

АЛГОРИТМ ОПТИМІЗАЦІЇ БОЙОВИХ МЕХАНІК І ШТУЧНОГО ІНТЕЛЕКТУ ВОРОГІВ У ГРІ

У розробці відеоігор важливим аспектом є досягнення балансу між складністю гри та ефективністю використання обчислювальних ресурсів. Цей баланс безпосередньо впливає на динаміку гри та здатність залучати гравців. Однією з основних проблем є те, що підвищення складності зазвичай призводить до значного збільшення обчислювальних витрат, що може погіршити продуктивність, особливо на пристроях з обмеженими ресурсами.

Метою цього дослідження є створення алгоритмів оптимізації механік та алгоритми AI ворогів, забезпечуючи баланс між складністю ігрового процесу та ефективністю використання ресурсів. Розроблені алгоритми повинні дозволити забезпечуючи стабільний ігровий досвід.

Для досягнення поставленої мети, було запропоновано алгоритм оптимізації бойових механік та інтелекту ворогів, який складається з кількох етапів:

1. Аналіз параметрів складності і ресурсів[1]:

На цьому етапі проводиться ідентифікація основних параметрів, що впливають на складність гри (кількість ворогів, їхня агресивність, швидкість реакцій) і ресурси, необхідні для їх обробки. Це дозволяє сформулювати модель для оптимізації ігрових процесів.

2. Моделювання бойових механік:

Розробка алгоритмів, які змінюють механіки боїв залежно від дій гравця. Зокрема, кількість ворогів і рівень їхньої агресивності змінюються в залежності від того, як швидко і ефективно гравець просувається в грі.

3. Моделювання витрат ресурсів:

Для оцінки ефективності оптимізації застосовуються математичні моделі, які описують взаємодію між складністю гри та обчислювальними витратами. Це дозволяє точно прогнозувати, як зміни в складності гри впливають на навантаження на систему і як оптимізувати ці процеси.

Математична модель для оцінки витрат ресурсів має такий вигляд:

$$R = E \cdot T \cdot S$$

де:

R — витрати ресурсів,

E — кількість ворогів,

T — час, необхідний для обробки кожного ворога,

S — складність бою.

На попередніх етапах у проєкт уже були інтегровані патерни[2] серед яких State (FiniteStateMachine) і DI-фреймворк Zenject, які забезпечують гнучкість управління станами ворогів і організацію залежностей між об'єктами. Однак аналіз показав, що зростання кількості ворогів призводить до збільшення часу обробки, а велика кількість викликів ін'єкцій у Zenject значно підвищує витрати ресурсів. Це вимагає подальшої оптимізації.

Після розробки алгоритмів та моделей, проводиться тестування[**Ошибка! Источник ссылки не найден.**] на прототипах гри. Результати тестування порівнюються з попередніми варіантами, що дозволяє оцінити продуктивність нових рішень і виявити найефективніші методи оптимізації.

У доповіді буде представлено результати оптимізації бойових механік і алгоритмів штучного інтелекту ворогів у грі. Та розгляд математичної моделі для оцінки складності і ресурсних витрат, а також запропоновані шляхи вдосконалення існуючих рішень через адаптацію сучасних патернів (State, Zenject).

Список використаних джерел

1. Game Learning Analytics: / C. Alonso-Fernández et al. *Journal of Learning Analytics*. 2022. P. 1–18. URL: <https://doi.org/10.18608/jla.2022.7633> (дата звернення: 22.11.2024).
2. Flores N. H., Paiva A. C. R., Letra P. Software Engineering Management Education through Game Design Patterns. *Procedia - Social and Behavioral Sciences*. 2016. Vol. 228. P. 436–442. URL: <https://doi.org/10.1016/j.sbspro.2016.07.067> (дата звернення: 22.11.2024).
3. Rosenfield Boeira J. N. Automated Testing with Unity. *Lean Game Development*. Berkeley, CA, 2023. P. 175–205. URL: https://doi.org/10.1007/978-1-4842-9843-5_15 (дата звернення: 22.11.2024).