

УДК 004

ONLINE GAME BASED ON SERVER-AUTHORITATIVE ARCHITECTURE

ОНЛАЙН-ГРА НА БАЗІ SERVER-AUTHORITATIVE АРХІТЕКТУРИ

Chykrii A.O. / Чикрій А. О.

student / студент

ORCID: 0009-0001-0972-7113

*National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute",**Kyiv, Prospect Beresteyskiy, 37, 03056**Національний технічний університет України «Київський політехнічний інститут**імені Ігоря Сікорського», Київ, Проспект Берестейський, 37, 03056*

Анотація. Реалізовано програмний продукт, що демонструє основні елементи змагальної онлайн-гри і має мінімальну вразливість щодо можливості ігрового шахрайства. Для реалізації системи було використано: ігровий рушій Unity Engine, бібліотека Netcode for GameObjects, мова програмування C#, середовище розробки Visual Studio Code, система контролю версій Git. Розроблений застосунок дозволяє грати у локальній мережі.

Ключові слова: гра, онлайн-гра, архітектура, авторитетний сервер, мережа, клієнт, сервер, користувач, гравець, чит, сесія, Unity, Netcode, C#.

Abstract. The purpose of developing an online game based on Server-Authoritative architecture is to improve the quality of the game process and reduce the vulnerability to the possibility of game cheating. The following were used to implement the system: Unity Engine game engine, Netcode for GameObjects library, C# programming language, Visual Studio Code development environment, Git version control system. The developed application allows you to play in a local network.

Key words: game, online game, architecture, authoritative server, network, client, server, user, player, cheating, session, Unity, Netcode, C#.

Вступ.

У сучасному світі індустрія цифрових розваг є однією з найбільш швидко зростаючих галузей, яка має величезний вплив на економіку та культуру суспільства. Індустрія цифрових розваг включає в себе виробництво відеоігор, веб-сайтів, мобільних додатків, анімаційних фільмів, музики, книг та інших видів цифрового контенту. Індустрія цифрових розваг також має значний вплив на розвиток технологій та інновацій. Через постійний пошук нових ідей та

технологій, ця галузь стимулює розвиток інформаційних технологій та допомагає у вирішенні складних технічних завдань.

Комп'ютерні ігри є однією з найпопулярніших форм розваг у сучасному світі, а онлайн-ігри стають все більш популярними. Ці ігри мають великий вплив на життя людей та суспільства, сприяючи розвитку креативності, критичного мислення, соціальних навичок та навичок співпраці. Онлайн-ігри, зокрема, є важливим засобом взаємодії між людьми з різних країн та культур. Вони дозволяють користувачам об'єднуватися в онлайн-спільноти та взаємодіяти між собою у віртуальному світі. Це дає можливість людям розширювати соціальне коло знайомств, вчитися комунікувати та співпрацювати з іншими людьми, навчатися вирішувати конфлікти та розвивати навички лідерства. Крім того, онлайн-ігри можуть стимулювати розвиток економіки. Багато онлайн-ігор є комерційними продуктами, які залучають велику кількість гравців та споживачів. Це сприяє розвитку цифрової економіки, створенню нових робочих місць та підтримує інновації в галузі інформаційних технологій.

Проблема читів в онлайн-іграх є однією з найбільш серйозних в ігровій індустрії. Чити (англ. cheats) або чит-програми - це програми або компоненти програм, що надають гравцеві незаконну перевагу в грі. Це можуть бути різноманітні функції, такі як нескінченне здоров'я, невидимість, збільшення швидкості чи точності стрільби, та інші. Використання читів є найпоширенішим видом ігрового шахрайства. Одним з найбільших наслідків використання читів є те, що вони порушують баланс гри, знижуючи її рівень складності та справедливості. Гравці, які використовують чити, мають непорівнянну перевагу перед тими, хто грають чесно, що веде до негативного досвіду для останніх. Крім того, використання читів може мати серйозні наслідки для онлайн-громади та ігрової індустрії в цілому. Щоб запобігти використанню читів, розробники онлайн-ігор використовують різні заходи безпеки, такі як античит-системи, що можуть виявляти та блокувати використання читів у грі. Також можуть бути введені санкції проти гравців, які

використовують чити, такі як блокування акаунту або тимчасова заборона на гру. Але базовою запорукою мінімальної або відсутньої можливості гравців отримати незаконну перевагу є правильні архітектурні підходи на етапі проектування ігрової програми.

Узагальнюючи, проблема ігрового шахрайства в онлайн-іграх є досить складною та актуальною, оскільки вона підриває рівень рівноправності в грі, порушує баланс та впливає на імідж гри та індустрії в цілому. Важливо розробляти та вдосконалювати рішення для забезпечення максимальної безпеки та справедливості для гравців.

Проектування гри

У даній роботі розроблено онлайн-гру, що базується на Server-Authoritative архітектурі для підвищення якості ігрового процесу та зменшення вразливості щодо можливості ігрового шахрайства. Головною ціллю розробки власної онлайн-гри є вирішення наступних проблем: погіршення якості ігрового процесу за великої кількості гравців в одній ігровій сесії або на одному сервері; вразливість гри до прямого втручання в ігровий процес клієнтами за допомогою стороннього ПЗ; наявність на клієнтській стороні даних, що не повинні бути видимі, та доступ до яких може надати перевагу; проблема оптимізації трафіку.

Для цього сформульовано наступні критерії: кожна ігрова сесія повинна обслуговуватися окремим сервером і повинна містити невелику кількість гравців; архітектура програми повинна прагнути до повної авторитетності серверу щодо критичних для ігрового процесу розрахунків; всі важливі для обчислення ігрового процесу дані повинні зберігатись на сервері, а клієнтам передаватися лише в тому обсязі, що призначений для відображення; усі неважливі для ігрового процесу обчислення не повинні синхронізуватись із сервером для економії трафіку.

Дана гра є екшн-грою сесійного виду, формату один проти одного. Сесійна гра – це така, в якій гравці зустрічаються в короткочасних сесіях, які, зазвичай, автоматично створюються для кожного матчу. Ігри такого виду ще називають МОБА (Multiplayer Online Battle Arena, англ. – Багатокористувацька онлайн

битва в арені). Такий формат на сьогодні є найбільш актуальним та поширеним, оскільки дозволяє гравцю самостійно визначати кількість часу, проведеного в грі. Під час розробки гра орієнтуватиметься на платформу ПК зі збереженням можливості подальшої міграції на мобільні пристрої.

Кожен з двох гравців керуватиме своєю групою юнітів (бойова одиниця, англ. unit), по чергово переключаючись між ними. Кожен юніт має певні початкові характеристики, такі як швидкість руху, запас одиниць здоров'я, енергії тощо. Також на всі юніти встановлено модулі, такі як озброєння (віднімає одиниці здоров'я юнітам опонента), модулі ремонту (відновлює одиниці здоров'я іншим юнітам) тощо; та модулі, що модифікуватимуть характеристики одного або декількох юнітів. Модулі поділяються на активні і пасивні. Пасивні надають свої ефекти весь час, в той час як активні потрібно активувати задля їх дії, а їх активація потребуватиме витрат енергії та матиме циклічний характер. Енергія відновлюватиметься з часом.

Кожна ігрова сесія (матч) обмежена за часом та проходить в обмеженій ділянці ігрової сцени (арені). Після вичерпання часу або за межами арени юнітам поступово змінюють деякі характеристики, що прискорить перемогу одного з гравців та закінчення матчу.

Ціллю та завершенням ігрового матчу є знищення всіх юнітів опонента. Задачею кожного гравця є позиціонування своїх юнітів на арені відносно опонента, контроль їх модулів, нанесення якомога більшого ураження юнітам опонента та збереження своїх юнітів шляхом їх своєчасного ремонту. Гравець, що перший знищив всі юніти опонента, вважатиметься переможцем матчу. Склад групи юнітів, їх початкові характеристики та перелік модулів кожного будуть налаштовуватися гравцем перед початком пошуку матчу. Така система забезпечує складову стратегічного мислення для гравців, вони можуть застосовувати різні підходи та креатив у формуванні своєї групи юнітів та тактиці гри безпосередньо в матчі.

Застосунок складається з наступних сцен: головне меню - надає можливість користувачу переходити до інших сцен, натискаючи відповідну

кнопку; ігрова сцена - синхронізована із ігровим сервером і на якій безпосередньо відбувається матч; меню налаштування групи юнітів (оснащення); меню статистики гравця.

Враховуючи потреби даного проєкту, основною і єдиною мережевою бібліотекою для використання була обрана Netcode for GameObjects. Головним фактором, що зумовлює цей вибір, є те, що ця бібліотека є розробкою команди Unity Technologies, що і забезпечує її високу продуктивність.

Реалізація гри

У межах даного проєкту реалізовано демонстраційну версію гри, що містить повноцінний ігровий процес та достатню кількість і різноманіття ігрових механік, реалізованих із дотриманням Server-Authoritative архітектури.

У даній роботі досліджується в першу чергу архітектурний підхід, принципи взаємодії компонентів гри. Деталі реалізації тих чи інших функціональностей здебільшого опущені. У проєкті ігровий процес було умовно поділено на механіки, що об'єднані в логічні групи: організація матчу, керування, збереження даних.

Основна увага приділена ігровій сцені, на якій проводиться матч, оскільки саме вона взаємодіє із ігровим сервером і демонструє Server-Authoritative архітектуру. Інші функції застосунку, такі як налаштування і збереження складу групи юнітів та їх модулів, а також збереження і відображення статистики (досягнень) гравця, розроблені, але не будуть реалізовані в повному обсязі. Застосунок функціонуватиме у локальній мережі.

Схема ініціалізації гравців зображена на рисунку 1.

Активацію, деактивацію та перебіг роботи модуля контролює компонент ModuleActivator. Він функціонує на серверній стороні, призначений кожному юніту, на клієнтській слугує лише для виклику ServerRPC, а також для виклику відтворення візуальних ефектів тих чи інших модулів. Запит на активацію модуля надсилає компонент Selector, коли отримує відповідний ввід від клієнта. В запиті передається id юніта-цілі та id модуля в переліку активних модулів поточного активного юніта. Сам перелік існує тільки у компоненті

UnitStatsHolder на сервері. Застосування ефекту модуля до UnitStatsHolder юніта-цілі (або цілей) відбувається на початку циклу для модулів, що діють на ціль, та в кожний такт обчислень для модулів, що діють по площі. Така різниця створена для того, щоб вплив ефекту дії по площі закінчувався одразу після виходу цілі із цієї площі. Схема активації модуля зображена на рисунку 2.

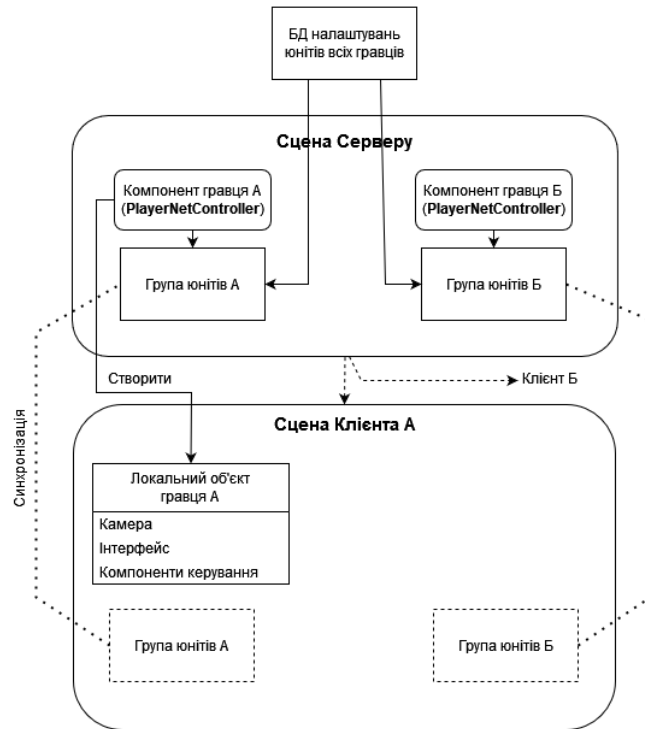


Рисунок 1 – Схема ініціалізації гравців

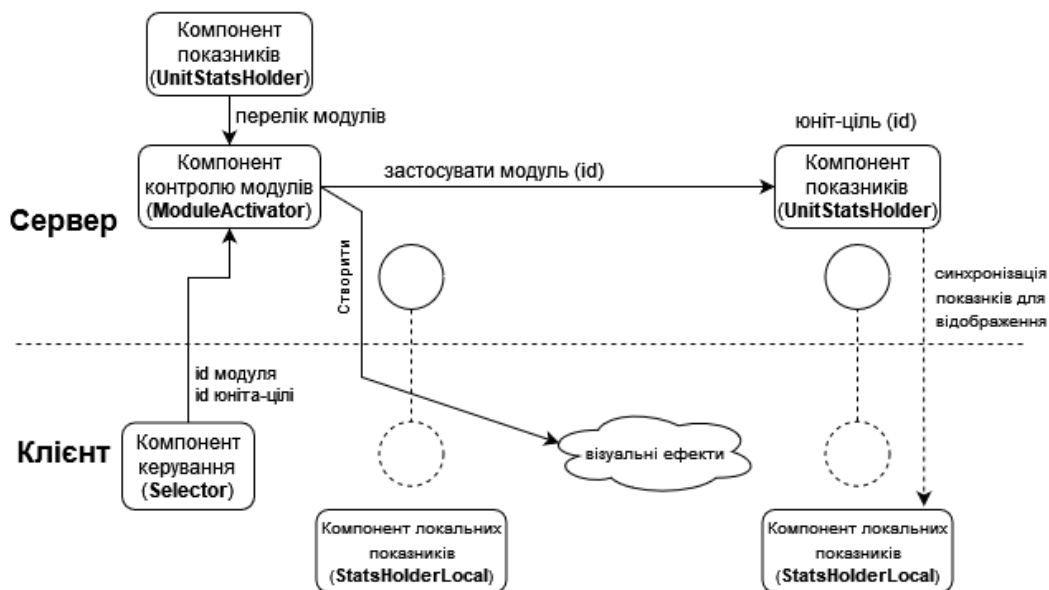


Рисунок 2 – Схема активації модуля

Збереження користувацьких налаштувань зазвичай виконується локально на комп'ютері користувача та зберігається в окремих файлах. Робота застосунку зі збереженими даними виглядатиме так, як зображено на рисунку 2.

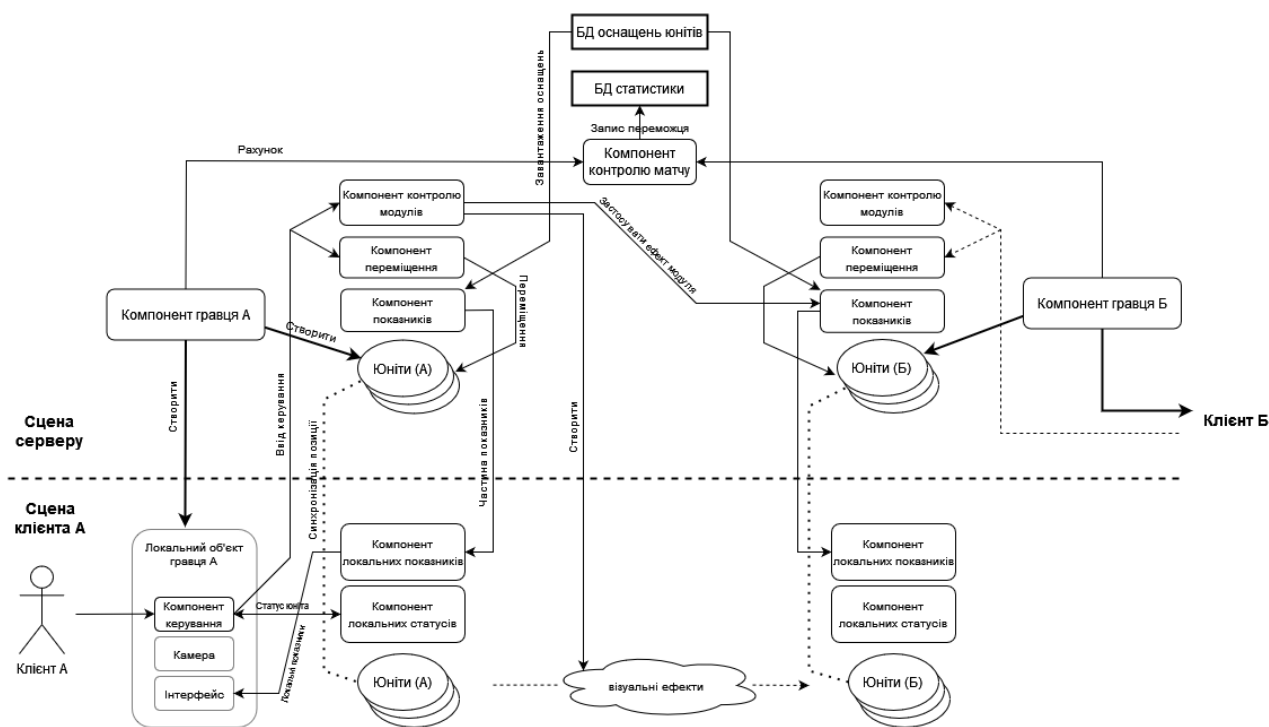


Рисунок 3 – Структура ігрової сцени

Висновки

У роботі розроблено онлайн-гру на базі архітектури з авторитетним сервером. В першу чергу, було визначено пріоритет розробки, а саме ігрова сцена, на якій проходить матч. Була виокремлена та досліджена загальна структура ігрової мережевої сесії. Визначено, що мережеві ігрові сцени (всесвіти) на сервері та клієнтах існують паралельно, синхронізуючи стани. Досліджено відповідні компоненти Netcode, що забезпечують синхронізацію сцени мережею. Основний механізм, що використовувався для написання клієнт-серверної логіки ігрового процесу – Remote Procedure Call. Обґрунтовано використання його імплементацій в Netcode (ServerRPC та ClientRPC) для тих чи інших завдань при розробці. Визначено основні ігрові механіки для реалізації в даному проєкті. Ці механіки являють собою основу застосунку, імплементують ключові елементи дизайну гри, та відповідають визначеним критеріям для проєкту.

Описано архітектуру механіки створення гравців в новій ігровій сесії, створення груп їх юнітів. Описано архітектуру механіки ініціалізації показників юнітів і їх модулів в новій ігровій сесії. Описано архітектуру підсистеми контролю перебігу матчу. Описано архітектуру підсистеми забезпечення вводу користувача для виклику відповідних дій на сервері. Описано архітектури підсистем переміщення юнітів та активації модулів кожного юніта.

Для всіх описаних механік схематично зображено взаємодію відповідних компонентів клієнта та сервера. У роботі доведено відповідність архітектури всіх описаних механік принципам авторитетності сервера та критеріям проекту, а саме: клієнтські екземпляри застосунку не містять даних, відображення яких не передбачено; клієнти не мають змоги вплинути втручатися в основні ігрові обчислення, що відбуваються на сервері.

Науковий керівник: к.т.н., доц. Галушко Д.О.

Стаття відправлена: 10.07.2023 р.

© Чикрій А.О.