

GRA SIECIOWA FPS Z UŻYCIEM SILNIKA UNITY

Mateusz Urban, Grzegorz Zych

*Uniwersytet Kazimierza Wielkiego
Wydział Informatyki
ul. Mikołaja Kopernika 1, 85-074 Bydgoszcz
e-mail: mateusz.urban@student.ukw.edu.pl*

Streszczenie: Niniejsza praca dotyczy opracowania sieciowej gry FPS (First-Person Shooter) przy użyciu środowiska Unity. Praca zawiera szczegółowy opis gry, projekt aplikacji, model bazy danych oraz diagramy interakcji. Omówiono także napotkane problemy podczas tworzenia gry, a także przedstawiono zastosowane technologie i wykorzystane narzędzia.

Słowa kluczowe: Unity, gra sieciowa, gra wieloosobowa, moce, strzelanka

ONLINE FPS GAME CREATED WITH UNITY ENGINE

Abstract: This paper focuses on the development of an online FPS (First-Person Shooter) game using the Unity environment. It includes a detailed description of the game, application design, database model, and interaction diagrams. Additionally, it discusses the challenges encountered during the development process and presents the applied technologies and utilized tools.

Keywords: Unity, online game, multiplayer game, powers, shooter

1. WSTĘP

Gry komputerowe stanowią istotny element współczesnej kultury i gospodarki, odgrywając kluczową rolę w sektorze rozrywki oraz technologii informatycznych. Są zjawiskiem wielowymiarowym, które przyciąga uwagę zarówno naukowców, jak i przedsiębiorców ze względu na ich potencjał ekonomiczny, społeczny oraz edukacyjny. Ich popularność wynika z możliwości zapewnienia immersyjnych doświadczeń, szerokiej dostępności technologicznej oraz różnorodności tematycznej, co czyni je medium o ogromnym zasięgu oddziaływania [1-3].

Współczesna branża gier komputerowych rozwija się dynamicznie i ma znaczący wpływ na globalny rynek. Prognozy wskazują na jej dalszy wzrost, napędzany postępem technologicznym oraz rosnącym zainteresowaniem grami jako formą spędzania wolnego czasu. Złożoność tej branży obejmuje nie tylko tworzenie i dystrybucję gier, ale także ich analizę pod kątem aspektów społecznych i psychologicznych [4-6].

Gry takie jak "Minecraft", "Lethal Company" czy "Stardew Valley" dowodzą, że nawet proste mechaniki mogą zdobyć szeroką popularność i zaangażować miliony graczy [7-9].

Celem niniejszej pracy było opracowanie projektu gry "Heir of Death", będącej próbą połączenia klasycznych elementów strzelanek pierwszoosobowych (FPS) z nowoczesnymi rozwiązaniami w zakresie projektowania rozrywki. Kluczowym założeniem było stworzenie gry o niskich wymaganiach sprzętowych, co pozwala na dotarcie do szerokiego grona odbiorców, zapewniając jednocześnie angażujące doświadczenie oparte na rywalizacji i strategii. Gra łączy łatwość przyswojenia podstawowych zasad z głębią rozrywki, wymagającą czasu i zaangażowania gracza.

Zakres projektu obejmuje szczegółowe określenia wymagań funkcjonalnych i technicznych gry, analizę narzędzi wspierających proces tworzenia oraz implementację kluczowych komponentów, takich jak

grafika, mechanika rozgrywki, interfejs użytkownika oraz systemy dźwiękowe. W projekcie uwzględniono również proces testowania, diagnozowania błędów i ich eliminacji, co pozwoliło na stworzenie stabilnego prototypu.

"Heir of Death" czerpie inspiracje z klasycznych tytułów, takich jak "Quake" i "Unreal Tournament", jednocześnie wprowadzając nowoczesne elementy rozgrywki znane z gier typu "Apex Legends". Dzięki temu projekt łączy dynamiczną, pełną akcji rozgrywkę z nowoczesnymi mechanikami, oferując unikalne doświadczenie dla graczy.

2. Oprogramowanie oraz biblioteki wykorzystane do tworzenia gry

2.1. Oprogramowanie

Projekt gry został oparty na Unity 2022.3.10f1, które oferuje kompleksowy zestaw narzędzi do tworzenia gier, w tym renderowanie 3D, animację, skryptowanie, symulację fizyki oraz zarządzanie dźwiękiem [2, 10-14]. Dzięki swojej wszechstronności i obszernej dokumentacji Unity jest preferowanym wyborem zarówno dla początkujących, jak i doświadczonych twórców.

Do pisania, debugowania i testowania skryptów w języku C# wykorzystano zintegrowane środowisko programistyczne Visual Studio [15-16]. Jego rozbudowane funkcje ułatwiły implementację złożonej logiki gry oraz zapewniły wysoką jakość kodu.

W projekcie użyto również open-source'owego oprogramowania do modelowania 3D – Blendera. Posłużył on do tworzenia szczegółowych modeli broni, postaci i elementów środowiska. Dzięki swojej wszechstronności i bogatemu zestawowi funkcji Blender umożliwił autorom projektowanie atrakcyjnych wizualnie i zoptymalizowanych modeli 3D [17].

Do projektowania komponentów interfejsu użytkownika, takich jak ikony i menu, wykorzystano edytor grafiki wektorowej Inkscape. Charakteryzuje się on wysoką precyzją i skalowalnością, co było kluczowe dla stworzenia przejrzystego i intuicyjnego interfejsu [18].

Za dodanie efektów dźwiękowych odpowiadał program Audacity – darmowy i otwarty edytor audio. Umożliwił on nagrywanie, edycję i poprawę efektów dźwiękowych w grze. Jego prostota i funkcjonalność pozwoliły zespołowi na

tworzenie wysokiej jakości zasobów audio, wzbogacających immersję w grze [19].

2.2. Biblioteki sieciowe

W projekcie wykorzystano Netcode for GameObjects – zintegrowaną z Unity bibliotekę zaprojektowaną do tworzenia gier wieloosobowych. Narzędzie to odegrało kluczową rolę w implementacji synchronizacji stanu gry w czasie rzeczywistym między klientami w trybie multiplayer.

Z kolei PlayFab [20], platforma backendowa dostarczana przez Microsoft, została użyta do uwierzytelniania użytkowników, przechowywania danych oraz zarządzania rankingami. Jej zastosowanie usprawniło integrację kluczowych funkcji sieciowych i zwiększyło skalowalność komponentów online gry.

3. Koncepcja i projekt gry

3.1. Aspekty gry

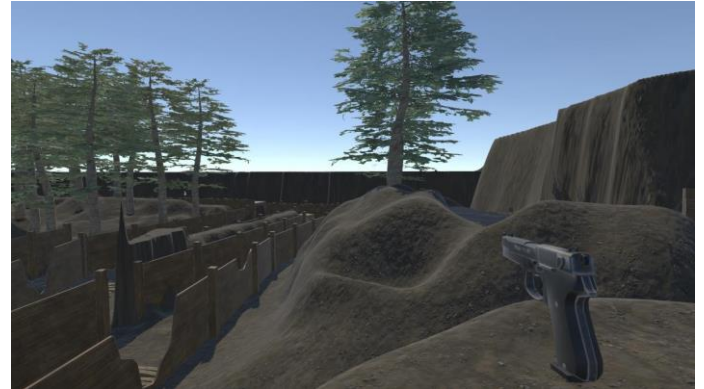
Faza koncepcyjna obejmowała definiowanie kluczowych elementów gry, tak aby były zgodne zarówno z wizją zespołu, jak i preferencjami docelowej grupy odbiorców. Proces projektowania skupił się na następujących aspektach:

- Tryby gry

W pierwszej kolejności opracowano tryb treningowy dla jednego gracza, który pozwala użytkownikom zapoznać się z mechaniką gry, mapami oraz sterowaniem, bez konieczności połączenia z Internetem. Następnie zaprojektowano tryb wieloosobowy, obejmujący konkurencyjny format 1v1, skoncentrowany na strategii i umiejętnościach graczy.



Rys. 1. Menu główne gry



Rys. 3 Mapa "Okopy"

- Mapy
 - "Las" – gęste środowisko leśne z drzewami, rozrzuconymi budynkami oraz strategicznymi punktami ukrycia.
 - "Okopy" – poligon bitewny w klimacie wojennym, składający się z połączonych okopów, sprzyjający taktycznej rozgrywce.



Rys. 2. Mapa "Las"

- Mechanika rozgrywki
 - Broń – arsenał obejmuje karabiny snajperskie, karabiny maszynowe i pistolety, zróżnicowane pod względem obrażeń, szybkostrzelności oraz prędkości przeładowania.
 - Umiejętności – gracze mogą wybierać spośród czterech umiejętności: widzenie przez ściany, teleportacja, niewidzialność oraz ciche poruszanie się, co pozwala na uzyskanie strategicznej przewagi podczas meczu.

3.2. Proces tworzenia gry

Proces tworzenia i testowania gry został podzielony na kilka kluczowych etapów:

1. Planowanie i podział ról

Projekt został rozbity na mniejsze, zarządzalne zadania, które przypisano członkom zespołu zgodnie z ich specjalizacjami. Wśród ról znalazły się m.in.:

- modelowanie 3D,
- skryptowanie mechanik rozgrywki,
- projektowanie interfejsu użytkownika,
- implementacja funkcji sieciowych.

2. Implementacja logiki gry

Prace rozpoczęto od opracowania diagramów UML, które przedstawiały relacje między kluczowymi elementami gry, takimi jak Gracz, Broń, Umiejętności i Mapa.

Następnie opracowano skrypty obsługujące podstawowe mechaniki, w tym:

- ruch gracza,
- strzelanie,
- zarządzanie zdrowiem,
- aktywację umiejętności.

Mechanizm strzelania uwzględniał obliczanie trajektorii pocisków, obrażenia zależne od stref trafienia oraz ich synchronizację między klientami w trybie multiplayer.

3. Implementacja funkcji sieciowych

Netcode for GameObjects zapewnił płynną synchronizację działań graczy w trybie wieloosobowym, takich jak ruch, strzelanie i używanie umiejętności.

PlayFab został zintegrowany w celu zarządzania uwierzytelnianiem użytkowników, danymi kont oraz rankingami, co usprawniło przechowywanie statystyk graczy i zapewniło bezpieczny dostęp do funkcji online.

4. Testowanie i debugowanie

Testowanie iteracyjne prowadzono przez cały cykl produkcji, skupiając się na identyfikacji błędów, balansowaniu mechanik rozgrywki oraz optymalizacji wydajności.

Scenariusze testowe obejmowały zarówno sesje lokalne, jak i online, aby zagwarantować stabilność i responsywność rozgrywki.

Gra była testowana na komputerach o różnej konfiguracji sprzętowej, co pozwoliło na wykrycie wąskich gardeł wydajności i optymalizację pod kątem różnych systemów.

4. Funkcjonalność gry „Heir of Death”

„Heir of Death” to sieciowa strzelanka FPS, oferująca dynamiczną rozgrywkę dla dwóch graczy. Gra wyróżnia się prostym i intuicyjnym interfejsem, zróżnicowanymi mapami, szerokim wyborem umiejętności i broni oraz systemem rankingowym, który dodaje rywalizacyjny charakter rozgrywki.

Tryb wieloosobowy i formuła rozgrywki:

- Pojedynki 1 vs 1 – gracze rywalizują bezpośrednio ze sobą w dynamicznych starciach.
- System „Best of X Rounds” – mecz składa się z określonej liczby rund, a zwycięzcą zostaje gracz,

który jako pierwszy osiągnie wymaganą liczbę wygranych.

- Limit czasowy rundy – każda runda trwa określony czas, co zwiększa presję i wymusza szybkie decyzje.
- Restartowanie rundy – jeśli żaden z graczy nie wyeliminuje przeciwnika w regulaminowym czasie, runda zostaje zrestartowana, co zapewnia równe szanse na zwycięstwo.

Mapy

W grze dostępne są dwie mapy:

- Las – gęsto zalesione środowisko z licznymi drzewami i kilkoma budynkami, oferujące strategiczne punkty ukrycia.
- Okopy – mapa inspirowana polem bitwy, wyposażona w system połączonych okopów, sprzyjających taktycznej rozgrywce.

Obie mapy mają ograniczony obszar gry, co zapobiega nadmiernemu oddalaniu się graczy i zwiększa intensywność starć na bliskim dystansie.

Umiejętności:

Gracze mogą wybierać spośród czterech umiejętności, które zapewniają dodatkowe możliwości taktyczne:

- Widzenie przez ściany – podświetla sylwetkę przeciwnika, umożliwiając jego lokalizację nawet za przeszkodami.
- Teleportacja – pozwala na natychmiastowe przeniesienie się w inne miejsce na mapie, co może być wykorzystane zarówno ofensywnie, jak i defensywnie.
- Niewidzialność – ukrywa model gracza, utrudniając jego wykrycie i umożliwiając zaskakujące ataki lub unikanie starć.
- Ciche poruszanie się – eliminuje dźwięki kroków, co pozwala na niezauważone podejście do przeciwnika. Umiejętność ta jest szczególnie skuteczna w połączeniu z niewidzialnością.



Rys. 4. Wybór mocy/umiejętności

Bronie:

W grze dostępnych jest pięć rodzajów broni, dostosowanych do różnych stylów gry:

- Karabiny wyborowe (2) – szybkie, automatyczne lub półautomatyczne. Skuteczne na długim dystansie, ale wymagające precyzyjnego celowania.
- Karabiny maszynowe (2) – automatyczne, charakteryzujące się wysoką szybkostrzelnością. Idealne na bliski i średni dystans, lecz posiadają duży rozrzut.
- Pistolet (1) – półautomatyczny, o średniej szybkostrzelności. Pełni rolę broni pomocniczej, skutecznej na krótkim dystansie.

System rankingowy:

Ranking oparty jest na platformie PlayFab, zapewniającej backendowe usługi dla gier wieloosobowych. W rankingu uwzględniono następujące informacje:

- Nazwa gracza – pozwala na identyfikację użytkowników.
- Liczba zwycięstw – informuje o liczbie wygranych meczów.
- Liczba porażek – wskazuje na liczbę przegranych starć.

System rankingowy dodaje do gry element rywalizacji, motywując graczy do poprawy swoich wyników i wspinania się po drabince liderów.

Interfejs użytkownika (UI):

Interfejs gry został zaprojektowany z myślą o intuicyjności, co zapewnia łatwą nawigację i wygodne korzystanie z dostępnych funkcji. Składa się z następujących ekranów:

- Ekran powitalny – umożliwia logowanie, rejestrację oraz resetowanie hasła.
- Menu główne – zapewnia dostęp do trybów gry, ustawień, rankingu i innych funkcji.
- Ekran wyboru poziomu i umiejętności – pozwala graczom wybrać mapę oraz umiejętność przed rozpoczęciem meczu.
- Ekran wyboru postaci – umożliwia wybór postaci, którą gracz będzie sterować.
- Ekran rankingu – prezentuje listę najlepszych graczy.
- Ekran ustawień – pozwala na dostosowanie parametrów gry do własnych preferencji.
- HUD (Head-Up Display) – wyświetla kluczowe informacje podczas rozgrywki, takie jak:
 - Pasek zdrowia
 - Czas rundy
 - Wybrana umiejętność i czas jej odnowienia
 - Aktualna liczba nabojojów
- Ekran wygranej/przegranej/pauzy – pojawia się po zakończeniu rundy lub w trakcie przerwy w grze.

Dodatkowe mechaniki gry:

- System zadawania obrażeń – określa wartość obrażeń dla każdej broni i umiejętności.
- Interaktywne elementy map – gracze mogą wchodzić w interakcję z wybranymi obiektami, np. apteczkami odnawiającymi zdrowie.
- Optymalizacja – zastosowano technikę Occlusion Culling, aby zwiększyć wydajność gry poprzez dynamiczne ukrywanie niewidocznych obiektów.

5. Podsumowanie i wnioski

Podczas tworzenia gry zespół napotkał kilka wyzwań, które wymagały kreatywnych rozwiązań. Oto niektóre z nich:

- Nieprawidłowa synchronizacja animacji między graczami w trybie sieciowym: Animacje były

odtworzane dla obu graczy jednocześnie, mimo że tylko jeden z nich się poruszał. Problem rozwiązano przez przypisanie osobnych komponentów animatorów dla każdej postaci i odpowiednie skonfigurowanie właściciela pliku, co umożliwiło poprawne odbieranie i wysyłanie informacji o animacjach między serwerem a klientami.

- Problemy z eksportem modeli broni z Blendera do Unity: Po eksporcie modeli broni z Blendera do Unity, szczegóły tekstur były tracone, a obiekty wyświetlały się jedynie w jednolitych kolorach. Rozwiązaniem było stworzenie map normalnych dla każdego obiektu, co zapewniło prawidłowe odwzorowanie szczegółów tekstur. Wcześniej tekstury zawierały tylko mapy diffuse, które przechowują jedynie informacje o kolorach obiektów.
- Spadki wydajności: Na niektórych częściach mapy gra doświadczała spadków liczby klatek na sekundę. Aby poprawić wydajność, z mapy usunięto modele o nadmiernej liczbie poligonów, a inne zostały zoptymalizowane, co zmniejszyło ich wpływ na wydajność.
- Problem z teleportacją: Podczas korzystania z umiejętności teleportacji, gracz czasami pojawiał się w innym miejscu niż to, które wskazał. Problem rozwiązano poprzez przeniesienie funkcji teleportacji z metody Update(), wywoływanej w każdej klatce, do metody FixedUpdate(), która jest wywoływana w równych odstępach czasu. Dzięki temu teleportacja przebiegała płynnie.
- Opóźnienia synchronizacji obiektów w grze: Synchronizacja obiektów w sieci lokalnej działała zbyt wolno, co powodowało problemy takie jak podwójne obrażenia od jednej kuli lub brak obrażeń. Aby rozwiązać ten problem, należało zmienić sposób wymiany danych między graczami, przechodząc na dedykowany serwer gry, zoptymalizowany pod kątem szybszej synchronizacji obiektów.

Kierunki rozwoju:

Aby rozwinąć obecny prototyp, zaplanowano następujące ulepszenia:

1. Rozszerzona zawartość:
 - Dodanie nowych map, broni oraz umiejętności, aby uroznać różnorodność gry.

- Ulepszony multiplayer:
- Wprowadzenie obsługi większej liczby graczy oraz trybów drużynowych.

2. Integracja AI:

- Dodanie przeciwników sterowanych przez sztuczną inteligencję, co wzbogaci doświadczenie w trybie dla pojedynczego gracza i da możliwość treningu dla nowych graczy.

3. Optymalizacja wydajności:

- Dalsze usprawnienia w zakresie wydajności, umożliwiające grę na szerszym zakresie konfiguracji sprzętowych.

Literatura

- [1] Robert Nystrom. Game Programming Patterns. Wydawnictwo Genever Benning, listopad 2014.
- [2] Joe Hocking Unity in Action: Multiplatform Game Development in C# with Unity 5, Helion, sierpień 2017
- [3] Unreal Engine 4 Game Development in 24 Hours, Sams Publishing, czerwiec 2016
- [4] Celia Hodent . The Gamer's Brain: How Neuroscience and UX Can Impact Video Game Design, CRC Press, sierpień 2017
- [5] Raph Koster A Theory of Fun for Game Design, O'Reilly Media, grudzień 2013
- [6] Rachel Kowert, Thorsten Quandt. Video Games and Well-being: Press Start, 2019
- [7] Minecraft Revenue and Usage Statistics (2024) - Business of Apps
- [8] Lethal Company - Steam Charts
- [9] Stardew Valley - Steam Charts
- [10] Harrison Ferrone Learning C# by Developing Games with Unity 2021", Packt Publishing, październik 2021
- [11] Matt Smith. Unity Game Development Cookbook, Packt Publishing, wrzesień 2021
- [12] Ferrone Harrison. Stwórz grę w Unity, a nauczysz się programowania w C#! Pisanie kodu, które sprawia radość, Packt Publishing, listopad 2022
- [13] Hardman Casey. Programowanie gier przy użyciu Unity i C#, Promise, grudzień 2020
- [14] Unity - Manual: Netcode for GameObjects
- [15] Sosna Łukasz Visual Studio 2022, C# i .NET. Programowanie kontrolki, Helion, styczeń 2023
- [16] Matulewski Jacek. Visual Studio 2022. Wprowadzenie do .NET MAUI, Helion, marzec, 2023
- [17] Bociak Bogdan. Blender. Podstawy modelowania, Helion, czerwiec 2007

- [18] Cieśla Krzysztof. Inkscape. Zaawansowane funkcje programu, Helion, październik 2013
- [19] <https://audacity.pl/>
- [20] What is PlayFab? - PlayFab | Microsoft Learn