

**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.

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

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

strengths. Modern programming languages and libraries often provide thread and process management tools to simplify the development of multithreaded and multiprocessing applications in Linux. The increase in game software performance due to threads and processes in the Linux operating system can vary significantly depending on the nature of the game, the hardware it's running on, and how well the game is optimized for multithreading and multiprocessing. Let's consider several ways threads and processes can impact game performance.

The first way is multithreading. Parallel execution based on multithreading can improve game performance by allowing specific tasks to run concurrently on multi-core CPUs. For example, game engines can use separate threads for rendering, physics simulations, and AI calculations, utilizing all available CPU cores. Responsiveness based on multithreading can help maintain a smooth and responsive user interface (UI) even when computationally intensive tasks are running in the background. This is important for preventing frame rate drops or input lag. Faster load times based on background loading of assets (textures, models, etc.) in a separate thread can reduce load times and provide a seamless gaming experience.

The second way is multiprocessing (multiple processes). Isolation, used for running game logic and other components in separate processes, can enhance stability and security. If one component crashes, it's less likely to affect the entire game. Resource management, used for multiprocessing, can help manage resource-heavy tasks that are best isolated from the primary game process, such as video encoding or complex physics simulations. This can prevent these tasks from adversely affecting game performance. Parallelism, used for specific game servers and backend processes, especially in online multiplayer games, can benefit from multiprocessing to handle concurrent connections or game instances. However, it's important to note that adding more threads or processes does not always guarantee a linear increase in performance. That is why there are challenges to consider. These problems and considerations are as follows. Synchronization overhead is that synchronization mechanisms like mutexes or semaphores may be required when multiple threads or processes need to access shared data. If not managed carefully, these can introduce overhead and potential bottlenecks. Complexity appears when multithreaded and multiprocess programming can be complex, and managing concurrency issues, such as race conditions and deadlocks, can be challenging.

For this, Amdahl's Law states that the speedup from parallelization is limited by the fraction of the program that cannot be parallelized. That is why some game tasks are inherently sequential, meaning they can't be parallelized effectively. Resource constraints are that adding more threads or processes consumes additional memory and CPU resources. Not managed well can lead to resource contention and reduced performance.

Overall, the performance gains from threads and processes in Linux for game software depend on careful optimization, hardware capabilities, and the specific demands of the game. That is why game developers often employ profiling and benchmarking tools to identify performance bottlenecks and make informed decisions about where to implement multithreading or multiprocessing to achieve the best balance between performance and resource utilization.

UDC 681.51

## **BEAM SCHEME DEVELOPMENT WORK BASED ON ARDUINO PRO MICRO C USING SOLAR PANEL**

VLADYSLAV.YEVSIIEV (vladyslav.yevsieiev@nure.ua),  
Kharkiv National University of Radio Electronics

*This work is devoted to the coverage of BEAM robotics development issues. The author provides a comparative analysis between the BEAM robot and a classic mobile robot according to various criteria and has developed a BEAM robot scheme based on Arduino Pro Micro using a Solar Panel.*

BEAM robotics (Biology, Electronics, Aesthetics, Mechanics) is an approach to creating minimalist robots that combines a biologically inspired approach with electronics, mechanics and aesthetics [1].