



UNIWERSYTET ZIELONOGÓRSKI  
Wydział Informatyki, Elektrotechniki i Automatyki  
Instytut Sterowania i Systemów Informatycznych

Programowanie gier 3D – Projekt  
Prowadzący: Mgr inż. Marcin Skobel

Wściekły Maks

Stanisław Mól, Erwin Konkel  
Grupa dziekańska: 33INF-SSI-SP

Data oddania projektu: 22.01.2021

Ocena: .....

## Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>3</b>
<b>2</b>	<b>Charakterystyka techniczna</b>	<b>3</b>
2.1	Statystyki techniczne . . . . .	3
2.2	Najciekawsze rozwiązania techniczne zastosowane w grze . . . . .	3
2.2.1	Skrypty . . . . .	3
2.2.2	Animator . . . . .	7
<b>3</b>	<b>Scenariusz gry</b>	<b>7</b>
<b>4</b>	<b>Rozgrywka</b>	<b>8</b>
<b>5</b>	<b>Opis wkładu własnego w realizację projektu</b>	<b>9</b>
<b>6</b>	<b>Spis zastosowanych assetów z krótką charakterystykę</b>	<b>10</b>
<b>7</b>	<b>Podsumowanie</b>	<b>11</b>

## Spis listingów

1	Skrypt 'Timer' - wykonywanie akcji po ustawionym uprzednio czasie . . . . .	3
2	Skrypt 'EnemyAi' - zachowanie oraz poruszanie się przeciwników . . . . .	5

---

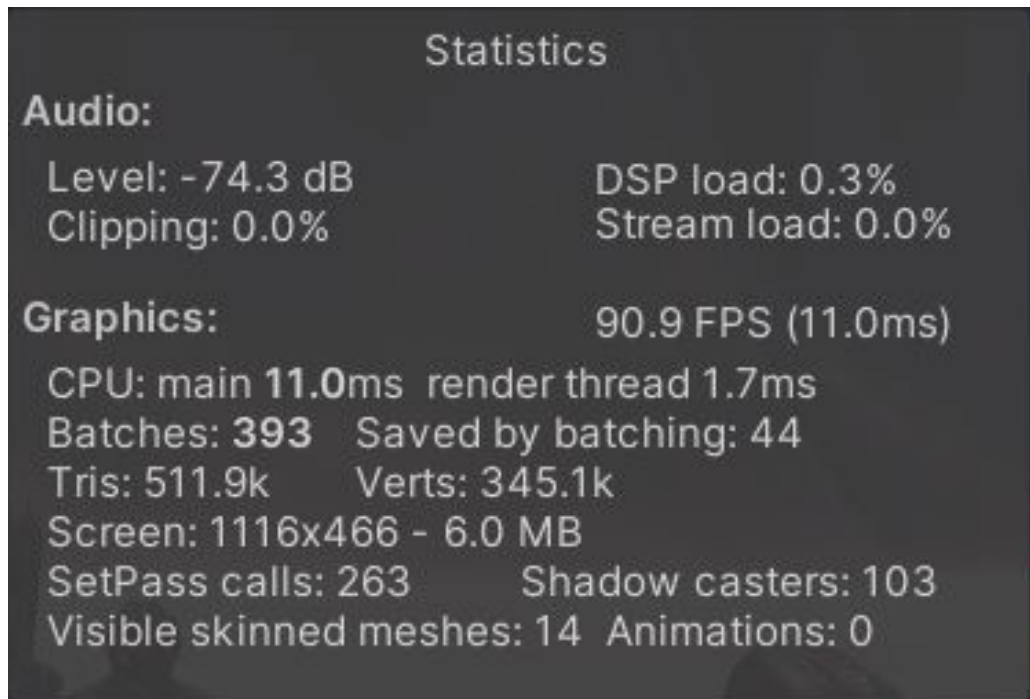
# 1 Wprowadzenie

Projekt jest to gra typu przygodowo-bijatykowej o otwartym wielkim świecie. Trafiamy na obszar startowy, w którym pomagają nam postacie w grze i mówią co mamy robić jako zadania w świecie gry. Mapa jest wielką pustynią z charakterystycznymi punktami i trzema wioskami w których otrzymujemy zadania. Polega to na podróżowaniu między wioskami i walce z pojawiającymi się potworami. Platforma sprzętowa to komputery z systemem Windows

## 2 Charakterystyka techniczna

### 2.1 Statystyki techniczne

Całkowita rozmiar gry to 575MB, średnia ilość FPS w grze waha się pomiędzy 60-90. Najbardziej obciążającą lokalizacją jest wioska "Na prawo".



Rysunek 1: Statystyki gry.

### 2.2 Najciekawsze rozwiązania techniczne zastosowane w grze

#### 2.2.1 Skrypty

Listing 1: Skrypt 'Timer' - wykonywanie akcji po ustawionym uprzednio czasie

```
using UnityEngine;
using UnityEngine.Events;
using System;
using System.Collections;

public enum TimerMode
{
    CountToDown,
    CountToUp
};

public class Timer : MonoBehaviour
{

    public float timerCounter;
```

```

public float timerTarget;
public bool isRun;
public TimerMode timerMode;
public UnityEvent OnTimerComplete;
public UnityEvent OnTimerCount;

public void InitializeTimer(float startValue, float targetValue, TimerMode mode)
{
    try
    {
        setTimerCounter(startValue);
        setTimerMode(mode);
        setTimerTarget(targetValue);
    }
    catch (ArgumentException)
    {
        Debug.Log(" Initialize Timer had set invalid parameters!");
    }
}

public void run()
{
    isRun = true;
}

public void stop()
{
    isRun = false;
}

private void Update()
{
    if (isRun)
    {
        OnTimerCount.Invoke();

        bool condition = false;

        if (timerMode == TimerMode.CountToUp)
        {
            condition = (timerCounter > timerTarget) ? true : false;
            timerCounter += Time.deltaTime;
        }
        else
        {
            condition = (timerCounter < timerTarget) ? true : false;
            timerCounter -= Time.deltaTime;
        }

        if (condition == true)
        {
            stop();
            OnTimerComplete.Invoke();
        }
    }
}

public void setTimerCounter(float timerCounter)
{
    if (timerCounter == 0)
    {
        throw new ArgumentException("Time must be greater than 0!", "timerCounter");
    }

    this.timerCounter = timerCounter;
}

```

```

    }

    public float getTimerCounter()
    {
        return timerCounter;
    }

    public void setTimerTarget(float timerTarget)
    {
        if (timerTarget < 0)
        {
            throw new ArgumentException("Timer duration must be greater than 0!", "timerDuration");
        }

        this.timerTarget = timerTarget;
    }

    public float getTimerTarget()
    {
        return timerTarget;
    }

    public void setTimerMode(TimerMode timerMode)
    {
        if (!Enum.IsDefined(typeof(TimerMode), timerMode))
        {
            throw new ArgumentException("Mode of Timer must set right!", "timerCountMode");
        }

        this.timerMode = timerMode;
    }

    public TimerMode getTimerCountMode()
    {
        return timerMode;
    }

    public void ResetTimer()
    {
        timerCounter = 0;
    }
}

```

Listing 2: Skrypt 'EnemyAi' - zachowanie oraz poruszanie się przeciwników

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.AI;

public class EnemyAi : MonoBehaviour
{
    public int health = 100;
    public NavMeshAgent agent;
    public Transform player;
    public LayerMask whatIsGrounded, whatIsPlayer;

    public Vector3 walkPoint;
    bool walkPointSet;
    public float walkPointRange;

    public float timeBetweenAttacks;
    bool alreadyAttacked;
    public GameObject projectile;

    public float sightRange, attackRange;
    public bool playerInSightRange, playerInAttackRange;
}

```

```

public Animator animator;

private void Awake()
{
    agent = GetComponent<NavMeshAgent>();
    player = GameObject.FindGameObjectWithTag("Player").transform;
}

private void SearchWalkPoit()
{
    float randomZ = Random.Range(-walkPointRange, walkPointRange);
    float randomX = Random.Range(-walkPointRange, walkPointRange);

    walkPoint = new Vector3(transform.position.x + randomX, transform.position.y, transform.position.z + randomZ);

    if (Physics.Raycast(walkPoint, -transform.up, 2f, whatIsGrounde))
        walkPointSet = true;
}

private void Patrolling()
{
    animator.SetBool("isAttack", false);
    animator.SetBool("isDead", false);

    if (!walkPointSet) SearchWalkPoit();
    if (walkPointSet) agent.SetDestination(walkPoint);

    Vector3 distanceToWalkPoint = transform.position - walkPoint;

    if (distanceToWalkPoint.magnitude < 1f) walkPointSet = false;
}

private void ChasePlayer()
{
    agent.SetDestination(player.position);
}

private void AttackPlayer()
{
    animator.SetBool("isAttack", true);
    agent.SetDestination(transform.position);
    transform.LookAt(player);

    if (!alreadyAttacked)
    {
        alreadyAttacked = true;
        Invoke(nameof(ResetAttack), timeBetweenAttacks);
        RaycastHit theHit;

        if (Physics.Raycast(transform.position, transform.TransformDirection(Vector3.forward), out theHit, walkPointSet ? walkPoint.magnitude : 10f))
            theHit.transform.SendMessage("HitByEnemy");
    }
}

private void ResetAttack()
{
    animator.SetBool("isAttack", false);
    alreadyAttacked = false;
}

private void TakeDamage(int damage)
{
    health -= damage;
    if (health <= 0) Invoke(nameof(DestroyEnemy), 0.5f);
}

private void DestroyEnemy()
{
    animator.SetBool("isAttack", false);
}

```

```

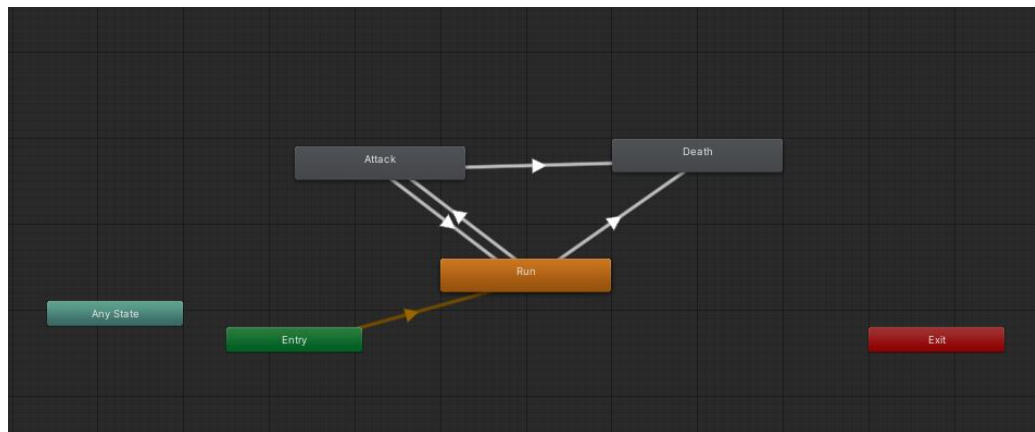
    animator.SetBool("isDead", true);
}

void Update()
{
    playerInSightRange = Physics.CheckSphere(transform.position, sightRange, whatIsPlayer);
    playerInAttackRange = Physics.CheckSphere(transform.position, attackRange, whatIsPlayer);

    if (!playerInSightRange && !playerInAttackRange) Patrolling();
    if (playerInSightRange && !playerInAttackRange) ChasePlayer();
    if (playerInSightRange && playerInAttackRange) AttackPlayer();
}
}

```

### 2.2.2 Animator



Rysunek 2: Animator - EnemyController

## 3 Scenariusz gry

Wcielamy się w rolę fałszywie oskarżonego łowcy przygód, który w trakcie jednej ze swoich wypraw zostaje złapany przez straż panującą w tej krainie. W trakcie ucieczki gubi straż i natrafia na pustynię, na której znajduje obóz w którym wykończony mdlejesz. Okazuje się, że jest to obóz ludzi którzy również mają z prawem na pieńku. Po obudzeniu się wita nas kapitan obozu, który daje nam pierwsze zadanie abyśmy dotarli do mędrca w wiosce na wschód. Po drodze atakują nas potwory, które władza wysłała w to miejsce, ponieważ nie potrafi nic innego z nimi zrobić. Okazuje się, że nasze dłonie są magiczne i jesteśmy w stanie te potwory pokonać. W trakcie podróży musimy walczyć o przetrwanie i wydostać się z tej krainy aby udowodnić swoją niewinność. Na końcu spotykamy straszną zjawę, która okazuje się być źródłem powstawania potworów. Kiedy ją pokonujemy, kraina zostaje uratowana i wszyscy mogą spokojnie żyć.

## 4 Rozgrywka



## 5 Opis wkładu własnego w realizację projektu

### **Stanisław Mól:**

- Postać gracza
- Poruszanie się gracza
- Pierwszy przeciwnik
- Teren poruszania się przeciwnika
- System animacji przeciwnika
- Atakowanie / poruszanie się przeciwnika (pojawily się błędy)
- Naprawa błędów
- Zmiana przeciwnika na Humanoidalnego
- Dodanie MainMenu

### **Erwin Konkel:**

- Skybox
- Design mapy
- Tekstury
- Wieże
- Wioska początkowa
- Granice mapy
- Ulepszenie ścieżki
- Poprawione oświetlenie
- Rozwijanie wioski
- NPC
- Dodanie nowego poziomu (nowa wioska)

## 6 Spis zastosowanych assetów z krótką charakterystykę

- Third Person Controller - Basic Locomotion FREE - <https://assetstore.unity.com/packages/tools/utilities/third-person-controller-basic-locomotion-free-82048>
- Free Low Poly Desert Pack - <https://assetstore.unity.com/packages/3d/environments/free-low-poly-desert-pack-106709>
- Fantasy Skybox Free - <https://assetstore.unity.com/packages/2d/textures-materials/sky/fantasy-skybox-free-18353>
- Ruined Tower Free - <https://assetstore.unity.com/packages/3d/environments/ruined-tower-free-66495>
- Spider Green - <https://assetstore.unity.com/packages/3d/characters/animals/insects/spider-green-11869>
- Campfire Pack - <https://assetstore.unity.com/packages/3d/environments/fantasy/campfire-pack-11256>

## 7 Podsumowanie

opisanie trudności oraz rozwiązania, wskazanie pozytywnych cech projektu oraz ewentualnych planów i szans na dalszy rozwój projektu.

---

Koniec dokumentu.