

POLITECHNIKA POZNAŃSKA

INFORMATYKA SEMESTR 6

PODSTAWY TELEINFORMATYKI – PROJEKT

KONFERENCJA AUDIO-VIDEO

Prowadzący:

mgr inż. Przemysław Walkowiak



Skład grupy:

Maciej Olejnik

numer indeksu: 125067

e-mail: maciej.olejnik@student.put.poznan.pl

Bartosz Tyczyński

numer indeksu: 126873

e-mail: bartosz.j.tyczynski@student.put.poznan.pl

Erwin Majewski

numer indeksu: 126793

e-mail: erwin.majewski@student.put.poznan.pl

Damian Stube

numer indeksu: 120758

e-mail: damian.stube@student.put.poznan.pl

Spis treści

Charakterystyka ogólna	5
Wybór tematu	5
Opis produktu	5
Podział prac	6
Wymagania funkcjonalne	7
Ogólna funkcjonalność	7
Klasyfikacja funkcjonalności	7
Wymagania нефunkcjonalne	8
Wymagania produktowe	8
Wymagania organizacyjne	8
Wymagania zewnętrzne	9
Charakterystyka użytkowników	9
Narzędzia i środowisko programistyczne	10
Środowisko	10
Narzędzia	10
Model aplikacji	12
Aktorzy	12
Diagram aktywności	12
Diagram sekwencji	14
Problemy i ich rozwiązania	16
Synchronizacja ekranu gry dla obydwu graczy.	16
Przechwytywanie obrazu z kamery.	16
Zapis zmian wprowadzanych przez klienta w grze do stanu i transmisja na serwer.	16
Testy aplikacji	17
Testy funkcjonalności gry przeprowadzane na maszynie lokalnej - dla jednego gracza.	17
Testy funkcjonalności gry z zastosowaniem transmisji - dla 2 graczy.	17
Testy przesyłania obrazu oraz dźwięku w konferencji.	17

Interfejs graficzny	18
Warstwa graficzna aplikacji klienckiej.	18
Instrukcja użytkowania	23
Uruchomienie aplikacji	23
Obsługa aplikacji	23
Dalszy rozwój	28
Podsumowanie	29
Bibliografia	30

1. Charakterystyka ogólna

Dokumentacja zawiera wymagania związane ze stworzeniem serwisu wideo konferencji z funkcjonalnością gier. Niniejszy dokument zawiera jasno sprecyzowane wymagania dotyczące aplikacji takie jak wygląd, funkcjonalność oraz charakterystykę poszczególnych modułów. Zamieszczona została również instrukcja korzystania z serwisu internetowego oraz regulamin. Celem pracy było utworzenie serwisu umożliwiającego użytkownikom interaktywną zabawę prowadzoną poprzez konferencje audiowizualną. Wideokonferencja jest to interaktywna komunikacja multimedialna, realizowana za pomocą urządzeń podłączonych do sieci np. laptop, komputer PC czy też smartphome. Polega ona na przesyłaniu z dużą prędkością bez wielkiej straty jakości danych, dźwięku oraz obrazu w czasie rzeczywistym między odległymi lokalizacjami. Wideokonferencja umożliwia rozmowę dwóch lub kilku osób znajdujących się daleko od siebie zupełnie za darmo. Stanowisko uczestnika wideokonferencji musi być wyposażone w komputer lub urządzenie mobilne z łączem telekomunikacyjnym o szybkości min. 128 kb/s, kamerę wideo, mikrofon i głośniki oraz odpowiednie oprogramowanie. Efektywność systemów konferencji mierzona jest w czasie rzeczywistym w klatkach na sekundę.

1.1 Wybór tematu

Temat zainteresował naszą grupę, ponieważ stwierdziliśmy, że przydałoby się stworzyć nową platformę do wideo konwersacji odróżniającą się od tych już istniejących. Innowacją w naszym systemie jest brak możliwości wyboru partnera do konwersacji. System automatycznie losuje partnera. Dodatkowo użytkownicy aby urozmaicić rozmowę wybierają grę w którą chcą zagrać. Poprzez wybór tego tematu chcieliśmy zgłębić więc dwa zagadnienia: konferencji wideo online i wieloosobowej gry online w formie z jaką się jeszcze nie spotkaliśmy. Takie połączenie, może posłużyć zarówno do zabawy jak i np. do nauki języków, przełamywaniu barier w mówieniu. W intensywnych rozgrywkach pod presją czasu gracz będzie mógł poćwiczyć swoje umiejętności językowe, komunikacyjne przy tym doskonale się bawiąc.

1.2 Opis produktu

Założeniem projektu jest stworzenie serwisu internetowego obsługującego interaktywne konferencje. Odmiennością od popularnych serwisów i aplikacji oferujących możliwość prowadzenia konferencji jest możliwość prowadzenia zabawy w formie gry pomiędzy uczestnikami konferencji. Po dołączeniu użytkownika do serwisu zestawiane jest połączenie z drugim aktywnym na stronie użytkownikiem. Użytkownicy nie muszą być tej samej narodowości i posługiwać się tymi samymi językami. Jedynym wymaganiem jest znajomość angielskiego. Jeden z użytkowników dostaje instrukcje dotyczącą danej wylosowanej gry. Drugiemu

użytkownikowi wyświetla się ekran z planszą gry np. bombą. Zadaniem obu użytkowników jest przekazanie informacji na temat wyświetlanej gry oraz sposobu jej rozwiązania. Użytkownicy muszą ustalić między sobą język jakim się będą porozumiewać oraz sposób przekazywania informacji. Zabawa odbywa się pod presją czasu. Każde zadanie (gra) ma wyznaczony limit czasu w którym powinno być rozwiązane. Zegar z pozostałym czasem znajduje się w widocznym miejscu dla obu użytkowników co powoduje dodatkowy dreszczyk emocji. Głównym celem serwisu jest umożliwienie użytkownikom spędzania miło wolnego czasu, zapoznanie się z nowymi ludźmi oraz poznanie kultury osób z innych krańców świata. Serwis jest dostępny bezpłatnie, jedyne co jest wymagane to posiadanie sprzętu wyposażonego w kamerę internetową, głośnik lub słuchawki oraz mikrofon. Ważnym elementem systemu jest posiadanie odpowiednio szybkiego łącza transmisji danych, aby zapewnić płynność transmisji wideo i dźwięku.

1.3 Podział prac

- Back-End
 - Maciej Olejnik
 - Praca nad dokumentacją.
 - Implementacja funkcjonalności komunikacji.
 -
 - Damian Stube
 - Praca nad dokumentacją.
 - Implementacja funkcjonalności.
 -
- Front-end
 - Erwin Majewski
 - Praca nad dokumentacją.
 - Implementacja funkcjonalności gier.
 - Projekt graficzny gier
 -
 - Bartosz Tyczyński
 - Praca nad dokumentacją.
 - Interfejs użytkownika.
 - Implementacja funkcjonalności interfejsu.
 - Wykorzystanie framework'u Bootstrap do utworzenia interfejsu.

2. Wymagania funkcjonalne

Rozdział opisuje wymagania dotyczące funkcjonalności aplikacji oraz zachowania systemu podczas korzystania z niego przez wielu użytkowników.

2.1 Ogólna funkcjonalność

Serwis powinien oferować następujące funkcje:

- Możliwość dołączenia do wideokonferencji.
- Dobór partnera losowo.
- Wybór gry do zabawy.
- Zmiana partnera w trakcie gry.
- Wybór języka gry i konwersacji.

Wszystkie powyższe funkcjonalności powinny działać bardzo szybko, aby zapewnić płynność transmisji i jakość konwersacji oraz gry. Transmisja wideo powinna być prowadzona w możliwie największej jakości. Zapewnione powinny być minimalne środki bezpieczeństwa dotyczące ochrony danych osobowych i wizerunku. Cała aplikacja powinna być obdarowana obszerną instrukcją obsługi umożliwiającą zapoznanie się z systemem i sposobem jego użytkowania.

2.2 Klasyfikacja funkcjonalności

Tabela 1 zawiera zebrane wszystkie funkcjonalności systemu wraz z ich typem oraz aktorami. Typ funkcjonalności określa jej priorytet dotyczący realizacji całego projektu. Typ konieczny to taki bez którego implementacja projektu się nie powiedzie. Typ oczekiwany to taki który gwarantuje zapewnienie podstawowych zasad działania aplikacji. Natomiast tryb możliwy to taki który nie warunkuje powodzenia projektu. Aktor jest to osoba odpowiedzialna za wykonanie/użytkowanie danej funkcjonalności. Wyróżniamy dwa typy aktorów: administrator mający pełen dostęp do zasobów aplikacji oraz użytkownik, czyli osoba mająca tylko pewną dostępną gamę funkcji.

Tabela 1 Podział funkcjonalności

Funkcjonalność	Typ	Aktorzy
Dołączenie do konwersacji	konieczny	Użytkownik
Dobór partnera	konieczny	Administrator
Wybór gry	oczekiwany	Użytkownik
Wybór języka aplikacji	oczekiwany	Użytkownik
Zmiana partnera	możliwy	Użytkownik
Zapewnienie bezpieczeństwa	oczekiwany	Administrator

3. Wymagania niefunkcjonalne

Rozdział ten formułuje wymagania i czynniki wpływające na implementację serwisu. Zawiera również listę ograniczeń nałożonych na projekt ze względu na stosowane technologie do rozwiązywania konkretnych problemów.

3.1 Wymagania produktowe

- Czas ładowania strony startowej z wyłączeniem komponentu wideo i komponentu gry nie powinien przekraczać 1 s.
- Czas ładowania komponentu wideo nie powinien przekraczać 2s.
- Czas ładowania komponentu gry nie powinien przekraczać 4s.
- Latency rozmowy wideo nie powinno przekraczać 100 ms.
- Jitter w rozmowie wideo nie powinien przekraczać 0.5 ms.
- Maksymalna strata pakietów nie powinna przekraczać 0.1%.
- Opóźnienie uaktualniania stanu gry między graczami nie powinno przekraczać 500 ms.
- System nie powinien być niedostępny dłużej niż 2h tygodniowo
- System powinien być dostępny w językach : angielskim
- System powinien działać na wszystkich platformach

3.2 Wymagania organizacyjne

- Kopia zapasowa bazy kodu i ustawień powinna być wykonywana przy każdej aktualizacji
- System powinien posiadać aktualną instrukcję użytkownika
- System powinien posiadać instrukcje instalacji dla członków projektu
- Projekt powinien być przygotowany w następującym środowisku:
 - node.js wersja 10.5.0
 - HTML 5
 - CSS 3
 - bootstrap wersja 4.0.0
- Wszystkie wersje produktu powinny być umieszczone na repozytorium
- Każda zmiana powinna być udokumentowana.

3.3 Wymagania zewnętrzne

- System nie powinien używać w grach grafik do których nie ma praw autorskich
- System musi być możliwy do zaprezentowania w sali wykładowej
- System powinien być łatwy w obsłudze, nawet dla osoby pierwszy raz korzystającej z aplikacji
- System powinien być elastyczny, modyfikowalny z aktualnymi trendami

3.4 Charakterystyka użytkowników

- Grupa wiekowa: głównie 16 - 40
- Zainteresowanie rozwojem umiejętności miękkich
- Zainteresowanie rozwojem umiejętności językowych
- Ciekawość świata i innych kultur
- Zainteresowanie grami online
- Zainteresowanie grami multiplayer
- Tylko osoby prywatne

4. Narzędzia i środowisko programistyczne

4.1 Środowisko

- **node.js.**

Środowisko uruchomieniowe zaprojektowane do tworzenia wysoce skalowalnych aplikacji internetowych, szczególnie serwerów www napisanych w języku JavaScript. Umożliwia tworzenie aplikacji sterowanych zdarzeniami wykorzystujących asynchroniczny system wejścia-wyjścia.

4.2 Narzędzia

- **GitHub.**

W zespołowej pracy nad projektem programistycznym nieodzowny jest system kontroli wersji. Narzędzie to usprawnia dzielenie się zmianami wprowadzonymi w kodzie, umożliwia przywrócenie poprzedniej wersji programu w przypadku wystąpienia nierozwiązywalnych problemów. Tworzy także kopię zapasową kodu programu na zdalnych serwerach. Wybór GitHuba spośród wielu systemów kontroli wersji został podyktowany faktem, że wszyscy członkowie zespołu mieli wcześniejsze doświadczenie z tą platformą oraz podzielają entuzjazm do korzystania z najlepszych rozwiązań pochodzących od firmy Microsoft.

- **Bootstrap.**

Interfejs graficzny strony internetowej został wystylizowany przy użyciu frameworka CSS - Bootstrapa. Dzięki zastosowaniu tej technologii uzyskano bardziej atrakcyjne wizualnie menu główne - uwzględniając takie elementy jak tło, stylizacja formularza, przyciski akcji, miejsca do wprowadzania danych (*inputy*).

- **npm.**

Menedżer pakietów środowiska node.js. Jego główną funkcjonalnością jest zarządzanie zewnętrznymi bibliotekami, potrzebnymi do zrealizowania projektu. Umożliwia pobieranie znanych pakietów do katalogu zawierającego pliki projektu. Po zainstalowaniu pakietu, dodaje go do listy zależności w pliku package.json w głównym katalogu projektu. Pozwala zbudować aplikację komendą *npm install*. Instalacja sprawdza, czy pakiety znajdujące się na liście zależności faktycznie znajdują się w projekcie. Wykorzystanie menedżera pakietów jest szczególnie przydatne w pracy zespołowej, gdzie tylko programista wykorzystujący dany pakiet musi posiadać wiedzę że należy go umieścić w projekcie. Dla reszty zespołu, dodanie tego pakietu jest zautomatyzowane i nie zaburza harmonogramu pracy (brak konieczności szukania plików biblioteki i ręcznego ich wgrywania do projektu).

- **PixiJS.**

Silnik służący do generowania grafiki dwuwymiarowej. Umożliwia wyświetlanie, animowanie oraz zarządzanie interaktywnymi grafikami.

- **Pixi Particles.**

Biblioteka do PixiJS umożliwiająca emitowanie cząsteczek. Emisja cząsteczek umożliwia wizualne naśladowanie zjawisk przyrodniczych (np. deszczu, śnieżycy, płomieni itp.). W projekcie zostało użyte do symulowania eksplozji.

- **Pixi Particles Editor.**

Interaktywny edytor do tworzenia efektów z cząsteczek. Po dobraniu kilkunastu parametrów (takich jak: częstotliwość emisji cząsteczek, maksymalna ilość istniejących cząsteczek, przyspieszenie i szybkość cząsteczek), następuje wygenerowanie pliku JSON z wartościami parametrów. Plik ten należy przesłać do obiektu typu Emitter z biblioteki Pixi Particles.

- **Inkscape.**

Program do tworzenia grafiki wektorowej, znalazł zastosowanie w projekcie przy sporządzaniu grafik do gry. Projekt bomby został wykonany z zestawienia kilku mniejszych grafik, załadowanych i rozmieszczonych za pomocą współrzędnych kartezjańskich w rendererze. Program ten generuje grafikę wektorową w formacie SVG. Każda narysowana ścieżka jest zapisana jako zestaw współrzędnych, przez które jest poprowadzona. Z punktu widzenia programisty, wykonywanie operacji na grafice wektorowej w formacie SVG wygląda analogicznie do operacji przetwarzania plików XML. Możliwe jest usuwanie lub podmienianie pewnych tagów. Przykładowo, w przypadku rozbrajania bomby następuje podmiana pliku reprezentującego kabel zwarty na rozcięty. Reprezentacja w formacie SVG ułatwia wykonywanie niewielkich zmian w reprezentacji graficznej. Zastosowano to do poruszania wskazówką zegara na bombie - wydzielono ścieżkę reprezentującą wskazówkę do jednego tagu, który następnie modyfikowano. W przypadku wykorzystania grafiki rastrowej, konieczna byłaby zmiana większej części obrazu (np. całej tarczy zegarowej).

5. Model aplikacji

Niniejszy rozdział dokumentacji zawiera opis wybranych elementów systemu oraz ich architekturę i sposób działania. Przestrzeganie poniższych zasad ułatwi korzystanie z funkcjonalności systemu

5.1 Aktorzy

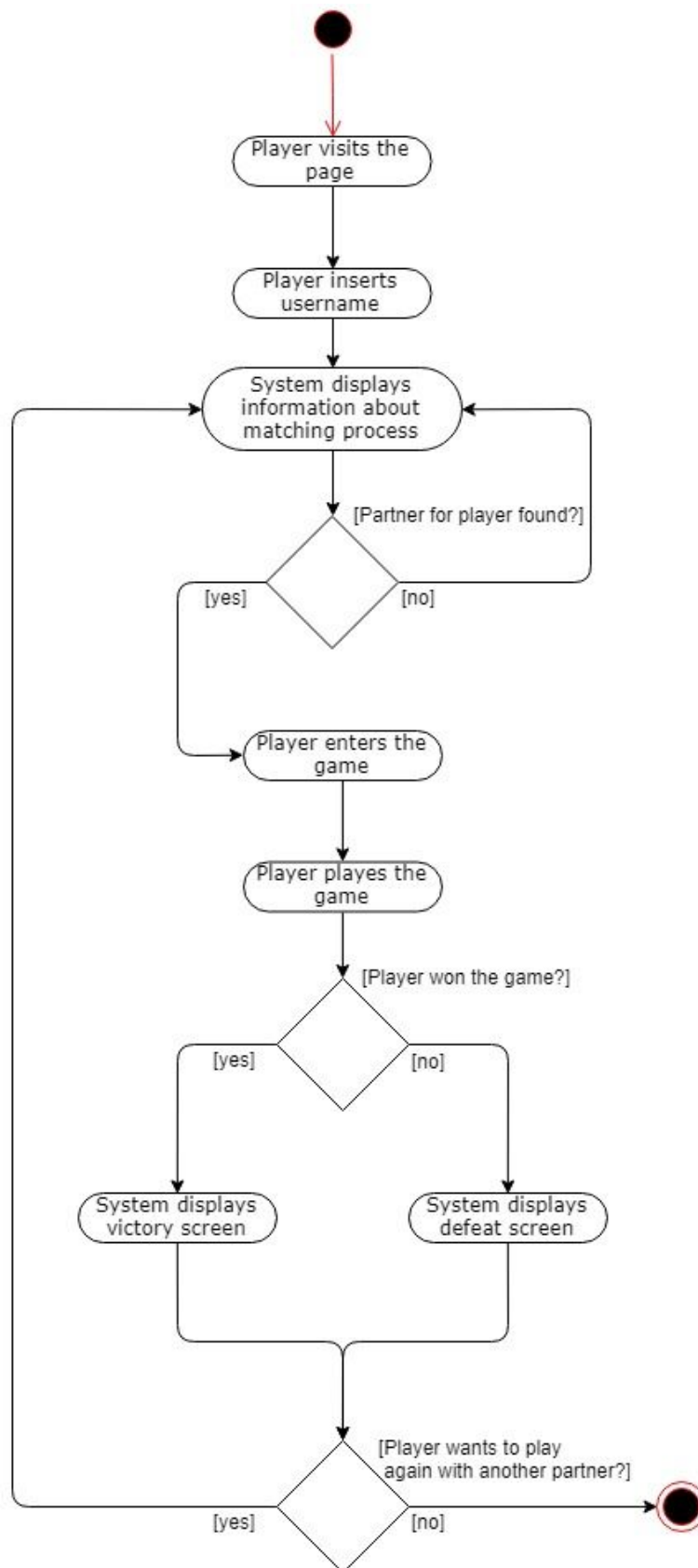
W systemie aplikacji wyróżniamy dwa typy aktorów: administrator z ogólnym dostępem do wszystkich funkcjonalności aplikacji oraz zwykły użytkownik z ograniczonym dostępem. Poniższa tabela zawiera opis poszczególnych użytkowników oraz ich uprawnienia.

Tabela 2 Zbiór aktorów

Aktor	Opis	Uprawnienia
Użytkownik	Osoba prywatna korzystająca z systemu	dołączenie do rozmów, wykonywanie połączeń, możliwość gry
Administrator	Programista, osoba zarządzająca aplikacją	uprawnienia użytkownika podniesione o modyfikacje rozgrywek, dostęp do kodu aplikacji oraz bazy danych

5.2 Diagram aktywności

Diagram czynności (ang. activity diagram) jest diagramem interakcji, który służy do modelowania dynamicznych aspektów systemu. Jego zasadniczą funkcją jest przedstawienie sekwencji kroków, które są wykonywane przez modelowany fragment systemu. Na poniższym diagramie (Rys. 1) możemy zaobserwować kroki wykonywane przez użytkownika w momencie dołączenia do systemu.

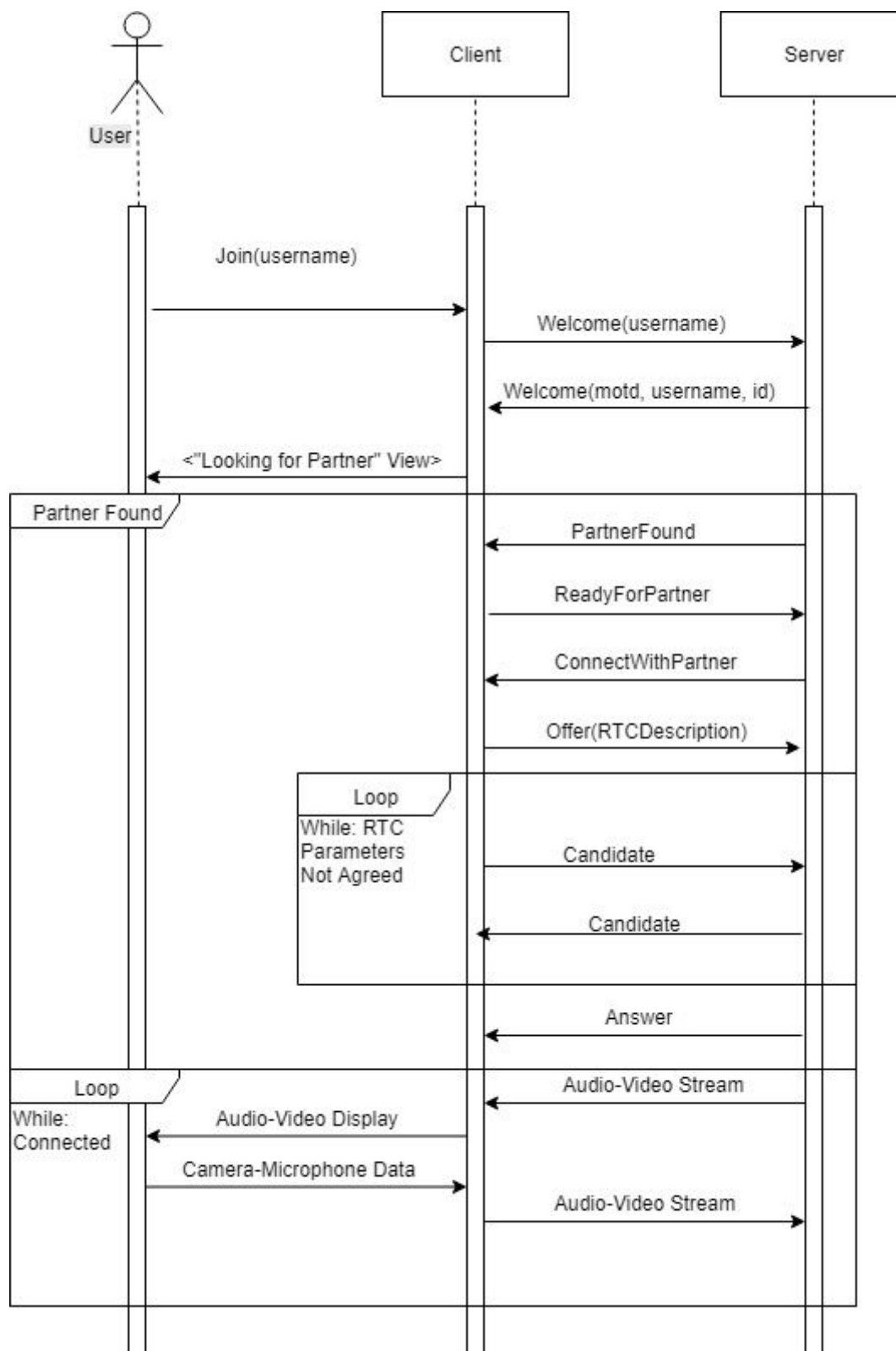


Rys. 1 Diagram aktywno ci

5.3 Diagram sekwencji

Diagram sekwencji (ang. sequence diagram) służy do prezentowania interakcji pomiędzy obiektami wraz z uwzględnieniem w czasie komunikatów, jakie są przesyłane pomiędzy nimi. Na diagramie sekwencji obiekty ułożone są wzdłuż osi X, a komunikaty przesyłane są wzdłuż osi Y. Zasadniczym zastosowaniem diagramów sekwencji jest modelowanie zachowania systemu w kontekście scenariuszy przypadków użycia. Diagramy sekwencji pozwalają uzyskać odpowiedź na pytanie, jak w czasie przebiega komunikacja między obiektami.

Na poniższym diagramie (Rys. 2) przedstawiona jest komunikacja RTC w czasie nawiązywania połączenia audio-video



Rys. 2 Diagram sekwencji

6. Problemy i ich rozwiązania

6.1 Synchronizacja ekranu gry dla obydwu graczy.

6.2 Przechwytywanie obrazu z kamery.

6.3 Zapis zmian wprowadzanych przez klienta w grze do stanu i transmisja na serwer.

7. Testy aplikacji

7.1 Testy funkcjonalności gry przeprowadzane na maszynie lokalnej - dla jednego gracza.

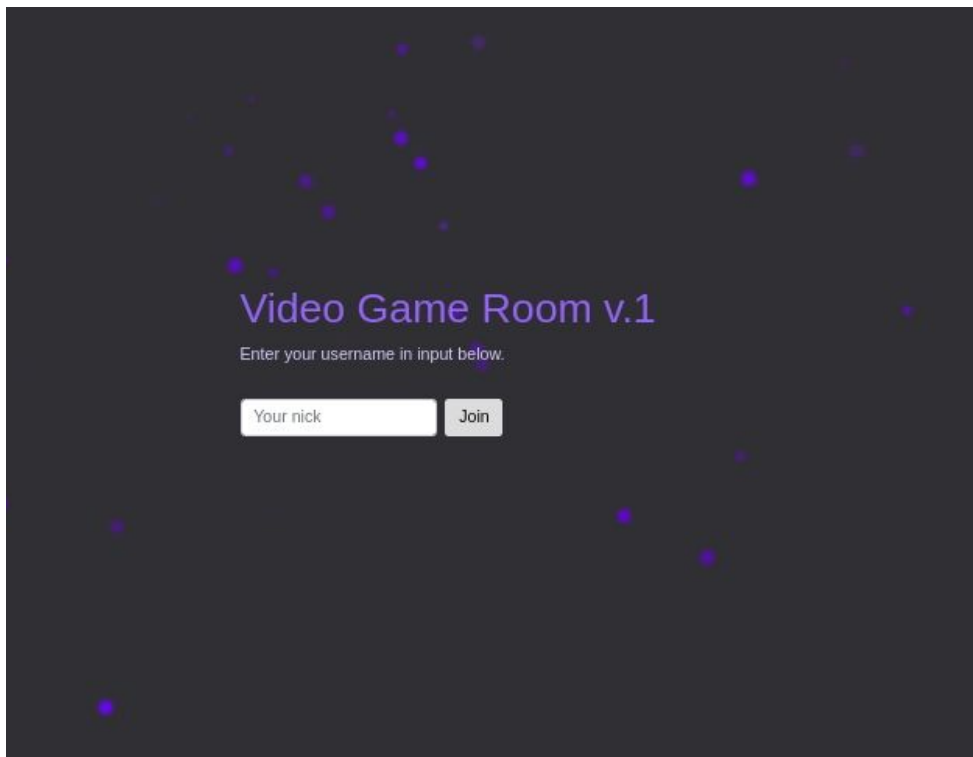
7.2 Testy funkcjonalności gry z zastosowaniem transmisji - dla 2 graczy.

7.3 Testy przesyłania obrazu oraz dźwięku w konferencji.

8. Interfejs graficzny

8.1 Warstwa graficzna aplikacji klienckiej.

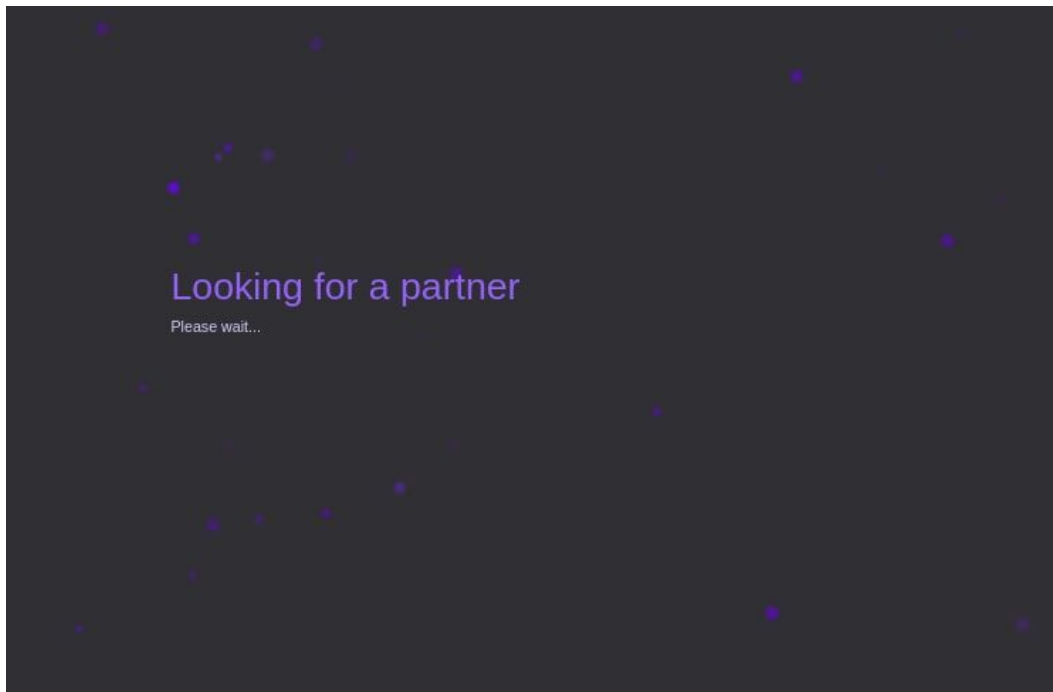
- I - ekran powitalny (Rys. 3).



Rys. 3 Ekran powitalny

Na powyższym rysunku został ukazany startowy ekran aplikacji. Użytkownik aby mieć możliwość uczestniczenia w konferencji musi podać swój identyfikator czyli nick. Podanie nazwy użytkownika spowoduje przejście do ekranu oczekiwania na partnera do rozmowy (gry) ukazanego na kolejnym rysunku.

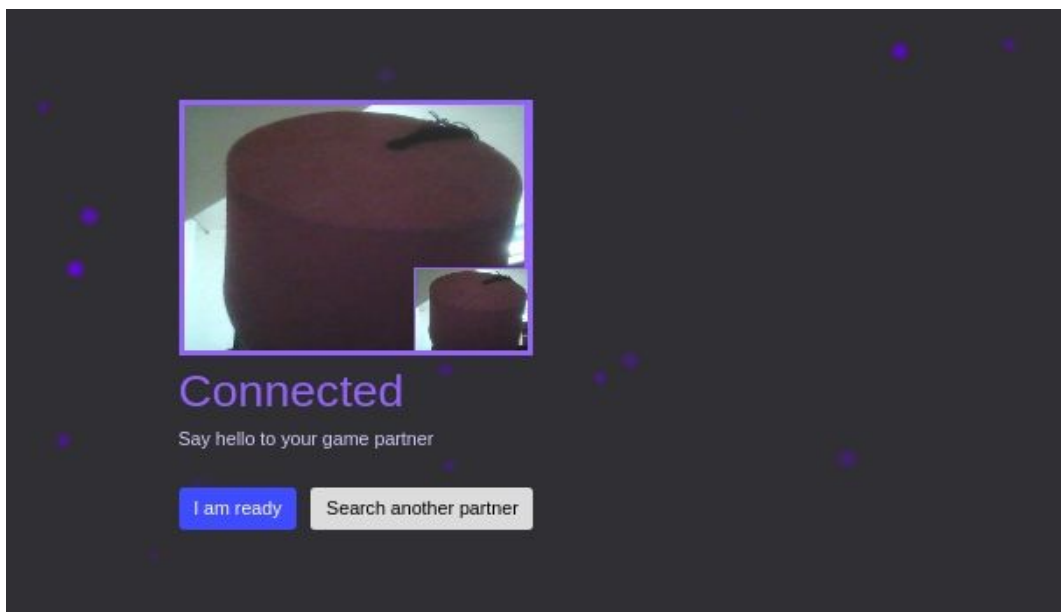
- **II - ekran oczekiwania (Rys. 4).**



Rys. 4 Ekran oczekiwania

Ekran oczekiwania widoczny dla zalogowanych użytkowników oczekujących na partnera do gry. Użytkownik będący w tym stanie aplikacji oczekują na partnera przydzielonego przez system do gry oraz rozmowy. System przydziela użytkowników losowo tzn. spośród osób będących zalogowanych w systemie wybiera dwóch użytkowników łączą ich w parę i przydziela pokój do rozmowy

- **III - lobby gry (Rys. 5).**



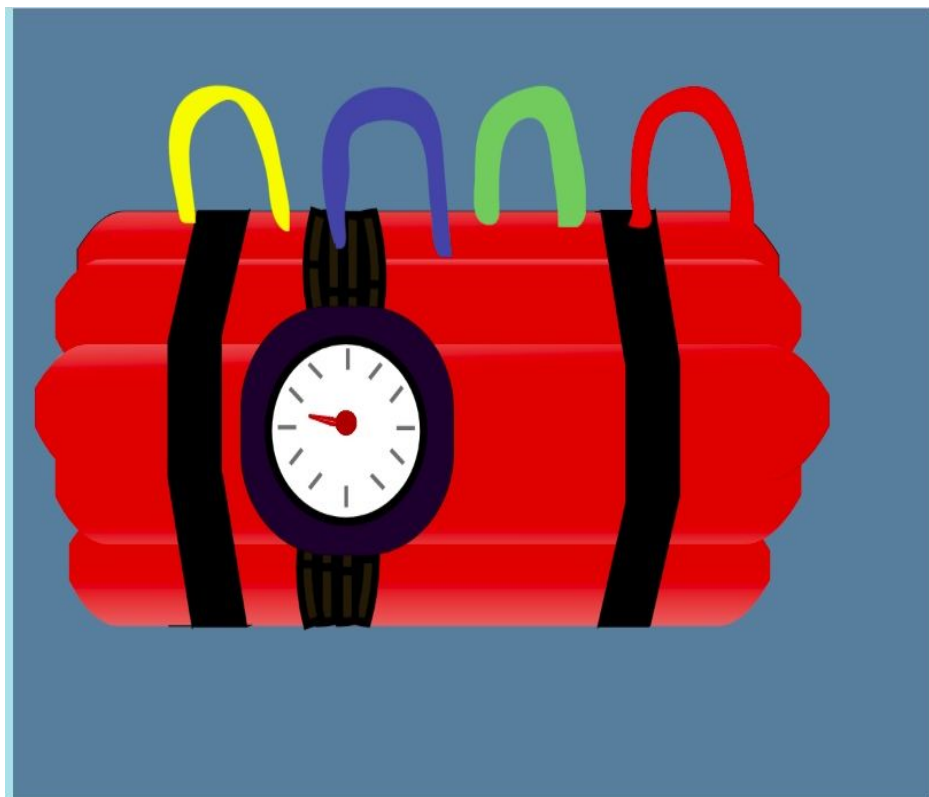
Rys. 5 Lobby gry

Po dołączeniu do pokoju użytkownik ma możliwość przetestowania swojej kamery, komunikacji z użytkownikiem przeciwnym oraz wykonania testu mikrofonu. Gdy obaj gracze upewnią się, że wszystko jest mogą przystąpić do rozgrywki prowadzonej między dwoma losowo wybranymi osobami. Również każdy z graczy ma możliwość zmiany partnera do rozmowy gdy ten mu nie odpowiada.

- **Ekran I (Rys. 6) - główny ekran gry, wyświetlany po uruchomieniu gry.**

Gra polegająca na rozbrajaniu bomby została wykonana z użyciem narzędzia PixiJS. Jest to renderer, który umożliwia wyświetlanie interaktywnych obrazów w czasie rzeczywistym. Obraz bomby jest wykonany w technice grafiki wektorowej. Składa się z kilkunastu oddzielnych plików w formacie *Scalable Vector Graphics*. W rendererze pliki są wczytane i rozmieszczone w odpowiednich miejscach. Takie zastosowanie grafiki wektorowej ułatwia obsługę interaktywnej funkcjonalności - np. umożliwia podmianę pliku z kabelkami bomby z zawartych na rozwarte, w przypadku gdy użytkownik postanowi rozciąć kabel kliknięciem.

Obrót wskazówek zegara został ustawiony poprzez funkcję odmierzenia czasu (atrybut *ticker* dla obiektu *PIXI.Application*). Każdy obiekt posiada pole *rotation*. W funkcji odmierzenia czasu, *rotation* dla obiektu Zegar jest nieznacznie zwiększone wraz z określonym upływem czasu.

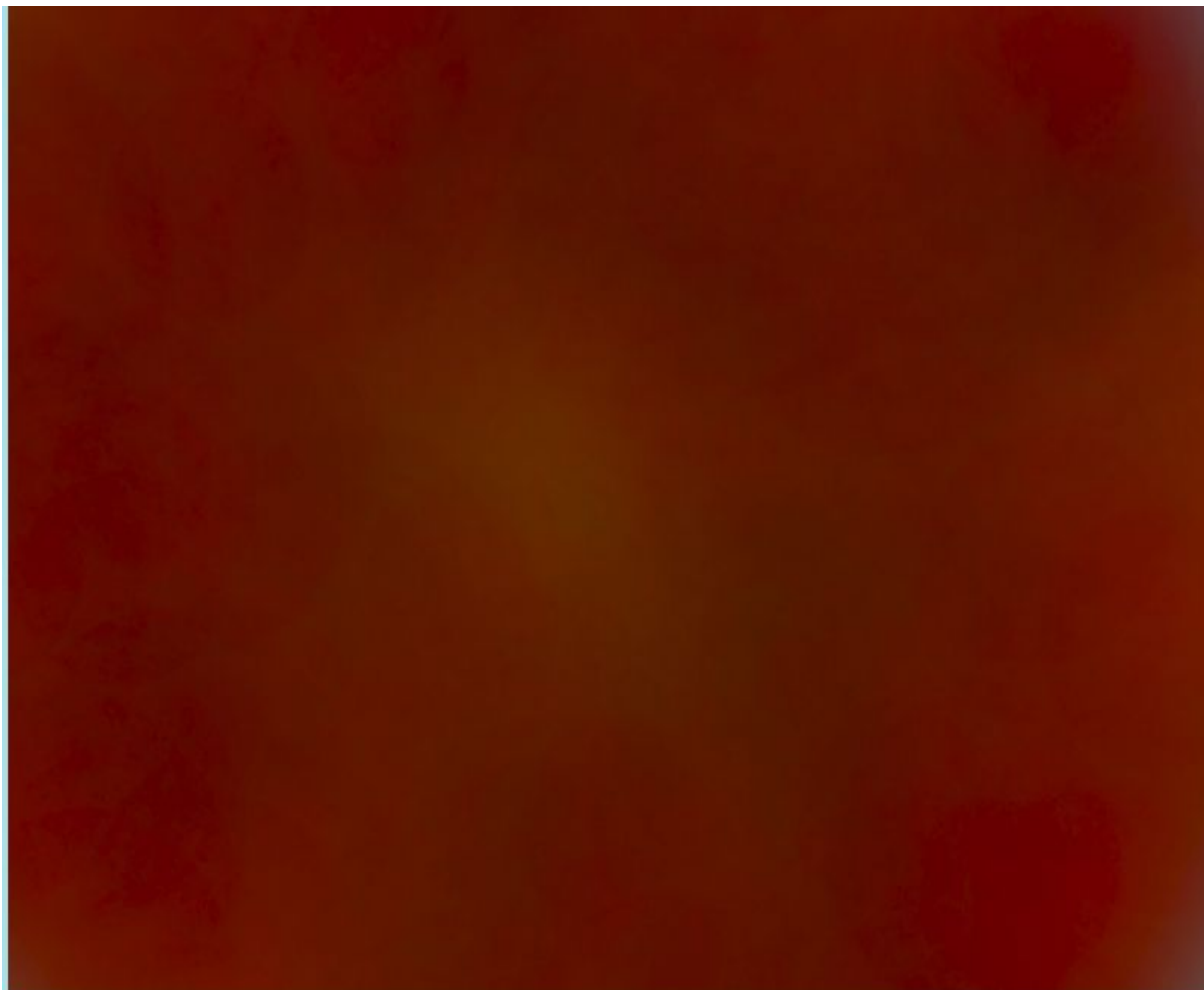


Rys. 6 Ekran z widokiem na bombę renderowany z użyciem PixiJS. Wyświetlany po dołączeniu użytkownika do gry.

W przypadku gdy gracz zdecyduje się rozbroić bombę, możliwe są dwa warianty:

- **Ekran II (Rys. 7) - renderowania efektu eksplozji.**

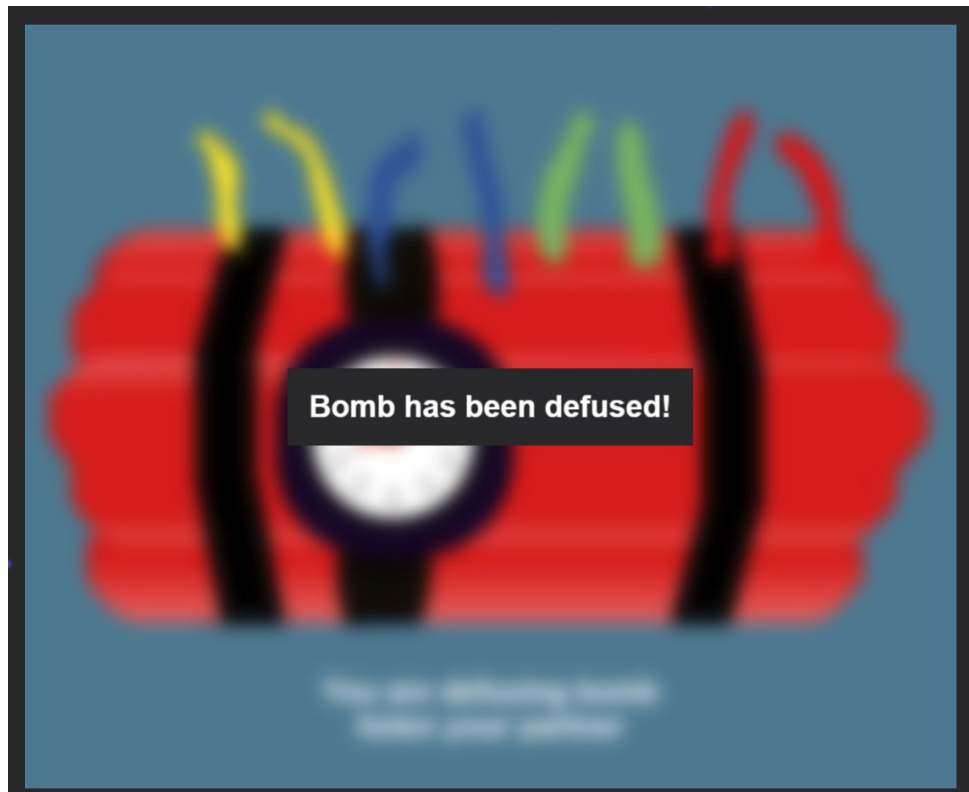
Wybór niewłaściwego kabla spowoduje wygenerowanie efektu eksplozji z użyciem biblioteki *Pixi Particles*, która pokryje cały ekran. Efekty do wyemitowania zostały sporządzone z użyciem edytora *Pixi Particles Editor* oraz zapisane w pliku JSON.



Rys. 7 Widok ekranu wyświetlany użytkownikom po nieudanym rozbrojeniu bomby.

- **Ekran III (Rys. 8) - rozbrojenie bomby.**

Jeśli graczowi uda się rozbroić bombę, obrót wskazówek zegara zostanie zatrzymany. Gracz zostanie poinformowany o powodzeniu wykonanej akcji odpowiednim komentarzem. Podmieniony zostanie plik w formacie SVG reprezentujący kabel. Niemal cały obraz zostanie rozmazany efektem *Blur* - pochodzi on z biblioteki filtrów PIXI.filters. Jedynym nierozmazanym obiektem pozostanie zegar.



Rys. 8 Widok ekranu wyświetlany u użytkownikowi w przypadku udanego rozbrojenia bomby. Obraz zostaje rozmyty efektem *Blur*, a zegar jest zatrzymywany.

9. Instrukcja użytkowania

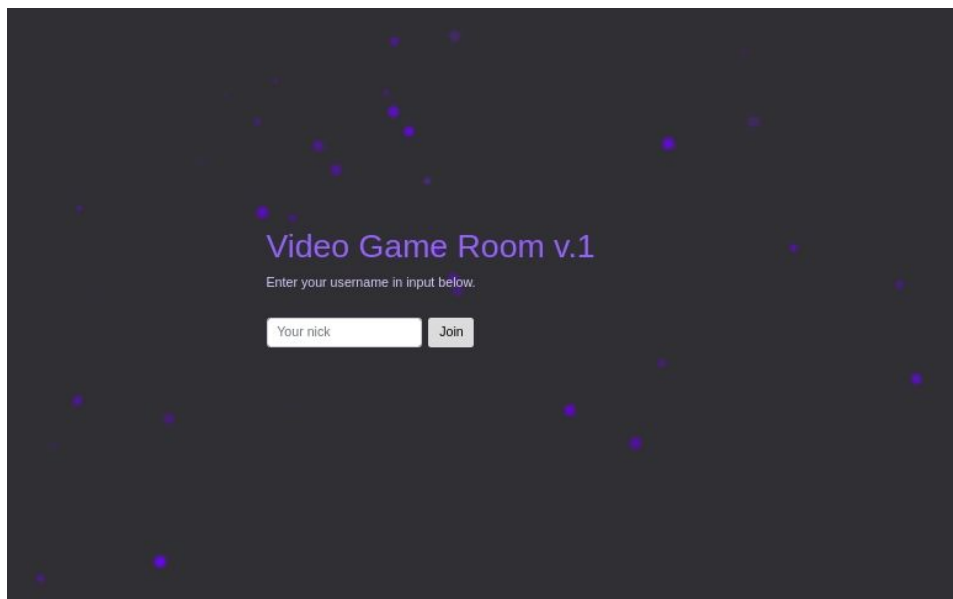
Niniejszy rozdział dokumentacji zawiera opis elementów dostępnych dla użytkownika oraz sposób poruszania się pomiędzy kolejnymi ekranami aplikacji. Aplikacja przejrzyste przechodzi do następnych etapów zabawy i rozmowy pomiędzy użytkownikami, przełączając się płynnie pomiędzy kolejnymi ekranami.

9.1 Uruchomienie aplikacji

Do uruchomienia aplikacji potrzebna nam będzie dowolna przeglądarka internetowa. Po uruchomieniu przeglądarki należy przejść na adres **ADRES** a następnie postępować zgodnie z wyświetlanymi ekranami i odpowiedziami. Aby móc w pełni korzystać z aplikacji nasz komputer powinien być wyposażony w wejściowe urządzenie audio - mikrofon oraz wyjściowe urządzenie audio - głośniki/słuchawki i dodatkowo kamerę internetową

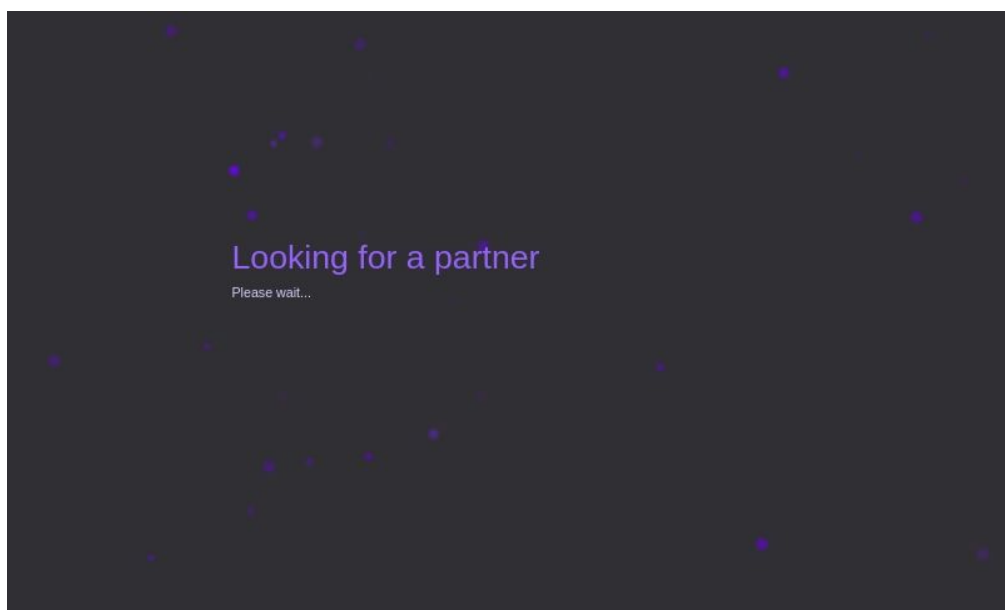
9.2 Obsługa aplikacji

Po przejściu na stronę aplikacji ukaże nam się okno (Rys. 9) umożliwiające nam wpisanie naszego identyfikatora i przejście do kolejnego etapu.



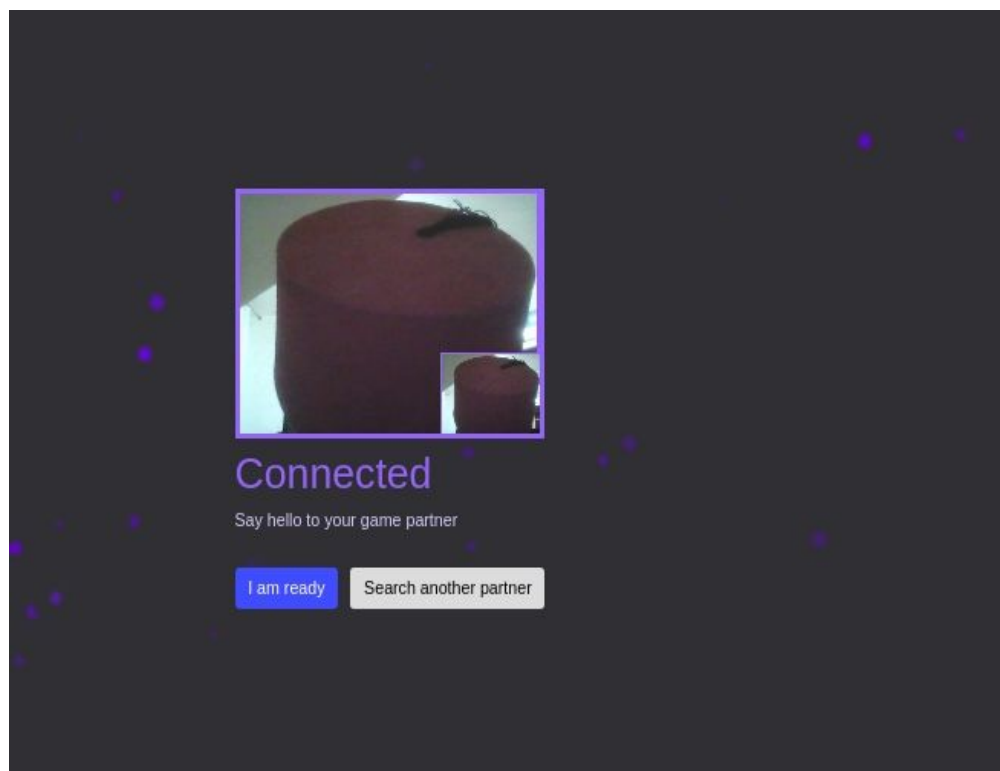
Rys. 9 Ekran powitalny

Gdy już wpiszę nazwę identyfikującą naszą osobę i klikniemy przycisk *Join* zostaniemy przeniesieni do ekranu poczekalni (Rys. 10) w której będziemy oczekiwać na przydzielenie nam partnera do gry i rozmów.



Rys. 10 Ekran poczekalni

Ekran poczekalni. W tym momencie serwer z dostępnych użytkowników w systemie przydziela nam losowo osobę do gry i rozmowy. Operacja może potrwać chwilę gdy na serwerze nie ma dostępnych osób, jednak jak jest więcej niż jedna oprócz nas operacja jest wykonywana natychmiast. Gdy zostanie znaleziony partner zostaniemy przeniesieni do pokoju rozgrywki (Rys. 11).



Rys. 11 Lobby gry

W pokoju rozgrywki mamy możliwość przetestowania połączenia jak i rozmowy audio-wideo. Użytkownik może na początek porozmawiać sobie ze swoim partnerem i się zapoznać, ustalić taktykę i strategię gry oraz komunikaty jakimi będą się porozumiewać. Gdy użytkownik akceptuje partnera klika przycisk że jest gotowy. Gdy obaj użytkownicy potwierdzają chęć dalszej zabawy przechodzą do ekranu z grą. Natomiast gdy użytkownikowi nie pasuje partner do gry ma możliwość porzucenia aktualnego gracza i wylosowania innego partnera.

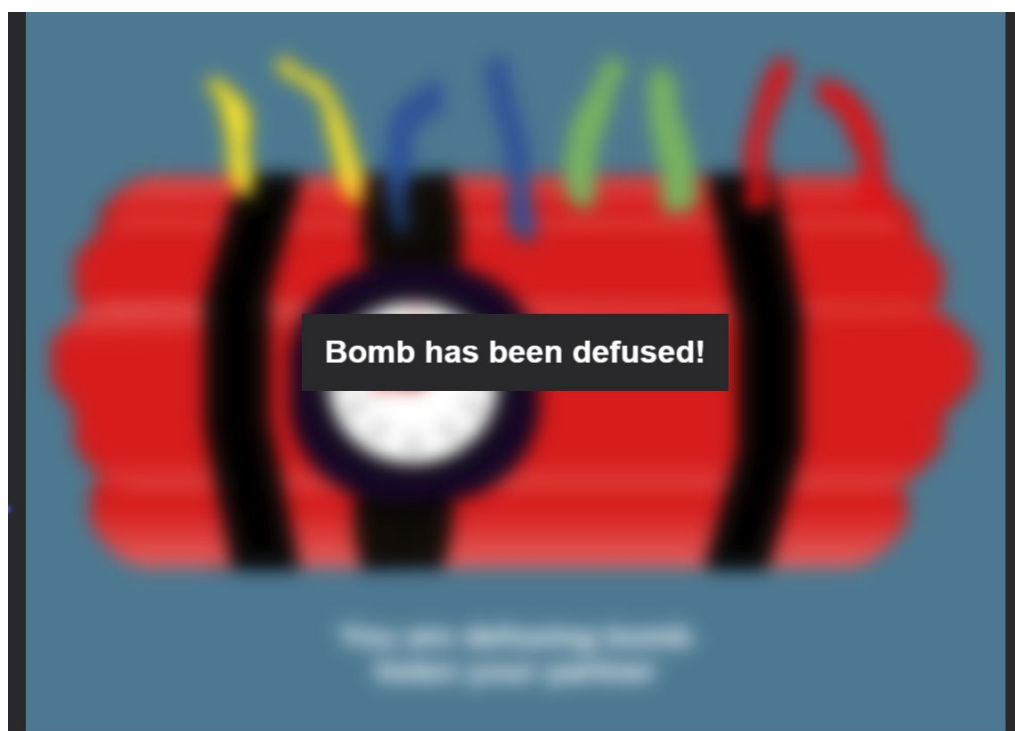


Rys. 12 Widok gracza rozbrajającego bombę

Użytkownik zostaje poinformowany o roli jaką pełni w grze. Może być "saperem" (Rys. 12) lub "trenerem" (Rys. 13). W przypadku, gdy zostanie mu przydzielona rola trenera, jego zadaniem jest przekazać partnerowi sekwencję kabli do rozcięcia. Saper powinien potrafić porozumieć się z rozmówcą i dobrze zrozumieć co ma do przekazania.

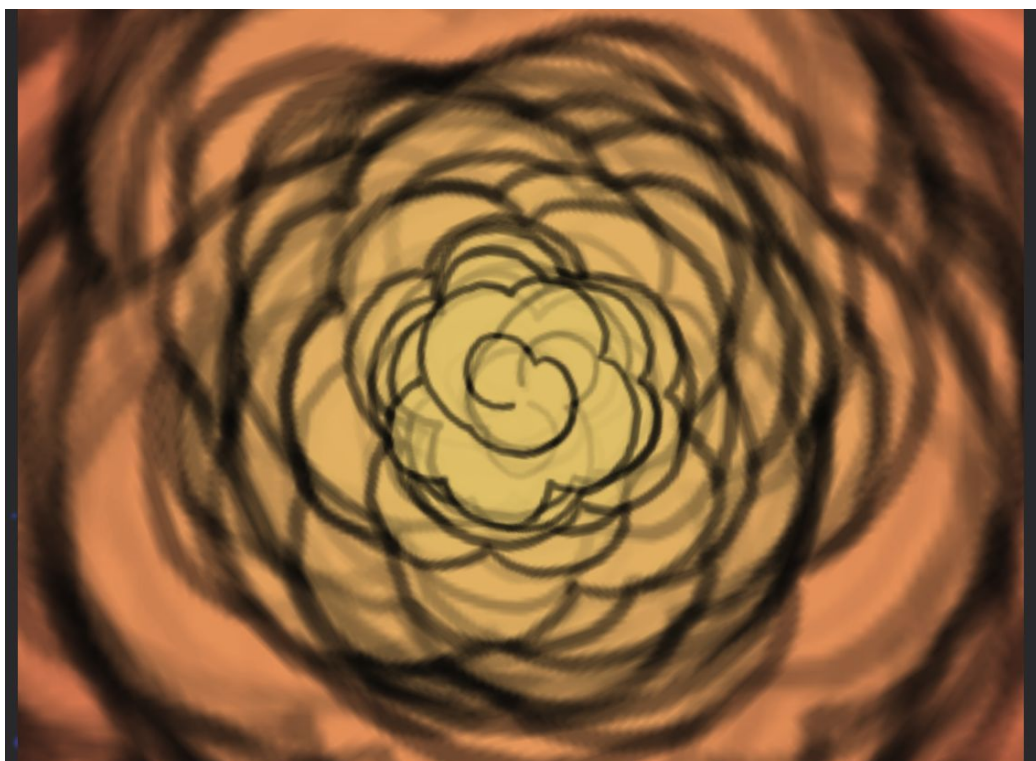


Rys. 13 Widok gracza przekazuj cego instrukcje



Rys. 14 Rozbrojona bomba

W nowszej wersji programu, po rozbrojeniu bomby użytkownik otrzymuje komunikat na środku ekranu (Rys. 14). Została także zmieniona grafika emitowana przy wybuchu. Nowy *asset* jest komiksowy, co podkreśla lekki charakter gry oraz zmniejsza ryzyko wystąpienia epilepsji u graczy znajdujących się w grupie ryzyka.



Rys. 15 Wybuch bomby

10. Dalszy rozwój

W wersji projektu zaprezentowanej na ostatnich zajęciach projektowych udało się uzyskać pełną zadowalającą funkcjonalność komunikatora z przesyłaniem obrazu oraz dźwięku. System umożliwia parowanie użytkowników zgłaszających chęć do gry. Poprawnie funkcjonuje także renderowanie gry w przeglądarce internetowej składającej się z uprzednio przygotowanych grafik oraz kodu Źródłowego w JavaScriptcie z zastosowaniem frameworka PIXIJS. Istotnym zagadnieniem, które wymagało dużego nakładu pracy, było zsynchronizowanie wyświetlanego u obydwu użytkowników obrazu. Informacje o stanie gry są przesyłane do serwera - jest to np. informacja o rozciętych kablach. Na podstawie przesłanej informacji renderowana jest grafika.

Po przygotowaniu takiego zaplecza technologicznego, oraz zapoznaniu się z programowaniem przy użyciu narzędzi wykorzystywanych w opisywanym projekcie, naturalną drogą rozwoju systemu wydaje się rozbudowanie jej do pełnoprawnej platformy z minigramami stawiającymi na interakcję między graczami.

Duży potencjał ma dodanie do aplikacji gry platformowej, w której gracze musieliby współpracować, aby przemierzać kolejne etapy odwiedzanego świata. Gry platformowe nie straciły popularności, dodatkowo bazują na nostalgii graczy, którzy kojarzą je ze wcześniejszymi etapami swojego życia. Możliwość współpracy dodała do tego gatunku elementy, które są wśród graczy wysoce pożądane. O popularności "platformówek" świadczy sukces, który odniosły takie gry jak *Magicka* czy *Trine*. Opublikowanie gry wzorowanej na którymś z tych tytułów dostarczyłoby szandarowy produkt dla aplikacji.

Innymi grami, które można by dodać do projektu, są klasyczne gry towarzyskie, np. kalambury. Ze względu na kooperacyjny charakter platformy, nie są przewidziane implementacje klasycznych gier planszowych, ponieważ większość z nich oparta jest na rywalizacji między graczami.

11. Podsumowanie

Prace nad aplikacją przebiegały harmonijnie. Podział obowiązków w grupie był czytelny i każdy jej członek wiedział, na jakich zagadnieniach ma się w danym momencie skupić. Zespół potrafił się zorganizować i skupić na pracy nad projektem, dzięki czemu po większości iteracji był w stanie zaprezentować wersję produktu z wprowadzoną nową funkcjonalnością lub znaczącymi zmianami wizualnymi. Udało się zrealizować założenia projektowe.

12. Bibliografia

1. Learning Pixi
<https://github.com/kittykatattack/learningPixi>, dostęp 23.06.2018
2. PixiJS — The HTML5 Creation Engine,
<https://github.com/pixijs/pixi.js> dostęp 23.06.2018
3. Pixi Particles,
<https://github.com/pixijs/pixi-particles> dostęp 23.06.2018
4. Forum internetowe dla programistów,
<https://stackoverflow.com> dostęp 23.06.2018
5. Node.js
<https://nodejs.org/en/> dostęp 23.06.2018
6. Bootstrap
<https://getbootstrap.com/> dostęp 23.06.2018
7. W3School
<https://www.w3schools.com> dostęp 23.06.2018