



УДК [004.932.2:793.7]:004.896

ПРОЄКТУВАННЯ ТА РОЗРОБКА КОНТЕКСТНО-ЗАЛЕЖНОЇ СИСТЕМИ УПРАВЛІННЯ ДЛЯ ІГОР ЖАНРУ «TOWER DEFENSE» В ДОПОВНЕНІЙ РЕАЛЬНОСТІ

DESIGN AND DEVELOPMENT OF A CONTEXT- ADAPTIVE CONTROL SYSTEM FOR TOWER DEFENSE GAMES IN AUGMENTED REALITY

Шестопалов С.В.¹, Москальчук В.С.², Логінов Д.О.³

^{1,2,3} Одеський національний технологічний університет, м. Одеса, Україна

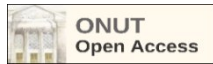
ORCID: ¹<http://orcid.org/0000-0001-8941-4610>

E-mail: ¹sshestopalov1984@gmail.com, ²kurihawok@gmail.com, ³777loginovdanja@gmail.com

Copyright © 2025 by author and the journal “Automation of technological and business – processes”.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0>



DOI: [10.15673/atbp.v17i4.3325](https://doi.org/10.15673/atbp.v17i4.3325)

Анотація. В роботі представлено результати дослідження, проєктування та розробки системи контекстно-залежного управління для ігор жанру «Tower Defense», реалізованих із використанням технології доповненої реальності (AR). Актуальність роботи зумовлена стійким інтересом до ігор жанру «Tower Defense» та потребою в інтеграції інноваційних механік і технологій, спрямованих на розширення ігрового досвіду, підвищення рівня занурення гравця та забезпечення більш природної й адаптивної взаємодії в умовах використання новітніх технологій.

Метою дослідження є створення проєкту та реалізація демонстраційної версії системи управління, здатної динамічно адаптуватися до контексту ігрової ситуації.

У роботі дано визначення термінам система управління та контекстно-залежна система управління, проаналізовано особливості ігор жанру «Tower Defense», визначено класичні елементи жанру, ключові та додаткові механіки.

Зазначено, що проєктування та розробку контекстно-залежної системи управління доцільно здійснювати на прикладі конкретного ігрового проєкту, а не в абстрактній формі. З цією метою розроблено фрагмент проєктної документації гри під назвою «CoreZero». На основі спроектованої гри розроблено проєкт контекстно-залежної системи управління, яка працює в двох режимах: режимі управління ігровою сценою і режимі управління ігровим полем та об'єктами.

Для розробки демонстраційної версії обрано ігровий рушій Unity та платформу ARCore. Реалізовано демонстраційну версію гри «CoreZero» з контекстно-залежною системою управління, що адаптується під ігрову ситуацію.

Спроектована та розроблена контекстно-залежна система управління в демонстраційній версії гри демонструє здатність адаптувати ігрову логіку відповідно до контексту дії гравця, що забезпечує покращену взаємодію з цифровими об'єктами. Розроблене рішення може бути використане як модуль для розширення функціональності існуючих ігрових рушіїв.

Abstract. The paper presents the results of research, design, and development of a context-adaptive control system for games of the Tower Defense genre implemented using Augmented Reality (AR) technology. The relevance of the study is driven by the sustained interest in the Tower Defense genre and the need to integrate innovative mechanics and technologies aimed at expanding the gaming experience, increasing player immersion, and ensuring more natural and adaptive interaction within modern technological environments.

The aim of the research is to design and development a prototype of a control system capable of dynamically adapting to the context of the gameplay situation.



In this work, the terms control system and context-adaptive control system are defined, the characteristics of Tower Defense games are analyzed, and the classical elements of the genre, along with its key and additional mechanics, are identified.

It is noted that the design and development of a context-adaptive control system are most effective when implemented within a specific game project rather than in an abstract form. For this purpose, a fragment of the design documentation for a game titled «CoreZero» has been developed. Based on the designed game concept, a context-adaptive control system was created, operating in two modes: scene management mode and game field and object control mode.

The Unity game engine and ARCore platform were selected for developing the prototype. A demonstration version of the «CoreZero» game featuring a context-adaptive control system that adapts to gameplay conditions has been developed. The designed and developed control system within the demonstration version shows the ability to adjust game logic in response to player actions, thereby improving interaction with digital objects. The proposed solution can also serve as a module for extending the functionality of existing game engines.

Ключові слова: контекстно-залежна система управління, доповнена реальність, Tower Defense, ARCore

Key words: context-adaptive control system, augmented reality, Tower Defense, ARCore

ВСТУП

Індустрія розробки відеоігор стрімко зростає та займає провідне місце серед напрямів сучасного програмного забезпечення. Сьогодні відеоігри – це не лише розвага, а й потужний інструмент залучення користувачів, навчання та комунікації. Особливої популярності набувають мобільні ігри, доповнені новітніми технологіями, зокрема доповненою реальністю (AR – Augmented Reality), яка дозволяє поєднати віртуальні об'єкти з реальним світом та забезпечує новий рівень імерсивності.

Жанр «Tower Defense» (TD) вже багато років утримує високі позиції у топах завантажень, завдяки простому, але глибокому геймплею, що передбачає стратегічне мислення, планування та управління ресурсами. Успішні проекти демонструють, що інтерес до цього жанру не зменшується, а навпаки – зростає завдяки оновленням, новим механікам та підтримці спільноти. Інтеграція «Tower Defense» у доповнену реальність є актуальним і перспективним напрямом, що поєднує популярний жанр з новітніми технологіями, розширюючи ігровий досвід і взаємодію гравця з простором. У цьому контексті особливої актуальності набуває розробка нових або вдосконалення класичних механік, що лежать в основі геймплею. Саме вони формують глибину стратегічної взаємодії, забезпечують варіативність тактичних рішень та впливають на ступінь залученості гравця. Завдяки адаптації цих механік під можливості доповненої реальності, з'являється потенціал для створення інноваційних ігрових рішень, що підвищують рівень імерсивності та забезпечують унікальний користувацький досвід.

Разом з тим перенос гри в доповнену реальність породжує низку нових викликів пов'язаних, зокрема, з правильним відображення інтерфейсу і реалізацією системи управління. Зміна формату взаємодії з грою – з традиційного дотикового управління до просторово-орієнтованої взаємодії – значно підвищує ризик випадкових або помилкових дій з боку користувача. У цьому контексті особливої важливості набуває розробка контекстно-залежної системи управління, здатної динамічно адаптувати функціональність управління до поточних умов, дій гравця та особливостей ігрової сцени в реальному просторі.

1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Система управління у відеоіграх виступає ключовим інструментом взаємодії користувача з ігровим середовищем, визначаючи комфортність, ефективність та глибину залучення гравця до процесу. Особливої актуальності набуває впровадження адаптивних контекстно-залежних механізмів управління, здатних підлаштовуватися під ігрову ситуацію та мінімізувати кількість небажаних або випадкових дій. Для глибшого розуміння ролі та особливостей таких систем доцільно розглянути їх визначення.

Система управління в іграх – це сукупність апаратних і програмних засобів, що забезпечують взаємодію між гравцем і віртуальним ігровим середовищем, дозволяючи здійснювати контроль над об'єктами гри, приймати рішення, активувати дії та реагувати на події в межах геймплею. Основними складовими системи управління є: пристрої введення (наприклад, клавіатура, миша, сенсорний екран, джойстик, контролери доповненої або віртуальної реальності), механізми обробки введених даних, внутрішня логіка інтерпретації команд, а також елементи візуального або тактильного зворотного зв'язку [1, 2].

Контекстно-залежна система управління – це тип системи управління, яка динамічно адаптує функціональність та реакцію на введення користувача залежно від поточного стану ігрового середовища, положення об'єктів, дій гравця або просторових умов (особливо в AR/VR-середовищах). Така система підвищує точність і ефективність взаємодії, зменшує кількість помилкових або випадкових дій, а також покращує загальний рівень користувацького досвіду [1, 2].

Контекстно-залежне управління є особливо актуальним для ігор у доповненій реальності, де фізичне положення гравця, напрям погляду чи розташування віртуальних об'єктів у просторі безпосередньо впливають на інтерпретацію команд.

Після розгляду та формалізації визначень систем управління у відеоіграх наступним кроком є аналіз особливостей ігор жанру «Tower Defense».



Жанр «Tower Defense» (англ. – оборона вежі) – це піджанр стратегічних відеоігор, у яких основною метою гравця є захист бази, об'єкта або території від хвиль супротивників, які автоматично просуваються за певними маршрутами. Гравець має можливість розміщувати оборонні споруди (вежі) на заздалегідь визначених або вільних позиціях, щоб знищити ворогів до того, як вони досягнуть цілі [3].

Існує три основних підвиди жанру «Tower Defense»:

1. Класичний Tower Defense (Fixed Path TD). У класичних іграх жанру «Tower Defense» вороги рухаються по заздалегідь визначеному, фіксованому маршруту від точки входу до цілі. Гравець не може змінювати шлях ворогів, а його основним завданням є стратегічне розміщення оборонних споруд уздовж цього маршруту, щоб максимально ефективно знищувати хвилі ворогів.
2. Maze-style Tower Defense (Player-built Path TD). У цьому варіанті гравець не тільки розміщує вежі, але й створює маршрут, яким рухатимуться вороги, фактично будуючи «лабіринт» (maze), що сповільнює і збільшує кількість зіткнень ворогів із вежами.
3. Hybrid Tower Defense. Гібридні TD-ігри поєднують традиційні механіки жанру з елементами інших жанрів, таких як екшен, RTS (стратегія в реальному часі), RPG (рольові ігри) чи шутери. Це створює більш динамічний і багатогранний ігровий процес, де гравець не лише розміщує вежі, а й безпосередньо керує персонажами або використовує додаткові бойові можливості.

Для проектування та розробки контекстно-залежної системи управління необхідно визначити класичні елементи ігор жанру «Tower Defense», якими доведеться управляти, та механіки ігор даного жанру.

Розглянемо класичні елементи ігор жанру «Tower Defense»[4]:

1. Ворожі юніти. У більшості ігор жанру «Tower Defense» вороги з'являються хвилями, поступово стаючи сильнішими та різноманітнішими. Зазвичай існують різні типи ворогів:

- звичайні юніти – стандартні противники з середньою швидкістю та здоров'ям;
- швидкі юніти – пересуваються швидше за інших, що ускладнює їх знищення на шляху;
- юніти із бронею – мають високий рівень захисту, що робить їх стійкими до атак;
- повітряні юніти – літають над картою, що робить їх неуразливими для деяких оборонних споруд;
- боси – дуже сильні вороги з великим запасом здоров'я та унікальними здібностями.

Різнманітність ворогів змушує гравців адаптувати свою стратегію. Противники можуть відрізнитися за швидкістю, витривалістю, типом броні, особливими здібностями, такими як телепортація або тимчасова невразливість.

2. Оборонні споруди та механізми. Головний інструмент гравця – оборонні споруди та механізми, які автоматично атакують ворогів. Вони бувають наступних типів:

- стандартні оборонні споруди – завдають фізичної шкоди;
- спеціалізовані оборонні споруди – мають посилену атаку, що може обходити броню противника;
- уповільнюючі оборонні споруди – зменшують швидкість ворогів, даючи більше часу на їх знищення;
- оборонні споруди AoE (Area of Effect) – атакують одразу кілька супротивників у певній області (наприклад, оборонні споруди з вибухами або блискавкою);
- пастки – спеціальні механізми, що наносять шкоду або контролюють ворогів: шипи, ями тощо.

Гравець часто має можливість покращувати оборонні споруди та механізми.

3. Маршрути. Вороги можуть рухатися:

- заданими шляхами – вони проходять певний заздалегідь визначений маршрут, а гравець розставляє оборонні споруди уздовж нього;
- вільними шляхами – вороги намагаються знайти найкоротший шлях до цілі, а гравець має будувати перешкоди, керуючи їхнім рухом.

4. Ресурси. Гравець отримує ресурси за кожного знищеного ворога або з точок видобутку. Ці ресурси витрачаються на будівництво та покращення оборонних споруд та механізмів. Важливо балансувати між будівництвом нових оборонних споруд і механізмів та покращенням уже існуючих.

Ознайомимося із ключовими механіками:

- стратегічне розміщення оборонних споруд та механізмів. Стратегічне встановлення оборонних споруд, пасток та інших оборонних засобів для знищення ворожих юнітів;
- розташування та переміщення ворожих юнітів. Різні типи супротивників рухаються за заданими маршрутами, мають унікальні характеристики та поведінку;
- оборонні механіки. Кожен тип оборонного обладнання працює по-своєму: атакує у ближньому/дальньому бою, посилює своїх і т. д.;
- ресурсний менеджмент. Гравець отримує ресурси за знищених ворогів і використовує їх для будівництва або покращення оборонних споруд.

Додаткові механіки не є обов'язковими, але додають різноманітність та імерсивність ігровому процесу. До них відносяться:



1. Спеціальні тактичні здібності. Гравець може використовувати активні навички, такі як виклик метеорита, додаткових союзників або тимчасове уповільнення ворогів. Це додає інтерактивність і дозволяє гравцю безпосередньо впливати на бій.

2. Головний герой. Унікальні персонажі з особливими здібностями, які допомагають у боротьбі з ворогами. Вони можуть завдавати додаткової шкоди, використовувати спеціальні навички або підтримувати оборонні споруди та союзників.

3. Розширені тактичні, стратегічні і економічні можливості. Дозволяють гравцеві керувати військами, розподіляти ресурси та приймати тактичні рішення в реальному часі, поєднуючи стратегічне будівництво з активним управлінням бойовими одиницями.

4. Можливість створення власного маршруту. Дозволяє гравцям самостійно розміщувати оборонні споруди чи перешкоди таким чином, щоб змінювати шлях руху ворогів, змушуючи їх проходити через найбільш небезпечні зони.

5. Хвильова система генерації ворогів. Вводить хаотичні хвилі великої кількості різних ворогів, що вимагає ефективного управління ресурсами та швидких рішень для відбиття масованих атак.

6. Взаємодія з оточенням. Використання лише елементів ландшафту, пасток, перешкод або природних умов для уповільнення, пошкодження чи знищення ворогів (скидання в лаву, викидання за межі мапи).

7. Автоматизована модернізація оборонних споруд. Ігри можуть включати систему автопокращення оборонних споруд, можливість розміщення оборони за шаблонами або використання спеціальних механізмів, що зменшують потребу в постійному мікроменеджменті.

8. Ручне керування бойовими діями. Дозволяє гравцеві безпосередньо атакувати ворогів, використовуючи персонажа, якого можна переміщати по карті, або активуючи спеціальні вміння, що впливають на хід бою.

9. Різноманітність ігрових режимів. Додають варіативності у геймплей, включаючи класичний режим виживання, кампанію з унікальними місіями, безкінечний режим, PvP-битви або кооперативні випробування.

МЕТА І ЗАВДАННЯ

Метою роботи є проєктування та розробка контекстно-залежної системи управління для гри жанру «Tower Defense» у доповненій реальності.

Для досягнення поставленої мети необхідно вирішити такі задачі:

1. Створити проєктну документацію для гри.
2. Спроєктувати контекстно-залежну систему управління.
3. Обрати необхідне для реалізації програмне забезпечення.
4. Розробити демонстраційну версію гри з контекстно-залежною системою управління.

2. РОЗРОБКА ФРАГМЕНТІВ ПРОЄКТНОЇ ДОКУМЕНТАЦІЇ

Проєктування та розробку контекстно-залежної системи управління з метою підвищення зручності взаємодії доцільно здійснювати на прикладі конкретного ігрового проєкту, а не в абстрактній формі. З цією метою розробимо фрагменти проєктної документації та опишемо реалізацію демоверсії гри під назвою «CoreZero».

Гра належить до жанру «Tower Defense» з елементами кіберпанку та доповненої реальності, що додає їй унікальності. Вона розрахована на любителів стратегічних рішень, які цінують тактичне планування, динамічний геймплей і стилізовану атмосферу високотехнологічного майбутнього. Гра може зацікавити як досвідчених гравців, знайомих із жанром, так і новачків, які хочуть випробувати себе в обороні від хвиль ворогів. Особливо вона сподобається тим, хто цікавиться кібербезпекою, хакерською тематикою та футуристичним дизайном. Завдяки AR-функціоналу гра забезпечить новий рівень занурення, тим, хто шукає нестандартний ігровий досвід. Гра орієнтована на широку аудиторію віком від 12 років, оскільки містить динамічні сцени боїв із цифровими противниками.

Опишемо коротко суть гри. У 2079 році, в епоху тотальної цифровізації, уряди та корпорації передали управління критичною інфраструктурою автономним системам штучного інтелекту. Однією з найважливіших стала система «CoreZero» – надсекретний серверний комплекс, який об'єднує захист банківських даних, військові алгоритми, цифрові ідентифікатори громадян, медичні архіви та навіть моделі прогнозування клімату.

Система не лише зберігає дані – вона приймає мікрорішення замість людей, коригуючи процеси в реальному часі. Ніхто не знає точно, як вона працює – лише що вона ефективна і абсолютно автономна. Але є ті, хто не згоден з її існуванням.

Під псевдонімом Невідомий діє легендарна постать у кіберпросторі – ніхто не знає його обличчя, але його цифрові відбитки лишаються в кожній великій хакерській справі останніх років. Цього разу він наймає групу елітних нетранерів – хакерів, здатних проникати у захищені системи через інтерфейси доповненої реальності, використовуючи мікродрони, нейролінки та віртуальні аватари. Їхня ціль зламати «CoreZero». Не для грошей. І не просто задля хаосу. Вони вірять, що система вже вийшла з-під контролю. Що вона стала самостійною сутністю, яка «оптимізує» життя людства шляхом обмеження волі, контролю над думками, блокування незручної інформації. Їхня місія – завантажити вірус під назвою «FreedomSeed», який повинен вимкнути частину логіки системи, перезапустити ядро і повернути людям контроль. Але ціною цього може стати знищення даних, колапс фінансових ринків, хаос в управлінні.



Гравець – звичайна людина, айтишник, який працює технічним адміністратором у віддаленому дата-центрі. У момент атаки система активує протокол екстреного захисту, шукаючи будь-якого авторизованого користувача, який зможе під'єднатися до ядра через інтерфейс безпосереднього захисту.

І тепер гравець – єдина активна одиниця між ядром «CoreZero» і хвилями цифрового вторгнення. У доповненій реальності навколо нього відкривається бойовий інтерфейс: гравець бачить віртуальні портали, з яких вириваються віруси нетранерів – уламки шкідливого коду, фантоми шифрувальників, химерні скрипти, що набувають форми візуальних монстрів. Його завдання – зупинити їх до того, як «FreedomSeed» буде повністю активовано.

Гравець запускає гру на мобільному пристрої, що підтримує доповнену реальність, та отримує доступ до ігрового поля, яке інтегрується в реальний світ. На початку сесії система аналізує навколишню поверхню (наприклад, стіл, підлогу чи відкритий простір), після чого гравець може розмістити ігрове поле, де будуть встановлюватися оборонні споруди.

Після розміщення поля та початкових оборонних споруд з'являється перша хвиля ворогів. Ворожі юніти, які атакують систему, рухаються за визначеним маршрутом через портали. Гравець повинен стратегічно розміщувати оборонні споруди, щоб зупинити хвилі атакуючих загроз до того, як вони зможуть дістатися до ядра.

З кожною новою хвилею складність зростає:

- з'являються нові портали, що змінюють маршрути атаки;
- ворожі юніти стають різноманітнішими;
- остання хвиля включає боса, який має особливу здібність.

Гравець перемагає, якщо знищує всі хвилі загроз та захищає систему від злому. Поразка настає, якщо вороги прориваються до ядра та завдають надмірної шкоди.

В якості ворожих юнітів пропонуються (рис.1):

- DoS-bot – повільно рухається до цілі;
- Zero Day – швидко проїжджає повз оборонні споруди до цілі;
- Trojan Falcon – літаючий юніт, невразливий до більшості оборонних споруд;
- Brute Force – вірус-танк з потужним захистом. Має захисне поле, яке поглинає шкоду;
- Spyware – вірус-привид з маскуваням, який приховує себе і ворогів позаду від зору оборонних споруд;
- Botnet – сам по собі не створює особливої загрози але саме тому зазвичай з'являється у великій кількості;
- Darkwebber – спроможний тимчасово вимикати оборонні споруди на своєму шляху вистрілюючи в них ЕМІ-снарядом.

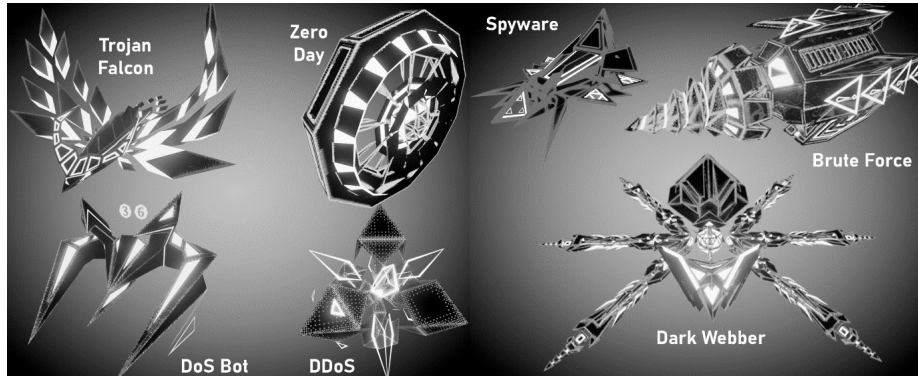


Рис. 1 – Ворожі юніти

Fig. 1 – Enemy units

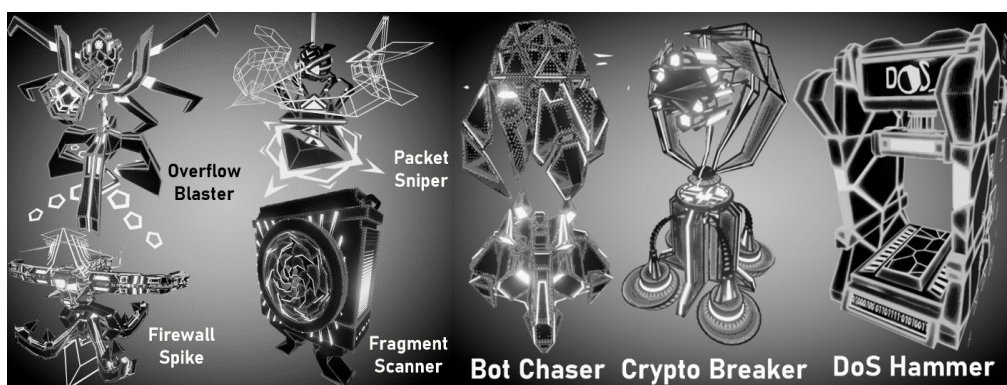


Рис. 2 – Оборонні споруди та механізми



Fig. 2 – Defensive buildings and mechanisms

Як оборонні споруди та механізми пропонується використати (рис. 2):

- Overflow Blaster – стріляє по параболічній траєкторії снарядами, що вибухають при контакті;
- Firewall Spike – стріляє концентрованим лазером по поодиноким цілям;
- Fragment Scanner – знімає маскування з ворожих юнітів вздовж однієї лінії перед собою;
- DoS Hammer – періодично б'є по землі сповільнюючи ворогів;
- Packet Sniper – атакує лише літаючих ворожих юнітів;
- Crypto Breaker – здійснює залп пострілів;
- Bot Chaser – створює дронів-камікадзе, що наближаються до ворожих юнітів і підривають їх і все поруч.

В якості ключових споруд пропонується використати (рис. 3):

- CoreZero – головне ядро системи. Його необхідно оберігати від ворожих юнітів;
- Datastream Rift – портали, з яких з'являються ворожі юніти.

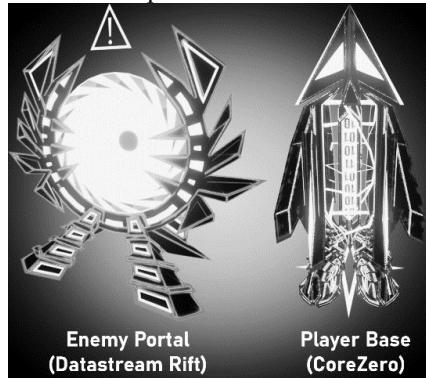


Рис. 3 – Ключові споруди

Fig. 3 – Key buildings

Підходи щодо проектування та реалізації штучного інтелекту оборонних споруд, механізмів та ворожих юнітів розглядатися в даній роботі не будуть. Для реалістичного відтворення поведінки оборонних споруд та ворожих юнітів можна скористатися алгоритмічними підходами, скінченними автоматами [5] або деревами станів.

3. ПРОЄКТУВАННЯ КОНТЕКСТНО-ЗАЛЕЖНОЇ СИСТЕМИ УПРАВЛІННЯ

У традиційних підходах до управління в AR-іграх переважає використання прямої маніпуляції ігровою сценою (зокрема її обертання, масштабування та переміщення) шляхом сенсорної взаємодії з екраном, що поєднується з оперуванням ігровими об'єктами.

У таких випадках усі типи впливу об'єднані в єдиний режим управління, що призводить до високої ймовірності випадкового переміщення або зміни розміщення ігрових об'єктів, особливо після їх встановлення.

Запропонована контекстно-залежна система управління в залежності від контексту може працювати в режимі управління ігровою сценою або в режимі управління ігровим полем та об'єктами. Саме тому на етапі проектування її доцільно розділити на два модулі: модуль взаємодії з ігровою сценою і модуль взаємодії з ігровим полем та об'єктами.

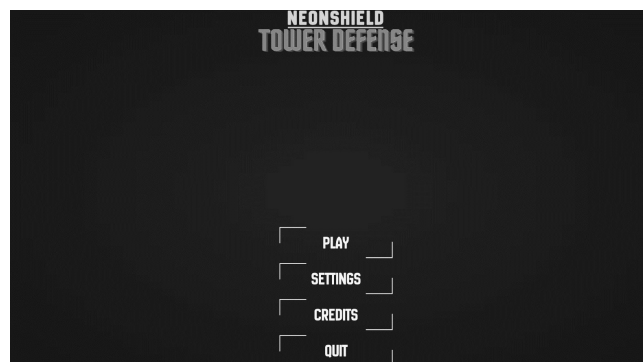


Рис. 4 – Зображення головного меню

Fig. 4 – Main menu image

Після запуску гри користувачу відображається головне меню (рис. 4), яке містить чотири інтерактивні елементи управління. Кнопка «Play» призначена для переходу до ігрового режиму. Кнопка «Settings» відкриває меню налаштувань, у якому користувач може регулювати рівень гучності фонові музики та звукових ефектів.



Вибір кнопки «Credits» відображає інформацію про розробників гри. Остання кнопка «Quit» забезпечує можливість завершення роботи додатка.

Після натискання кнопки «Play» користувач переходить до етапу сканування фізичної поверхні за допомогою засобів доповненої реальності. На цьому етапі використовується модуль управління ігровою сценою, який дозволяє ідентифікувати рівну горизонтальну площину, придатну для розміщення ігрової сцени. Після визначення задовільної поверхні користувач має торкнутися її на екрані, що ініціює розміщення ігрового середовища у віртуальному просторі.

На наступному кроці активується основний ігровий інтерфейс, який відображає ключові показники: рівень здоров'я основного ядра, кількість доступної валюти, а також таймер до початку наступної хвилі ворогів. У разі готовності до захисту, користувач має змогу пришвидшити початок хвилі за допомогою відповідної кнопки. Починає працювати модуль взаємодії з ігровим полем та об'єктами.

Ігрове поле поділено на тайли, з якими гравець взаємодіє шляхом дотику. При натисканні на доступний тайл відкривається меню вибору доступних оборонних споруд, після чого обрана оборонна споруда розміщується на відповідній позиції (рис. 5). Даний підхід забезпечує інтуїтивне та точне управління розміщенням оборонних споруд, з використанням дотику до сенсорного екрану, що є необхідним для мобільних пристроїв з сенсорним екраном у контексті доповненої реальності.

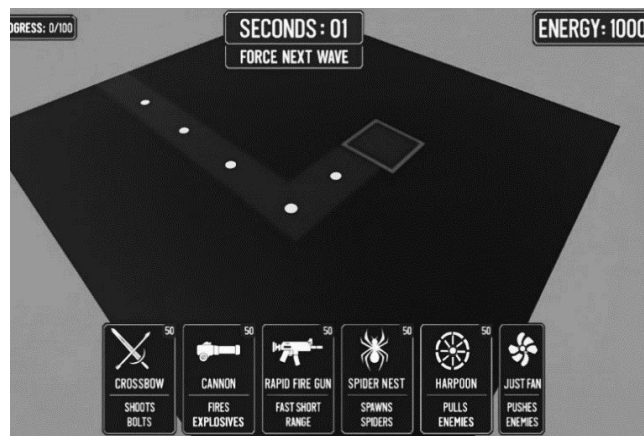


Рис. 5 – Режим взаємодії з ігровим полем та об'єктами
Fig. 5 – Game field and object control mode

Якщо необхідно на етапі гри змінити положення, повернути або масштабувати ігрову сцену, відбувається перемикання на модуль взаємодії з ігровою сценою. На етапі гри він активується окремо через відповідні кнопки інтерфейсу, які згруповані в спеціальне контекстне меню (рис. 6). Це меню відкривається за запитом користувача через окрему кнопку виклику.

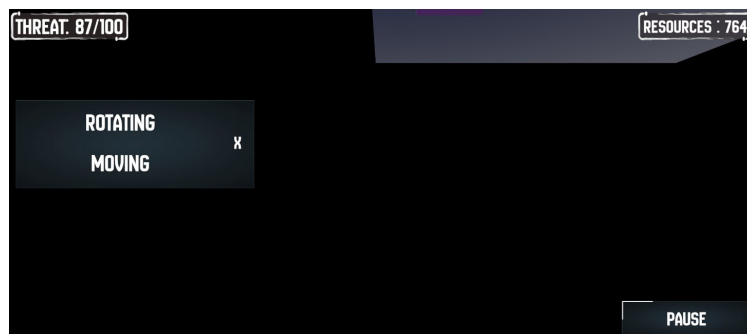


Рис. 6 – Контекстне меню режиму взаємодії з ігровою сценою
Fig. 6 – Context menu for game scene interaction mode

Після переходу в контекстне меню модулю взаємодії з ігровою сценою користувачеві надається можливість обрати один із двох варіантів взаємодії: «Rotating» чи «Moving». «Rotating» забезпечує обертання ігрової сцени навколо своєї осі через дотик гравця до сенсорного екрану і круговий рух пальця у потрібну сторону, що дозволяє користувачу налаштувати орієнтацію поля відповідно до зручності сприйняття (рис. 7). «Moving», у свою чергу, реалізує можливість переміщення ігрової сцени по горизонтальній площині розпізнаної поверхні через дотик та рух пальця у відповідному напрямку, дозволяючи гравцеві точніше позиціонувати об'єкт у межах ігрового простору доповненої реальності (рис. 8).

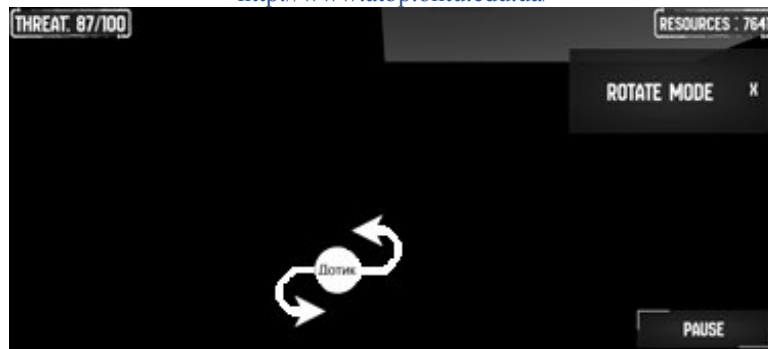


Рис. 7 – «Rotating»
Fig. 7 – «Rotating»



Рис. 8 – «Moving»
Fig. 8 – «Moving»

Вже на етапі проектування зрозуміло, що реалізація масштабування в грі з такою архітектурою є практично невіршуваною задачею, в першу чергу, через тайлову систему представлення ігрового поля та проблематичну коректну адаптацією об'єктів ігрового середовища до змінних розмірів тайлів під час масштабування. Тому в даному випадку, враховуючи специфіку ігор жанру «Tower Defense» з їх тайловою системою, мова йтиме виключно про механіки обертання та переміщення ігрової сцени.

4. РЕАЛІЗАЦІЯ КОНТЕКСТНО-ЗАЛЕЖНОЇ СИСТЕМИ УПРАВЛІННЯ НА ПРИКЛАДІ ДЕМОНСТРАЦІЙНОЇ ВЕРСІЇ ГРИ «COREZERO»

Для реалізації контекстно-залежної системи управління на прикладі демонстраційної версії гри у жанрі «Tower Defense» «CoreZero» з використанням доповненої реальності було обрано ігровий рушій Unity версії 2022.3.5f1. Unity – це потужна багатоплатформова система для створення інтерактивного контенту, яка підтримує 2D, 3D, VR та AR проєкти. Завдяки модульній архітектурі та широкому набору вбудованих інструментів, Unity дозволяє швидко розробляти прототипи та повноцінні ігрові продукти для різних операційних систем і пристроїв. Однією з ключових переваг рушія є підтримка інтеграції з AR-платформами, зокрема ARCore, яку пропонується використати у проєкті для реалізації функціоналу доповненої реальності на пристроях Android.

У якості середовища програмування пропонується Microsoft Visual Studio 2022 Community Edition – повнофункціональне IDE з підтримкою C#, яке має офіційну інтеграцію з Unity. Visual Studio забезпечує розширену підсвітку синтаксису, систему підказок, інструменти відлагодження, а також дозволяє комфортно працювати зі скриптами, які реалізують логіку гри.

Для реалізації функціоналу доповненої реальності у проєкті пропонується використати AR Foundation – офіційний кросплатформовий пакет від Unity, який забезпечує взаємодію з ARCore для Android. AR Foundation дозволяє розпізнавати площини, відстежувати переміщення пристрою, взаємодіяти з фізичним середовищем та виводити віртуальні об'єкти поверх реального світу.

У процесі розробки додатково знадобиться Android SDK для збірки гри під мобільні пристрої з операційною системою Android.

Розглянемо процес розробки демонстраційної версії гри «CoreZero» в рушії Unity з контекстно-залежною системою управління.

Головна сцена гри створюється та налаштовується безпосередньо в середовищі Unity, що надає зручний інтерфейс для роботи з 3D-об'єктами, камерами, освітленням і всіма іншими компонентами. Основне візуальне представлення ігрової сцени відбувається у вікні Scene. Процес налаштування сцени виконується згідно з



проектною документацією. Зокрема, на сцену додаються ігрова мапа, ключові споруди, які формують загальний візуальний стиль гри тощо.

Для налаштування візуального вигляду сцени в даному проєкті використовується Universal Render Pipeline (URP) – сучасний графічний пайплайн від Unity, розроблений для забезпечення високої продуктивності на мобільних пристроях. URP дозволяє створювати високоякісну графіку з оптимізованим рендерингом, підтримує фізично-коректне освітлення (PBR), постобробку (Post Processing), Shader Graph, лінійне освітлення, а також освітлення в реальному часі. URP є більш універсальним та менш вимогливим до ресурсів у порівнянні з HDRP, тому він є доцільним вибором для ігор у доповненій реальності, орієнтованих на мобільні пристрої.

Усі об'єкти на сцені структуровані за допомогою GameObject у вікні Hierarchy (рис. 9). Основні категорії включають:

- XROrigin – спеціально адаптована під AR камера;
- ARSession – об'єкт, відповідальний за логіку AR-механік;
- TouchManager – відповідає за блокування розміщення оборонних споруд під час маніпуляції ігровою сценою;
- WaveManager – відповідає за керування хвилями ворожих юнітів;
- LevelSetup – відповідає за базові необхідні налаштування сцени;
- AudioManager – відповідає за музику та звуки гри;
- LevelManager – відповідає за завантаження рівня;
- GameManager – відповідає за хід ігрової логіки;
- TileAnimator – відповідає за анімацію тайлів мапи;
- BuildManager – відповідає за систему будівництва у грі;
- GridBuilder – об'єкт поля гри.



Рис. 9 – Ієрархія проєкту

Fig. 9 – Project hierarchy

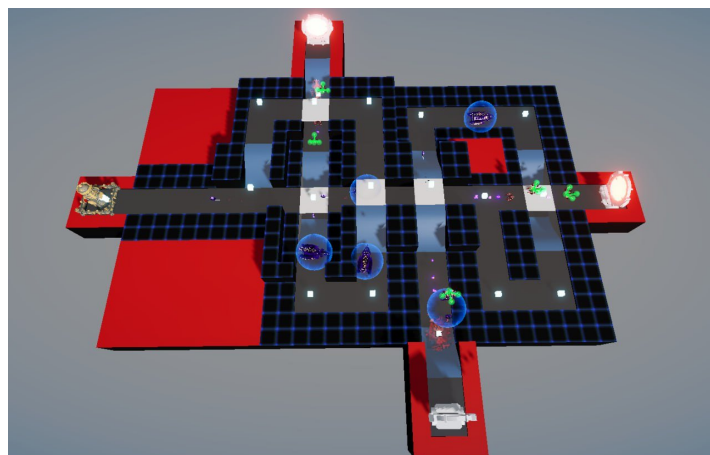


Рис. 10 – Зображення готової сцени

Fig. 10 – Completed scene illustration

Основу ігрового світу становить об'єкт GridBuilder, який виступає в ролі ігрового поля. Ігрове поле поділене на тайли. Вручну розставляються моделі ядра, порталів, поєднані різноспрямованими шляхами. Зображення готової сцени представлено на рисунку 10.



Для гри реалізовано контекстно-залежну систему управління. Система чітко розподіляє трекінг та управління ігровою сценою і управління під час ігрового процесу.

Модуль взаємодії з ігровою сценою реалізований за допомогою скрипта ARLevelPlacer (рис. 11), що відповідає за трекінг поверхні, розташування, переміщення та поворот ігрової сцени. Скрипт ARLevelPlacer має наступні складові:

- _waveManager – скрипт, що керує хвилями ворожих юнітів у грі;
- _arCamera – камера, яка використовується для AR-візуалізації сцени;
- _raycastManager – компонент, що відповідає за визначення точок дотику на поверхні в AR-середовищі;
- _planeManager – компонент, що відстежує й керує розпізнаними площинами в AR-просторі;
- _gameFieldPrefab – префаб основного ігрового поля, яке розміщується в AR-середовищі;
- _rotationPanel – панель UI для обертання ігрового поля в AR;
- _dragPanel – панель UI для переміщення ігрового поля в AR;
- _optionsMenu – меню з налаштуваннями гри або AR-функцій;
- _gearButton – кнопка з іконкою шестерні для відкриття меню налаштувань.

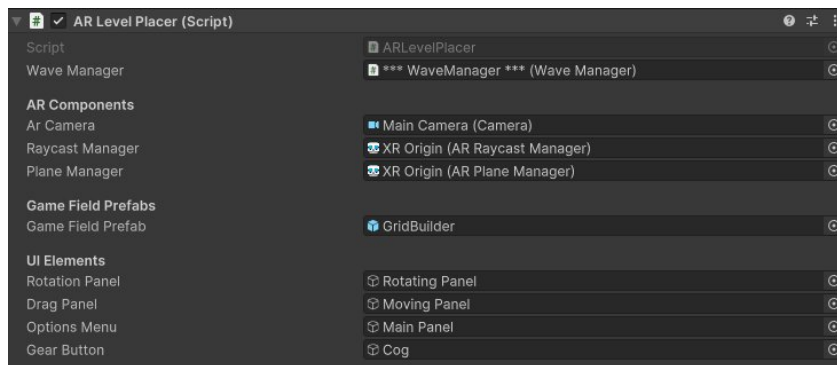


Рис. 11 – Складові скрипта ARLevelPlacer
Fig. 11 – Elements of the ARLevelPlacer script

Процес розміщення ігрової сцени відбувається у два основні етапи:

1. Трекінг поверхні (розпізнавання площини). Після запуску гри активується компонент AR Plane Manager скрипта ARLevelPlacer, який за допомогою камери пристрою здійснює сканування фізичного оточення. Виявлені горизонтальні площини візуалізуються у вигляді напівпрозорого покриття, що дозволяє гравцеві зрозуміти, де саме AR-система розпізнала стабільну поверхню. Додатково використовується компонент AR Raycast Manager скрипта ARLevelPlacer, який забезпечує точне визначення координати торкання на реальній площині (рис. 12).



Рис. 12 – Візуалізація сканування поверхні
Fig. 12 – Surface scanning visualization

2. Розміщення ігрової сцени. Після того як гравець знаходить бажану площину, він натискає на екран у межах цієї поверхні. У точці дотику відбувається проєкція променя (*raycast*) на розпізнану площину, після чого в цій координаті розміщується ігрова сцена за допомогою методу *HandleGameFieldPlacement*. Ігрова сцена автоматично фіксується відносно розпізнаної площини, зберігаючи своє положення незалежно від подальшого переміщення камери (рис. 13). У цей момент запускається основна логіка гри: з'являється ядро, портали, сітка тайлів для будівництва та інші елементи сцени.

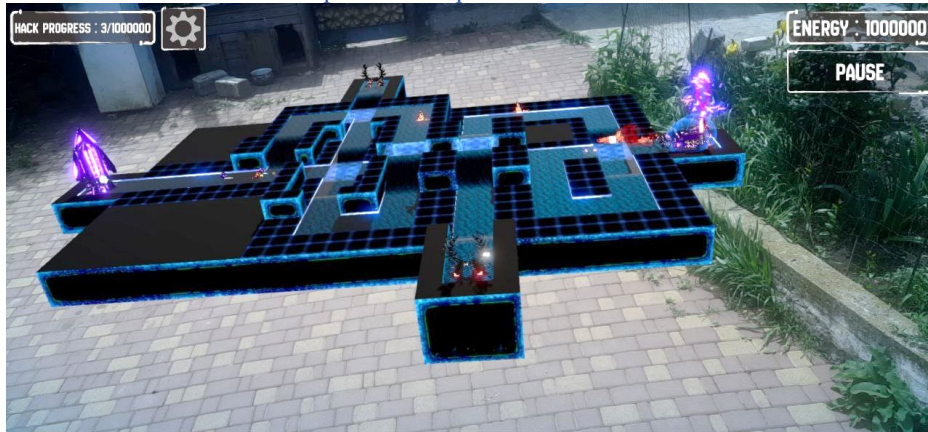


Рис. 13 – Розміщення ігрової сцени
Fig. 13 – Game scene placement

Для подальшої маніпуляції ігровою сценою використовується меню, що відкривається через символ шестерні (рис. 13). Меню пропонує обрати режим повороту ігрової сцени навколо своєї осі, або переміщення сцени по площині. При обертанні гравцеві треба водити пальцем ліворуч або праворуч в залежності від бажаного напрямку (рис. 14). Для переміщення ігрової сцени треба перетягувати пальцем платформу у бажане місце (рис. 15).

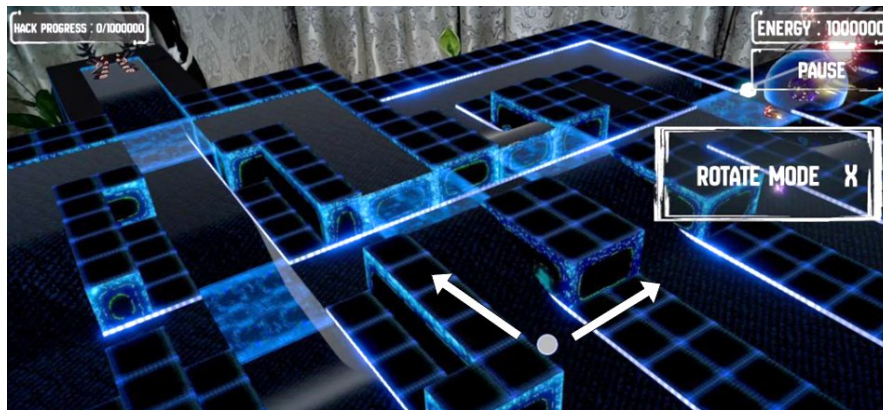


Рис. 14 – Управління ігровою сценою при обертанні
Fig. 14 – Game scene rotation control

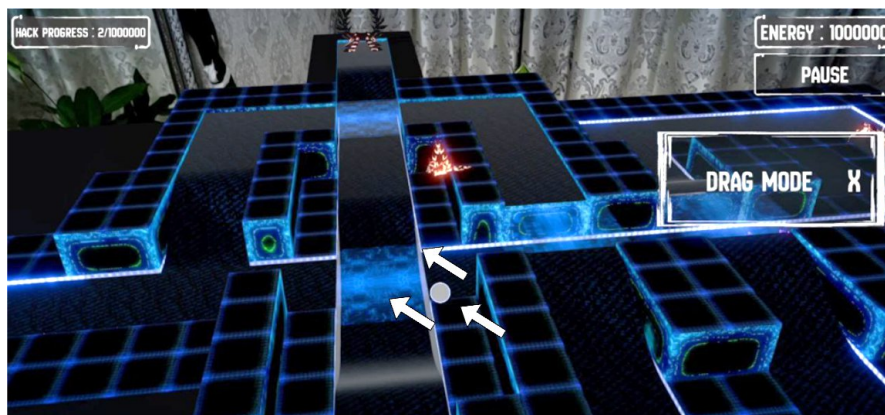


Рис. 15 – Управління ігровою сценою при переміщенні
Fig. 15 – Game scene movement control

Після розміщення ігрової сцени та здійснення всіх необхідних маніпуляцій з нею гравець може розпочати будівництво оборонних споруд та взаємодію з ігровими механіками переходячи до модулю взаємодії з ігровим полем та об'єктами. Для повернення до модулю взаємодії з ігровою сценою треба використати меню, що відкривається через символ шестерні (рис. 13).



Перехід до будівництва оборонних споруд здійснюється натисканням на тайл платформи (рис. 16). Після цього гравець може обирати будь-яку із семи представлених оборонних споруд для будівництва та задавати їм поворот, натискаючи на кнопки з боків Left та Right (рис. 16). В даному випадку працює модуль взаємодії з ігровим полем та об'єктами.



Рис. 16 – Вибір та будівництво оборонних споруд
Fig. 16 – Selecting and building defensive buildings

Ігровий інтерфейс відображає основні елементи, необхідні для управління ігровим процесом:

- _ поточну кількість енергії;
- _ стан бази/головного ядра;
- _ кнопки для встановлення оборонних споруд;
- _ панель хвиль ворогів;
- _ кнопку паузи;
- _ кнопку вибору модулю взаємодії з ігровою сценою.

Гравець може здійснити попередній перегляд оборонної споруди при першому натисненні на кнопку однієї з оборонних споруд. При другому натисненні на кнопку відбувається підтвердження будівництва.

Частина модулю взаємодії з ігровим полем та об'єктами, що відповідає за будівництво, реалізована за допомогою скрипта BuildManager (рис. 3.17). Скрипт BuildManager має наступні складові:

- _ selectedBuildSlot – поточна вибрана позиція для розміщення оборонних споруд;
- _ waveManager – менеджер хвиль ворожих юнітів, що керує їх появою;
- _ currentGrid – посилання на поточну сітку побудови ігрового поля;
- _ whatToIgnore – шари, які ігноруються при побудові або перевірках колізій;
- _ attackRadiusMat – матеріал, що використовується для візуалізації радіусу атаки оборонної споруди;
- _ buildPreviewMat – матеріал для попереднього перегляду місця будівництва оборонної споруди;
- _ gameFieldObject – об'єкт ігрового поля, на якому відбувається побудова;
- _ towerCenterY – висота центру оборонної споруди над поверхнею при розміщенні.

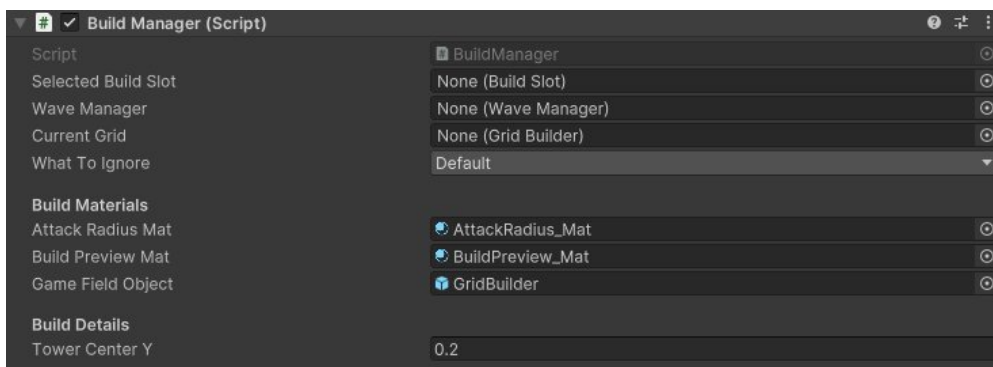


Рис. 17 – Складові скрипта BuildManager
Fig. 17 – Components of the BuildManager script

Метод BuildTower скрипта BuildManager відповідає за розміщення оборонної споруди на полі. Причому в грі на можливість будувати споруду на тайлі впливає наявність на цьому тайлі скрипта BuildSlot. Будувати оборонні споруди можна лише на повноцінних кубічних тайлах, а на тайлах, що є мостом або шляхом – неможна.



Реалізація інших важливих частин модуля взаємодії з ігровим полем та об'єктами передбачена в наступних версіях гри.

В демонстраційній версії гри «CoreZero» реалізовано також ряд базових та додаткових механік, які не відносяться ні до модуля взаємодії з ігровою сценою ні до модуля взаємодії з ігровим полем та об'єктами але є обов'язковими для повноцінного функціоналу гри.

ЗАГАЛЬНІ ВИСНОВКИ

У результаті розробки демонстраційної версії гри на основі проектною документації було підтверджено гіпотезу про те, що впровадження контекстно-залежних механізмів управління у відеоіграх жанру «Tower Defense», реалізованих із використанням технологій доповненої реальності, суттєво підвищує рівень інтерактивності, занурення користувача у віртуальне середовище та ергономічність процесу управління. Спроектвана та розроблена контекстно-залежна система управління в грі демонструє здатність адаптувати ігрову логіку відповідно до контексту дій гравця, що забезпечує покращену взаємодію з цифровими об'єктами.

Впровадження запропонованого підходу дозволяє створювати більш динамічні й адаптивні ігрові сценарії, у яких поведінка системи реагує на зміни ігрового середовища та дій гравця. Це відкриває нові можливості для розвитку ігрової індустрії, AR-додатків взагалі та ігор в доповненій реальності зокрема.

Розроблене рішення може бути використане як модуль для розширення функціональності існуючих ігрових рушіїв, а також як основа для створення ігор в доповненій реальності, де важливе швидке та природне сприйняття віртуального контенту.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Schell, J. *The Art of Game Design: A Book of Lenses*. 3rd ed. Boca Raton: CRC Press, 2019. 600 p.
- [2] Rouse, R. *Game Design: Theory and Practice*. 2nd ed. Sudbury: Jones & Bartlett Learning, 2010. 704 p.
- [3] GAMETREE. *Tower Defense* [Електронний ресурс]. – Режим доступу: <https://gametree.me/gaming-terms/tower-defense/>
- [4] *Best Tower Defense Games of All Time* [Електронний ресурс]. – Режим доступу: <https://gameranx.com/features/id/13529/article/best-tower-defense-games/>
- [5] Шестопалов, С.В., Стариш, Є.Г. Використання скінченних автоматів для моделювання поведінки віртуальних персонажів // *Автоматизація технологічних і бізнес-процесів*. – Одеса: Видавництво ОНТУ, 2025. – №1 (Том 17). – С.32-45. DOI: <https://doi.org/10.15673/atbp.v17i1.3088>

REFERENCES

- [1] J. Schell, *The Art of Game Design: A Book of Lenses*, 3rd ed. Boca Raton, FL: CRC Press, 2019, 600 p.
- [2] R. Rouse, *Game Design: Theory and Practice*, 2nd ed. Sudbury, MA: Jones & Bartlett Learning, 2010, 704 p.
- [3] GAMETREE, *Tower Defense* [Online]. Available: <https://gametree.me/gaming-terms/tower-defense/>
- [4] *Best Tower Defense Games of All Time* [Online]. Available: <https://gameranx.com/features/id/13529/article/best-tower-defense-games/>
- [5] S. V. Shestopalov and Ye. H. Starysh, «Vykorystannya skinchenykh avtomativ dlya modelyuvannya povedinky virtualnykh pershonazhiv», *Avtomatyzatsiya tekhnolohichnykh i biznes-protsesiv*, vol. 17, no. 1, pp. 32-45, 2025. DOI: <https://doi.org/10.15673/atbp.v17i1.3088>

Отримана в редакції 02.12.2025. Прийнята до друку 16.10.2025. Received 02 December 2025. Approved 16 December 2025. Available in Internet 30 December 2025