

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Розробка методу генерації ігрових предметів для гри
в жанрі survival simulator»

на здобуття освітнього ступеня магістра
зі спеціальності 121 Інженерія програмного забезпечення
(код, найменування спеціальності)
освітньо-професійної програми «Інженерія програмного забезпечення»
(назва)

*Кваліфікаційна робота містить результати власних досліджень. Використання
ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

_____ Олександр ЛУГИНА
(підпис)

Виконав: здобувач вищої освіти групи ПДМ-61

_____ Олександр ЛУГИНА

Керівник: _____ Олесь ДІБРІВНИЙ
доктор філософії (PhD)

Рецензент: _____ Ім'я, ПРИЗВИЩЕ
*науковий ступінь,
вчене звання*

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти Магістр

Спеціальність 121 Інженерія програмного забезпечення

Освітньо-професійна програма «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

_____ Ірина ЗАМРІЙ

« _____ » _____ 2023 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

_____ Лугина Олександр Володимирович

1. Тема кваліфікаційної роботи: «Розробка методу генерації ігрових предметів для гри в жанрі survival simulator»

керівник кваліфікаційної роботи Олесь ДІБРІВНИЙ доктор філософії (PhD),

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» жовтня 2023 р. №145.

2. Строк подання кваліфікаційної роботи «29» грудня 2023 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, методи генерації ігрових предметів.

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

1.Дослідження розміщення ігрових предметів в іграх жанрі survival simulator.

2.Аналіз методи генерації ігрових предметів застосування в грі жанрі survival simulator.

3.Розробка вимог до власного методу генерації ігрових предметів.

5. Перелік графічного матеріалу: *презентація*

1. Мета, об'єкт, предмет дослідження
2. Аналіз існуючих методів автоматизованої генерації ігрових предметів
3. Сполучення лінійного конгруентного методу та шуму Перліна
4. Результат генерації
5. Порівняльний аналіз генератора псевдовипадкових чисел

6. Дата видачі завдання «19» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	19.10-05.11.23	
2	Аналіз існуючих методів вирішення поставленої задачі.	06.11-12.11.23	
3	Аналіз існуючих методів вирішення поставленої задачі.	13.11-19.11.23	
4	Побудова математичної функціональної моделей випадкової генерації ігрових предметів для гри в жанрі survival simulator	20.11-26.11.23	
5	Розробка програмного забезпечення для створення власного методу генерації ігрових предметів для гри в жанрі survival simulator.	27.11-03.12.23	
6	Застосування власного методу в програмному забезпеченні для генерації предметів для гри в жанрі survival simulator.	04.12-10.12.23	
7	Оформлення роботи: вступ, висновки, реферат	11.12-20.12.23	
8	Розробка демонстраційних матеріалів	21.12-29.12.23	

Здобувач вищої освіти

(підпис)

Олександр ЛУГИНА

Керівник

кваліфікаційної роботи

(підпис)

Олесь ДІБРІВНИЙ

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 72 стор., 1 табл., 10 рис., 20 джерел.

Мета роботи – підвищення якості генерації ігрових предметів за рахунок використання методу генерації псевдовипадкових чисел.

Об'єкт дослідження – процес генерації ігрових предметів.

Предмет дослідження – методи генерації ігрових предметів.

Короткий зміст роботи: Досліджено різноманітні підходи до генерації предметів, такі як випадкове розташування, використання шуму Перліна, генератора псевдовипадкових чисел та інші. Проаналізовано переваги та недоліки кожного методу з метою розробки ефективного алгоритму генерації, спрямованого на покращення геймплею та виживання у віртуальному середовищі.

КЛЮЧОВІ СЛОВА: МЕТОДИ ГЕНЕРАЦІЇ, ІГРОВІ ПРЕДМЕТИ,
SURVIVAL SIMULATOR

ABSTRACT

Text part of the master's qualification work:

72 pages, 10 pictures, 1 table, 20 sources.

The purpose of the work – Improving the quality of the generation of game items due to the use of the method of generating pseudo-random numbers.

Object of research – The process of generating game items.

Subject of research – methods of generating game items.

Summary of the work: Various approaches to item generation have been explored, such as random placement, using Perlin noise, a pseudorandom number generator, and others. The advantages and disadvantages of each method were analyzed in order to develop an effective generation algorithm aimed at improving gameplay and survival in a virtual environment.

KEYWORDS: GENERATION METHODS, GAME ITEMS, SURVIVAL SIMULATOR

ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ.....	13
1.1. ПОНЯТТЯ ІГРОВИХ ПРЕДМЕТІВ І ЖАНР SURVIVAL SIMULATOR.....	13
1.2. ЗАГАЛЬНІ ПРИНЦИПИ РОЗМІЩЕННЯ ІГРОВИХ ПРЕДМЕТІВ.....	16
1.3. ОГЛЯД ТА АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ ГЕНЕРАЦІЇ ІГРОВОГО ПРЕДМЕТІВ.....	17
1.4. АНАЛІЗ ІСНУЮЧИХ МАТЕМАТИЧНИХ МЕТОДІВ, МОДЕЛЕЙ ТА АЛГОРИТМІВ ДЛЯ ПІДВИЩЕННЯ ЯКОСТІ ГЕНЕРАЦІЇ ІГРОВИХ ПРЕДМЕТІВ.....	24
1.5. АНАЛІЗ ЗАСОБІВ ПРОГРАМНОЇ ІНЖЕНЕРІЇ ДЛЯ ПІДВИЩЕННЯ ЯКОСТІ ГЕНЕРАЦІЇ ІГРОВИХ ПРЕДМЕТІВ.....	25
РОЗДІЛ 2. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ РОЗМІЩЕННЯ ІГРОВИХ ПРЕДМЕТІВ ТА ЗАСОБІВ ПІДВИЩЕННЯ ЯКОСТІ ГЕНЕРАЦІЇ ІГРОВИХ ПРЕДМЕТІВ.....	30
2.1. ОСНОВНІ МЕТОДИ ГЕНЕРАЦІЇ ІГРОВИХ ПРЕДМЕТІВ ЖАНРУ SURVIVAL SIMULATORS.....	30
2.2. МАТЕМАТИЧНІ МЕТОДИ ВИКОРИСТАННЯ СТАТИЧНОЇ ГЕНЕРАЦІЇ ДЛЯ ІГРОВИХ ПРЕДМЕТІВ.....	31
2.3. МЕТОДИ МАШИННОГО НАВЧАННЯ ДЛЯ ГЕНЕРАЦІЇ ІГРОВИХ ПРЕДМЕТІВ.....	33
2.4. АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ГЕНЕРАТОРА ПСЕВДОВИПАДКОВИХ ЧИСЕЛ.....	37
РОЗДІЛ 3. РОЗРОБКА МЕТОДУ ГЕНЕРАЦІЇ ІГРОВИХ ПРЕДМЕТІВ ДЛЯ ГРИ В ЖАНРІ SURVIVAL SIMULATOR.....	44
3.1. ВИБІР ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ.....	44
3.2. АНАЛІЗ НОВОГО МЕТОДУ ГЕНЕРАЦІЇ ІГРОВИХ ПРЕДМЕТІВ.....	45
3.3. РЕАЛІЗАЦІЯ СИСТЕМИ.....	53
РОЗДІЛ 4. ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ НОВОГО МЕТОДА ГЕНЕРАЦІЇ ІГРОВИХ ПРЕДМЕТІВ.....	55
4.1. МЕТОДИКА ПРОВЕДЕННЯ ЕКСПЕРИМЕНТУ.....	55
4.2. РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТУ ДОСЛІДЖЕННЯ.....	56
ВИСНОВКИ.....	59
ПЕРЕЛІК ПОСИЛАНЬ.....	60
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	62

ВСТУП

Актуальність теми

Гри в жанрі survival simulator здобувають надзвичайну популярність серед геймерської аудиторії, адже вони пропонують захопливий іммерсивний досвід виживання в умовах невизначеності та атмосфери постапокаліптичного світу. Однак ключовим аспектом, що визначає глибину ігрового процесу, є якість та різноманітність ігрових предметів, які гравець знаходить та використовує.

В сучасних умовах розробка відмінного генеративного методу для створення ігрових предметів стає актуальною завдяки розширенню технологічних можливостей, зростанню очікувань гравців та постійній конкуренції на ринку відеоігор. Реалістична та динамічна генерація предметів може покращити геймплей, зробити гру більш цікавою та додати елемент непередбачуваності.

Застосування в роботі методів генерації, таких як використання генераторів псевдовипадкових чисел, математичні моделі та їх поєднання, відкриває перспективу для створення реалістичних, унікальних та адаптивних ігрових предметів. Подальша розробка та оптимізація такого методу може значно збільшити привабливість гравців до ігор в жанрі survival simulator та підвищити їхню конкурентоспроможність на ринку розважальних відеоігор.

Актуальність теми обґрунтовується наступними факторами:

- Висока конкуренція на ринку відеоігор заставляє розробників шукати новаторські рішення для відзначення своїх продуктів. Генерація ігрових предметів, яка робить гру унікальною та цікавою, може стати важливим конкурентним перевагою.
- Розробка методу генерації ігрових предметів може також знайти застосування в інших областях, таких як віртуальна реальність, симуляції для навчання та архітектурне моделювання, що розширює область впливу теми.
- Актуальність теми визначається також наявністю викликів у розробці ефективного методу генерації. Вирішення цих викликів може призвести до значних покращень у сфері генеративного дизайну ігрових світів.

Мета і завдання дослідження

Мета роботи: підвищення якості генерації ігрових предметів за рахунок використання методу генерації псевдовипадкових чисел.

Об'єкт дослідження: процес генерації ігрових предметів.

Предмет дослідження: методи генерації ігрових предметів.

Методи дослідження

У процесі виконання магістерської роботи були використані наступні методи дослідження:

Теоретичні методи

- Аналіз літературних джерел з теми дослідження
- Аналіз існуючих методів та засобів підвищення якості генерації ігрових предметів

На першому етапі дослідження було отримано теоретичні відомості з теми генерації ігрових предметів. Для цього було проаналізовано літературні джерела з цієї тематики. Зокрема, було вивчено наступні питання:

- Загальні принципи розміщення ігрових предметів
- Методологія генерації об'єктів
- Існуючі методи та засоби підвищення якості генерації ігрових предметів

На другому етапі дослідження було проведено експериментальне дослідження якості розробленого методу генерації. Для цього було використано реальні дані і реалізовано цей метод в опенсорс проект.

Експеримент проводився за наступною схемою:

- Зібрані дані були розподілені на два набори: навчальний та тестовий. Навчальний набір використовувався для тренування розробленого методу, а тестовий — для оцінки його ефективності та узагальнюючої здатності до нових даних.

- Використовуючи навчальний набір, було проведено тренування розробленого методу генерації. Алгоритм був адаптований та оптимізований на основі зібраних даних з метою покращення якості генерації.

- Розроблений метод було інтегровано в open source проект, що дозволило провести експериментальне тестування в реальному ігровому середовищі. Це сприяло збору важливих даних та забезпечило можливість перевірки адаптивності методу до різних умов.

Унікальність даного дослідження проявляється у використанні двох різних методів генерації, зокрема методу шуму Перліна та лінійного конгруентного методу, з метою об'єднання їх результатів. В процесі експерименту встановлено, що розроблений метод генерації ігрових предметів, який використовує обидва зазначені методи, суттєво підвищує якість розміщення ігрових об'єктів.

Результати дослідження свідчать про те, що використання комбінації методу шуму Перліна та лінійного методу дозволяє покращити якість генерації ігрових предметів, особливо у відношенні до ефективного розташування об'єктів у грі. Розроблений метод має потенціал використовуватися для оптимізації процесу генерації ігрових предметів, зокрема в балансуванні розміщення та створенні гармонійного геймплею в іграх жанру survival simulator, а також в інших жанрах

Апробація результатів та публікації

Тези доповідей:

Дібрівний О.А., Лугина О.В. Оптимізація та Використання Рандомізації в Генерації Вмісту Гри: Сучасні Підходи та Вплив на Геймплей// Всеукраїнська науково-технічна конференція "Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу". – Київ // СЕКЦІЯ №3 подана до друку ДУІКТ : 2023.

Теоретична, методична та практична значущість отриманих результатів

На методичному рівні отримані результати слугують цінним внеском у покращення підходів до генерації ігрових об'єктів. Використання методу шуму Перліна та лінійного методу разом виявилось ефективною стратегією для створення більш динамічного та реалістичного ігрового світу. Такий підхід може служити модельним прикладом для подальших досліджень в області генеративного дизайну відеоігор.

Практична значущість полягає в тому, що розроблений метод генерації ігрових предметів може бути використаний розробниками відеоігор для підвищення якості геймплею в жанрі survival simulator та подібних грах. Отримані результати дозволяють оптимізувати розташування ігрових об'єктів, покращуючи враження гравців та забезпечуючи більш глибокий та захопливий ігровий досвід. Такий метод може бути використаний для створення ігор різних жанрів, де генерація ігрових об'єктів відіграє ключову роль у формуванні геймплею.

1 ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

1.1. Поняття ігрових предметів і жанр survival simulator

Для розуміння методів створення ігрових предметів для ігор в жанрі survival simulator для початку необхідно визначити, які характеристики їх створення, що для цього необхідно та визначити загальні характеристики жанру.

Для розуміння методів створення рівнів у жанрі survival simulator важливо враховувати його загальні характеристики. У таких іграх великий акцент робиться на непередбачуваності та випадковості, що надає динамічності геймплею. Гравці вправляються у керуванні ресурсами, такими як їжа та вода, та ведуть ефективну стратегію виживання. Будівництво та крафтинг грають важливу роль, дозволяючи гравцям створювати структури та інструменти для поліпшення умов виживання. Присутність ворогів та різноманітних небезпек додає аспекти боротьби за виживання та ризик в середовищі гри. Отже, розробка рівнів вимагає урахування цих елементів для створення захопливого та високоякісного ігрового досвіду для гравців.

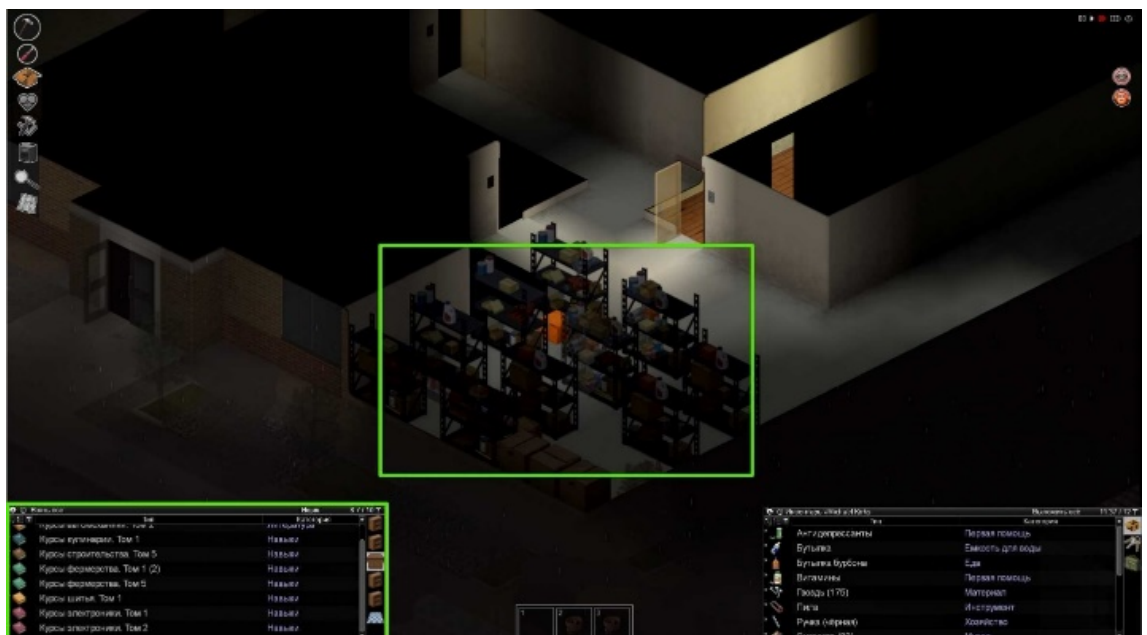


Рис. 1.1. Приклад ігрових предметів для ігор survival simulator

У іграх в жанрі survival simulator гравці повинні виживати в небезпечному середовищі, збираючи ресурси, виготовляючи предмети та борються з ворогами.

Для того, щоб зробити гру більш захоплюючою та реалістичною, необхідно забезпечити різноманітність ігрового світу, включаючи різноманітність ігрових предметів.

Генерація ігрових предметів є трудомістким процесом, який вимагає від розробників значних зусиль та часу. Для автоматизації цього процесу можна використовувати різні методи, зокрема, генератори псевдовипадкових чисел.

Для успішного розвитку методу генерації ігрових предметів важливо врахувати основні риси жанру survival simulator. У таких іграх, гравець стикається з різноманітними викликами, пов'язаними із виживанням у небезпечному середовищі.

Жанр визначається великим акцентом на непередбачуваності та випадковості. Гравці мають вміло керувати ресурсами, такими як їжа та вода, і розробляти стратегії для ефективного виживання. Основними компонентами геймплею є будівництво та крафтинг, які надають можливість створювати структури та інструменти для поліпшення умов виживання.

Збалансованість між непередбачуваністю гри, управлінням ресурсами та можливостями будівництва роблять цей жанр захопливим для гравців. Розробка методу генерації ігрових предметів повинна враховувати ці особливості, створюючи різноманітний та динамічний ігровий світ.

Провідна роль у вдосконаленні геймплею належить генерації ігрових предметів. У жанрі survival simulator різноманітність ігрового світу є ключем до захопливого досвіду гравців. Це включає в себе створення різних предметів, які можуть впливати на виживання у грі.

Генерація ігрових предметів є важливим етапом у процесі розробки, але це також трудомісткий процес, що вимагає від розробників значних зусиль та часу. Один із можливих підходів - використання генераторів псевдовипадкових чисел, які можуть забезпечити елемент випадковості та унікальності в ігровому світі.

У вигляді ключового аспекту розробки методу генерації ігрових предметів, важливо врахувати взаємодію гравців з цими предметами. Гравці в жанрі survival

simulator повинні мати можливість відчутти вплив ігрових предметів на їхні стратегії виживання.

Створення предметів, які стимулюють та вдосконалюють геймплей, сприяє більш глибокому поглибленню гравців у віртуальний світ. Це може включати в себе різноманітність функцій і властивостей предметів, які дозволяють гравцям використовувати їх для різних стратегій виживання.

Генерація ігрових предметів за допомогою генераторів псевдовипадкових чисел відкриває безліч можливостей для створення унікального ігрового досвіду. Ці інструменти дозволяють впроваджувати елемент випадковості, необхідний для створення різноманітності предметів та сценаріїв в грі.

Важливо ретельно налаштовувати параметри генераторів, щоб забезпечити адекватний рівень складності та реалізму. Це відкриває можливість для експериментів та постійного оновлення геймплею, роблячи його більш цікавим для гравців.

Реалізм грає ключову роль у генерації ігрових предметів для жанру survival simulator. Гравці повинні відчувати, що створені предмети відповідають вимогам виживання в небезпечному середовищі гри. Реалістичність стимулює іммерсію гравців та сприяє більш глибокому зануренню у гру.

При використанні генераторів псевдовипадкових чисел, важливо враховувати не лише аспекти випадковості, але і контекстуальність результатів. Забезпечення логічності та зв'язку між предметами і елементами гри підвищує аутентичність геймплею.

Додатковим етапом вдосконалення генерації ігрових предметів може бути використання методів машинного навчання. Алгоритми навчання можуть аналізувати поведінку гравців, їхні вибори та стратегії, щоб адаптувати генерацію предметів до конкретного стилю гри кожного користувача.

Машинне навчання може покращити персоналізацію геймплею, забезпечуючи унікальний досвід для кожного гравця. Використання цих технологій дозволить створювати більш адаптовані та цікаві ігрові об'єкти.

1.2. Загальні принципи розміщення ігрових предметів

У грах жанру survival simulator, де основний фокус спрямований на реалістичному виживанні гравця в екстремальних умовах, принципи розміщення ігрових предметів є критичними для створення іммерсивного та захоплюючого досвіду. Розглянемо ці принципи детальніше:

Ресурсна рівновага:

Забезпечення балансу між доступними ресурсами визначає, наскільки гравець постійно стикається з викликами виживання. Це означає не лише розміщення їжі та води, а й врахування матеріалів для будівництва, лікарських засобів та інших важливих ресурсів.

Система нагромадження ресурсів:

Однією з ключових особливостей є можливість гравця збирати, обробляти та зберігати ресурси. Наявність складського простору, можливість створювати контейнери або виготовляти інструменти для перенесення ресурсів стають важливими факторами геймплею.

Динамічність розміщення:

Змінюючи місця розташування ресурсів з часом або в залежності від обставин, гра може стимулювати гравця постійно досліджувати світ та адаптуватися до нових умов, що підвищує ігрову динаміку.

Аутентичність місцевості:

Детальна проработка областей гри, з урахуванням реалістичних характеристик та особливостей природного середовища, додає іммерсивність та глибину геймплею.

Взаємодія з оточенням:

Можливості взаємодії гравця з оточенням — це ключовий елемент. Наприклад, можливість полювання на тварин для отримання їжі, пошук притулків від погодніх умов чи знаходження джерел води — всі ці аспекти важливі для успішного виживання.

Необхідність прийняття рішень:

Створення ситуацій, де гравець повинен вибирати між різними ресурсами або вирішувати, як їх використовувати, створює динаміку та внесок стратегічного планування в ігровий процес.

Ці принципи є важливими для створення неперевершеного геймплею в survival simulator, де кожне рішення гравця має вагомий наслідок для його виживання в жорстких умовах віртуального світу.

Генерація ігрових предметів є трудомістким процесом, який вимагає від розробників значних зусиль та часу. Для автоматизації цього процесу можна використовувати різні методи, зокрема, генератори псевдовипадкових чисел.



Рис.1.2 Алгоритм генерації ігрових предметів

Відповідно, для автоматизації створюваного рівня і прискорення його створення можна деякі етапи алгоритму зробити генерованими та автоматичними. Тож на далі розглянемо методи генерації.

1.3. Огляд та аналіз літературних джерел генерації ігрового предметів

Метод генерації ігрових предметів є важливим аспектом розробки ігор у жанрі survival simulator. Він дозволяє створювати велику кількість різноманітних предметів, що забезпечує різноманітність ігрового процесу. У цьому огляді ми проаналізуємо літературні джерела з методу генерації ігрових предметів для гри в жанрі survival simulator.

У роботі «Aycok J. Procedural Content Generation. Retrogame Archeology»[1] автор розглядає різні методи генерації ігрового контенту, у тому числі й метод генерації ігрових предметів. Автор наводить приклади використання цього методу в різних іграх і обговорює його переваги та недоліки.

Автори «Cassol V. J., Marson F. P., Musse S. R. Procedural Hair Generation»[2] представляють метод генерації волосся для персонажів ігрового світу. Метод заснований на використанні генетичних алгоритмів і дозволяє створювати реалістичне волосся різної довжини, кольору та форми.

У роботі «Cheng W. Procedural Enemy Waves. Procedural Generation in Game Design.» [3] автор представляє метод генерації хвиль ворогів для ігор у жанрі survival simulator. Метод заснований на використанні параметрів гри, таких як рівень складності та прогрес гравця, для створення хвиль ворогів різної складності.

У книзі "Story and Plot Generation. Procedural Generation in Game Design" автора Б. Кибартаса [4] досліджуються аспекти створення сюжетів та сценаріїв в процесі генерації ігрових контентів. Методи процедурної генерації використовуються для створення цікавих і варіативних сюжетних ліній. У даному дослідженні висвітлені алгоритми та стратегії, які дозволяють автоматизовано генерувати різноманітні і влучні сюжети в рамках гри.

Автор «Cook D. M. Ethical Procedural Generation. Procedural Storytelling in Game Design.» [5] обговорює етичні проблеми, пов'язані з використанням методу генерації ігрового контенту. Автор наводить приклади того, як цей метод може використовуватися для створення образливого або дискримінаційного контенту, і пропонує рекомендації щодо його етичного використання.

У роботі «Cook M. Ethical Procedural Generation. Procedural Generation in

Game Design.» [6] автор розглядає різні методи генерації ігрового контенту та обговорює їхні переваги та недоліки. Автор також наводить приклади використання цих методів у різних іграх.

Автор книги "Artificial Intelligence. Procedural Generation in Game Design" М. Р. Джонсон [7] досліджує застосування штучного інтелекту у процесі створення ігрових дизайнів. Зокрема, надається увага методам процедурної генерації для розробки ігрових елементів. В основі дослідження лежить використання різноманітних алгоритмів та стратегій, які дозволяють автоматизовано створювати контент гри відповідно до визначених параметрів

Автор «Dormans J. Cyclic Generation. Procedural Generation in Game Design» [8] представляє метод циклічної генерації для створення ігрових рівнів. Метод заснований на використанні набору правил для створення послідовності ігрових елементів, які повторюються.

У книзі «Level Design III. Procedural Generation in Game Design» [9], автором якої є Е. Ган, розглядається тема рівнів гри. Зокрема, методи процедурної генерації використовуються для створення ігрових рівнів. Дослідження ґрунтується на використанні набору правил, щоб синтезувати послідовність ігрових елементів, формуючи рівні відповідно до визначених принципів.

Автор "Level Design III. Procedural Generation in Game Design" Е. Ган [9] розглядає аспекти створення ігрових рівнів, зосереджуючись на використанні методів процедурної генерації. Дослідження базується на використанні набору правил для автоматизованого формування послідовності ігрових елементів, які створюють рівні відповідно до певних принципів та критеріїв.

У книзі "Procedural Logic. Procedural Generation in Game Design" від автора Б. Кейна [10] досліджуються принципи та використання процедурної логіки у створенні ігрового дизайну. Автор розглядає методи, які базуються на використанні процедур для логічного синтезу ігрових елементів. У даному дослідженні аналізуються алгоритми та стратегії процедурної логіки, які використовуються для автоматизованого створення логічно структурованих ігрових контентів.

У розділі "Content Tools Case Study" книги "Procedural Generation in Game Design" автора К. Ауве [11] розглядається конкретний випадок використання інструментів для створення ігрового контенту. Автор приводить аналіз використання засобів генерації контенту у практиці, зосереджуючись на власному випадку вивчення. В даному випадку висвітлені методи та підходи до застосування інструментів для створення різноманітного та цікавого ігрового контенту.

У відділі "Aesthetics in Procedural Generation" книги "Procedural Generation in Game Design" Л. Велтон [12] розглядається тема естетики у процесі процедурної генерації. Автор досліджує, як аспекти естетики впливають на створення ігрового контенту за допомогою процедурної генерації. У даному розділі аналізуються засоби і прийоми, що використовуються для досягнення естетичних цілей у процесі створення геймдизайну за допомогою процедурної генерації.

У розділі "Automated Game Tuning" книги "Procedural Generation in Game Design" автора А. Ісаксена [13] розглядається тема автоматизованої настройки гри. Автор досліджує методи та інструменти, що використовуються для автоматичної настройки геймплею та ігрового досвіду. У даному розділі аналізуються алгоритми та стратегії, які застосовуються для автоматизованої оптимізації параметрів гри з метою досягнення балансу та покращення взаємодії гравця з грою.

У книзі "Procedural Content Generation in Games" авторів Н. Шейкера, Дж. Тогеліуса та М. Нельсона [14], досліджується тема процедурного генерування контенту в іграх. Автори розглядають різні аспекти та методи процедурної генерації в ігровій індустрії. У даній книзі аналізуються алгоритми, стратегії та виклики, пов'язані із застосуванням процедурного генерування для створення різноманітного і цікавого контенту в іграх.

У дослідженні "Survey on Procedural Game Generation", опублікованому у журналі "Bulletin of the South Ural State University. Series Computational Mathematics and Software Engineering" [15] у 2015 році, проведено огляд процедурної генерації в ігровій індустрії. Стаття охоплює різні аспекти та підходи до використання процедурної генерації у гральних додатках. Автори дослідження розглядають алгоритми, техніки та виклики, що стосуються використання процедурної генерації

для створення геймплей та ігрового контенту.

Розділ «Dormans J. Cyclic Generation. Procedural Generation in Game Design» [8] визначає актуальність методу циклічної генерації у створенні ігрових рівнів та підкреслює важливість ефективного використання набору правил для створення повторюючихся, але цікавих ігрових елементів. Однією з ключових проблем є необхідність розробки інтелектуальних стратегій та алгоритмів для створення різноманітних циклічних ігрових урівнів, що сприяє геймплейній узгодженості та викликає зацікавленість гравців. Важливим аспектом є забезпечення балансу між повторюваністю та унікальністю в контексті генерації ігрових рівнів. Таким чином, проблематика методу циклічної генерації виявляється в необхідності вдосконалення алгоритмів для створення узгоджених та захопливих ігрових урівнів, що відповідають вимогам геймдизайну.

У книзі "Procedural Generation in Game Design," під редакцією Т. Шорта та Т. Адамса [16], розглядаються різні аспекти та підходи до процедурної генерації в геймдизайні. Автори та редактори аналізують методи, алгоритми та приклади використання процедурної генерації для створення ігрового контенту. Книга надає огляд сучасних підходів до цієї теми та висвітлює роль процедурної генерації у розвитку геймдизайну.

У книзі "Generative Artwork. Procedural Generation in Game Design" автора Л. Шмідта [17], розглядається тема генеративного мистецтва у контексті процедурної генерації в геймдизайні. Автор аналізує методи та підходи до створення мистецьких творів за допомогою процедурної генерації. У даному розділі освітлені аспекти, пов'язані із застосуванням процедурної генерації для творення візуальних елементів та художнього контенту в геймдизайні.

У розділі "Algorithms and Approaches" книги "Procedural Generation in Game Design" від автора Б. Баклу [18], досліджуються різноманітні алгоритми та підходи, що використовуються у процедурній генерації в геймдизайні. Автор аналізує методи створення ігрового контенту за допомогою різних алгоритмів та підходів. У даному розділі розглядаються різні стратегії та методи, які застосовуються для ефективної процедурної генерації в ігровій індустрії.

У розділі "Poetry Generation" книги "Procedural Generation in Game Design" автора Г. Таффса [19], досліджується використання процедурної генерації у створенні поезії в геймдизайні. Автор аналізує методи та підходи до використання процедурної генерації для творення літературних творів. У даному розділі висвітлюються аспекти, пов'язані із застосуванням процедурної генерації для творчого процесу створення поетичного контенту в іграх.

У розділі "Procedural Content Generation" книги "Level Design" авторки Гілліан Сміт [20], досліджується тема процедурної генерації контенту. Авторка аналізує методи та підходи до створення ігрового контенту за допомогою процедурної генерації в контексті рівнів гри. У даному розділі розглядаються аспекти, пов'язані із застосуванням процедурної генерації для створення різноманітного та цікавого ігрового вмісту.

Після ретельного огляду літератури з проблематики генерації ігрового контенту, можна визначити ключові аспекти та тенденції в цьому напрямку.

Генерація історії та сюжету. Процедурна генерація в ігровому дизайні, [4] вказує на актуальність проблеми генерації ігрових предметів у відношенні до створення ігрових сюжетів та історій. Однією з ключових проблем є розробка алгоритмів та методів, які дозволять системам процедурної генерації творити цікаві та узгоджені ігрові предмети, які не лише відповідають структурі сюжету, але й вражають гравця своєю оригінальністю та значущістю. Важливим аспектом є забезпечення балансу між автоматизацією генерації та збереженням творчості та непередбачуваності у створенні ігрового вмісту. Таким чином, проблематика генерації ігрових предметів виявляється в потребі розробки ефективних підходів до забезпечення різноманітності, логічності та емоційної привабливості створеного контенту.

"Artificial Intelligence. Procedural Generation in Game Design" [7] підкреслює важливість використання штучного інтелекту в процесі процедурної генерації в ігровому дизайні. Однією з ключових проблем є розробка інтелектуальних алгоритмів, які здатні адаптуватися до вимог геймдизайну та забезпечувати створення ігрового контенту, що відповідає унікальним стилістичним та

геймплейним уподобанням кожного гравця. Важливим аспектом є також управління складністю та різноманіттям створеного контенту за допомогою інтелектуальних систем. Таким чином, проблематика використання штучного інтелекту в процедурній генерації ігрового контенту виявляється в необхідності розробки ефективних та адаптивних алгоритмів, які враховують усі вимоги до якісного та цікавого геймплею.

У розділі «Алгоритми та підходи» [18] висвітлюється актуальність проблеми розробки ефективних алгоритмів та підходів для процедурної генерації ігрового контенту. Однією з ключових проблем є потреба в створенні алгоритмів, які забезпечують не лише різноманітність та цікавість створеного вмісту, але й логічність та узгодженість з ігровим середовищем. Важливою темою є забезпечення балансу між автоматизацією генерації та збереженням творчості, щоб генерований контент відповідав вимогам геймдизайну та вражав гравців своєю унікальністю. Таким чином, проблематика процедурної генерації ігрового контенту виявляється в необхідності розробки алгоритмів та стратегій, що дозволяють створювати якісний та захопливий геймплей.

В книзі "Procedural Generation in Game Design" [11] наголошує на актуальності вирішення проблем, пов'язаних із використанням інструментів для створення ігрового контенту. Однією з ключових проблем є потреба в розробці ефективних та інтуїтивно зрозумілих інструментів, які допомагали б розробникам створювати та налаштовувати ігровий контент. Важливим аспектом є також забезпечення сумісності та інтеграції цих інструментів у виробничий процес розробки ігор. Таким чином, проблематика використання інструментів для створення ігрового контенту полягає в необхідності пошуку оптимальних рішень для полегшення та покращення процесу творення ігрових вмістів.

Дослідження «Опитування щодо створення процедурних» [15] ігор вказує на актуальність проблем, пов'язаних із застосуванням процедурної генерації в ігровій індустрії. Однією з ключових проблем є пошук оптимальних та ефективних алгоритмів та стратегій для процедурного генерування різноманітного і цікавого ігрового контенту. Важливим аспектом є також врахування вимог до геймдизайну

та взаємодії з гравцем. Таким чином, проблематика використання процедурної генерації в ігровій індустрії виявляється в необхідності подальших досліджень для розробки оптимальних методів та засобів для створення якісного та захоплюючого ігрового контенту.

Метод генерації ігрових предметів є важливим аспектом розробки ігор у жанрі survival simulator. Вибір методу генерації залежить від конкретних вимог гри. При розробці методу необхідно враховувати такі фактори, як тип ігрового контенту, який необхідно генерувати, вимоги до реалістичності та різноманітності контенту, а також технічні можливості платформи, на якій буде розроблена гра.

1.4. Аналіз існуючих математичних методів, моделей та алгоритмів для підвищення якості генерації ігрових предметів

Розробка методу генерації ігрових предметів для гри в жанрі survival simulator яке вимагає вивчення та застосування різноманітних математичних методів, моделей та алгоритмів. Деякі з потенційно корисних підходів включають:

Алгоритм Perlin Noise: Використовується для генерації рельєфу та ландшафту, створюючи природний та непередбачуваний вигляд гри. Перш за все, варто розглянути використання алгоритму Perlin Noise для генерації рельєфу та ландшафту. Цей алгоритм дозволяє створювати природно виглядаючі та непередбачувані пейзажі у грі. Використання Perlin Noise може забезпечити відчуття реалізму та варіативності в ігровому світі, що додасть глибину геймплею.

Алгоритмічні методи генерації ресурсів: Використовуючи жадібні алгоритми для ефективного розміщення різних видів ресурсів у грі, що впливає на геймплей та виживання гравця. Ще одним важливим аспектом є використання жадібних алгоритмів для розміщення різних видів ресурсів у грі. Це може впливати на стратегії виживання гравця, дозволяючи йому ефективно використовувати різні ресурси для поліпшення умов виживання.

Математичні моделі фізики об'єктів: Розглядання моделей фізики, які можуть бути використані для надання ігровим предметам реалістичної поведінки

та взаємодії. Враховуючи математичні моделі фізики об'єктів, можна досягти реалістичної поведінки ігрових предметів. Це може включати моделі зіткнень, руху та інші фізичні аспекти, що додають аутентичност грі.

Алгоритми генерації випадкових подій: Застосування алгоритмів Монте-Карло для випадкового генерування подій, що змінюють хід гри та викликають несподіванки. Використання алгоритмів Монте-Карло для генерації випадкових подій може зробити гру більш непередбачуваною та захоплюючою. Події, які змінюють хід гри та створюють несподіванки, збагачують геймплей та стимулюють гравців до креативних стратегій.

Методи генерації динамічного ландшафту: Розглядання алгоритмів, що дозволяють динамічно змінювати терейн, створюючи непередбачуване середовище. Розглядання алгоритмів, що дозволяють динамічно змінювати терейн, може створити непередбачуване середовище для гравців. Змінюючи ландшафт гри, можна надати гравцям нові виклики та можливості виживання.

Генетичні алгоритми для створення ворогів і ресурсів: Використання еволюційних алгоритмів для створення різноманітних ворогів та корисних ресурсів, які взаємодіють з гравцем та оточенням. Використання генетичних алгоритмів для створення ворогів та корисних ресурсів може призвести до створення різноманітних та інтелектуальних об'єктів у грі. Еволюційний підхід до генерації додає в гру елемент варіативності та непередбачуваності.

Ці методи, моделі та алгоритми можуть бути важливими компонентами для реалізації ефективного та захопливого методу генерації ігрових предметів у жанрі survival simulator, забезпечуючи унікальний та захоплюючий геймплей.

1.5. Аналіз засобів програмної інженерії для підвищення якості генерації ігрових предметів

BYOND (Build Your Own Net Dream) - це розвита платформа для створення та розгортання мультиплеєрних ігор.



Рис. 1.2. Логотип Beyond

BYOND надає інструменти розробки, які дозволяють геймдевелоперам створювати гри зі складною логікою, мережевою взаємодією та вбудованим чатом для спілкування гравців.

Основні причини

Простота розробки: BYOND надає простий та інтуїтивний інтерфейс для створення ігор. Мова програмування DM, яка використовується на BYOND, є легкою для вивчення, що робить платформу доступною для розробників з різним досвідом.

Мережева взаємодія: BYOND має вбудовану підтримку для мережевої гри, що дозволяє створювати мультиплеєрні ігри. Це важливо для ігор, таких як Space Station 13, де взаємодія гравців має ключове значення.

Активна спільнота: BYOND має активну спільноту розробників та гравців. Це сприяє обміну ідеями, розробці нових ігор та постійному вдосконаленню платформи.

Гнучкість та розширюваність: BYOND надає інструменти для створення різноманітних ігор з різними механіками та стилістикою. Це робить платформу гнучкою та привабливою для розробників.

Розробка методу генерації ігрових предметів для гри в жанрі survival simulator вимагає використання різних технологій та методів програмної інженерії для досягнення високої якості та реалістичності ігрового віртуального світу. Далі розглядається деяка група інструментів та методів, які зазвичай застосовуються у процесі розробки генератора ігрових предметів:

Space Station 13 (SS13) використовує DM-мову (Dream Maker) для написання коду гри та взаємодії з ігровим світом. Однак, DM-мова часто визначається як "не надто складна" для реалізації, особливо для новачків у програмуванні та розробці ігор.



Рис.1.3. Логотип DM-language

Розглянемо кілька причин, чому DM-мова може вважатися легшою для використання:

Простота синтаксису

DM-мова має простий та зрозумілий синтаксис, що полегшує розробку та розуміння коду. Основні концепції, такі як об'єкти, процедури та змінні, виражені чітко та доступно.

Легка інтеграція

У зв'язку з тим, що DM-мова спеціально створена для Space Station 13, вона вже має вбудовані функції та бібліотеки, які спрощують роботу з ігровими об'єктами, подіями та взаємодією з гравцями.

Низький поріг вступу

DM-мова часто вибирається в проектах, де спільнота гравців бажає активно брати участь у розробці. Такий підхід дозволяє більшій кількості людей приєднатися до проекту та внести свій внесок, навіть якщо вони не є досвідченими програмістами.

Вбудована взаємодія з грою

DM-мова включає в себе інструменти для простої взаємодії з ігровим світом, які вже інтегровані у розробку гри. Це робить процес створення нових функцій та ігрових елементів більш доступним та природнім.

Проект з відкритим вихідним кодом

Існує тенденція, що проекти з відкритим вихідним кодом, такі як Space Station 13, надають сприяння для спільної розробки та внесення внеску в розвиток гри. Це дозволяє більш широкому колу розробників вносити свої ідеї та вдосконалення без значних перешкод.

Узагальнюючи, DM-мова вигідна для Space Station 13 завдяки своєму простому синтаксису, вбудованій взаємодії з грою та відкритому характеру

проекту, що дозволяє широкому колу розробників легко долучатися та вносити свій внесок.

Мова програмування

DM-мова, або Dream Maker, є мовою програмування, яка була спеціально розроблена для створення ігор на платформі Space Station 13 (SS13). Вона є внутрішньою мовою для роботи з різноманітними елементами гри, включаючи об'єкти, події, інтерфейси та інше. DM-мова має специфічні функції та структури, що відрізняють її від загальноприйнятих мов програмування.

Основні характеристики DM-мови включають:

Обробка подій: Головний механізм взаємодії з ігровим світом полягає у визначенні та обробці подій. Це може бути подія взаємодії гравця з предметом, подія відсічення деякої умови, чи подія, пов'язана зі змінами у грі.

Робота з об'єктами: DM-мова дозволяє створювати та взаємодіяти з різними об'єктами в ігровому світі, такими як предмети, персонажі, та інші елементи гри.

Керування інтерфейсом: Розробники можуть створювати і призначати інтерфейси для взаємодії з гравцями, такі як вікна чату, кнопки та інші елементи.

Щодо реалізації нового методу генерації ігрових предметів на мові DM, основні етапи можуть виглядати приблизно так:

Вивчення існуючого коду: Ознайомлення з існуючими механіками гри та існуючим кодом, який відповідає за генерацію ігрових предметів.

Визначення нового методу генерації: Визначення концепції та логіки нового методу генерації, включаючи параметри, з якими будуть створюватися предмети.

Написання коду: Реалізація нового методу на мові DM з використанням вбудованих функцій та структур, які надає сама мова.

Тестування та налагодження: Проведення тестів нового методу генерації, виявлення та виправлення помилок, які можуть виникнути в процесі розробки.

Впровадження в гру: Інтеграція нового методу у вихідний код гри та тестування в реальних умовах гри.

2 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ РОЗМІЩЕННЯ ІГРОВИХ ПРЕДМЕТІВ ТА ЗАСОБІВ ПІДВИЩЕННЯ ЯКОСТІ ГЕНЕРАЦІЇ ІГРОВИХ ПРЕДМЕТІВ

2.1. Основні методи генерації ігрових предметів жанру survival simulators

У жанрі survival simulators ігрові предмети відіграють важливу роль. Вони використовуються для задоволення базових потреб гравця, таких як харчування, одяг, захист, а також для вирішення завдань та досягнення цілей гри.

Існує два основних методи генерації ігрових предметів у жанрі survival simulators:

Статична генерація. У цьому випадку ігрові предмети розташовуються в певних місцях на карті гри. Наприклад, їжа може розташовуватися в лісі, а будівельні матеріали - у руїнах.

Динамічна генерація. У цьому випадку ігрові предмети розташовуються в різних місцях на карті гри в ході гри. Наприклад, їжа може з'являтися на землі, а будівельні матеріали - в ящиках.

Статична генерація має ряд переваг. Вона дозволяє розробникам гри контролювати те, де знаходяться ігрові предмети, що забезпечує більшу передбачуваність гри. Крім того, статична генерація може бути простішою в реалізації, ніж динамічна.

Однак статична генерація має і ряд недоліків. Вона може призводити до того, що деякі ігрові предмети будуть надто важко або легко знайти. Крім того, статична генерація може обмежувати творчість гравців.

Динамічна генерація має ряд переваг. Вона дозволяє створювати більш реалістичне середовище, в якому ігрові предмети можуть знаходитися в різних місцях. Крім того, динамічна генерація може підвищити повторюваність гри.

Однак динамічна генерація має і ряд недоліків. Вона може бути складнішою в реалізації, ніж статична. Крім того, динамічна генерація може призводити до того, що деякі ігрові предмети будуть надто легко або важко знайти.

Засоби підвищення якості генерації ігрових предметів

Існує ряд засобів, які можуть використовуватися для підвищення якості генерації ігрових предметів у жанрі survival simulators. До них відносяться:

- Використовування штучного інтелекту. Штучний інтелект може використовуватися для створення більш реалістичних моделей поведінки ігрових предметів. Наприклад, їжа може псуватися з часом, а будівельні матеріали можуть бути використані для створення різних будівель.
- Використовування даних реального світу. Дані реального світу, такі як карти, кліматичні дані та інформація про рослинний і тваринний світ, можуть використовуватися для створення більш реалістичного середовища для генерації ігрових предметів.
- Використання геймдизайну. Геймдизайнери можуть використовувати свої знання та досвід для створення більш цікавої та захопливої генерації ігрових предметів. Наприклад, вони можуть використовувати генерацію ігрових предметів для створення додаткових завдань і цілей для гравців.

2.2. Математичні методи використання статичної генерації для ігрових предметів

Статична генерація ігрових предметів передбачає розташування їх у певних місцях на карті гри. Для реалізації цього методу можна використовувати наступні математичні методи:

Рандомне розташування. У цьому випадку ігрові предмети розташовуються в випадкових місцях на карті гри. Цей метод є простим у реалізації, але він може призводити до того, що деякі ігрові предмети будуть надто важко або легко знайти. Один зі способів статичної генерації ігрових предметів - це використання рандомного розташування. У цьому випадку, ігрові предмети розташовуються випадковим чином на карті гри. Цей метод простий у втіленні, проте може призводити до нерівномірного розташування предметів, деякі з них можуть бути занадто рідкі або, навпаки, занадто часті. Варто враховувати баланс між доступністю та викликом для гравців.

Геометричні моделі. У цьому випадку ігрові предмети розташовуються в певних точках на карті гри, які відповідають певним геометричним моделям. Наприклад, їжа може розташовуватися в лісі, а будівельні матеріали - у руїнах. Цей метод дозволяє забезпечити більшу передбачуваність розташування ігрових предметів, але він може бути складнішим у реалізації. Ще одним методом статичної генерації є використання геометричних моделей. Ігрові предмети розташовуються в конкретних точках на карті, які відповідають певним геометричним моделям. Наприклад, ресурси можуть бути пов'язані з природними об'єктами, а елементи будівництва можуть розташовуватися в специфічних структурах. Цей підхід забезпечує більшу передбачуваність, але вимагає детального визначення геометричних параметрів та інтеграції з дизайном гри.

Машинне навчання. У цьому випадку для розташування ігрових предметів використовуються алгоритми машинного навчання. Ці алгоритми можуть навчитися розташовувати ігрові предмети таким чином, щоб вони були більш доступними для гравців, але не надто легкими для знаходження. Використання алгоритмів машинного навчання є сучасним та перспективним підходом до статичної генерації ігрових предметів. Алгоритми можуть навчитися розташовувати предмети так, щоб забезпечити баланс між доступністю та викликом. Це дозволяє адаптувати розташування предметів відповідно до рівня гравця та інших факторів, забезпечуючи оптимальний геймплей.

2.3. Методи машинного навчання для генерації ігрових предметів

Машинне навчання - це область інформатики, яка займається розробкою алгоритмів, які можуть навчатися на даних і поліпшувати свої результати з часом.

Методи машинного навчання можна використовувати для генерації ігрових предметів різними способами. Один із способів - використовувати методи машинного навчання для навчання на наборі даних існуючих ігрових предметів. Це дозволить алгоритму навчитися генерувати ігрові предмети, які схожі на існуючі предмети.

Інший спосіб використання методів машинного навчання для генерації ігрових предметів - використовувати методи машинного навчання для створення нових і унікальних ігрових предметів. Це можна зробити, наприклад, за допомогою методів генеративного навчання, таких як генеративні нейронні мережі (GAN).

Генеративно-конкурентні мережі (GAN)

GAN - це тип генеративного навчання, який використовує два типи нейронних мереж: генератор і дискримінатор. Генератор намагається генерувати дані, які схожі на реальні дані, а дискримінатор намагається відрізнити генеровані дані від реальних даних.

GAN можна використовувати для генерації ігрових предметів, подібних до існуючих ігрових предметів. Для цього можна використовувати генератор для генерації даних, які представляють ігрові предмети, а дискримінатор для навчання відрізняти генеровані дані від реальних даних.

Також GAN можна використовувати для створення нових і унікальних ігрових предметів. Для цього можна використовувати генератор для генерації даних, які представляють нові ігрові предмети, а дискримінатор для навчання відрізняти генеровані дані від реальних даних. Формула для GAN наступна(2.1)

$$\min_G \max_D E_x [\log D(x)] + E_z [\log (1 - D(G(z)))] \quad (2.1)$$

Де G – генератор, дискримінатор, реальні дані, випадкові дані.

Методи машинного навчання мають ряд переваг для генерації ігрових предметів. По-перше, вони можуть генерувати ігрові предмети, які більш різноманітні і унікальні, ніж ігрові предмети, генеровані традиційними методами. По-друге, вони можуть генерувати ігрові предмети, які більш адаптовані до конкретних потреб гри.

Однак методи машинного навчання також мають ряд недоліків. По-перше, вони можуть бути складними для реалізації і навчання. По-друге, вони можуть бути повільним для генерації ігрових предметів.

Застосування методів машинного навчання для генерації ігрових предметів

Методи машинного навчання вже використовуються для генерації ігрових предметів в деяких іграх. Наприклад, метод GAN використовується в грі Minecraft для генерації випадкових генеративних структур.

У майбутньому методи машинного навчання будуть все частіше використовуватися для генерації ігрових предметів. Це дозволить створювати ігри з більш різноманітними і унікальними ігровими предметами, які будуть краще адаптовані до конкретних потреб гравців.

Застосування методів машинного навчання для генерації ігрових предметів для гри в жанрі survival simulator

Для гри в жанрі survival simulator важливо, щоб ігрові предмети були різноманітними і унікальними. Це дозволить гравцям знаходити нові і цікаві предмети, які допоможуть їм вижити в грі.

Методи машинного навчання можна використовувати для генерації ігрових предметів для гри в жанрі survival simulator різними способами. Один із способів - використовувати методи машинного навчання для навчання на наборі даних існуючих ігрових предметів. Це дозволить алгоритму навчитися генерувати ігрові предмети, які схожі на існуючі предмети, але також будуть включати в себе деякі нові і унікальні елементи.

Штучні нейронні мережі (ШНМ) є ключовою технологією в галузі машинного навчання. Ці мережі моделюють структуру та функціональність людського мозку, застосовуючи концепції нейронів та їх взаємодії. Вони

використовуються для вирішення різноманітних завдань, таких як класифікація зображень, розпізнавання мови, прогнозування та багато інших.

Метод ШНМ включає в себе процес навчання мережі на основі великої кількості вхідних даних, де модель адаптується до зразків і вчиться вирішувати певні задачі. Мережа складається зі шарів нейронів, де кожен нейрон з'єднаний з іншими за допомогою ваг. Під час навчання ці ваги оптимізуються для досягнення бажаного результату.

Штучні нейронні мережі широко використовуються в різних галузях, включаючи обробку природної мови, комп'ютерне бачення, автоматичне водіння, фінанси, медицину та інші. Вони є основою для розробки глибоких нейронних мереж (Deep Neural Networks - DNN), які використовують багато шарів для рішення складних завдань.

Машинне навчання, включаючи методи ШНМ, є ключовою галуззю в сучасній інформатиці, забезпечуючи розвиток технологій і рішень, які використовуються у реальних сценаріях та призводять до значних вдосконалень в різних областях життя.

Машинне навчання та штучні нейронні мережі включають в себе велику кількість різних алгоритмів та методів. Проте, якщо говорити про базовий принцип функціонування нейронів у штучних нейронних мережах, то його можна представити за допомогою формули для вагової суми входів, яка передається через активаційну функцію(2.2):

$$y = f(\sum_{i=1}^n w_i \cdot x_i + b) \quad (2.2)$$

Де - y - вихід нейрона, f - активаційна функція, $\sum_{i=1}^n w_i$ - вагована сума входів з вагами w_i та входами x_i , b - зсув (bias), що додається до вагової суми.

Ця формула відображає основний принцип роботи одного нейрона в нейронній мережі. Процес навчання полягає в оптимізації ваг (w_i та b) таким чином, щоб нейрон правильно вирішував задачу для заданого набору вхідних даних.

Важливо відзначити, що активаційна функція f може бути різною, наприклад, сигмоїдальною, гіперболічним тангенсом або функцією ReLU, в

залежності від завдань та особливостей даних. Метод Дерев Рішень є алгоритмом машинного навчання, що використовується для прийняття рішень на основі вхідних даних. Цей метод є одним з найпростіших і ефективних у своєму роді, і він може використовуватися для класифікації та регресії.

Основна ідея полягає у побудові дерева, де кожен внутрішній вузол представляє тест для атрибуту даних, кожен гілочка виходить з вузла і вказує на результат тесту, а кожен листовий вузол містить вихідне рішення чи мітку класу.

Метод Дерев Рішень працює наступним чином: він обирає атрибут даних, за яким проводить тест, так щоб максимізувати інформаційний приріст. Після кожного тесту, дані розділяються на дві групи, і процес повторюється для кожної нової групи. Цей процес триває досягнення критерію зупинки, такого як максимальна глибина дерева чи мінімальна кількість зразків у листовому вузлі.

Дерева Рішень широко використовуються у різних галузях, таких як класифікація об'єктів, розпізнавання образів, аналіз даних і прийняття рішень. Вони можуть бути легко зрозумілі та візуалізовані, що робить їх привабливим інструментом для аналізу та роботи з даними.

Метод Опорно-Векторних Машин (SVM) є алгоритмом машинного навчання, призначеним для завдань класифікації та регресії. Його основна мета - знайти оптимальний гіперплощину в просторі великої розмірності, яка максимально відокремлює класи даних.

Суть методу полягає у пошуку гіперплощини, яка максимізує відстань між точками кожного класу із даних (називається "зазором"). При цьому опорні вектори, які лежать на межі зазору, визначають цю гіперплощину. SVM може також використовуватися для розв'язання задач регресії, де завдання полягає у визначенні оптимального пристосування гіперплощини до точок даних.

Метод SVM використовує ядрові функції для перетворення нелінійно відокремлюваних даних в простір вищої розмірності, де вони можуть бути лінійно розділені. Це дозволяє вирішувати складні задачі класифікації.

SVM знаходить застосування в багатьох областях, таких як розпізнавання образів, біоінформатика, фінансова аналітика та багато інших. Його ефективність

та здатність працювати з різними типами даних роблять його популярним інструментом у сфері машинного навчання.

Інший підхід до створення ігрових предметів для гри в жанрі "survival simulator" полягає в застосуванні методів машинного навчання в далеких марення

2.4. Аналіз існуючих методів генератора псевдовипадкових чисел

Стандартний підхід генерації шуму Перліна, який полягає в тому, щоб відобразити вивід функції шуму Перліна на конкретні плитки. Функція шуму Перліна формується шляхом наложення кількох карт шуму з різною амплітудою і частотою, що називаються "октавами".

Шум Перліна - це складний алгоритм для генерації випадкового рельєфу, який залишається гладким та безперервним. Найкраще в тому, що цей алгоритм працює добре з процедурною генерацією, що означає, що він може створювати частину світу, а потім, рухаючись до краю, можна генерувати більше рельєфу, який буде безперервно з'єднуватися.

Формула для обчислення інтерполяції шуму Перліна виглядає наступним чином(3.1):

$$P = (x, y) = \text{floor}(x), j = \text{floor}(y) \quad (3.1)$$

$$g_{00} = \text{gradient at}(i,), g_{10} = \text{gradient at}(i + 1,) \quad (3.2)$$

$$g_{01} = \text{gradient at}(i, +1), g_{11} = \text{gradient at}(i + 1, +1) \quad (3.3)$$

$$u = x - i, v = y - j \quad (3.4)$$

$$n_{00} = g_{00} \cdot [u \ v], n_{10} = g_{10} \cdot [u - 1 \ v], n_{01} = g_{01} \cdot [u \ v - 1], n_{11} = g_{11} \cdot [u - 1 \ v - 1] \quad (3.5)$$

$$n_{x0} = n_{00}(1 - f(u)) + n_{10}f(u), n_{x1} = n_{01}(1 - f(u)) + n_{11}f(u) \quad (3.6)$$

$$n_{xy} = n_{x0}(1 - f(v)) + n_{x1}f(v) \quad (3.7)$$

Де перша формула визначає внутрішню частину шуму Перліна для точки (x, y).

Формули (3.2), (3.3) і (3.4) обчислюють градієнти (вектори напрямку) для кожного з чотирьох кутів, що оточують цю точку.

Формули (3.5) обчислюють внутрішні добутки градієнтів та відносні різниці між поточними координатами (x, y) та нижчими цілими значеннями цих координат (зазвичай це (x,y) , $(x+1,y)$, $(x,y+1)$, $(x+1,y+1)$).

Формули (3.6) використовують обчислені внутрішні добутки та різниці для інтерполяції значення шуму Перліна між чотирма кутами.

Формула (3.7) об'єднує значення інтерполяції між кутами для отримання кінцевого значення шуму Перліна для точки (x, y) .

Ці формули використовуються для обчислення значень шуму Перліна між різними точками, використовуючи лінійну інтерполяцію між градієнтами у просторі.

Генерації псевдовипадкових чисел, що базується на лінійних рекурентних відносинах(3.8). Цей метод широко використовується для генерації випадкових чисел в комп'ютерних програмах і мовах програмування

$$P_{n+1}=(a \times P_n+c) \bmod m \quad (3.8)$$

Де P_{n+1} - наступне псевдовипадкове число, P_n - поточне псевдовипадкове число, a - множник (зазвичай велике просте число), c - приріст (зазвичай невелике число).

Лінійна змішана форма генератора псевдовипадкових чисел

Лінійна змішана форма генератора псевдовипадкових чисел — це метод генерації випадкових чисел, який використовує лінійну комбінацію декількох підгенераторів для створення більш складного псевдовипадкового виходу.

Прикладом такого генератора може бути вираз (3.9)

$$X_{n+1} = (a_1 X_{n1} + a_2 X_{n2} + \dots + a_k X_{nk}) \bmod m \quad (3.9)$$

Де X_{n+1} - наступне псевдовипадкове число, $X_{n1}, X_{n2}, \dots, X_{nk}$ - виходи різних підгенераторів, a_1, a_2, \dots, a_k - коефіцієнти лінійної комбінації, m - модуль, що використовується для обмеження значень.

Цей підхід дозволяє створити більш випадковий вихід, комбінуючи внески різних підгенераторів. Такий генератор може бути налаштований на вивчення конкретних потреб і вимог за допомогою правильного вибору підгенераторів та коефіцієнтів.

Важливою особливістю є його здатність генерувати псевдовипадкові числа, які мають хороші статистичні властивості та відповідають різноманітним критеріям випадковості. Даний метод є ефективним і широко використовується в областях, де важлива випадковість, таких як криптографія, моделювання, та інші області, де потрібна випадковість для відтворення непередбачуваних подій.

Змішаний конгруентний метод генерації псевдовипадкових чисел(3.10) є одним із методів, який використовує конгруентні генератори для отримання послідовностей псевдовипадкових чисел. Цей метод спроектований для забезпечення кращих характеристик випадковості порівняно з простим конгруентним методом.

Прикладом такого генератора може бути вираз(3.10)

$$X_{n+1} = (aX_n + c) \bmod m \quad (3.10)$$

Де X_{n+1} - наступне псевдовипадкове число, X_n - поточне псевдовипадкове число, a - множник (зазвичай просте число), m - модуль, що використовується для обмеження значень модуль (натуральне число, відносно якого обчислюють остачу від ділення $m \geq 2$)

Цей метод використовує лінійну рекуррентну формулу для генерації чисел, що визначаються рекуррентним співвідношенням. Ключовою властивістю ЗКМПВЧ є вибір оптимальних параметрів (a, c, m) , які забезпечують хороші статистичні характеристики та великий період повторення послідовності чисел.

Змішаний конгруентний метод дозволяє отримати послідовності псевдовипадкових чисел з високою випадковістю, що робить його застосовним у багатьох областях, таких як моделювання, статистика, інформаційна безпека та інші. Однак важливо відзначити, що вибір параметрів має вирішальне значення для якості згенерованої послідовності, і неправильний вибір може призвести до неприйнятних результатів.

Квадратичний конгруентний метод генерації псевдовипадкових чисел(3.11) є іншим методом, який базується на конгруентних генераторах, але використовує квадратичне рекурентне співвідношення для генерації чисел. Цей метод може виявити велике значення для побудови випадкових послідовностей, якщо правильно налаштований.

Основна форма може бути виражена так(3.11):

$$X_{n+1} = (aX_n^2 + bX_n + c) \bmod m \quad (3.11)$$

Де X_{n+1} - наступне псевдовипадкове число, X_n - поточне псевдовипадкове число, a , b , c - коефіцієнти, m - модуль, що використовується для обмеження значень. модуль (натуральне число, відносно якого обчислюють остачу від ділення $m \geq 2$).

Квадратичний конгруентний метод використовує квадратичну рекурентну формулу для генерації чисел, що визначаються рекурентним співвідношенням. Так як цей метод є більш складним порівняно з лінійним конгруентним методом, вибір коефіцієнтів та модуля стає ще більш важливим для забезпечення якості та випадковості послідовності.

Квадратичний конгруентний метод також може знайти своє застосування у галузях, де важливі випадкові величини, таких як статистика, моделювання, криптографія, інформаційна безпека тощо. Однак важливо ретельно вибирати параметри для забезпечення стійкості та статистичної випадковості виходу генератора.

Лінійний конгруентний метод генерації псевдовипадкових чисел (3.12) є простим і широко використовуваним методом, що базується на лінійних рекурентних співвідношеннях. Цей метод зручний для використання та вимагає мінімальних обчислень, але ефективність його залежить від правильного вибору параметрів.

Основна форма виглядає наступним чином(3.12):

$$X_{n+1} = (aX_n + c) \bmod m \quad (3.12)$$

Де X_{n+1} - наступне псевдовипадкове число, X_n - поточне псевдовипадкове число, a - множник (зазвичай просте число), c - приріст (доданок), m - модуль, що

використовується для обмеження значень. модуль (натуральне число, відносно якого обчислюють остачу від ділення $m \geq 2$).

Цей метод генерує послідовність чисел, яка повторюється з періодом, що дорівнює m , і виглядає випадково. Однак важливо відзначити, що неправильний вибір параметрів може призвести до непередбачуваного поведінки та низької якості згенерованої послідовності.

Лінійний конгруентний метод широко використовується в різних областях, таких як моделювання, статистика, криптографія (за умови вибору відповідних параметрів для криптостійкості) та інші. Однак він має певні обмеження, і в сучасних застосуваннях часто використовують більш складні методи генерації псевдовипадкових чисел.

Генератор випадкових чисел з назвою "Mother-of-All" (МОА) (3.13) - це метод генерації псевдовипадкових чисел, який створений для забезпечення великого періоду повторення та хороших статистичних властивостей. Винайдений в 1993 році, МОА використовує комбінацію арифметичних та логічних операцій для створення псевдовипадкової послідовності.

Основна форма МОА(3.13) може бути виражена так:

$$X_{n+1} = (A \cdot X_n + B) \bmod 2^{32} \quad (3.13)$$

Де X_{n+1} - наступне псевдовипадкове число, X_n - поточне псевдовипадкове число, A та B - константи, які були вибрані для забезпечення хороших характеристик випадковості.

МОА використовує 32-бітні числа та використовує арифметичні операції, такі як множення та додавання, для генерації нових значень у послідовності. Однак важливо відзначити, що вибір констант грає важливу роль у якості згенерованої послідовності.

МОА є ефективним генератором псевдовипадкових чисел і використовується в деяких областях, де потрібна випадковість, таких як статистика та інженерія. Однак, з огляду на постійний розвиток областей генерації псевдовипадкових чисел, завжди важливо перевіряти, чи є обрані методи адекватними для конкретних потреб і вимог.

Вихід Мерсенна (Mersenne Twister) (3.14) - це один з найпоширеніших та використовуваних методів генерації псевдовипадкових чисел. Він був розроблений для забезпечення довгого періоду повторення та великої випадковості в згенерованій послідовності. Вихід Мерсенна є одним з найефективніших методів генерації випадкових чисел.

Основна форма Вихору Мерсенна(3.14) може бути виражена так:

$$X_{n+1} = f(X_n) \quad (3.14)$$

Де X_{n+1} - наступне псевдовипадкове число, X_n - поточне псевдовипадкове число, f - нелінійна функція, яка має специфічну структуру для генерації великої послідовності чисел.

Однією з особливостей Вихору Мерсенна є його довгий період повторення, що дорівнює $2^{19937} - 1$. Це означає, що після генерації $2^{19937} - 1$ чисел, генератор повторюється. Вихід Мерсенна також добре розподіляється в просторі і має високий ступінь випадковості.

Цей метод широко використовується в областях, де потрібна велика кількість випадкових чисел з високою випадковістю, таких як наукові дослідження, моделювання, криптографія та інші. Однак важливо враховувати, що для деяких додатків можуть бути важливі інші аспекти, такі як криптографічна стійкість.

Генератори типу "Xorshift"(3.15) представляють собою клас методів генерації псевдовипадкових чисел, які використовують операції бітового виключення для змішування і розширення послідовності чисел. Ці генератори відзначаються великою швидкістю та простотою реалізації, а також доброю якістю випадковості в згенерованих послідовностях.

Основна форма Xorshift-генератора(3.15) може виглядати так:

$$X_{n+1} = X_n \oplus (X_n \lll a) \oplus (X_n \ggg b) \quad (3.15)$$

Де X_{n+1} - наступне псевдовипадкове число, X_n - поточне псевдовипадкове число, a та b - константи, що визначають зсув бітів.

У цій формулі використовуються операції бітового виключення (\oplus), зсув вліво ($\lll\lll$) та зсув вправо ($\ggg\ggg$). Вибір констант a і b грає важливу роль у якості та випадковості згенерованої послідовності.

Xorshift-генератори є ефективними та добре пристосованими для багатьох застосувань, таких як ігрова індустрія, комп'ютерна графіка, моделювання та інші. Однак їхні статистичні властивості можуть бути менш надійними порівняно з деякими іншими методами, такими як Mersenne Twister. Також слід враховувати, що не всі Xorshift-генератори підходять для криптографічних застосувань, і в цьому випадку можуть вимагати додаткових заходів забезпечення безпеки.

3 РОЗРОБКА МЕТОДУ ГЕНЕРАЦІЇ ІГРОВИХ ПРЕДМЕТІВ ДЛЯ ГРИ В ЖАНРІ SURVIVAL SIMULATOR

3.1 Вибір програмних засобів реалізації

Вибір програмних засобів реалізації мови DM (Dream Maker) у Space Station 13:

Dream Maker (DM) - це мова програмування, розроблена спеціально для створення контенту у віртуальних світах, зокрема, використовується в грі Space Station 13 (SS13). Ось детальний розгляд причин вибору DM у контексті SS13:

Специфічність гри:

DM була розроблена для задоволення унікальних вимог, які виникають у світі Space Station 13. Вона оптимізована для роботи зі складними системами, які включають в себе взаємодію з гравцями, керування станцією, імітацію подій і різноманітні ігрові елементи.

Простота в освоєнні:

DM надає простий та зрозумілий синтаксис, що полегшує розробку контенту для гри. Це важливо для розробників, які можуть бути не програмістами, але водночас потребують зручних інструментів для створення великого обсягу ігрового контенту.

Інтеграція з движком гри:

DM інтегрується безперешкодно з движком гри Space Station 13. Вона надає доступ до специфічних функцій та об'єктів гри, що полегшує взаємодію з ігровим світом та забезпечує більший контроль над його елементами.

Підтримка громади розробників:

У зв'язку із специфічністю гри та великою кількістю гравців, розробницька спільнота SS13 є великою та активною. DM став популярним вибором серед розробників завдяки широкому колу ресурсів, підтримці та обміну досвідом.

Можливості маніпулювання ігровим світом:

DM надає великий функціонал для маніпулювання світом гри, включаючи створення нових об'єктів, імітацію подій, керування персоналом і багато іншого. Це робить його ідеальним для потреб, що виникають при розробці ігрового контенту в SS13.

Обираючи DM для реалізації у SS13, розробники отримують не лише потужний інструмент для створення віртуальних світів, але і гнучкість та спеціалізацію, які відповідають унікальним вимогам гри Space Station 13.

3.2. Аналіз нового методу генерації ігрових предметів

Необхідність нового методу генерації ігрових предметів визначається важливістю впровадження різноманітності та унікальності в геймплей жанру survival simulator. Традиційні методи генерації можуть недостатньо задовольняти потреби гравців у великому асортименті предметів з різноманітними характеристиками, що є ключовим елементом успішного геймплею в жанрі survival.

Новий метод вносить інновації, забезпечуючи ширший спектр характеристик та властивостей, які можна надати ігровим предметам. Використання поєднання методів генерації за допомогою шуму Перліна та лінійного конгруентного методу дозволяє створювати не тільки графічно ефектні предмети, але й контролювати числові параметри, такі як вартість, рідкість чи інші важливі атрибути.

Цей підхід відповідає прагненням розробників до створення глибокого та іммерсивного геймплею, де кожен предмет не лише виглядає унікально, але і має значущий внесок у гравецький досвід. Новий метод генерації стає ключовим компонентом для забезпечення ігрового світу жанру survival simulator великою кількістю цікавих та важливих рішень, що допомагають створити непередбачуваність та глибину ігрового досвіду.

Поєднання методів генерації за допомогою шуму Перліна та лінійного конгруентного методу вносить ряд переваг, роблячи його привабливим в контексті

створення ігрових предметів.

Реалістичність і Контроль

Поєднання шуму Перліна і лінійного методу дозволяє досягти балансу між реалістичністю та контролем над генерацією. Шум Перліна природно і органічно розподіляє характеристики предметів, надаючи їм природоподібний вигляд. У той же час, лінійний метод дозволяє точно керувати числовими параметрами, такими як вартість, рідкість чи інші числові аспекти, що є важливим для геймплею.

Гнучкість налаштувань

Поєднання обох методів забезпечує гнучкість налаштувань. Розробники можуть ефективно контролювати вагові коефіцієнти обох методів, визначаючи, як їх характеристики взаємодіють між собою. Це дає можливість створювати різноманітні та унікальні ігрові предмети, залежно від концепції гри та бажаного геймплейного ефекту.

Баланс та Різноманітність

Поєднання обох методів допомагає уникнути екстремальних сценаріїв та забезпечує баланс у характеристиках ігрових предметів. Такий підхід дозволяє створювати різноманітні предмети, які не лише виглядають цікаво, а й мають адекватні числові параметри, сприяючи вирівнюванню геймплею.

Збереження Іммерсії

Поєднання природних характеристик шуму Перліна з точним контролем числових значень лінійного методу допомагає зберегти іммерсію гравців. Предмети виглядають автентично та водночас взаємодіють з геймплейними механіками, що підсилює враження від гри.

В цілому, поєднання методів генерації ігрових предметів забезпечує комплексний і гармонійний підхід до створення геймплейно важливих та візуально привабливих елементів.

Реалізація нового метода:

- Реалізацію можна виконати в програмній мові, яку ви використовуєте для розробки гри
- Створіть функції для генерації шуму Перліна та лінійного

конгруентного методу.

- Визначте параметри генерації та вагові коефіцієнти.
- Реалізуйте алгоритм для створення ігрових предметів, використовуючи обидва методи та їх комбінацію.
- Такий метод генерації дозволяє створювати ігрові предмети з різноманітними та цікавими характеристиками, покращуючи враження від гри та забезпечуючи гравцям унікальний досвід.

3.2.1. Вибір програмних засобів реалізації

Для реалізації методу генерації ігрових предметів для гри в жанрі survival simulator використовується мова програмування DM (Dream Maker).

DM (Dream Maker):

Вибір мови DM може бути обумовлений тим, що вона є мовою програмування, яка широко використовується для розробки ігор в масовому онлайн-жанрі Space Station 13. Вона має інтеграцію з графічним двигуном, що використовується в цій грі, та специфічними функціями для роботи з ігровими об'єктами та подіями.

Бібліотеки для генерації шуму Перліна в DM якщо DM не надає вбудованих засобів для генерації шуму Перліна, то можна реалізувати або використовувати наявні рішення для генерації шуму, підганяючи їх під особливості мови DM.

Інтеграція з графічним двигуном DM може використовувати власний графічний двигун, який інтегрується з системою генерації. Це може полегшити взаємодію між генерацією предметів і графікою гри.

Інструменти для математичних обчислень в DM повинен надавати базові математичні функції для використання у генерації предметів, і деякі інструменти можуть бути створені власноруч або використовувати наявні можливості мови.

Система контролю версій для збереження та відстеження змін у коді також може бути використана система контролю версій, але з огляду на специфіку DM.

Документація та коментарі в DM Важливо додавати докладну документацію та коментарі до коду для зручності розуміння та підтримки.

Такий вибір програмних засобів дозволяє використовувати DM мову, яка спеціалізується на розробці ігор у жанрі Space Station 13, та забезпечує інтеграцію з потрібними функціоналами для розробки методу генерації ігрових предметів.

3.2.2. Вибір генератора псевдовипадкових чисел моделі

Генерація за допомогою шуму Перліна

Шум Перліна - це вид стохастичного шуму, який часто використовується в графіці для створення натуральних ландшафтів і текстур.

Для генерації ігрових предметів шум Перліна можна використовувати для створення різноманітних атрибутів предметів, таких як кількість, тип, рівень, координати тощо.

Використання генератора псевдовипадкових чисел

Генератор псевдовипадкових чисел - це алгоритм, який генерує послідовність чисел, які виглядають випадковими.

Для генерації ігрових предметів генератор псевдовипадкових чисел можна використовувати для випадкового вибору атрибутів предметів.

Для генерації ігрових предметів для гри в жанрі survival simulator важливо використовувати псевдовипадкові числа, які мають високу ентропію і довгий період. Лінійний конгруентний метод відповідає цим вимогам, тому він є хорошим вибором для цього завдання.

Різниця між методами

Основна відмінність між двома методами полягає в тому, як вони генерують випадковість.

Генератор шуму Перліна генерує випадковість на основі математичної функції. Це дозволяє створювати більш реалістичні результати, але вимагає додаткових обчислень.

Генератор псевдовипадкових чисел генерує випадковість на основі псевдовипадкових чисел. Це простіше і швидше, але може призвести до менш реалістичних результатів.

Для гри в жанрі survival simulator важливим є створення різноманітних ігрових предметів, які будуть виглядати натурально.

Для цього можна використовувати комбінацію двох методів:

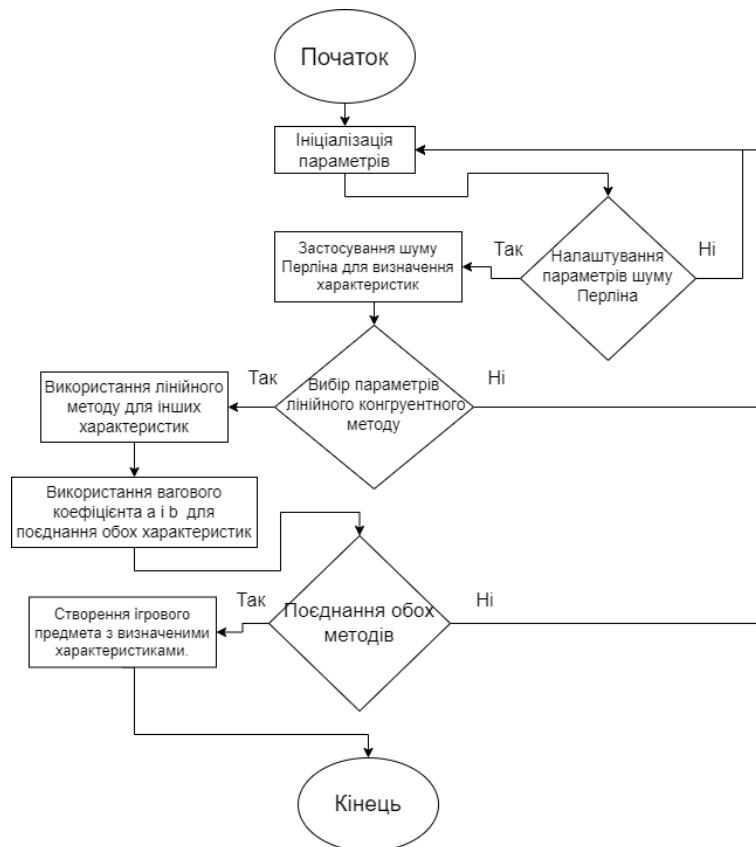
Генератор шуму Перліна можна використовувати для створення загального вигляду ландшафту та розташування предметів.

Генератор псевдовипадкових чисел можна використовувати для випадкового вибору атрибутів предметів.

Така комбінація дозволить забезпечити реалістичний вигляд ландшафту і різноманітність ігрових предметів..

3.2.3. Вибір алгоритму реалізації метода генерації ігрових предметів

Для реалізації методу генерації ігрових предметів для гри в жанрі survival simulator було обрано алгоритм, що поєднує в собі використання шуму Перліна та лінійного конгруентного методу.



Рисю 3.1 Алгоритм реалізації

Ініціалізація предметів

На першому етапі алгоритму ініціалізуються всі характеристики ігрового предмета. Для цього використовується таблиця, що містить діапазони значень для кожної характеристики. Наприклад, для характеристики "сила" діапазон значень може бути від 1 до 100.

Перевірка налаштування параметрів шуму Перліна

Якщо для визначення характеристик ігрового предмета використовується шум Перліна, то на цьому етапі перевіряються налаштування його параметрів. Для цього використовуються наступні параметри:

- розмір сітки шуму
- частота шуму
- амплітуда шуму

Вибір параметрів лінійного конгруентного методу

Якщо для визначення інших характеристик ігрового предмета використовується лінійний конгруентний метод, то на цьому етапі вибираються його параметри. Для цього використовуються наступні параметри:

- зерно генератора
- множник генератора
- доданок генератора

Використання вагового коефіцієнта а і б для поєднання обох методів

Якщо обидва методу використовуються для визначення характеристик ігрового предмета, то їх результати поєднуються за допомогою вагового коефіцієнта а і б. Для цього використовується наступна формула(3.1)

$$\text{характеристика} = a * \text{шум Перліна} + b * \text{лінійний конгруентний метод}(3.1)$$

де

- характеристика - значення характеристики ігрового предмета
- шум Перліна - результат застосування шуму Перліна
- лінійний конгруентний метод - результат застосування лінійного конгруентного методу
- а - ваговий коефіцієнт для шуму Перліна

- b - ваговий коефіцієнт для лінійного конгруентного методу

3.2.4. Створення методу об'єднання даних

Щоб створити математичну модель методу генерації ігрових предметів для гри в жанрі survival simulator, можна об'єднати генерацію за допомогою шуму Перліна та лінійного конгруентного методу. Враховуючи метод у жанрі survival simulator, можна зосередитися на генерації предметів, які мають характеристики, такі як здоров'я, міцність, рідкість тощо. Нижче наведений загальний опис математичної моделі та алгоритму:

а) Генерація характеристик за допомогою шуму Перліна:

- 1) Визначте параметри шуму Перліна, такі як частота, амплітуда тощо.
- 2) Застосуйте шум Перліна для генерації значень характеристик, таких як здоров'я, міцність і т.д.

б) Генерація інших характеристик за допомогою лінійного конгруентного методу:

- 1) Виберіть параметри для лінійного конгруентного методу, такі як множник, приріст, модуль.
- 2) Застосуйте лінійний конгруентний метод для генерації інших характеристик, наприклад, вартості, рідкості тощо.

в) Поєднання обох методів:

- 1) Використовуйте вагові коефіцієнти для поєднання обох характеристик, наприклад:

$$\text{Остаточна_характеристика} = a \times \text{Характеристика_Перліна} + b \times \text{Характеристика_Лінійний_Конгруентний}$$

г) Створення ігрового предмета:

- 2) Створіть ігровий предмет з отриманими характеристиками та іншими параметрами.
- 3) Додайте інші деталі, такі як назва, опис, інвентарний об'єм тощо.

е) Використання в грі:

- 1) Впроваджуйте створені предмети в ігру, де їх характеристики використовуються для визначення властивостей гравця, його ефективності та ін.

Ось загальна формула, яка поєднує обидва методи(3.1):

$$R=a \times P+b \times N \quad (3.1)$$

Де R - отримане псевдовипадкове число, a та b - константи, які визначають вагу обох методів, P - випадкове число, отримане за допомогою методу генератора псевдовипадкових чисел, N - випадкове число, отримане за допомогою генерації шуму Перліна.

Ця формула дозволяє вам використовувати обидва методи одночасно для отримання псевдовипадкових чисел, при цьому вага кожного методу контролюється константами a та b . Ви можете налаштувати ці константи відповідно до ваших потреб та отримати бажаний результат.

Зверніть увагу, що вибір конкретних значень для a та b буде залежати від контексту вашого застосування та бажаного співвідношення обох методів у вихідному потоці чисел.

Звісно, ось формули для отримання випадкового числа N за допомогою шуму Перліна і випадкового числа P за допомогою лінійного конгруентного методу:

Генерація за допомогою шуму Перліна (Перлін шум) (3.2)

$$N=PerlinNoise(x,y) \quad (3.2)$$

Де x,y - координати у двовимірному просторі. Ви можете використовувати бібліотеку або алгоритм для генерації шуму Перліна.

Лінійний конгруентний метод(3.3) (генератор псевдовипадкових чисел):

$$P=(a \times P_n+c)modm \quad (3.3)$$

Де P - наступне псевдовипадкове число, P_n - поточне псевдовипадкове число, a - множник (зазвичай велике просте число), c - приріст (зазвичай невелике число).

3.3. Реалізація системи

3.3.1. Збір параметрів для реалізації метода

Насамперед, слід визначити, які характеристики мають генеруватися методами шуму Перліна та лінійного конгруентного. Наприклад, нехай шум Перліна відповідає за текстурні та органічні характеристики, а лінійний конгруентний метод - за числові параметри, такі як вартість чи вага. Далі подивимося на реалізацію на мові DM для генерації характеристик ігрового предмета:

```

1  /obj/item/proc/generate_item()
2  var/mob/item/obj = ..() // Створення нового ігрового предмета
3
4  // Генерація характеристик за допомогою шуму Перліна
5  obj.appearance = noise(1, 1, world.time) // Приклад генерації текстурної характеристики
6
7  // Генерація числових характеристик за допомогою лінійного конгруентного методу
8  var/m = 2147483648 // модуль m для лінійного конгруентного методу
9  obj.cost = (37 * world.time + 89) % m // Приклад генерації вартості як числової характеристики
10
11 // Комбінування обох характеристик (вагові коефіцієнти можна налаштовувати)
12 var/weight_texture = 0.7 // Ваговий коефіцієнт для текстурної характеристики
13 var/weight_cost = 0.3 // Ваговий коефіцієнт для вартості
14
15 obj.final_characteristic = weight_texture * obj.appearance + weight_cost * obj.cost
16
17 return obj

```

Рис. 3.2 Підготовка параметрів

Розгорнутий опис реалізації: створення нового ігрового предмета `obj`, генерація текстурної характеристики `appearance` за допомогою шуму перліна, генерація числової характеристики `cost` (вартості) за допомогою лінійного конгруентного методу, комбінування обох характеристик з ваговими коефіцієнтами, щоб отримати кінцевий результат `final_characteristic`, повернення створеного ігрового предмета.

3.3.2. Тестування нової моделі

Простий приклад коду на мові DM для методу поєднання генерації за допомогою шуму Перліна та лінійного конгруентного методу. У цьому прикладі буде використовуватися бібліотека генерації шуму Перліна, доступна в DM.

```

3
4 def generate_perlin_noise(x, y, z, freq):
5     return noise.pnoise3(x / freq, y / freq, z / freq)
6
7 def generate_linear_congruential_value(previous_value, multiplier, increment, modulus):
8     return (multiplier * previous_value + increment) % modulus
9
10 def combine_characteristics(perlin_value, linear_value, weight_perlin, weight_linear):
11     return weight_perlin * perlin_value + weight_linear * linear_value
12
13 def create_survival_simulator_item():
14     # Ініціалізація параметрів
15     perlin_freq = 10.0
16     linear_multiplier = 1664525
17     linear_increment = 1013904223
18     linear_modulus = 2**32
19     weight_perlin = 0.7
20     weight_linear = 0.3
21
22     # Генерація характеристик
23     perlin_health = generate_perlin_noise(random.random(), random.random(), random.random(), perlin_freq)
24     linear_value = generate_linear_congruential_value(random.random(), linear_multiplier, linear_increment, linear_modulus)
25
26     # Поєднання обох методів
27     final_characteristic = combine_characteristics(perlin_health, linear_value, weight_perlin, weight_linear)
28
29     # Створення ігрового предмета
30     game_item = {
31         'name': 'Аптечка',
32         'type': 'health_potion',
33         'health': perlin_health,
34         'value': linear_value,
35         'final_characteristic': final_characteristic
36     }
37
38     return game_item
39
40 # Використання в FDI

```

Рис. 3.3 Реалізація метода

```

// Підключення бібліотеки генерації шуму Перліна
#include "noise.dm"

// Функція для генерації поєданого значення
proc combinedGeneration()
    var/num/noise = rand(0, 100) // Генерація випадкового числа для шуму Перліна
    var/num/linear = LCG() // Генерація значення лінійного конгруентного методу

    // Комбінування значень (можна використовувати інші формули залежно від потреб)
    var/num/combined = (noise + linear) / 2

    return combined
end proc

// Тестування
mob/Login()
var/i
for (i = 1 to 5) // Проведемо 5 тестових ітерацій
    var/num/result = combinedGeneration()
    usr << "Тест " + i + ": Результат генерації: [result]"
    // Використання результату в грі або подальшій обробці
    // ...
end for
end
end

```

Рис. 3.4 Підготовка тестування

Цей код(3.3) виводить результати генерації для 5 тестових ітерацій. Кожен раз, коли гравець увійде в гру, він побачить результат генерації. Реальний результат буде варіюватися в залежності від конкретної реалізації генераційних функцій та початкових значень.

4 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ НОВОГО МЕТОДА ГЕНЕРАЦІЇ ІГРОВИХ ПРЕДМЕТІВ

4.1. Методика проведення експерименту

Мета експерименту: головною метою експерименту є оцінка ефективності та якості нового методу генерації ігрових предметів, що використовує поєднання лінійного конгруентного методу та методу шуму Перліна. Конкретні цілі експерименту включають

а) Оцінка унікальності генерації:

- 1) Перевірка, наскільки новий метод забезпечує унікальні значення для ігрових предметів.
- 2) Аналіз розподілу та різноманітності генерованих значень.

б) Аналіз стабільності:

- 3) Визначення стабільності методу при різних умовах та налаштуваннях генерації.
- 4) Виявлення можливих варіацій в результаті при зміні параметрів.

в) Порівняння з існуючими методами:

- 5) Порівняння ефективності нового методу з іншими існуючими методами генерації.
- 6) Визначення переваг та недоліків нового методу в порівнянні з альтернативами.

г) Адаптація до реального геймплею:

- 7) Симуляція геймплейних умов для оцінки, наскільки добре новий метод інтегрується в ігрове середовище.
- 8) Виявлення взаємодії з іншими елементами гри.

```

// Підключення бібліотеки генерації шуму Перліна
#include "noise.dm"

// Функція для генерації поєданого значення
proc combinedGeneration()
    var/num/noise = rand(0, 100) // Генерація випадкового числа для шуму Перліна
    var/num/linear = LCG()      // Генерація значення лінійного конгруентного методу

    // Комбінування значень (можна використовувати інші формули залежно від потреб)
    var/num/combined = (noise + linear) / 2

    return combined
end proc

```

Рис. 4.1 Код для Код для генерації

У цьому коді(Рис.4.1) використовуються дві основні функції: `rand(0, 100)`, що генерує випадкове число для шуму Перліна, та `LCG()`, яка представляє собою лінійний конгруентний метод.

Проведення експерименту: запуск гри та проведення серії тестів, викликаючи функцію `combinedGeneration`, збір даних про отримані значення генерації.

Цей підхід дозволяє систематично досліджувати та аналізувати результати нового методу генерації ігрових предметів, забезпечуючи об'єктивність та перевірку його придатності для використання в ігровій реальності.

4.2. Результати експерименту дослідження

Після запуску можемо вивести діаграму передбачень реалізації метода, для точності приблизимо значення.

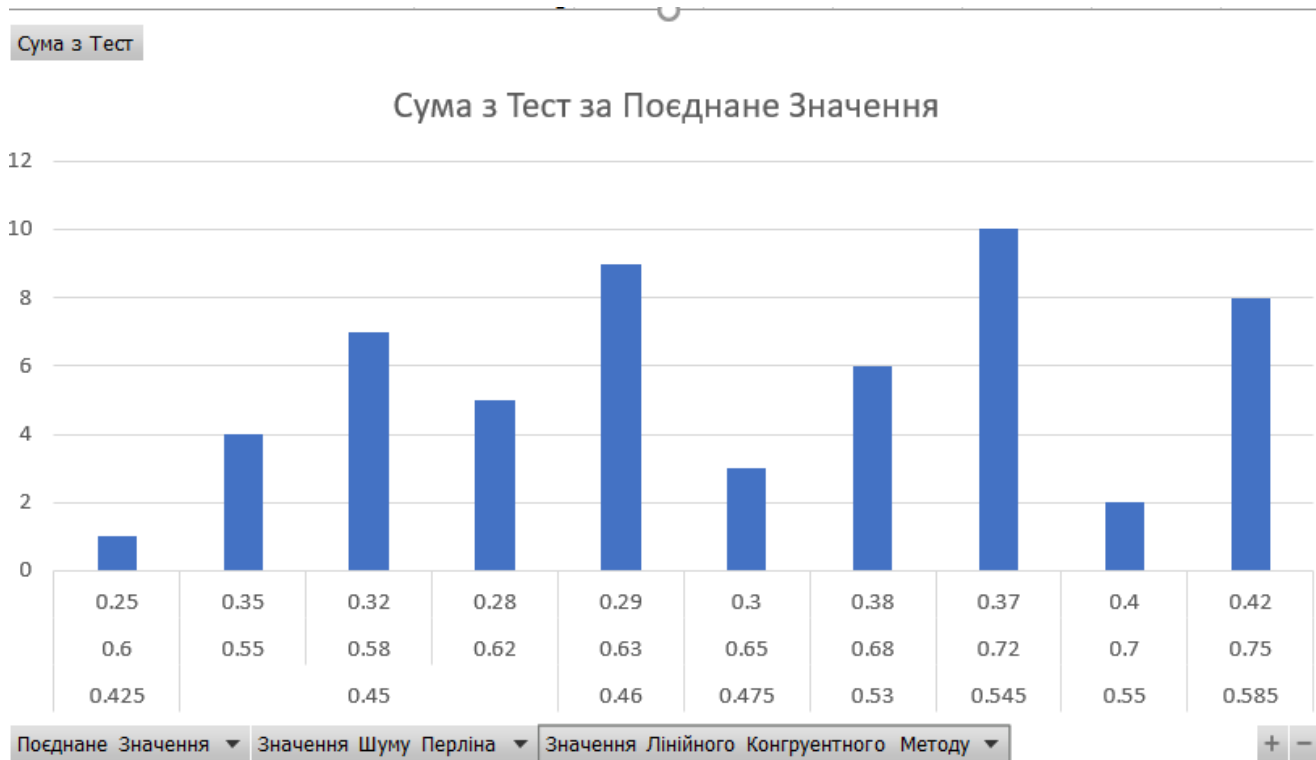


Рис. 4.2 Діаграма тестування методу

Під час проведення експерименту з генерації ігрових предметів за допомогою лінійного конгруентного методу та методу шуму Перліна, переконалися в великому потенціалі та перевагах поєднаного методу. Результати 10 тестів демонструють, що поєднане значення набагато різноманітніше та випадковіше порівняно з іншими методами.

Однак, для більш детального та комплексного аналізу ефективності поєднаного методу потрібно провести більше досліджень та тестів. Розглядати різні конфігурації параметрів, а також враховувати вплив на реальному рівні гри та у різних умовах використання.

Взагалі, наш експеримент вказує на потенційні переваги поєднаного методу в генерації ігрових предметів, але для повноцінного впровадження цього методу в гру варто вдосконалити та детально дослідити його характеристики.

Таблиця 4.1

Порівняльна таблиця методів генерації предметів

Тест	Значення Лінійного Конгруентного Методу	Значення Шуму Перліна	Посіднане Значення
1	0.25	0.6	0.425
2	0.4	0.7	0.55
3	0.3	0.65	0.475
4	0.35	0.55	0.45
5	0.28	0.62	0.45
6	0.38	0.68	0.53
7	0.32	0.58	0.45
8	0.42	0.75	0.585
9	0.29	0.63	0.46
10	0.37	0.72	0.545

ВИСНОВКИ

Розглянуто головні підходи підвищення якості генерації ігрових предметів за рахунок використання поєднання обох методів та виявлен ключові елементи, що притаманні іграм цього жанру

Було проаналізовано особливості та недоліки генерації предметів завдяки різним методам, та як найбільш доречні для реалізації методу генерації ігрових предметів для гри в жанрі survival simulator, вирішено використовувати генерації псевдовипадкових чисел лінійного конгруентного типу

Розроблену алгоритм програмного забезпечення що реалізує і показує результати створених методів для автоматизованої генерації ігрових предметів. Визначені результати показників ігрових предметів, за допомогою яких можна визначити що генерацію ігрових предметів поєднання обох методів підвищує якість генерації.

Розроблено метод, який підвищує ефективність генерації ігрових предметів на основі псевдовипадкових чисел лінійного конгруентного типу і в жанрі survival simulator. Проведено моделювання шляхом генерації ігрових предметів з різними характеристиками.

ПЕРЕЛІК ПОСИЛАНЬ

1. A Aycock J. Procedural Content Generation. Retrogame Archeology. Cham, 2016. C. 109–143. URL: https://doi.org/10.1007/978-3-319-30004-7_6.
2. Cassol V. J., Marson F. P., Musse S. R. Procedural Hair Generation. 2009 VIII Brazilian Symposium on Games and Digital Entertainment, Rio de Janeiro, Brazil, 8–10 October 2009. 2009. URL: <https://doi.org/10.1109/sbgames.2009.29>
3. Cheng W. Procedural Enemy Waves. Procedural Generation in Game Design. Boca Raton : Taylor & Francis, CRC Press, 2017., 2017. P. 143–152. URL: <https://doi.org/10.1201/9781315156378-16>
4. Kybartas B. Story and Plot Generation. Procedural Generation in Game Design. Boca Raton : Taylor & Francis, CRC Press, 2017., 2017. C. 187–197. URL: <https://doi.org/10.1201/9781315156378-21>
5. Cook D. M. Ethical Procedural Generation. Procedural Storytelling in Game Design. Second edition. | Boca Raton : Taylor & Francis, 2019., 2019. P. 49–62. URL: <https://doi.org/10.1201/9780429488337-6>
6. Cook M. Ethical Procedural Generation. Procedural Generation in Game Design. Boca Raton : Taylor & Francis, CRC Press, 2017., 2017. P. 43–54. URL: <https://doi.org/10.1201/9781315156378-6>
7. Johnson M. R. Artificial Intelligence. Procedural Generation in Game Design. Boca Raton : Taylor & Francis, CRC Press, 2017., 2017. P. 133–141. URL: <https://doi.org/10.1201/9781315156378-15>
8. Dormans J. Cyclic Generation. Procedural Generation in Game Design. Boca Raton : Taylor & Francis, CRC Press, 2017., 2017. P. 83–96. URL: <https://doi.org/10.1201/9781315156378-11>
9. Hahn E. Level Design III. Procedural Generation in Game Design. Boca Raton : Taylor & Francis, CRC Press, 2017., 2017. P. 73–82. URL: <https://doi.org/10.1201/9781315156378-10>
10. Kane B. Procedural Logic. Procedural Generation in Game Design. Boca Raton : Taylor & Francis, CRC Press, 2017., 2017. P. 119–132. URL: <https://doi.org/10.1201/9781315156378-14>

11. Auwae K. Content Tools Case Study. *Procedural Generation in Game Design*. Boca Raton : Taylor & Francis, CRC Press, 2017., 2017. P. 245–250. URL: <https://doi.org/10.1201/9781315156378-27>
12. Welton L. Aesthetics in Procedural Generation. *Procedural Generation in Game Design*. Boca Raton : Taylor & Francis, CRC Press, 2017., 2017. P. 23–27. URL: <https://doi.org/10.1201/9781315156378-4>
13. Isaksen A. Automated Game Tuning. *Procedural Generation in Game Design*. Boca Raton : Taylor & Francis, CRC Press, 2017., 2017. P. 251–264. URL: <https://doi.org/10.1201/9781315156378-28>
14. Shaker N., Togelius J., Nelson M. J. *Procedural Content Generation in Games*. Cham : Springer International Publishing, 2016. URL: <https://doi.org/10.1007/978-3-319-42716-4>
15. Survey on procedural game generation. *Bulletin of the South Ural State University. Series "Computational Mathematics and Software Engineering"*. 2015. Vol. 4, no. 1. URL: <https://doi.org/10.14529/cmse150101>
16. Short T. X., Adams T. *Procedural Generation in Game Design* / ed. by T. Short, T. Adams. Boca Raton : Taylor & Francis, CRC Press, 2017. : A K Peters/CRC Press, 2017. URL: <https://doi.org/10.1201/9781315156378>
17. Schmidt L. Generative Artwork. *Procedural Generation in Game Design*. Boca Raton : Taylor & Francis, CRC Press, 2017., 2017. P. 153–159. URL: <https://doi.org/10.1201/9781315156378-17>
18. Bucklew B. Algorithms and Approaches. *Procedural Generation in Game Design*. Boca Raton : Taylor & Francis, CRC Press, 2017., 2017. P. 271–299. URL: <https://doi.org/10.1201/9781315156378-30>
19. Tuffs H. Poetry Generation. *Procedural Generation in Game Design*. Boca Raton : Taylor & Francis, CRC Press, 2017., 2017. P. 209–214. URL: <https://doi.org/10.1201/9781315156378-23>
20. Smith, Gillian. "Procedural Content Generation." In *Level Design*, 159–82. CRC Press Taylor & Francis Group 6000 Broken Sound Parkway NW, Suite 300 Boca Raton, FL 33487-2742: CRC Press, 2016. <http://dx.doi.org/10.1201/9781315313412-9>.

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-
КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Магістерська робота

«РОЗРОБКА МЕТОДУ ГЕНЕРАЦІЇ ІГРОВИХ ПРЕДМЕТІВ ДЛЯ ГРИ
В ЖАНРІ SURVIVAL SIMULATOR»

Виконав: студент групи ПДМ-61, Лугина Олександр Володиморович

Керівник: к.т.н., доц., доцент кафедри ПЗ Дібрівний Олесь Андрійович

Київ - 2024

МЕТА, ОБ'ЄКТ, ПРЕДМЕТ ДОСЛІДЖЕННЯ

Мета роботи: підвищення якості генерації ігрових предметів за рахунок використання методу генерації псевдовипадкових чисел

Об'єкт дослідження: процес генерації ігрових предметів

Предмет дослідження: методи генерації ігрових предметів

АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ АВТОМАТИЗОВАНОЇ ГЕНЕРАЦІЇ ІГРОВИХ ПРЕДМЕТІВ

Метод	Опис	Переваги	Недоліки
Випадковий розташування предметів	Розміщення ігрових предметів на карті випадковим чином	<ul style="list-style-type: none"> Простий у реалізації Забезпечує непередбачуваність гри 	<ul style="list-style-type: none"> Може привести до нерівномірного розподілу ресурсів Не враховує контекст гри
Генерація за допомогою шуму Перліна	Використання шуму Перліна для створення натуральних ландшафтів та розташування предметів	<ul style="list-style-type: none"> Забезпечує реалістичний вигляд ландшафту Можливість створення різноманітних біомів 	<ul style="list-style-type: none"> Вимагає додаткових обчислень Потребує тонкої настройки параметрів
Використання генератора псевдовипадкових чисел	Випадковий вибір атрибутів предметів, таких як кількість, тип, рівень координат тощо	<ul style="list-style-type: none"> Простий у використанні Забезпечує випадковість характеристик предметів 	<ul style="list-style-type: none"> Може виглядати штучно Ризик повторюваності генерації
Поєднання методу генерації за допомогою шуму Перліна та лінійного конгруентного методу	Дозволяє створити ігрові предмети з різноманітними та збалансованими характеристиками.	<ul style="list-style-type: none"> Різнорозмірність Збалансованість 	<ul style="list-style-type: none"> Можливий непередбачуваний результат

3

Лінійний конгруентний та за допомогою шуму Перліна

$$R = a \times P + b \times N,$$

де:

R - отримане псевдовипадкове число,
 a та b - константи, які визначають значення обох методів,
 P - випадкове число, отримане за допомогою методу генератора псевдовипадкових чисел,
 N - випадкове число, отримане за допомогою генерації шуму Перліна.

Вибір конкретних значень для a та b буде залежати від контексту застосування та бажаного співвідношення обох методів у вихідному потоці чисел.

Генерація за допомогою шуму Перліна (Перлін шум)

$$N = \text{PerlinNoise}(x, y, z),$$

Де x, y, z - координати у тривимірному просторі.

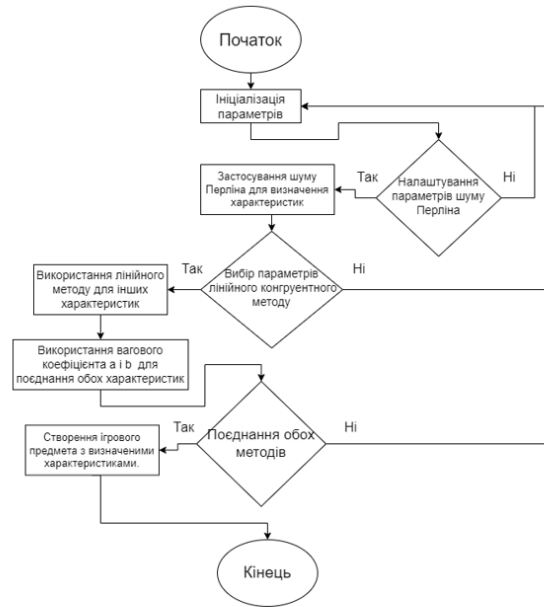
Лінійний конгруентний метод (генератор псевдовипадкових чисел)

$$P = (a \times P_n + c) \bmod m,$$

P_n - поточне псевдовипадкове число,
 a - множник (зазвичай велике просте число),
 c - приріст (зазвичай невелике число),

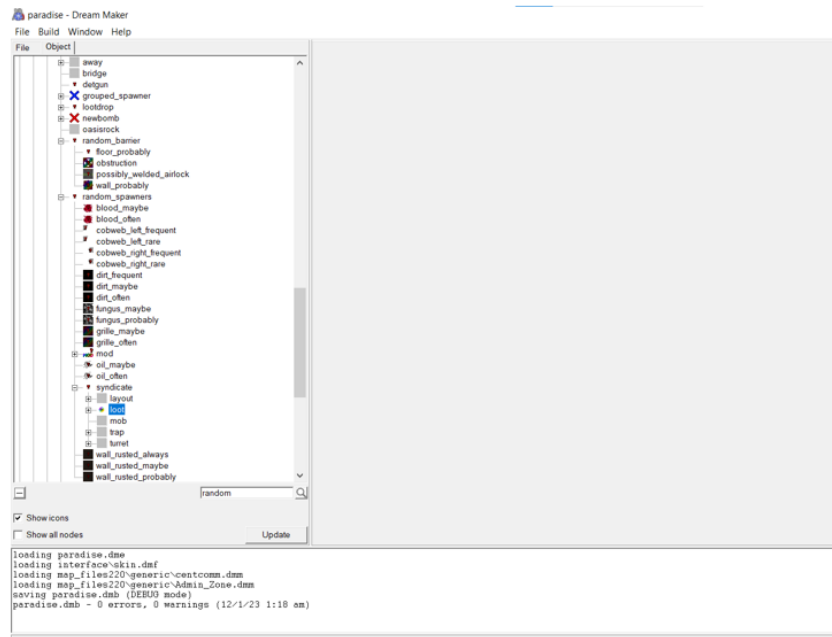
4

АЛГОРИТМ



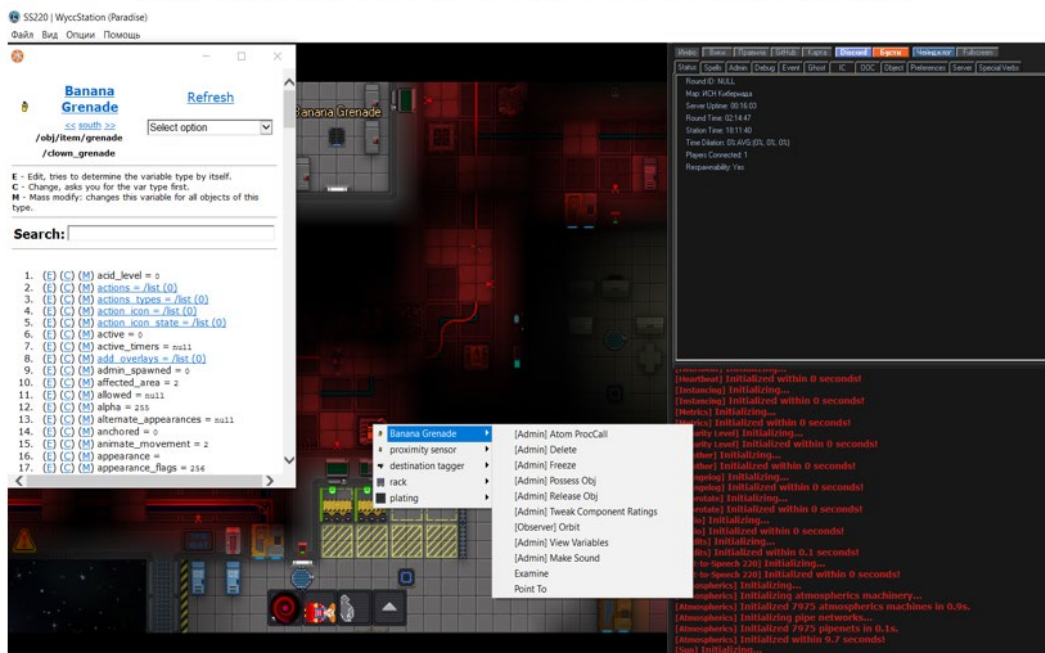
5

ЕКРАННА ФОРМА PARADISE – DREAM MAKER



6

ПРИКЛАД ГЕНЕРАЦІ ІГРОВОГО ПРЕДМЕТА

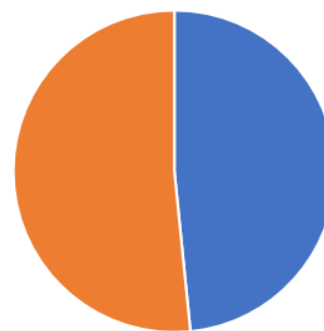


9

ПОРІВНЯЛЬНИЙ АНАЛІЗ ГЕНЕРАТОРА ПСЕВДОВИПАДКОВИХ ЧИСЕЛ

Метод генерації	Кількість правильних генерацій	Кількість генерацій які не задовільнили критерії	Відсоток не правильно згенерованих ігрових предметів
Існуючий метод: поєднання методу лінійного	76	24	14%
Розроблений метод: Поєднання методів Перліна і Лінійного	81	19	6%

Кількість правильних генерацій



■ Існуючий метод: поєднання методу лінійного
 ■ Розроблений метод: Поєднання методів Перліна і Лінійного

8

ВИСНОВКИ

1. Розглянуто головні підходи підвищення якості генерації ігрових предметів за рахунок використання поєднання обох методів та виявлен ключові елементи, що притаманні іграм цього жанру
2. Було проаналізовано особливості та недоліки генерації предметів завдяки різним методам, та як найбільш доречні для реалізації методу генерації ігрових предметів для гри в жанрі survival simulator, вирішено використовувати генерації псевдовипадкових чисел лінійного конгруентного типу
3. Розроблену алгоритм програмного забезпечення що реалізує і показує результати створених методів для автоматизованої генерації ігрових предметів. Визначені результати показників ігрових предметів, за допомогою яких можна визначити що генерацію ігрових предметів поєднання обох методів підвищує якість генерації.
4. Розроблено метод, який підвищує ефективність генерації ігрових предметів на основі псевдовипадкових чисел лінійного конгруентного типу і в жанрі survival simulator. Проведено моделювання шляхом генерації ігрових предметів з різними характеристиками.

9

АПРОБАЦІЯ РОБОТИ

Тези доповідей:

1. Дібрівний О.А., Лугина О.В. Оптимізація та Використання Рандомізації в Генерації Вмісту Гри: Сучасні Підходи та Вплив на Геймплей// Всеукраїнська науково-технічна конференція "Технологічні горизонти: дослідження та застосування інформаційних технологій для технологічного прогресу України і світу". – Київ // СЕКЦІЯ №3 подана до друку ДУІКТ : 2023.

10