



Uniwersytet Rzeszowski  
Kolegium Nauk Przyrodniczych  
Instytut Informatyki

# Projektowanie Gier Komputerowych

Sprawozdanie projektu pt. „Settlers – The Origin”

Autorzy:

Patryk Arendt 113705,  
Damian Jamroży 113729

Prowadzący:

Dr inż. Wojciech Koziół

## Spis treści

|  |    |
|--|----|
| Wstęp .....  | 2  |
| Opis technologii.....  | 2  |
| Unity .....  | 2  |
| Mixamo.....  | 3  |
| Visual studio code .....                                       | 4  |
| Unity Assets Store .....                                       | 4  |
| Blender .....  | 5  |
| Wymagania sprzętowe.....                                       | 6  |
| Minimalne.....   | 6  |
| Rekomendowane .....  | 6  |
| Projekt świata gry .....                                       | 7  |
| Projekt interfejsu gracza .....                                | 8  |
| Projekt postaci .....  | 10 |
| Postać główna.....   | 10 |
| Postacie fabularne.....  | 11 |
| Postacie niezależne.....                                       | 11 |
| Opis fabuły .....  | 13 |
| Wątki fabularne .....  | 13 |
| Quest 1. Prologue - Przynieś miecz dziwnemu nieznajomemu ..... | 13 |
| Quest 2. Zbieranie pomidorów .....                             | 14 |
| Quest 3. Poszukiwania straszego stworzenia .....               | 15 |
| Przykładowe mechaniki rozgrywki .....                          | 16 |
| Poruszanie się .....   | 16 |
| Interakcje z przedmiotami .....                                | 20 |
| Atak mieczem.....  | 22 |
| Strzelanie z łuku .....  | 23 |
| Pora dnia.....   | 24 |
| Poziom wytrzymałość postaci .....                              | 25 |
| AI postaci niezależnych.....                                   | 26 |
| Zaimplementowane skrypty .....                                 | 27 |
| Podział obowiązków.....  | 29 |
| Spis ilustracji .....  | 30 |
| Netografia .....   | 31 |

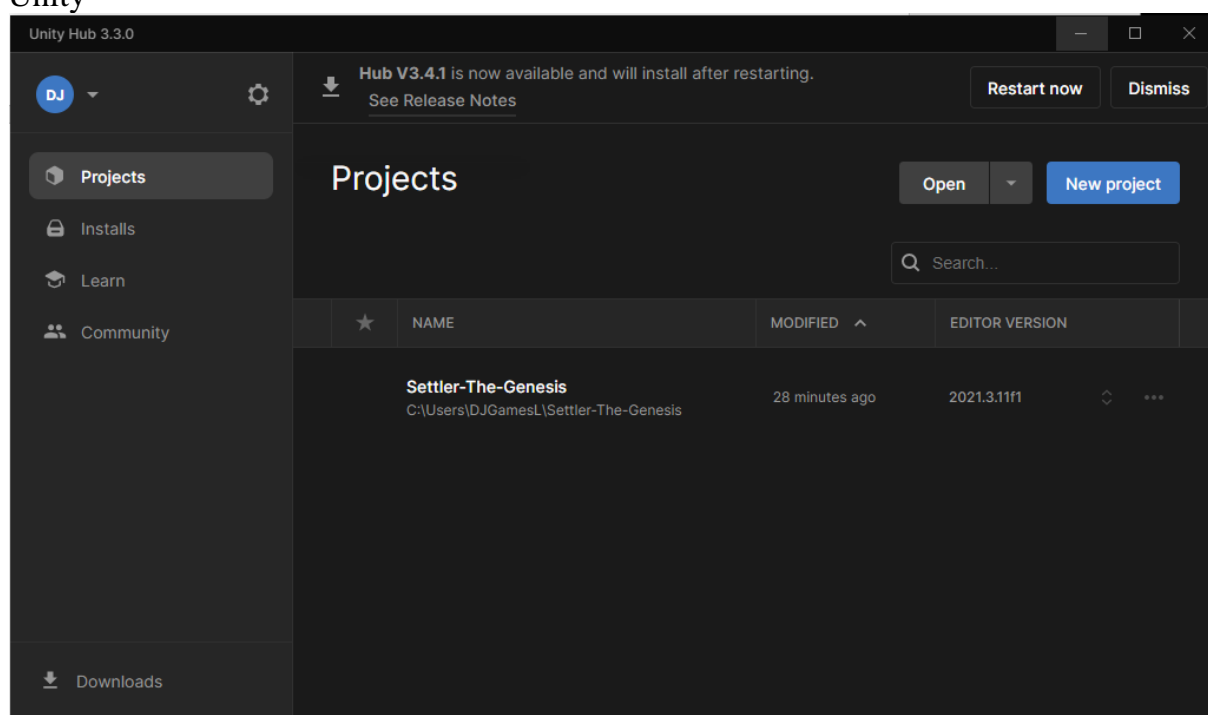
## Wstęp

Gra *Settler – The Genesis*, stworzona została przez dwóch studentów Uniwersytetu Rzeszowskiego. W założeniu, skierowana jest do grupy docelowej 16+, a rodzaj gry określony został jako RPG. Oferuje ona rozgrywkę z trzecio-osobowej perspektywy w technologii 3D. Fabuła gry zorientowana jest w alternatywnym średniowieczu, gdzie na gracza czekają zadania różnej wielkości i złożoności. Gra we wczesnej wersji beta bazuje na licencji beerware.

## Opis technologii

Do wykonania projektu wykorzystano liczne oprogramowania graficzne jak służące stricte w branży game-dev. Każde z wykorzystanych narzędzi oferuje darmowe funkcje, pozwalające na przyspieszenie lub polepszenie pracy nad produkowaną grą.

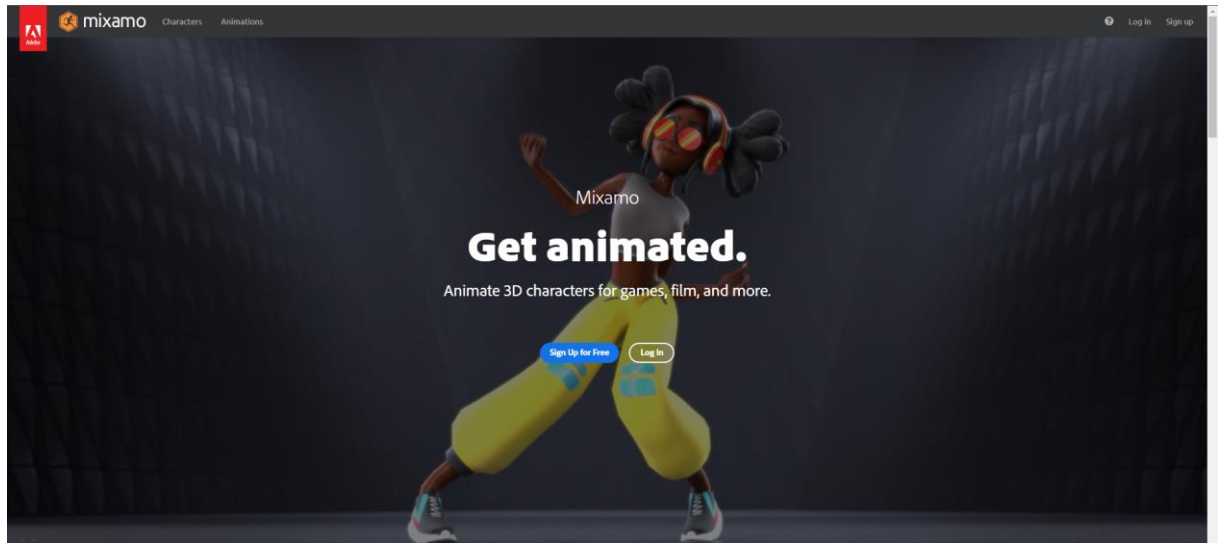
## Unity



Ryc. 1 Interfejs programu Unity Hub

Głównym narzędziem wykorzystywanym do integracji części graficznej z częścią programistyczną została platforma Unity. Oprogramowanie to oferuje ogromne możliwości, a co najważniejsze jest ono udostępnione w wersji darmowej do nisko-komercyjnych celów.

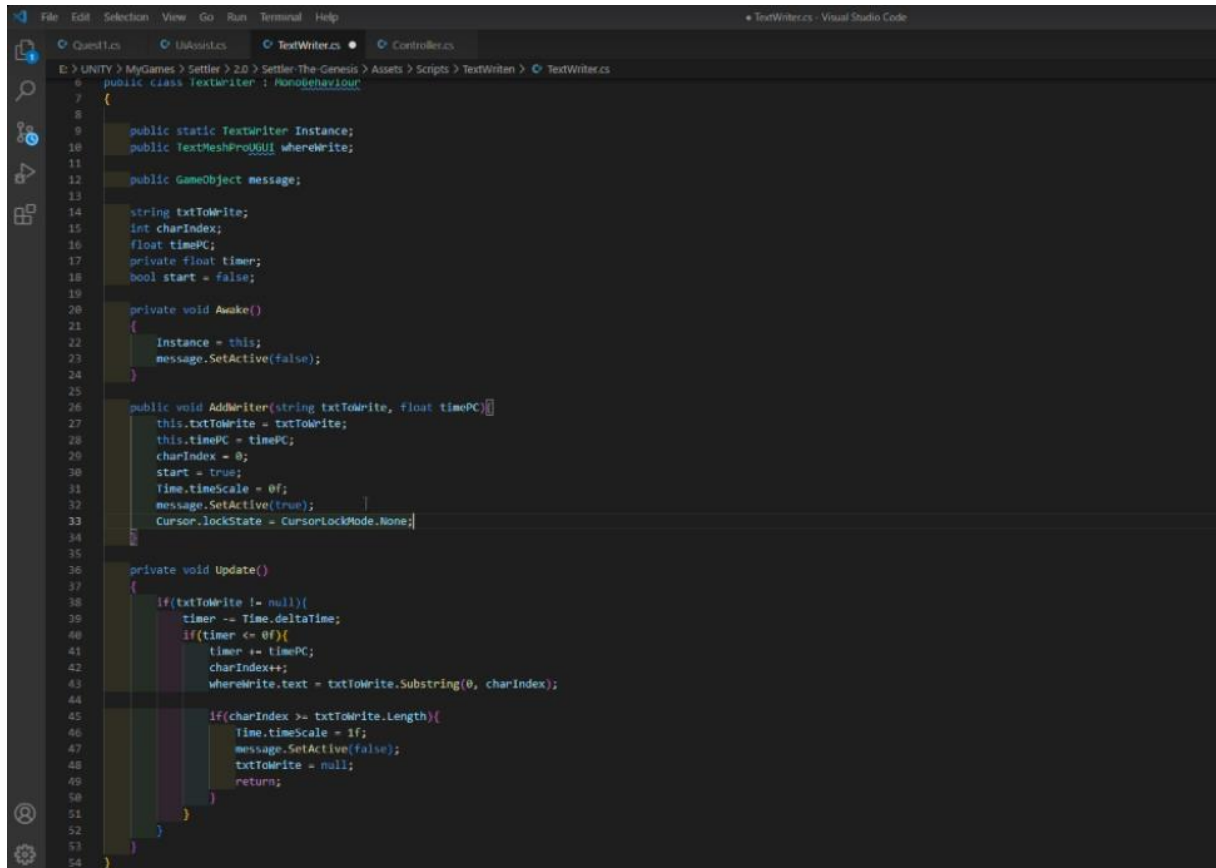
## Mixamo



Ryc. 2 Strona główna witryny Mixamo

Mixamo to narzędzie do tworzenia i animowania 3D, które dostępne jest jako usługa w chmurze. Umożliwia ono tworzenie, edytowanie, animowanie i publikowanie zaawansowanych animacji 3D za pomocą łatwego w użyciu interfejsu. Mixamo ułatwia tworzenie postaci 3D, animacji ruchu i szczegółowej animacji twarzy. Jest wykorzystywany przez wielu twórców gier, animatorów i reżyserów filmowych. W projekcie *Settler The Genesis* posłużyło głównie do animacji gotowych assetów, które nie posiadały kości i gotowych animacji.

## Visual studio code



```
using UnityEngine;
using TMPro;

public class TextWriter : MonoBehaviour
{
    public static TextWriter Instance;
    public TextMeshProUGUI whereWrite;

    public GameObject message;

    string txtToWrite;
    int charIndex;
    float timePC;
    private float timer;
    bool start = false;

    private void Awake()
    {
        Instance = this;
        message.SetActive(false);
    }

    public void AddWriter(string txtToWrite, float timePC)
    {
        this.txtToWrite = txtToWrite;
        this.timePC = timePC;
        charIndex = 0;
        start = true;
        Time.timeScale = 0f;
        message.SetActive(true);
        Cursor.lockState = CursorLockMode.None;
    }

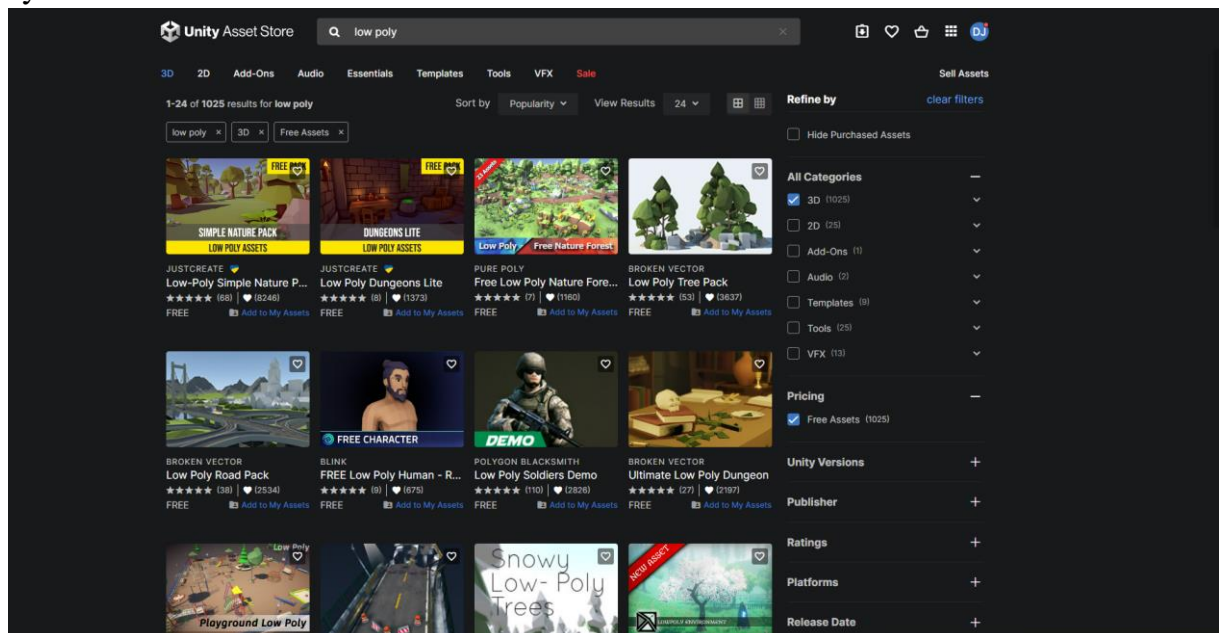
    private void Update()
    {
        if(txtToWrite != null)
        {
            timer -= Time.deltaTime;
            if(timer <= 0f)
            {
                timer += timePC;
                charIndex++;
                whereWrite.text = txtToWrite.Substring(0, charIndex);

                if(charIndex >= txtToWrite.Length)
                {
                    Time.timeScale = 1f;
                    message.SetActive(false);
                    txtToWrite = null;
                    return;
                }
            }
        }
    }
}
```

Ryc. 3 Interfejs graficzny programu Visual Studio Code

Oprogramowanie to posłużyło do obsługi języka C#, wykorzystanego w pełni w kwestiach programistycznych.

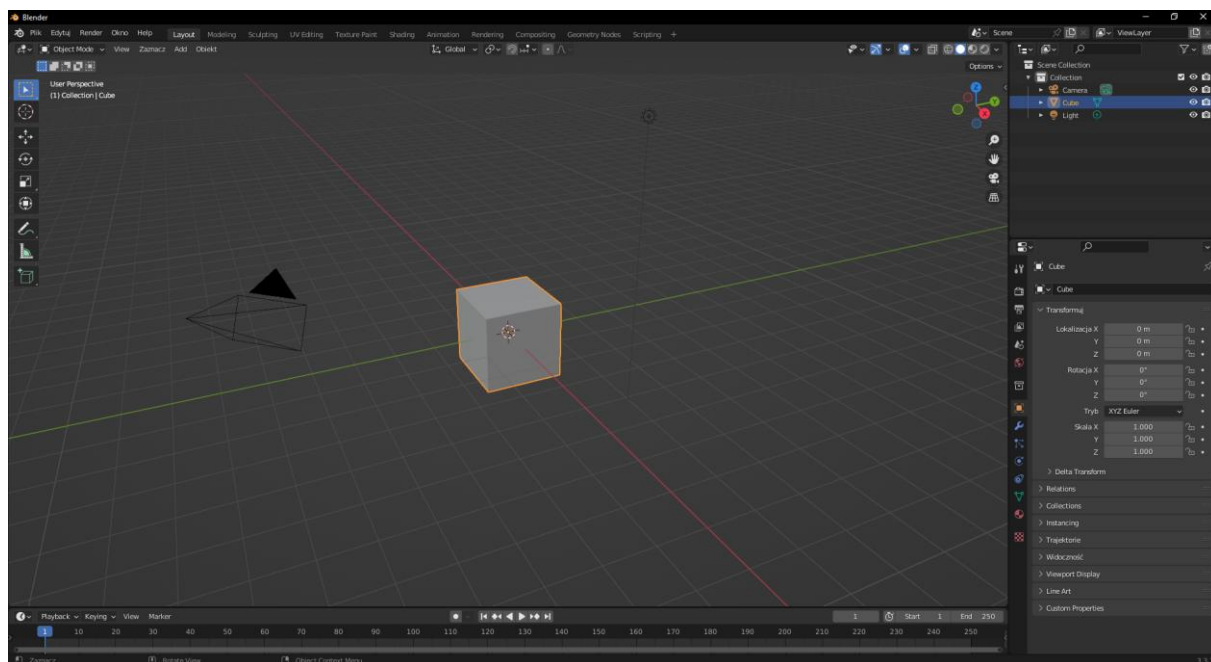
## Unity Assets Store



Ryc. 4 Strona internetowa Unity Assets Store

Strona pozwala na darmowe pobieranie gotowych assetów, zarówno w technologii 2D jak i 3D.

## Blender



Ryc. 5 Interfejs graficzny programu Blender

Blender to narzędzie pozwalające na tworzenie oraz edycję grafiki 3D. To zaawansowane narzędzie jest w stanie obsłużyć zarówno małe projekty, jak i filmy animowane.

## Wymagania sprzętowe

### Minimalne

|                           |  |
|---------------------------|--|
| System operacyjny         | Windows 10   |
| Pamięć RAM                | 4GB  |
| Przestrzeń dyskowa        | 5 GB   |
| Procesor                  | Intel: Core i3-530<br>AMD: A6-3650   |
| Karta graficzna           | NVidia: GeForce 9600GT AMD: HD 6570<br>Intel: zintegrowana karta Intel HD 4600 |
| Karta dźwiękowa           | Zawiera  |
| Zalecana rozdzielczość    | 1024x768   |
| Standard karty graficznej | Sprzęt zgodny z DX10   |

### Rekomendowane

|                           |  |
|---------------------------|--|
| System operacyjny         | Windows 10                                 |
| Pamięć RAM                | 8GB  |
| Przestrzeń dyskowa        | 5 GB                                       |
| Procesor                  | Intel: Core i5-3300<br>AMD: Ryzen 3 1200   |
| Karta graficzna           | NVidia: GeForce 560<br>AMD: Radeon HD 6950 |
| Karta dźwiękowa           | Zawiera                                    |
| Zalecana rozdzielczość    | 1920x1080                                  |
| Standard karty graficznej | Sprzęt zgodny z DX11                       |



## Projekt świata gry

Projekt świata gry składa się z otwartego terenu, w którego skład wchodzi neutralny biome w kolorystyce zielono-żółtej. Utworzony projekt mapy składa się z dwóch części głównych, które oddziela most nad potokiem.



Ryc. 6 Projekt świata - perspektywa ogólna

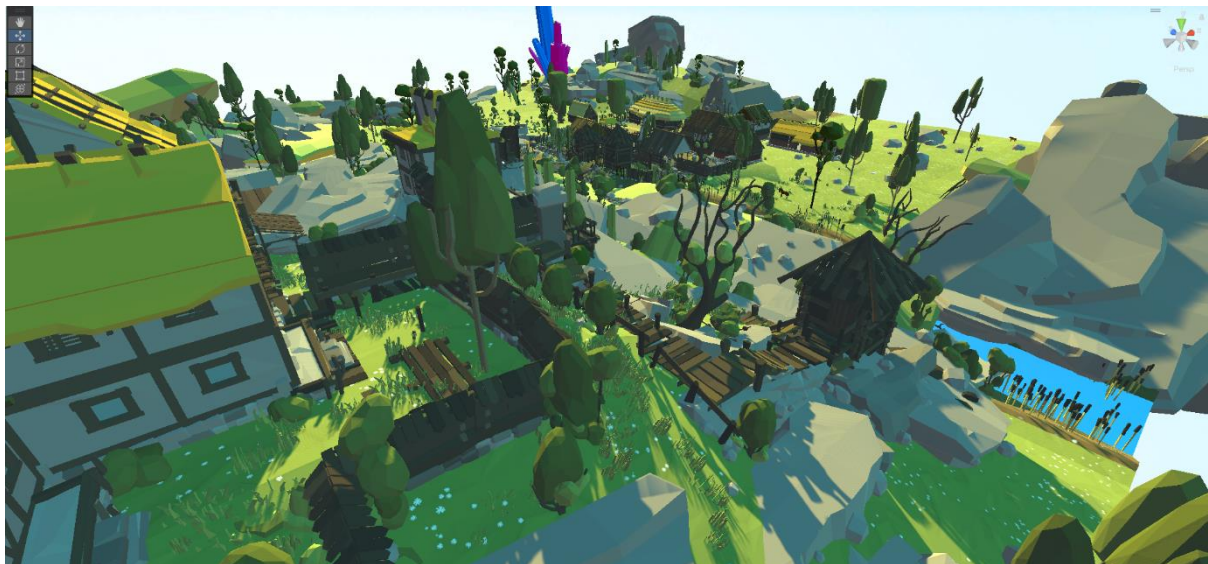
Teren o dużym zrównoważeniu płaszczyzny, zawierający w swoim obszarze kryształ jest głównym obszarem początkowej rozgrywki. To na jego terenie odbywają się wszystkie zadania udostępnione w wersji beta. Obejmuje on taki obszar jak karczma, domy mieszkalne, drzewa, góry, farmę oraz pola uprawne.



Ryc. 7 Projekt świata - perspektywa części fabularnej



Teren po drugiej stronie potoku jest wyłączony spoza aktualnej aktywności gracza. Jest on jedynie udostępniony w celach eksploracyjnych. Nie zawiera jednak żadnych większych elementów interaktywnych. W założeniu, teren ten zostanie przystosowany pod tereny mieszkalne dla gracza.



Ryc. 8 Projekt świata - perspektywa pozostałej części

## Projekt interfejsu gracza

Interfejs graficzny gracza składa się z wielu niewielkich rozmiarów okienek. Gracz ma do dyspozycji główny ekran ekwipunku, którego otwarcie zatrzymuje czas, a za jego pomocą można sprawdzić status zadań do wykonania wraz z ich licznikiem. Każde wykonane zadanie zwiększa licznik o 1 punkt. Do dyspozycji gracza jest również okno tak zwanego plecaka. Znajdują się tam wszystkie podniesione przez gracza obiekty, które są w jakimś stopniu interaktywne np. jedzenie. Takowe obiekty można spożyć – przywracając sobie tym samym w błyskawiczny sposób wytrzymałość. Jeżeli natomiast ekwipunek byłby przepełniony to możemy również usunąć wybrane elementy poprzez wybranie funkcji „Enable Remove” i kliknięcie przycisku X przy przedmiocie, którego gracz chce się pozbyć.



*Ryc. 9 Interfejs aktywnego zadania*



*Ryc. 10 Interfejs ekwipunku*

Aby nawigacja po mapie była łatwiejsza, gracz ma do swojej dyspozycji minimapę, która znajduje się w lewym dolnym rogu ekranu.



*Ryc. 11 Interfejs minimapy*

Użytkownik posiada również dwa paski, które znajdują się w prawym dolnym rogu ekranu. Obrazują one jego stan zdrowia (pasek czerwony), a także poziom jego wytrzymałości (kolor pomarańczowy).



Ryc. 12 Interfejs życia oraz wytrzymałości

## Projekt postaci

### Postać główna

Postacią główną jest starszy mężczyzna (ryc.13). Do swojej dyspozycji posiada on dwie bronie. Broń dystansową, jaką jest łuk, a także broń białą jaką jest miecz.



Ryc. 13 Postać sterowana przez gracza

### Postacie fabularne

W aktualnej wersji zaprezentowanej gry, dostępne są trzy postacie fabularne. Każda ze wskazanych postaci posiada dla gracza jedno zadanie do wykonania. Ze względu na liniowość fabuły, gracz musi odwiedzać postacie w wyznaczonej z góry kolejności.



Ryc. 14 Postacie fabularne

### Postacie niezależne

Do postaci niezależny należą zwierzęta. Nie są one bezpośrednio związane z wątkami fabularnymi, a gracz nie jest w stanie wchodzić z nimi w interakcję (za wyjątkiem walki).



Ryc. 15 Postacie niezależne 1

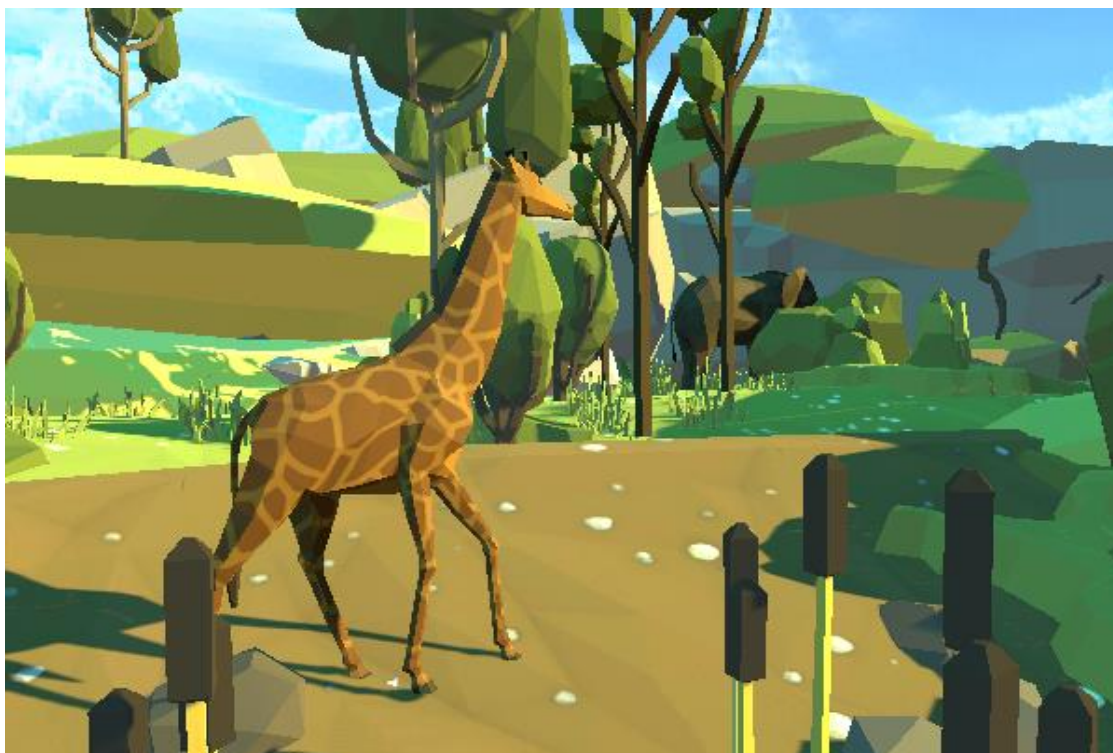


Zaimplementowany świat zwierząt jest bardzo bogaty i posiada wiele modeli zwierząt, wraz z ich spersonalizowanymi animacjami.



*Ryc. 16 Postacie niezależne 2*

Postacie niezależne dzielą się na dwie kategorie: pasywne oraz agresywne. Pasywne są to stworzenia, które nie są w stanie podejmować interakcji z niczym z wyjątkiem broni gracza. Agresywne stworzenia, są w stanie zaatakować inne pasywne zwierzęta.



*Ryc. 17 Postacie niezależne 3*

## Opis fabuły

Fabuła utworzonej gry jest zorientowana liniowo. Jest to spowodowane obszernością projektu. Nieliniowość dodałaby wiele rozgałęzień fabuły, przez co prezentowany materiał znacznie zwiększyłby wymagany nakład pracy oraz samą czasochłonność wykonywanych czynności. Głównym założeniem fabuły miała być lekka i przyjemna historia w tematyce średniowiecza, z lekką dozą poczucia humoru oraz absurdów oraz easter eggów, które urozmaicą rozgrywkę.

## Wątki fabularne

### Quest 1. Prologue - Przynieś miecz dziwnemu nieznajomemu

Fabuła rozpoczyna się gry gracz podejmuje interakcję z postacią, która stoi nieopodal jego lokalizacji. Podczas rozmowy, nieznana postać prosi nas o odnalezienie jego miecza. Po odczytaniu informacji, zadanie zostaje aktywowane, a gracz rusza w poszukiwaniu miecza.

Po odnalezieniu ostrza, gracz ma za zadanie odnieść go do właściciela. Niestety okazuje się że ostrze nie należy do mężczyzny przez co nasz bohater przywłaszcza sobie go dla siebie (od tego momentu postać nosi miecz przy pasku).

Kończąc zadanie, nieznany mężczyzna odsyła nas do karczmy.



Ryc. 18 Początek pierwszego zadania

## Quest 2. Zbieranie pomidorów

Po dotarciu do karczmy, gracz może wejść w interakcję z karczmarką w czerwonych włosach. Ona zleca nam przyniesienie kilku dorodnych pomidorów w celu przygotowania sosu pomidorowego na pizzę. Gracz musi odnaleźć 3 pomidory rozrzucone po mapie. Po odniesieniu ich do karczmarki, gracz otrzymuje kawałek pizzy i zostaje odesłany do rolnika.



Ryc. 19 Początek drugiego zadania



### Quest 3. Poszukiwania strasznego stworzenia

Po odszukaniu rolnika, gracz może aktywować ostatniego dostępnego na ten moment zadania. Polega ono na zlokalizowaniu przyczyny obaw rolnika. Sądzi on, iż w okolicy grasuje straszne stworzenie, które zagraża hodowanym przez niego zwierzętom. Gracz musi zlokalizować to stworzenie i je wyeliminować. Po wykonaniu zadaniu należy powrócić do mężczyzny, który tym samym kończy nasze ostatnie zlecenie.



Ryc. 20 Początek trzeciego zadania

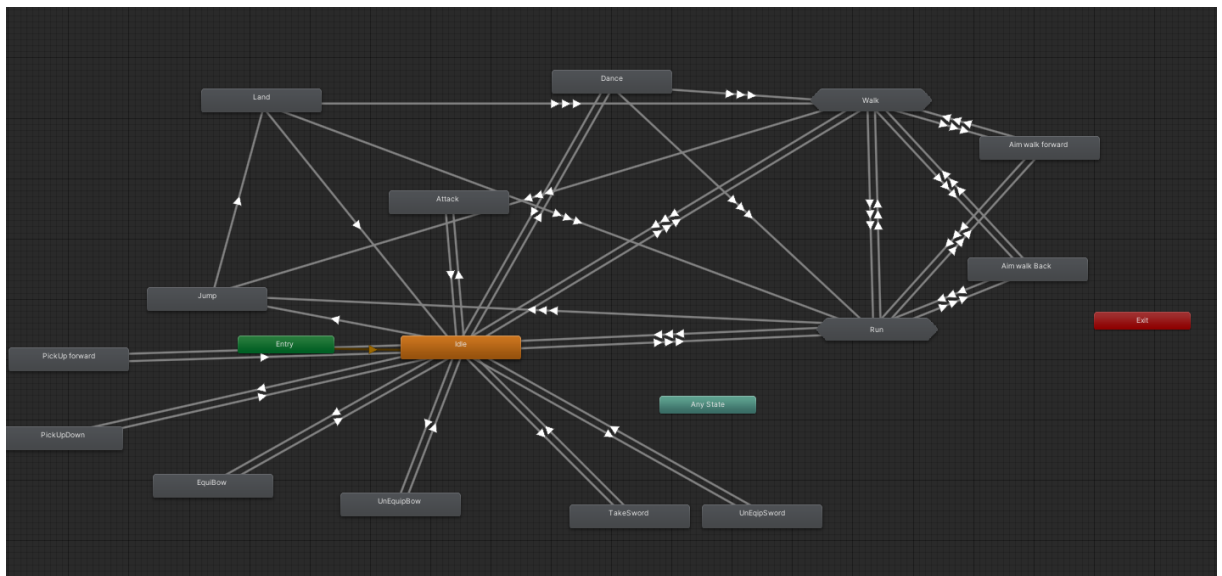
## Przykładowe mechaniki rozgrywki

Gracz do swojej dyspozycji posiada kilka mechanik, które zwiększają interaktywność z grą. Odpowiadają za to kontrolery, które umożliwiają graczowi przemieszczanie się, skakanie, uderzanie, strzelanie z łuku jak i również podnoszenie wybranych przedmiotów.

## Poruszanie się

Do podstawowych mechanik rozgrywki zalicza się podstawową fizykę gry jak i również poruszanie się postaci. Bohater gry *Settlers The Origin*, może wykonywać ruchy do przodu oraz do tyłu, korzystając z klawiszy W oraz S. Aby skręcić lub rozejrzeć się gracz musi skorzystać z myszy bądź innego narzędzia do kontrolowania kursora myszy.

Postać może przełączać się pomiędzy trybami pracy poprzez rozbudowany kontroler ruchu.



Ryc. 21 Główny kontroler ruchu

Gracz może zmieniać prędkość chodzenia poprzez kombinację przycisków Shift + W bądź Shift + S. Wciśnięcie znaków kombinacyjnych wymusza zmianę animacji postaci, a kontroler pomaga zachować jej płynność.



Ryc. 22 Animacja chodzenia

Postać sterowana przez gracza, może również pokonywać trudno dostępne dystanse poprzez zastosowanie funkcji skoku. Odpowiada za nią klawisz spacebar.



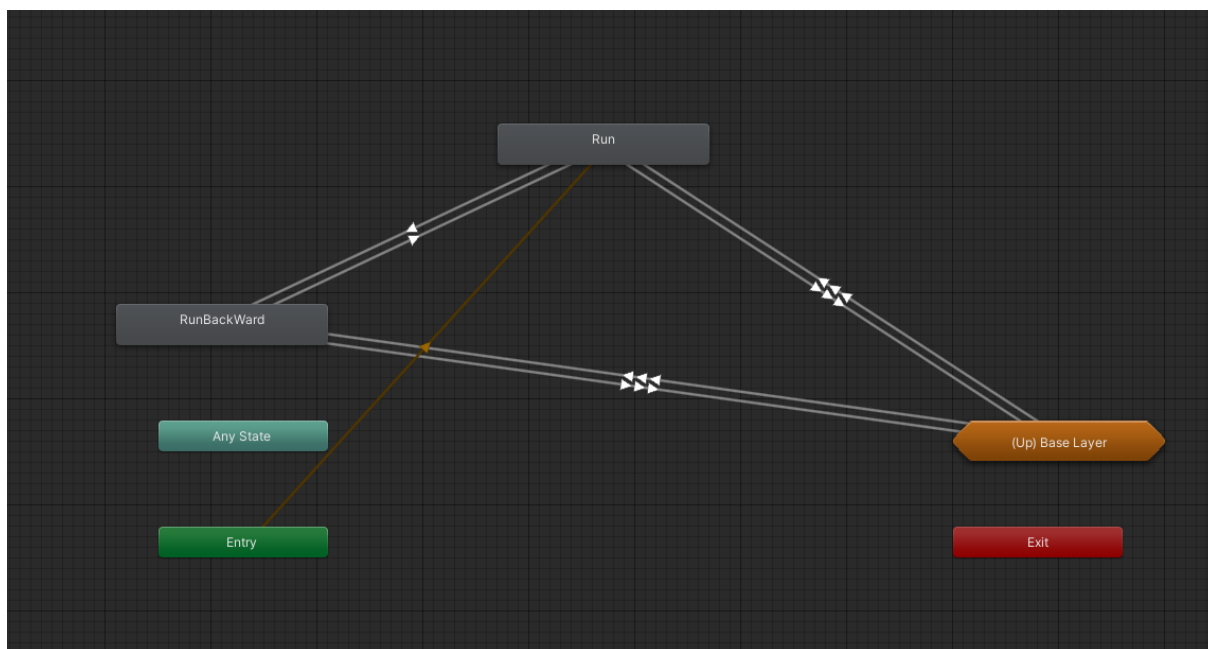
Ryc. 23 Animacja skoku

Postać ta w celach rozrywkowych posiada również ukrytą możliwość wykonania animacji tańca.

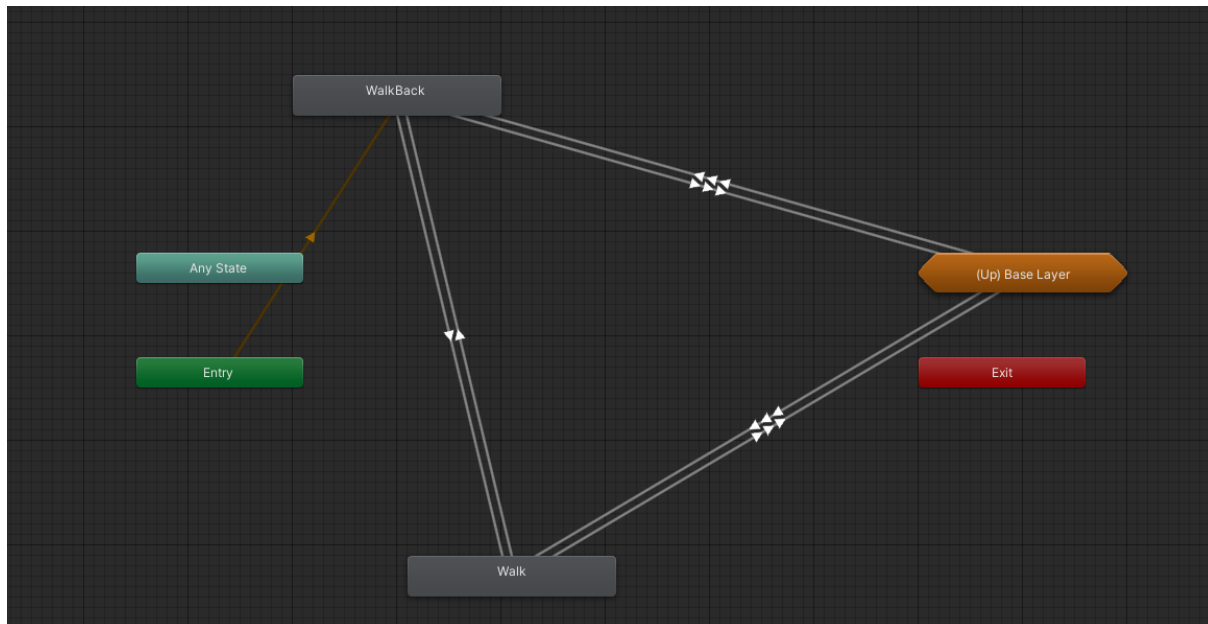


Ryc. 24 Animacja tańca

Poniżej zamieszone zostały również innego typu kontrolery, które zostały zintegrowane z postacią dla jego podstawowych funkcji ruchowych.



Ryc. 25 Kontroler ruchu 2



Ryc. 26 Kontroler ruchu 3

## Interakcje z przedmiotami

Postać sterowana przez gracza może również wchodzić w interakcję nie tylko z postaciami fabularnymi, ale również z niektórymi przedmiotami, które znajdują się w obszarze gry. Do takowych przedmiotów należy miecz oraz jedzenie.

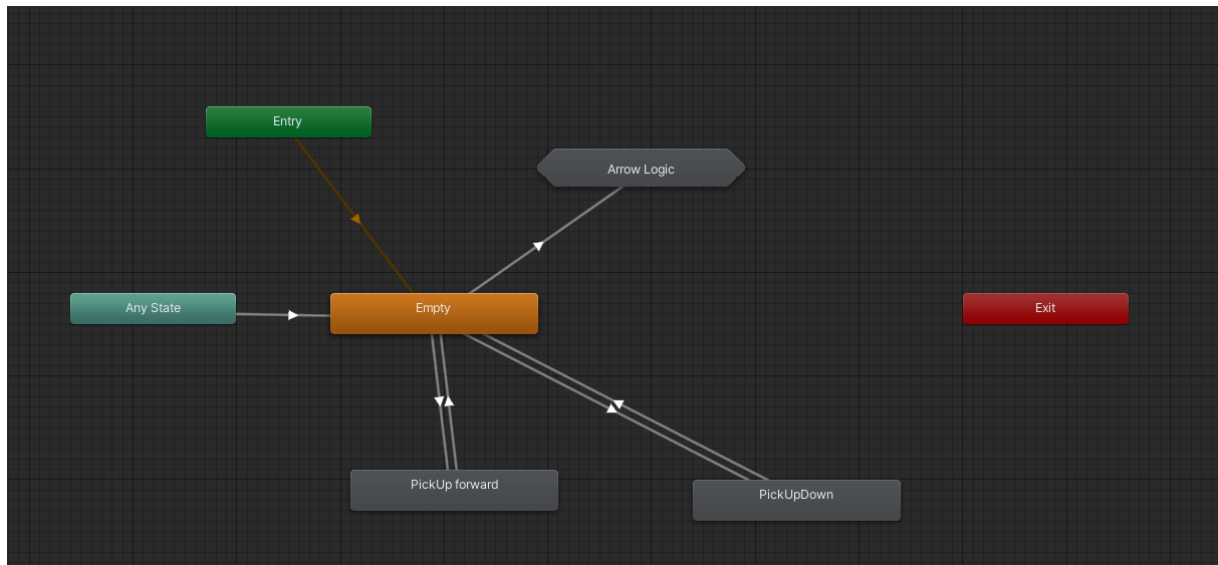


Ryc. 27 Obiekt interaktywny - pomidor



Ryc. 28 Dwa typy animacji podnoszenia obiektów w zależności od ich położenia

Dzięki oskryptowaniu obiektów interaktywnych, postać może dostosować typ wykonanej animacji. Obiekt może znaleźć się przed nami bądź pod nami. W zależności od czynnika wysokości, postać dostosowuje rodzaj uruchamianej animacji.



Ryc. 29 Kontroler podnoszenia obiektów

Zaimplementowana została również możliwość otwierania drzwi. Standardowo są one zamknięte, jednakże po wejściu z nimi w interakcję możemy zmienić ich kont położenia przez co zostają otwarte, a gracz może wejść do pomieszczenia.



Ryc. 30 Otwieranie drzwi



## Atak mieczem

Po zakończeniu pierwszego zadania, bohater gry wchodzi w posiadanie miecza, z którego może korzystać przeciwko postacią niezależnym. Atak mieczem posiada własną animację. Gracz ma możliwość przełączania się pomiędzy bronią za pomocą klawiszy 1 oraz 2.



Ryc. 31 Animacja ataku mieczem

## Strzelanie z łuku

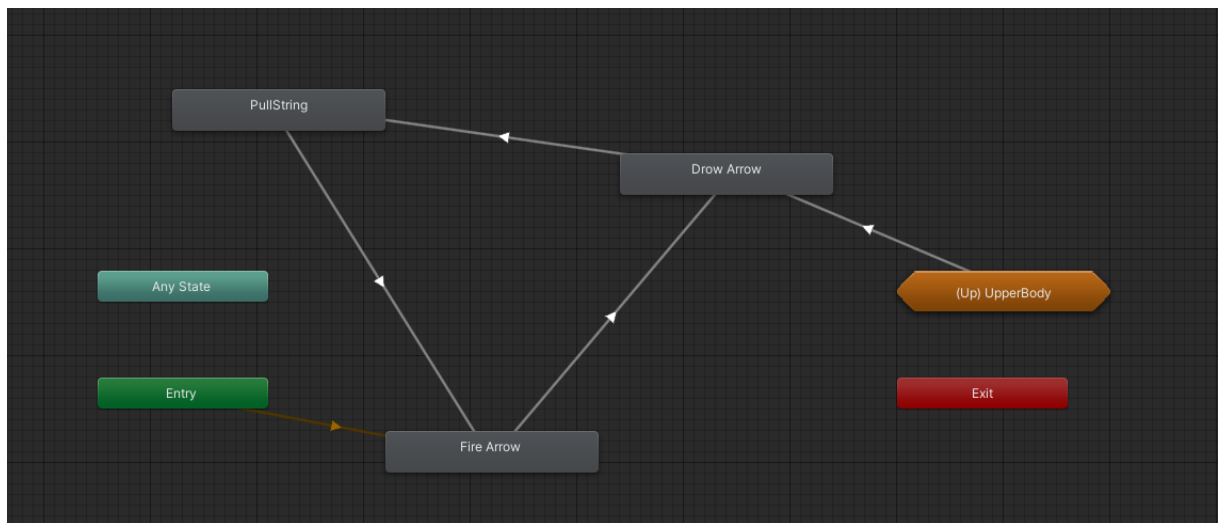
Drugą bronią należącą do gracza jest łuk. Jest to zdecydowanie lepsza broń niżeli miecz. Posiada ona lepszy zasięg, a sama ilość strzał nie jest w żaden sposób ograniczona. Podczas mierzenia z broni (prawy przycisk myszy), kamera zostaje zbliżona do postaci, aby cel był bardziej widoczny, stosując tym samym lekki stopień skupienia. Po wystrzeleniu strzały, zostaje ona wbita w najbliższy możliwy cel, który posiada funkcję kolizji obiektów.



Ryc. 32 Animacja strzelania z łuku



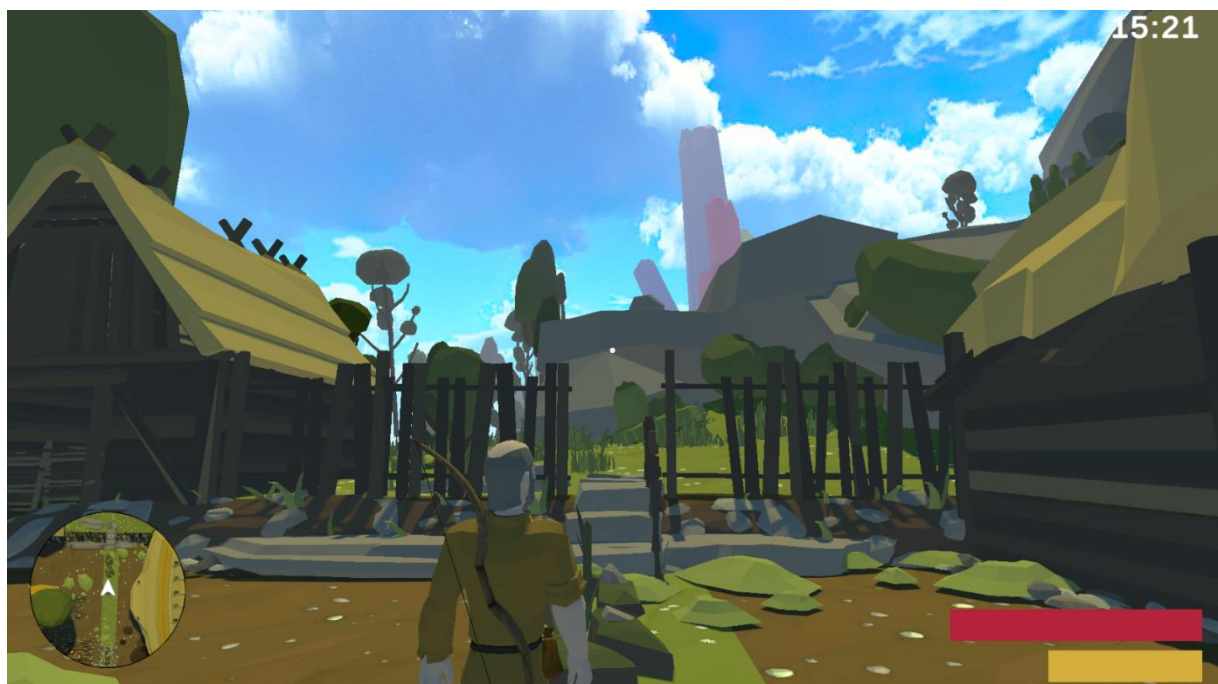
Ryc. 33 Strzala wbita w ścianę



Ryc. 34 Kontroler obsługujący strzelanie z łuku

## Pora dnia

Utworzona gra oferuje również system mierzenia czasu. Po upływie odpowiedniej godziny zostaje zaimplementowany kod, który wymusza zmianę pory dnia z dnia na noc i odwrotnie.



Ryc. 35 Pora dnia - dzień



Ryc. 36 Pora dnia - noc

### Poziom wytrzymałość postaci

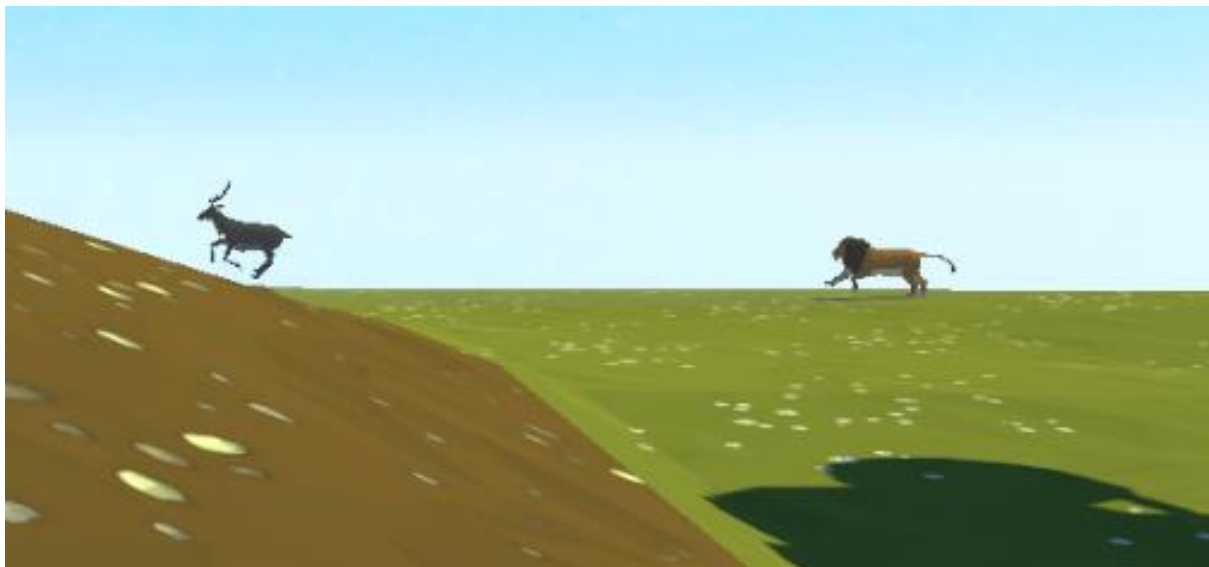
Postać posiada realnie działający pasek wytrzymałości. Wytrzymałość tracona jest poprzez skrajny wysiłek fizyczny. Gdy gracz biega lub skacze, to poziom wytrzymałości spada. Gdy spadnie do zera, gracz zaczyna tracić życie. Pasek ten można szybko odzyskać poprzez zjedzenie jakiegoś pożywienia. Jest możliwość również odzyskać wytrzymałość po upływie dłuższego okresu czasu – jest on automatycznie regenerowany.



Ryc. 37 Paski zdrowia oraz wytrzymałości

## AI postaci niezależnych

Postacie niezależne dzielą się na pasywne oraz agresywne. Pasywne zwierzęta nie wchodzi w interakcję z innymi lub od nich uciekają. Agresywne natomiast szukają ofiary, a gdy jest ona niedaleko to stara się ją zaatakować.



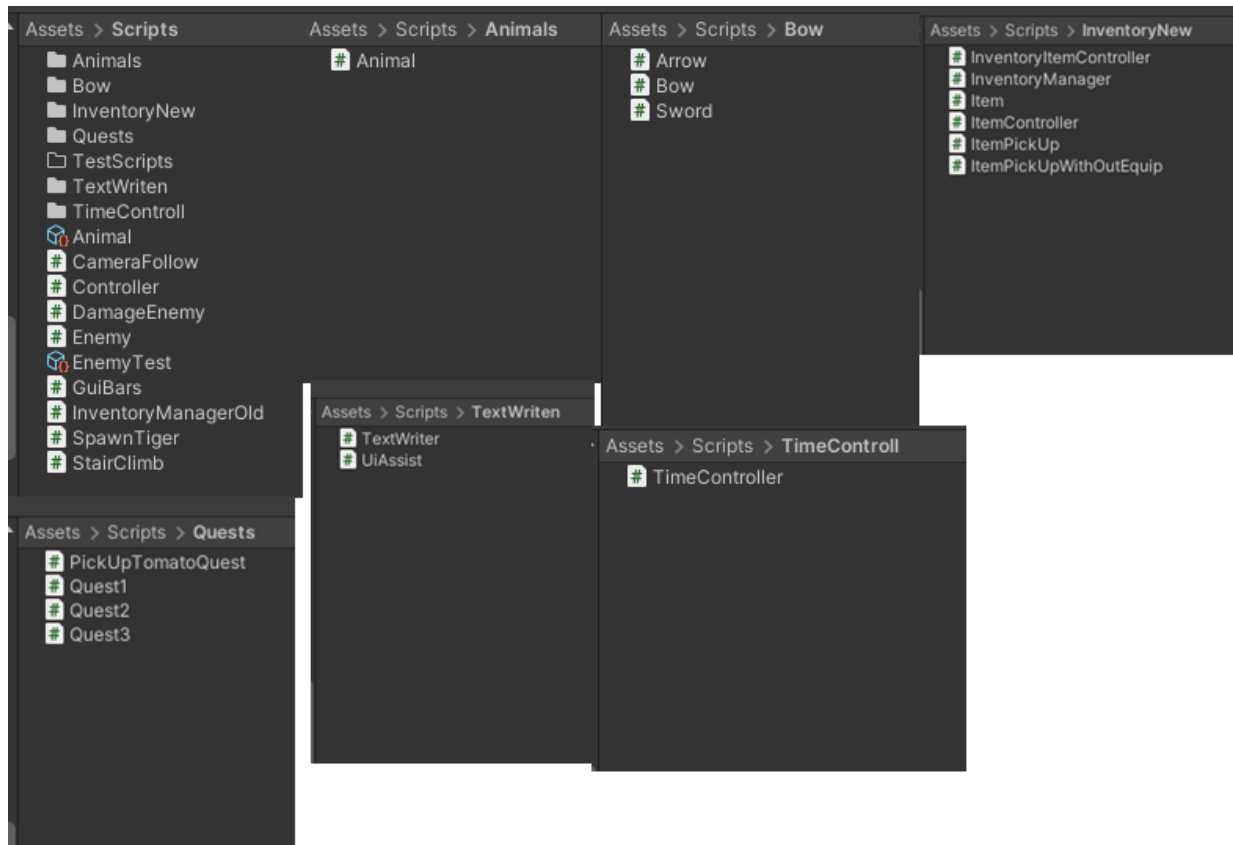
*Ryc. 38 Atak lwa na jelenia*



*Ryc. 39 Walka pomiędzy tygrysami*

## Zaimplementowane skrypty

Wszystkie skrypty zastosowane w projekcie są autorstwa twórców gry, a część kodu wraz z jego rozmieszczeniem została przedstawiona poniżej.



Ryc. 40 Schemat rozmieszczenia autorskich skryptów

Poniżej zaprezentowane zostały części kodu oprogramowania, które służą do zadawania obrażeń postaciom, a także do obsługi różnych stanów zwierząt (tj. poruszania się).

```
public void TakeDamage(float damage)
{
    toughness -= damage;
    if (toughness <= 0f)
    {
        Die();
        Destroy(gameObject, 20f);
    }
}
public void Die()
{
    SetState(WanderState.Dead);
}
```

Ryc. 41 Skrypt zadawania obrażeń



```

        targetPosition = position + Vector3.ProjectOnPlane(position - primaryPursuer.transform.position, Vector3.up);
        if (!IsValidLocation(targetPosition))
        {
            targetPosition = startPosition;
            ValidatePosition(ref targetPosition);
            FaceDirection((targetPosition - position).normalized);
            stamina -= Time.deltaTime;
            if (stamina <= 0f)
            {
                UpdateAI();
                break;
            }
        }
        case WanderState.Wander:
        {
            stamina = Mathf.MoveTowards(stamina, stats.stamina, Time.deltaTime);
            targetPosition = wanderTarget;
            Debug.DrawLine(position, targetPosition, Color.yellow);
            FaceDirection((targetPosition - position).normalized);
            var displacementFromTarget = Vector3.ProjectOnPlane(targetPosition - transform.position, Vector3.up);
            if (displacementFromTarget.magnitude < contingencyDistance)
            {
                SetState(WanderState.Idle);
                UpdateAI();
            }
        }
        break;
        case WanderState.Idle:
        {
            stamina = Mathf.MoveTowards(stamina, stats.stamina, Time.deltaTime);
            if (Time.time >= idleUpdateTime)
            {
                SetState(WanderState.Wander);
                UpdateAI();
            }
        }
        break;
    }

    if (navMeshAgent)
    {
        navMeshAgent.destination = targetPosition;
        navMeshAgent.speed = moveSpeed;
        navMeshAgent.angularSpeed = turnSpeed;
    }
    else
    {
        characterController.SimpleMove(moveSpeed * UnityEngine.Vector3.ProjectOnPlane(targetPosition - position, Vector3.up).normalized);
    }
}

```

Ryc. 42 Fragment skryptu do zmiany stanu zwierząt

Zaprezentowany powyżej kod obsługuje sztuczną inteligencję, która w czasie rzeczywistym steruje postaciami niezależnymi. Zwierzęta, które zostały objęte działaniem skryptów mogą przechodzić pomiędzy stanami takimi jak np. brak czynności, swobodne poruszanie się, bieganie oraz atakowanie. Stany te zależne są od wielu czynników, chociażby od odległości postaci niezależnej od potencjalnej ofiary lub oprawcy.



## Podział obowiązków

| Patryk Arendt  | Damian Jamróży  |
|--|---|
| <ul style="list-style-type: none"><li>• Pomoc przy opracowywaniu zakresu prac nad projektem. (Dokumentacja)</li><li>• Wsparcie merytoryczne przy tworzeniu questów.</li><li>• Dostosowywanie postaci do stworzonego świata gry.</li><li>• Wdrożenie pozyskanych assetów.</li><li>• Oskryptowanie zaawansowanych mechanik gry, takich jak okna dialogowe, strzelanie z łuku itp.</li><li>• Wygenerowanie animacji postaci.</li><li>• Implementacja pory dnia.</li><li>• Wykonanie poradnika dotyczącego oprogramowania make human oraz maximo</li><li>• Prezentacja gameplaya</li></ul> | <ul style="list-style-type: none"><li>• Opracowanie głównego zakresu prac.</li><li>• Przygotowanie oraz nadzór nad repozytorium github.</li><li>• Przygotowanie zarysu fabuły.</li><li>• Opracowanie questów.</li><li>• Pozyskanie assetów terenu oraz postaci.</li><li>• Skompletowanie dokumentacji projektu.</li><li>• Implementacja świata gry.</li><li>• Oskryptowanie fizyki gry.</li><li>• Pomoc przy opracowaniu poradnika dotyczącego oprogramowania make human oraz maximo</li><li>• Nagranie gameplaya</li></ul> |

## Spis ilustracji

|  |    |
|--|----|
| Ryc. 1 Interfejs programu Unity Hub .....  | 2  |
| Ryc. 2 Strona główna witryny Mixamo .....  | 3  |
| Ryc. 3 Interfejs graficzny programu Visual Studio Code .....                       | 4  |
| Ryc. 4 Strona internetowa Unity Assets Store .....                                 | 4  |
| Ryc. 5 Interfejs graficzny programu Blender .....                                  | 5  |
| Ryc. 6 Projekt świata - perspektywa ogólna .....                                   | 7  |
| Ryc. 7 Projekt świata - perspektywa części fabularnej .....                        | 7  |
| Ryc. 8 Projekt świata - perspektywa pozostałej części .....                        | 8  |
| Ryc. 9 Interfejs aktywnego zadania.....  | 9  |
| Ryc. 10 Interfejs ekwipunku.....   | 9  |
| Ryc. 11 Interfejs minimapy .....   | 9  |
| Ryc. 12 Interfejs życia oraz wytrzymałości .....                                   | 10 |
| Ryc. 13 Postać sterowana przez gracza .....  | 10 |
| Ryc. 14 Postacie fabularne .....   | 11 |
| Ryc. 15 Postacie niezależne 1 .....  | 11 |
| Ryc. 16 Postacie niezależne 2 .....  | 12 |
| Ryc. 17 Postacie niezależne 3 .....  | 12 |
| Ryc. 18 Początek pierwszego zadania.....   | 13 |
| Ryc. 19 Początek drugiego zadania.....   | 14 |
| Ryc. 20 Początek trzeciego zadania .....   | 15 |
| Ryc. 21 Główny kontroler ruchu .....   | 16 |
| Ryc. 22 Animacja chodzenia.....  | 17 |
| Ryc. 23 Animacja skoku.....  | 18 |
| Ryc. 24 Animacja tańca.....  | 18 |
| Ryc. 25 Kontroler ruchu 2 .....  | 19 |
| Ryc. 26 Kontroler ruchu 3 .....  | 19 |
| Ryc. 27 Obiekt interaktywny - pomidor.....   | 20 |
| Ryc. 28 Dwa typy animacji podnoszenia obiektów w zależności od ich położenia ..... | 20 |
| Ryc. 29 Kontroler podnoszenia obiektów .....                                       | 21 |
| Ryc. 30 Otwieranie drzwi.....  | 21 |
| Ryc. 31 Animacja ataku mieczem .....   | 22 |
| Ryc. 32 Animacja strzelania z łuku.....  | 23 |
| Ryc. 33 Strzała wbita w ścianę.....  | 23 |
| Ryc. 34 Kontroler obsługujący strzelanie z łuku.....                               | 24 |
| Ryc. 35 Pora dnia - dzień .....  | 24 |
| Ryc. 36 Pora dnia - noc .....  | 25 |
| Ryc. 37 Paski zdrowia oraz wytrzymałości.....                                      | 25 |
| Ryc. 38 Atak lwa na jelenia.....   | 26 |
| Ryc. 39 Walka pomiędzy tygrysami .....   | 26 |

## **Netografia**

- 1) <https://learn.unity.com/tutorials>
- 2) <https://www.mixamo.com>
- 3) <https://assetstore.unity.com/>
- 4) <https://www.blender.org/support/tutorials/>