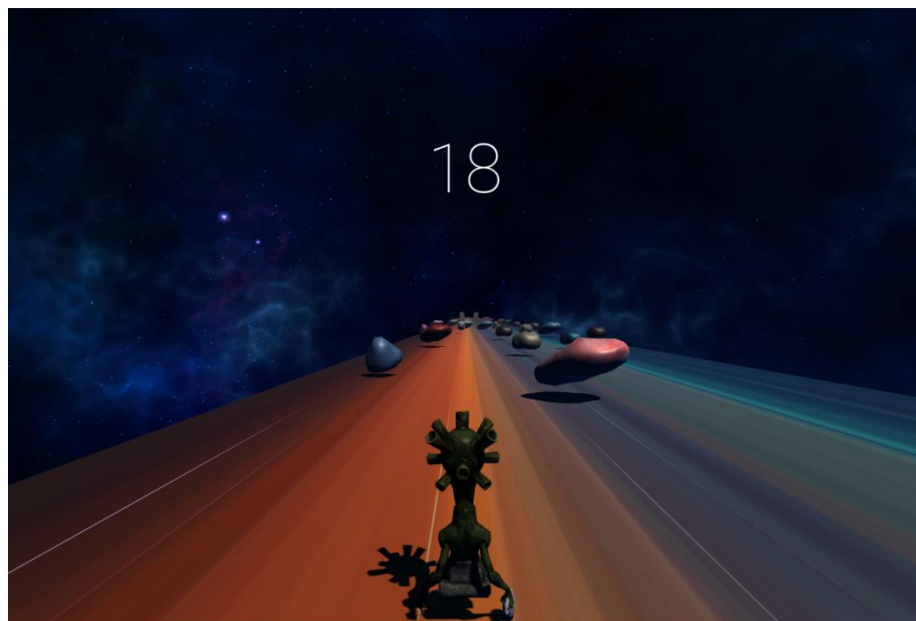


Рисунок 3.7 - Загальний вигляд інтерфейсного вікна "Першого рівня" в комп'ютерній грі "Space Dust".



Рисунок 3.8 - Загальний вигляд інтерфейсного вікна "Другого рівня" в комп'ютерній грі "Space Dust".

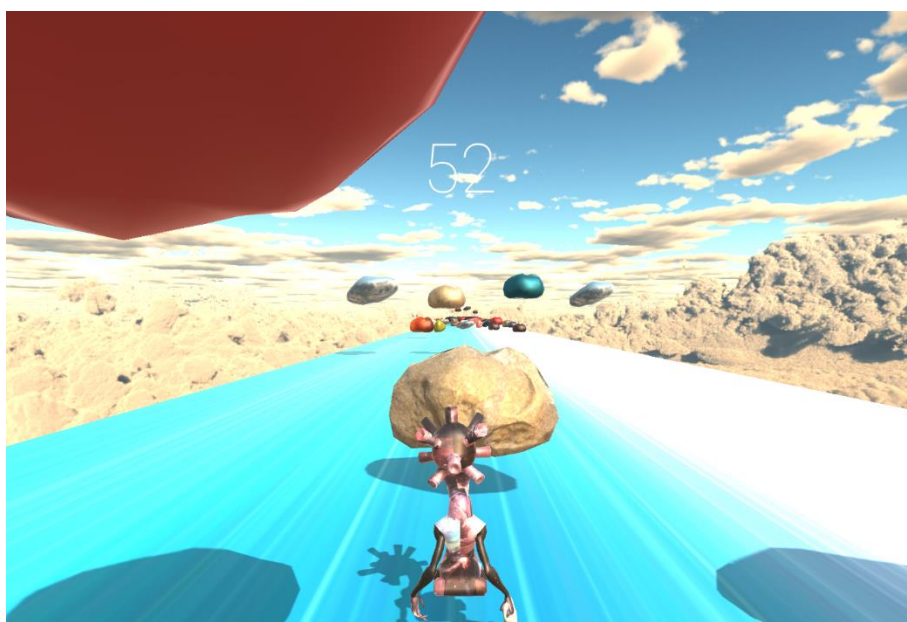


Рисунок 3.9 – Загальний вигляд інтерфейсного вікна "Третього рівня" в комп'ютерній грі "Space Dust".

При з'єднанні персонажа з об'єктом гра завершується (рис 3.10)

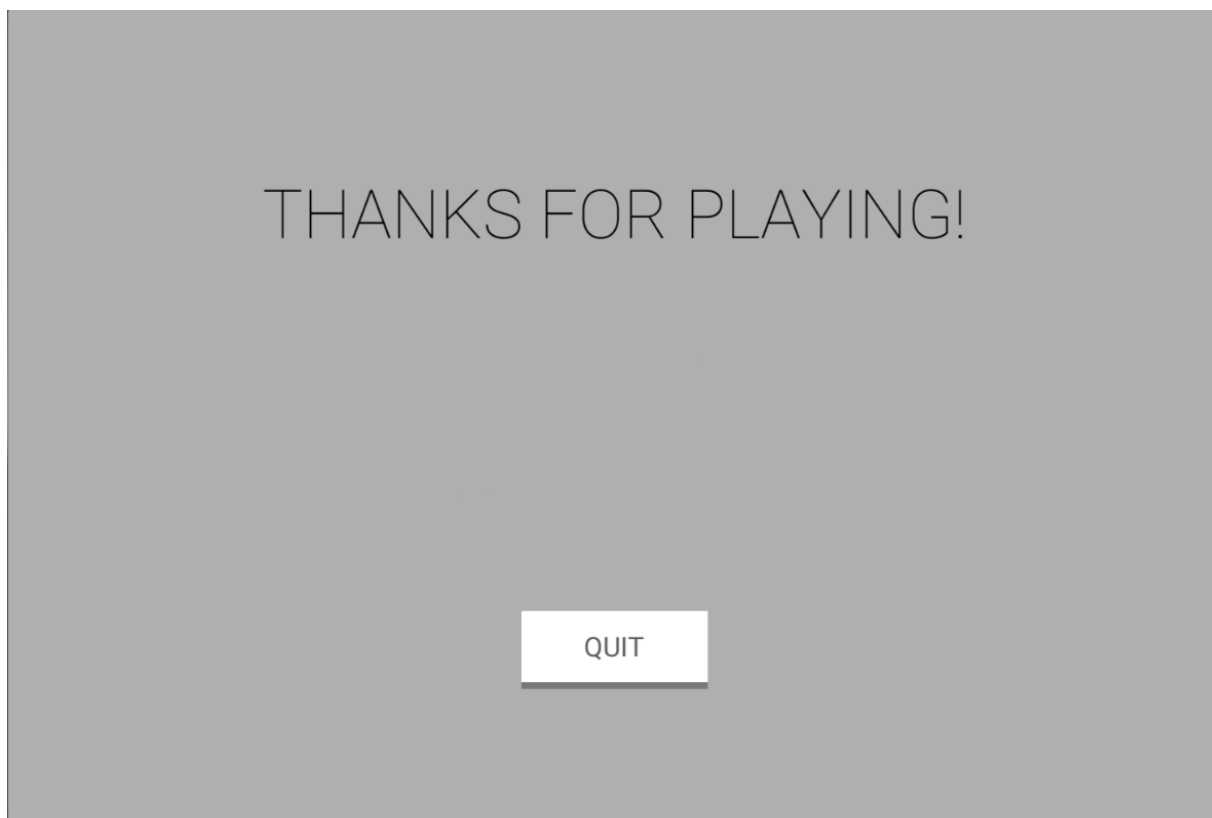


Рисунок 3.10 – Загальний вигляд інтерфейсного вікна типу «Кінець гри»

За результатами тестів було підтверджено, що комп'ютерна гра "Space Dust" працює згідно з усіма функціями, що були визначені в процесі розробки програмного модуля.

3.4 Висновок до розділу 3

У цьому розділі порівняли Unreal Engine, CryEngine та Unity 3D для розробки "Space Dust". Обрали Unity 3D для цього проекту. Разом з необхідними засобами було розроблено основний модуль гри з механікою, рівнями та перешкодами. Тестування підтвердило його працездатність.

Це важливий крок у розробці "Space Dust", створюючи основу для подальшої роботи над проектом.

ВИСНОВКИ

У цій бакалаврській дипломній роботі було успішно розроблено програмний модуль для 3D гри «Space Dust» з використанням ігрового рушія Unity. Проект включав дослідження жанру, створення ключових компонентів гри, таких як управління персонажем, генерація рівнів, перешкод, користувацький інтерфейс і саундтрек. Завдяки процедурній генерації рівнів та різноманітним перешкодам, гра забезпечує захопливий і нескінченний ігровий процес.

Під час розробки використовувалася компонентна архітектура Unity 3D та мова програмування C#, що забезпечило модульність та гнучкість коду. Для підтвердження працездатності гри та відповідності всім запланованим функціям було проведено тестування.

Цей проект є важливим кроком у розробці мобільних 3D раннерів та демонструє потенціал Unity 3D для створення захоплюючих та якісних ігрових продуктів. У майбутньому планується розширення функціоналу гри, додавання нових рівнів, персонажів і механік, а також оптимізація продуктивності для різних платформ.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Електронний ресурс. – Режим доступу:
https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D1%86%D0%B5%D0%B4%D1%83%D1%80%D0%BD%D0%B0_%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%86%D1%96%D1%8F
2. Educational discourse [Електронний ресурс]: Режим доступу:
https://journal-discourse.com/files/pdf/OD-41_7-9_-2022.pdf
3. Resident Evil 2 [Електронний ресурс]: Режим доступу:
https://store.steampowered.com/app/883710/Resident_Evil_2/
4. Portal 2 [Електронний ресурс]: Режим доступу:
<https://store.steampowered.com/app/400/Portal>
5. Stardew Valley [Електронний ресурс]: Режим доступу:
https://store.steampowered.com/app/413150/Stardew_Valley/?l=ukrainian&curator_clanid=7785583
6. Devil my cry 5 [Електронний ресурс]: Режим доступу:
https://store.steampowered.com/app/601150/Devil_May_Cry_5
7. Civilization V [Електронний ресурс]: Режим доступу:
https://store.steampowered.com/app/8930/Sid_Meiers_Civilization_V
8. Elder ring [Електронний ресурс]: Режим доступу:
https://store.steampowered.com/app/1245620/ELDEN_RING/
9. Basic procedures of the graphic pipeline [Електронний ресурс]: Режим доступу:
<https://ir.lib.vntu.edu.ua/handle/123456789/35568>

10. Blender [Электронный ресурс]: Режим доступа:
<https://www.blender.org/>

11. Subway Surfers [Электронный ресурс]: Режим доступа:
https://uk.wikipedia.org/wiki/Subway_Surfers

12. Temple Run [Электронный ресурс]: Режим доступа:
https://uk.wikipedia.org/wiki/Temple_Run

13. Collection of scientific [Электронный ресурс]: Режим доступа:
[papershttp://www.philosophy.univ.kiev.ua/uploads/editor/Files/Vydanna/Gumanitarni%20studii/%D0%93%D0%A120.pdf](http://www.philosophy.univ.kiev.ua/uploads/editor/Files/Vydanna/Gumanitarni%20studii/%D0%93%D0%A120.pdf)

14. dspace bent functions of 20 multi-valued logic [Электронный ресурс]:
 Режим доступа:

<http://dspace.op.edu.ua/jspui/bitstream/123456789/13205/1/%D0%A5%D0%B8%D0%BC%D0%B5%D0%BD%D0%BA%D0%BE%20%D0%9C%D0%B8%D1%85%D0%B0%D0%B9%D0%BB%D0%BE%20%D0%A0%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B0%20%D0%B3%D0%B5%D0%BD%D0%B5%D1%80%D0%B0%D1%82%D0%BE%D1%80%D0%B0%20%D0%BA%D0%BB%D1%8E%D1%87%D0%BE%D0%B2%D0%B8%D1%85%20%D0%BF%D0%BE%D1%81%D0%BB%D1%96%D0%B4%D0%BE%D0%B2%D0%BD%D0%BE%D1%81%D1%82%D0%B5%D0%B9%20%D0%BD%D0%B0%20%D0%BD1%86%D1%96%D0%B9%20%D0%B1%D0%B0%D0%B3%D0%B0%D1%82%D0%BE%D0%B7%D0%BD%D0%B0%D1%87%D0%BD%D0%BE%D1%97%20%D0%BB%D0%BE%D0%B3%D1%96%D0%BA%D0%B8.pdf>

15. Microsoft. C# documentation [Электронный ресурс]: Режим доступа:
<https://learn.microsoft.com/en-us/dotnet/csharp/>.

ДОДАТОК А
ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: Програмний модуль комп'ютерної гри «Space Dust»

Тип роботи: бакалаврська дипломна робота

(БДР, МКР)

Підрозділ кафедра комп'ютерних наук, ФІТА

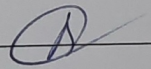
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність	<u>97,26%</u>	Схожість	<u>2.72%</u>
----------------	---------------	----------	--------------

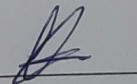
Аналіз звіту подібності (відмітити потрібне):

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку  Озеранський В.С.

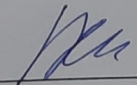
Ознайомлені з повним звітом подібності, який був згенерований системою Unicheck щодо роботи.

Автор роботи



Севастьянов І.В.

Керівник роботи



Денисюк В.О.

ДОДАТОК Б

Інструкція користувача

1. Відкриваєм рушій Unity
2. Запускаємо комп'ютерну гру під назвою «Space Dust»

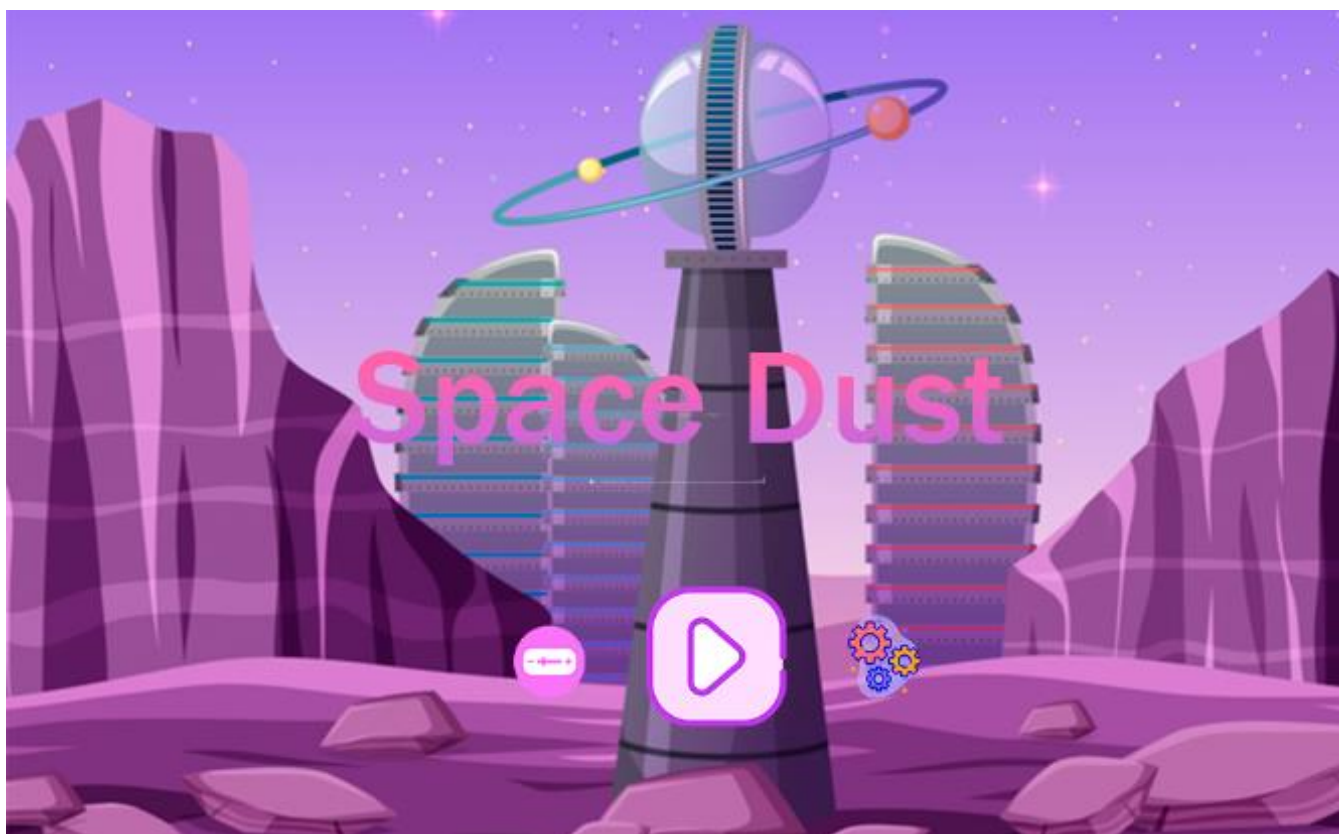


Рисунок Б.1 – Головне меню гри

3. Натискаємо на середню кнопку «Грати», щоб запустити гру



Рисунок Б.2 – Загальний вигляд комп'ютерної гри

4. За допомогою клавішах «A», «D» або «←», «→» персонаж переміщається вліво та вправо, «Пробіл» задіє стрибок персонажа.

ДОДАТОК В

Лістинг програми

```

void FixedUpdate()
{
    rb.AddForce(0, a_forwardMovement * Time.deltaTime);

    if (Input.GetKey("a") || Input.GetKey(KeyCode.LeftArrow))
    {
        rb.AddForce(-sidewaysMovement * Time.deltaTime, 0,
a_ForceMode_VelocityChange);
    }

    if (Input.GetKey("d") || Input.GetKey(KeyCode.RightArrow))
    {
        rb.AddForce(sidewaysMovement * Time.deltaTime, 0,
a_ForceMode_VelocityChange);
    }

    if (rb.position.y < -1f)
    {
        FindObjectOfType<GameManger>().EndGame();
    }
}
}
using UnityEngine;

public class PlayerCollision : MonoBehaviour
{
    public PlayerMovement movement;

    private void OnCollisionEnter(Collision collision)
    {
        if (collision.collider.tag == "Obstacle")
        {
            movement.enabled = false;
            FindObjectOfType<GameManger>().EndGame();
        }
    }
}
using UnityEngine;
using UnityEngine.SceneManagement;

public class Menu : MonoBehaviour
{
    public void StartGame()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }
}
using UnityEngine;
using UnityEngine.SceneManagement;

public class LevelComplete : MonoBehaviour
{
    public void LoadNextLevel()

```



```

        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }
}
using UnityEngine;
using UnityEngine.SceneManagement;
public class GameManager : MonoBehaviour
{
    bool gameHasEnded = false;
    public float restartDelay = 1f;
    public GameObject completeLevelUI;
    public void CompleteLevel()
    {
        completeLevelUI.SetActive(true);
    }
    public void EndGame()
    {
        if (gameHasEnded == false)
        {
            Debug.Log("Game Over");
            gameHasEnded = true;
            restart();
            Invoke("restart", restartDelay);
        }
    }

    public void restart()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().name);
    }
}
using UnityEngine;

public class FollowPlayer : MonoBehaviour
{
    public Transform player;
    public Vector3 offset;

    void Update()
    {
        transform.position = player.position + offset;
    }
}
using UnityEngine;

public class End : MonoBehaviour
{
    public GameManager gameManager;

    private void OnTriggerEnter(Collider other)
    {
        gameManager.CompleteLevel();
    }
}
using UnityEngine;

public class Credits : MonoBehaviour
{
    public void Quit()

    {
        Debug.Log("QUIT");
        Application.Quit();
    }
}

```

ДОДАТОК Г
Графічний матеріали

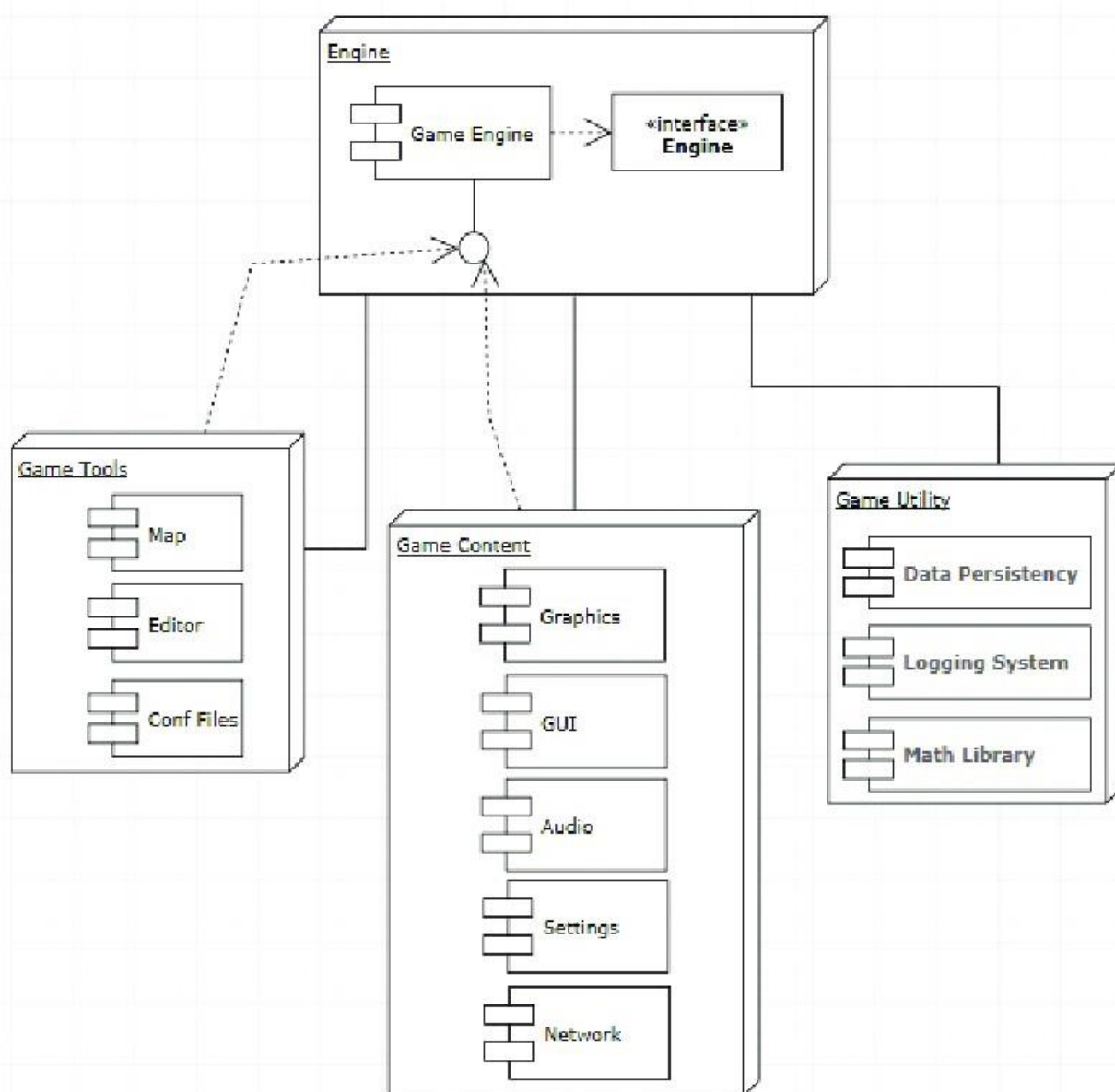


Рисунок Г.1 – Діаграма основних компонентів архітектури програмного модуля відеогри «Space Dust»

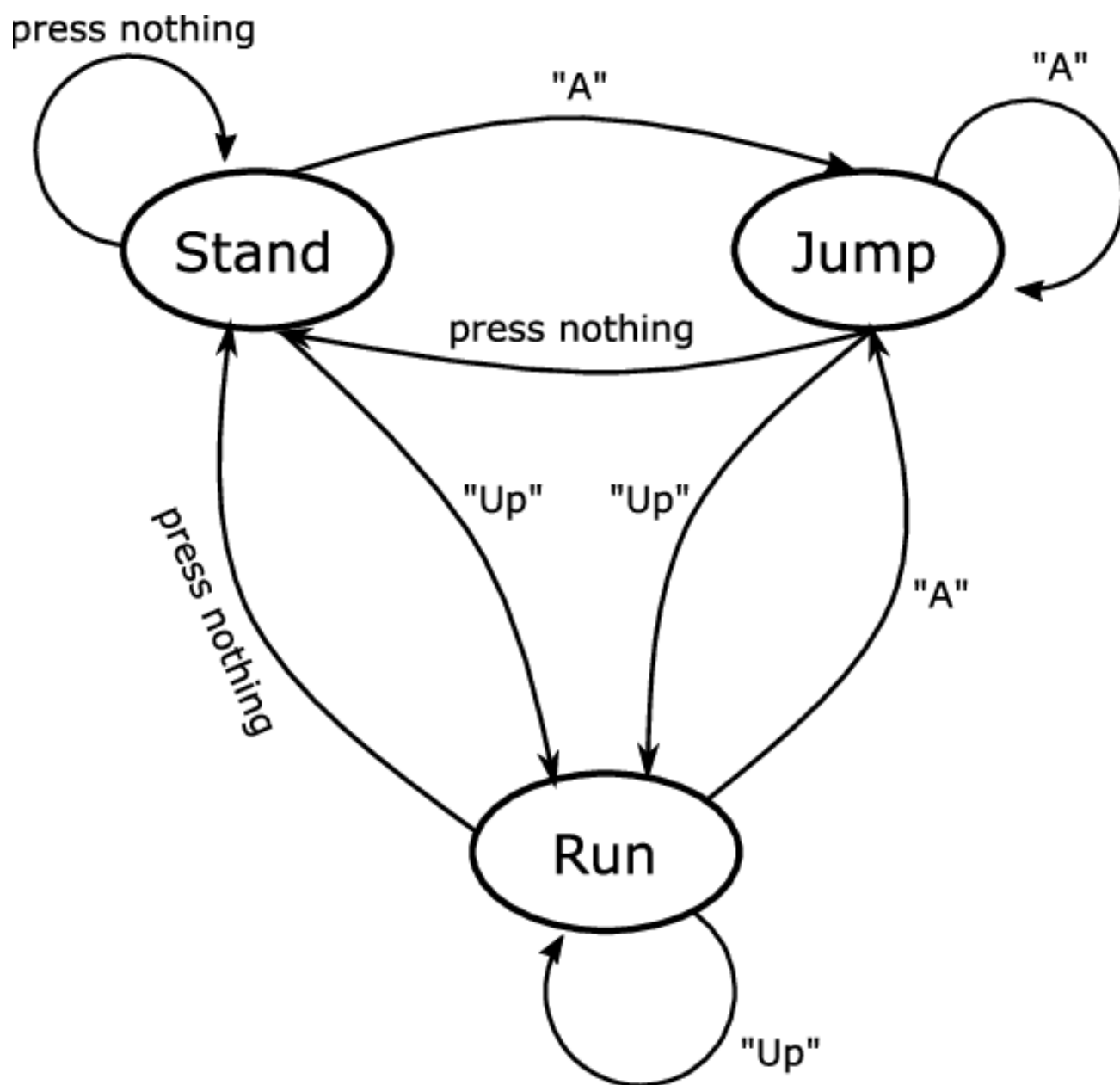


Рисунок Г.2 – Діаграма модель станів персонажа

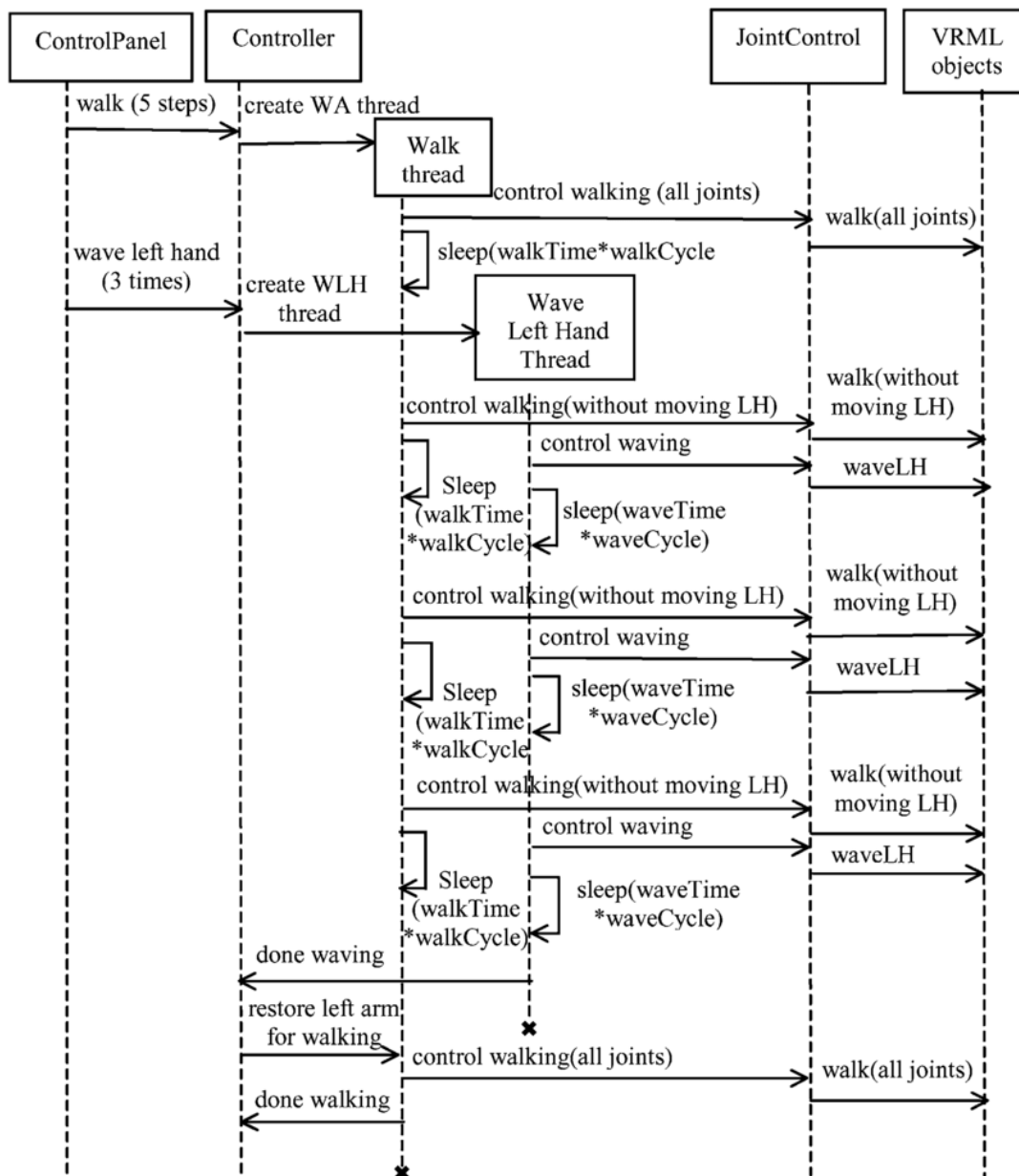


Рисунок Г.3 Діаграма взаємодії персонажа з перешкодою з додаванням анімаційного ефекту.

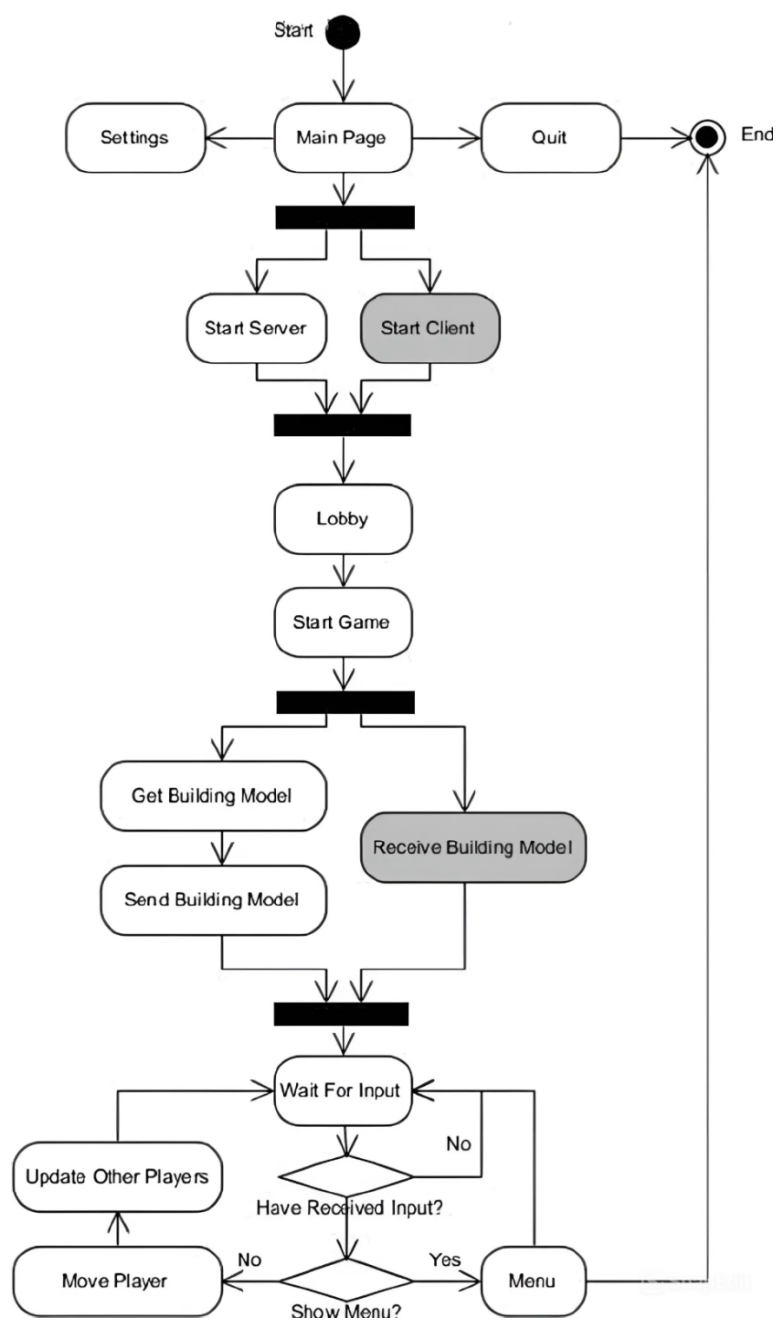


Рисунок Г.4 – Алгоритм роботи програмного модулю комп'ютерної гри «Space Dust»

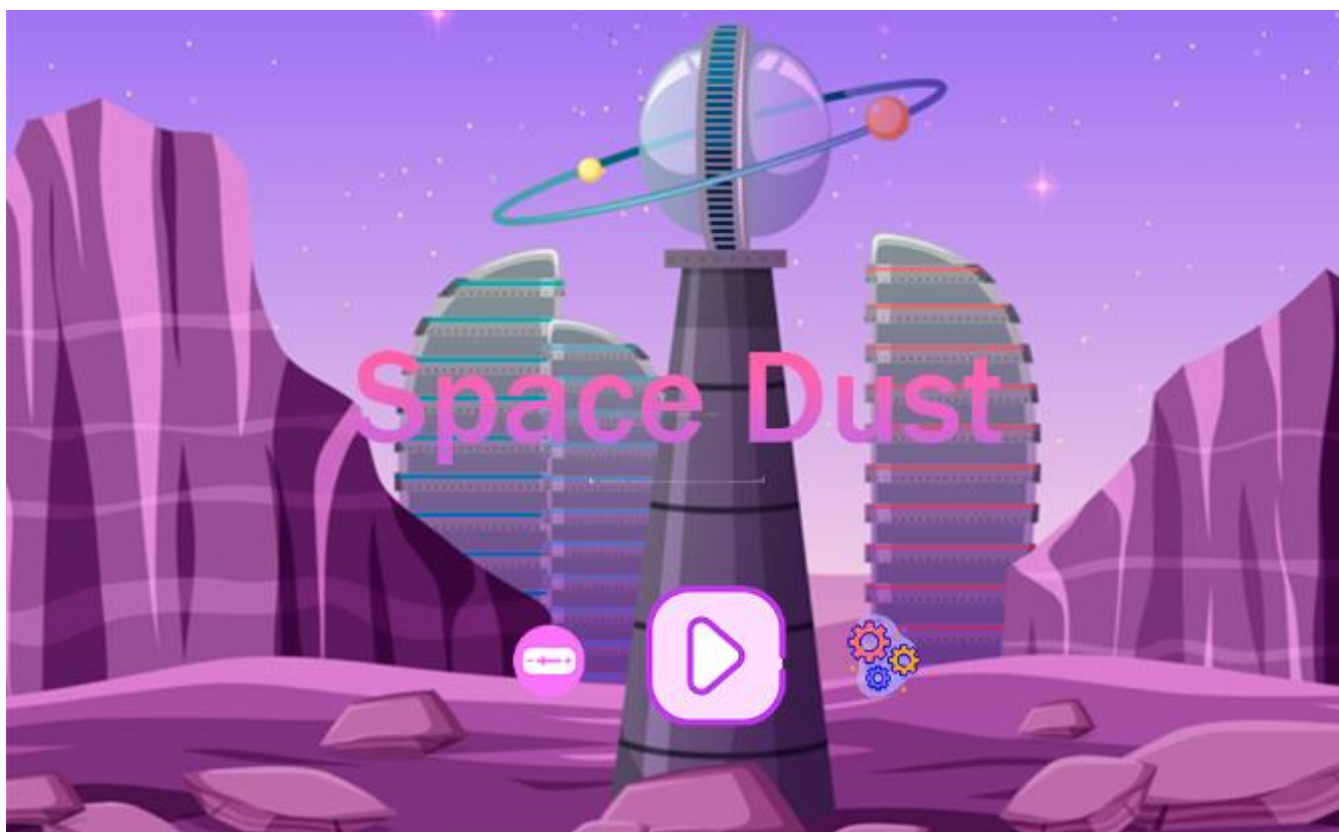


Рисунок Г.5 – Головне меню комп'ютерної гри «Space Dust»



Рисунок Г.6 – Перший рівень комп'ютерній грі "Space Dust".



Рисунок Г.7 – Другий рівень комп'ютерній грі "Space Dust".

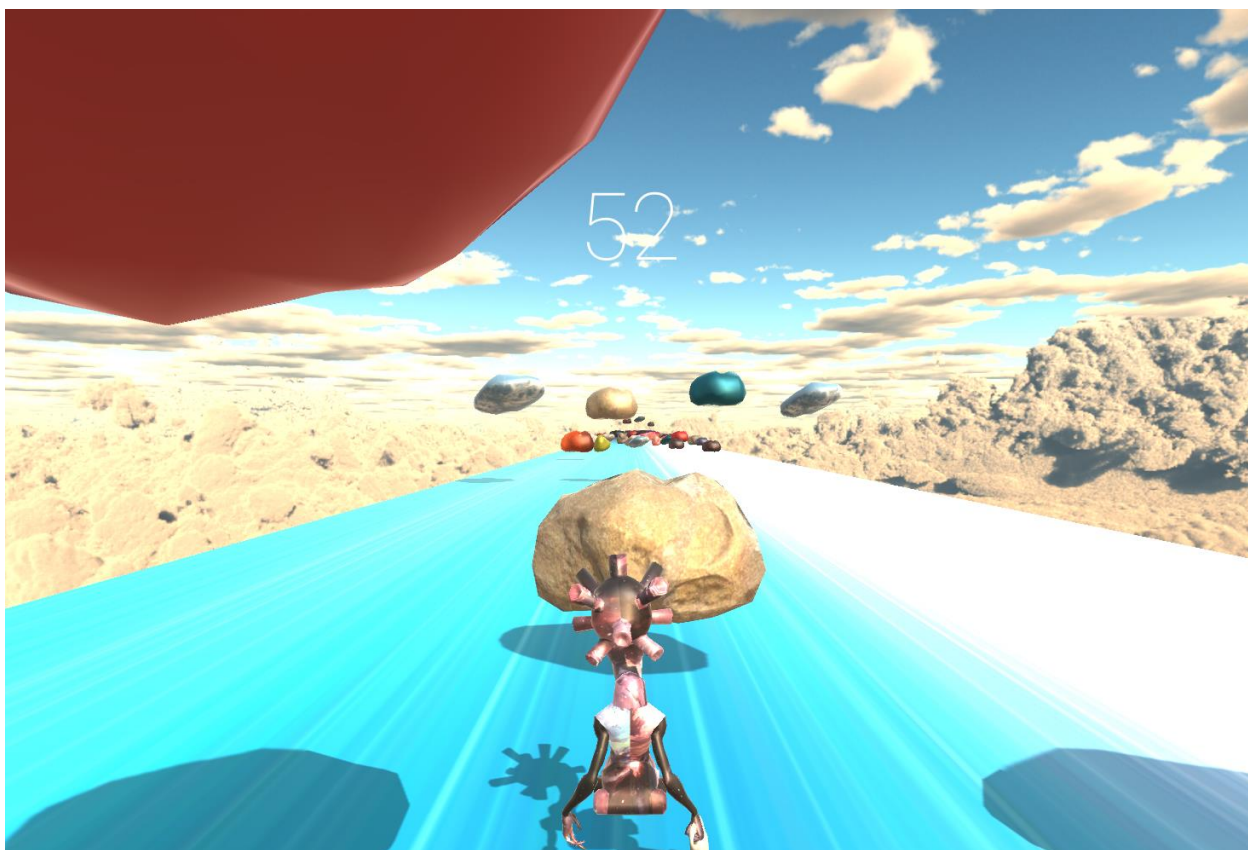


Рисунок Г.8 – Третій рівень комп'ютерній грі "Space Dust".