



**Wydział Zastosowań
Informatyki
i Matematyki**

Raport finalny

Specyfikacja projektu zaliczeniowego

Mateusz Ługowski

14 stycznia 2024

Spis treści

1	Wstęp	2
1.1	Ogólny opis projektu	2
1.2	Fabula oryginalnej gry	2
1.3	Fabula WZiMong Us	2
2	Planowanie pracy i komunikacja	3
2.1	Podział pracy występujący w projekcie	3
2.2	Kanały komunikacji	4
3	Wykorzystane programy i technologie	5
3.1	Silnik gry i język programowania	5
3.2	Program kontroli wersji	5
3.3	Łączenie się pomiędzy graczami (Multiplayer)	6
3.4	Grafika	6
4	Elementy Rozgrywki	7
4.1	Cel gry	7
4.2	Niezbędne elementy rozgrywki	7
4.3	Opis i podsumowanie elementów rozgrywki	8

1 Wstęp

1.1 Ogólny opis projektu

Celem naszego projektu jest stworzenie desktopowej gry multiplayer, na podstawie popularnego tytułu z 2018 roku studia InnerSloth - "Among Us". Swoją aplikację nazwaliśmy **"WZiMong Us"** i staramy się wiernie odwzorować klimat, rozgrywkę i grafikę jaka występuje w oryginalnej wersji gry. Gra ma charakter towarzyski w stylu "wszyscy przeciw jednemu", a jej uniwersum osadzone jest na wydziale Zastosowań Informatyki i Matematyki Szkoły Głównej Gospodarstwa Wiejskiego w Warszawie.

Aplikacja posłuży do promocji naszego wydziału, możliwość zagrania w nią będzie m.in. na dniach otwartych, ale też udostępniemy ją publicznie, żeby każdy zainteresowany mógł pobawić się na "WZiMie" wspólnie ze swoimi znajomymi.

1.2 Fabuła oryginalnej gry

Fabuła "Among Us" obraca się wokół załogi kosmicznego statku, która wykonuje różne zadania na różnych pokładach statku. Jednak jedna lub więcej postaci wśród załogi to tajni zdrajcy, znani jako "Impostorzy". Ich zadaniem jest eliminacja reszty załogi, sabotując zadania i oskarżając niewinnych. Gracze muszą współpracować, aby ukończyć zadania, jednocześnie starając się odkryć tożsamość oszustów poprzez dyskusje i głosowanie. Gdy na pokładzie zginie wystarczająco wielu członków załogi lub zostaną ukończone wszystkie zadania, gra kończy się.

1.3 Fabuła WZiMong Us

Studia na kierunku informatyka nie należą do najłatwiejszych, a na pewno jeżeli jest to kierunek na jednej z najlepszych i największych uczelni w kraju. Każdy ze studentów kombinuje, żeby zdać z jak najlepszym wynikiem. Biorąc pod uwagę nasze przeżycia związane ze studiami, postanowiliśmy zmodyfikować fabułę "Among Us" w taki sposób, aby rzetelnie ją odwzorować, jednocześnie osadzając ją w akademickim uniwersum. Oryginalni członkowie załogi statku kosmicznego, w naszym projekcie są studentami Wydziału Zastosowań Matematyki i Informatyki SGGW w Warszawie i za wszelką cenę próbują ułatwić sobie życie przed zapowiedzianą sesją, która zbliża się wielkimi krokami, a materiału do nauki wcale nie ubywa. Impostorzy zamienieni są na wykładowców, których celem jest obalenie każdego studenta. Mogą oni sabotować zadania studentów i oskarżać innych żeby ich się pozbyć. Studenci muszą kooperować, rozwiązywać zadania, robić ściągę i wyeliminować wykładowców, zanim ich obleją.



Rysunek 1: Ikona WZiMong US

2 Planowanie pracy i komunikacja

2.1 Podział pracy występujący w projekcie

Pracując przy tak dużym projekcie, niezbędny jest odpowiedni podział pracy. Nasza grupa jest dość duża, liczy 20 osób, więc postanowiliśmy podzielić się na poszczególne zespoły:

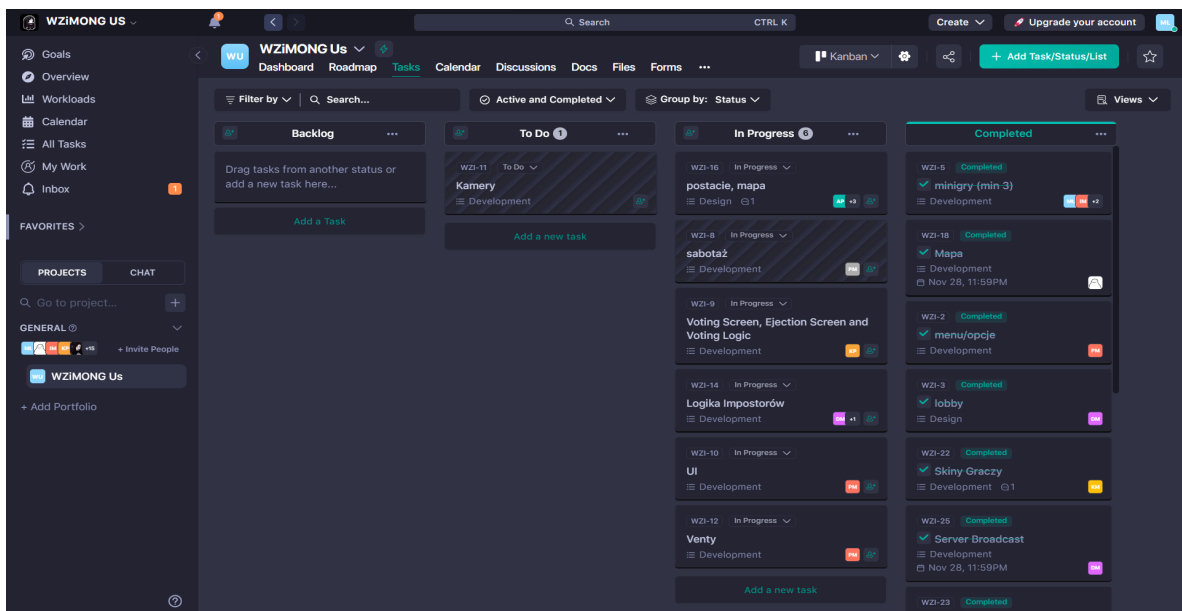
- Deweloperzy - osoby odpowiedzialne za pisanie kodu i dodawanie nowych funkcjonalności
- Testerzy - osoby odpowiedzialne za sprawdzanie poprawności działania nowych funkcjonalności i zgłaszanie błędów, razem z możliwymi rozwiązaniami
- Art Design - osoby odpowiedzialne za szatę graficzną gry i interfejsu
- Netcode - osoby odpowiedzialne za stworzenie działającego systemu multiplayer i dostosowywanie do niego nowych funkcjonalności

Każdy zespół posiada swojego kierownika, który rozdziela zadania i kontroluje pracę podległych mu osób, samemu jednocześnie rozwiązując konflikty w kodzie podczas aktualizowania głównej wersji. Zespoły liczą od 2 do 6 osób, w których jedna osoba zajmuje się maksymalnie jednym zadaniem. Naszym zdaniem jest to najlepszy podział pozwalający na zmaksymalizowanie i zoptymalizowanie pracy.

Przypisywanie zadań do wykonawców ułatwi nam, aplikacja Nifty: Project Management. Możliwe jest w niej wykonanie diagramu Gantta, który pozwala określić kolejność w jakiej powinny być wykonywane kolejne etapy projektu. Aplikacja jest prosta, przejrzysta, ułatwia przeglądanie zadań i ich organizację.

Imię	Nazwisko	Zespół
Mateusz	Ługowski	Project Manager
Igor	Osiakowski	Dev Manager
Szymon	Młonek	Test Manager
Dawid	Maliszewski	Netcode Manager
Daryna	Mala	Art Design Manager
Ilya	Pauliuk	Netcode
Michał	Łyszkowicz	Test
Tomasz	Niewiadomski	Test
Jakub	Mierzwa	Test
Kacper	Malik	Test
Amelia	Posiadała	Art Design
Błażej	Józwik	Art Design
Zuzanna	Majewska	Art Design
Aleksander	Maciejewski	Art Design
Kacper	Pełka	Dev
Patryk	Matuszewski	Dev
Piotr	Matoszka	Dev
Ivan	Matveichyk	Dev
Kamil	Marszałek	Dev
Sebastian	Matejak	Dev

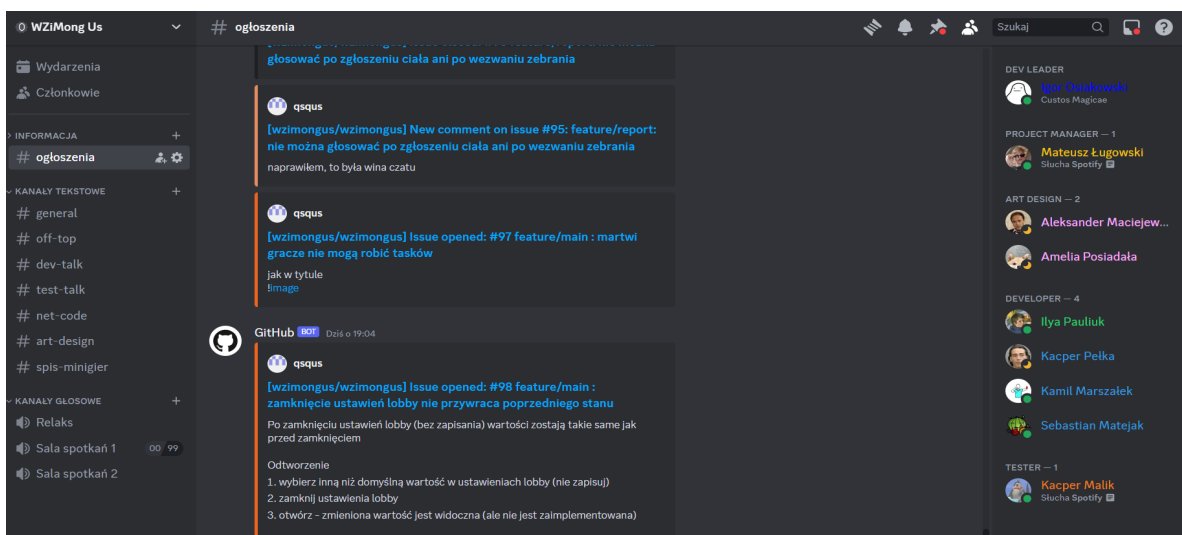
Tabela 1: Tabela przedstawiająca przypisanie osób do poszczególnych zespołów



Rysunek 2: Interfejs aplikacji Nifty: Project Management

2.2 Kanały komunikacji

Podczas pracy nad tak dużym projektem, przed którym nieliczni mieli jakiekolwiek doświadczenie z projektami w branży IT, komunikacja i wspólna pomoc jest niezwykle ważna. Naszym głównym kanałem komunikacji jest serwer Discord, na którym systematycznie co wtorek odbywają się spotkania mające na celu podsumowanie cotygodniowej pracy i rozdzielenia nowych zadań. Aplikacja Discord jest świetnie przystosowana do takiego typu projektów. Możliwe jest zaprojektowanie bota, który będzie informował nas o wszystkich zmianach, a dzięki wbudowanej funkcji Discorda, jaką jest Code Blocks, przesłany kod jest przejrzysty i przyjemny dla oka. Jest również możliwość stworzenia tematycznych kanałów dla każdego z zespołów, aby mogli wymieniać się informacjami i nowymi pomysłami.



Rysunek 3: Wykorzystywany przez nas serwer Discord

3 Wykorzystane programy i technologie

3.1 Silnik gry i język programowania

Zgodnie z poprzednimi założeniami, do stworzenia naszej gry posłużymy się silnikiem Godot Engine w wersji 4.1.x, używając języka GDScript. Wykorzystanie Godot Engine jest najlepszym możliwym wyborem do tworzenia gry 2D. Jest łatwy w obsłudze, przejrzysty oraz posiada duże wsparcie społeczności. GDScript ułatwia nam przeprowadzenie testów jednostkowych naszej aplikacji i z pewnością wpłynie pozytywnie na optymalizację gry.

Wykorzystane biblioteki języka GDScript

- Netfox - Biblioteka pozwalająca na dodanie do gry rollbacku, usprawniając grę online
- Lan Server Broadcast - Biblioteka pozwalająca wykonać broadcast serwera, umożliwia stworzenie wyszukiwarki serwerów.
- GUnit4 - Biblioteka pozwalająca na skuteczne przeprowadzenie testów jednostkowych

3.2 Program kontroli wersji

Jeżeli chodzi o program do kontroli wersji, wybór mógłby być tylko jeden - Git razem z Githubem. Git oferuje niezrównaną prędkość i niezawodność, co pozwala skoncentrować się na tworzeniu i udoskonalaniu kodu źródłowego, zamiast martwić się o bieżący stan projektu. Możliwość tworzenia i scalania gałęzi projektu ułatwia równoczesną pracę nad różnymi funkcjonalnościami, a jego otwarty charakter i darmowa dostępność umożliwiają współpracę. Git pozwala również na skuteczne śledzenie historii projektu, co znacząco ułatwia pracę kierownikom.

Konwencje, nazewnictwo i zasady obowiązujące w repozytorium na GitHubie

- Nazwy scen, skryptów, testów - snake_case
- Nazwy Node'ów i klas - PascalCase

Wszystkie pliki potrzebne do odpalenia wybranych scen znajdują się w folderze scenes, który dzieli się na konkretne foldery, odpowiadające poszczególnym rodzajom scen.

- game
- maps
- minigames
- player
- ui

W każdym z tych folderów, znajduje się podfolder, w którym jest konkretna funkcjonalność/minigra/część mapy. Sprawia to że kontrola projektu jest bardzo przyjemna i wszystko jest uporządkowane

Branche i Commity

Korzystamy z tzw. feature branchów. Podstawowe założenie feature branch polega na tym, że każda nowa funkcja lub zadanie jest rozwijane w osobnej gałęzi, a nie bezpośrednio na głównej gałęzi (main). Każdy commit jest nazwany w języku angielskim, gdzie tytuł informuje o temacie commita, a opis dokładnie dokumentuje co zostało zrobione.

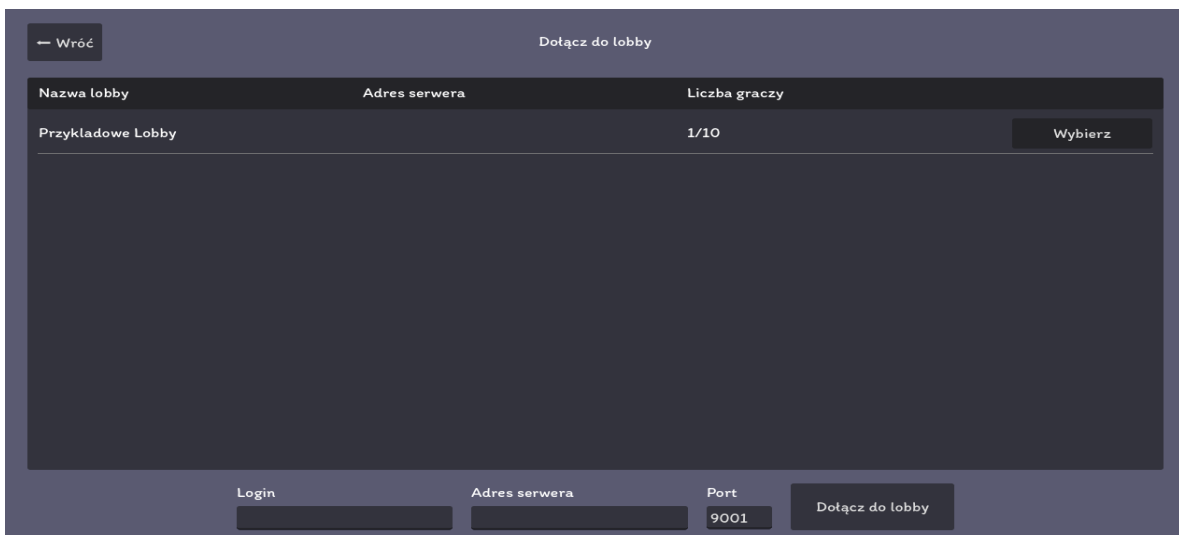
3.3 Łączenie się pomiędzy graczami (Multiplayer)

System sieciowy

Kwestia multiplayer została rozwiązana w najprostszy możliwy sposób. Gracze łączą się do jednego założyciela poczekalni (Hosta). Korzystamy z połączenia P2P (peer-to-peer). Aby dołączyć do gry, należy wybrać odpowiedni serwer za pomocą zaimplementowanej wyszukiwarki i podać login, który wyświetlany będzie nad postacią gracza

System logowania

W naszej grze nie występuje system progresji, osiągnięć, ani zapisu, więc nie ma konieczności tworzenia prywatnego konta. Gracz będzie zobowiązany tylko do podania nicku podczas dołączania do poczekalni.



Rysunek 4: Wyszukiwarka serwerów

3.4 Grafika

Grafika w naszej grze została stworzona od zera i wszystkie elementy zostały wykonane przez nasz zespół Art Design. Głównym programem wykorzystywanym przez naszych grafików jest Ink Scape. Pozwala on tworzyć grafikę wektorową, używać szablonów, jednocześnie będąc bardzo przyjaznym dla użytkownika. Założenia graficzne postawione przed nami to m.in. wykonanie mapy, ikony gry, postaci, przycisków UI, animacji poruszania i oblewania.



Rysunek 5: logo Ink Scape

4 Elementy Rozgrywki

4.1 Cel gry

Celem gry jest wyeliminowanie jednej ze stron, albo poprzez oblanie wszystkich studentów, albo przez wyeliminowanie wykładowców. Każda ze stron ma konkretne warunki wygrania, które muszą być spełnione, aby gra się zakończyła.

Warunki wygrania rozgrywki

Dla studentów:

- Studenci wygrywają, gdy wykonają wszystkie swoje zadania na wydziale.
- Studenci wygrywają, gdy zidentyfikują i wyeliminują wszystkich Wykładowców, jest osiągnięte poprzez głosowanie i eliminację podejrzanych.

Dla wykładowców:

- Wykładowcy wygrywają, gdy ich liczba jest równa lub przewyższa liczbę pozostałych w grze Studentów.

4.2 Niezbędne elementy rozgrywki

Nasza gra wymaga wprowadzenia niezbędnych elementów, aby była możliwość przeprowadzenia rozgrywki. Jako zespół ustaliliśmy, że mimo obowiązkowych elementów wprowadzimy kilka dodatkowych elementów, aby rozgrywka była przyjemniejsza.

Lista niezbędnych elementów rozgrywki

- Mapa, na wzór budynku nr 34
- System głosowania
- Różniący się bohaterowie
- Zadania do wykonania przez studentów
- Sabotaż wykładowcy
- Czat pozwalający na komunikację
- System eliminacji
- System "ventów"
- Ustawienia rozgrywki pozwalające określić ilość graczy i wykładowców

Opcjonalne elementy, które sprawiają, że rozgrywka staje się przyjemniejsza

- Pasek postępu zadań
- Rozbudowane ustawienia
- System kamer
- Poczekałnia, w której można wybrać wygląd postaci
- Większa liczba zadań

Lista zadań do wykonania przez studentów

- Skopiowanie pliku z rozwiązaniami do egzaminu z programowania
- Otwarcie szafki z egzaminami, poprzez odtworzenie wcześniej wyświetlonej sekwencji przycisków
- Stworzenie ściągi z wzorami matematycznymi
- Zrobienie zdjęcia egzaminu
- Dopasowanie odpowiednich sal do wykładowców
- Wpisanie poprawnego hasła
- Włączenie światła poprzez odpowiednie ustawienie przełączników
- Dopasowanie wykresu funkcji $y = \sin(x)$ do podanego wzoru
- Odkręcenie tablicy

4.3 Opis i podsumowanie elementów rozgrywki

Element rozgrywki	Minimalne rozwiązanie	Przewidziane rozwiązanie
Mapa	Pomieszczenia wyglądem przypominające sale dydaktyczne w budynku 34	Pomieszczenia wyglądem przypominające sale dydaktyczne w budynku 34
System "Ventów"	Połączenie dwóch pokoi ze sobą za pomocą systemu ventów	Połączenie więcej niż dwóch pokoi za pomocą systemu ventów
System kamer	Brak	Dodanie systemu kamer umożliwiającego przeglądanie paru miejsc na raz
Bohaterowie	10 różnokolorowych bohaterów w jednolitych kolorach	12 spersonalizowanych bohaterów nawiązujących do typów studentów
Zadania	3 zadania po jednym z każdego typu: krótkie, średnie, długie	9 zadań po 3 z każdego typu
Pasek postępu zadań	Brak	Dodanie pasku pokazującego ile zadań zostało do końca rozgrywki
Sabotaż	Jeden sabotaż ułatwiający rozgrywkę impostorowi	Dodanie dwóch sabotaży
Czat	Wprowadzenie prostego i przejrzystego czatu tekstowego	Wprowadzenie prostego i przejrzystego czatu tekstowego
Głosowanie i eliminacja przegłosowanego	Wprowadzenie ekranu głosowania na wzór tego z Among Us po przegłosowaniu wyświetlenie informacji o odpadnięciu z rozgrywki	Dodanie animacji przy eliminacji przegłosowanego
Poczekalnia i przydzielanie modeli postaci	Brak poczekalni, modele postaci będą przydzielane na podstawie kolejności dołączania do gry	Stworzenie graficznej poczekalni i umożliwienie graczom wyboru wyglądu postaci
System eliminacji	Wyświetlenie informacji o eliminacji gracza	Wykonanie animacji, na której widoczna jest eliminacja gracza
Ustawienia rozgrywki	Wymagane: -ilość graczy -ilość impostorów(wykładowców)	Opcjonalne: -Czas na głosowanie i dyskusje -Zakres pola widzenia -Ustawienia prędkości graczy -Czas oczekiwania między sabotażami -Ilość zadań na mapie