

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Одеський національний технологічний університет
Університет Інформатики і прикладних знань, м.Лодзь, Польща
Національний технічний університет України «Київський
політехнічний інститут»
Навчально-науковий інститут комп'ютерних систем і технологій
«Індустрія 4.0» ім. П.М. Платонова

XXIII Всеукраїнська науково-технічна конференція
молодих вчених, аспірантів та студентів

«СТАН, ДОСЯГНЕННЯ ТА ПЕРСПЕКТИВИ
ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ»

Матеріали конференції



Одеса

20-21 квітня 2023 р.

Стан, досягнення та перспективи інформаційних систем і технологій / Матеріали XXIII Всеукраїнської науково-технічної конференції молодих вчених, аспірантів та студентів. Одеса, 20-21 квітня 2023 р. - Одеса, Видавництво ОНТУ, 2023 р. – 449 с.

Збірник включає матеріали доповідей учасників конференції, які об'єднані за тематичними напрямками конференції.

Збірник буде корисним як для фахівців і працівників фірм, зайнятих в області ІТ, так і для викладачів, магістрів і студентів вищих навчальних закладів, які навчаються за напрямками і спеціальностями програмного забезпечення, обчислювальної техніки і автоматизованих систем, прикладної математики та обробки інформації, буде корисним професіоналам з комп'ютерного моделювання та розробки комп'ютерних ігор.

Результати досліджень у збірнику представляють собою своєрідний зріз сучасного стану справ в перерахованих галузях знань, який може допомогти як фахівцям, так і студентам університетів скласти загальну картину розвитку інформаційних технологій та пов'язаних з ними питань.

Наукові праці згруповані за напрямками роботи конференції та наведені в алфавітному порядку прізвищ авторів.

Матеріали (тези доповідей) друкуються в авторській редакції. Відповідальність за якість та зміст публікацій несе автор.

Матеріали подано українською та англійською мовами.

Редактор збірника Котлик С.В.

університет)	
30. Концепція інтелектуальної інформаційної системи аналізу та обробки антропометричних даних. Бондаренко А.С. (Національний університет «Одеська політехніка»)	373
Розділ 7: Комп'ютерні ігри і WEB-дизайн	375
1. The role of game projects in teaching programming. Рап А., Kim Ye.R. (University "Turan", Kazakhstan)	375
2. Development of the computer game "fade out" on the unity platform in the mode of 2d modeling. Tarasov V.A., Lagutkin B.D., Bibik I.V., Kasimova A.K., Mamurova A.K. (College "turan", University "Turan", Kazakhstan)	377
3. Особливості дизайну для людей з особливими потребами. Альпашкін М.І., Романюк О. Н., Романюк О.В., Котлик С.В. (Вінницький національний технічний університет, Одеський національний технологічний університет)	380
4. Вплив трасування променів на рендер game-ready моделей. Бойцова М.П., Ломовцев П.Б. (Одеський національний технологічний університет)	382
5. Аналіз безкоштовних графічних редакторів для веб-дизайну. Бондаренко Н.О., Романюк О.Н. (Вінницький національний технічний університет)	383
6. Розробка ігрового додатку в середовищі UNITY. Булах В.О., Сахарова С.В., Іванова Л.В. (Одеський національний технологічний університет, Одеський технічний фаховий коледж ОНТУ)	385
7. Дослідження засобів бібліотеки react для розробки текстових блогів. Гандзюк К. Р. (Волинський національний університет імені Лесі Українки).	388
8. Штучний інтелект у комп'ютерних іграх. Геселева Н.В., Коваль А. К. (Державний торговельно-економічний університет)	389
9. Розробка бізнес-гри для навчання керівничого персоналу. Горбатюк М.В., Стельмашенко А.В. (Український державний хіміко-технологічний університет)	391
10. Кастомізабельний Тетріс на основі рушія HGE. Данченков В.О., Данченков Я.В. (Національний університет водного господарства та природокористування)	393
11. Розробка інтелектуальної гри-вікторини за допомогою UNITY 3D. Деркач Т.М., Вітер В.В. (Національний університет "Полтавська політехніка імені Юрія Кондратюка")	395
12. Дослідження та розробка анімаційної візуалізації інтер'єру у середовищі Unreal Engine. Єрощенко М.В., Ломовцев П.Б. (Одеський національний технологічний університет)	397
13. Комп'ютерні ігри і WEB-дизайн. Жилін М. (Національний університет "Одеська політехніка")	398
14. Аналіз ігор жанру «Симулятор колонії». Зелененький А.О., Ненов О.Л. (Одеський національний технологічний університет)	399
15. Розробка гри в жанрі ENDLESS RUNNER. Карякін Д. (Національний університет кораблебудування ім. адмірала Макарова)	402
16. Середовище розробки ігор Gamedeaker: огляд основних функцій та можливостей. Костюк В. В., Мельник А. В. (Житомирський державний університет ім. Івана Франка)	403
17. Комп'ютерні ігри та WEB-дизайн. Кравцова А. О. (Національний університет «Одеська політехніка».)	406
18. Розробка відеогри для стимуляції покращення зорових функцій. Курашин Є.О. (Український державний хіміко-технологічний університет)	407
19. Переваги RPG жанру відеоігор. Кухаренко В.С., Гайдаєнко О.В. (Національний університет кораблебудування імені адмірала Макарова)	410
20. Комп'ютерна RPG-гра «FATEFUL STORY». Лисогурський М.Л., Швець Н.В. (ВСП "Фаховий коледж промислової автоматики та інформаційних технологій ОНТУ)	412

DEVELOPMENT OF THE COMPUTER GAME "FADE OUT" ON THE UNITY PLATFORM IN THE MODE OF 2D MODELING

V.A. TARASOV¹ (vovochka.tarasov.04@mail.ru) **B.D. LAGUTKIN¹, I.V. BIBIK¹,
A.K. KASIMOVA¹, A.K. MAMYROVA²**
¹ College "Turan", ² University "Turan"

This article presents the stages of designing a computer game developed in the Unity environment, the objects are made in Adobe Photoshop, the code listing is written in C# programming language and has been tested using artificial intelligence ChatGPT.

The aim of the work is to develop the game "Fade Out" in the 2D platformer genre. To achieve this goal, the following tasks must be solved:

- to analyze similar projects;
- to analyze and select the means of implementing the game;
- describe the concept of the game;
- design a software system;
- implement the game;
- to test the implemented game.

Software tools. We used the program "Unity", "Photoshop" to assist and edit graphic images, as well as a neural network ChatGPT for detecting errors in the program code. Unity is a game engine used to develop mobile games and projects for PC (Windows, iOS, Linux) and consoles, such as Xbox, PlayStation. [1, 3] It has various components for working with graphics, animation, object physics, sound, templates and scripts. Adobe Photoshop is a multifunctional editor for working with photo and video files (raster images and several vector tools). [2] The software product works on PCs with operating systems macOS, Windows and mobile versions of iOS, Windows phone and Android. The Unity editor has a simple Drag&Drop interface, which consists of various windows, so you can debug the game right in the editor. The engine uses C# to write scripts. The editor has an object inheritance system; child objects will repeat all changes in the position, rotation and scale of the parent object. Scripts in the editor are attached to objects as separate components. [3]

Development algorithm:

1. Search for ideas and concepts.
 - 1.1. Search game genre.
 - 1.2. Setting search.
 - 1.3. Search for game design details.
2. Adapting already implemented ideas and concepts to your idea.
3. Choosing a software environment (Engine, programming language) for the implementation of the planned game design plan.
4. Creation from scratch of game sources.
 - 4.1. Search for ready-made sprites.
 - 4.2. Creating animations from scratch.
 - 4.3. Creation of scripts from scratch.
5. Creation of prefabs based on a combination of sprites, animations and scripts.
6. Creation of game scenes from prefabs.
7. Create a connection between game scenes.
8. Creation of an intuitive, working interface for the game.
9. Creation of game levels from scenes.
10. The final building of the game from the working environment into a separate program.
 - 10.1. Platform - PC.
 - 10.2. Platform - Android / IOS.

10.3. Platform - consoles.

Software implementation. Control. There are different player states. Depending on the state, which is determined by input from the outside, the corresponding animation will be played. (Fig. 1)

```
53     {
54         Flip();
55     }
56     if (inputHorizontal < 0 && facingRight)
57     {
58         Flip();
59     }
60 }
61
62 void Flip()
63 {
64     Vector3 currentScale = gameObject.transform.localScale;
65     currentScale.x *= -1;
66     gameObject.transform.localScale = currentScale;
67     facingRight = !facingRight;
68 }
69
70 private void Update()
71 {
72     if (isGrounded && !isAttacking) State = States.idle;
73
74     if (!isAttacking && Input.GetButton("Horizontal"))
75         Run();
76     if (!isAttacking && isGrounded && Input.GetButtonDown("Jump"))
77         Jump();
78     if (Input.GetButtonDown("Fire1"))
79         Attack();
80 }
81
82 private void Run()
83 {
84     if (isGrounded) State = States.run;
85
86     Vector3 dir = transform.right * Input.GetAxis("Horizontal");
87     transform.position = Vector3.MoveTowards(transform.position, transform.position + dir, speed * Time.deltaTime);
88 }
89
90 private void Jump()
91 {
92     rb.AddForce(transform.up * jumpForce, ForceMode2D.Impulse);
93 }
94
95 private void Attack()
96 {
97     if (isGrounded && isRecharged)
98     {
99         State = States.attack;
100        isAttacking = true;
101        isRecharged = false;
102        StartCoroutine(AttackAnimation());
103        StartCoroutine(AttackCoolDown());
104    }
```

Figure 1 - Listing animation of the main character.

Also developed the implementation of the enemy. The PathFinding library is used, which allows you to find the shortest paths to the player. Unlike the player, the opponent does not have states, since they are not implemented together with the player search algorithms. (Fig. 2)

```
53         isAttackReady = false;
54         lastAttackTime = Time.time;
55         animator.Play("attack");
56         StartCoroutine(AttackCoolDown());
57     }
58 }
59 private IEnumerator AttackCoolDown()
60 {
61     yield return new WaitForSeconds(attackCooldown);
62     isAttackReady = true;
63 }
64 private void OnDrawGizmosSelected()
65 {
66     Gizmos.color = Color.red;
67     Gizmos.DrawWireSphere(attackPoint.position, attackRange);
68 }
69 private void OnAttackAnimationEnd()
70 {
71     animator.SetBool("isAttacking", false);
72 }
73 }
74 }
```

Figure 2 - Testing the program using artificial intelligence.

Software implementation. Health bar. An object was created on the stage, which is obtained in the script, then there is a manipulation with the Transform parameter of this object. (Fig. 3)

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class CurrentHealth : MonoBehaviour
6 {
7     private RectTransform CurHealth;
8
9     void Start()
10    {
11        CurHealth = GetComponent<RectTransform>();
12    }
13
14    void Update()
15    {
16        CurHealth.localScale = new Vector2(Player.CurrentLives*0.01f, 0.3317f);
17    }
18 }
19
```

Figure 3 - Listing of the health bar code.

The same is true for the maximum amount of health, expressed as a box on the scale.

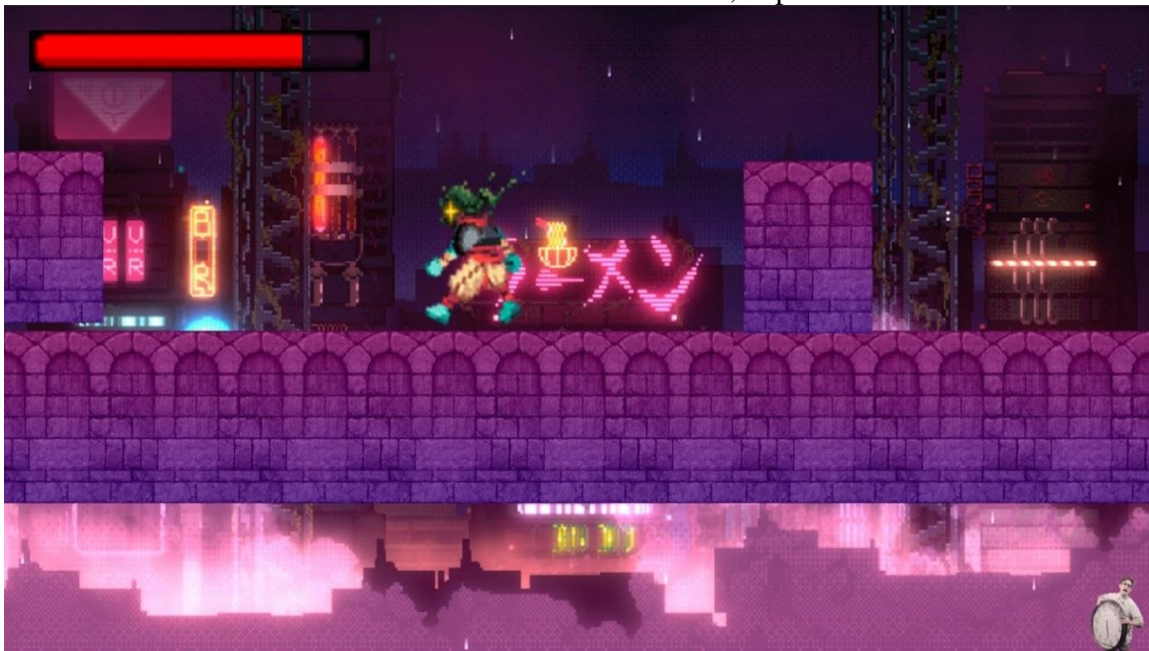


Figure 4 - Screenshot of the game interface.

Conclusions. The result of the project was the development of the game - 2D platformer. The Unity game engine was ideal for solving the tasks due to its simplicity and flexibility of functionality. The game has a system of levels and player respawn. The initial game scene is the main menu, which has a blank for settings. The game has a health system not only for the player, but also for opponents. At the same time, means of visual display of the state of health have been developed. There is a non-playable zone, when you enter it, the level starts from zero, there are also portals to go to the next level. An interface has been developed to implement a pause and exit to the main menu, as well as to exit the game. (Fig. 4) Opponents have artificial intelligence and are able to find the best route to pursue the player.

LIST OF USED LITERATURE.

1. <https://docs.unity3d.com/Manual/index.html>
2. <https://chat.openai.com/chat>