Паламарчук М. П., магістрант,
Граф М. С.,Ph.D,
Колос К. Р., д.пед.н., професор
Державний університет «Житомирська політехніка»

АЛГОРИТМИ СИНХРОНІЗАЦІЇ ПРОЦЕСІВ В ОНЛАЙН ГРІ

У сучасних он-лайн іграх синхронізація процесів є одним з найважливіших аспектів. Вона впливає не тільки на плавність ігрового процесу, а і на загальні враження гравців. Якщо для покрокових стратегій синхронізація є чимось другорядним, то для шутерів, навпаки – перетворюється на визначальний фактор. В таких іграх кожен момент стає критичним і найменші затримки можуть вирішально вплинути на результат взаємодії гравців, визначити перемогу чи поразку. Саме тому вдосконалення синхронізації є важливим аспектом при створенні якісних он-лайн ігор. Метою дослідження є аналіз основних алгоритмів і підходів до синхронізації процесів в он-лайн грі.

Найбільша проблема при синхронізації процесів в он-лайн іграх – це час, затрачений на передачу даних. При цьому не завжди є доступ до швидкого та надійного мережевого з'єднання. Щоб компенсувати різницю у часі між сервером та гравцями, використовують обчислювальні потужності гаджетів та різноманітні алгоритми синхронізації процесів. Завдяки цьому, можна скинути частину навантаження з мережі на процесор. Це можна зробити за допомогою таких методів:

- **Застосування алгоритмів стиснення даних:** дозволяє зменшити розмір даних без втрати інформації. Однак цей варіант підходить лише для великих об'ємів інформації, адже час на кодування та розшифрування при невеликій кількості даних може навіть негативно вплинути на розмір даних та витратити час на обробку, що тільки сповільнить передачу [2].
- **Використання шаблонів, заготовок, словників тощо:** будь-яку інформацію, що часто використовується, можна скоротити. Це дещо схоже на алгоритми стиснення даних, але використовується лише для конкретної інформації, заміняючи довгі рядки короткою позначкою. Наприклад, ігровий простір можна розбити на квадрати, додавши кожному буквену позначку. І вже семизначна координата перетворюється на одну букву з двома-трьома цифрами, що змінились. Назви предметів замінити на унікальний номер, при передачі даних.
- **Використання алгоритмів передбачення стану об'єктів:** дозволяє зменшити кількість даних, що передаються, оскільки не завжди необхідно передавати повний стан об'єкта. Наприклад, можна передбачати напрямок руху персонажа на основі його попереднього стану, що дозволить промалювати дії гравця без жодної затримки [1].
- **Оптимізація та фільтрування інформації, що надсилається:**
  - Ігнорування незначних змін в стані об'єктів.

Наприклад, персонаж змінює нахил камери на 1 градус. Такі зміни навіть візуально непомітні для інших гравців і їх не обов'язково передавати на сервер. Також можна проігнорувати анімації та події, що не несуть в собі певної інформації для інших гравців.

  - Періодична синхронізація стану об'єктів.

Наприклад, персонаж горить і здоров'я гравця поступово падає. Серверу не обов'язково фіксувати кожну одиницю здоров'я гравця, ще й надсилати іншим.

  - Паралельна обробка інформації.

Сервер може розподілити обчислювання інформації не тільки між ядрами своїх процесорів, а й змусити гравців самостійно обраховувати події. Повертаючись до попереднього прикладу, сервер може отримати та надіслати лише факт початку події горіння, при цьому кожен гравець самостійно обрахує падіння здоров'я конкретного персонажа, а в кінці сервер тільки перевірить коректність результату.

  - Відправляти лише ту інформацію, яка стосується гравця.

На масштабній ігровій локації гравець бачить лише невелику частину, тому недоцільно відправляти йому усе що відбувається в кожному кутку гри.

Отже, оптимізація синхронізації в он-лайн іграх через використання алгоритмів і технік, що переносять частину навантаження з мережі на процесори, може значно поліпшити ігровий досвід, забезпечуючи ефективну взаємодію гравців при обмежених ресурсах мережі.

Список використаних джерел

1. Ferretti S., Roccetti M., Palazzi C. E. An Optimistic Obsolescence-Based Approach to Event Synchronization for Massively Multiplayer Online Games. International Journal of Computers and Applications. 2007. Vol. 29, no. 1. P. 33–43. URL: https://doi.org/10.1080/1206212x.2007.11441830.
2. Gregory J., Peters A. Game Engine Architecture. CRC Press, 2009. 1240 p. URL: https://doi.org/10.1201/b10681.