

КРИЛИК ЛЮДМИЛА

Вінницький національний технічний університет

<https://orcid.org/0000-0001-6642-754X>e-mail: lyudmila.krylik@gmail.com**ЕРМОЛАЄВА МІЛЕНА**

Вінницький національний технічний університет

e-mail: mlen.cr.003@gmail.com

ОСОБЛИВОСТІ РОЗРОБКИ АВТОРСЬКОЇ 2D ГРИ З ВИКОРИСТАННЯМ ІГРОВОГО РУШІЯ UNREAL ENGINE 5

У роботі подано особливості розробки авторської 2D гри з використанням Unreal Engine 5. Метою роботи є покращення ігрового процесу 2D гри за допомогою розширення функціональних можливостей ігрового рушія Unreal Engine 5. Unreal Engine 5 – це сучасний ігровий рушій для розробки ігор, який має інструменти для створення захоплюючих просторових світів, анімацій та ігрових механік. Цей рушій обмежений у розробці 2D ігор, але існують додаткові інструменти для полегшення та покращення роботи з розробкою. Вбудований плагін Paper2D та безкоштовний плагін PaperZD значно полегшили та прискорили розробку 2D гри, а саме, налаштування різнонаправлених анімацій, створення state machine, створення AnimationInstance, AnimationSource Blueprints. Крім того, Unreal Engine потребував певних налаштувань, які потрібні були не тільки для покращення якості графіки, а також для уникнення графічних артефактів та покращення роботи гри. Під час розробки було виявлено проблему з налаштуванням колізій для взаємодії та якорем персонажа. Як результат дослідження було знайдено рішення цієї проблеми та впроваджено у 2D гру. Враховуючи всі особливості роботи з 2D іграми, у ігровому рушії Unreal Engine 5 можна розробити гру з покращеним ігровим процесом, якісною графікою за менший період часу. Порівнюючи розроблену авторську 2D гру з іншим 2D проектом, розробленим у Unreal Engine 5 без використання описаних технік та плагіну PaperZD, встановлено, що використання плагіну PaperZD та використання файлів з розширенням json значно прискорили та покращили розробку 2D гри. Змінення налаштувань проекту та використання модифікованого PostProcessing Volume значно покращили графіку та продуктивність гри. Розроблена авторська 2D гра успішно пройшла тестування та має покращений ігровий процес, якісну графіку та належну продуктивність.

Ключові слова: 2D ігри, Unreal Engine 5, gamedev, інди-розробка.

KRYLIK LYUDMILA**YERMOLAIEVA MILENA**

Vinnytsia National Technical University

FEATURES OF DEVELOPING A 2D AUTHORIZED GAME USING THE UNREAL ENGINE 5

The work presents the features of developing an author's 2D game using Unreal Engine 5. The method of work is to improve the gameplay of a 2D game by expanding the functionality of the Unreal Engine 5 game engine. Unreal Engine 5 is a modern game engine for game development, which has tools for creating exciting spatial worlds, animations and game mechanisms. This engine is limited in the development of 2D games, but other additional tools to facilitate and improve the development work. The built-in Paper2D plugin and the free PaperZD plugin have greatly facilitated and accelerated the development of a 2D game, as well as setting up multi-directional animations, creating a state machine, creating AnimationInstance, AnimationSource Blueprints. In addition, Unreal Engine required certain settings that are needed not only to improve the quality of graphics, but also to avoid graphic artifacts and improve the game. During development, a problem was discovered with the settings of collisions for interaction and the character anchor. The research found a solution to this problem and implemented it in a 2D game. Taking into account all the features of working with 2D games, in the Unreal Engine 5 game engine, it is possible to develop a game with improved gameplay, high-quality graphics in a shorter period of time. Comparing the development of the author's 2D game with another 2D project developed in Unreal Engine 5 without using the described technique and the PaperZD plugin, it was found that using the PaperZD plugin and using files with the json extension significantly accelerated and improved the development of the 2D game. Changing the project settings and using a modified PostProcessing Volume significantly improved the graphics and performance of the game. The developed author's 2D game successfully passed testing and has improved gameplay, high-quality graphics and proper performance.

Keywords: 2D games, Unreal Engine 5, gamedev, indie development.

Стаття надійшла до редакції / Received 11.04.2025

Прийнята до друку / Accepted 26.04.2025

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

У сучасному світі ігрова індустрія постійно розвивається, технології для їх розробки вдосконалюються та оновлюються. Нині це одна із провідних галузей, яка стимулює розвиток комп'ютерних технологій. Для розробки 2D ігор існують різні ігрові рушії, а саме Unity, Unreal Engine 5, Godot, GameMaker та багато інших. Донедавна існував стереотип, що за допомогою Unreal Engine 5 неможливо зробити якісну 2D гру, але протягом останніх років з'явилися нові інструменти та техніки, що полегшують розробку 2D ігор.

Unreal Engine 5 – це один з провідних та найсучасніших ігрових рушіїв, застосування якого можливе не тільки для розробки гіпер-реалістичних AAA 3D ігор, а також для розробки 2D ігор, проте є деякі особливості, з якими можна зіштовхнутись під час розробки.

Аналіз досліджень та публікацій

На сьогодні існує велика кількість інструментів та технік для розробки 3D ігор на ігровому рушії Unreal Engine 5, але їх не так багато для розробки 2D ігор. Вважається, що розробка 2D ігор за допомогою Unreal Engine 5 є дуже непрактичною або навіть неможливою, але існують маловідомі інструменти та техніки, які полегшують цей процес. Сам по собі Unreal Engine 5 обмежений для розробки 2D ігор, саме тому виникає потреба у поєднанні різних плагінів та технік, щоб розширити функціонал ігрового рушія та покращити якість програмного продукту [1–3].

Формулювання цілей статті

Метою роботи є покращення ігрового процесу 2D гри за допомогою розширення функціональних можливостей ігрового рушія Unreal Engine 5. Це сприятиме підвищенню якості гри та її графіки, також забезпечить кращу роботу анімацій у грі, що також покращить ігровий процес та ігровий досвід користувача. Застосування різних технік полегшить та прискорить розробку гри, це сприятиме економії часу, потрібного для розробки гри.

Для досягнення цієї мети потрібно вирішити такі задачі:

- 1) провести аналіз існуючих інструментів та технік для покращення розробки авторської 2D гри з використанням ігрового рушія Unreal Engine 5;
- 2) проаналізувати проблеми, які виникли під час розробки авторської 2D гри на ігровому рушії Unreal Engine 5 та обґрунтувати рішення для усунення цих проблем;
- 3) провести тестування розробки;
- 4) зробити висновки на основі проведених досліджень.

Виклад основного матеріалу

Розроблена авторська 2D гра за допомогою ігрового рушія Unreal Engine 5 – це приклад використання сучасних технологій для розробки класичних ретро-ігор. Найбільша проблема у розробці 2D гри за допомогою ігрового рушія Unreal Engine 5 – це налаштування анімацій, щоб вони виглядали реалістично та без багів. Для розробки 2D ігор у Unreal Engine 5 існує вбудований плагін Paper2D. Paper 2D – це система ігрового процесу та анімації Unreal Engine, яку можна використовувати для створення 2D та 2D/3D гібридних проєктів [4, 5]. Основний тип ресурсу Paper 2D називається спрайтом. Спрайти – це статичні двовимірні текстури, які можна використовувати для представлення персонажів або об'єктів під час розробки проєктів у Unreal Engine.

Для коректного відображення текстури спрайта потрібно встановити певні налаштування. За замовчуванням усі 2D-зображення, імпортовані в Unreal Engine, будуть імпортовані як текстурні ресурси. Обробка зображень за замовчуванням, яка призначена для обробки текстур для 3D-моделей і об'єктів, згладжує та пом'якшує файли з низькою роздільною здатністю, наприклад 2D-персонажі та об'єкти. Тому для того, щоб зберегти вигляд спрайтів із низькою роздільною здатністю, потрібно застосувати до ресурсу налаштування двовимірної текстури паперу. Крім того, необхідно змінити фільтр текстури з Default на Nearest. На рис. 1 подано порівняння текстур спрайта головного персонажа з правильними налаштуваннями та без них.

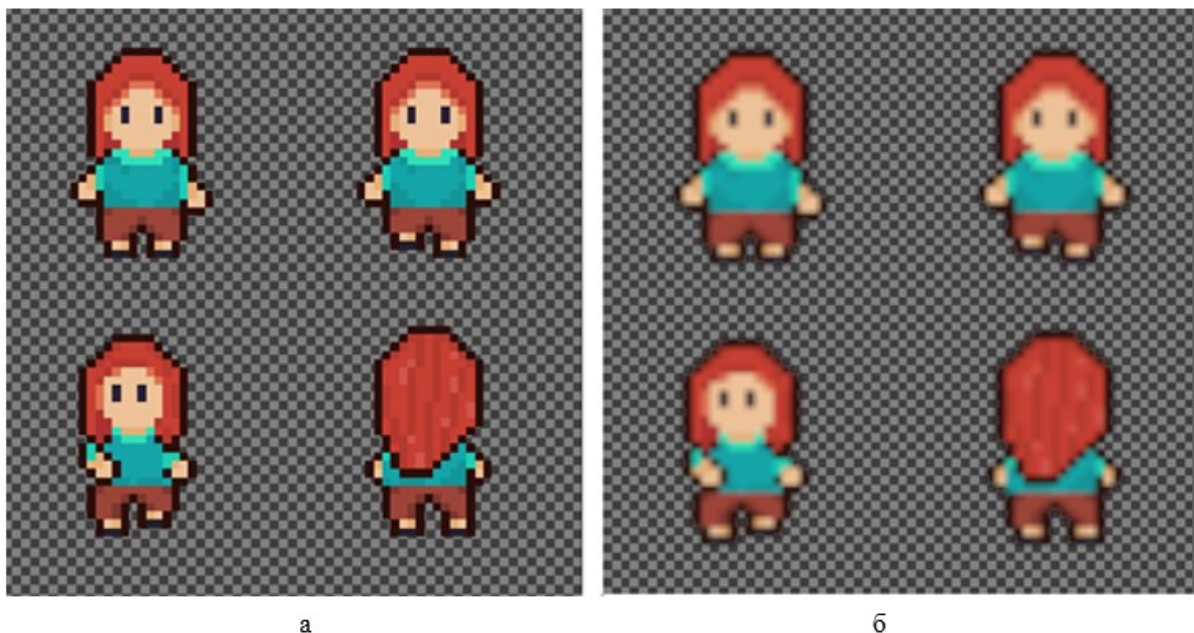


Рис. 1. Порівняння текстур спрайта головного персонажа: а - з правильними налаштуваннями; б – без правильних налаштувань

Крім того, для пришвидшення створення фліпбуків найкраще імпортувати спрайти у Unreal Engine 5 у форматі json. Це значно прискорює ділення спрайт файла на однакові кадри та створює

правильні фліпбуки для кожної анімації. Такий метод прискорює імпорт файлів в рази. Зауважимо, якщо є тільки файл спрайта у форматі png, тоді потрібно розділи лист спрайтів на кадри. Для цього найкраще використати режим сітки, а не авто та розділити висоту на кількість рядків, а ширину на кількість стовпців. Таким чином лист спрайтів поділиться правильно і анімаційні фліпбуки будуть відображатись коректно [6–8]. На рис. 2 наведено приклад ділення листа спрайта на кадри з використанням автоділення та з ручним діленням.



Рис. 2. Порівняння ділення листа спрайта на кадри: а – з використанням автоділення; б – ручне ділення

Для налаштування анімацій для певних подій найкраще використовувати плагін PaperZD. За допомогою цього плагіну є можливість швидко налаштувати направлену анімацію, а також створити анімаційні сповіщення, які можна використати для налаштування звукових та візуальних ефектів, а також для створення більш живого ефекту від користування зброєю. Плагін PaperZD має певні можливості, а саме, створювати власних анімованих персонажів та об'єктів, керуючи їх анімацією за допомогою простого у використанні, але потужного графіка анімації; додавати сповіщення анімації до послідовностей, дозволяючи запускати події в дуже конкретних кадрах без необхідності керувати ними самостійно; реалізувати власну логіку сповіщень у C++, Blueprints або навіть як Custom сповіщення у власному Animation Blueprint; легко створювати 2-вимірну TopDown гру, ізометричну або іншу «спрямовану» 2D-гру, дозволяючи налаштувати та вибрати кут візуалізації для використання; створювати власні анімаційні послідовності на «Sequencer» та запускати всі сповіщення та налаштування, уже встановлені у вашому персонажі; легко розширювати плагін відповідно до своїх потреб. З PaperZD можна мати інструменти, які потрібні для створення неймовірних анімаційних проєктів, водночас керуючи ними дуже легко та інтуїтивно зрозуміло [9]. На рис. 3 представлено налаштування різнонаправленої анімації.

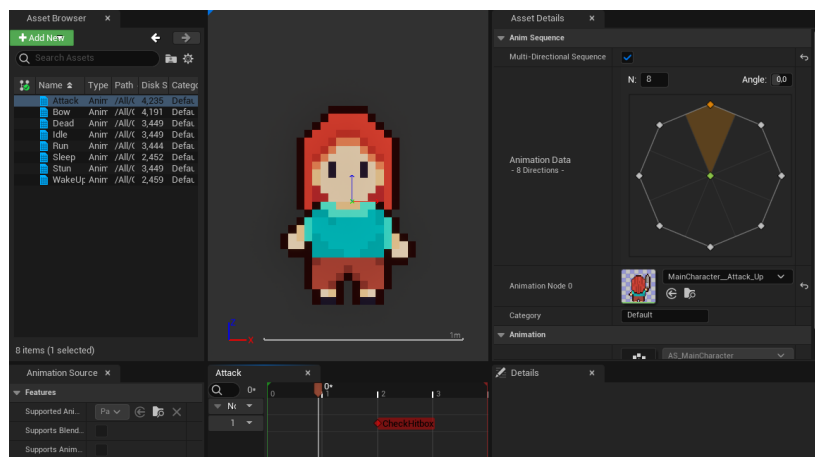


Рис. 3. Налаштування різнонаправленої анімації

Для налаштування руху та відображення правильних анімацій створено *AnimationSource Blueprints*. За допомогою нього можна створити State Machine, де налаштовують умови для відображення потрібних анімацій. Таким чином налаштування відображення певних анімацій значно оптимізованіше, ніж без AnimationSource Blueprints. Під час розроблення State Machine можна створити умови для зміни анімацій в залежності від направленості персонажа [10]. На рис. 4 представлено налаштування State Machine Locomotion.



Рис. 4. Налаштування State Machine Locomotion

Таке налаштування більш гнучке та дозволяє пришвидшити та оптимізувати розробку. Також у більш масштабних проєктах таке налаштування оптимізує гру, тому що для налаштування потрібно менше нодів, ніж налаштування зміни анімацій з умовами без плагіну PaperZD. Використання State Machine у AnimationSource Blueprints економить час, адже налаштування анімацій з більшими напрямленнями та для більшої кількості акторів було б громіздким, якщо налаштовувати здійснювати в Actor Blueprints. На рис. 5 показано неоптимізоване налаштування зміни анімацій для ходьби та бездіяльності.

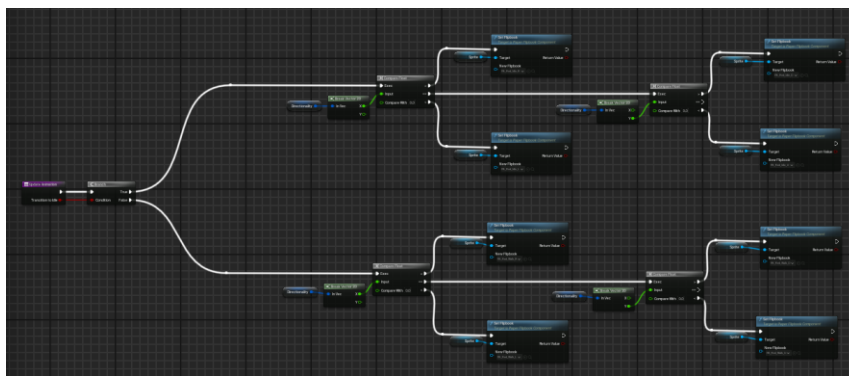


Рис. 5. Неоптимізоване налаштування зміни анімацій для ходьби і бездіяльності

Під час розробки виникла проблема із створенням колізій для взаємодії. Потрібно було зробити так, щоб персонаж міг заходити за предмети навколишнього середовища, наприклад, стояти за палаткою чи деревом. Для такої реалізації потрібно було змістити якір персонажа нижче. Тут виникли такі проблеми, наприклад, якщо змістити якір персонажа режиму Action, тоді зламаються всі колізії пов'язані з атакою: колізія отримання шкоди та колізія меча. Тобто якщо вдарити біля персонажа, а не по ньому, персонаж однаково отримає шкоду, використання меча також буде дивне, тому що зона його ураження не буде збігатися з нарисованим мечем. Для персонажа режиму RPG також виникнуть проблеми, через те, що висота персонажа 25 пікселів, то середина зміщується некоректно, або 12, або 13 пікселів. Також за таких налаштувань у режимі RPG колізія взаємодії працювала некоректно, замість того, щоб перекриватись з колізіями інших персонажів, вона штовхала їх у певному напрямку, навіть, якщо всі налаштування колізій виставлені правильно. В процесі дослідження колізій, виявлено, що плагін PaperZD змінює основний Capsule Component у Actor Blueprint, також цей Capsule Component неможливо видалити, тому що це основний компонент актора [11]. Під час експериментів було знайдено рішення. Дерева було поділено на два окремі компоненти: листя та стовбур, у стовбура було зменшено колізію до половини, а у листя колізії немає взагалі, тому коли персонаж проходить вище умовного центра стовбура, тоді це буде виглядати як персонаж проходить позаду дерева. Щодо палаток, каміння, кущів та інших предметів для них просто зменшено колізію, тобто верхня частина без колізії. Таким чином було знайдено рішення та ігровий процес виглядає більш природньо. На рис. 6 наведено приклад, як персонаж проходить позаду предметів.

Крім того, іншою проблемою було те, що на слабких ПК після удару персонажа відкидало на меншу відстань, ніж на потужніших ПК з більшою частотою оновлення екрана. Тобто, маючи частоту оновлення екрана 60 Гц, персонажа відкидало на умовну відстань X, але при частоті оновлення екрана 165 Гц, ця відстань становила $1,5 \div 2X$, що надавало перевагу для гравців з кращим ПК. Для розв'язання цієї проблеми було змінено налаштування функції відкидання без прив'язки до delta time, яке використовує частоту оновлення екрана для обчислення відстані відкидання. Таким чином гра працює однаково на різних ПК, для оптимізування роботи гри краще змінити графічні налаштування проєкту.



Рис. 6. Приклад, як персонаж проходить позаду предметів

За замовчуванням всі налаштування у Unreal Engine 5 встановлені для 3D проєктів, деякі з них є надлишковими для 2D проєктів та можуть значно погіршити продуктивність гри. Тому для оптимізації продуктивності потрібно змінити деякі налаштування у налаштуваннях проєктів. Спочатку потрібно вимкнути Auto Exposure, встановити Auto Exposure на Manual та Auto Exposure Bias – на 0.0. Зауважимо, що Auto Exposure повільно робить темні сцени світлішими, що надає реалістичного ефекту, але це впливає на кольори у піксельній грі, тому найкраще вимкнути Auto Exposure. Потрібно вимкнути Motion Blur, що є неактуальним для 2D графіки, встановити Anti Aliasing на None, щоб уникнути дивного вигляду пікселів. Встановити Dynamic Global Illumination Method та Reflection Method на None, ці методи дуже ресурсно затратні та в основному використовуються з налаштуванням світла, якого немає у 2D проєктах [12]. Встановити Shadow Map Method на Shadow Maps, вимкнути Ambient Occlusion та Ambient Occlusion Static Fraction, оскільки гра немає жодних тіней, тому ці методи непотрібні. Якщо увімкнути Forward Shading, то це переключить проєкт із Forward Shading на direct. Для створення VR або 2D ігор це зазвичай більш продуктивно. Встановлено, що кількість кадрів за секунду (FPS) збільшилось на 7% на ноутбучі та на 33% – на Steam Deck [13].

Під час розробки потрібно налаштувати PostProcessing Volume. За замовчуванням PostProcessing Volume вплине лише на камери, які знаходяться всередині області дії. Але щоб застосувати ефект до всього етапу, потрібно додати PostProcessing Volume до кожного рівня гри та налаштувати його. Спочатку потрібно розширити розміри PostProcessing Volume, тобто увімкнути Infinite Extent (Unbound). Встановити Exposure Compensation, MinEV100, MaxEV100, Vignette Intensity, Expand Gamut, Tone Curve Amount на 0.0. Це допоможе налаштувати більш реальну кольоропередачу без додаткових фільтрів [13]. Також існує платний плагін Pixel Perfect Tools, який покращує вигляд піксел графіки у Unreal Engine 5 [14].

Всі ці особливості були використанні при розробці 2D гри на ігровому рушії Unreal Engine 5 та допомогли значно покращити ігровий процес та розробку авторської гри. Порівнюючи розроблену авторську 2D гру з іншим 2D проєктом розробленим у Unreal Engine 5 без використання описаних технік та плагіну PaperZD встановлено, що використання плагіну PaperZD та використання файлів з розширенням json значно прискорили та покращили розробку 2D гри. Використання всіх описаних вище технік збільшило швидкість розробки гри на 50%, а змінення налаштувань проєкту та використання модифікованого PostProcessing Volume значно покращили графіку та продуктивність гри. Отже, розроблена авторська 2D гра пройшла тестування, має покращений ігровий процес та графіку, а саме: чітку кольоро-передачу, відсутність розмиття, яскраві кольори та продуктивність: гра працює плавно та без лагів навіть на слабких ПК.

Висновки з даного дослідження і перспективи подальших розвідок у даному напрямі

1. Проведено дослідження існуючих технологічних рішень в ігровій індустрії. Проаналізовано особливості, інструменти та техніки для розробки 2D гри з використанням ігрового рушія Unreal Engine 5. Встановлено, що ігровий рушій Unreal Engine 5 обмежений для розробки двовимірних ігор, тому потребує додаткових плагінів для покращення розробки. Не зважаючи на свої обмеження Unreal Engine 5 – це гнучкий та сучасний ігровий рушій, доступний для використання не тільки професіональним програмістам, а й графічним дизайнерам.

2. Використання вбудованого плагіну Paper2D разом із плагіном PaperZD значно полегшило та прискорило розробку гри. Використання State Machine для налаштувань анімацій оптимізувало розробку та покращило продуктивність гри. Використання файлів формату json пришвидшило розробку авторської 2D гри на 20%. Крім того, під час розробки виявилась проблема з налаштуванням колізій для взаємодії та якорем персонажа. В процесі дослідження було знайдено рішення для цієї проблеми та впроваджено у 2D гру. Для покращення продуктивності ігрового процесу було встановлено певні графічні налаштування до проєкту та кожного рівня.

3. Під час тестування авторської 2D гри програмний модуль показав високу продуктивність: плавність гри, як на слабких, так і потужних ПК; якісну піксельну графіку: яскраві кольори,

недеформовані пікселі. Всі описані особливості розробок 2D ігор на ігровому рушії Unreal Engine 5 були використанні під час розроблення авторської 2D гри та оптимізували процес її розроблення. Внаслідок цього швидкість розробки гри збільшена на 50%, це вплинуло на якість ігрового процесу та графіки.

Література

1. Єрмолаєва М. А., Крилик Л. В. Особливості розробки програмного модуля авторської 2D TopDown гри на ігровому рушії Unreal Engine 5. *LIV Всеукраїнська науково-технічна конференція факультету інтелектуальних інформаційних технологій та автоматизації*. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2025/paper/view/23096/19120> (дата звернення: 02.04.2025).
2. Paper 2D – How To. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/how-to-use-paper-2d-in-unreal-engine> (дата звернення: 02.04.2025).
3. Paper 2D. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/paper-2d-overview-in-unreal-engine?application_version=5.4 (дата звернення: 02.04.2025).
4. Getting Started with Your Paper 2D Project. URL: <https://dev.epicgames.com/community/learning/courses/LwA/unreal-engine-getting-started-with-your-paper-2d-project/n8l/unreal-engine-getting-started-with-your-paper-2d-project-introduction> (дата звернення: 02.04.2025).
5. Paper 2D Sprites. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/how-to-import-and-use-paper-2d-sprites-in-unreal-engine?application_version=5.4 (дата звернення: 02.04.2025).
6. Paper 2D Flipbooks. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/paper-2d-flipbooks-in-unreal-engine?application_version=5.2 (дата звернення: 02.04.2025).
7. How to make 2D Games with Unreal Engine 5 [2024 Edition]. URL: <https://dev.epicgames.com/community/learning/tutorials/n1aL/how-to-make-2d-games-with-unreal-engine-5-2024-edition> (дата звернення: 02.04.2025).
8. Unreal Engine Ready 2D Assets. URL: <https://cobracode.notion.site/Unreal-Engine-Ready-2D-Assets-9349db17b6f04b71923e6b9d63dbf4c7> (дата звернення: 02.04.2025).
9. PaperZD. URL: <https://www.fab.com/listings/6664e3b5-e376-47aa-a0dd-f7bbbd5b93c0> (дата звернення: 02.04.2025).
10. State Machines. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/state-machines-in-unreal-engine> (дата звернення: 02.04.2025).
11. Collision Overview. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/collision-in-unreal-engine---overview> (дата звернення: 02.04.2025).
12. Global Illumination. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/global-illumination-in-unreal-engine> (дата звернення: 02.04.2025).
13. Best 2D Settings for Unreal Engine 5. URL: <https://cobracode.notion.site/Best-2D-Settings-for-Unreal-Engine-5-0895133a56924f1ca7f2149cf111983b#fe336dea026f4e559ab9203c662bc7b8> (дата звернення: 02.04.2025).
14. Pixel Perfect Tools. URL: <https://cobracode.notion.site/Pixel-Perfect-Tools-f03c0475060e415486656ce07f77f02c> (дата звернення: 02.04.2025).

References

1. Yermolaieva M. A., Krylyk L. V. Osoblyvosti rozrobky prohramnoho modulja avtorskoi 2D TopDown hry na ihrovomu rushii Unreal Engine 5. *LIV Vseukrainska naukovo-tekhnichna konferentsiia fakultetu intelektualnykh informatsiynykh tekhnolohii ta avtomatyzatsii*. URL: <https://conferences.vntu.edu.ua/index.php/all-fksa/all-fksa-2025/paper/view/23096/19120> (data zvernennia: 02.04.2025).
2. Paper 2D - How To. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/how-to-use-paper-2d-in-unreal-engine> (data zvernennia: 02.04.2025).
3. Paper 2D. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/paper-2d-overview-in-unreal-engine?application_version=5.4 (data zvernennia: 02.04.2025).
4. Getting Started with Your Paper 2D Project. URL: <https://dev.epicgames.com/community/learning/courses/LwA/unreal-engine-getting-started-with-your-paper-2d-project/n8l/unreal-engine-getting-started-with-your-paper-2d-project-introduction> (data zvernennia: 02.04.2025).
5. Paper 2D Sprites. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/how-to-import-and-use-paper-2d-sprites-in-unreal-engine?application_version=5.4 (data zvernennia: 02.04.2025).
6. Paper 2D Flipbooks. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/paper-2d-flipbooks-in-unreal-engine?application_version=5.2 (data zvernennia: 02.04.2025).

7. How to make 2D Games with Unreal Engine 5 [2024 Edition]. URL: <https://dev.epicgames.com/community/learning/tutorials/n1aL/how-to-make-2d-games-with-unreal-engine-5-2024-edition> (data zvernennia: 02.04.2025).
8. Unreal Engine Ready 2D Assets. URL: <https://cobracode.notion.site/Unreal-Engine-Ready-2D-Assets-9349db17b6f04b71923e6b9d63dbf4c7> (data zvernennia: 02.04.2025).
9. PaperZD. URL: <https://www.fab.com/listings/6664e3b5-e376-47aa-a0dd-f7bbbd5b93c0> (data zvernennia: 02.04.2025).
10. State Machines. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/state-machines-in-unreal-engine> (data zvernennia: 02.04.2025).
11. Collision Overview. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/collision-in-unreal-engine---overview> (data zvernennia: 02.04.2025).
12. Global Illumination. URL: <https://dev.epicgames.com/documentation/en-us/unreal-engine/global-illumination-in-unreal-engine> (data zvernennia: 02.04.2025).
13. Best 2D Settings for Unreal Engine 5. URL: <https://cobracode.notion.site/Best-2D-Settings-for-Unreal-Engine-5-0895133a56924f1ca7f2149cf111983b#fe336dea026f4e559ab9203c662bc7b8> (data zvernennia: 02.04.2025).
14. Pixel Perfect Tools. URL: <https://cobracode.notion.site/Pixel-Perfect-Tools-f03c0475060e415486656ce07f77f02c> (data zvernennia: 02.04.2025).