

**Міністерство освіти і науки України
Одеський національний технологічний університет
Вінницький національний технічний університет
Інститут комп'ютерної інженерії, автоматизації,
робототехніки та програмування ім.П.Н.Платонова**



ПРОГРАМА

**III ВСЕУКРАЇНСЬКОЇ
НАУКОВО – ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ
МОЛОДИХ ВЧЕНИХ, АСПІРАНТІВ
ТА СТУДЕНТІВ**

**«КОМП'ЮТЕРНІ ІГРИ І МУЛЬТИМЕДІА
ЯК ІННОВАЦІЙНИЙ ПІДХІД
ДО КОМУНІКАЦІЇ - 2023»**

**28-29 вересня 2023 р.
ОДЕСА**

ПРЕЗИДІЯ ТА ОРГКОМІТЕТ КОНФЕРЕНЦІЇ

ГОЛОВА ПРЕЗИДІЇ

Єгоров Б.В., Президент ОНТУ, академік НААН України, д.т.н., професор

ЧЛЕНИ ПРЕЗИДІЇ

Іванченкова Л.В., Ректор Одеського національного технологічного університету, д.е.н., професор

Поварова Н.М., проректор з наукової роботи, к.т.н., доцент

ГОЛОВА ОРГКОМІТЕТУ

Котлик С.В., директор навчально-наукового інституту комп'ютерної інженерії, автоматизації, робототехніки та програмування ОНТУ, к.т.н., доц.

ЗАСТУПНИК ГОЛОВИ ОРГКОМІТЕТУ

Сергій Шестопапов, к.т.н., доц., каф. Комп'ютерної інженерії, ОНТУ

ЧЛЕНИ ОРГКОМІТЕТУ

Олексій Извалов, регіональний координатор Global Game Jam в Східній Європі, ETI ім.Ельворті,

Сергій Артеменко, зав.каф. Комп'ютерної інженерії, ОНТУ,

Михайло Кисленко, Unity Developer, DAL'S Games,

Олександр Романюк, зав.каф. Програмного забезпечення, ВНТУ,

Ольга Чолишкіна, директор Інституту комп'ютерно-інформаційних технологій і дизайну, МАУП,

Олександр Терьошин, Unity 3d developer, BlueGoji,

Павло Івасюк, Senior Snapchat JS Developer, BeVisioned,

Петро Горват, зав.каф. Комп'ютерних систем і мереж, ДВНЗ "Ужгородський національний університет".

УДК 004.01/08

Комп'ютерні ігри та мультимедіа як інноваційний підхід до комунікації - 2023 / Матеріали III Всеукраїнської науково-технічної конференції молодих вчених, аспірантів і студентів, Одеса, 28-29 жовтня 2023 р. - Одеса, Видавництво ОНТУ, 2023 р. – 270 с.

Збірник включає матеріали доповідей учасників конференції, які об'єднані за тематичними напрямками конференції.

Збірник буде корисним як для фахівців і працівників фірм, зайнятих в області розробки та просування комп'ютерних ігор, так і для викладачів, магістрів і студентів вищих навчальних закладів, які навчаються за напрямками і спеціальностями програмного забезпечення, комп'ютерних наук, комп'ютерної інженерії, прикладної математики та обробки інформації, буде корисним професіоналам у сферах гейміфікації, кіберспорту, стрімінгу, віртуальної реальності, доповненої реальності, штучного інтелекту, машинного навчання, геймдизайну, саунддизайну.

Результати досліджень у збірнику представляють собою своєрідний зріз сучасного стану справ в перерахованих галузях знань, який може допомогти як фахівцям, так і студентам університетів скласти загальну картину розвитку комп'ютерних ігор та мультимедіа та пов'язаних з ними питань.

Наукові праці згруповані за напрямками роботи конференції та наведені в алфавітному порядку прізвищ авторів.

Матеріали (тези доповідей) друкуються в авторській редакції. Відповідальність за якість та зміст публікацій несе автор.

Матеріали подано українською та англійською мовами.
Редактор збірника Котлик С.В.

Trends and prospects for the development of artificial intelligence and neural networks in the modern world. Kazantsev R., Zharikov T., Kim Ye.R. (Turan University, Kazakhstan)	132
Problems of evaluating and eliminating performance bottlenecks in computer games. Khoshaba O.M. (Vinnitsia National Technical University)	133
Research on the estimation of process modeling effort and cost. Andrii Kopp, Ibrahim Dag (National Technical University «Kharkiv Polytechnic Institute»)	135
Software tool for bpmn diagrams evaluation against modeling rules. Andrii Kopp, Gulden Egemen (National Technical University «Kharkiv Polytechnic Institute»)	138
Software tool for business process model comprehensibility assessment. Andrii Kopp, Vadym Sheveliev, Yagiz Ali Turgut (National Technical University «Kharkiv Polytechnic Institute»)	141
Educational school of English language. Niyazdzhanov R.R., Ismailova R.T. (Turan University)	144
Analysis of hard drive operating methods for gaming software. Oliinik M., Khoshaba O. (National Technical University, Vinnitsia, Ukraine)	147
Research application of the spam filtering and spammer detection algorithms on computer games communications. Oliinyk V., Podorozhniak A., Liubchenko N. (National Technical University “Kharkiv Polytechnic Institute”)	148
The impact of the development of embedded processor systems on gaming software. Ovod D., Khoshaba O. (National Technical University, Vinnitsia, Ukraine)	151
General methods for investigating performance bottlenecks in game software. Sychenko V., Khoshaba O. (National Technical University, Vinnitsia, Ukraine)	153
Increasing game software performance due to threads and processes in the Linux operating system. Yavorskyi D., Khoshaba O. (National Technical University, Vinnitsia, Ukraine)	154
Beam scheme development work based on arduino pro micro c using solar panel. Vladyslav Yevsieiev (Kharkiv National University of Radio Electronics)	155
Аналіз продуктивності мобільних застосунків на базі кросплатформених фреймворків. Антонова А.Р., Очеретенко Д.В. (Одеський національний технологічний університет)	158
Метод практичної побудови розпізнавача об'єктів у реальному світі. Башта А.Р., Павлова О.О. (Хмельницький національний університет, м.Хмельницький)	160
Розробка методу та програмного забезпечення модуля штучного інтелекту для гри "Монополія". Богомазов Д.В., Кательніков Д.І. (Вінницький національний технічний університет)	162
Метод організації самонавчальної комп'ютерної гри алгоритмом еволюційного навчання штучних нейронних мереж. Божик І.С., Мазурець О.В., Багрій Р.О., Кліменко В.І., Тищенко О.О. (Хмельницький національний університет)	165

GENERAL METHODS FOR INVESTIGATING PERFORMANCE BOTTLENECKS IN GAME SOFTWARE

SYCHENKO V., KHOSHABA O. (Oleksandr.Khoshaba@gmail.com)
National Technical University, Vinnitsia, Ukraine

The work describes general methods for investigating performance bottlenecks in gaming software. Particular attention is paid to the relevance and necessity of methods for identifying bottlenecks in gaming software using the example of program code optimization. This example highlights the impact of hardware in supporting modern computer games. For commonly used methods for studying gaming software performance bottlenecks, methods for profiling, optimizing program code and graphics, and using multithreading and multiprocessing in program code are described.

Relevance and necessity of performance bottleneck methods in gaming software.

Performance bottleneck analysis and optimization methods are highly relevant and necessary in gaming software development for several reasons. It is known that the quality of a game's performance directly impacts the user experience. Players expect smooth, responsive gameplay with high frame rates, low input lag, and minimal interruptions. Identifying and addressing performance bottlenecks is crucial for delivering a gaming experience.

Gaming software runs on various hardware configurations, from high-end gaming PCs to mobile devices and consoles. As a result, performance bottlenecks can vary depending on the hardware, so optimization is necessary to ensure the game performs well across different platforms.

Modern computer games become increasingly complex, with intricate graphics, physics simulations, AI, and large open worlds. This complexity can strain hardware resources, making performance bottlenecks more likely. Some game designs also require high-performance hardware to achieve their intended gameplay experiences. Optimization is necessary to meet these goals and ensure the game is playable.

Program code optimization improves performance and efficiently uses system resources such as CPU, GPU, memory, and storage. That is why efficient resource usage can extend the game's reach to lower-end hardware.

Thus, computer games may offer us various graphics settings and performance profiles to accommodate a range of hardware capabilities. Identifying and optimizing performance bottlenecks allows for better scalability across different settings.

Often used methods for research performance bottlenecks in gaming software.

There are a lot of methods for researching performance bottlenecks in gaming software. Let's look at some of them in more detail. Profiling tools help developers identify specific performance bottlenecks by analyzing where the most time or resources are spent during gameplay. Profilers like Intel VTune, NVIDIA Nsight, or built-in profiling tools in game engines help developers pinpoint precisely where performance bottlenecks occur. They provide insights into CPU and GPU usage, memory allocation, and other performance metrics.

Code optimization allows developers to optimize code to reduce computational or memory overhead. This may involve optimizing algorithms, reducing unnecessary calculations, or improving data structures. Code optimization involves making code more efficient by reducing unnecessary operations or improving algorithms. Game developers often use profiling data to identify specific parts of the code that need optimization.

Multithreading and multiprocessing can significantly boost game performance by utilizing multiple CPU cores efficiently. Leveraging multithreading and multiprocessing can distribute tasks across multiple CPU cores to improve performance, especially in rendering, physics, and AI calculations.

Graphics are a critical aspect of gaming performance. Techniques like LOD models and efficient shader code can significantly impact GPU performance. Reducing the graphical workload through techniques like occlusion culling also helps. Graphics optimization can help use graphics rendering techniques, using level-of-detail (LOD) models, and employing efficient shader code can improve GPU

performance. Also, compressing textures, reducing polygon counts in models, and optimizing asset loading can reduce memory and storage overhead.

Efficient memory management is essential for preventing performance degradation over time. Minimizing memory leaks and fragmentation ensures stable and consistent performance during gameplay. Efficient memory management practices, such as minimizing memory leaks and reducing memory fragmentation, are essential for stable performance in computer games.

Reducing input latency and maintaining a consistent frame rate is essential for a smooth gaming experience. Techniques like frame rate capping, adaptive sync, and proper input handling help achieve this. Input and frame timing allow us to reduce input latency, and ensuring consistent frame timing can result in smoother gameplay experiences.

Efficient asset streaming and loading mechanisms minimize load times and prevent stutter during gameplay. This becomes more critical as games become more open-world and data-heavy. Streaming and loading let users implement efficient streaming and loading mechanisms to reduce load times and minimize stutter during gameplay. That is why regular testing and benchmarking on various hardware configurations help ensure performance remains consistent and acceptable in computer games.

In summary, performance bottleneck analysis and optimization are critical aspects of gaming software development. They must deliver a satisfying gaming experience, meet design goals, and remain competitive in the gaming industry. Developers must continually assess and address performance issues to create successful games that run smoothly on diverse hardware.

UDK 004.451

INCREASING GAME SOFTWARE PERFORMANCE DUE TO THREADS AND PROCESSES IN THE LINUX OPERATING SYSTEM

YAVORSKYI D., KHOSHABA O. (Oleksandr.Khoshaba@gmail.com)

National Technical University, Vinnitsia, Ukraine

The work describes the increase in gaming software performance due to threads and processes in the Linux operating system. Particular attention is paid to the comparative characteristics of threads and processes. The most effective conditions for their use in gaming software are described.

Using threads and processes in the Linux operating system.

In the Linux operating system, threads and processes are fundamental for managing and executing tasks. A process is an independent and self-contained unit of execution. Each process has its memory space, file descriptors, and system resources. This isolation provides security and stability, as one process cannot directly interfere with or corrupt the memory of another. The operating system schedules processes for execution and ensures they have access to the needed resources. Multiple processes can run concurrently on multi-core CPUs, enabling true parallelism. Each process runs independently of others, and communication between processes often involves inter-process communication (IPC) mechanisms like pipes, sockets, or message queues. Threads in the same process can easily communicate and cooperate through shared memory, making it useful for tasks that require close coordination, like multithreaded server applications.

In summary, processes provide higher isolation and security but are relatively heavyweight, making them suitable for running separate, independent tasks. Threads, however, are lightweight and share resources within a process, making them suitable for concurrent tasks that must collaborate closely and efficiently. The choice between processes and threads depends on the application's specific requirements. Modern programming languages and libraries often provide thread and process management tools to simplify the development of multithreaded and multiprocessing applications.

Increased performance of gaming software due to threads and processes in the Linux operating system.

The choice between processes and threads depends on the application's specific requirements. In many cases, processes and threads are used together in a single application to take advantage of their