

# **ЕЛЕКТРОННІ ІНФОРМАЦІЙНІ РЕСУРСИ: СТВОРЕННЯ, ВИКОРИСТАННЯ, ДОСТУП**

**ЗБІРНИК МАТЕРІАЛІВ**

**Міжнародної науково-практичної Інтернет-конференції**

**20-21 листопада 2023 р.**

**Міністерство освіти і науки України**  
**Вінницький національний технічний університет**  
**Національна академія Державної прикордонної служби України**  
**ім. Богдана Хмельницького**  
**Вінницький національний медичний університет ім. М.І. Пирогова**  
**КЗВО «Вінницька академія безперервної освіти»**  
**КЗ «Сумський обласний інститут післядипломної педагогічної освіти»**  
**Інститут комп'ютерних систем і технологій "Індустрія 4.0"**  
**ім. П. Н. Платонова**  
**Люблінська політехніка (Польща)**  
**Університет Бельсько-Бяльський (Польща)**

**«ЕЛЕКТРОННІ ІНФОРМАЦІЙНІ  
РЕСУРСИ: СТВОРЕННЯ, ВИКОРИСТАННЯ,  
ДОСТУП»**

**ЗБІРНИК МАТЕРІАЛІВ**

**Міжнародної науково-практичної Інтернет-конференції**  
**20-21 листопада 2023 р.**

**Суми/Вінниця**  
**НІКО/КЗВО «Вінницька академія безперервної освіти»**  
**2023**

**УДК 004**  
**ББК 32.97**  
**Е50**

Рекомендовано до видання Вченою радою КЗВО «Вінницька академія безперервної освіти» (протокол № 8 від 20.11.2023 р.)

**Електронні інформаційні ресурси: створення, використання, доступ.**  
Збірник матеріалів Міжнародної науково-практичної Інтернет конференції 20-21 листопада 2023 р. – Суми/Вінниця: НІКО/КЗВО «Вінницька академія безперервної освіти», 2023. – 336 с.

**ISBN 978-617-7422-23-4**

Збірник містить матеріали Міжнародної науково-практичної Інтернет конференції «Електронні інформаційні ресурси: створення, використання, доступ. Матеріали збірника подано у авторській редакції. Автори опублікованих матеріалів несуть повну відповідальність за підбір, точність наведених фактів, цитат, статистичних даних, власних імен та інших відомостей, Матеріали відтворюються зі збереженням змісту, орфографії та синтаксису текстів, наданих авторами.

**УДК 004**  
**ISBN 978-617-7422-23-4**

**© КЗВО «Вінницька академія безперервної освіти», 2023**  
**© Вид-во Суми, НІКО, 2023**

Прус О.В., Майданюк В.П.,	WEBASSEMBLY: інтеграція та інновації у побудові графіків та інтерактивних веб-інтерфейсів	212
Рейда М.О., Черній А.О., Романюк О.Н., Рейда О.М.	Аналіз DIRECTX	217
Рейда О.М., Коваленко О.О., Антипенко Я.Д.	Програмні продукти підтримки педагогічних квестів	220
Рижавська Т.М.	Електронні інформаційні ресурси. Google для освіти	221
Рижий Я.О., Мельник М.М., Стецюк В.М.	Технологія цифрового підпису з використанням атрибутів в системах електронного документообігу	223
Рижков А. К., Войцеховська О.В., Городецька О. С.	Аналіз методів авторизації при проектуванні серверної частини веб-застосунку	225
Романюк О. Н., Станіславенко Є. Г., Мельник А. В., Романюк С. О.	Використання програмного пакета SUBSTANCE PAINTER для розробки 3Д моделей	227
Романюк О. Н., Корягіна Д. О.,	Аналіз сучасних програмних продуктів для розробки web-сайтів	230
Романюк О.Н., Бойко О.П., Мельник А.В., Чехмestрук Р.Ю.	Елементи штучного інтелекту в програмі ADOBE PHOTOSHOP	232
Романюк О.Н., Мазур В.В., Глоба А. Р., Снігур А.В.	Аналіз вбудованих графічних процесорів	233
Салабай Б.С.	Forecasting sales using exponential smoothing methods	235
Саланчій Т.О.	Дослідження та порівняння методів класифікації рослинних хвороб на розмитих зображеннях для підвищення ефективності сільського господарства та біологічних досліджень	239
Самарасінгхе Д.С.В., Рейда О.М.	Дослідження методів оптимізації ігрових рушіїв ACTION ігор мобільних додатків	243
Сафо В.В.	Мікросервісна архітектура для системи управління обігом антикваріату	246
Сентюрін Є.Є., Кочнев Є.А., Антонюк В.В., Ліщинський А.С., Бабюк Н.П.	Аналіз додатків-порадників для молоді та напрямки їх удосконалення	249

доступу до особистих даних на сервері. Токени повинні бути збережені безпечно та шифровані. Крім того, слід використовувати методи автентифікації з використанням токенів, такі як OAuth, щоб забезпечити безпеку та прозорість взаємодії з ресурсами [10].

Для забезпечення безпеки даних також важливо правильно реалізувати процес автентифікації користувачів. Необхідно використовувати сучасні методи автентифікації, такі як двофакторна автентифікація, входження за допомогою відбитків пальців або розпізнавання обличчя, щоб запобігти несанкціонованому доступу.

Загальна безпека даних у мобільних додатках - це процес, який постійно розвивається та оновлюється. Використовуючи вищезазначені практики та ресурси, ви зможете підтримувати найвищий стандарт безпеки в ваших Flutter-додатках і надавати користувачам впевненість у збереженні їхніх даних.

### **Висновки**

Шифрування та безпека збереження даних в Flutter-додатках є критично важливими аспектами розробки. Захист даних користувачів повинен бути високого рівня, і розробники повинні вкласти зусилля в забезпечення безпеки локальних і переданих даних. Використовуючи належні методи шифрування, безпечні сховища для ключів і токенів, а також правильні методи автентифікації, ви можете створити надійний та безпечний Flutter-додаток, якому користувачі можуть довіряти.

### **Список використаних джерел**

1. Dominic Chell, Tyrone Erasmus, Shaun Colley, Ollie Whitehouse. (2015). The Mobile Application Hacker's, 18.
2. Flutter [Електронний ресурс] – режим доступу: <https://flutter.dev> Дата звернення: 28 жовтня 2023.
3. Timothy Speed, Darla Nykamp, Mari Heiser, Joseph Anderson. (2013). Mobile Security: How to Secure, Privatize, and Recover Your Devices, 20-23, 56-58.
4. Himanshu Dwivedi, Chris Clark, (2012) Mobile Application Security, 16, 40, 68.
5. Encrypt [Електронний ресурс] – Режим доступу: <https://pub.dev/packages/encrypt> Дата звернення: 28 жовтня 2023.
6. Pointycastle [Електронний ресурс] – Режим доступу: <https://pub.dev/packages/pointycastle> Дата звернення: 28 жовтня 2023.
7. Introduction to HTTP and HTTPS [Електронний ресурс] – Режим доступу: <https://learn.microsoft.com/en-us/azure/rtos/netx-duo/netx-duo-web-http/chapter1> Дата звернення: 28 жовтня 2023.
8. Http [Електронний ресурс] – Режим доступу: <https://pub.dev/packages/http> Дата звернення: 28 жовтня 2023.
9. Understanding OAuth2 Landscape [Електронний ресурс] – Режим доступу: <https://sagarag.medium.com/understanding-oauth2-landscape-1b80cc9ed303> Дата звернення: 28 жовтня 2023.
10. Aaron Parecki (2012), OAuth 2.0: The Definitive Guide 156-160.

*ПРУС О.В., МАЙДАНЮК В.П.,  
Вінницький національний технічний університет*

## **WEBASSEMBLY: ІНТЕГРАЦІЯ ТА ІННОВАЦІЇ У ПОБУДОВІ ГРАФІКІВ ТА ІНТЕРАКТИВНИХ ВЕБ-ІНТЕРФЕЙСІВ**

*Анотація: Стаття розглядає роль та важливість WebAssembly (Wasm) у побудові графіків та інтерактивних інтерфейсів на веб-сторінках. WebAssembly надає можливості оптимізації та виконання високоякісного коду на веб-платформі, що робить його потужним інструментом для розробки веб-додатків. Стаття аналізує основні аспекти використання WebAssembly у графіці та інтерактивних інтерфейсах, а також надає огляд фреймворків і бібліотек для роботи з Wasm. Висвітлюються найкращі практики та інноваційні рішення для покращення веб-графіки та інтерактивності.*

*Ключові слова: WebAssembly, Wasm, графіка, інтерактивність, веб-інтерфейси, фреймворки, бібліотеки, оптимізація, інновації, веб-розробка, візуалізація, віртуальні інтерфейси, графічний дизайн.*

*Abstract: The article explores the role and significance of WebAssembly (Wasm) in building graphics and interactive interfaces on web pages. WebAssembly provides optimization capabilities and the execution of high-quality code on the web platform, making it a powerful tool for web application development. The article analyzes the key aspects of using WebAssembly in graphics and interactive interfaces, as well as provides an overview of frameworks and libraries for working with Wasm. Best practices and innovative solutions for enhancing web graphics and interactivity are highlighted.*

*Keywords: WebAssembly, Wasm, graphics, interactivity, web interfaces, frameworks, libraries, optimization, innovations, web development, visualization, virtual interfaces, graphic design.*

## **Вступ**

З розвитком сучасних веб-додатків, вимоги до якості графічних інтерфейсів та інтерактивності зростають з кожним днем. Веб-середовище стає все більш вимогливим до продуктивності та швидкодії, а користувачі очікують на відмінну якість візуального відображення та плавну роботу інтерфейсів. Однак для команд, що розробляють веб-додатки в багатопроєктних середовищах, це завдання може бути надзвичайно складним і вимагає ефективних рішень [1, 2].

WebAssembly (або коротко Wasm) - це відкритий стандарт, який представляє собою низькорівневу віртуальну машину, призначену для виконання веб-додатків у браузері. Завдяки WebAssembly розробники мають можливість виконувати високопродуктивний код, спеціально оптимізований для веб-середовища, що відкриває нові можливості для створення веб-додатків з високоякісною графікою та інтерактивністю [3].

**Актуальність** використання WebAssembly (Wasm) у побудові графіків та інтерактивних інтерфейсів полягає в тому, що сучасні веб-додатки стають все складнішими та вимогливішими щодо графічного вмісту та інтерактивності. Від користувачів очікується швидкодія та висока якість візуального сприйняття, що робить важливим завданням оптимізацію відтворення графіки та взаємодії на веб-сторінках [4].

**Метою** цієї статті є подальший розвиток дослідження можливостей та переваг використання WebAssembly у веб-графіці та інтерактивних інтерфейсах. Ми дослідимо інноваційні методи та рішення, які дозволяють покращити продуктивність та якість графічних інтерфейсів веб-додатків.

## **Аналіз досліджень та постановка завдання**

На сьогоднішній день існують численні дослідження та публікації, що стосуються використання WebAssembly у веб-розробці загалом. Вони демонструють, що Wasm може бути потужним інструментом для оптимізації веб-додатків у відтворенні графіки та інтерактивності. Однак варто відзначити, що існують невирішені проблеми та нез'ясовані аспекти використання Wasm у цій контексті, а також багато можливостей для подальших досліджень [5].

Аналіз попередніх досліджень та публікацій у галузі використання WebAssembly вказує на те, що цей інструмент вже знайшов застосування в різних аспектах веб-розробки. Деякі з найбільш важливих робіт та публікацій, які стосуються даної проблеми, включають:

- John Royle займався роботою присвяченою основам WebAssembly та дослідженню його можливостей та переваг у контексті веб-розробки. Стаття визначає, що WebAssembly є новим виконавчим середовищем, яке може використовуватися для побудови веб-додатків з високоякісною графікою та швидкою продуктивністю [6]. Проте робота не докладно розглядає конкретні приклади використання цього інструменту у веб-графіці.

- Lin Clark у своєму дослідженні звертає увагу на WebAssembly Interface Types, які розширюють можливості взаємодії між WebAssembly та звичайним JavaScript кодом. Він вказує на важливість покращення взаємодії між цими двома середовищами для забезпечення більшої зручності та ефективності веб-розробки [7]. Проте ця робота не вирішує всі аспекти, пов'язані із використанням WebAssembly у веб-графіці та інтерфейсах.

- Senthil Padmanabhan та Pranav Jha приводять приклад реального використання WebAssembly на платформі eBay. Їхній досвід показує, що WebAssembly може бути успішно використаним у великих веб-додатках для покращення продуктивності та користувацького досвіду [8]. Однак ця робота також не розглядає всі аспекти та потенціал WebAssembly у

побудові графіків та інтерактивних інтерфейсів.

Як показує аналіз, хоча й існує певний обсяг досліджень, присвячених використанню WebAssembly у веб-графіці та інтерфейсах, однак деякі питання залишаються не вирішеними:

- Оптимізація графічного відтворення: Як забезпечити оптимальне відтворення графіки на веб-сторінках за допомогою WebAssembly для досягнення максимальної продуктивності та візуальної якості?

- Інтерактивність та реактивність: Як підвищити інтерактивність веб-інтерфейсів за допомогою WebAssembly, забезпечуючи швидку взаємодію з користувачем?

- Сумісність та інтеграція: Як забезпечити сумісність WebAssembly з існуючим JavaScript кодом та іншими веб-технологіями?

Враховуючи аналіз проблеми та попередніх досліджень, основними завданнями цієї статті є:

- Розглянути можливості модульності та перевикористання коду з використанням WebAssembly у контексті інтерактивних веб-інтерфейсів.

- Провести аналіз особливостей використання бібліотек та фреймворків для створення графіків та інтерактивних інтерфейсів з використанням WebAssembly.

- Представити огляд кращих практик та інноваційних рішень у використанні WebAssembly для покращення веб-графіки та інтерактивності.

Ці завдання спрямовані на розширення розуміння можливостей WebAssembly та виявлення шляхів оптимізації веб-графіки та інтерактивних інтерфейсів для подальшого розвитку веб-розробки.

### **Виклад основного матеріалу**

Можливості модульності та перевикористання коду з використанням WebAssembly у контексті інтерактивних веб-інтерфейсів дозволяють розробникам здобути ряд конкурентних переваг, включаючи швидкість розробки, високу продуктивність та гнучкість систем. Ключові аспекти WebAssembly включають:

#### 1. Модульна структура:

- Незалежність компонентів: WASM-модулі можна розробляти як незалежні блоки, що спрощує їх інтеграцію та тестування. Ця незалежність дозволяє командам розробників працювати паралельно над різними частинами проекту [9].

- Легкість обслуговування: Завдяки модульності, внесення змін або вдосконалень до конкретного модуля не вимагає перегляду всього застосунку, що пришвидшує процес обслуговування та виправлення помилок.

#### 2. Повторне використання коду:

- Зменшення розробки: Використання вже існуючих бібліотек або модулів, скомпільованих у WASM, зменшує час та ресурси, потрібні для розробки нового застосунку [10].

- Стандартизація рішень: Повторне використання коду сприяє створенню стандартних рішень для типових завдань, що забезпечує їх більш стабільну роботу та сумісність [9].

#### 3. Інтеграція з JavaScript:

- Комбіновані можливості: WASM працює в гармонії з JavaScript, дозволяючи розробникам використовувати переваги обох технологій: продуктивність WASM та гнучкість JavaScript.

- Взаємодія із DOM: Інтеграція WASM і JavaScript спрощує взаємодію з DOM та іншими браузерними API, надаючи можливість створювати більш реактивні та інтерактивні інтерфейси [11].

#### 4. Безпека:

- Обмежений доступ: WASM-модулі виконуються в ізольованому середовищі, яке обмежує доступ до основних системних ресурсів, забезпечуючи таким чином високий рівень безпеки.

- Стабільність виконання: Ізольоване середовище також забезпечує стабільність виконання коду, зменшуючи ймовірність впливу зовнішніх чинників на роботу програми [12].

#### 5. Висока продуктивність:

- Нативний підхід: WASM забезпечує виконання коду на рівні, близькому до нативного, що гарантує високу продуктивність навіть для вимогливих застосунків.

- Оптимізація ресурсів: Завдяки ефективній роботі з пам'яттю та ресурсами, застосунки на WASM споживають менше ресурсів, що підвищує їх продуктивність та відгук на дії користувача [13].

Дедалі більше веб-розробників вибирають WebAssembly як ключовий інструмент для створення сучасних, високопродуктивних і гнучких веб-інтерфейсів. Його модульність, можливість перевикористання коду та інтеграція з існуючими технологіями роблять його незамінним у сучасному світі веб-розробки.

WebAssembly надає можливість використовувати багато різних бібліотек і фреймворків для створення графічних та інтерактивних елементів веб-додатків. Аналіз цього аспекту включає наступні особливості:

- Відкритість і сумісність: WebAssembly підтримує стандарти відкритої веб-розробки, що дозволяє використовувати велику кількість бібліотек і фреймворків, які були розроблені для JavaScript. Це важливо для розробників, оскільки вони можуть перевикористовувати існуючий код та інструменти [14].

- Швидкість і продуктивність: Використання WebAssembly дозволяє підвищити продуктивність графічних операцій та інтерактивних операцій, оскільки він компілюється в машинний код та виконується набагато швидше, ніж інтерпретований JavaScript. Це особливо корисно для важкозавантажених графічних додатків та ігор [15].

- Графічні бібліотеки: WebAssembly підтримує різні графічні бібліотеки, такі як WebGL, OpenGL, та інші. Це дозволяє розробникам створювати складну графіку та візуалізації у веб-додатках. Також, існують спеціалізовані бібліотеки для відображення 2D та 3D графіки [16, 17].

- Фреймворки для інтерактивності: Існують фреймворки, які допомагають створювати взаємодію користувача з графічними компонентами. Такі фреймворки надають можливість створювати анімації, обробляти події та реагувати на дії користувача у веб-додатках (Emscripten, AssemblyScript, Blazor WebAssembly, Three.js та інші) [15].

- Розширені можливості бібліотек: Деякі бібліотеки та фреймворки мають розширені можливості для роботи з графікою та взаємодією. Наприклад, деякі дозволяють створювати веб-VR та веб-AR додатки, розробляти ігри, а також робити візуалізації для великих обсягів даних [16, 17].

- Підтримка розробки на різних мовах програмування: WebAssembly підтримує різні мови програмування, що дозволяє розробникам вибирати мову, яка найкраще підходить для конкретної задачі. Наприклад, для обчислень можна використовувати мови, які спеціалізуються на чисельних операціях.

Загалом, використання WebAssembly разом з бібліотеками та фреймворками дозволяє створювати більш продуктивні та візуально привабливі веб-додатки з високоякісною графікою та інтерактивністю.

WebAssembly відкриває безліч можливостей для покращення веб-графіки та інтерактивності. Нижче представлено огляд кращих практик та інноваційних рішень у використанні WebAssembly для створення високоякісних веб-графічних інтерфейсів:

- Використання WebGL та OpenGL: WebGL та OpenGL є потужними графічними бібліотеками, які підтримуються WebAssembly. Вони надають можливість реалізовувати 2D та 3D графіку з високою швидкістю та якістю [16, 17].

- Використання готових фреймворків: Деякі фреймворки, такі як Three.js для 3D графіки, PixiJS для 2D графіки, та Phaser для створення ігор, дозволяють розробникам швидко створювати візуально привабливі інтерактивні інтерфейси без необхідності писати весь код з нуля [15].



- Використання графічних шейдерів: WebAssembly дозволяє розробникам використовувати графічні шейдери, що забезпечують ефектну обробку графіки та анімації. Це особливо корисно для створення візуально складних інтерфейсів [16].

- Оптимізація використання ресурсів: Використання WebAssembly для обчислень та графічних операцій може бути ресурсомістким. Кращі практики включають управління пам'яттю, уникання зайвого копіювання даних та мінімізацію звернень до браузерних API [12].

- Відзначення потенціалу WebAssembly Interface Types: Веб-стандарт WebAssembly Interface Types відкриває нові можливості для взаємодії з JavaScript. Це дозволяє розробникам ефективно комунікувати між WebAssembly та JavaScript кодом, що полегшує роботу з графічним інтерфейсом [18].

- Оптимізація завантаження та завантаження по запиту: Для оптимізації швидкості завантаження веб-графіки та інтерактивних ресурсів рекомендується завантажувати лише необхідні дані при необхідності і використовувати кешування [15].

- Спільне використання бібліотек та компонентів: Розробники можуть спільно використовувати бібліотеки та компоненти між проектами для підвищення ефективності розробки та забезпечення єдності стилю та функціональності.

- Моніторинг та аналіз продуктивності: Важливим аспектом є моніторинг продуктивності веб-графіки та інтерактивних інтерфейсів, щоб виявляти та виправляти можливі помилки та збої [19].

Ці кращі практики та інноваційні рішення допомагають розробникам створювати веб-інтерфейси, які поєднують високоякісну графіку та інтерактивність, що відповідає сучасним вимогам користувачів. Розвиток WebAssembly продовжує відкривати нові можливості для покращення веб-графіки та інтерактивності в майбутньому.

### **Висновки**

У цій статті ми розглянули застосування WebAssembly в контексті веб-графіки та інтерактивних інтерфейсів. Було представлено основні можливості цієї технології та огляд кращих практик та інноваційних рішень, які дозволяють розробникам створювати високоякісні веб-інтерфейси.

Ми проаналізували основні дослідження та публікації, пов'язані із застосуванням WebAssembly у веб-графіці та інтерактивних інтерфейсах, та виділили невирішені проблеми та можливості для подальших досліджень.

Важливим аспектом є використання графічних бібліотек, таких як WebGL та OpenGL, а також фреймворків, що спрощують створення графіки. Графічні шейдери дозволяють досягати вражаючих ефектів та анімацій.

Оптимізація ресурсів, використання WebAssembly Interface Types та спільне використання бібліотек та компонентів є важливими частинами розробки веб-графічних інтерфейсів.

За допомогою WebAssembly розробники мають змогу створювати захоплюючі та ефективні веб-графічні інтерфейси, які задовольняють потреби сучасних користувачів. Розвиток цієї технології продовжує відкривати нові можливості та горизонти для веб-розробки.

### **Список використаних джерел**

1. Zachary, P., & Nguyen, T. (2017). An Introduction to WebAssembly: Design, Concepts, and Implementations. *Journal of Web Development*, 3-4, 33-47.
2. Eich, B. (2019). *WebAssembly: The Future of JS and a Multi-Language Web*. O'Reilly Media, 211-223.
3. Haas, A., & Rossberg, A. (2017). Bringing the web up to speed with WebAssembly. *Proceedings of the 38th ACM SIGPLAN Conference on Programming Language Design and Implementation*, 185-200.
4. W3C. (2021). *WebAssembly Core Specification*. [Електронний ресурс] – режим доступу: <https://www.w3.org/TR/wasm-core-1/>. Дата звернення: 24 жов. 2023.

5. Parker, R. (2020). *Optimizing WebAssembly: Techniques and Practices for Web Developers*. Addison-Wesley Professional, 322-329.
6. John Royle. (2023) *WebAssembly: A New Era of High-Performance, Portable, and Secure Web Applications* [Електронний ресурс] – режим доступу: <https://www.linkedin.com/pulse/webassembly-new-era-high-performance-portable-secure-web-john-royle/> Дата звернення: 24 жов. 2023.
7. Lin Clark. (2019) *WebAssembly Interface Types: Interoperate with All the Things!* [Електронний ресурс] – режим доступу: <https://hacks.mozilla.org/2019/08/webassembly-interface-types/> Дата звернення: 24 жов. 2023.
8. Senthil Padmanabhan, Pranav Jha. (2019) *WebAssembly at eBay: A Real-World Use Case* [Електронний ресурс] – режим доступу: <https://innovation.ebayinc.com/tech/engineering/webassembly-at-ebay-a-real-world-use-case/> Дата звернення: 24 жов. 2023.
9. Green, J., & Brown, L. (2019). *Graphics Rendering: Techniques for Modern Web*. Springer. 110-126.
10. Miller, K. (2018). *Interactive and Reactive Web: A Deep Dive*. John Wiley & Sons. 54-70.
11. MDN WEB Docs. (2023) *WebAssembly* [Електронний ресурс] – режим доступу: <https://developer.mozilla.org/en-US/docs/WebAssembly/> Дата звернення: 24 жов. 2023.
12. *Webassembly*. [Електронний ресурс] – режим доступу: <https://webassembly.org/> Дата звернення: 24 жов. 2023.
13. *Майбутнє веб-розробки з WebAssembly* [Електронний ресурс] – режим доступу: <https://devzone.org.ua/post/maibutnje-veb-rozrobki-z-webassembly/> Дата звернення: 24 жов. 2023.
14. *Awesome Open Source. Awesome WebAssembly Languages*. [Електронний ресурс] – режим доступу: <https://awesomeopensource.com/project/appcypher/awesome-wasm-langs> Дата звернення: 24 жов. 2023.
15. Rossberg, A. (2018). *WebAssembly: Basics & Beyond*. Google Developers Blog. [Електронний ресурс] – режим доступу: <https://developers.google.com/web/updates/2018/03/emscripting-a-c-library/> Дата звернення: 24 жов. 2023.
16. *WebGL Fundamentals*. [Електронний ресурс] – режим доступу: <https://webglfundamentals.org/>. Дата звернення: 24 жов. 2023.
17. *WebGPU API* [Електронний ресурс] – режим доступу: [https://developer.mozilla.org/en-US/docs/Web/API/WebGPU\\_API/](https://developer.mozilla.org/en-US/docs/Web/API/WebGPU_API/) Дата звернення: 24 жов. 2023.
18. *WebAssembly Interface Types*. (n.d.). [Електронний ресурс] – режим доступу: <https://github.com/WebAssembly/interface-types/> Дата звернення: 24 жов. 2023.
19. *How do you develop web solutions that are innovative and compatible with different audiences?* [Електронний ресурс] – режим доступу: <https://www.linkedin.com/advice/0/how-do-you-develop-web-solutions-innovative-compatible/> Дата звернення: 24 жов. 2023.

*РЕЙДА М. О., ЧЕРНІЙ А. О, РОМАНЮК О. Н., РЕЙДА О. М.  
Вінницький національний технічний університет;*

### **АНАЛІЗ DIRECTX**

DirectX [1-6] — це набір API функцій, розроблених для ефективного вирішення завдань відображення та обробки графічних даних у процесі розробки програм для операційної системи Microsoft Windows. В основному застосовується для створення відеоігор. Пакет засобів розробки DirectX від Microsoft є безкоштовним на сайті Microsoft. Він включений в стандартний набір програмного забезпечення Windows. Актуальні версії DirectX часто включенні до файлів, які його потребують. Це пов'язано з регулярним оновленням API для найефективнішої обробки даних.

Основні переваги роботи DirectX:

**ЕЛЕКТРОННІ ІНФОРМАЦІЙНІ РЕСУРСИ:  
СТВОРЕННЯ, ВИКОРИСТАННЯ, ДОСТУП:**

Збірник матеріалів  
Міжнародної науково-практичної Інтернет-конференції  
20-21 листопада 2023 р.

Редактор С.А.Пойда, М.С. Ніколаєнко  
Комп'ютерне верстання С.А.Пойда, М.С. Ніколаєнко

Підписано до друку 15.11.2023 Гарнітура Times New Roman  
Формат 60x84/16 Папір офсетний  
Друк цифровий Ум. друк. арк. 19,4  
Тираж 300 пр. Зам. № 2/23

Видавництво НІКО  
м.Суми, вул.Харківська, 54  
Свідоцтво про внесення до Державного реєстру  
суб'єктів видавничої справи України  
серія СМв № 044  
від 15.10.2012  
E-mail: ms.niko@i.ua  
Телефон для замовлень: +38(066) 270-64-68