

GRA SIECIOWA FPS Z UŻYCIEM SILNIKA UNITY

Jakub Witkowski, Grzegorz Zych

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

Niniejsza praca koncentruje się na opracowaniu gry sieciowej typu FPS (First-Person Shooter) z wykorzystaniem środowiska Unity. W pracy przedstawiono szczegółowy opis gry, projekt aplikacji, model bazy danych, diagramy interakcji, a także opis napotkanych problemów podczas procesu tworzenia gry. Ponadto uwzględniono informacje dotyczące wykorzystanych narzędzi oraz zastosowanych rozwiązań technologicznych.

Unity, FPS, gra wieloosobowa, moce

NETWORK FPS GAME USING THE UNITY ENGINE

Abstract: This work focuses on the development of a network-based FPS (First-Person Shooter) game using the Unity environment. The paper provides a detailed description of the game, the application design, the database model, interaction diagrams, as well as a discussion of the challenges encountered during the game development process. Additionally, it includes information about the tools used and the technological solutions applied.

Keywords: Unity, FPS, multiplayer game, powers

1. Wstęp

Gry komputerowe stanowią istotny element współczesnej kultury oraz gospodarki, odgrywając kluczową rolę w sektorze rozrywki i technologii informatycznych. Są one zjawiskiem wielowymiarowym, które przyciąga uwagę zarówno naukowców, jak i przedsiębiorców ze względu na ich potencjał ekonomiczny, społeczny oraz edukacyjny. Popularność gier wynika z ich zdolności do zapewniania immersyjnych doświadczeń, dostępności technologicznej oraz różnorodności tematycznej, co czyni je medium o niezwykle szerokim zasięgu oddziaływania [1-3]. Współczesna branża gier komputerowych charakteryzuje się dynamicznym rozwojem i znaczącym wpływem na rynek globalny. Prognozy wskazują na jej dalszy wzrost, co wynika z postępu technologicznego oraz rosnącego zainteresowania grami jako formą spędzania czasu wolnego. Złożoność tej branży obejmuje nie tylko tworzenie i dystrybucję gier, ale także ich analizę pod kątem społecznych i psychologicznych aspektów oddziaływania [4-6]. Gry takie jak "Minecraft", "Lethal Company" czy "Stardew Valley" stanowią dowód na to, że nawet proste mechaniki mogą zdobyć szeroką popularność i zaangażowanie milionów graczy [7-9]. Celem niniejszej

pracy było stworzenie projektu pod nazwą "Heir of Death". Jest to próba połączenia klasycznych elementów strzelanek pierwszoosobowych (FPS) z nowoczesnymi rozwiązaniami w zakresie projektowania rozrywki. Główne założenia projektu obejmują stworzenie gry o niskich wymaganiach sprzętowych, co umożliwi dotarcie do szerokiego grona odbiorców, przy jednoczesnym zapewnieniu angażującego doświadczenia opartego na rywalizacji i strategii. Kluczową cechą gry jest jej dostępność, łącząca łatwość przyswojenia podstawowych zasad z głębią rozrywki wymagającą czasu i zaangażowania gracza. Zakres projektu obejmuje szczegółowe określenie wymagań funkcjonalnych i technicznych gry, analizę narzędzi wspierających proces tworzenia oraz implementację poszczególnych komponentów, takich jak grafika, mechanika gry, interfejs użytkownika oraz systemy dźwiękowe. Ponadto, w projekcie uwzględniono proces testowania, diagnozowania błędów oraz ich eliminacji, co pozwoliło na opracowanie stabilnego prototypu.

Ponadto "Heir of Death" to projekt łączący klasyczne inspiracje, takie jak "Quake" czy "Unreal Tournament", z nowoczesnymi elementami rozrywki, charakterystycznymi dla tytułów takich jak "Apex Legends".

2. Oprogramowanie oraz biblioteki wykorzystane do tworzenia gry

2.1 Oprogramowanie

Projekt gry oparty został o Unity 2022.3.10f1. Unity oferuje kompleksowy zestaw narzędzi do tworzenia gier, w tym renderowanie 3D, animację, skryptowanie, symulację fizyki i zarządzanie dźwiękiem [2, 10-14]. Jego wszechstronność i obszerna dokumentacja czynią go preferowanym wyborem zarówno dla początkujących, jak i doświadczonych twórców.

Kolejne użyte narzędzie to Visual Studio tj zintegrowane środowisko programistyczne (IDE) używane do pisania, debugowania i testowania skryptów w języku C# [15-16]. Rozbudowane funkcje Visual Studio ułatwiły implementację złożonej logiki gry i zapewniły jakość kodu. W Projekcie wybrano oprogramowanie do modelowania 3D open-source jakim jest Blender. Oprogramowanie to używane jest do tworzenia szczegółowych modeli broni, postaci i elementów środowiska. Wszechstronność i bogaty zestaw funkcji Blendera pozwoliły autorom zaprojektować atrakcyjne wizualnie i zoptymalizowane modele 3D [17].

Jako edytor grafiki wektorowej używany do projektowania komponentów interfejsu użytkownika, takich jak ikony i menu wykorzystany został program Inkscape, charakteryzujący się precyzją i skalowalnością narzędzia [18]. Oprogramowanie to było kluczowe w tworzeniu przejrzystego i intuicyjnego interfejsu. W celu dodanie efektów audio wybrano program Audacity. Jest to darmowy i otwarty edytor audio używany do nagrywania, edytowania i poprawiania efektów dźwiękowych w grze. Jego prostota i funkcjonalność umożliwiły zespołowi tworzenie wysokiej jakości zasobów audio, wzbogacających immersję w grze [19].

2.2. Biblioteki sieciowe

Netcode for GameObjects wykorzystano jako zintegrowaną z Unity bibliotekę zaprojektowaną do tworzenia gier wieloosobowych. To narzędzie było kluczowe dla implementacji synchronizacji stanu gry w czasie rzeczywistym między klientami w trybie multiplayer. Z kolei PlayFab [20] jest używana jako platforma backendowa dostarczana przez Microsoft, używana do uwierzytelniania użytkowników, przechowywania danych i zarządzania rankingami. PlayFab usprawnił integrację kluczowych funkcji sieciowych i zwiększył skalowalność komponentów online gry.

3. Koncepcja i projekt gry

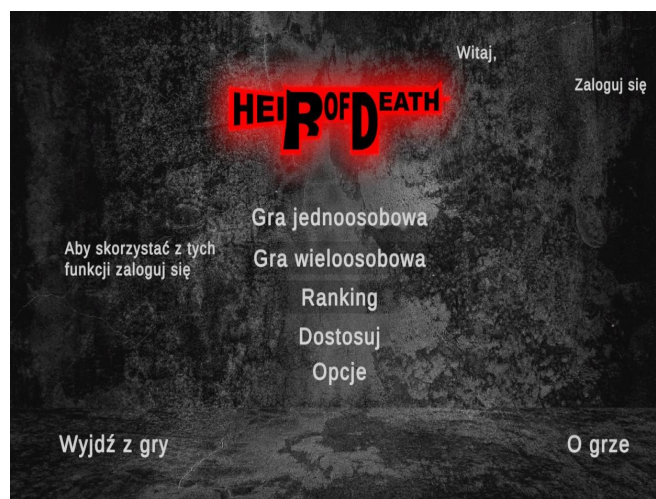
3.1 Aspekty gry

Faza koncepcyjna obejmowała zdefiniowanie kluczowych elementów gry, aby dostosować je zarówno do wizji zespołu, jak i preferencji docelowej grupy odbiorców. Proces projektowania koncentrował się na następujących aspektach:

- Tryby gry:

W tym celu zaprojektowano tryb treningowy dla jednego gracza. Został on zaprojektowany w celu zapoznania graczy z mechaniką gry, mapami i sterowaniem bez potrzeby połączenia z Internetem. W kolejnym etapie zaprojektowano tryb wieloosobowy: Konkurencyjny format 1v1, koncentrujący się na strategii i umiejętnościach.

Projekt grafiki gry pt „Heir of Death” przedstawiono na rysunku 1.



Rys. 1. Interfejs gry - Menu główne

- Mapy:
 - "Las": Gęste środowisko leśne z drzewami, rozrzuconymi budynkami i strategicznymi punktami ukrycia.
 - "Okopy": Poligon bitewny w klimacie wojennym z połączonymi okopami, sprzyjający taktycznej rozgrywce.

Na rysunku 2 przedstawiono poziom „Las” a na rysunku 3 poziom „Okopy”.



Rys. 2. Poziom „Las”



Rys. 3. Poziom „Okopy”

- Mechanika rozgrywki:
 - Broń: Arsenal obejmuje karabiny snajperskie, karabiny maszynowe i pistolety, zróżnicowane pod względem obrażeń, szybkostrzelności i prędkości przeładowania.
 - Umiejętności: Gracze mogą wybierać spośród czterech umiejętności (widzenie przez ściany, teleportacja, niewidzialność, ciche poruszanie się), aby zyskać strategiczną przewagę podczas meczu.

3.2 Proces tworzenia gry

Proces tworzenia i testowania gry rozłożony został na następujące etapy”

1. Planowanie i role zespołowe:

- Projekt został podzielony na mniejsze, zarządzalne zadania przypisane członkom zespołu na podstawie ich specjalizacji. Obejmowały one role takie jak modelowanie 3D, skryptowanie mechanik rozgrywki, projektowanie interfejsu użytkownika i implementacja funkcji sieciowych.

2. Implementacja logiki gry:

- Tworzenie logiki gry rozpoczęło się od opracowania diagramów UML, które mapowały relacje między kluczowymi elementami gry, w tym klasami dla Gracza, Broni, Umiejętności i Mapy.
- Opracowano skrypty obsługujące podstawowe mechaniki, takie jak ruch gracza, strzelanie, zarządzanie zdrowiem i aktywacja umiejętności. Na przykład mechanizm strzelania obejmował obliczanie trajektorii pocisków, obrażeń w zależności od stref trafienia oraz synchronizację tych działań między klientami w trybie multiplayer.

3. Implementacja sieci:

- Netcode for GameObjects zapewnił płynną synchronizację działań graczy, takich jak ruch, strzelanie i używanie umiejętności, w trybie wieloosobowym.
- PlayFab został zintegrowany w celu zarządzania uwierzytelnianiem użytkowników, danymi kont i funkcjonalnością rankingów. Ułatwiło to przechowywanie statystyk graczy i zapewniło bezpieczny dostęp do funkcji online.

4. Testowanie i debugowanie:

- Testowanie iteracyjne przeprowadzano przez cały cykl tworzenia. Zespół koncentrował się na identyfikowaniu błędów, równoważeniu mechanik rozgrywki i optymalizacji wydajności. Scenariusze testowe obejmowały sesje lokalne, jak i online, aby zapewnić stabilność i responsywność.
- Gra była testowana na różnych komputerach osobistych o zróżnicowanej konfiguracji sprzętowej, aby zapewnić spójność wydajności i kompatybilność. Takie podejście pomogło zidentyfikować i rozwiązać wąskie gardła wydajności, optymalizując grę dla różnych systemów.

4. Funkcjonalność gry "Heir of death"

Gra "Heir of Death" to sieciowa strzelanka FPS, oferująca dynamiczną rozgrywkę dla dwóch graczy. Gra charakteryzuje się prostym interfejsem, zróżnicowanymi mapami, umiejętnościami i bronią, a także systemem rankingowym, który dodaje do rozgrywki element rywalizacji.

- Tryb wieloosobowy i formuła rozgrywki:
 - Gra "Heir of Death" oferuje tryb wieloosobowy dla dwóch graczy (1 vs 1), co oznacza, że gracze rywalizują bezpośrednio ze sobą.
 - Rozgrywka toczy się w formule "Best of x rounds", gdzie "x" oznacza z góry określoną liczbę rund. Zwycięzcą meczu zostaje gracz, który jako pierwszy wygra określoną liczbę rund.
 - Każda runda trwa z góry ustalony czas, co dodaje do rozgrywki element presji czasu i wymusza na graczach podejmowanie szybkich decyzji.
 - Jeśli w regulaminowym czasie gry żaden z graczy nie wyeliminuje przeciwnika, runda jest restartowana, co zapewnia sprawiedliwość i daje obu graczom równe szanse na zwycięstwo.
- Mapy:
 - W grze dostępne są dwie mapy:
 - Las: Charakteryzuje się dużą ilością drzew i kilkoma budynkami.
 - Okopy: Zaprojektowana w klimacie wojennym, z okopami.
 - Obie mapy mają ograniczony obszar gry, co oznacza, że gracze nie mogą wyjść poza wyznaczone granice. Ograniczony obszar gry wpływa na dynamikę rozgrywki i zachęca do starć na bliskim dystansie.
- Umiejętności, Gracze mają do wyboru cztery umiejętności:
 - Widzenie przez ściany: Pozwala na podświetlenie modelu przeciwnika, co ułatwia jego zlokalizowanie, nawet gdy jest on ukryty za przeszkodami.
 - Teleportacja: Umożliwia natychmiastowe przeniesienie gracza w inne miejsce na mapie. Ta umiejętność może być wykorzystana zarówno do ataku, jak i do obrony.
 - Niewidzialność: Ukrywa model gracza, co utrudnia przeciwnikowi jego dostrzeżenie. Niewidzialność może być strategicznie wykorzystana do

przeprowadzania zaskakujących ataków lub unikania konfrontacji.

- Ciche poruszanie się: Eliminuje dźwięki kroków gracza, co pozwala mu na ciche podejście do przeciwnika. Ciche poruszanie się jest szczególnie przydatne w połączeniu z niewidzialnością.

Na rysunku 4 przedstawiono interfejs gry dotyczący wyboru umiejętności postaci.



Rys. 4. Interfejs gry - Wybór umiejętności postaci

- Bronie, w grze dostępnych jest pięć rodzajów broni:
 - Dwa karabiny wyborowe: Szybkostrzelne, automatyczne lub półautomatyczne. Karabiny wyborowe są skuteczne na długim dystansie, ale wymagają precyzji.
 - Dwa karabiny maszynowe: O dużej szybkostrzelności, automatyczne. Karabiny maszynowe są skuteczne na bliskim i średnim dystansie, ale mają dużą rozrzutność.
 - Jeden pistolet: O średniej szybkostrzelności, półautomatyczny. Pistolet jest bronią pomocniczą, skuteczną na bliskim dystansie.
- System rankingowy:
 - System rankingowy jest oparty o platformę PlayFab, która zapewnia usługi backendowe dla gier.
 - W rankingu wyświetlane są następujące informacje:
 - Nazwa gracza: Pozwala na identyfikację graczy w rankingu.
 - Liczba zwycięstw: Informuje o liczbie wygranych meczów przez danego gracza.

- Liczba porażek: Informuje o liczbie przegranych meczów przez danego gracza.
- System rankingowy dodaje do gry element rywalizacji i motywuje graczy do osiągania jak najlepszych wyników.
- Interfejs użytkownika (UI):
 - Interfejs użytkownika został zaprojektowany z myślą o intuicyjności i prostocie, co ułatwia graczom nawigację po grze.
 - Składa się z następujących ekranów:
 - Ekran powitalny: Wyświetlany po uruchomieniu gry, umożliwia logowanie, rejestrację i resetowanie hasła.
 - Menu główne: Daje dostęp do różnych trybów gry, ustawień, rankingu i innych funkcji.
 - Ekran wyboru poziomu i umiejętności: Pozwala graczom na wybór mapy i umiejętności przed rozpoczęciem gry.
 - Ekran wyboru postaci: Umożliwia graczom wybór postaci, którą będą grać.
 - Ekran rankingu: Wyświetla listę najlepszych graczy.
 - Ekran ustawień gry: Pozwala graczom na dostosowanie ustawień gry do swoich preferencji.
 - HUD (Head-up Display): Wyświetla informacje o stanie gry podczas rozgrywki, takie jak pasek zdrowia, czasomierz rundy, wybraną umiejętność, czas odnowienia umiejętności i dostępną amunicję.
 - Ekran wygranej/przegranej/pauzy: Wyświetlany po zakończeniu rundy lub w trakcie pauzy.

Dodatkowe informacje:

- Pozostałe cechy gry, takie jak:
 - System zadawania obrażeń: Określa, ile obrażeń zadają poszczególne bronie i umiejętności.
 - Interaktywne elementy map: Na mapach znajdują się obiekty, z którymi gracze mogą wchodzić w interakcję, np. apteczki uzupełniające zdrowie.
 - Optymalizacja: Zastosowano technikę OcclusionCulling, aby zwiększyć wydajność gry.
 - Źródła nie zawierają szczegółowych informacji na temat wszystkich funkcji i mechanik gry.

5. PODSUMOWANIE I WNIOSKI

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

- Nieprawidłowa synchronizacja animacji między graczami w sieci: Problem polegał na tym, że animacje były odtwarzane dla obu graczy jednocześnie, mimo że tylko jeden z nich się poruszał. Rozwiązanie polegało na przypisaniu osobnego komponentu animatora dla każdej postaci w sieci i odpowiednim skonfigurowaniu właściciela pliku, aby serwer mógł poprawnie odbierać i wysyłać informacje o animacjach do klientów.
- Problemy z eksportem modeli broni z Blendera do Unity: Po eksporcie modeli broni z Blendera do Unity, modele traciły szczegóły tekstur i wyświetlały tylko kolory. Rozwiązanie problemu polegało na stworzeniu map normal dla każdego obiektu, aby zapewnić prawidłowe odwzorowanie szczegółów tekstur. Wcześniej tekstury zawierały tylko mapę diffuse, która przechowuje informacje tylko o kolorach obiektu.
- Spadki wydajności: W niektórych miejscach na mapie gra doświadczała spadków liczby klatek na sekundę. Aby rozwiązać ten problem, z mapy usunięto modele o zbyt dużej liczbie poligonów, a niektóre z nich zostały zoptymalizowane, aby zmniejszyć ich wpływ na wydajność.
- Problem z teleportacją: Podczas korzystania z umiejętności teleportacji, gracz niekiedy pojawiał się w innym miejscu niż to, które wskazał. Problem został rozwiązany poprzez zmianę sposobu wywoływania funkcji teleportacji. Zamiast wywoływać ją w metodzie Update(), która jest uruchamiana w każdej klatce, funkcja została przeniesiona do metody FixedUpdate(), która jest wywoływana w równych odstępach czasu. Dzięki temu przenoszenie wszystkich GameObjectów podczas teleportacji przebiegało płynniej.
- Opóźnienia synchronizacji obiektów w grze: Synchronizacja obiektów w sieci lokalnej działała zbyt wolno. Powodowało to problemy, takie jak podwójne obrażenia od jednej kuli lub brak obrażeń. Problem wynikał z niewystarczającej szybkości przesyłania informacji o nowo utworzonych obiektach w sieci. Aby rozwiązać ten problem, należałoby zmienić sposób wymiany danych między graczami na dedykowany serwer

gry, który byłby zoptymalizowany pod kątem synchronizacji obiektów w grze.

Kierunki rozwoju

Aby rozwinąć obecny prototyp, proponowane są następujące ulepszenia:

1. Rozszerzona zawartość:
 - o Wprowadzenie dodatkowych map, broni i umiejętności, aby urozmaicić rozgrywkę.
2. Ulepszony multiplayer:
 - o Obsługa większych meczów z większą liczbą graczy i wprowadzenie trybów drużynowych.
3. Integracja AI:
 - o Dodanie przeciwników AI w celu wzbogacenia doświadczenia dla jednego gracza i zapewnienia możliwości treningu dla nowych graczy.
4. Optymalizacja wydajności:
 - o Dalsza optymalizacja gry dla szerszego zakresu konfiguracji sprzętowych.

Tworzenie gry "Heir of Death" stanowi znaczące osiągnięcie dla autorów, demonstrując ich zdolność do konceptualizacji, projektowania i wdrażania funkcjonalnego prototypu gry. Łącząc klasyczne elementy FPS z nowoczesnymi innowacjami, gra przemawia do szerokiego grona odbiorców, jednocześnie pozostając dostępna. Projekt nie tylko spełnił swoje cele edukacyjne, ale także stworzył solidne podstawy dla przyszłych postępów w tworzeniu gier.

Główne założenia projektu obejmowały dynamiczną rozgrywkę wieloosobową, możliwość personalizacji postaci oraz strategię opartą na unikalnych umiejętnościach graczy. Dzięki zastosowaniu współczesnych technologii oraz skupieniu na minimalizacji wymagań sprzętowych, gra stanowi przykład nowoczesnego podejścia do projektowania gier komputerowych, które łączy funkcjonalność z dostępnością. Proces realizacji projektu dostarczył cennych doświadczeń i umożliwił głębsze zrozumienie złożoności oraz etapów tworzenia zaawansowanych produktów informatycznych.

Literatura

1. Robert Nystrom. Game Programming Patterns. Wydawnictwo GeneverBenning, 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. CeliaHodent . The Gamer's Brain: How Neuroscience and UX CanImpact 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, ThorstenQuandt. Video Games and Well-being: Press Start, 2019
7. MinecraftRevenue and UsageStatistics (2024) - Business of Apps
8. Lethal Company - SteamCharts
9. Stardew Valley - SteamCharts
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. HardmanCasey. 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 kontrolek, Helion, styczeń 2023
16. Matulewski Jacek. Visual Studio 2022. Wprowadzenie do .NET MAUI, Helion, marzec, 2023
17. Bocię Bogdan. Blender. Podstawy modelowania, Helion, czerwiec 2007
18. Cieśla Krzysztof. Inkscape. Zaawansowane funkcje programu, Helion, październik 2013
19. <https://audacity.pl/>
20. WhatisPlayFab? - PlayFab | Microsoft Learn