

РОЗРОБКА 3D ГРИ «ROBOT ESCAPE» В ЖАНРІ ПЛАТФОРМЕР

Кропивка Владислав, Пузікова Анна

Центральноукраїнський державний педагогічний університет імені

Володимира Винниченка, м. Кропивницький, Україна

Популярність комп'ютерних ігор зростає з кожним роком, а з розвитком інформаційних технологій з'являється потреба створювати нові більш інноваційні програмні продукти, які будуть задовольняти потреби сучасних користувачів.

У статті наведено стислий опис досвіду розробки відеогри в жанрі платформер за допомогою рушія Unity для операційної системи Windows. Окреслені питання розробки сценарію гри, створення базових 3D-моделей персонажів та налаштування їх анімації, написання скриптів руху персонажа, створення локацій. Зазначені програмні засоби, які були використані на кожному з етапів розробки програмного продукту.

Ключові слова: комп'ютерна гра, платформер, рушій Unity.

Development of 3D game "Robot Escape" in the genre of platformer

V. Kropivka, A. Puzikova

Volodymyr Vynnychenko Central Ukrainian State Pedagogical University,

Kropyvnytsky, Ukraine

The popularity of computer games is growing every year, and with the development of information technology, there is a need to create new, more innovative software products that will meet the needs of their users.

The article provides a brief description of the experience of developing a platformer video game using the Unity engine for the Windows operating system. The issues of developing a game scenario, creating basic 3D models of characters and adjusting their animation, writing scripts for character movement, and creating locations are outlined. The software tools that were used at each stage of the software product development are indicated.

Key words: computer game, platformer, Unity engine.

На сьогоднішній день ринок комп'ютерних ігор росте і розвивається найшвидше у світі. Щоб втримати конкурентоспроможність, розробники, які обмежені у здатності створювати дороговартісну та складну графіку, покладаються на інноваційність ігрового процесу. Так, серед ігор, які здобули міжнародного успіху завдяки своїй творчості розробників і задали новий тренд в

індустрії, стали Cuphead, Ghost of a Tale та Hollow Knight. Ці ігрові відносяться до платформних ігор.

Платформер – жанр комп'ютерних ігор, де основною рисою ігрового процесу є стрибання по платформах, лазіння по сходах, збирання предметів, які необхідні для завершення рівня. Колекційні предмети, зброя та паверапи збираються зазвичай простим дотиком протагоніста і не вимагають спеціальних дій із боку гравця [5].

Сучасний ігровий ринок переповнений іграми жанру платформер, оскільки цей жанр є одним із найстаріших і має велику фан-базу. Але більшість цих ігор мають однотиповий ігровий процес, побудований на стрибках і збиранні предметів, та виконані в стилі 2D. Зважаючи на те, що на сьогоднішній день у гравців одним із найважливіших критеріїв вибору гри є її графіка, платформні ігри, виконані в 3D-стилі, складають значну конкуренцію іншим проектам.

Таким чином, розробка власної платформної гри виконувалась з урахуванням вимог ринку, у 3D-стилі, із забезпеченням значної кількості різноманітних локацій, а також особливостей механіки для кожного наступного рівня, що сприяє утриманню уваги і зацікавленості гравця.

Для створення гри було обрано рушій Unity¹, який на сьогоднішній день є найпопулярнішим серед розробників, оскільки за допомогою його потужних можливостей процес розробки гри можна прискорити в кілька разів, а отже з його використанням невеликі команди можуть скласти конкуренції великим студіям. До того ж база інструментів цього рушія розширюється з кожним роком. Також зауважимо, що великі студії часто створюють свої власні ігрові рушії, що вимагає значних затрат часу.

Метою статті є стислий опис досвіду розробки відеогри «ROBOT ESCAPE» в жанрі платформер за допомогою рушія Unity.

¹ Unity – це крос-платформний ігровий рушій, розроблений Unity Technologies. Рушій поступово розширюється для підтримки різноманітних настільних, мобільних, консольних та віртуальних платформ.

Вхідними даними розробленої програми є ігрові налаштування, які обирає користувач, а також управління аватаром під час ігрового процесу, за допомогою клавіатури.

Вихідними даними – є графічна інтерпретація процесу гри на екрані та звук, який її супроводжує. Дії гравця впливають на стан ігрової сцени.

Відповідно до обраного жанру гри, однією з вимог до функціоналу розробленого продукту була така: «Основні механіки гри полягають в можливості гравця взаємодіяти з необхідними для проходження рівнів об'єктами, збирати предмети та використовувати їх для взаємодії з іншими об'єктами, пересувати об'єкти, перестрибувати та оминати перешкоди».

Алгоритм розробки комп'ютерної гри [4] збігається з алгоритмом розробки більшості програмних продуктів та включає в себе такі стадії: проектування, розробка, видання та підтримка.

В процесі роботи над продуктом було реалізовано розробку сценарію гри, ігрового персонажу, перешкод, противників, предметів для взаємодії персонажа, ігрових локацій; ігрового інтерфейсу, основні функції та механіки гри.

Для досягнення поставлених задач використовувались такі програмні засоби:

- Blender 3D [3] – для створення 3D моделей та їх анімації.
- Adobe Illustrator [1] – для малювання векторних зображень та елементів користувацького інтерфейсу.
- Adobe Substance 3D Painter [2] – для створення текстур та матеріалів.
- Visual Studio Code – для написання C# скриптів.
- Unity Editor – для розробки самої гри.

Сценарій гри містить короткий опис рівнів гри, її основні елементи та події. Відповідно до сценарію були розроблені ескізи ігрових персонажів та їх базові моделі. Базова 3D-модель головного героя – робота, яка була розроблена на 3 модулі відповідно до кожного ігрового рівня, наведена на рисунку 1.

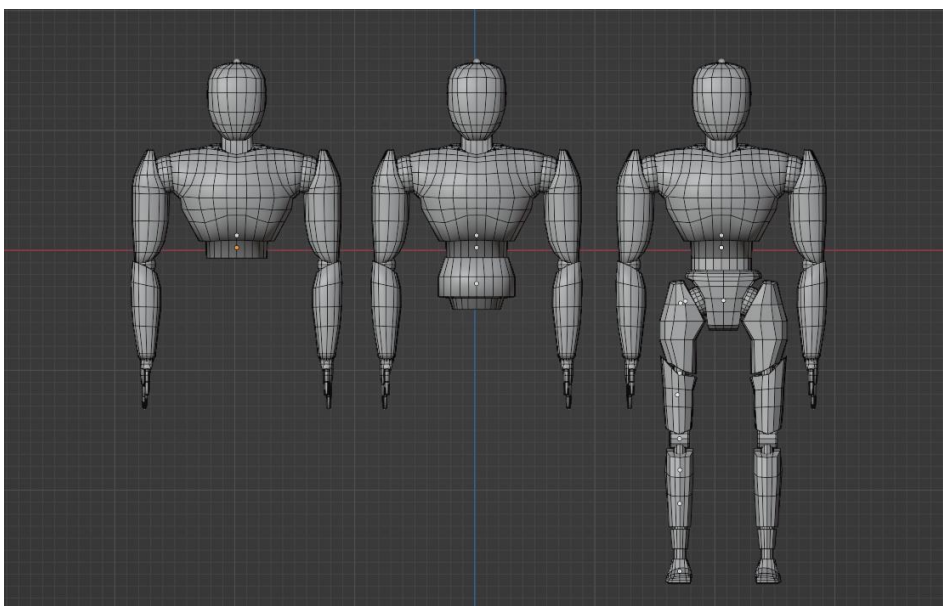


Рис. 1. Базова 3D-модель головного героя

При моделюванні персонажа гри було враховано те, що потрібно дотримуватися норми в кількості полігонів моделі, оскільки за великої кількості полігонів з'являється значне навантаження на графічний процесор.

Після створення 3D-моделей персонажів у програмному засобі Substance Painter було створено та налаштовано карту для кожної моделі (UV-розгортку), як показано на рисунку 2, а також – текстури та матеріали для кожного з персонажів.

Скелет персонажа є одним з ключових компонентів для створення анімації гуманоїдних персонажів. Для створення та налаштування скелета (див. рис. 3) було використано вбудовані можливості програми Blender.

Базові анімації для персонажа було створено також з використанням можливостей Blender. За сценарієм гри в головного героя є три різні стани: без модулів, з модулем левітації та з модулем ніг. Кожен з цих станів передбачав створення свого набору анімацій, таких як: спокій; рух та швидкий рух; взаємодія з об'єктом; взаємодія з предметом; стрибок (для стану з модулем левітації або з модулем ніг); програш; штовхання об'єкта (для стану з модулем польоту або з модулем ніг). Також додатково були створені анімації для супроводження зміни модулів.

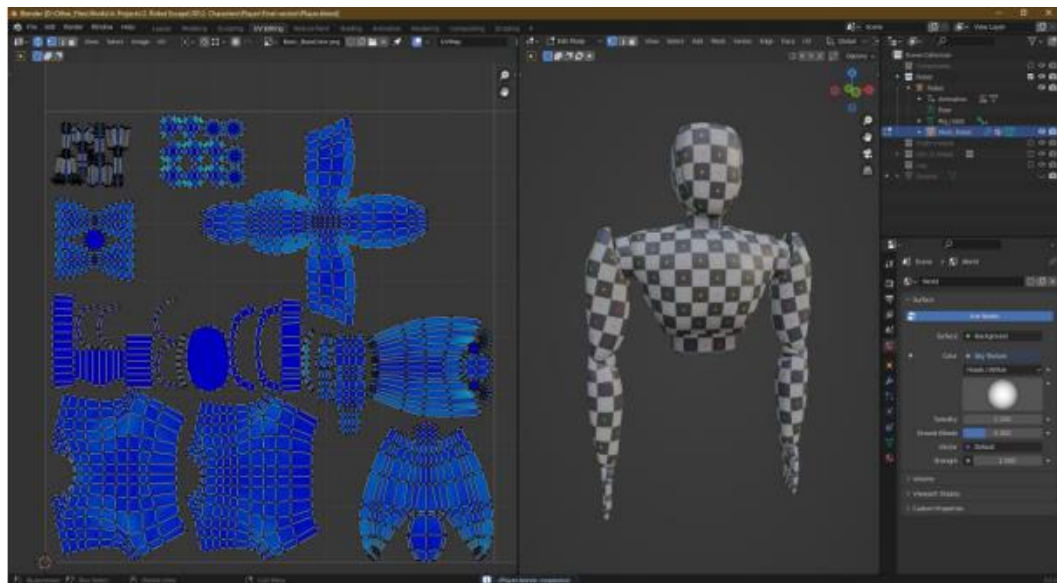


Рис. 2. Розгортка базової моделі головного героя

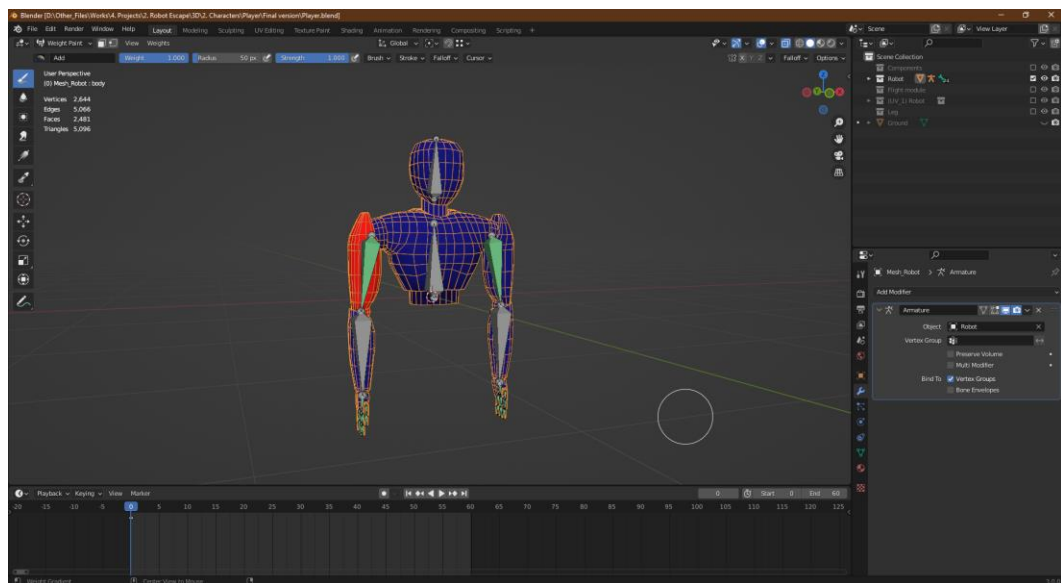


Рис. 3. Налаштування віртуального скелета

Наступний етап полягав у написанні скриптів руху персонажа та задання умов відтворення кожної анімації. Нижче наведено перелік оголошених компонент та змінних, використаних в лістингах анімаційних методів:

- `public CharacterController controller` – основний компонент, який має власні параметри та змінні, за допомогою яких можна задавати рух;
- `public Vector3 move` – змінна, що відповідає за параметри координат в тривимірному просторі;

- public float playerSpeed – змінна, що відповідає за швидкість руху персонажа.
- public float jumpForce – змінна, що задає висоту стрибка.
- [SerializeField] private float gravity – змінна, що відповідає за силу тяжіння.
- public float targetAngle; public float angle; public float turnSmoothTime = 0.4f; public float turnSmoothVelocity; – змінні для задання куту повороту персонажа та його швидкості і плавності.

Нижче наведено лістинг методу «Move», виконання якого задає напрямки руху, а також дозволяє виконувати плавний поворот та повне керування діями.

```
private void Move(){
move = new Vector3(Input.GetAxis("Horizontal"), 0, Input.GetAxis("Vertical"));
if (move != Vector3.zero){
targetAngle = Mathf.Atan2(move.x, move.z) * Mathf.Rad2Deg;
angle      =  Mathf.SmoothDampAngle(transform.eulerAngles.y, targetAngle, ref
turnSmoothVelocity, turnSmoothTime);
transform.rotation = Quaternion.Euler(0f, angle, 0f);
}}

```

На етапі планування також були розроблені схеми локацій. Зокрема, схема локації «Зона перевірки» містить кімнати, сходи, сканери, лазер та усі його компоненти, лампи для джерела світла, ящики та ін. (див. рис. 4); 3D-модель цієї локації наведена на рисунку 5.

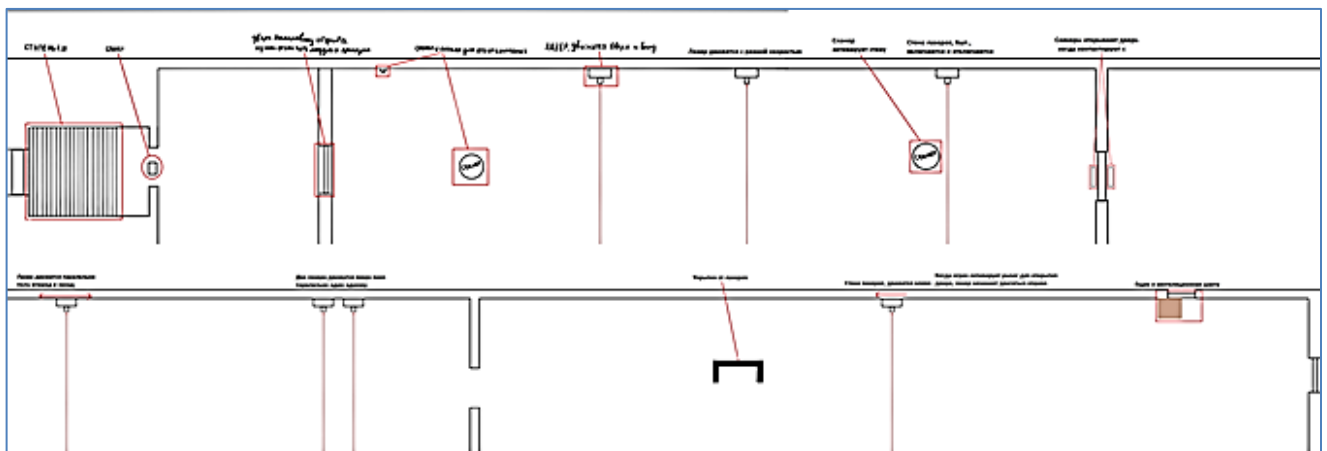


Рис. 4. Схематичний план локації «Зона перевірки»

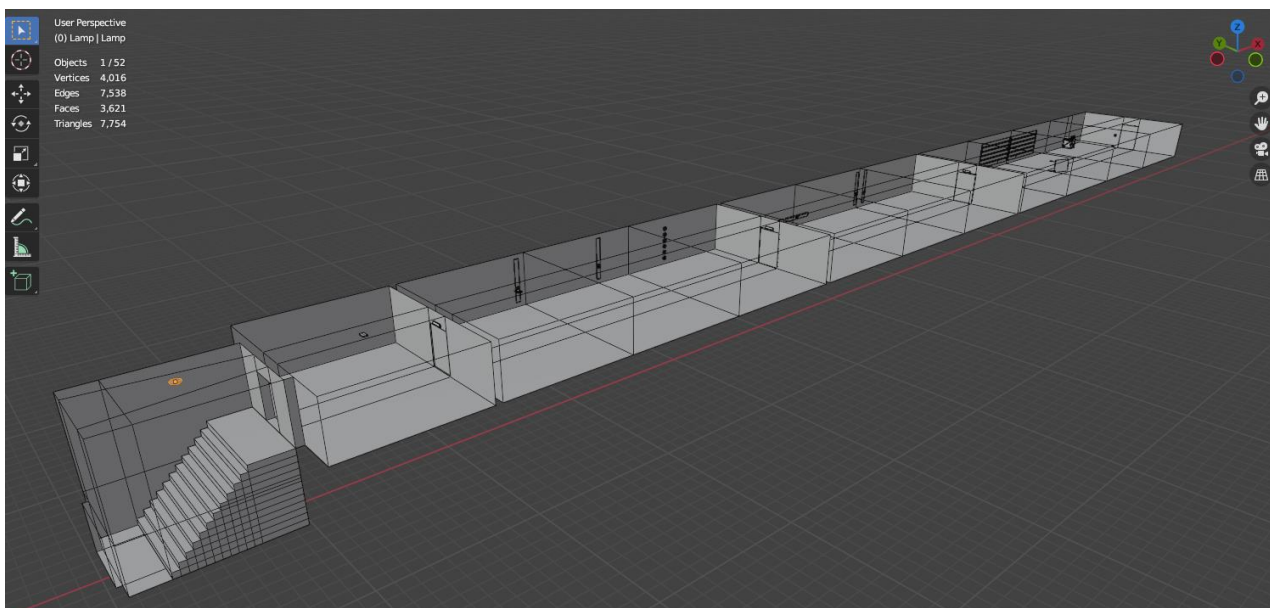


Рис. 5. 3D-модель локації «Зона перевірки»

Створені моделі були імпортовані в редактор Unity та розміщені на сцені.

На етапі реалізації функціональної частини, створено перешкоди у вигляді сканера, який активує червоне світло лазерів та під час тривоги. Також написано скрипти для запуску руху автоматичних дверей (які розміщені майже на всіх переходах між кімнатами), механіки об'єктів (що використовуються як укриття від лазерів), для зміни модуля руху по завершенню проходження локації, тощо.

Зазначимо, що для полегшення програмної реалізації гри було спроектовано UML – діаграми класів, розроблено блок-схему алгоритму здійснення ходу, а також змодельовано основну діаграму варіантів використання.

Після реалізації функціоналу було виконано графічні налаштування оформлень матеріалів і джерел світла відповідно до обраного стилю, окремий результат яких наведено на рисунку 6.

В останню чергу в процесі розробки здійснювалось налаштування головного меню, основною вимогою до якого є: «Головне меню не повинно бути перевантаженим, але в той самий час має бути функціональним та інформативним». Таким чином, головне меню складається з таких компонентів: почати гру; продовжити гру; налаштування; допомога; вихід (див. рис. 6).

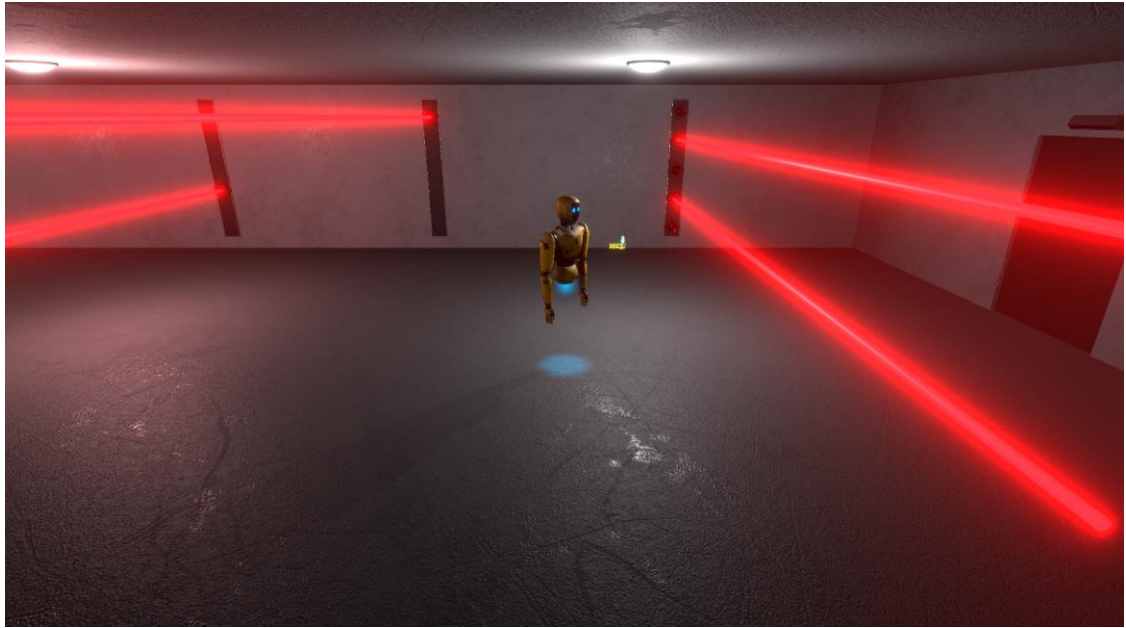


Рис. 6. Скріншот локації «Конвеєрне приміщення»

Набутий під час створення описаної вище відеогри досвід дозволяє стверджувати, що можливості додатку Blender включають необхідні інструменти для швидкого і легкого створення 3D-моделей та анімації, особливо завдяки системі інверсної кінематики, для використання якої непотрібно багато зусиль.

Також зауважимо, що завдяки добротній вбудованій бібліотеці текстур, матеріалів та інших інструментів програмного додатку Adobe Substance Painter вдалося створити гарну та досить деталізовану графіку.

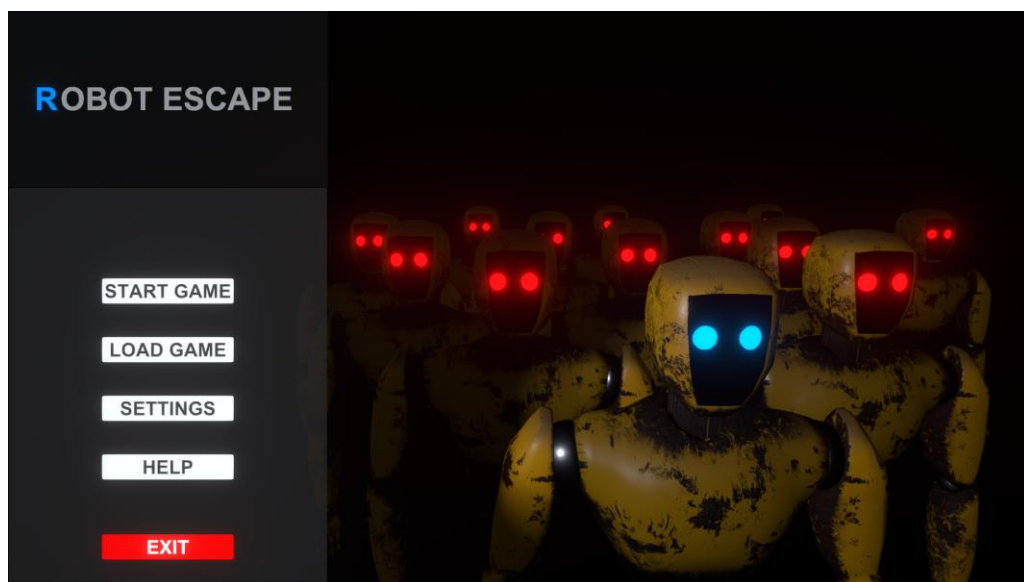


Рис.6. Головне меню відеогри «ROBOT ESCAPE» в жанрі платформер

В свою чергу, можливості ігрового рушія Unity були використані для створення сцен, анімацій, написання скриптів, налаштування об'єктів, створення UI та компіляції проекту.

Таким чином, створений програмний продукт є прикладом відеогри, який можна використовувати як для розвитку логічного мислення та тренування реакційних навичок гравця, так і в навчальному процесі при вивченні курсу «Програмування».

Список використаної літератури

1. Adobe Illustrator. URL: <https://www.adobe.com/ua/products/illustrator.html> (дата звернення: 28.11.2022).
2. Adobe Substance 3D Painter. URL: <https://www.adobe.com/products/substance3d-painter.html> (дата звернення: 28.11.2022).
3. Blender. URL: <https://www.blender.org/> (дата звернення: 28.11.2022).
4. Hocking J. Unity in Action: Multiplatform game development in C# with Unity 5. Manning Publications, 2015. P. 484.
5. What is a Platform Game? | 10 Design Types & Video Game Examples. By: iD Tech. URL: <https://www.idtech.com/blog/10-types-of-platforms-in-platform-video-games> (дата звернення: 28.11.2022).