Jakub Maliszewski,

Mateusz Lewczak,

Krzysztof Blankiewicz,

Konrad Polański

Raport – Projekt zespołowy

**Gra typu FPS(First-Person Shooter) w stylu gry Doom93**

# Wstęp

# Założenia projektowe

Nasz projekt polega na stworzeniu pierwszoosobowej gry typu FPS (First Person Shooter) w perspektywie 2,5D inspirowaną stylem klasycznej gry Doom z 1993 roku. Doom jest jednym z legendarnych tytułów w historii gier wideo, często uznawanym za pioniera gatunku FPS, który zdefiniował wiele podstawowych mechanik, które stały się podstawą tego typu gier do dzisiaj. W grze zastosowana została perspektywa gracza 2,5D, która łączy dwuwymiarowe tekstury z trójwymiarową przestrzenią, tworząc iluzję głębi i realizmu przy zachowaniu prostoty graficznej – bez skomplikowanych modeli 3D.

Celem projektu jest nie tylko odtworzenie kultowej mechaniki rozgrywki, w tym najważniejsze aspekty rozgrywki, takie jak ruch gracza po mapie, walka z przeciwnikami oraz zbieranie przedmiotów i interakcje tych mechanik ze sobą. Projekt dał nam również możliwość zapoznania się z technicznymi aspektami tworzenia silników gier z perspektywą 2,5D. Projekt pozwoli nam także zgłębić zagadnienia takie jak optymalizacja kodu pod kątem wydajności, efektywne zarządzanie pamięcią, renderowanie grafiki w czasie rzeczywistym oraz implementacja podstawowej sztucznej inteligencji przeciwników. Gra będzie naśladować klimat i działanie oryginalnej gry – szybka akcja połączona z prostym sterowaniem, walka z przeciwnikami. Kluczowym elementem będzie zachowanie estetyki retro, z uwzględnieniem oryginalnej grafiki, jednak z pewnymi uproszczeniami na potrzeby projektu zespołowego.

Oryginalnego Dooma93 napisano w języku C, a w pewnych krytycznych aspektach, np. do obsługi grafiki oraz dźwięku wykorzystano assemblera, co zapewniło dużą wydajność na komputerach z tego okresu []. My natomiast do skonstruowania naszego silnika gry sięgnęliśmy po język C++ w standardzie C++11, również ze względów wydajnościowych. Do utworzenia okna gry oraz rysowania elementów w grze użyliśmy popularnej i prostej w użytkowaniu biblioteki SFML 2.6.1. Do efektywnego zarządzania kodem źródłowym i umożliwienie wygodnej pracy zespołowej w projekcie wykorzystaliśmy system kontroli wersji Git, który pozwala na śledzenie zmian w kodzie i łatwe integrowanie pracy wszystkich członków zespołu.

W projekcie zastosowaliśmy wzorzec projektowy ECS (Entity-Component-System) do zarządzania logiką silnika gry. Wzorzec pozwoli na elastyczne tworzenie i zarządzanie obiektami gry poprzez podział na trzy główne typy elementów: Encje (Entities), Komponenty (Components) oraz Systemy (Systems). Takie podejście projektowe ułatwia dodawanie nowych elementów gry i modyfikację już istniejących co poprawia skalowalność i elastyczność projektu [].

Proces rozwoju gry będzie obejmował regularnie przeprowadzane testy kolejnych implementowanych modułów, aby zapewnić stabilność i jakość tworzonej gry. Końcowym wynikiem projektu będzie gotowa, grywalna wersja demo, prezentująca kluczowe mechaniki gry, dostępna do gry na komputerach PC z przynajmniej jednym poziomem.

# Harmonogram i spis zadań

Harmonogram projektu został rozpisany na dwa półsemestry, aby zapewnić wystarczająco dużo czasu na projektowanie, implementację, testowanie oraz dokończenie projektu. Daje to także margines na rozwiązanie problemów, które mogą pojawić się w trakcie realizacji projektu. Podczas przydziału zadań podzieliliśmy grę na odpowiednie moduły i systemy, tak aby w miarę możliwości każdy był odpowiedzialny za swoją autonomiczną część projektu, co na końcu pozwoli scalić wszystko w całość.

1. Przygotowanie środowiska programistycznego (Wiosna)
   1. Wybór narzędzi programistycznych (marzec):

* Decyzja o wyborze narzędzi programistycznych, takich jak kompilator, biblioteki, system kontroli wersji oraz środowiska IDE (cały zespół),
* Utworzenie repozytorium i konfiguracja systemu kontroli wersji dla całego zespołu (cały zespół),
  1. Instalacja i konfiguracja narzędzi (marzec – kwiecień)
* Instalacja wybranych narzędzi na komputerach roboczych każdego z członków zespołu (cały zespół),
* Konfiguracja środowiska programistycznego, ustawienia projektu w IDE (cały zespół),
* Stworzenie początkowej struktury katalogów w repozytorium (Mateusz Lewczak),
* Przetestowanie instalacji poprzez uruchomienie przykładowego projektu SFML (cały zespół)

1. Implementacja podstawowej struktury silnika (Wiosna)
   1. Struktura klasy GameEngine (kwiecień) (Jakub Maliszewski),
   2. Implementacja wzorca projektowego ECS (Entity – Component – System) (kwiecień – maj) (Mateusz Lewczak),
   3. Implementacja klasy WADLoader wczytującej dane z pliku o rozszerzeniu .WAD z poziomem (kwiecień – maj) (Mateusz Lewczak),
2. Implementacja głównych systemów logiki gry oraz potrzebnych komponentów (Lato) (do połowy sierpnia)
   1. System kontroli gracza i komponent stanu gracza (Jakub Maliszewski),
   2. System ruchu gracza i komponent ruchu (Jakub Maliszewski),
   3. System broni i komponent broni (Krzysztof Blankiewicz),
   4. System obrażeń i komponenty życia oraz obrażeń (Krzysztof Blankiewicz),
   5. System przeciwników i komponent przeciwnika (Jakub Maliszewski),
3. Implementacja systemów odpowiedzialnych za renderowanie grafiki (Lato)
   1. System renderowania minimapy i komponentu spritu minimapy (Mateusz Lewczak),
   2. System renderowania grafiki i komponentu spritu w grze (Mateusz Lewczak),
   3. System renderowania interfejsu gracza (Jakub Maliszewski).
4. Testowanie, optymalizacja i dokumentacja (Lato)
   1. Testowanie i optymalizacja (sierpień)
      * Przeprowadzenie pełnych testów funkcjonalnych i wydajnościowych, aby upewnić się, że gra działa zgodnie z założeniami projektowymi (cały zespół),
      * Poprawa ewentualnych błędów i optymalizacja kodu w krytycznych obszarach silnika (cały zespół),
   2. Opracowanie dokumentacji i końcowego raportu (sierpień – wrzesień)

* Poprawienie komentarzy w kodzie opisujących strukturę projektu (cały zespół),
* Sporządzenie raportu końcowego, podsumowującego wyniki projektu, napotkane problemy i ich rozwiązania oraz wnioski z realizacji projektu (cały zespół).

# Podstawy teoretyczne

# Opis realizowanego projektu

System Kontroli gracza,

System ruchu gracza,

System Przeciwników,

- opis zrealizowanego projektu, dane techniczne, wykonane testy, omówienie  
uzyskanych wyników, ...

# Podsumowanie

-- dyskusja otrzymanych wyników,  
-- co się udało zrobić a co nie, proponowane dalsze prace