

УДК 004.75

**Верницький І. Р. студент**

(Тернопільський національний технічний університет ім. І. Пулюя)

## **РЕАЛІЗАЦІЯ СИНХРОНІЗАЦІЇ ТА УЗГОДЖЕННЯ ДАНИХ У БРАУЗЕРНІЙ ГРІ**

UDC 004.75

**Vernytskyi I. R. Student**

## **IMPLEMENTING DATA SYNCHRONIZATION AND RECONCILIATION IN A BROWSER GAME**

В сучасному світі комп'ютерні ігри займають важливу роль у житті багатьох людей. Суть комп'ютерних ігор – отримання задоволення від процесу. Такий ефект досягається завдяки простій схемі – за певну дію, чи послідовність дій гравцю видається винагорода. В більшості ігор цей алгоритм складніший – дія повинна бути правильною, вчасною, комбінуватись з рядом інших, або навіть усе разом. В цій концепції швидкий відгук ігрового стану на дії гравця грає дуже велику роль. Деякі із жанрів ігор буквально зав'язані на правильному, а головне – вчасному натисканні клавіш. Будь то від погано оптимізованого ігрового процесу, чи великого пінгу до сервера, затримка може стати критичним чинником в таких іграх, та перетворити задоволення в цілий ряд негативних емоцій. Адже це руйнує основну концепцію – гравець зробив все правильно та вчасно, але через затримку не отримав нагороди. І якщо оптимізувавши ігровий код доволі просто досягнути непомітної для людського ока затримки – в випадку з пінгом в онлайн іграх ця задача стає в рази складнішою. Адже час проходження пакету від клієнта до сервера зазвичай не залежить від розробника. І поки сервер отримає дані про дію гравця – ця дія може бути уже не актуальною. Рішенням цієї проблеми є прогнозування та узгодження. Після дії гравця клієнт гри одразу відображає її, прогножуючи поведінку ігрового стану. Паралельно клієнт відправляє дані про дію, свою реакцію, та часову мітку на сервер. Сервер тримає в пам'яті декілька десятків останніх ігрових станів. За допомогою часової мітки він визначає під час якого із станів дія була здійснена, повертається до нього, та перевіряє правильність реакції клієнта. Проводиться перерахунок усіх ігрових циклів від моменту натискання гравцем клавіші до актуального на даний момент стану. Після цього сервер відправляє часову мітку, та актуальні дані усім підключеним клієнтам. Отримавши дані, опираючись на часову мітку, та ігрову логіку, клієнти розраховують актуальний ігровий стан, та з використанням інтерполяції переходять в нього.

Таким чином клієнт може майже миттєво реагувати на дії гравця, сервер не втрачає можливості перевіряти інформацію, а на інші клієнти відображають найбільш актуальний ігровий стан. Особливо ефективно такий алгоритм показує себе в іграх, де дії гравця породжують інерційні зміни в ігрових станах, наприклад – клік по ігровому полю генерує шлях, по якому рухається персонаж до точки кліку. В такому випадку, хоч початок руху персонажа на різних клієнтах все ж відбудеться із затримкою – основна частина шляху, та прибуття в кінцеву точку будуть практично синхронними. Для браузерних ігор однією із проблем такого рішення є відсутність однієї часової шкали для всіх пристроїв. На кожному пристрої, включаючи сервер, час зазвичай відрізняється. Ця різниця може коливатись від декількох мілісекунд до декількох секунд. Таким чином часова мітка, створена клієнтом, буде не актуальною. Для вирішення цієї проблеми на початку ігрового процесу потрібно синхронізувати годинники. Сервер відправляє низку пінг-понг запитів, а клієнт до кожного з них додає часову мітку отримання. Різниця часу визначається як усереднене значення усіх різниць часу між клієнтом і сервером з вирахуванням половини величини пінгу кожного запиту.

### **Література.**

1. Стаття What Every Programmer Needs To Know About Game Networking. URL: [https://gafferongames.com/post/what\\_every\\_programmer\\_needs\\_to\\_know\\_about\\_game\\_networking/](https://gafferongames.com/post/what_every_programmer_needs_to_know_about_game_networking/).
2. Стаття On-line игры: взаимодействие с сервером <https://gamedev.ru/code/articles/?id=4255>
3. Стаття NTP. URL: <https://ru.wikipedia.org/wiki/NTP>.