

Akademia Kultury Społecznej i Medialnej

WYDZIAŁ NAUK SPOŁECZNYCH
I KOMUNIKACJI MULTIMEDIALNEJ

przedmiot

**Projekt z metod programowania
Laboratorium**



Projekt grupowy

Dokumentacja projektu „Quack Hunt”

prowadzący

dr inż. Sławomir Jeżewski

TORUŃ, 29 października 2023

Spis treści

1.	Wstęp	3
2.	Wymagania	3
3.	Wymagania dotyczące pistoletu	3
4.	Dokumentacja Software.....	4
4.1.	Wprowadzenie	4
4.1.1.	Cel.....	4
4.1.2.	Zakres	4
4.1.3.	Docelowi odbiorcy.....	4
4.2.	Struktura Gry.....	4
4.2.1.	Klasa „GameCore”	4
4.2.2.	Klasa „BirdsObj”.....	4
4.2.3.	Klasa BirdsFeather	4
4.2.4.	Klasa GameBirds	5
4.3.	Stałe i Konfiguracja	5
4.3.1.	Stałe	5
4.3.2.	Konfiguracja	5
4.4.	Interfejs użytkownika	5
4.5.	Zasoby i skrypty	6
4.5.1.	Skrypty	6
4.5.2.	Struktura kodu HTML.....	6
4.5.3.	Struktura kodu w JavaScript:	8
4.6.	Grafika	10
5.	Model pistoletu.....	11
6.	Architektura systemowa.....	13
7.	Wymagania dotyczące produktu	15

1. Wstęp

- Quack Hunt jest grą elektroniczną integrującą ruchowość, która wymaga sprawności fizycznej od gracza.
- Posługujemy się pistoletem i ekranem do obsługi gry.

2. Wymagania

- 2.1. Quack Hunt wymaga telewizora o wielkości 800x600 cm do gry.
- 2.2. Quack Hunt wymaga specjalizowanego pistoletu do gry.
- 2.3. Quack Hunt wymaga komputera z interfejsem WiFi.
- 2.4. Quack Hunt wymaga pomieszczenia o wielkości 5x5 m.

3. Wymagania dotyczące pistoletu

- 3.1. Pistolet musi mieć możliwość druku na drukarce 3D.
- 3.2. Projekt mechaniczny pistoletu ma być opracowany w Adobe Inventor.
- 3.3. Mechaniczny projekt pistoletu ma być udostępniony społeczności internetowej.
- 3.4. Projekt elektryczny pistoletu ma być opracowany w AutoCAD.
- 3.5. Projekt elektryczny pistoletu ma być udostępniony społeczności internetowej.
- 3.6. Założenia projektu elektrycznego
 - 3.6.1. Pistolet ma być zasilany na baterie
 - 3.6.2. Pistolet ma być zasilany baterią o rozmiarze 18650
 - 3.6.3. Pistolet musi mieć przełącznik włączający i wyłączający elektronikę
 - 3.6.4. Pistolet komunikuje się z resztą systemu za pomocą Wi-Fi (udostępnione przez komputer)
 - 3.6.5. Konfiguracja pistoletu dostępna za pośrednictwem punktu dostępu pistoletu.
 - 3.6.6. Kontroler pistoletu ze spustem i ekranem LCD do wyświetlania adresu IP pistoletu i wyniku.
- 3.7. Założenia funkcjonalne
 - 3.7.1. Jak działa w obrębie systemu
 - 3.7.2. Jakie funkcje realizuje

4. Dokumentacja Software

4.1. Wprowadzenie

Dokument projektowy gry "Quack Hunt" to formalny dokument, który pełni kluczową rolę w procesie projektowania i implementacji gry komputerowej. Zawiera on szczegółowy opis struktury gry, jej funkcji oraz architektury kodu, mając na celu dostarczenie kompleksowego zrozumienia dla zespołu programistycznego.

4.1.1. Cel

Głównym celem dokumentu jest dostarczenie spójnego planu dla programistów, który określa strukturę modułową, funkcje oraz sposób interakcji między poszczególnymi elementami gry. Ponadto, dokument ma na celu ułatwienie zrozumienia koncepcji gry "Quack Hunt" i jej wizji, umożliwiając jednocześnie zgodność z założeniami projektowymi.

4.1.2. Zakres

Gra "Quack Hunt" to strzelanka, w której gracz ma za zadanie strzelać do obiektów reprezentujących ptaki. Projekt obejmuje różne poziomy trudności, a osiągnięcie celu zależy od precyzyjnych strzałów gracza. Zakres dokumentu obejmuje główne klasy i moduły związane z logiką gry, a także stałe i konfiguracje.

4.1.3. Docelowi odbiorcy

Dokument jest dedykowany programistom pracującym nad implementacją gry "Quack Hunt". Stanowi on kluczowe źródło informacji, umożliwiające zrozumienie założeń projektu, co przyczynia się do efektywnej pracy zespołu programistycznego.

4.2. Struktura Gry

4.2.1. Klasa „*GameCore*”

Klasa *GameCore* pełni centralną rolę w grze, przechowując statystyki i odpowiadając za aktualizację stanu gry w każdej klatce. Dokładne omówienie interakcji tej klasy z innymi modułami jest kluczowe dla zrozumienia logiki gry.

4.2.2. Klasa „*BirdsObj*”

Klasa *BirdsObj* reprezentuje obiekty ptaków w grze. Szczegółowy opis struktury tej klasy obejmuje informacje o położeniu, prędkości i zachowaniu ptaków. Wskazane jest także przedstawienie sposobu obsługi trafień.

4.2.3. Klasa *BirdsFeather*

Klasa *BirdsFeather* odpowiada za reprezentację piór, które pojawiają się po trafieniu ptaka. Opisuje ona sposób, w jaki pióra poruszają się w grze i jak są renderowane.

4.2.4. Klasa GameBirds

Klasa `GameBirds` rozszerza funkcjonalność klasy `GameCore` o elementy związane z ptakami i piórami. Dokładny opis roli tej klasy w strukturze gry oraz jej interakcji z innymi modułami jest niezbędny dla zrozumienia całości.

4.3. Stałe i Konfiguracja

4.3.1. Stałe

Difficulties – Zestaw poziomów trudności gry, definiujących parametry takie jak szybkość i rozmiar obiektów. Przykładowe scenariusze dla różnych poziomów trudności mogą być dodatkowo opisane.

Colors – Kody kolorów używane w grze, z ich znaczeniem i zastosowaniem. Szczegóły dotyczące estetyki i atmosfery gry.

BirdsVariety – Rodzaje ptaków dostępne w grze, z opisem unikalnych cech każdego rodzaju.

4.3.2. Konfiguracja

config.dark – Flaga określająca tryb ciemny w grze. Wpływ na doświadczenie wizualne gracza.

config.difficulty – Poziom trudności gry, z zaleceniami dotyczącymi zbalansowania rozgrywki. Przykładowe scenariusze dla różnych poziomów trudności.

config.name – Nazwa gracza, z informacjami o tym, jak będzie wykorzystywana w grze.

config.ip – Adres IP gracza, jeśli relewantny w kontekście gry wieloosobowej.

config.treshold – Próg wartości dla detekcji strzału, z opisem, jak wpływa na mechanikę gry.

4.4. Interfejs użytkownika

Gra "Quack Hunt" oferuje użytkownikowi prosty interfejs składający się z ekranu gry i panelu konfiguracyjnego dostępnego dzięki bibliotece Tweakpane. Dokładny opis poszczególnych elementów interfejsu oraz ich funkcji przyczyni się do spójnego zrozumienia doświadczenia gracza.

4.5. Zasoby i skrypty

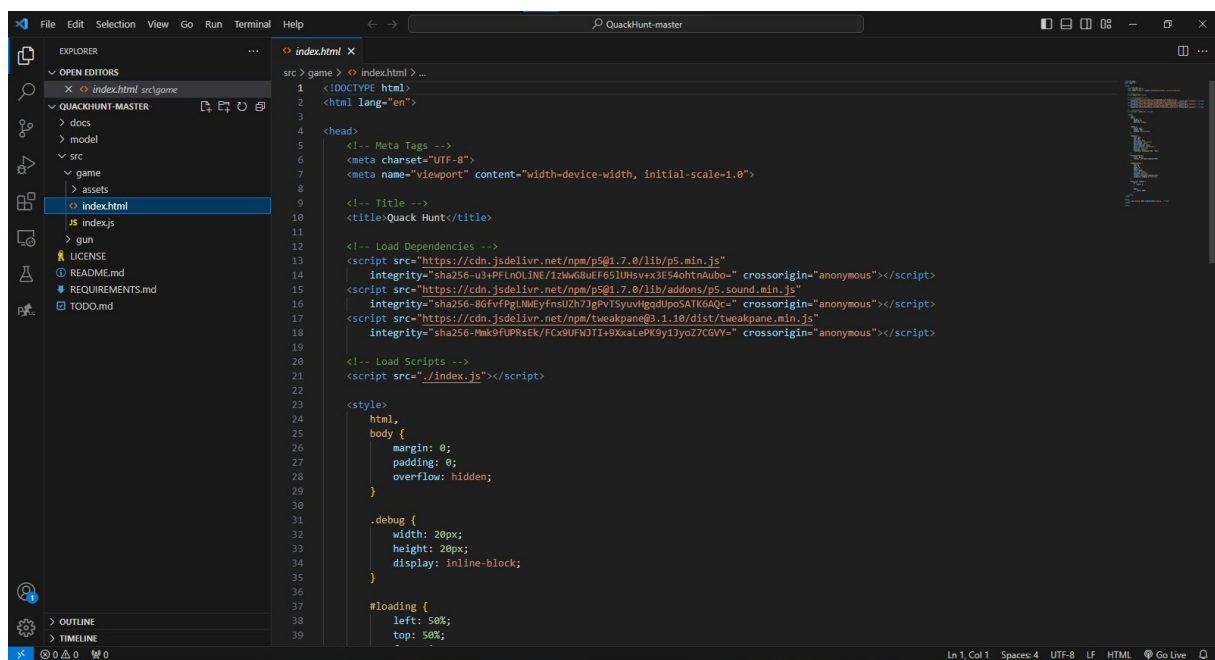
Gra korzysta z różnych zasobów, takich jak dźwięki strzałów i trafień, obrazy tła oraz animacje ptaków. Szczegółowy opis każdego zasobu, wraz z informacjami o sposobie ich integracji z grą, jest kluczowy dla zapewnienia odpowiedniego działania gry.

4.5.1. Skrypty

Skrypty definiują logikę gry oraz obsługę interakcji z graczem. Dokładny opis struktury kodu oraz komunikacji między skryptami pozwoli zrozumieć mechanizmy rządzące rozgrywką.

Dokument ten stanowi fundament projektu gry "Quack Hunt" i będzie regularnie aktualizowany w miarę postępu prac, dostarczając zespołowi programistycznemu klarownego i kompleksowego przewodnika.

4.5.2. Struktura kodu HTML



Rys. 1 Program aplikacji webowej napisany w języku HTML

```
<!DOCTYPE html>
```

Deklaracja typu dokumentu. Informuje przeglądarkę, że plik jest dokumentem HTML5.

```
<html lang="en">
```

Otwarcie tagu HTML z określeniem języka dokumentu (angielski).

<head>

Początek sekcji nagłówkowej dokumentu, która zawiera metadane, style, skrypty i inne informacje nieprezentacyjne.

charset="UTF-8"

Określa kodowanie znaków jako *UTF-8*, co pozwala obsługiwać znaki z różnych języków.

viewport

Określa widok dla urządzeń mobilnych, co pozwala na responsywność strony.

<title>

Tytuł strony, który będzie wyświetlany na karcie przeglądarki.

Load Dependencies

Sekcja, w której ładowane są zależności, takie jak biblioteka *p5.js* do tworzenia grafiki i dźwięku, oraz *Tweakpane* do tworzenia interfejsu konfiguracyjnego.

integrity i crossorigin

Zabezpieczenia, które sprawdzają, czy pliki są niezmienione i pochodzą z bezpiecznego źródła.

Load Scripts

Sekcja, w której ładowany jest skrypt JavaScript z pliku *index.js*.

style

Sekcja zawierająca definicje stylów CSS dla elementów na stronie.

<body>

Początek sekcji treści strony.

<div id="loading" data-loading-text="Loading..."></div>

Element *<div>* z identyfikatorem „*loading*”, który będzie używany do wyświetlania komunikatu ładowania. Dane atrybutu „*data-loading-text*” zawierają tekst wyświetlany podczas ładowania.

</body>

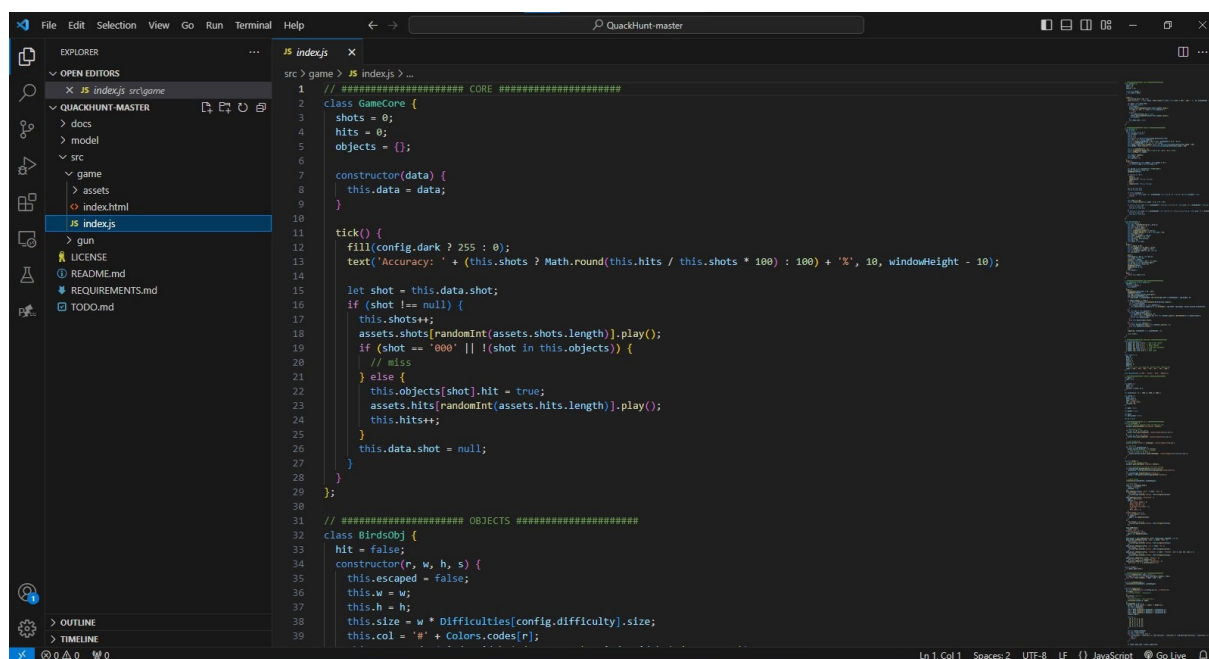
Zamknięcie sekcji treści strony.

</html>

Zamknięcie tagu HTML, oznaczające zakończenie dokumentu.

Kod zawiera również style CSS dla elementów, takie jak animacja ładowania. Kluczowym elementem jest tu wykorzystanie bibliotek *p5.js* i *Tweakpane*, które wprowadzają funkcje związane z grafiką, dźwiękiem i interfejsem użytkownika do gry "Quack Hunt".

4.5.3. Struktura kodu w JavaScript:



Rys. 2 Program aplikacji webowej napisany w języku JavaScript

Klasa *GameCore*

- Klasa ta stanowi rdzeń gry i zawiera informacje na temat strzałów, trafień oraz obiektów w grze.
- Metoda „tick” odpowiada za aktualizację stanu gry, rysowanie informacji o trafieniach i przetwarzanie strzałów.

Klasa *BirdsObj*

- Klasa reprezentuje obiekt ptaki w grze.
- Konstruktor inicjalizuje właściwości ptaki, takie jak położenie, prędkość, kierunek poruszania się, itp.
- Metoda „draw” odpowiada za rysowanie obiektu ptaki na ekranie.

Klasa *BirdsFeather*

- Klasa reprezentuje pióra, które pojawiają się po trafieniu ptaka.
- Konstruktor inicjalizuje właściwości pióra, takie jak położenie, kierunek poruszania się, itp.

- Metoda „draw” odpowiada za rysowanie pióra na ekranie, uwzględniając animację opadania.
- Metoda „end” zwraca informację, czy pióro zakończyło swoją animację.

Klasa *GameBirds*

- Klasa ta rozszerza klasę *GameCore* i dodaje funkcjonalność związana z elementami gry, takimi jak ptaki i pióra.
- Metoda „tick” odpowiada za aktualizację stanu gry, rysowanie obiektów, zarządzanie ptakami i piórami.

Stałe i konfiguracja

- Zdefiniowane są stałe, takie jak poziomy trudności gry („Difficulties”), kolory („Colors”), różne rodzaje ptaków („BirdsVariety”).
- Ustawienia konfiguracyjne dla gry, takie jak tryb ciemny, poziom trudności, nazwa gracza, adres IP, próg wartości dla detekcji strzału („config”).

Globalne zmienne

- Zdefiniowane są globalne zmienne, takie jak „data” (związane z danymi gry), „assets” (związane z zasobami gry), „calibration” (dane kalibracyjne), „config” (ustawienia konfiguracyjne gry), „game” (obiekt gry), „packet” (pakiet danych), „pane” (panel konfiguracyjny), „pane_player” (panel konfiguracyjny dla gracza).

Funkcje p5.js

Funkcje takie jak „preload”, „setup”, „draw”, „windowResized” są częścią biblioteki *p5.js* i są używane do ładowania zasobów, konfiguracji początkowej, rysowania klatek i obsługi zdarzeń okna.

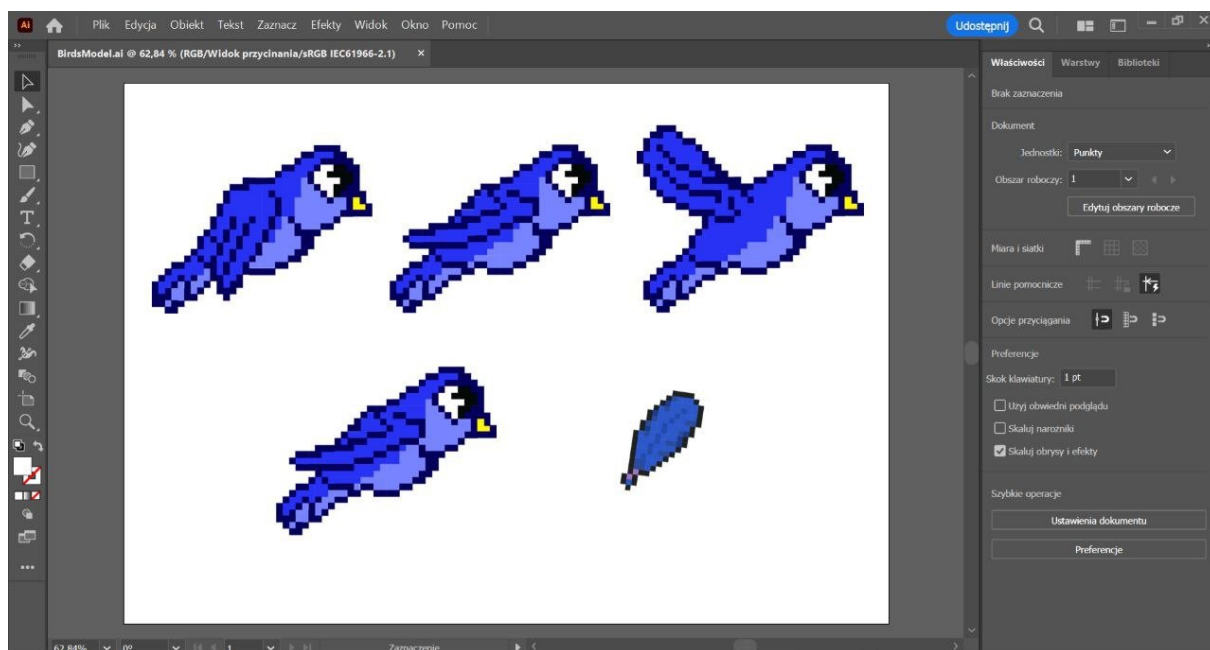
MISC (Różne)

- Funkcje pomocnicze takie jak „randomInt”, które generuje losową liczbę całkowitą, są używane w różnych miejscach w kodzie.

4.6. Grafika

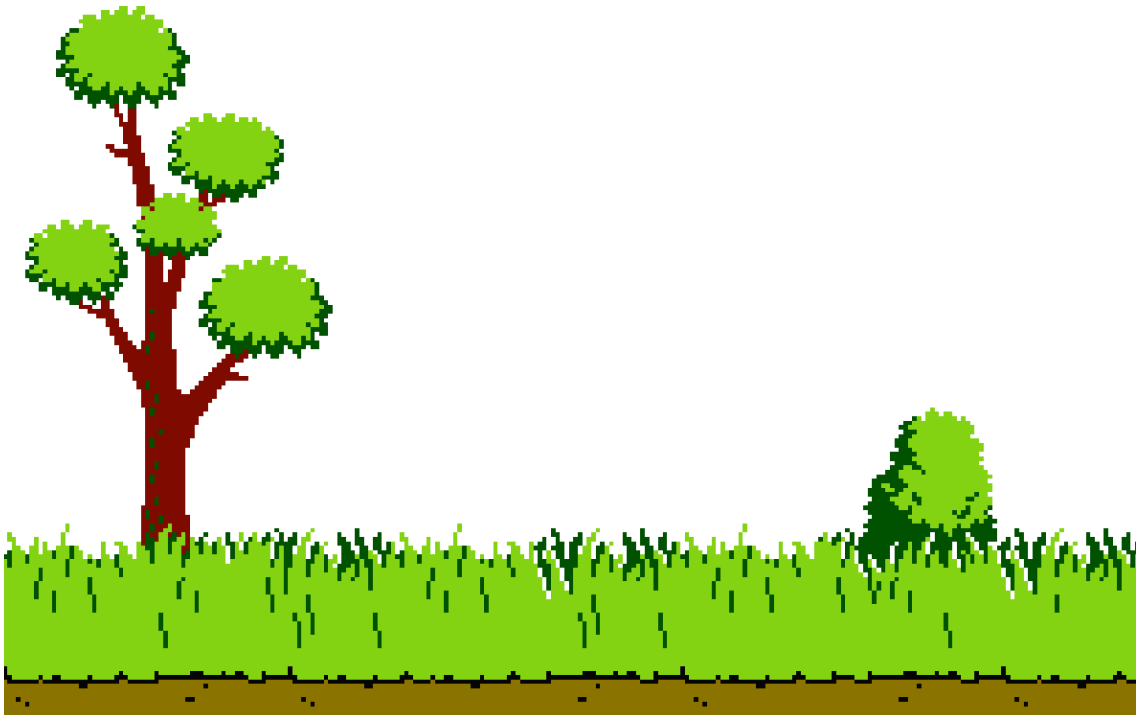
Grafika do gry została stworzona w programie Adobe Illustrator. To zaawansowany program graficzny stworzony do projektowania i edycji grafiki wektorowej, będący flagowym produktem firmy Adobe Inc. Program jest kompatybilny zarówno z systemem Windows, jak i macOS, a projekty są zapisywane w formacie plików .ai.

Adobe Illustrator zajmuje jedno z czołowych miejsc wśród programów do tworzenia i edycji grafiki wektorowej na świecie, obok konkurencyjnego CorelDRAW. Ilustracje tworzone w tym programie, zapisywane w formacie .ai, są kompatybilne z innymi narzędziami Adobe do grafiki, wideo i animacji. Niemniej jednak, pliki o rozszerzeniu .cdr, specyficzne dla Corela, nie są otwieralne w Illustratorze.



Rys. 3 birds wykonane w Adobe Illustrator

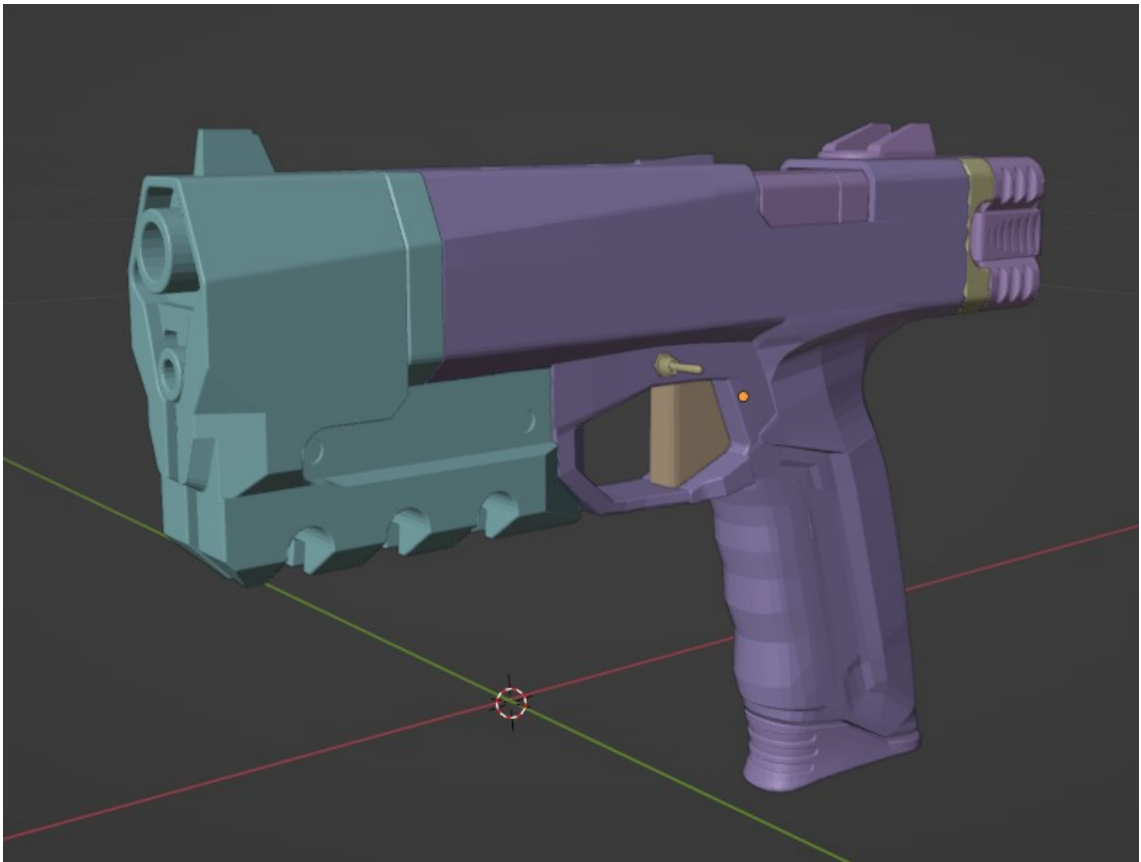
Powyższy rysunek przedstawia poszczególne fazy, zmieniającej się grafiki w trakcie gry. Obrazy są wykonane w czterech wariantach kolorów: niebieskim, fioletowym, żółtym oraz czerwonym.



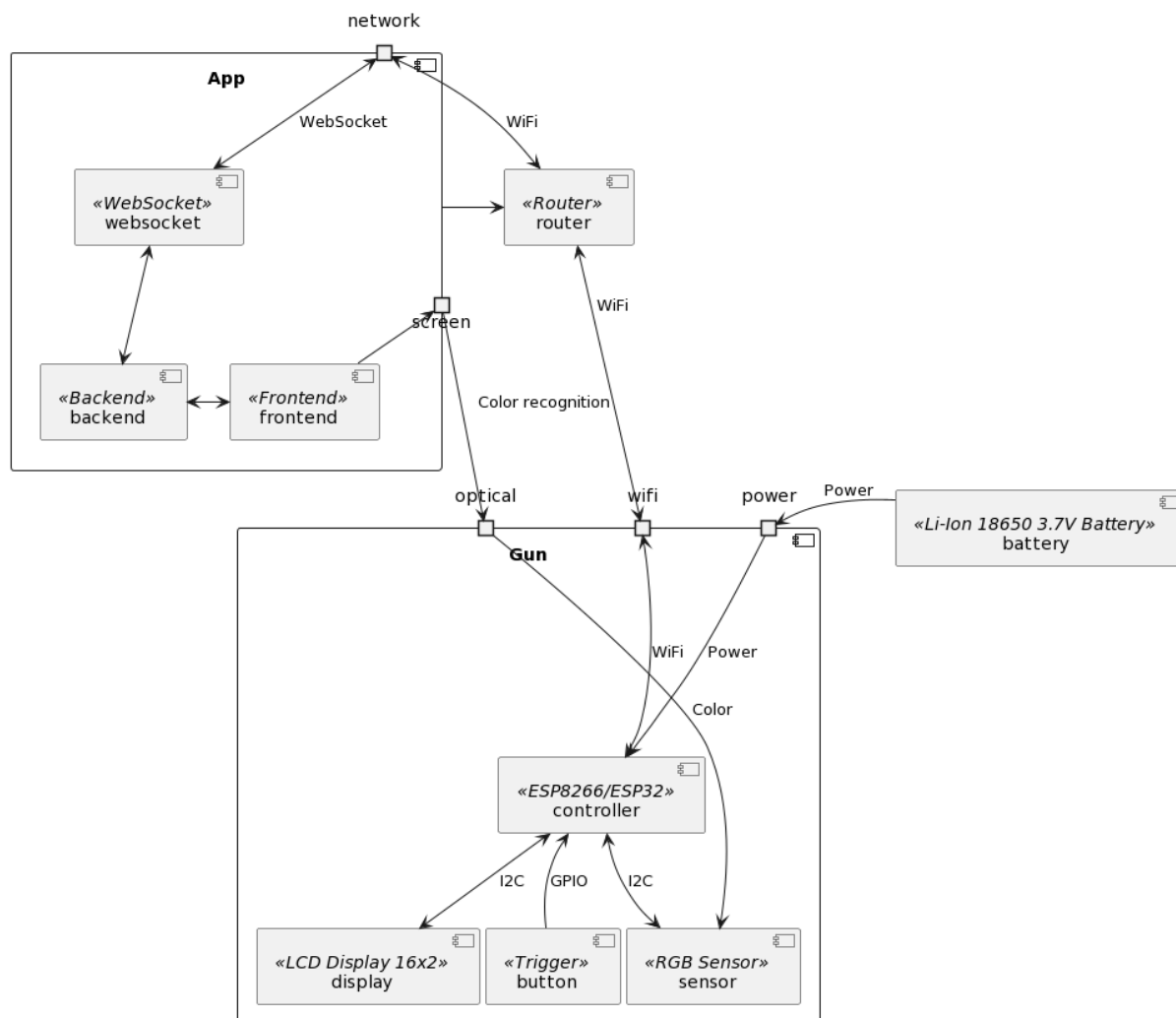
Rys. 4 Tło gry

5. Model pistoletu





6. Architektura systemowa



Rys. 1 Schemat pistoletu zasilanego na baterię

Schemat przedstawia zestaw komponentów i ich powiązania w aplikacji sterującej bronią (prawdopodobnie karabinem) zasilaną przez baterię. Oto opis poszczególnych komponentów:

6.1. battery (Bateria Li-Ion 18650 3.7V)

Komponent reprezentujący baterię, która zasilana jest z zewnątrz.

6.2. Gun (pistolet)

Główny komponent reprezentujący pistolet, który składa się z:

- **controller** (Kontroler): Układ mikrokontrolera ESP8266/ESP32 odpowiedzialny za sterowanie bronią.
- **button** (Przycisk): Przycisk służący jako wyzwalacz, do strzelania.
- **sensor** (Czujnik RGB): Czujnik do rozpoznawania kolorów, jest wykorzystywany w celu identyfikacji celów.

- **display** (Wyświetlacz LCD 16x2): Wyświetlacz LCD służący do wyświetlania informacji o stanie broni.
- **Porty wejściowe:**
 - **power** (Port zasilania): Port zasilania broni.
 - **optical** (Port optyczny): Port optyczny, prawdopodobnie do komunikacji z czujnikiem.
- **Port wyjściowy:**
 - **wifi** (Port WiFi): Port WiFi, prawdopodobnie do komunikacji z aplikacją lub serwerem.

6.3. router (Router)

Komponent reprezentujący router, prawdopodobnie wykorzystywany do zestawiania połączenia Wi-Fi.

6.4. App (Aplikacja)

Komponent reprezentujący aplikację sterującą bronią, składającą się z:

- **websocket** (WebSocket): Komunikacja poprzez protokół WebSocket, zapewniająca dwustronną komunikację między aplikacją a serwerem lub urządzeniem.
- **frontend** (Warstwa frontendowa): Warstwa interfejsu użytkownika, która może być widoczna dla użytkownika.
- **backend** (Warstwa backendowa): Warstwa obsługująca logikę biznesową i przetwarzanie danych.
- **Port wyjściowy:**
 - **screen** (Ekran): Port wyjściowy do ekranu, prawdopodobnie do wyświetlania informacji o stanie broni.

6.5 frontend (Warstwa frontendowa)

Jest połączony z ekranem (screen) za pomocą portu wyjściowego i połączony z backendem.

6.6. screen (Ekran)

Jest połączony z portem wyjściowym aplikacji (screen) i połączony z portem wejściowym optycznym pistoletu (optical) za pomocą portu wyjściowego.

6.7. sensor (Czujnik RGB)

Jest połączony z portem wejściowym optycznym pistoletu (optical) za pomocą portu wyjściowego.

Schemat pokazuje hierarchię komponentów i relacje między nimi, takie jak zasilanie, komunikacja WiFi, oraz interakcje między warstwami aplikacji.

7. Wymagania dotyczące produktu

7.1. Rozgrywka

Tryby singleplayer i multiplayer, pozwalające na udział jednego lub więcej graczy.

7.2. Kalibracja

Funkcja kalibracji umożliwiająca dostosowanie czułości i dokładności kontrolera pistoletu.

7.3. Interakcja z grą

- Integracja kontrolera pistoletu z aplikacją gry/stroną internetową przy użyciu adresu IP pistoletu.
- Realistyczna mechanika celowania i strzelania, podobna do oryginalnej gry Duck Hunt.

7.4. Cechy rozgrywki

- Różne poziomy trudności dla graczy na wszystkich poziomach umiejętności.
- Wysokiej jakości grafika i dźwięk poprawiające wrażenia z gry.
- Śledzenie wyników i funkcja tabeli liderów do rywalizacji.

7.5. Funkcja dla wielu

- Obsługa lokalnego trybu wieloosobowego z wieloma kontrolerami broni.
- Dostępność i interfejs użytkownika.
- Przyjazny dla użytkownika interfejs do konfigurowania pistoletu i uruchamiania gry.