УДК 004.42

РОЗРОБКА ROGUELIKE ГРИ НА UNITY

Валентин Кириченко

DEVELOPMENT OF A ROGUELIKE GAME ON UNITY

Valentyn Kyrychenko

Волинський національний університет імені Лесі Українки, просп. Волі, 13, Луцьк, 43025, Україна

Abstract. *This paper presents the development process of a RogueLike game using the Unity game engine. The research explores key aspects of the genre and the technical implementation within Unity. The result is a playable prototype demonstrating core RogueLike mechanics. The findings highlight the capabilities of Unity for indie game development.*

Актуальність. Жанр RogueLike ігор вирізняється високою реіграбельністю та можливістю створення унікального ігрового досвіду за рахунок процедурної генерації рівнів, випадкових подій та перманентної смерті персонажа. Популярність інді-ігор, а також доступність потужних і гнучких ігрових рушіїв, таких як Unity, роблять розробку ігор у цьому жанрі актуальним завданням. Створення власної RogueLike гри дозволяє дослідити принципи процедурної генерації контенту, розробки складних ігрових механік та оптимізації продуктивності.

Мета роботи. Метою роботи є розробка функціональної RogueLike гри з використанням ігрового рушія Unity. В рамках досягнення мети планується реалізувати ключові елементи жанру, такі як процедурна генерація підземель, система покрокового бою, інвентар та розвиток персонажа, а також забезпечити базову візуальну та звукову складові.

Результати. В рамках дослідження було реалізовано повноцінний прототип RogueLike гри на базі ігрового рушія Unity. Ключовим досягненням стала розробка та інтеграція системи процедурної генерації підземелля. Був розроблений та імплементований алгоритм, що дозволяє динамічно створювати унікальні ігрові рівні при кожному новому проходженні, використовуючи структуру кімнат, поєднаних коридорами. Ця система забезпечує високу реіграбельність та непередбачуваність дослідження ігрового світу.

Паралельно з генерацією середовища, було розроблено та вдосконалено основні ігрові механіки жанру. Реалізовано покрокову бойову систему, де гравець та ігрові сутності (вороги) взаємодіють у послідовних ходах. Кожна сутність оснащена компонентами, що визначають її характеристики (Fighter.cs) та поведінку (AI.cs). Були розроблені різні типи супротивників з різною логікою поведінки, включаючи агресивних ворогів (HostileEnemy.cs) та сутності зі зміненою поведінкою під впливом ефектів (ConfusedEnemy.cs).

Важливою частиною проєкту стала розробка системи інвентарю (Inventory.cs) та інтерактивних предметів (Item.cs, Consumable.cs). Були реалізовані різноманітні споживані предмети, такі як зілля для відновлення здоров'я (Healing.cs) та магічні сувої з активними ефектами (наприклад, Fireball.cs для атаки по площі, Lightning.cs для точкової атаки, Confusion.cs для зміни поведінки ворогів). Ці предмети додають тактичну глибину ігровому процесу та розширюють можливості гравця.

Для забезпечення навігації супротивників та визначення областей видимості гравця на процедурно згенерованих рівнях були інтегровані та адаптовані відповідні алгоритми. Зокрема, реалізовано алгоритм пошуку шляху A* (AStar.cs) для ефективного переміщення

ворогів до цілі та алгоритми визначення видимості (AdamMilVisibility.cs, BresenhamLine.cs), що дозволяє коректно відображати "туман війни" та поле зору персонажа.

Технічна реалізація проєкту здійснювалась з використанням скриптів на мові програмування C# в інтегрованому середовищі розробки Unity. Код проєкту має модульну структуру, розділену на логічні компоненти (сутності, алгоритми, елементи карти), що сприяло ефективній розробці та подальшому масштабуванню. Були інтегровані додаткові інструменти (як видно з папки Plugins, наприклад, OdinSerializer для серіалізації даних та TextMesh Pro для роботи з текстом), що оптимізувало певні аспекти розробки. Також реалізовано систему збереження та завантаження прогресу гри (SaveManager.cs) та базовий інтерфейс користувача (UIManager.cs), включаючи головне меню (MainMenu.cs), для забезпечення повноцінного ігрового циклу.

Процес розробки включав проєктування архітектури ігрових класів на C# в середовищі Unity, роботу зі сценами та ігровими об'єктами, налагодження взаємодії між різними компонентами, а також створення базового інтерфейсу користувача (UIManager.cs) для відображення ігрової інформації. Було враховано необхідність збереження прогресу гри (SaveManager.cs). Отриманий в результаті роботи прототип є функціональним, демонструє реалізацію основних механік RogueLike жанру та готовий до подальшого розширення та вдосконалення.

Висновок. Таким чином, виконана робота підтвердила можливість та ефективність використання ігрового рушія Unity для розробки комплексних ігор жанру RogueLike силами одного розробника. Успішна реалізація процедурної генерації рівнів, покрокової бойової системи, системи інвентарю з різноманітними предметами, а також інтеграція алгоритмів навігації та видимості демонструє гнучкість та потужність Unity як платформи для інді-розробки. Створений прототип є не просто концептуальним підтвердженням, а функціональною основою, яка закладає фундамент для майбутнього розвитку проєкту. Перспективи подальших досліджень та розробок включають розширення контенту гри (нові типи ворогів, предметів, механік), вдосконалення алгоритмів генерації та ШІ, покращення візуальної та звукової складових, а також оптимізацію продуктивності. Результати роботи мають практичну цінність як демонстрація можливостей Unity для створення ігор з процедурним контентом та складними системними взаємодіями, а також як фундамент для подальших експериментів у рамках жанру RogueLike.

Бібліографія

1. RogueBasin: The Official Rogue Wiki [Електронний ресурс] // RogueBasin. URL: https://roguebasin.com/
2. Unity Manual [Електронний ресурс] // Unity Technologies. URL: https://docs.unity3d.com/Manual/
3. Unity Scripting API [Електронний ресурс] // Unity Technologies. URL: https://docs.unity3d.com/ScriptReference/