

# Sprawozdanie z przedmiotu Zarządzanie Projektem Informatycznym

## Zakres projektu

Celem Projektu jest stworzenie gry komputerowej pt. "Mafia". Gra przeznaczona ma być do działania w przeglądarce i umożliwiać rozgrywkę wieloosobową online.

Gra w stylu mafia to popularna gra towarzyska, która wymaga od uczestników sprytu, strategicznego myślenia i umiejętności dedukcji. Gracze są podzieleni na dwie frakcje: "miasto" i "mafia". Celem miasta jest odkrycie tożsamości członków mafii i ich wyeliminowanie, podczas gdy mafia stara się eliminować członków miasta, zachowując swoją tożsamość nieujawnioną.

Gra toczy się na etapach, które obejmują dzień i noc. W ciągu dnia gracze dyskutują i głosują, aby wyeliminować podejrzane osoby z gry. Nocą mafia dokonuje zabójstw, a także inne role mogą wykonywać swoje specjalne zdolności.

W grze mogą występować różne role, które dodają głębi i złożoności rozgrywce, takie jak detektyw, lekarz, czy szeryf. Te role posiadają unikalne umiejętności, które mogą pomóc w identyfikacji i eliminacji mafii lub chronić niewinne osoby.

Gra w stylu mafia wymaga od graczy umiejętności komunikacji, manipulacji oraz analizy zachowań innych graczy, co czyni ją ekscytującą i emocjonującą grą towarzyską.

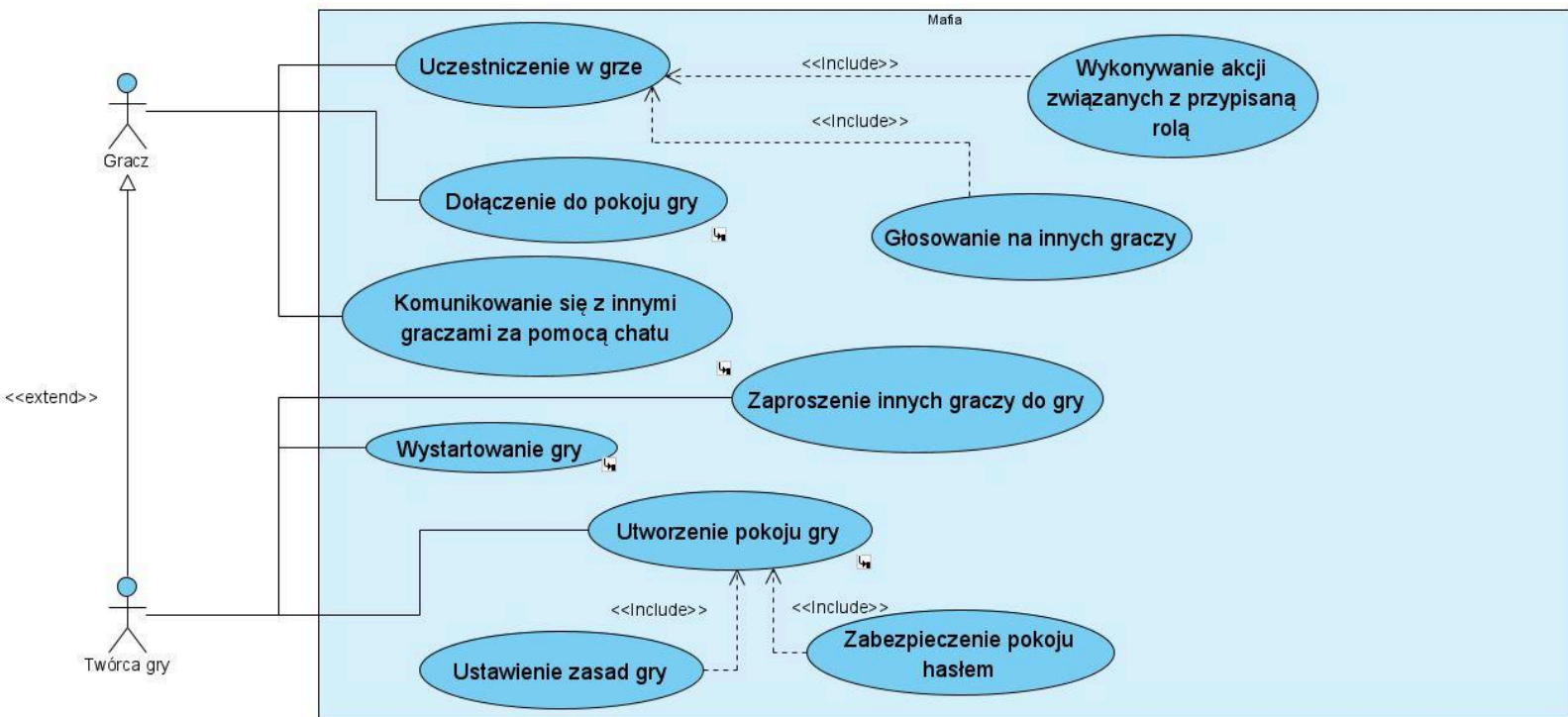
Gra powinna zostać zrealizowana jako gra przeglądarkowa. Są to gry dostępne w Internecie, do których uruchomienia nie jest wymagana instalacja oprogramowania na komputerze. Wystarczy jedynie posiadać przeglądarkę internetową i dostęp do sieci. Gry te cechują się prostotą instalacji, niskimi wymaganiami sprzętowymi oraz szeroką dostępnością. W ostatnich latach gry przeglądarkowe stają się coraz bardziej popularne, oferując graczom coraz lepszą grafikę, bardziej rozbudowaną rozgrywkę i wciągające historie.

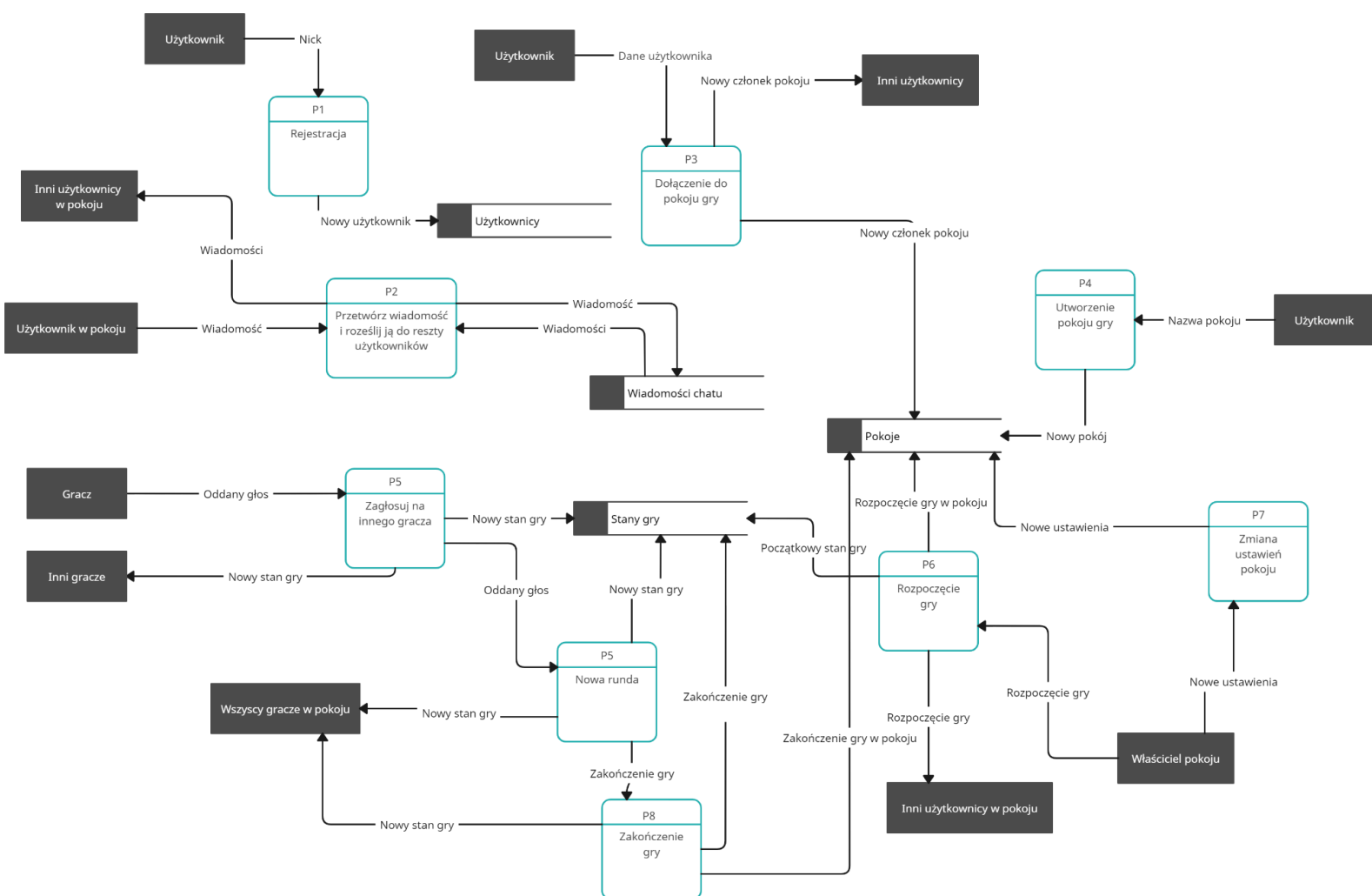
**Zarządzanie projektem stworzenia gry** w stylu "Mafia" wymaga zespołu odpowiednio wykwalifikowanych specjalistów, którzy będą mogli pokryć wszystkie aspekty produkcji gry. Proponowany zespół może obejmować następujące role i zadania:

1. **Project Manager:** Odpowiada za ogólne zarządzanie projektem, harmonogramowanie zadań, kontrolę postępu prac oraz koordynację pracy całego zespołu. Powinien posiadać doświadczenie w zarządzaniu projektami IT oraz umiejętność efektywnej komunikacji.

2. **Game Designer:** Odpowiada za opracowanie koncepcji rozgrywki, zasad gry, oraz struktury gry. Powinien posiadać kreatywność, umiejętność projektowania gier oraz zrozumienie mechanik gier towarzyskich.
3. **Lead Developer:** Odpowiada za techniczną stronę projektu, w tym programowanie gry, zarządzanie kodem źródłowym oraz zapewnienie prawidłowego działania gry w przeglądarce. Powinien posiadać doświadczenie w programowaniu gier oraz znajomość technologii webowych, takich jak HTML5, JavaScript, czy WebGL.
4. **UI/UX Designer:** Odpowiada za projektowanie intuicyjnego interfejsu użytkownika oraz doświadczenia użytkownika w grze. Powinien posiadać umiejętności w zakresie projektowania interfejsów oraz zrozumienie zasad UI/UX.
5. **Graphic Designer:** Odpowiada za projektowanie grafik, animacji oraz elementów wizualnych gry. Powinien posiadać umiejętności w zakresie projektowania grafiki komputerowej oraz znajomość narzędzi do grafiki komputerowej, takich jak Adobe Photoshop czy Illustrator.
6. **QA Tester:** Odpowiada za testowanie gry pod kątem błędów, niedociągnięć oraz zapewnienie wysokiej jakości gry. Powinien posiadać umiejętność identyfikacji błędów oraz znajomość technik testowania oprogramowania.
7. **Network Engineer:** Odpowiada za zapewnienie prawidłowego działania serwerów oraz infrastruktury sieciowej obsługującej grę wieloosobową online. Powinien posiadać doświadczenie w konfigurowaniu i zarządzaniu serwerami oraz sieciami komputerowymi.
8. **Scrum master:** to lider i przewodnik zespołu Scrum, który czuwa nad prawidłowym przebiegiem procesu scrumowego. Ułatwia współpracę między zespołem, klientem i właścicielem produktu, usuwa przeszkody i zapewnia optymalne warunki do tworzenia wartościowego oprogramowania.

# Diagram przypadków użycia oraz diagram przepływu danych





## Szczegółowy opis prac realizowanych w projekcie:

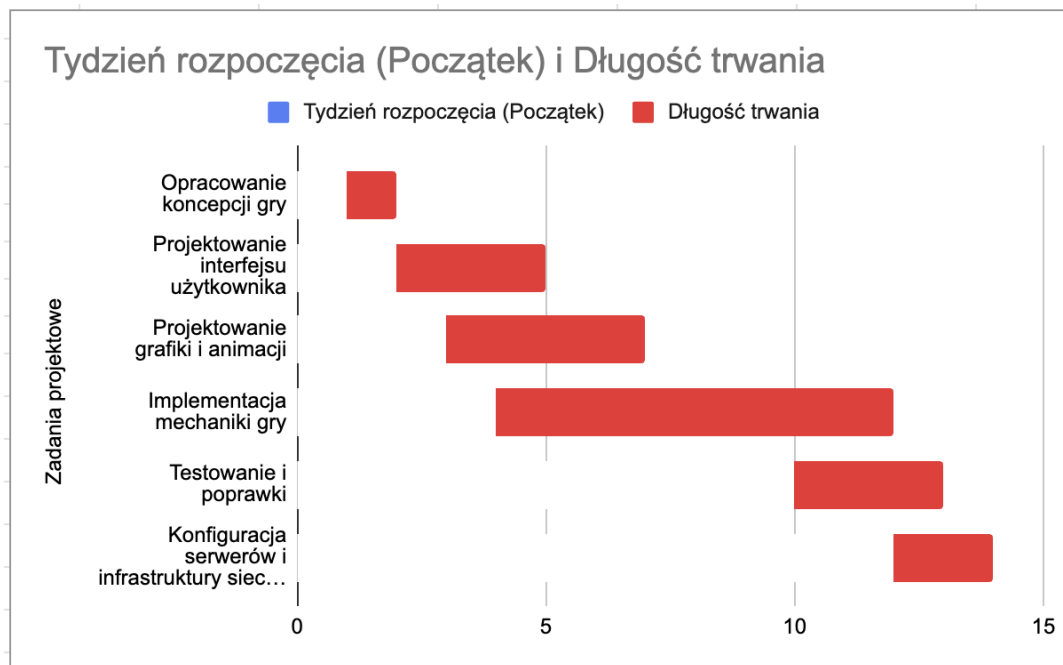
1. Opracowanie koncepcji gry:
  - a. Zadanie: Określenie głównych zasad rozgrywki, mechaniki gry oraz funkcji specjalnych.
  - b. Metoda kontroli: Regularne spotkania zespołu projektowego w celu omówienia postępów i ewentualnych zmian w koncepcji.
  - c. Metoda odbioru: Przedstawienie finalnej koncepcji gry zespołowi zarządzającemu oraz uzyskanie akceptacji.
2. Projektowanie interfejsu użytkownika (UI) i doświadczenia użytkownika (UX):
  - a. Zadanie: Stworzenie prototypów interfejsu oraz mapowania doświadczenia użytkownika w grze.
  - b. Metoda kontroli: Przegląd regularnych wersji prototypów przez zespół projektowy i zbieranie opinii.

- c. Metoda odbioru: Finalna wersja prototypu jest akceptowana przez zespół projektowy oraz testerów.
- 3. Projektowanie grafiki i animacji:
  - a. Zadanie: Tworzenie grafik, animacji oraz elementów wizualnych gry zgodnie z założeniami projektowymi.
  - b. Metoda kontroli: Regularne przeglądy postępów przez zespół projektowy oraz porównanie z założeniami projektowymi.
  - c. Metoda odbioru: Akceptacja finalnych wersji grafik i animacji przez zespół projektowy.
- 4. Implementacja mechaniki gry i programowanie:
  - a. Zadanie: Programowanie głównych mechanik gry, interakcji między graczami oraz innych funkcji specjalnych.
  - b. Metoda kontroli: Regularne testowanie funkcjonalności przez programistów oraz raportowanie o postępach.
  - c. Metoda odbioru: Przeprowadzenie testów integracyjnych i akceptacja przez zespół projektowy.
- 5. Testowanie i poprawki:
  - a. Zadanie: Testowanie gry pod kątem błędów, niedociągnięć oraz zapewnienie wysokiej jakości.
  - b. Metoda kontroli: Systematyczne raportowanie błędów i niedoskonałości oraz ich poprawa przez programistów.
  - c. Metoda odbioru: Pozytywny wynik testów akceptowany przez zespół projektowy i testerów.
- 6. Konfiguracja serwerów i infrastruktury sieciowej:
  - a. Zadanie: Konfiguracja serwerów obsługujących grę wieloosobową online oraz zapewnienie stabilnej infrastruktury sieciowej.
  - b. Metoda kontroli: Monitorowanie działania serwerów i infrastruktury sieciowej oraz szybka reakcja na wszelkie problemy.
  - c. Metoda odbioru: Potwierdzenie stabilnego działania przez network engineer'a oraz testy obciążeniowe.

#### **Czas realizacji zadań:**

- Opracowanie koncepcji gry - 1 tydzień
  - Uzasadnienie: Czas ten przeznaczony jest na przeprowadzenie burzy mózgów, dyskusje nad pomysłami, wykrystalizowanie głównych zasad gry, mechaniki i funkcji specjalnych.
  - Ryzyko: Brak jasnej wizji gry na początku projektu może wydłużyć czas tego etapu.
- Projektowanie interfejsu użytkownika i doświadczenia użytkownika - 3 tygodnie
  - Uzasadnienie: Ten etap obejmuje stworzenie prototypów interfejsu, testowanie ich z użytkownikami, zbieranie opinii i wprowadzanie iteracji.
  - Ryzyko: Zmiany w koncepcji gry na późniejszym etapie mogą wymagać przeprojektowania interfejsu, co wydłuży czas realizacji.

- Projektowanie grafiki i animacji - 4 tygodnie
  - Uzasadnienie: Czas ten zależy od złożoności grafiki, animacji i ilości elementów wizualnych w grze.
  - Ryzyko: Zmiany w koncepcji gry na późniejszym etapie mogą wymagać przerobienia grafik i animacji, co wydłuży czas realizacji.
- Implementacja mechaniki gry i programowanie - 8 tygodni
  - Uzasadnienie: Czas ten zależy od złożoności mechaniki gry, ilości interakcji i funkcji specjalnych.
  - Ryzyko: Napotkanie błędów lub problemów technicznych może wydłużyć czas realizacji.
- Testowanie i poprawki - 3 tygodnie
  - Uzasadnienie: Czas ten przeznaczony jest na gruntowne testowanie gry, identyfikację i naprawę błędów, dopracowanie grywalności i wyważenie rozgrywki.
  - Ryzyko: Znalezienie dużej ilości błędów lub problemów może wydłużyć czas testowania.
- Konfiguracja serwerów i infrastruktury sieciowej - 2 tygodnie
  - Uzasadnienie: Czas ten przeznaczony jest na konfigurację serwerów, wdrożenie oprogramowania sieciowego i testowanie infrastruktury.
  - Ryzyko: Niespodziewane problemy techniczne lub wzrost liczby graczy mogą wymagać dodatkowych konfiguracji.



# Zasoby wykorzystywane do stworzenia projektu:

## Komputery dla programistów, designerów i analityków:

- Z systemem Windows: 7000 PLN/szt. + Licencja Windows 1099 PLN
- Z systemem MacOS: 8999 PLN/szt.
- Z systemem Linux: 7000 PLN/szt.

Propozycje:

### System Windows:

#### 1. Dell XPS 15 (2023)

- Cena: 7499 PLN
- Procesor: Intel Core i7-12800H
- Pamięć RAM: 16 GB DDR5
- Dysk: 512 GB SSD PCIe NVMe
- Karta graficzna: NVIDIA GeForce RTX 3050 Ti
- Ekran: 15.6" FHD+ (1920x1200) IPS, 500 nits, Anti-Glare
- Bateria: Do 13 godzin

#### 2. Lenovo ThinkPad X1 Extreme Gen 5 (2023)

- Cena: 7999 PLN
- Procesor: Intel Core i9-12900H
- Pamięć RAM: 32 GB DDR5
- Dysk: 1 TB SSD PCIe NVMe
- Karta graficzna: NVIDIA GeForce RTX 3060
- Ekran: 16" UHD+ (3840x2400) IPS, OLED, 600 nits, Anti-Glare
- Bateria: Do 16 godzin

#### 3. Microsoft Surface Laptop Studio (2022)

- Cena: 7699 PLN
- Procesor: Intel Core i7-11370H
- Pamięć RAM: 16 GB DDR4x
- Dysk: 512 GB SSD PCIe NVMe
- Karta graficzna: NVIDIA GeForce RTX 3050
- Ekran: 14.4" FHD+ (3200x1440) IPS, 120 Hz, Touchscreen, Surface Pen compatible
- Bateria: Do 18 godzin

### System macOS:

#### 1. Apple MacBook Pro 14" (2021)

- Cena: 8999 PLN
- Procesor: M1 Pro 8-core CPU, 14-core GPU
- Pamięć RAM: 16 GB
- Dysk: 512 GB SSD
- Ekran: 14.2" Liquid Retina XDR, 3024x1964 pixels, 500 nits, True Tone
- Bateria: Do 17 godzin

#### 2. Apple MacBook Air M2 (2022)

- Cena: 6499 PLN
- Procesor: M2 8-core CPU, 10-core GPU
- Pamięć RAM: 8 GB
- Dysk: 256 GB SSD

- Ekran: 13.6" Retina display, 2560x1600 pixels, 500 nits, True Tone
- Bateria: Do 18 godzin

### 3. Apple iMac 24" (2021)

- Cena: 8499 PLN
- Procesor: M1 8-core CPU, 8-core GPU
- Pamięć RAM: 8 GB
- Dysk: 256 GB SSD
- Ekran: 24" 4.5K Retina display, 4480x2880 pixels, 500 nits, True Tone
- Bateria: Do 15 godzin

### System Linux:

#### 1. System76 Oryx Pro (2023)

- Cena: 7499 PLN
- Procesor: AMD Ryzen 9 7950X
- Pamięć RAM: 32 GB DDR5
- Dysk: 1 TB SSD PCIe NVMe
- Karta graficzna: NVIDIA GeForce RTX 3070
- Ekran: 17.3" FHD+ (1920x1080) IPS, 300 nits, Anti-Glare
- Bateria: Do 8 godzin

#### 2. Tuxedo Pulse 15 Gen 11 (2023)

- Cena: 7999 PLN
- Procesor: Intel Core i7-12800H
- Pamięć RAM: 16 GB DDR5
- Dysk: 512 GB SSD PCIe NVMe
- Karta graficzna: NVIDIA GeForce RTX 3060
- Ekran: 15.6" FHD+ (1920x1200) IPS, 300 nits, Anti-Glare
- Bateria: Do 8

### Oprogramowanie:

- **Adobe Creative Cloud:** 1 034,90 PLN/rok (Licencja dla 1 użytkownika)
- **IntelliJ IDEA IDE:** 3 361 PLN/rok (Licencja na 10 użytkowników)
- **Visual Studio Code** – darmowe
- **Docker business** 1200 PLN/rok (Licencja dla 1 użytkownika)

### Hosting Azure:

- Klaster 2 wirtualne maszyny do hostingu produkcyjnej wersji aplikacji (2 wirtualne procesory, 8 GB pamięci RAM, 50 GB pamięci magazynu tymczasowego) 8280 PLN/rok
- 5 maszyn wirtualnych do testowania (2 wirtualne procesory, 8 GB pamięci RAM, 50 GB pamięci magazynu tymczasowego) 33300 PLN/rok

### Licencja Udemy business: 2000 PLN/rok

Zakup licencji Udemy Business dla projektu gry może przynieść wiele korzyści, które znacząco usprawnią proces produkcji i zwiększą szanse na sukces gry. Jeżeli w zespole zabraknie kompetencji w poszczególniej dziedzinie dostęp do platformy szkoleniowej pozwala szybko zażegnać problemy.



**Dostęp do bogatej bazy kursów:** Platforma Udemy oferuje szeroki wachlarz kursów online z zakresu tworzenia gier, programowania, projektowania, marketingu i wielu innych dziedzin, które są niezbędne dla udanego projektu gry. Licencja Business zapewnia nieograniczony dostęp do wszystkich kursów, co pozwala członkom zespołu na poszerzanie swoich umiejętności i zdobywanie wiedzy w różnych obszarach.

**Nauka od ekspertów:** Kursy Udemy prowadzone są przez doświadczonych instruktorów z branży gier wideo, którzy dzielą się swoją wiedzą i praktycznym doświadczeniem. Członkowie zespołu mogą uczyć się od najlepszych w branży i poznawać najnowsze trendy i technologie.

**Szkolenie z różnych dziedzin:** Platforma oferuje kursy z różnych dziedzin związanych z tworzeniem gier, takich jak programowanie, grafika 3D, projektowanie poziomów, tworzenie muzyki i dźwięków, animacja, testowanie i wiele innych. To pozwala członkom zespołu na rozwijanie swoich umiejętności w różnych obszarach i tworzenie gry o wysokiej jakości.

**Dostęp do aktualnych treści:** Udemy regularnie dodaje nowe kursy i aktualizuje istniejące treści, co zapewnia członkom zespołu dostęp do najnowszych informacji i trendów w branży gier wideo. Pozwala im to na tworzenie gry zgodnej z aktualnymi oczekiwaniami graczy.

**Możliwość nauki w dowolnym czasie i miejscu:** Kursy Udemy dostępne są online, co oznacza, że członkowie zespołu mogą uczyć się w dowolnym czasie i miejscu, korzystając z komputera, tabletu lub smartfona. To zapewnia im elastyczność i pozwala na dopasowanie nauki do swoich indywidualnych potrzeb.

**Oszczędność czasu i pieniędzy:** Zakup licencji Udemy Business może być bardziej opłacalny niż wysyłanie członków zespołu na pojedyncze kursy lub szkolenia. Platforma oferuje szeroki zakres kursów w przystępnej cenie, co pozwala zaoszczędzić czas i pieniądze.

**Zwiększenie szans na sukces gry:** Dostęp do bogatej bazy wiedzy, możliwość nauki od ekspertów i rozwijanie umiejętności w różnych dziedzinach może znacząco zwiększyć szanse na sukces gry "Mafia". Inwestowanie w rozwój zespołu jest kluczowym czynnikiem w tworzeniu udanych gier wideo.

Oprócz wyżej wymienionych korzyści, licencja Udemy Business oferuje również inne funkcje, takie jak możliwość pobierania kursów offline, tworzenia własnych grup dyskusyjnych i korzystania z certyfikatów ukończenia kursów. Wszystkie te funkcje mogą dodatkowo usprawnić proces produkcji gry i zwiększyć motywację członków zespołu.

# Ścieżki komunikacji w zespole

## Ogólne zasady:

- Komunikacja w zespole powinna odbywać się w sposób transparentny i efektywny.
- Należy wykorzystać odpowiednie kanały komunikacji dla różnych typów informacji i zadań.
- Ważne jest, aby dokumentować wszystkie istotne decyzje i ustalenia.
- Należy regularnie spotykać się i omawiać postępy w projekcie.

## Szczegółowe ścieżki komunikacji:

- **Project Manager:**
  - Otrzymuje zadania od klienta lub zarządu firmy.
  - Przekazuje zadania członkom zespołu i określa terminy realizacji.
  - Komunikuje się z klientem lub zarządem firmy na temat postępów w projekcie.
  - Spotyka się z członkami zespołu w celu omówienia postępów i rozwiązywania problemów.
- **Game Designer:**
  - Współpracuje z Project Managerem w celu określenia wymagań funkcjonalnych gry.
  - Opracowuje koncepcję rozgrywki, zasady gry i strukturę gry.
  - Konsultuje swoje pomysły z innymi członkami zespołu.
  - Dokumentuje swoje pomysły i decyzje.
- **Lead Developer:**
  - Otrzymuje specyfikację techniczną od Game Designera.
  - Opracowuje architekturę systemu gry.
  - Programuje grę i zarządza kodem źródłowym.
  - Współpracuje z UI/UX Designerem w celu integracji interfejsu użytkownika.
  - Testuje grę i naprawia błędy.
- **UI/UX Designer:**
  - Współpracuje z Game Designerem w celu określenia wymagań dotyczących interfejsu użytkownika.
  - Opracowuje prototypy interfejsu użytkownika i testuje je z użytkownikami.
  - Tworzy grafiki interfejsu użytkownika.
  - **Współpracuje** z Lead Developerem w celu integracji interfejsu użytkownika z grą.
- **Graphic Designer:**
  - Otrzymuje specyfikację graficzną od Game Designera.
  - Tworzy grafiki tła, postaci, przedmiotów i innych elementów wizualnych gry.
  - Animuje elementy gry.
  - Współpracuje z UI/UX Designerem w celu integracji elementów wizualnych z interfejsem użytkownika.
- **QA Tester:**
  - Otrzymuje grę od Lead Developera.
  - Testuje grę pod kątem błędów i niedociągnięć.
  - Zgłasza błędy Lead Developerowi.

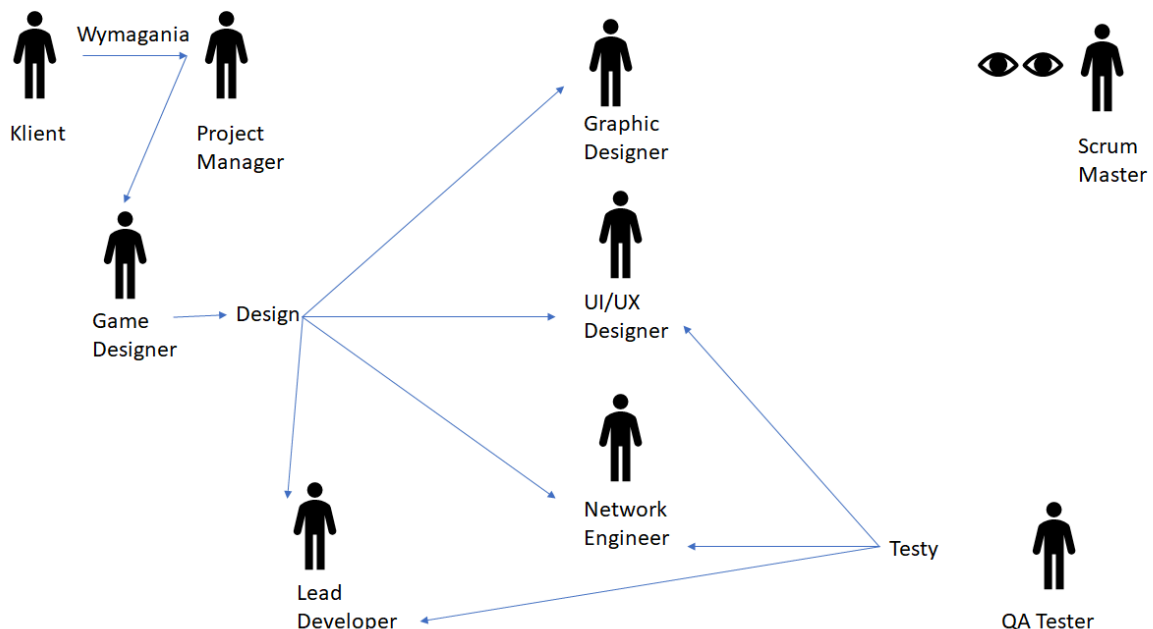
- Testuje grę ponownie po naprawieniu błędów.
- **Network Engineer:**
  - Współpracuje z Project Managerem w celu określenia wymagań infrastruktury sieciowej.
  - Konfiguruje serwery i infrastrukturę sieciową.
  - Monitoruje wydajność gry i serwerów.
  - Rozwiązuje problemy z siecią.

### Sprawdzanie wykonania zadań:

- **Project Manager** regularnie **spotyka się** z członkami zespołu, aby omówić postępy w realizacji zadań.
- **Lead Developer recenzuje kod** napisany przez innych programistów.
- **UI/UX Designer testuje prototypy** interfejsu użytkownika z użytkownikami.
- **Graphic Designer prezentuje swoje prace** innym członkom zespołu i otrzymuje od nich opinie.
- **QA Tester pisze raporty** z testów i przedstawia je Project Managerowi.
- **Network Engineer monitoruje logi** serwerów i zgłasza wszelkie problemy Project Managerowi.

### Kanały komunikacji:

- **Spotkania** twarzą w twarz
- **Narzędzia do komunikacji online**, takie jak Slack, Discord, czy Jira
- **Poczta elektroniczna**
- **System kontroli wersji (VCS)**, taki jak Git



## Koszt poszczególnych faz projektu:

### Faza 1: Opracowanie koncepcji gry (koszt: 10 000 PLN)

- Celem tej fazy jest dopracowanie szczegółów koncepcji gry, w tym:
  - Określenie szczegółowych zasad i mechanik rozgrywki
  - Zdefiniowanie różnych ról i ich unikalnych umiejętności
  - Stworzenie scenariuszy dla różnych etapów gry (dzień i noc)
  - Opracowanie systemu głosowania i eliminacji graczy
  - Określenie wymogów technicznych gry

W tej fazie kluczową rolę odgrywa zespół projektowy, składający się z:

- **Game Designera:** odpowiedzialnego za stworzenie szczegółowej dokumentacji projektowej gry, uwzględniającej wszystkie aspekty rozgrywki
- **Programisty:** odpowiedzialnego za ocenę wykonalności koncepcji gry pod kątem technicznym
- **Testera:** odpowiedzialnego za identyfikację potencjalnych problemów i błędów w koncepcji gry

### Faza 2: Projektowanie interfejsu użytkownika i doświadczenia użytkownika (koszt: 25 000 PLN)

Celem tej fazy jest stworzenie intuicyjnego i estetycznego interfejsu użytkownika (UI) oraz zapewnienie płynnego i przyjemnego doświadczenia użytkownika (UX) podczas gry. Obejmuje to:

- Stworzenie prototypów interfejsu użytkownika i przeprowadzenie testów z użytkownikami
- Zaprojektowanie ekranów gry, menu, ikon i innych elementów graficznych
- Opracowanie systemu nawigacji i interakcji w grze
- Zapewnienie dostępności gry dla osób z niepełnosprawnościami

W tej fazie kluczową rolę odgrywa zespół UI/UX, składający się z:

- **UI/UX Designerów:** odpowiedzialnych za zaprojektowanie interfejsu użytkownika i zapewnienie spójnego doświadczenia użytkownika
- **Grafików:** odpowiedzialnych za stworzenie elementów graficznych interfejsu użytkownika
- **Testerów UX:** odpowiedzialnych za ocenę użyteczności i przyjemności korzystania z interfejsu użytkownika

### Faza 3: Projektowanie grafiki i animacji (koszt: 15 000 PLN)

Celem tej fazy jest stworzenie wysokiej jakości grafiki i animacji, które budują atmosferę gry i zapewniają wciągające doznania wizualne. Obejmuje to:

- Stworzenie koncepcji graficznych i style przewodnie gry
- Opracowanie modeli 3D postaci i obiektów

- Tworzenie animacji postaci, ruchów i efektów specjalnych
- Zaprojektowanie interfejsów graficznych i elementów menu

W tej fazie kluczową rolę odgrywa zespół graficzny, składający się z:

- **Art Directora:** odpowiedzialnego za nadzorowanie całego procesu tworzenia grafiki i zapewnienie spójności stylu wizualnego gry
- **Grafików 3D:** odpowiedzialnych za tworzenie modeli 3D postaci i obiektów
- **Animatorów:** odpowiedzialnych za tworzenie animacji postaci, ruchów i efektów specjalnych
- **UI/UX Designerów:** odpowiedzialnych za integrację grafiki z interfejsem użytkownika

#### **Faza 4: Implementacja mechaniki gry i programowanie (koszt: 50 000 PLN)**

Celem tej fazy jest przekształcenie koncepcji gry w działający produkt. Obejmuje to:

- Napisanie kodu gry w wybranym języku programowania
- Implementacja mechanik gry, zasad i logiki rozgrywki
- Zintegrowanie grafiki, animacji i dźwięku z kodem gry
- Opracowanie systemu sieciowego dla rozgrywki wieloosobowej

W tej fazie kluczową rolę odgrywa zespół programistów, składający się z:

- **Programistów Game:** odpowiedzialnych za napisanie kodu gry, implementację mechanik gry i logiki rozgrywki
- **Programistów Sieciowych:** odpowiedzialnych za opracowanie systemu sieciowego dla rozgrywki wieloosobowej
- **Testerów:** odpowiedzialnych za identyfikowanie i naprawianie błędów w kodzie gry

#### **Faza 5: Testowanie i poprawki (koszt: 20 000 PLN)**

Celem tej fazy jest dopracowanie gry i usunięcie wszelkich błędów lub problemów. Obejmuje to:

- Przeprowadzenie testów funkcjonalnych i integracyjnych gry, aby upewnić się, że wszystkie funkcje działają poprawnie i są ze sobą spójne.
- Testowanie gry z różnymi grupami użytkowników, aby zidentyfikować potencjalne problemy z grywalnością, interfejsem użytkownika i doświadczeniem użytkownika.
- Naprawianie błędów i problemów zgłoszonych przez testerów.
- Optymalizacja wydajności gry.

W tej fazie kluczową rolę odgrywa zespół testerów, składający się z:

- **Testerów Funkcjonalnych:** odpowiedzialnych za przeprowadzanie testów funkcjonalnych i integracyjnych gry
- **Testerów UX:** odpowiedzialnych za ocenę grywalności i interfejsu użytkownika
- **Testerów Wydajności:** odpowiedzialnych za ocenę wydajności gry i identyfikację potencjalnych problemów z płynnością
-

## Faza 6: Konfiguracja serwerów i infrastruktury sieciowej (koszt: 10 000 PLN)

Celem tej fazy jest przygotowanie gry do rozgrywki wieloosobowej online. Obejmuje to:

- Wykupienie i konfiguracji serwerów do hostowania gry
- Wdrożenie oprogramowania sieciowego i zabezpieczeń
- Przeprowadzenie testów obciążeniowych, aby upewnić się, że infrastruktura sieciowa może obsłużyć dużą liczbę graczy
- Monitorowanie wydajności serwerów i infrastruktury sieciowej

W tej fazie kluczową rolę odgrywa zespół inżynierów sieciowych, składający się z:

- **Inżynierów Sieciowych:** odpowiedzialnych za konfigurację serwerów, wdrożenie oprogramowania sieciowego i zapewnienie bezpieczeństwa
- **Administratorów Systemu:** odpowiedzialnych za monitorowanie wydajności serwerów i infrastruktury sieciowej
- **Testerów Sieciowych:** odpowiedzialnych za przeprowadzenie testów obciążeniowych i identyfikację potencjalnych problemów z siecią

## Kosztorys projektu

Pozycja	Przykładowa stawka/rata	Koszt
Koszty pracy personelu:		
Project Manager	80 PLN/godz. x 480 godz.	38400 PLN
Game Designer	70 PLN/godz. x 480 godz.	33600 PLN
Lead Developer	100 PLN/godz. x 480 godz.	48000 PLN
UI/UX Designer	60 PLN/godz. x 480 godz.	28800 PLN
Graphic Designer	70 PLN/godz. x 480 godz.	33600 PLN
QA Tester	50 PLN/godz. x 480 godz.	24000 PLN
Network Engineer	90 PLN/godz. x 480 godz.	43200 PLN
Scrum Master	50 PLN/godz. x 480 godz.	24000 PLN

Koszty sprzętu komputerowego:		
Komputery (10 pracowników)	4000 PLN/komputer x 10	40000 PLN
Serwery (2 sztuki)	10000 PLN/serwer x 2	20000 PLN
Koszty oprogramowania:		
Licencje oprogramowania	5000 PLN/licencja x 7	35000 PLN
Pozostałe koszty:		
Hosting (6 miesięcy)	1000 PLN/miesiąc x 6	6000 PLN
Koszty testowania	5000 PLN (jednorazowo)	5000 PLN
Koszty marketingu	10000 PLN (szacowane)	10000 PLN
Łączny koszt projektu:		340600 PLN

## Raport postępu prac dla wybranego okresu wraz z przypisaną listą zasobów wykorzystanych w tym okresie oraz poniesionych kosztów

Wybrany okres: **Faza 2 - projektowanie i implementacja interfejsu użytkownika i doświadczenia użytkownika**

Faza 2 projektu gry "Mafia" skupiała się na projektowaniu i implementacji interfejsu użytkownika (UI) oraz doświadczenia użytkownika (UX). Celem tej fazy było stworzenie intuicyjnego, estetycznego oraz funkcjonalnego interfejsu, który umożliwiłby graczom płynne i satysfakcjonujące korzystanie z gry.

W fazie projektowania skoncentrowaliśmy się na analizie potrzeb użytkowników oraz zdefiniowaniu kluczowych funkcji i elementów interfejsu. Przeprowadziliśmy badania ankietowe oraz wywiady z potencjalnymi graczami, aby lepiej zrozumieć ich oczekiwania i preferencje.

### Stworzone elementy Interfejsu Użytkownika:

- **Menu główne:** Stworzyliśmy prosty, ale przejrzysty ekran główny, zawierający opcje takie jak rozpoczęcie nowej gry, dołączenie do istniejącej sesji, czy opcje konfiguracyjne.
- **Lobby gry:** Wprowadziliśmy interaktywne lobby, gdzie gracze mogą dołączyć do istniejącej gry, zapraszać znajomych oraz dostosować ustawienia przed rozpoczęciem rozgrywki.
- **Panel gry:** Zaprojektowaliśmy panel, który prezentuje informacje o aktualnej sesji gry, takie jak lista graczy, ich role oraz aktualny etap rozgrywki.
- **Karty postaci:** Każdy gracz ma teraz dostęp do swojej karty postaci, na której znajdują się informacje o roli, umiejętnościach i zadaniach do wykonania.
- **System głosowania:** Stworzyliśmy interfejs umożliwiający graczom oddanie swojego głosu podczas procesu głosowania na podejrzanego gracza.

Dotychczasowy koszt wykonania projektu - 1 miesiąc (wg. szczegółowego opisu prac w projekcie): 74 800 PLN

Wykorzystane zasoby: komputery dla 4 członków zespołu, licencja Adobe Creative Cloud, Figma.

Wykorzystane metody zarządzania jakością w tej fazie:

1. **Analiza Heurystyczna:** Przegląd interfejsu przez doświadczonego eksperta w celu identyfikacji potencjalnych problemów zgodnie z zestawem heurystyk projektowania interakcji użytkownika.
2. **Testowanie A/B:** Porównywanie dwóch różnych wersji interfejsu wśród użytkowników, aby określić, która wersja jest bardziej skuteczna pod względem wydajności, satysfakcji użytkownika itp. Pozwala to na podejmowanie świadomych decyzji projektowych na podstawie danych.
3. **Badania Ankiety:** Przeprowadzanie ankiet wśród użytkowników, aby zbadać ich preferencje dotyczące interfejsu, wyglądu, funkcjonalności itp. Pozwala to na lepsze zrozumienie potrzeb użytkowników i dostosowanie interfejsu do ich oczekiwań.

W trakcie realizacji projektu tworzenia gry "Mafia" mogą pojawić się różnorodne ryzyka, które mogą zakłócić harmonię procesu i wpłynąć na ostateczny rezultat. Oto kilka potencjalnych ryzyk:

1. **Opóźnienia w harmonogramie:** Mogą być spowodowane nieprzewidzianymi problemami technicznymi, trudnościami w koordynacji pracy zespołu, błędami w oszacowaniu czasu lub zmianami wymagań klienta.



2. **Niedoskonała komunikacja:** Brak klarownej komunikacji między członkami zespołu może prowadzić do nieporozumień, opóźnień w dostarczaniu zadań oraz niewłaściwego wykonywania zadań.
3. **Brak zasobów:** Może wystąpić niedostateczna ilość środków finansowych, ludzkich lub technicznych potrzebnych do zrealizowania projektu, co może prowadzić do opóźnień lub obniżenia jakości produktu.
4. **Zmiany w wymaganiach:** Niezrozumienie lub niestabilność wymagań klienta może prowadzić do częstych zmian, które mogą wpływać na harmonogram, jakość i koszty projektu.
5. **Problemy techniczne:** Mogą obejmować awarie sprzętu, błędy programistyczne, problemy ze zgodnością platformy, co może prowadzić do opóźnień w dostarczaniu funkcjonalności.
6. **Niewłaściwa ocena ryzyka:** Brak identyfikacji i oceny ryzyka na wczesnym etapie projektu może skutkować niewłaściwym przygotowaniem na potencjalne zagrożenia, co może prowadzić do nieprzewidzianych problemów w trakcie realizacji projektu.
7. **Konflikty w zespole:** Napięcia między członkami zespołu, niejasne podziały obowiązków czy różnice wizji mogą prowadzić do konfliktów, które mogą zakłócić efektywność pracy.
8. **Brak zaangażowania klienta:** Jeśli klient nie jest zaangażowany w projekt, może to prowadzić do niejasności co do oczekiwań i wymagań, co z kolei może prowadzić do niezadowolenia klienta z końcowego produktu.
9. **Bezpieczeństwo danych:** W przypadku gier online istnieje ryzyko naruszenia prywatności danych użytkowników, ataków hakerskich lub wycieku danych, co może zaszkodzić reputacji projektu.
10. **Nieprzewidziane zdarzenia zewnętrzne:** Mogą obejmować zmiany regulacyjne, katastrofy naturalne, pandemie czy inne czynniki zewnętrzne, które mogą mieć wpływ na realizację projektu.

Skuteczne zarządzanie ryzykiem w trakcie realizacji projektu obejmuje identyfikację potencjalnych zagrożeń, opracowanie strategii ich zarządzania oraz monitorowanie i reagowanie na zmiany w trakcie trwania projektu. Poniżej przedstawiono sposoby minimalizacji ryzyk podczas realizacji projektu:

1. **Przygotowanie Realistycznego Planu Projektu:** Staranne oszacowanie zasobów, harmonogramu i wymagań projektu pozwoli uniknąć niepotrzebnych opóźnień i niedomówień, minimalizując ryzyko niemożności dotrzymania terminów.
2. **Regularna Komunikacja:** Zachęcanie do otwartej, regularnej komunikacji między członkami zespołu oraz z klientem może zminimalizować ryzyko nieporozumień i utrzymać wszystkich zaangażowanych na bieżąco.
3. **Monitorowanie Postępów i Przewidywanie Problemów:** Regularne sprawdzanie postępów projektu i śledzenie wskaźników wydajności pozwala na wcześniejsze wykrycie potencjalnych problemów i podjęcie działań zaradczych.
4. **Szkolenia i Rozwój Zespołu:** Inwestowanie w rozwój kompetencji i umiejętności członków zespołu może pomóc w zapobieganiu błędom i minimalizowaniu ryzyka związanego z niedoskonałościami w pracy.
5. **Regularne Przeglądy Kodu i Testy Jakości:** Stałe przeglądy kodu, testy jednostkowe i testy integracyjne pomagają wykryć błędy i niedociągnięcia na

wczesnych etapach, co minimalizuje ryzyko pojawienia się poważnych problemów w późniejszych fazach projektu.

6. **Zapewnienie Zasobów:** Staranne zarządzanie zasobami finansowymi, ludzkimi i technicznymi pozwala na uniknięcie niedostatków w trakcie realizacji projektu, minimalizując ryzyko opóźnień i niedoskonałości.
7. **Regularne Spotkania z Klientem:** Regularne spotkania z klientem pozwalają na bieżące monitorowanie zmian w wymaganiach i dostosowywanie planów projektowych, minimalizując ryzyko niezgodności z oczekiwaniami klienta.
8. **Zastosowanie Metodologii Zarządzania Projektem:** Wykorzystanie sprawdzonych metodologii zarządzania projektem, takich jak Agile lub Scrum, może pomóc w elastycznym reagowaniu na zmiany i minimalizowaniu ryzyka niemożności dotrzymania terminów.
9. **Backup i Plan Awaryjny:** Stworzenie planu awaryjnego i regularne tworzenie kopii zapasowych danych zapewniają ochronę przed nieoczekiwanymi awariami sprzętu lub oprogramowania, minimalizując ryzyko utraty danych i opóźnień.
10. **Analiza i Wnioski po Zakończeniu Projektu:** Po zakończeniu projektu warto przeprowadzić szczegółową analizę, aby zidentyfikować czynniki, które przyczyniły się do sukcesu lub niepowodzenia projektu, co pozwoli na lepsze przygotowanie się do przyszłych projektów.

## Planowanie zarządzania projektem gry "Mafia" z wykorzystaniem metodyki zwinnej

**Metodyka zwinna** to podejście do zarządzania projektem, które opiera się na krótkich cyklach rozwojowych, iteracjach i stałej komunikacji między zespołem i klientem. W przypadku projektu gry "Mafia" można zastosować jedną z odmian metodyki zwinnej, np. **Scrum**.

**Scrum** to zwinna struktura zarządzania projektem, która dzieli projekt na krótkie iteracje zwane sprintami. Sprints zazwyczaj trwają 1-4 tygodnie i obejmują następujące etapy:

- **Planowanie sprintu:** Na początku sprintu zespół ustala cele sprintu i wybiera zadania, które zostaną zrealizowane w tym czasie.
- **Codzienne spotkania:** Codziennie zespół spotyka się, aby omówić postępy w pracy nad zadaniami, zidentyfikować potencjalne problemy i zaplanować dalsze działania.
- **Przegląd sprintu:** Na koniec sprintu zespół prezentuje klientowi zrealizowane zadania i zbiera od niego opinie.
- **Retrospektywa sprintu:** Po zakończeniu sprintu zespół omawia przebieg sprintu, identyfikuje obszary do poprawy i wprowadza zmiany w procesie pracy.

## Zastosowanie metodyki Scrum do projektu gry "Mafia":

### 1. Zespół:

- **Product Owner (PO):** Odpowiedzialny za reprezentowanie klienta i definiowanie backlogu produktu, czyli listy zadań do zrealizowania w projekcie.
- **Scrum Master:** Odpowiedzialny za nadzorowanie procesu Scrum, usuwanie przeszkód i zapewnienie optymalnych warunków do pracy zespołu.
- **Zespół programistów:** Odpowiedzialny za implementację mechaniki gry, programowanie, tworzenie grafiki i animacji.
- **Zespół testerów:** Odpowiedzialny za testowanie gry, identyfikowanie błędów i problemów.

### 2. Backlog produktu:

- Backlog produktu powinien zawierać szczegółowe opisy zadań do zrealizowania w projekcie, priorytetyzowane według ich ważności.
- Zadania powinny być podzielone na mniejsze, łatwiej mierzalne elementy, aby ułatwić ich planowanie i realizację.

### 3. Sprints:

- Sprints powinny trwać 1-2 tygodnie, aby zapewnić odpowiednią częstotliwość iteracji i feedbacku.
- Na początku każdego sprintu zespół planuje sprint, wybierając zadania z backlogu produktu, które zostaną zrealizowane w tym czasie.
- Codzienne spotkania powinny trwać około 15 minut i służyć omówieniu postępów w pracy nad zadaniami, zidentyfikowaniu potencjalnych problemów i zaplanowaniu dalszych działań.
- Przegląd sprintu powinien trwać około 1 godzinę i służyć prezentacji klientowi zrealizowanych zadań i zbierania od niego opinii.
- Retrospektywa sprintu powinna trwać około 1 godzinę i służyć omówieniu przebiegu sprintu, identyfikacji obszarów do poprawy i wprowadzeniu zmian w procesie pracy.

### 4. Artefakty:

- **Lista zadań (backlog produktu):** Zawiera wszystkie zadania do zrealizowania w projekcie, priorytetyzowane według ich ważności.
- **Burn-down chart:** Wizualizuje postępy w realizacji zadań ze sprintu.
- **Sprint backlog:** Zawiera listę zadań wybranych do realizacji w danym sprincie.

## Korzyści z zastosowania metodyki Scrum:

- **Szybkie dostarczanie wartości:** Metodyka Scrum pozwala na szybkie dostarczanie funkcjonalnych części gry klientowi, co umożliwia zbieranie od niego feedbacku i wprowadzanie zmian na bieżąco.

- **Lepsza komunikacja:** Codzienne spotkania i przeglądy sprintu zapewniają stałą komunikację między zespołem i klientem, co pomaga w identyfikowaniu i rozwiązywaniu problemów.
- **Większa elastyczność:** Metodyka Scrum pozwala na dostosowywanie planu projektu do zmieniających się potrzeb klienta.
- **Wyższa jakość:** Ciągłe testowanie i iterowanie pomaga w tworzeniu wysokiej jakości gry.

## Kalendarium pracy zespołu w metodyce scrum

### Backlog Planning

- Uczestnicy: Product Owner (PO), Scrum Master, Zespół programistów, Zespół testerów (opcjonalnie)
- Czas trwania: 2-4 godziny
- Cel: Ustalenie priorytetów dla Backlogu produktu i zaplanowanie prac na przyszłe Sprinty.

#### Scenariusz:

1. Product Owner przedstawia Backlog produktu, zawierający wszystkie zadania do wykonania w projekcie.
2. Zespół zadaje pytania i dyskutuje nad poszczególnymi zadaniami.
3. Product Owner wyjaśnia priorytety dla zadań w oparciu o potrzeby biznesowe i wartość dla użytkownika.
4. Zespół wspólnie szacuje czas realizacji dla priorytetowych zadań.
5. Product Owner, w porozumieniu z zespołem, ustala zakres prac na najbliższe Sprinty.
6. Scrum Master omawia proces pracy w ramach metodyki Scrum.

### Sprint Planning

- Uczestnicy: Product Owner, Scrum Master, Zespół programistów, Zespół testerów
- Czas trwania: 1-2 godziny
- Cel: Ustalenie celów Sprintu i zaplanowanie zadań do wykonania.

#### Scenariusz:

1. Product Owner przedstawia priorytety z Backlogu produktu wybrane na bieżący Sprint.
2. Zespół dyskutuje nad wybranymi zadaniami i dzieli je na mniejsze, mierzalne elementy (user stories).
3. Zespół szacuje czas realizacji dla poszczególnych user stories.
4. Zespół ustala cel Sprintu w oparciu o wybrane user stories.
5. Scrum Master omawia plan Sprintu i odpowiada na pytania zespołu.

### Daily Scrum

- Uczestnicy: Scrum Master, Zespół programistów, Zespół testerów (opcjonalnie)

- Czas trwania: 15 minut
- Cel: Codzienna synchronizacja zespołu i omówienie postępów w pracy.

#### **Scenariusz:**

1. Scrum Master rozpoczyna spotkanie i zadaje trzy pytania każdemu członkowi zespołu:
  - Co zrobiłem/am wczoraj w ramach Sprintu?
  - Co zamierzam zrobić/zrobić dziś w ramach Sprintu?
  - Czy napotkałem/am jakieś przeszkody lub problemy?
2. Zespół dyskutuje krótko nad odpowiedziami i ustala dalsze działania.
3. Scrum Master pomaga w identyfikowaniu i usuwaniu przeszkód.

#### **Backlog Refinement**

- Uczestnicy: Product Owner, Zespół programistów, Zespół testerów (opcjonalnie)
- Czas trwania: 1-2 godziny (regularnie w trakcie trwania Sprintu)
- Cel: Udoskonalanie i doprecyzowanie zadań w Backlogu produktu.

#### **Scenariusz:**

1. Product Owner przedstawia zadania z Backlogu produktu.
2. Zespół dyskutuje nad zadaniami, dzieli je na mniejsze elementy i szacuje ich czas realizacji.
3. Product Owner aktualizuje priorytety dla zadań w oparciu na nowe informacje.
4. Zespół wspólnie doprecyzowuje szczegóły zadań.

#### **Sprint Review**

- Uczestnicy: Product Owner, Scrum Master, Zespół programistów, Zespół testerów, Stakeholderzy (opcjonalnie)
- Czas trwania: 1 godzina (na tydzień Sprintu)
- Cel: Prezentacja zrealizowanych prac ze Sprintu i zbieranie feedbacku.

#### **Scenariusz:**

1. Zespół programistów prezentuje ukończone user stories i przeprowadza demo gry.
2. Product Owner omawia osiągnięte cele Sprintu.
3. Zespół testerów przedstawia raport z testów i zidentyfikowane błędy.
4. Stakeholderzy (jeśli obecni) zadają pytania i udzielają feedbacku.
5. Product Owner i zespół omawiają feedback i ustalają dalsze działania.

#### **Sprint Retrospective**

- Uczestnicy: Scrum Master, Zespół programistów, Zespół testerów
- Czas trwania: 1-2 godziny
- Cel: Omówienie przebiegu Sprintu i wprowadzenie usprawnień do procesu pracy.

#### **Scenariusz:**

6. Zespół wspólnie omawia przebieg Sprintu:
  - Co poszło dobrze?
  - Co poszło źle?
  - Co można poprawić w następnym Sprincie?
7. Zespół identyfikuje obszary do poprawy w zakresie:
  - Procesu pracy
  - Narzędzi
  - Współpracy
  - Komunikacji
8. Zespół ustala konkretne działania w celu wprowadzenia usprawnień.
9. Scrum Master dokumentuje ustalenia z retrospektywy.

### **Monthly Project Update**

- Uczestnicy: Product Owner, Scrum Master, Stakeholderzy
- Czas trwania: 30 minut - 1 godzina
- Cel: Przedstawienie ogólnego stanu projektu i omówienie kluczowych kwestii.

### **Scenariusz:**

1. Product Owner przedstawia raport z postępów w projekcie:
  - Zrealizowane zadania ze Sprintów
  - Otwarte elementy w Backlogu produktu
  - Aktualny stan budżetu
  - Kluczowe wskaźniki wydajności (KPI) projektu (np. liczba zgłoszonych błędów, poziom zadowolenia testerów)
2. Scrum Master omawia stan procesu Scrum i identyfikuje potencjalne problemy.
3. Stakeholderzy zadają pytania i udzielają feedbacku.
4. Product Owner i Scrum Master odpowiadają na pytania i omawiają plany na kolejne miesiące.
5. Ustalane są kluczowe decyzje i działania dotyczące dalszego rozwoju projektu.

## **Przykładowy sprint**

### **Dzień 1:**

#### **Backlog Planning:** 2 godziny (rano)

Ustalenie priorytetów dla Backlogu produktu i planowanie prac na Sprint 1.

### **Dzień 2:**

#### **Sprint Planning:** 1 godzina (rano)

Ustalenie celów Sprintu 1 i zaplanowanie zadań do wykonania.

### **Dni 3-10:**

#### **Daily Scrum:** 15 minut (codziennie rano)

Codzienna synchronizacja zespołu i omówienie postępów w pracy.

### **Dzień 5:**

#### **Backlog Refinement:** 1 godzina (po południu)

Udoskonalanie i doprecyzowanie zadań w Backlogu produktu.

#### **Dzień 10:**

**Sprint Review:** 1 godzina (po południu)

Prezentacja zrealizowanych prac ze Sprintu 1 i zbieranie feedbacku.

#### **Dzień 12:**

**Sprint Retrospective:** 1-2 godziny (po południu)

Omówienie przebiegu Sprintu 1 i wprowadzenie usprawnień do procesu pracy.

#### **Dzień 14:**

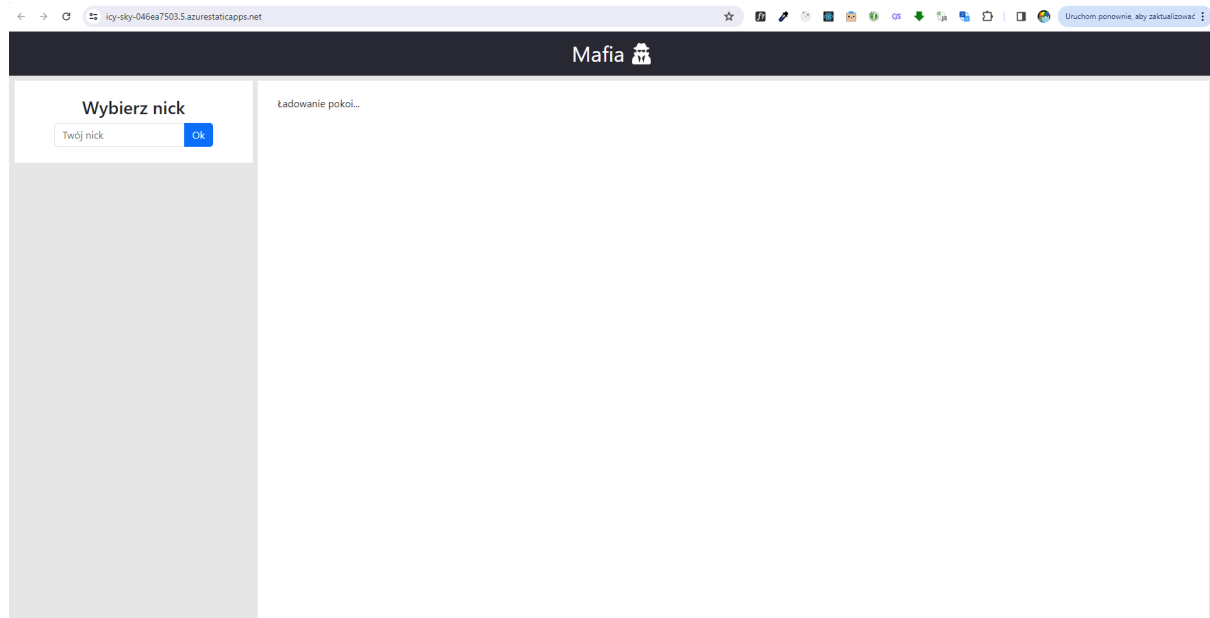
**Monthly Project Update:** 30 minut - 1 godzina (po południu)

Prezentowanie ogólnego stanu projektu i omówienie kluczowych kwestii.

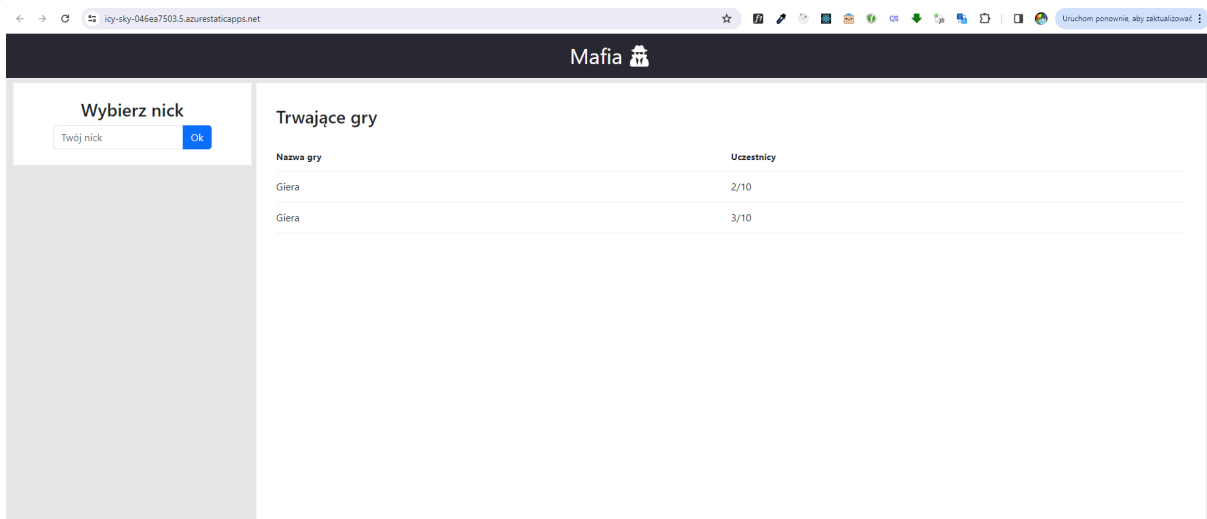
## Implementacja projektu

Implementacja projektu znajduje się w [repozytorium na serwisie github](#). Wersja live aplikacji dostępna jest obecnie [tutaj](#). Link ten może jednak ulec zmianie, lecz aktualny link do wersji live znajduje się zawsze na stronie repozytorium. Dostęp do aplikacji chroniony jest hasłem, w celu uniknięcia nieprzewidzianego ruchu który generuje dodatkowe koszty. Hasło do aplikacji to `zbzF>h%sf`Q-dkwDT2h:D,+C*3nmaTXN`.

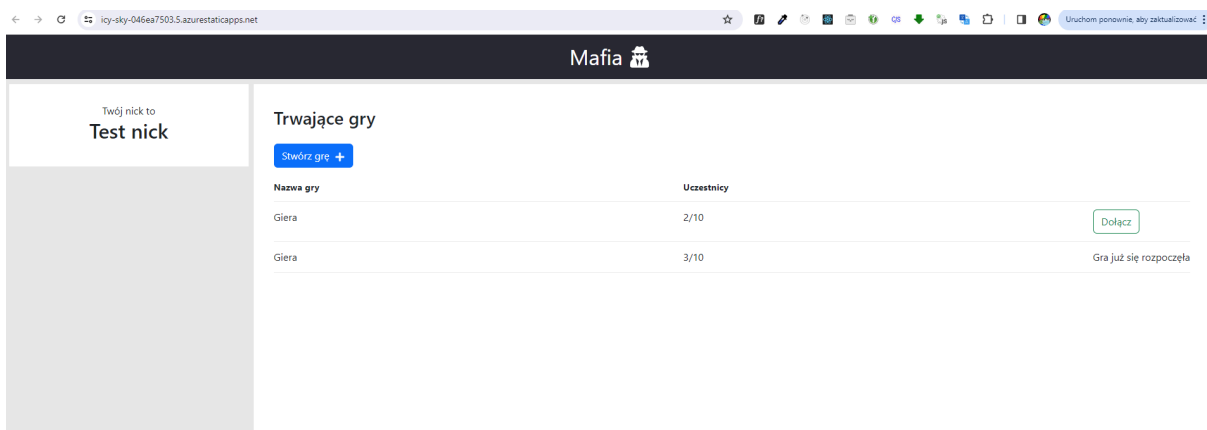
Poniżej znajdują się zrzuty ekranu prezentujący aktualny wygląd poszczególnych widoków aplikacji w przebiegu przykładowej sesji gry.



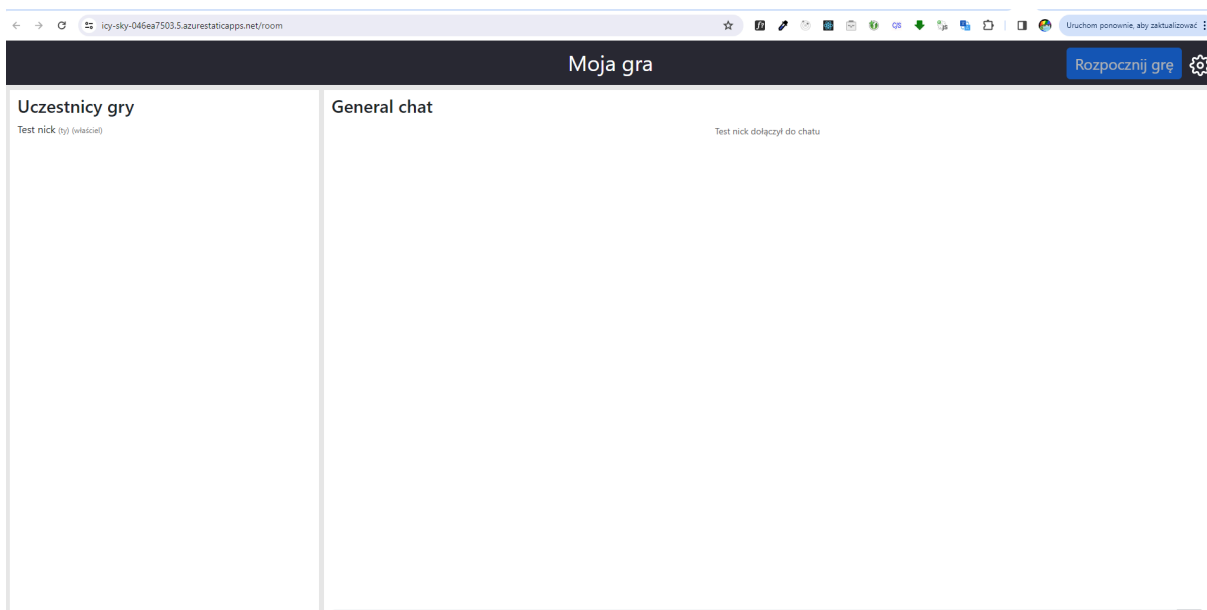
Rysunek 1: widok główny podczas ładowania dostępnych pokoi



Rysunek 2: lista dostępnych pokoi gry przed wybraniem nicku

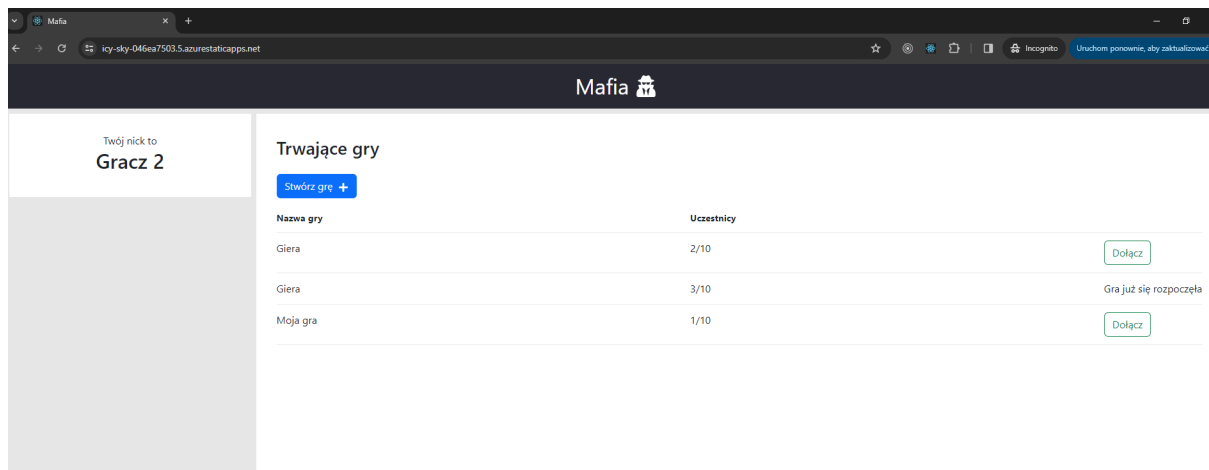


Rysunek 3: lista dostępnych pokoi gry po wybraniu nicku gracza

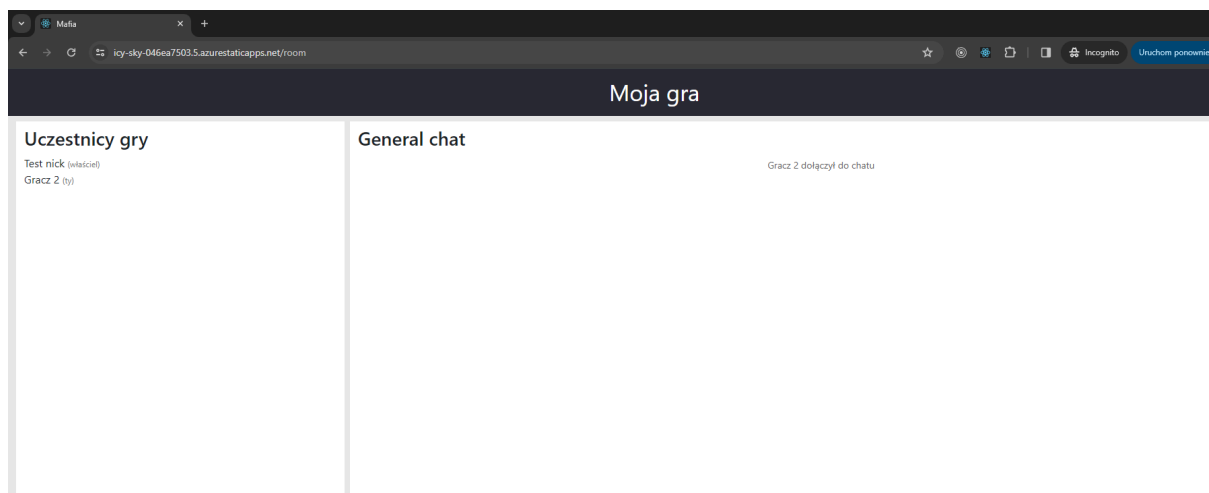


Rysunek 4: widok główny pokoju gry przed jej rozpoczęciem, widziany z perspektywy właściciela pokoju

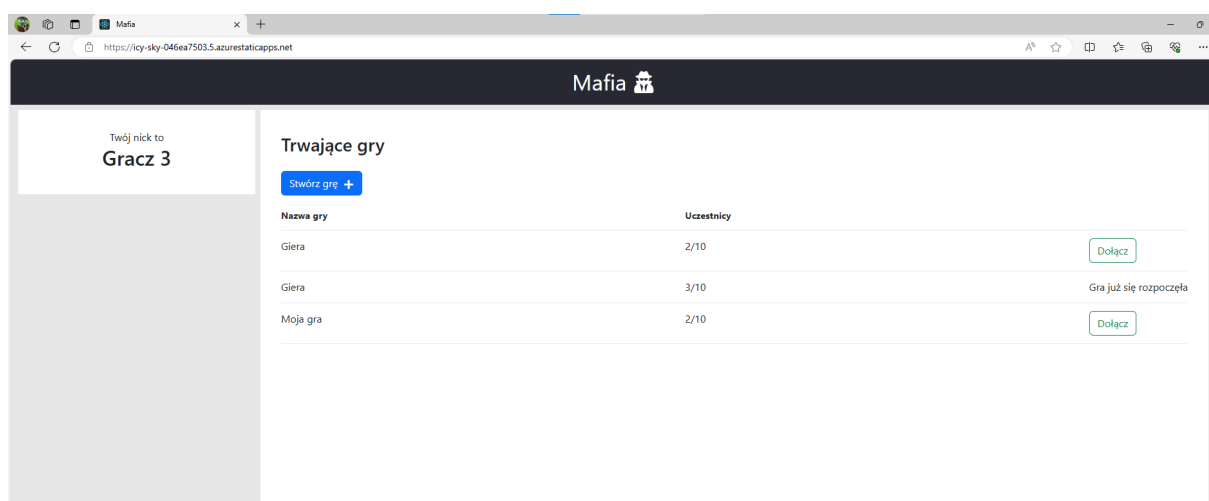




Rysunek 5: widok listy dostępnych pokoi z perspektywy drugiego gracza. Na ostatniej pozycji widać pokój stworzony przez pierwszego gracza na poprzednich zrzutach ekranu



Rysunek 6: widok pokoju gry przed jej rozpoczęciem z perspektywy drugiego gracza



Rysunek 7: widok listy pokoi z perspektywy gracza trzeciego

Rozpocznij grę

## Ustawienia gry

Maksymalna ilość graczy w pokoju

10

Ilość członków mafii

1

Długość rundy

00

min


50

sec

☒

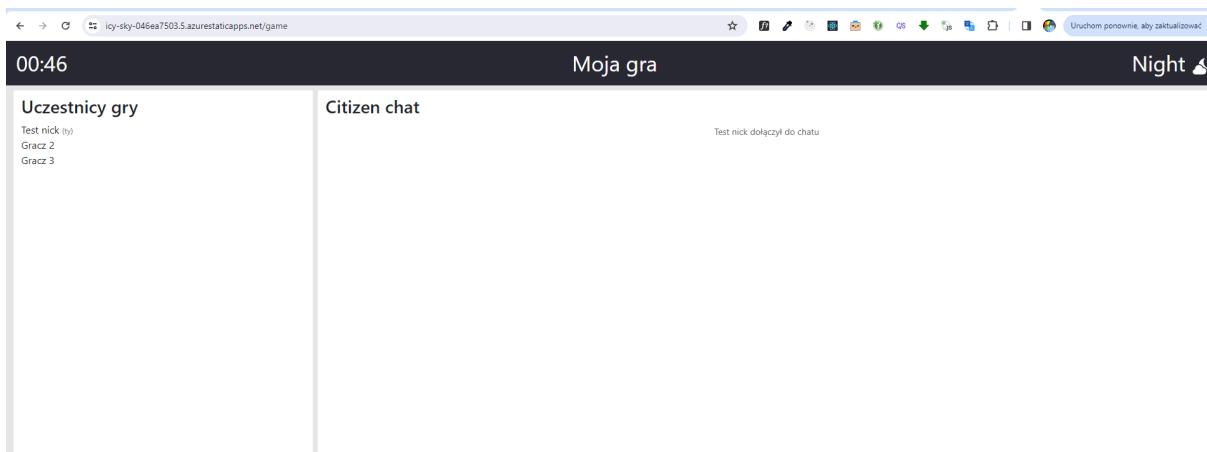
 Pokój publiczny

☒

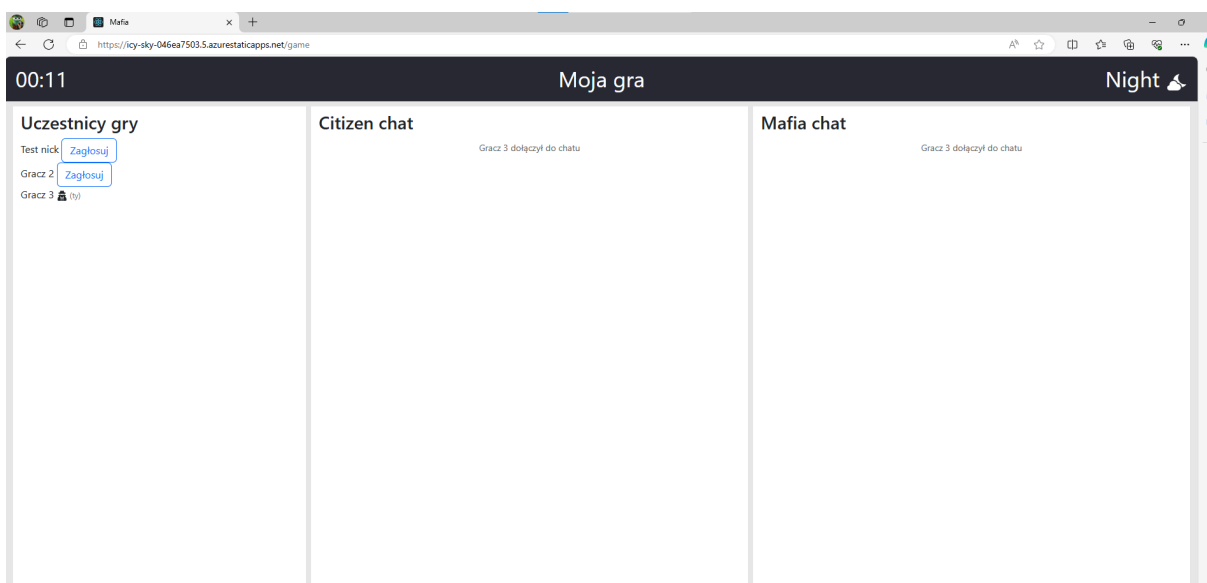
 Głosy widoczne dla innych graczy

Zapisz

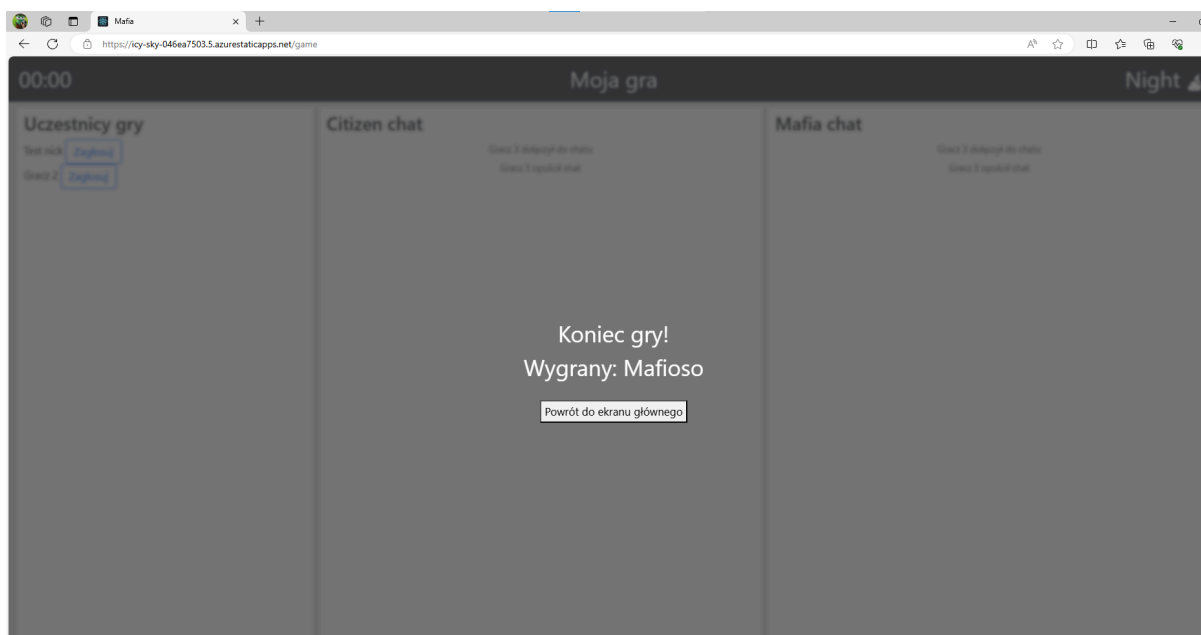
Rysunek 8: widok zmiany ustawień gry



Rysunek 9: widok trwającej gry z perspektywy pierwszego gracza, któremu została przypisana rola mieszkańca miasta. Trwa runda nocna.



Rysunek 10: widok trwającej gry z perspektywy gracza trzeciego, któremu została przypisana rola Mafiosa.



Rysunek 11: widok końcowy gry