

3. Mhatre T., Samel M., Chaudhari U., Banekar S. LostDune: A 2D Platformer Game Using Godot 4 Engine with Finite State Machine Architecture. IRE Journals. 2026. Vol. 9, No. 8. URL: <https://doi.org/10.64388/IREV9I8-1714549>.

4. Godot Engine 4 documentation. ShapeCast3D. URL: https://docs.godotengine.org/en/stable/classes/class_shapecast3d.html (дата звернення: 25.03.2026).

5. Vanhove S. Learning GDScript by Developing a Game with Godot 4: A Fun Introduction to Programming in GDScript 2.0 and Game Development Using the Godot Engine. Packt Publishing, 2024. 378 p.

РОЗРОБКА МІЖПЛАТФОРМНОЇ КОМП'ЮТЕРНОЇ ГРИ ЖАНРУ IDLE CLICKER «CANDY CLICKER» НА БАЗІ РУШІЯ GODOT ENGINE

Бухта Роман Тарасович

здобувач першого рівня вищої освіти спеціальності Комп'ютерні науки
Тернопільський національний педагогічний університет імені Володимира Гнатюка
buhta_rt@fizmat.tnpu.edu.ua

Вовкодав Олександр Валерійович

кандидат технічних наук, доцент кафедри інформатики та методики її навчання
Тернопільський національний педагогічний університет імені Володимира Гнатюка
o.vovkodav@tnpu.edu.ua

Сучасна індустрія відеоігор є одним із найбільш зростаючих сегментів глобального ринку цифрових технологій: за сукупним обсягом доходів вона перевищує показники кінематографічної та музичної галузей разом узятих. Загальна чисельність активних користувачів ігрових платформ у світі наразі становить понад 3 млрд. осіб. Серед значної різноманітності існуючих ігрових жанрів окреме місце посідають так звані інкрементальні ігри (англ. *Idle/Clicker games*) – жанр, що протягом останнього десятиліття еволюціонував від нішевого експериментального формату до комерційно самостійного ринкового сегменту. За даними галузевої аналітики, обсяг глобального ринку *IDLE* ігор у 2024 році сягнув 2,5 млрд. доларів США, а прогнозовані темпи зростання передбачають досягнення позначки 6,1 млрд. доларів до 2033 року [3]. Сукупність характеристик цього жанру – зокрема, відносно низький поріг входу для розробників програмного забезпечення, мінімальні вимоги до апаратних ресурсів кінцевих пристроїв, а також ефективні з точки зору поведінкової психології механіки залучення та утримання аудиторії – визначає його як перспективну дослідницьку платформу для вивчення фундаментальних принципів ігрового проектування та прикладних аспектів програмної інженерії.

Стрімкий розвиток відкритих ігрових рушіїв, зокрема *Godot Engine*, що розповсюджується під ліцензією MIT з відкритим вихідним кодом, суттєво розширює технологічні можливості для створення повнофункціональних ігрових застосунків без необхідності понесення ліцензійних витрат.

Актуальність представленого дослідження визначається кількома взаємопов'язаними чинниками. По-перше, існує об'єктивна потреба у систематизації та узагальненні практичного досвіду застосування рушія *Godot Engine* у контексті розробки ігрових продуктів жанру *Idle/Clicker*. По-друге, недостатньо дослідженими залишаються архітектурні патерни проектування програмного забезпечення, специфічні для *IDLE* ігрових систем. По-третє, практична реалізація збалансованої ігрової системи потребує інтеграції знань із суміжних галузей – зокрема, врахування психологічних механізмів залучення та утримання гравців, а також дотримання сучасних вимог до кросплатформної сумісності застосунків [1]. Метою дослідження є розробка повнофункціонального ігрового застосунку «*Candy Clicker*» з

використанням рушія *Godot Engine* версії 4.x та мови програмування *GScript*, що реалізує класичні механіки *IDLE* жанру в оригінальному тематичному оформленні кондитерської тематики та забезпечує збалансований ігровий досвід із можливістю збереження прогресу між сесіями [4].

Аналіз жанру *Idle/Clicker* показав, що він ґрунтується на п'яти ключових механіках: активний клік (безпосередня взаємодія гравця з ігровим елементом), система генераторів пасивного доходу, ієрархічна система покращень, система престижу та офлайн-прогрес. Феноменальна популярність жанру пояснюється здатністю ефективно активувати системи винагороди мозку через «дофаміновий цикл» та задовольняти фундаментальну людську потребу у відчутті прогресу. Порівняльний аналіз актуальних інструментальних засобів для двовимірної ігрової розробки – *Unity*, *Unreal Engine*, *GameMaker*, *Defold* та *Construct* – дозволив обґрунтувати вибір рушія *Godot Engine* як технологічно оптимального рішення для цілей даного дослідження. Визначальними критеріями такого вибору стали: відкритість вихідного коду та розповсюдження під ліцензією MIT, що унеможливує виникнення роялті-зобов'язань; наявність вбудованої мови програмування *GScript* із синтаксисом, типологічно близьким до *Python*, що забезпечує низький поріг освоєння; а також нативна підтримка мультиплатформного експорту на *Windows*, *Linux*, *Android* та *HTML5* [2].

На етапі проектування спроектовано концепцію гри в кондитерській тематиці з 11 типами генераторів пасивного доходу – Желейка, Мармеладка, Кексик, Ледяник, Пончик, Ріжок та ін. – кожен з яких має унікальну базову вартість і продуктивність. Архітектура програмного забезпечення побудована на компонентному підході із централізованим головним скриптом *main.gd* та системою сигналів *Godot* для слабкозв'язаної комунікації між модулями. Формалізовано математичну модель ігрової економіки з експоненційною функцією вартості генераторів $C(n) = C_0 \times r^n$ із коефіцієнтом зростання $r = 1.12$ та лінійною продуктивністю $P(n) = P_0 \times n$, що забезпечує збалансований темп прогресії з поступовим уповільненням на пізніх етапах гри [3]. Спроектовано систему збереження прогресу на основі *JSON*-серіалізації з записом у директорію *user://*, механізмом валідації даних за принципом *graceful degradation* та стратегією автозбереження, що поєднує періодичний (кожні 30 секунд) та подієвий підходи.

У процесі реалізації розроблено алгоритм офлайн-прогресу з обмеженням накопичення, модуль форматування великих чисел та основні ігрові механіки засобами *GScript*. Інтерфейс створено на базі системи вузлів *Godot* із забезпеченням адаптивності та зручності використання. Проведене тестування підтвердило стабільність роботи застосунку на різних платформах та його відповідність заданим вимогам. Виконано оптимізацію продуктивності, що забезпечило ефективну роботу та низьке енергоспоживання. Наукова новизна полягає у поєднанні архітектурних підходів *Godot* із механіками *IDLE* ігор, а практичне значення – у створенні готового ігрового продукту, придатного до використання та подальшого розвитку.

Список використаних джерел

1. Мельник П. П., Василенко Я. П. Особливості використання рушія *Godot* та *C#* для розробки ігрових застосунків. Збірник наукових праць. Тернопіль, 2025. С. 156.
2. Comparative Analysis of Unity and Godot for 2D Game Development / Preprints.org. 2025. URL: <https://www.preprints.org/manuscript/202511.1981> (дата звернення: 24.03.2026).
3. Buergi J. Idle Yet Engaged: How Idle Games Satisfy Our Needs for Competence and Autonomy: diss. Rensselaer Polytechnic Institute, 2025. 120 p. URL: <https://www.proquest.com/openview/daa4dd324eab2af807e492f8c945122a/1> (дата звернення: 26.03.2026).

4. Alharthi S. A., Toups Z. O., Alsaedi O., Tanenbaum J., Hammer J. Exploring Engagement in Idle Games. Proceedings of the Annual Symposium on Computer-Human Interaction in Play. ACM, 2018. URL: <https://dl.acm.org/doi/10.1145/3242671.3242700> (дата звернення: 26.03.2026).

РОЗРОБКА КРИПТОВАЛЮТНОГО ЗАСТОСУНКУ POINTC З ВИКОРИСТАННЯМ ANGULAR, TYPESCRIPT ТА АРХІТЕКТУРИ MVC НА ПЛАТФОРМІ .NET

Воропай Ігор Олександрович

здобувач першого рівня вищої освіти спеціальності Комп'ютерні науки
Тернопільський національний педагогічний університет імені Володимира Гнатюка
voropaj_io@fizmat.tnpu.edu.ua

Василенко Ярослав Пилипович

викладач кафедри інформатики та методики її навчання
Тернопільський національний педагогічний університет імені Володимира Гнатюка
yava@fizmat.tnpu.edu.ua

Сучасний розвиток цифрових технологій супроводжується активним впровадженням криптовалют у різні сфери економіки. Ринок цифрових активів демонструє стійке зростання, що зумовлює підвищений попит на програмні засоби для автоматизації процесів торгівлі та аналізу ринку. У зв'язку з цим розробка вебзастосунків для автоматизованого трейдингу є актуальним напрямом досліджень у галузі програмної інженерії [4].

Використання автоматизованих торгових систем дозволяє забезпечити безперервний моніторинг ринку, оперативне реагування на зміни цін та мінімізувати вплив людського фактору. Сучасні підходи до розробки програмного забезпечення передбачають використання модульної архітектури, що сприяє підвищенню якості та масштабованості систем [1].

Метою роботи є розробка криптовалютного вебзастосунку PointC, який забезпечує автоматизацію торгових операцій із використанням сучасних технологій веброзробки та дозволяє користувачам ефективно управляти торговими стратегіями.

Для реалізації клієнтської частини використано фреймворк Angular, який забезпечує створення односторінкових застосунків із високою продуктивністю та зручною структурою компонентів. Використання мови TypeScript дозволяє підвищити надійність програмного коду та спростити процес розробки [1].

Серверну частину реалізовано на платформі ASP.NET Core із застосуванням архітектури Model-View-Controller, що забезпечує розділення бізнес-логіки, інтерфейсу користувача та обробки даних. Такий підхід дозволяє підвищити гнучкість системи та спростити її подальше масштабування [2].

У процесі проектування було визначено основні функціональні компоненти системи: модуль управління користувачами, модуль торгових ботів, модуль обробки ринкових даних та модуль інтеграції з API криптовалютних бірж. Використання API платформи Binance дозволяє отримувати актуальні дані у режимі реального часу та виконувати торгові операції [3].

Архітектура застосунку побудована за принципом клієнт-серверної взаємодії із використанням REST API. Передача даних здійснюється у форматі JSON, що забезпечує ефективну взаємодію між компонентами системи. Для збереження інформації використано реляційну базу даних, яка містить дані про користувачів, налаштування ботів та історію торгових операцій [2].

Особливу увагу приділено питанням інформаційної безпеки. У застосунку реалізовано механізми автентифікації з використанням JWT-токенів, що забезпечує