

О. В. СОБКО

Ph.D., старший викладач кафедри комп'ютерних наук
Хмельницький національний університет
ORCID: 0000-0001-5371-5788

М. О. МОЛЧАНОВА

Ph.D., старший викладач кафедри комп'ютерних наук
Хмельницький національний університет
ORCID: 0000-0001-9810-936X

О. В. МАЗУРЕЦЬ

кандидат технічних наук,
доцент кафедри комп'ютерних наук
Хмельницький національний університет
ORCID: 0000-0002-8900-0650

П. В. ГНАТЮК

студент кафедри комп'ютерних наук
Хмельницький національний університет

АЛГОРИТМІЗАЦІЯ ПРОЦЕСІВ КЕРУВАННЯ ТА ІНТЕРАКЦІЙ В ІГРОВИХ ОБ'ЄКТНО-ОРІЄНТОВАНИХ ЗАСТОСУНКАХ

У статті досліджується алгоритмізація процесів керування і інтеракцій в ігрових об'єктно-орієнтованих застосунках на прикладі гри «Гольф», розробленої з використанням мови програмування C# та ігрового рушія Unity. Розглянуто основні аспекти механізмів вибору рівнів, управління ігровим об'єктом, проходження рівнів та нарахування винагород у вигляді зірок, а також обробку ситуацій поразки. Вказано, що інтеграція взаємозалежних функцій у відповідні класи дозволяє забезпечити модульність, масштабованість і повторне використання коду, що є ключовим для складних ігрових систем. Створена діаграма класів відображає структуру зв'язків між основними та допоміжними класами, включно з MonoBehaviour, який забезпечує взаємодію скриптів із рушієм Unity через події Start(), Update(), OnCollisionEnter() та інші, а також користувацькі класи, що реалізують поведінку м'яча, управління сценами, відображення інтерфейсу, обробку колізій та ефекти, логіку ворогів та рух об'єктів на сцені.

У статті показано, що застосування запропонованих алгоритмів вибору рівнів, керування м'ячем, проходження рівнів та нарахування зірок, поразки дозволяє створити реалістичний та керований ігровий процес, а також забезпечує точність розрахунку траєкторій, сили поштовху та реакцій на зіткнення. Розроблені механіки можуть слугувати основою для подальшого проектування більш складних ігор із високим рівнем інтерактивності, динамічною фізичною моделлю та керованими елементами інтерфейсу. Запропоновані у статті алгоритми демонструють ефективність об'єктно-орієнтованого підходу в програмуванні ігор, що сприяє підвищенню продуктивності розробки та полегшує масштабування проекту, що є важливим під час розробки та супроводу програмного забезпечення.

Ключові слова: алгоритмізація, об'єктно-орієнтоване програмування, Unity, керування об'єктами, фізика ігрового процесу, діаграма класів.

O. V. SOBKO

Ph.D., Senior Lecturer at the Computer Science Department
Khmelnytskyi National University
ORCID: 0000-0001-5371-5788

M. O. MOLCHANOVA

Ph.D., Senior Lecturer at the Computer Science Department
Khmelnytskyi National University
ORCID: 0000-0001-9810-936X

O. V. MAZURETS

Ph.D. in Engineering Science,
Associate Professor at the Computer Science Department
Khmelnitskyi National University
ORCID: 0000-0002-8900-0650

P. V. HNATIUK

Bachelor Student at the Computer Science Department
Khmelnitskyi National University

ALGORITHMISATION OF CONTROL AND INTERACTION PROCESSES IN OBJECT-ORIENTED GAME APPLICATIONS

The article investigates the algorithmization of control processes and interactions in object-oriented game applications using the example of the game "Golf", developed using the C# programming language and the Unity game engine. The main aspects of the level selection mechanisms, game object management, level passing and star reward accrual, as well as handling defeat situations are considered. It is indicated that the integration of interdependent functions into the corresponding classes allows for modularity, scalability and code reuse, which is key for complex game systems. The created class diagram reflects the structure of relationships between the main and auxiliary classes, including MonoBehaviour, which provides script interaction with the Unity engine via Start(), Update(), OnCollisionEnter() and other events, as well as user classes that implement ball behavior, scene management, interface display, collision handling and effects, enemy logic and object movement on the scene.

The article shows that the application of the proposed algorithms for level selection, ball control, level progression, and star and defeat accrual allows for the creation of a realistic and controllable gameplay, as well as ensuring the accuracy of trajectories, thrust forces, and collision reactions. The developed mechanics can serve as the basis for further designing more complex games with a high level of interactivity, a dynamic physical model, and controllable interface elements. The algorithms proposed in the article demonstrate the effectiveness of an object-oriented approach in game programming, which helps to increase development productivity and facilitates project scaling, which is important during software development and maintenance.

Key words: algorithmization, object-oriented programming, Unity, object management, gameplay physics, class diagram.

Постановка проблеми

Сучасна індустрія комп'ютерних ігор та інтерактивних застосунків відзначається високим рівнем складності програмних систем [1], що обумовлює необхідність ефективного моделювання поведінки ігрових об'єктів та організації їх взаємодії [2]. Об'єктно-орієнтоване програмування забезпечує структурований підхід до розробки таких систем завдяки використанню принципів інкапсуляції, успадкування та поліморфізму [3, 4], що дозволяє створювати модульні, масштабовані та повторно використовувані компоненти [5]. Ключовим аспектом розробки об'єктно-орієнтованих ігрових застосунків є алгоритмізація процесів керування та інтеракцій [6, 7]. Ефективна алгоритмізація включає розробку алгоритмів обробки подій [8], колізій, керування станами об'єктів. Сучасні ігрові рушії, зокрема Unity, у поєднанні з мовою програмування C# надають широкий спектр інструментів для реалізації таких алгоритмів [9, 10]. Середовище Unity і C# також сприяють масштабованості системи та повторному використанню кодових модулів [11], що є важливим для підтримки та розвитку складних ігрових проєктів [12].

Водночас розробка ефективних алгоритмів керування та інтеракцій у об'єктно-орієнтованих ігрових системах залишається проблематичною [13] через необхідність одночасної інтеграції різних підходів до поведінки об'єктів, адаптації до дій користувача та оптимізації продуктивності. Неоптимізована логіка призводить до затримок, зниження частоти кадрів та погіршення користувацького досвіду [14], тоді як відсутність системного підходу ускладнює масштабування та повторне використання компонентів.

Аналіз останніх досліджень і публікацій

Сучасні дослідження в галузі розробки об'єктно-орієнтованих ігрових застосунків підтверджують ефективність поєднання об'єктно-орієнтованого програмування з алгоритмізацією процесів керування та інтеракцій. Ряд наукових робіт показує, що об'єктно-орієнтоване програмування дозволяє структуровано організовувати поведінку ігрових об'єктів у вигляді класів та модулів, що підвищує масштабованість системи та спрощує повторне використання коду [15].

Стаття [16] зосереджена на розробці високорівневої структури гри з використанням Unity та мови програмування C#. Автори детально описують принципи проєктування, побудову основних модулів та застосування реальних прикладів. Вони підкреслюють важливість організації коду для підвищення ефективності розробки та якості Unity-проєктів.

У роботі [17] розглядається використання Unity для навчання програмуванню ігор. Автор описує розробку шаблону гри RPG, який використовується як навчальний матеріал для студентів. Робота акцентує увагу на важливості використання об'єктно-орієнтованого підходу та алгоритмізації в процесі навчання програмуванню ігор.

У статті [18] представлено досвід розробки адаптивних ігор з анімованими персонажами в середовищі Unity. Автори описують процес створення інтерактивного середовища, яке дозволяє розробляти ігри з анімованими персонажами, що реагують на дії користувача.

Стаття [19] досліджує методи ефективної розробки мобільних ігор на сучасних пристроях, акцентуючи увагу на оптимізації коду, алгоритмізації процесів керування та інтеракцій, використанні об'єктно-орієнтованого підходу та повному використанні обчислювальних і графічних можливостей мобільних платформ для створення складних і адаптивних ігрових систем.

Аналіз наукових джерел показує, що розробка об'єктно-орієнтованих ігрових застосунків значною мірою залежить від алгоритмізації процесів керування, оскільки саме вона забезпечує узгодженість взаємодії об'єктів, оптимізацію ресурсів і підтримку масштабованості проекту.

Формулювання мети дослідження

Метою дослідження є проектування алгоритмів процесів керування та інтеракцій в ігрових об'єктно-орієнтованих застосунках з використанням C# і Unity на прикладі гри «Гольф».

Викладення основного матеріалу дослідження

Розробка сучасних ігрових застосунків вимагає не лише високоякісного візуального оформлення, а й чіткої організації логіки керування та взаємодії між об'єктами [20]. Для досягнення цього важливим є використання об'єктно-орієнтованого підходу, який забезпечує модульність, масштабованість та зручність супроводу програмного коду [21]. У межах цього дослідження здійснено проектування алгоритмів процесів керування та інтеракцій в ігрових застосунках із застосуванням мови програмування C# та ігрового рушія Unity. Як приклад реалізації обрано створення гри «Гольф», що дало змогу продемонструвати особливості побудови ігрової логіки та взаємодії об'єктів у рамках компонентної архітектури Unity.

Алгоритм вибору рівнів передбачає активацію панелі з кнопками для запуску окремих сцен після натискання кнопки Play та виклик панелі підтвердження виходу при натисканні Quit, що забезпечує керувану навігацію між рівнями та узгодженість станів інтерфейсу. Псевдокод алгоритму подано далі.

Алгоритм 1. Алгоритм вибору рівнів.

приНатисканніКнопки(кнопка):

```
якщо кнопка == «Play»:
    деактивувати(КнопкаPlay)
    деактивувати(КнопкаQuit)
    активувати(ПанельРівнів)
якщо кнопка == «Quit»:
    деактивувати(ГоловнеМенюUI)
    активувати(ПанельПідтвердженняВиходу)
```

приНатисканніКнопки(підтвердженняВиходу):

```
якщо підтвердженняВиходу == «Так»:
    завершитиПрограму()
якщо підтвердженняВиходу == «Ні»:
    деактивувати(ПанельПідтвердженняВиходу)
    активувати(ГоловнеМенюUI)
```

приНатисканніКнопки(кнопкаРівня):

```
якщо кнопкаРівня у КнопкахРівнів:
    завантажитиСцену(кнопкаРівня.індексСцени)
```

приНатисканніКнопки(закритиПанельРівнів):

```
деактивувати(ПанельРівнів)
активувати(КнопкаPlay)
активувати(КнопкаQuit)
```

Ігровий процес реалізовано через алгоритм керування м'ячем, який поєднує обробку натискань миші, візуалізацію прицілювання, симуляцію руху з урахуванням сили поштовху, поступове гальмування та обробку зіткнень. Потрапляння м'яча в лунку фіксує проходження рівня, а у разі інших зіткнень застосовується відбиття за фізичними законами. Псевдокод алгоритму подано далі.

Алгоритм 2. Алгоритм керування м'ячем.

Оновлення():

```
якщо натиснутаЛіваКнопкаМиші:
    попадання = променевийВистріл(позиціяМиші, шарМ'яча)
```

```

якщо попадання == м'яч:
    режимПрицілювання = true
    початковаПозиція = позиціяМиші

якщо режимПрицілювання і утримуєтьсяЛіваКнопкаМиші:
    векторТяги = початковаПозиція - позиціяМиші
    сила = обмежити(довжина(векторТяги), 0, максСила)
    показатиЛініюПрицілювання(сила, векторТяги)

якщо режимПрицілювання і відпущенаЛіваКнопкаМиші:
    м'яч.швидкість = нормалізувати(векторТяги) * сила
    режимПрицілювання = false
    приховатиЛініюПрицілювання()
    збільшитиЛічильникХодів()
    відтворитиЗвукУдару()

Фізика():
    якщо м'яч.рухається:
        застосуватиГальмування(м'яч)
        якщо м'яч.швидкість < мінШвидкість:
            м'яч.швидкість = 0
            м'яч.рухається = false

приКолізії(м'яч, об'єкт):
    якщо об'єкт.тег == «loonka»:
        зупинитиМ'яч()
        активуватиПеремогуРівня()
    інакше:
        відбити(м'яч.швидкість, нормальКолізії)

```

Для оцінювання результатів гравця створено алгоритм підрахунку ходів і нарахування зірок, що активується при досягненні лунки. У цей момент зупиняється внутрішній ігровий час і з'являється панель перемоги з опціями переходу до наступного рівня, перезапуску чи повернення в меню. Псевдокод алгоритму подано далі.

Алгоритм 3. Алгоритм проходження рівнів і нарахування зірок.

```

колиМ'ячДосягЛунки():
    зупинитиІгровийЧас()
    ходи = отриматиЛічильникХодів()
    зірки = обчислитиЗірки(ходи, пороговіЗначенняРівня)
    показатиПанельПеремоги(зірки)

приНатисканніКнопки(кнопкаПанеліПеремоги):
    якщо кнопка == «Next Level»:
        завантажитиСцену(індексПоточноїСцени + 1)
    якщо кнопка == «Again»:
        перезапуститиПоточнуСцену()
    якщо кнопка == «Main Menu»:
        завантажитиСцену(0)
    відновитиІгровийЧас()

```

Передбачено також алгоритм поразки: при зіткненні з небезпечними об'єктами активується панель повідомлення про невдачу, зупиняється ігровий час, а гравцеві пропонуються лише варіанти перезапуску або виходу до меню. Псевдокод алгоритму подано далі.

Алгоритм 4. Алгоритм поразки.

```

приКолізії(м'яч, об'єкт):
    якщо об'єкт.тег == «перешкода»:
        зупинитиІгровийЧас()
        показатиПанельПоразки()

приНатисканніКнопки(кнопкаПанеліПоразки):
    якщо кнопка == «Again»:

```

```

перезапуститиПоточнуСцену()
якщо кнопка == «Main Menu»:
    завантажитиСцену(0)
    відновитиІгровийЧас()

```

У процесі розробки створено діаграму класів (рисунк 1), що відображає основні зв'язки та функціональне призначення класів і їх методів [22].

У розробленому ігровому проєкті не було необхідності створювати єдиний головний клас, що об'єднує всю програму, оскільки цю функцію виконує вбудований у рушій Unity клас `MonoBehaviour`. `MonoBehaviour`, що належить до простору імен `UnityEngine`, забезпечує скриптам взаємодію з рушієм через автоматично викликані події, такі як `Start()`, `Update()`, `OnCollisionEnter()` та інші. Серед користувацьких класів основними є `movement` та `SceneManagerScript`. Клас `movement` відповідає за логіку поведінки м'яча на рівнях, включаючи надання імпульсу, розрахунок траєкторії, початкової швидкості, зміни швидкості, визначення нового положення, обробку відбиттів від поверхні, контакт із лункою та підрахунок кількості ударів. Оскільки ці функції взаємозалежні, їх інтеграція в одному класі є доцільною.

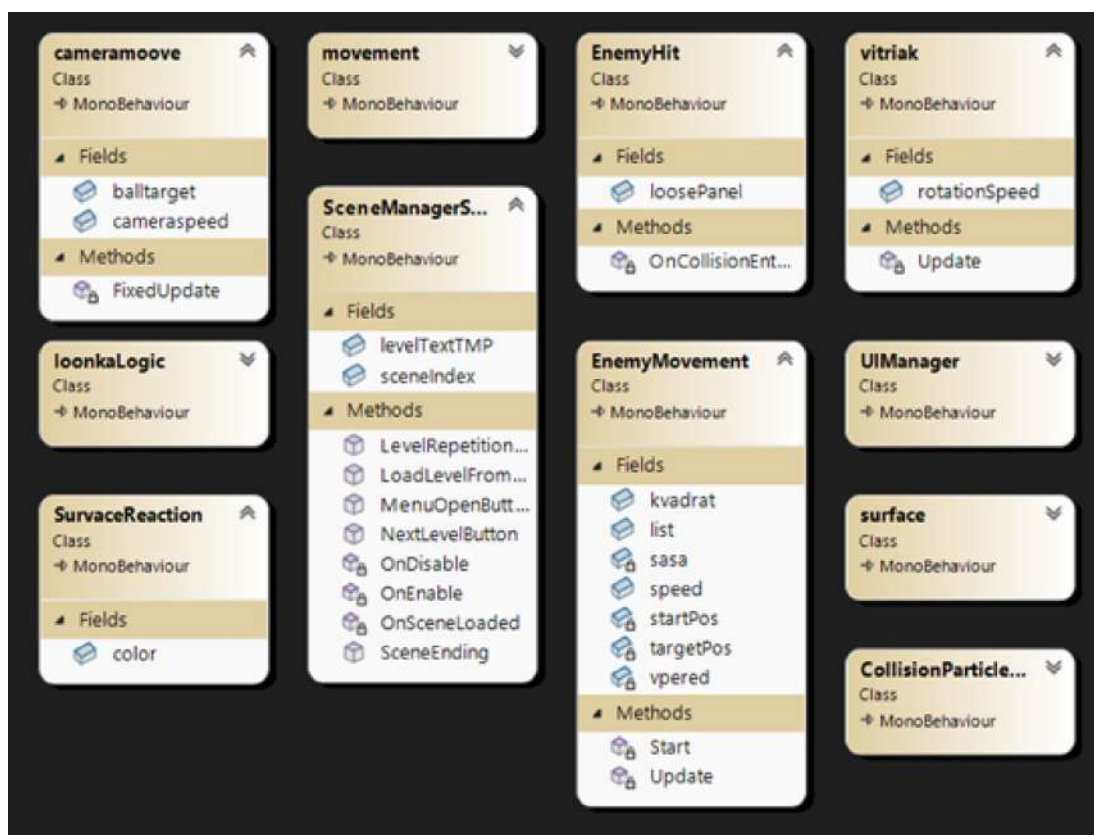


Рис. 1. Діаграма класів

Клас `SceneManagerScript` реалізує управління сценами за допомогою вбудованих механізмів `Unity Scene Management`. Додатково в проєкті використовуються класи допоміжного призначення: `loonkaLogic` – для відображення панелі перемоги та підрахунку ходів; `UIManager` – для відображення результатів гравця; `CollisionParticleManager` та `SurfaceReaction` – для системи ефектів при колізіях; `vitriak` – для обертання об'єктів на сцені; `EnemyHit` та `EnemyMovement` – для логіки поведінки ворожих об'єктів; `Surface` – для впливу властивостей поверхні на швидкість м'яча; `cameraoove` – для плавного слідування камерою за м'ячем.

Таким чином, усі класи взаємодіють між собою для забезпечення повноцінної ігрової логіки та інтерактивності об'єктів.

Висновки

У статті розглянуто алгоритмізацію процесів керування та інтеракцій в об'єктно-орієнтованих ігрових застосунках на прикладі гри «Гольф», розробленої з використанням `C#` та `Unity`. Проведено аналіз ключових алгоритмів, що забезпечують вибір рівнів, керування ігровим об'єктом, проходження рівня, нарахування винагород та обробку поразки.

Було продемонстровано, що інтеграція взаємозалежних функцій у відповідні класи дозволяє забезпечити модульність, масштабованість та ефективну взаємодію об'єктів у грі. Створена діаграма класів відображає структуру взаємозв'язків між основними і допоміжними класами, що забезпечує керованість ігрового процесу та гнучкість у розвитку проєкту. Розроблені алгоритми і класи можуть слугувати основою для подальшого проєктування складніших ігрових систем із високим рівнем інтерактивності та реалістичною фізичною моделлю.

Список використаної літератури

1. Kumar K. Game-changing intelligence: Unveiling the societal impact of artificial intelligence in game software / K. Kumar, N. Veena, T. Aravind, C. Bhatt, U. Kuppusamy, P. Jain // *Entertainment Computing*. 2025. Vol. 52. P. 100862. <https://doi.org/10.1016/j.entcom.2024.100862>
2. Mazurets O., Uspenska K., Vit R., Tyschenko O. Intelligent System for Determining the Object Attributes Values by Neural Networks Means by Graphic Images in Databases / O. Mazurets, K. Uspenska, R. Vit, O. Tyschenko // *Current Trends in the Development of Scientific Research in Today's Conditions: proceedings of XXV International Scientific and Practical Conference (May 29–31, 2024, Florence, Italy)*. Florence, Italy: International Scientific Unity, 2024. P. 86–91.
3. Молчанова М. О., Мазурець О. В., Собко О. В., Кліменко В. І., Андрощук В. І. Метод нейромережевого виявлення кібербулінгу з використанням хмарних сервісів та об'єктно-орієнтованої моделі / М. О. Молчанова, О. В. Мазурець, О. В. Собко, В. І. Кліменко, В. І. Андрощук // *Вісник Хмельницького національного університету. Серія: Технічні науки*. – Хмельницький, 2024. № 2 (333). С. 200–206. <https://doi.org/10.31891/2307-5732-2024-333-2>
4. Молчанова М. О., Мазурець О. В., Шевчук П. О., Кліменко В. І., Тищенко О. О. Підхід до тестування об'єктно-орієнтованих систем керування в електронній комерції / М. О. Молчанова, О. В. Мазурець, П. О. Шевчук, В. І. Кліменко, О. О. Тищенко // *Наука і техніка сьогодні*. Київ, 2025. № 4 (45). С. 1273–1285. [https://doi.org/10.52058/2786-6025-2025-4\(45\)-1273-1285](https://doi.org/10.52058/2786-6025-2025-4(45)-1273-1285)
5. Мазур Є. В., Мазурець О. В., Кліменко В. І., Собко О. В., Залуцька О. О. Алгоритми та програмна архітектура інформаційної системи нейромережевого аналізу постави людини / Є. В. Мазур, О. В. Мазурець, В. І. Кліменко, О. В. Собко, О. О. Залуцька // *Вісник Хмельницького національного університету. Серія: Технічні науки*. Хмельницький, 2025. № 3, Т. 1. С. 275–284. <https://doi.org/10.31891/2307-5732-2025-351-35>
6. Мазурець О. В., Молчанова М. О., Собко О. В., Дідур В. О., Багрій Р. О. Інтеграція об'єктно-орієнтованого підходу та реляційних баз даних при проєктуванні і розробленні веб-платформи для діячів мистецтва / О. В. Мазурець, М. О. Молчанова, О. В. Собко, В. О. Дідур, Р. О. Багрій // *Наука і техніка сьогодні*. Київ, 2025. № 8 (49). С. 1542–1555. [https://doi.org/10.52058/2786-6025-2025-8\(49\)-1542-1555](https://doi.org/10.52058/2786-6025-2025-8(49)-1542-1555)
7. Molchanova M., Mazurets O., Sobko O., Boiarchuk I. Object-Oriented Approach for Ethnic Enmity Detection in Text Messages by NLP / M. Molchanova, O. Mazurets, O. Sobko, I. Boiarchuk // *Proceedings of XXI International Scientific and Practical Conference «Scientific Achievements and Innovations as a Way to Success» (May 1–3, 2024, Vilnius, Lithuania)*. 2024. P. 73–77.
8. Молчанова М. О., Мазурець О. В., Собко О. В., Віт Р. В., Назаров В. В. Алгоритм виявлення аб'юзивного вмісту в україномовному аудіоконтенті для імплементації в об'єктно-орієнтовану інформаційну систему / М. О. Молчанова, О. В. Мазурець, О. В. Собко, Р. В. Віт, В. В. Назаров // *Вісник Хмельницького національного університету. Серія: Технічні науки*. Хмельницький, 2024. № 1 (331). С. 101–106. <https://doi.org/10.31891/2307-5732-2024-331-17>
9. Molchanova M., Mazurets O., Klimenko V., Kuflevsky Ev. Object-oriented model for neural network damage detection of mail packages / M. Molchanova, O. Mazurets, V. Klimenko, Ev. Kuflevsky // *Proceedings of XIV International Scientific and Practical Conference «Solving Scientific Problems Using Innovative Concepts» (March 13–15, 2024, Copenhagen, Denmark)*. 2024. P. 58–62.
10. Mazurets O., Sobko O., Vit R., Pasternak V. Practical Approach for Detection by Deep Learning of Target Objects of Subject Area Based on Semantic Connectivity Indicators in Audio Database / O. Mazurets, O. Sobko, R. Vit, V. Pasternak // *Proceedings of XXIV International Scientific and Practical Conference “Modern Scientific Challenges are the Driving Force of the Development of Scientific Research” (May 22–24, 2024, Bruges, Belgium)*. International Scientific Unity, 2024. P. 91–96.
11. Kok I. A., Kadyńska V. D., Zalutka O. O., Mazurets O. V. Object-Oriented Intelligent System for Automated Control of Smoking by Video Data / I. A. Kok, V. D. Kadyńska, O. O. Zalutka, O. V. Mazurets // *Current scientific goals, approaches and challenges: Proceedings of IV International Scientific and Theoretical Conference*. Dresden, Federal Republic of Germany, 2025. Pp. 156–164.
12. Mazurets O. V., Klimenko V. I., Molchanova M. O., Sultanov A. V. Object-Oriented Intelligent System for Neural Network Detection of Sugar Crystallization Zones / O. V. Mazurets, V. I. Klimenko, M. O. Molchanova, A. V. Sultanov // *Global Science: Prospects and Innovations: Proceedings of the 10th International Scientific and Practical Conference*. Liverpool, United Kingdom: Cognum Publishing House, 2024. P. 198–207.
13. Мазурець О. В., Молчанова М. О., Кліменко В. І., Собко О. В., Супрун П. К. Даталогічна модель бази даних для виявлення гендерної приналежності за SVM-аналізом дописів інтернет-мереж з використанням об'єктно-

орієнтованого проєктування / О. В. Мазурець, М. О. Молчанова, В. І. Кліменко, О. В. Собко, П. К. Супрун // Вісник Хмельницького національного університету. Серія: Технічні науки. Хмельницький, 2024. № 3, Т. 2 (337). С. 197–204. <https://doi.org/10.31891/2307-5732-2024-337-3-29>

14. Molchanova M., Didur V., Mazurets O., Sobko O., Zakharkevich O. Method for Construction and Demolition Waste Classification Using Two-Factor Neural Network Image Analysis / M. Molchanova, V. Didur, O. Mazurets, O. Sobko, O. Zakharkevich // CEUR Workshop Proceedings. 2025. Vol. 3970. P. 168–182. URL: <https://ceur-ws.org/Vol-3970/PAPER14.pdf> (date of access: 19.09.2025).

15. Molchanova M., Didur V., Sobko O., Mazurets O. Detection of Web Propaganda Patterns by Transformer Neural Networks: Improving Efficiency via Dataset Balancing / M. Molchanova, V. Didur, O. Sobko, O. Mazurets // CEUR Workshop Proceedings. 2025. Vol. 3988. P. 112–126. URL: <https://ceur-ws.org/Vol-3988/paper10.pdf> (date of access: 19.09.2025).

16. Zhang Q., Mao N., Li J., Zhou N., Miao J., Li G. Design of the Top-Level Code Framework for Unity Based on C / Q. Zhang, N. Mao, J. Li, N. Zhou, J. Miao, G. Li // ResearchGate. URL: https://www.researchgate.net/publication/391193129_Design_of_the_Top-Level_Code_Framework_for_Unity_Based_on_C_Language (date of access: 19.09.2025).

17. Golob S. P., Pagnon Eriksson N. Using Unity to teach Game Programming / S. P. Golob, N. Pagnon Eriksson // Malmö University, Game Development Program, Computer Science and Media Technology. URL: <https://www.diva-portal.org/smash/get/diva2%3A1867700/FULLTEXT02.pdf> (date of access: 19.09.2025).

18. Torun B., Karakurt S., Aydin T. B., Altunel Y. Game Development on Unity / B. Torun, S. Karakurt, T. B. Aydin, Y. Altunel // Turkish Online Journal of Educational Technology-TOJET. 2021. Vol. 20, No. 1. P. 39–43. URL: <https://eric.ed.gov/?id=EJ1290793> (date of access: 19.09.2025).

19. Fannoun Q. K. Escape the Planet: Revolutionizing Game Design with Novel OOP Techniques / Q. K. Fannoun. 2024. URL: <https://cornerstone.lib.mnsu.edu/etds/1406/> (date of access: 19.09.2025).

20. Wu C. H., Chien Y. C., Chou M. T., Huang Y. M. Integrating computational thinking, game design, and design thinking: a scoping review on trends, applications, and implications for education / C. H. Wu, Y. C. Chien, M. T. Chou, Y. M. Huang // Humanities and Social Sciences Communications. 2025. Vol. 12, No. 1. P. 1–12. <https://doi.org/10.1057/s41599-025-04502-x>

21. Mazurets O., Sobko O., Dydo R., Zalutska O., Molchanova M. Augmented reality audiostream creation using CNN: boosting inclusion and safety for visually impaired people / O. Mazurets, O. Sobko, R. Dydo, O. Zalutska, M. Molchanova // CEUR Workshop Proceedings. 2025. Vol. 4004. P. 347–361. URL: <https://ceur-ws.org/Vol-4004/paper26.pdf> (date of access: 19.09.2025).

References

1. Kumar, K., Veena, N., Aravind, T., Bhatt, C., Kuppusamy, U., & Jain, P. (2025). Game-changing intelligence: Unveiling the societal impact of artificial intelligence in game software. *Entertainment Computing*, 52, 100862. <https://doi.org/10.1016/j.entcom.2024.100862>

2. Mazurets, O., Uspenska, K., Vit, R., & Tyschenko, O. (2024). Intelligent system for determining the object attributes values by neural networks means by graphic images in databases. In *Current trends in the development of scientific research in today's conditions: Proceedings of XXV International Scientific and Practical Conference* (pp. 86–91). Florence, Italy: International Scientific Unity.

3. Molchanova M. O., Mazurets O. V., Sobko O. V., Klimenko V. I., Androshchuk V. I. (2024) Metod neiromerezhevoho vyznachennia kyberbulinhy z vykorystanniam khmarnykh servisiv ta ob'iektno-orientovanoi modeli [Neural network method for cyberbullying detection using cloud services and object-oriented model]. *Visnyk Khmelnytskoho natsionalnoho universytetu. Serii: Tekhnichni nauky*, № 2 (333), pp. 200–206. <https://doi.org/10.31891/2307-5732-2024-333-2>

4. Molchanova M. O., Mazurets O. V., Shevchuk P. O., Klimenko V. I., Tyshchenko O. O. (2025) Pidkhyd do testuvannia ob'iektno-orientovanykh system keruvannia v elektronni komertsii [Approach to testing object-oriented control systems in e-commerce]. *Nauka i tekhnika s'ohodni*, № 4 (45), pp. 1273–1285. [https://doi.org/10.52058/2786-6025-2025-4\(45\)-1273-1285](https://doi.org/10.52058/2786-6025-2025-4(45)-1273-1285)

5. Mazur Y. V., Mazurets O. V., Klimenko V. I., Sobko O. V., Zalutska O. O. (2025) Alhorytmy ta prohramna arkhitektura informatsiinoi systemy neiromerezhevoho analizu postavy liudyny [Algorithms and software architecture of an information system for neural network posture analysis]. *Visnyk Khmelnytskoho natsionalnoho universytetu. Serii: Tekhnichni nauky*, № 3, T. 1, pp. 275–284. <https://doi.org/10.31891/2307-5732-2025-351-35>

6. Mazurets O. V., Molchanova M. O., Sobko O. V., Didur V. O., Bahriy R. O. (2025) Intehratsiia ob'iektno-orientovanoho pidkhodu ta relatsiinykh baz danykh pry proiektuvanni i rozroblenni web-platfomy dlia diiachiv mystetstva [Integration of object-oriented approach and relational databases in design and development of a web platform for artists]. *Nauka i tekhnika siogodni*, № 8 (49), pp. 1542–1555. [https://doi.org/10.52058/2786-6025-2025-8\(49\)-1542-1555](https://doi.org/10.52058/2786-6025-2025-8(49)-1542-1555)

7. Molchanova M., Mazurets O., Sobko O., Boiarchuk I. (2024) Object-Oriented Approach for Ethnic Enmity Detection in Text Messages by NLP. In *Proceedings of XXI International Scientific and Practical Conference “Scientific Achievements and Innovations as a Way to Success”* (pp. 73–77). Vilnius, Lithuania.

8. Molchanova M. O., Mazurets O. V., Sobko O. V., Vit R. V., Nazarov V. V. (2024) Alhoritm vyviedennia ab'iuzyvnoho vmistu v ukraïnomovnomu audiokontenti dlia implementatsii v ob'iektno-orijentovanu informatsiinu systemu [Algorithm for detecting abusive content in Ukrainian audio content for implementation in an object-oriented information system]. *Visnyk Khmelnytskoho natsionalnoho universytetu: Serii Tekhnichni nauky*, 1(331), 101–106. <https://doi.org/10.31891/2307-5732-2024-331-17>
9. Molchanova M., Mazurets O., Klimenko V., Kuflevsky Ev. (2024) Object-oriented model for neural network damage detection of mail packages. In *Proceedings of XIV International Scientific and Practical Conference “Solving Scientific Problems Using Innovative Concepts”* (pp. 58–62). Copenhagen, Denmark.
10. Mazurets O., Sobko O., Vit R., Pasternak V. (2024) Practical approach for detection by deep learning of target objects of subject area based on semantic connectivity indicators in audio database. In *Proceedings of XXIV International Scientific and Practical Conference «Modern Scientific Challenges are the Driving Force of the Development of Scientific Research»* (pp. 91–96). Bruges, Belgium: International Scientific Unity.
11. Kok I. A., Kadynska V. D., Zalutska O. O., Mazurets O. V. (2025) Object-oriented intelligent system for automated control of smoking by video data. Current scientific goals, approaches and challenges: *Proceedings of IV International Scientific and Theoretical Conference*, pp. 156–164. Dresden, Federal Republic of Germany.
12. Mazurets O. V., Klimenko V. I., Molchanova M. O., Sultanov A. V. (2024) Object-oriented intelligent system for neural network detection of sugar crystallization zones. *Global Science: Prospects and Innovations: Proceedings of the 10th International Scientific and Practical Conference*, pp. 198–207. Liverpool, United Kingdom: Cognum Publishing House.
13. Mazurets O. V., Molchanova M. O., Klimenko V. I., Sobko O. V., Suprun P. K. (2024). Datalogichna model' bazy danykh dlia vyviedennia hendernoï prynalezhnosti za SVM-analizom dopysiv internet-merezh z vykorystanniam ob'iektno-orijentovanoho proiektuvannia [Datalogical model of database for detecting gender based on SVM analysis of internet posts using object-oriented design]. *Visnyk Khmelnytskoho Natsionalnoho Universytetu: Serii Tekhnichni Nauky*, 3(2) (337), 197–204. <https://doi.org/10.31891/2307-5732-2024-337-3-29>
14. Molchanova M., Didur V., Mazurets O., Sobko O., Zakharkevich O. (2025). Method for construction and demolition waste classification using two-factor neural network image analysis. *CEUR Workshop Proceedings*, 3970, 168–182. <https://ceur-ws.org/Vol-3970/PAPER14.pdf>
15. Molchanova M., Didur V., Sobko O., Mazurets O. (2025). Detection of web propaganda patterns by transformer neural networks: Improving efficiency via dataset balancing. *CEUR Workshop Proceedings*, 3988, 112–126. <https://ceur-ws.org/Vol-3988/paper10.pdf>
16. Zhang, Q., Mao, N., Li, J., Zhou, N., Miao, J., & Li, G. (n.d.). Design of the top-level code framework for Unity based on C. ResearchGate. https://www.researchgate.net/publication/391193129_Design_of_the_Top-Level_Code_Framework_for_Unity_Based_on_C_Language
17. Golob, S. P., & Pagnon Eriksson, N. (n.d.). Using Unity to teach game programming. Malmö University, Game Development Program, Computer Science and Media Technology. <https://www.diva-portal.org/smash/get/diva2%3A1867700/FULLTEXT02.pdf>
18. Torun, B., Karakurt, S., Aydin, T. B., & Altunel, Y. (2021). Game development on Unity. *Turkish Online Journal of Educational Technology-TOJET*, 20(1), 39–43. <https://eric.ed.gov/?id=EJ1290793>
19. Fannoun, Q. K. (2024). Escape the Planet: Revolutionizing game design with novel OOP techniques. <https://cornerstone.lib.mnsu.edu/etds/1406/>
20. Wu, C. H., Chien, Y. C., Chou, M. T., & Huang, Y. M. (2025). Integrating computational thinking, game design, and design thinking: a scoping review on trends, applications, and implications for education. *Humanities and Social Sciences Communications*, 12(1), 1–12. <https://doi.org/10.1057/s41599-025-04502-x>
21. Mazurets, O., Sobko, O., Dydo, R., Zalutska, O., & Molchanova, M. (2025). Augmented reality audiostream creation using CNN: Boosting inclusion and safety for visually impaired people. *CEUR Workshop Proceedings*, 4004, 347–361. <https://ceur-ws.org/Vol-4004/paper26.pdf>

Дата першого надходження рукопису до видання: 24.09.2025
Дата прийнятого до друку рукопису після рецензування: 21.10.2025
Дата публікації: 28.11.2025