



Uniwersytet Rzeszowski
Kolegium Nauk Przyrodniczych
Instytut Informatyki

Przedmiot - Technologie internetowe

Temat projektu: Battleships JavaScript

Prowadzący:

mgr Wojciech Gałka

Autor:

Konrad Pluta

nr albumu:

125152

Kierunek: Informatyka, grupa lab 2

1. Strona tytułowa

Tytuł projektu: Battleships JavaScript – gra w statki przy użyciu języka JavaScript oraz elementów HTML i CSS.

Projekt składa się z dwóch stron: **menu** oraz **gry**. Każda ze stron została zdefiniowana przy pomocy HTML i wystylizowana z użyciem dedykowanych dla stron plików CSS. Strony korzystają również z JavaScriptu w celu poprawnego działania oraz obsługi mechanik gry.

Opis struktury strony:

- **Strona główna (index.html)** – punkt startowy strony, posiada tytuł, pole do podania nazwy użytkownika, który będzie użyty do gry. Na stronie znajduje się również pole jednokrotnego wyboru w celu wybrania motywu kolorystycznego. Na samym dole występuje przycisk do rozpoczęcia gry.
- **Strona z grą (game.html)** – zawiera tytuł, dwie plansze: gracza i komputera, informacje na temat gry takie jak: aktualna tura, informacje o oddanym ruchu (trafienie/spudłowanie), zliczanie strzałów.
Pod planszą znajduje się kontener z dostępnymi statkami dla gracza oraz przyciski do obrócenia statku, zresetowania gry, rozpoczęcia gry i wyciszeniu dźwięków towarzyszących podczas rozgrywki.
Na samym dole występuje slider odpowiedzialny za powiększanie planszy.

Skrypty JavaScript oraz funkcjonalność:

- **menuGameSettings.js** – skrypt obsługuje dane takie jak wybrany motyw, nazwa użytkownika, następnie parsuje je do formatu JSON oraz przesyła do strony game.html po kliknięciu przycisku „Graj”.
- **app.js** – główny skrypt całego projektu, odpowiedzialny za całą mechanikę gry oraz części wyświetlanych elementów.

Stylizacja CSS:

- Każda ze stron posiada dedykowany arkusz stylów CSS odpowiedni dla ich treści i układu.

2. Zdefiniowanie problemu do realizacji.

Projekt zakłada stworzenie prostej i intuicyjnej gry w statki z prostymi efektami wizualnymi oraz dźwiękowymi.

Projekt umożliwia rozegranie partii z komputerem, który losowo rozkłada swoje statki oraz poprawnie – tzn. statki nie nachodzą na siebie oraz pozostają w wymiarach planszy.

Projekt zakłada, że interakcja pomiędzy użytkownikiem a grą będzie odbywała się za pomocą myszy. Użytkownik będzie miał możliwość wybierania poszczególnych statków oraz pól na planszy w celu wykonania ruchu. Ta metoda interakcji ma na celu zapewnienie intuicyjnego i płynnego doświadczenia dla gracza.

Problem do realizacji:

Problemem jest utworzenie poprawnie działającego skryptu, który:

- **Tworzenie plansz do gry:**
 - Skrypt tworzy dwie plansze dla użytkownika oraz dla komputera, składającą się z pól do zaznaczania w celu wykonania ruchów.
 - Rozmieszczenie statków komputera w losowych miejscach oraz zabezpieczenie statków przed błędnym umiejscowieniem tzn. brak możliwości wyjścia poza plansze, nie nachodzenie się statków.
- **Umiejscowienie statków gracza:**
 - Skrypt pozwala umieścić statki gracza na jego planszy, umożliwiając ich obrócenie w o 90 stopni w osi x lub y.
 - Skrypt sprawdza czy statek nie wystaje poza planszę oraz czy nie nachodzi na inną jednostkę.
- **Wykonywanie ruchów gracza i komputera:**
 - Skrypt odczytuje aktualną turę gracza lub komputera, weryfikuje oddany strzał, czy został trafiony statek lub nie, oraz czy nie trafiamy już w zestrzelony statek.
 - Skrypt liczy zestrzelone statki obu graczy i sprawdza czy zostały zestrzelone wszystkie.
 - Skrypt wyświetla aktualne statystyki gry takie jak: informacje o danej turze, informacje o oddanym strzale, informacje o zliczanych strzałach.
- **Funkcjonalności:**
 - Skrypt umożliwia zresetowanie partii gry, wyciszenie dźwięków oraz przybliżyć planszę za pomocą slidera.

- Skrypt pokazuje miejsce w którym zostanie położony przez użytkownika statek, tak aby mógł dokładnie zobaczyć jego lokalizację przed umiejscowieniem go na planszy.

- **Intuicyjność:**

- Gra jest intuicyjna dla każdego użytkownika bez względu na poziom doświadczenia.

Zaproponowane rozwiązania problemów:

W celu rozwiązania wyżej wymienionych problemów, proponowane są poniższe kroki:

- **Tworzenie plansz do gry:**

- Główny skrypt gry „app.js” za pomocą funkcji tworzy dwie tablice jako znacznik <div>, następnie przy użyciu pętli są wypełniane blokami <div>, które będą służyć jako pola planszy.
- Skrypt posiada funkcję odpowiedzialną za rozmieszczenie statków na planszy dla komputera. Działa ona w następujący sposób:
 - Najpierw wybiera losowo orientację statku (poziomo lub pionowo) za pomocą zmiennej.
 - Następnie wybiera losowy indeks startowy na planszy.
 - Za pomocą funkcji sprawdza, czy wybrane miejsce na planszy jest dostępne i czy statek może być tam umieszczony. Ta funkcja zwraca obiekt zawierający informacje o blokach statku, czy miejsce jest prawidłowe i czy nie jest już zajęte.
 - Jeżeli miejsce jest prawidłowe i nie jest zajęte, statek jest umieszczany na planszy poprzez dodanie odpowiednich klas do bloków planszy.
 - Jeżeli miejsce nie jest prawidłowe lub jest już zajęte, funkcja od dodawania statków jest wywoływana ponownie, aby znaleźć nowe miejsce dla statku.

W ten sposób komputer samodzielnie rozmieszcza swoje statki na planszy. Proces ten jest powtarzany dla każdego statku w tablicy.

- **Umiejscowienie statków gracza:**

- Użytkownik zaczyna przeciągać statek, co wywołuje funkcję, która ustawia zmienną na statek, który jest aktualnie przeciągany.
- Gdy użytkownik przeciąga statek nad planszą, wywołwana jest funkcja, która podświetla obszar, na który statek może zostać upuszczony.
- Gdy użytkownik upuszcza statek na planszy, wywołwana jest funkcja która następnie wywołuje inną funkcję z identyfikatorem bloku, na którym statek został upuszczony, oraz statkiem, który został upuszczony.
- Funkcja sprawdza, czy statek może zostać umieszczony w danym miejscu. Jeżeli miejsce jest prawidłowe i nie jest zajęte, statek jest umieszczany na planszy. W przeciwnym razie, statek nie jest

umieszczany, a zmienna odpowiedzialna za oznaczenie czy statek nie został umieszczony jest ustawiana na „prawda”.

- Jeżeli statek został prawidłowo umieszczony na planszy, jest on usuwany z listy statków do rozmieszczenia.

W ten sposób użytkownik może rozmieścić swoje statki na planszy, przeciągając je na wybrane miejsca. Proces ten jest powtarzany dla każdego statku, który użytkownik chce rozmieścić.

- **Wykonywanie ruchów gracza i komputera:**

- Funkcja symuluje ruch komputera w grze. Oto, jak to działa:
 - Komputer wybiera losowo pole na planszy gracza do strzału.
 - Jeżeli pole jest już trafione, komputer wykonuje kolejny ruch.
 - Jeżeli pole jest zajęte przez statek i nie jest jeszcze trafione, komputer trafia w statek, dodaje klasę statku do listy trafień komputera i sprawdza wynik.
 - Jeżeli pole nie jest zajęte, komputer chybia i kończy swoją turę.
 - Po wykonaniu ruchu, następuje tura gracza. Komputer aktywuje zdarzenie kliknięcia dla każdego pola na swojej planszy, umożliwiając graczowi wykonanie ruchu.

W ten sposób komputer wykonuje swoje ruchy w grze. Proces ten jest powtarzany, dopóki gra nie dojdzie do końca.

- Funkcja obsługuje ruchy gracza, następnie kolejna funkcja sprawdza stan gry. Oto, jak to działa:
 - Gdy gracz klika na pole na planszy komputera:
 - Jeżeli pole jest już trafione, gracz traci turę.
 - Jeżeli pole jest zajęte przez statek i nie jest jeszcze trafione, gracz trafia w statek i dodaje klasę statku do listy swoich trafień.
 - Jeżeli pole nie jest zajęte, gracz chybia i kończy swoją turę. Następnie, tura przechodzi do komputera.
 - Ta funkcja sprawdza, czy którykolwiek z graczy zatopił wszystkie statki przeciwnika.
 - Dla każdego rodzaju statku, sprawdza, czy wszystkie jego części zostały trafione. Jeżeli tak, statek jest zatopiony.
 - Jeżeli gracz zatopił wszystkie statki komputera, gracz wygrywa grę.
 - Jeżeli komputer zatopił wszystkie statki gracza, komputer wygrywa grę.

W ten sposób są obsługiwane ruchy gracza i sprawdzany jest stan gry. Proces ten jest powtarzany, dopóki gra nie dojdzie do końca

- **Funkcjonalności**

- Suwak umożliwia zmianę skali planszy gry. Działa to w następujący sposób:
 - Użytkownik przesuwając suwak, co wywołuje funkcję odpowiedzialną za skalę planszy.
 - Funkcja aktualizuje skalę planszy na wartość suwaka, co powoduje powiększenie lub pomniejszenie planszy.

- Dodatkowo, margines planszy jest dostosowywany proporcjonalnie do wartości suwaka, co utrzymuje planszę na środku ekranu podczas skalowania.

W ten sposób suwak umożliwia użytkownikowi dostosowanie rozmiaru planszy gry od 1.0 do 1.5 z krokiem co 0.1.

- Przycisk obracania statku: Gdy użytkownik kliknie ten przycisk, jest wywoływana funkcja, która zmienia kąt obrotu statków w kontenerze między 0 a 90 stopni, co pozwala na obracanie statków.
- Przycisk rozpoczęcia gry: Gdy użytkownik kliknie ten przycisk, funkcja jest wywoływana. Ta funkcja sprawdza, czy wszystkie statki gracza zostały rozmieszczone na planszy. Jeżeli tak, gra się rozpoczyna. W przeciwnym razie, gracz jest proszony o rozmieszczenie wszystkich swoich statków.
- Przycisk wyciszenia dźwięków: Gdy użytkownik zaznaczy to pole wyboru, wszystkie dźwięki w grze są wyciszane. Gdy pole wyboru jest odznaczone, dźwięki są odtwarzane.
- Przycisk resetowania gry: Gdy użytkownik kliknie ten przycisk, funkcja jest wywoływana. Ta funkcja odświeża stronę, co skutkuje zresetowaniem gry do stanu początkowego.
- **Intuicyjność:**
 - Dzięki wykorzystaniu HTML i CSS, gra posiada prosty i intuicyjny interfejs, który ułatwia nawigację i interakcję. Elementy interfejsu są zaprojektowane w taki sposób, aby były zrozumiałe i łatwe do użycia. Wyświetlane alerty i informacje dodatkowo wspierają gracza podczas rozgrywki, dostarczając niezbędnych wskazówek i informacji zwrotnych. Wszystko to przyczynia się do płynnej i przyjemnej rozgrywki.

Cel projektu:

Stworzenie prostej i intuicyjnej gry dla każdego użytkownika z elementami wizualnymi oraz dźwiękowymi, która może zachęcić do nauki języka JavaScript, HTML oraz CSS.

3. Propozycja rozwiązania problemu

1. Struktura oraz organizacja gry:

- Cel: Łatwe zarządzanie plikami gry, głównie kodem.
- Rozwiązanie:
 - Rozdzielenie menu startowego od gry, poprzez stworzenie dedykowanych plików JavaScript, HTML oraz CSS, które zawierają specyficzne dla siebie znaczniki, style oraz skrypty.
- Implementacja:
 - Kod jest zorganizowany względem wyżej wymienionych części, z przemyślanym nazewnictwem klas, identyfikatorów, funkcji oraz różnych zmiennych.
 - Wykorzystanie czytelnego formatowania aby ułatwić poruszanie się w kodzie.

2. Responsywność oraz wygląd na komputerach stacjonarnych oraz laptopach:

- Cel: Zapewnienie prostego, schludnego oraz funkcjonalnego wyświetlania na wyżej wymienionych urządzeniach.
- Rozwiązanie:
 - Stosowanie responsywnych technik jak media queries, flexbox w plikach CSS, aby szablon dostosowywał się do różnych wielkości ekranów.
- Implementacja:
 - Każdy element zostanie zaprojektowany dbając o estetykę oraz dostosowywanie się do różnych rozdzielczości.
 - Za pomocą elementów JavaScript użytkownik będzie mógł wybrać motyw kolorystyczny, który zostanie załadowany po wejściu do gry.
 - Za pomocą JavaScript, HTML i CSS gra umożliwia przybliżenie planszy.

3. Interaktywność:

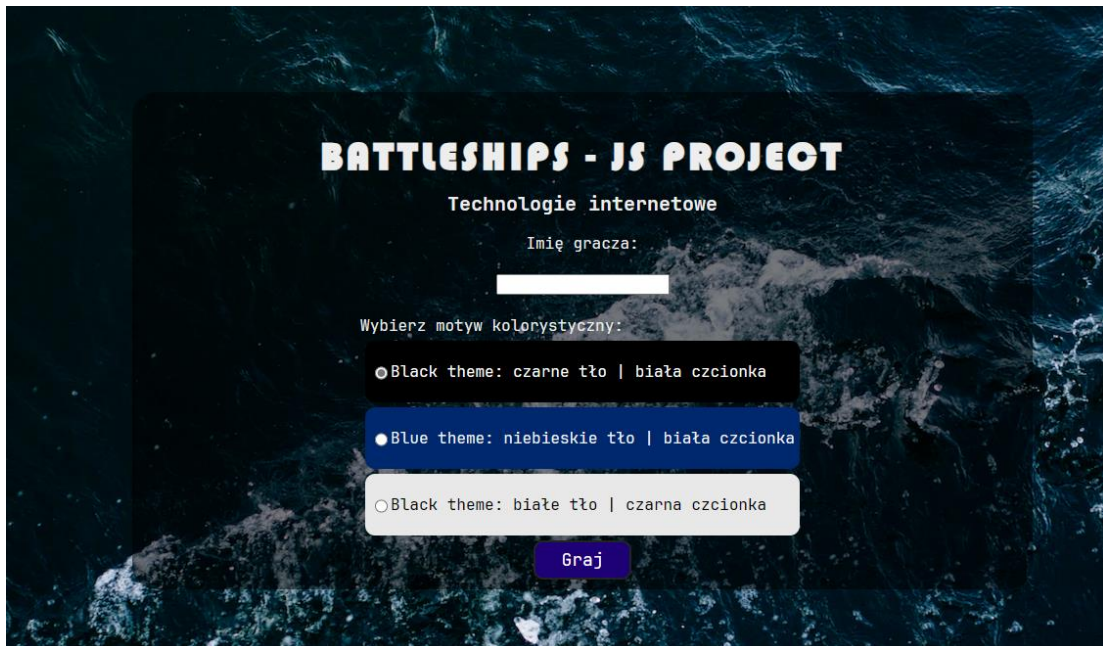
- Cel: stworzenie interaktywnej gry, w której umieszczamy statki poprzez przeciągnięcie ich na planszę, oddanie ruchu poprzez wybranie miejsca na planszy przeciwnika, występowanie efektów dźwiękowych oraz wizualnych.
- Rozwiązanie:
 - Za pomocą arkusza stylu CSS, ustawienie wyglądu statków, interfejsu oraz planszy. Wykorzystanie języka JavaScript w celu obsługi interaktywności oraz mechanik gry.
- Implementacja:
 - Stworzenie specjalnych funkcji które będą odpowiedzialne za obsługę dźwięków, oraz części wizualnych.
 - Odpowiednie dostosowanie stylu w CSS dla konkretnych elementów.

Podsumowanie:

Założeniem projektu jest stworzenie prostej, przyjemnej oraz interaktywnej gry w statki. Responsywność, interaktywne elementy, intuicyjny design mają za zadanie zaangażować gracza w rozgrywkę oraz ułatwić poruszanie się po grze niezależnie od stopnia zaawansowania użytkownika. Kolejną rzeczą jest możliwość zainteresowania użytkownika ww. technologiami aby spróbować odtworzyć np. taki projekt bądź inny.

4. Testowanie

1. Menu główne:

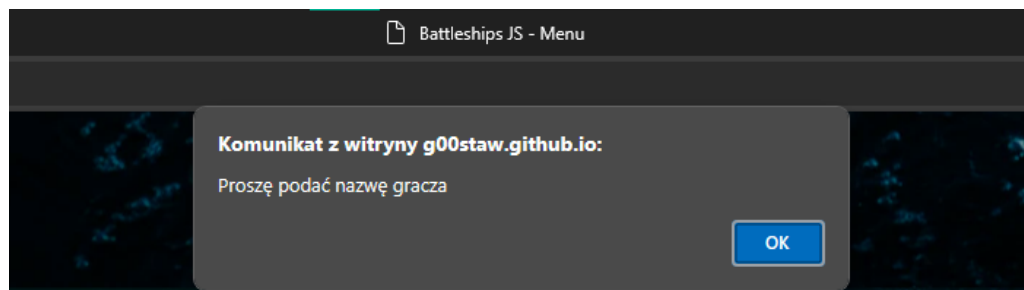


Rys. 1.0

Menu główne zawiera:

- Tytuł główny gry,
- Pole do wyboru gracza
- Wybór motywu kolorystycznego
- Przycisk graj.

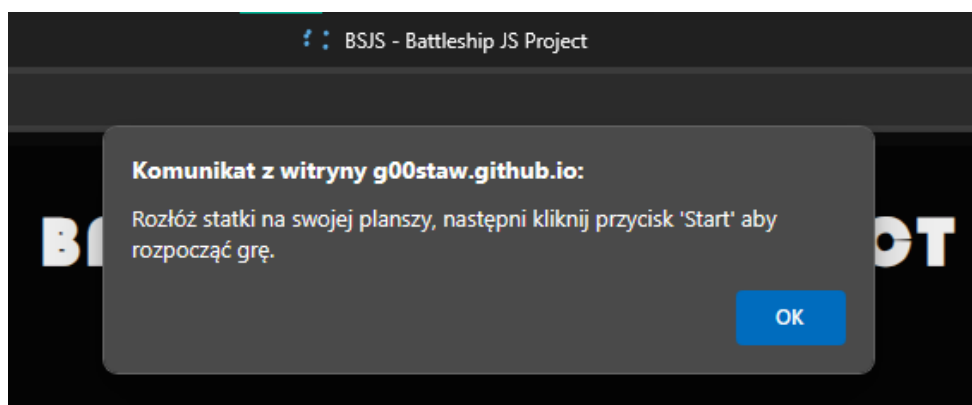
Do rozpoczęcia gry wymagane jest podanie nazwy gracza, które będzie użyte w trakcie rozgrywki. Jeśli pole zostanie puste, menu jest odpowiednio zabezpieczone i nie pozwoli na taki stan rzeczy. Użytkownik zostanie poinformowany o tym w komunikacie:



Rys. 1.1

Motyw graficzny domyślnie jest zaznaczony na „czarne tło | biała czcionka” jak widać na rysunku 1.0.

2. Gra.



Rys. 1.2

Po przekierowaniu podaniu nazwy użytkownika, pojawia się komunikat z instrukcją, aby użytkownik rozłożył dostępne statki a następnie wystartował grę.

Po zatwierdzeniu komunikatu, zostaje załadowana gra (rys. 1.3) z wybranym motywem kolorystycznym.

Na samej górze znajduje się tytuł gry, poniżej – okno informacyjne z takimi danymi jak:

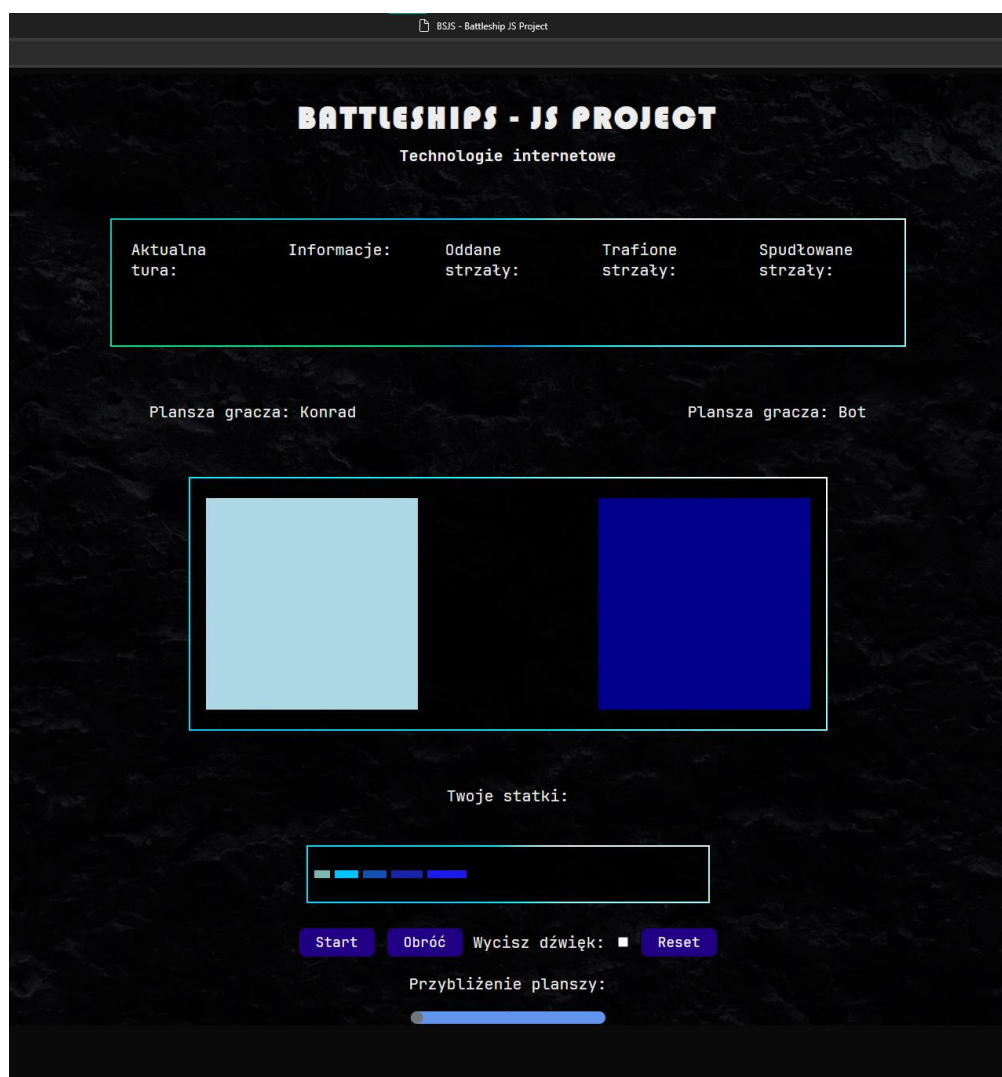
- Aktualna tura,
- Informacje – o spudłowaniu lub trafieniu celu przez aktualnego gracza
- Oddane, trafione i spudłowane strzały przez użytkownika.

Niżej znajdują się plansze gry z podpisem do kogo należy poszczególna plansza.

Pod planszami jest kontener ze statkami gracza które należy umieścić na planszy.

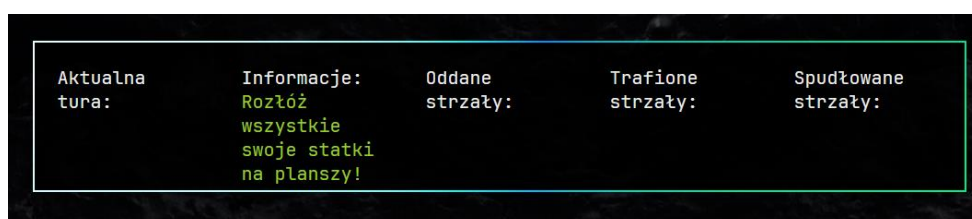
Na samym dole występują przyciski funkcyjne takie jak:

- Start – wystartowanie gry po rozłożeniu statków.
- Obróć – obrócenie statków o 90 stopni.
- Wycisz dźwięki – wyciszenie dźwięków występujących podczas rozgrywki.
- Reset – ponowne załadowanie strony, czyli zresetowanie aktualnej gry.
- Przybliżenie planszy – skaluje rozmiar planszy od 1.0 do 1.5 z przejściami co 0.1.



Rys. 1.3

W momencie wystartowania rozgrywki bez rozłożonych statków, użytkownik zostanie poinformowany o popełnionym błędzie.



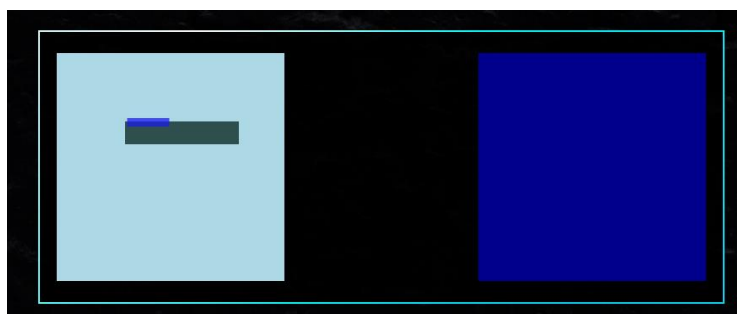
Rys. 1.4

Podczas rozkładania statków na planszy użytkownik może obrócić swoje statki:



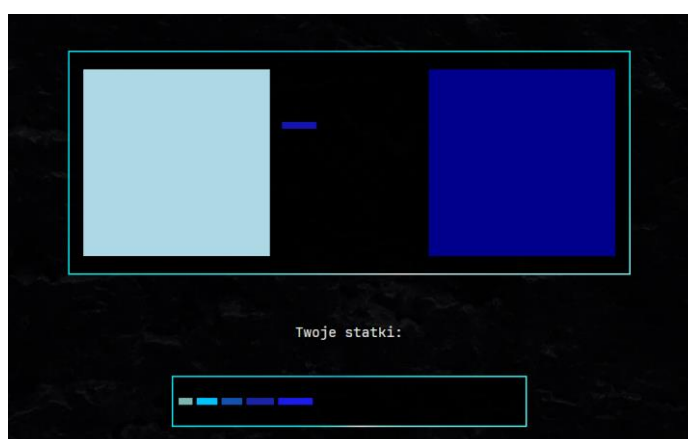
Rys. 1.5

W momencie przesuwania statków na plansze, wyświetla się cień w którym zostanie umieszczony statek.



Rys. 1.6

W przypadku błędnego położenia statku, zostanie on cofnięty do kontenera aby ponownie go umieścić.



Rys. 1.7

Po poprawnym umieszczeniu statku zostanie on pokazany na planszy oraz usunięty z kontenera.



Rys. 1.8

Każdy ze statków posiada unikalne grafiki oraz podczas rozmieszczania statki są zabezpieczone przed nachodzeniem na siebie.



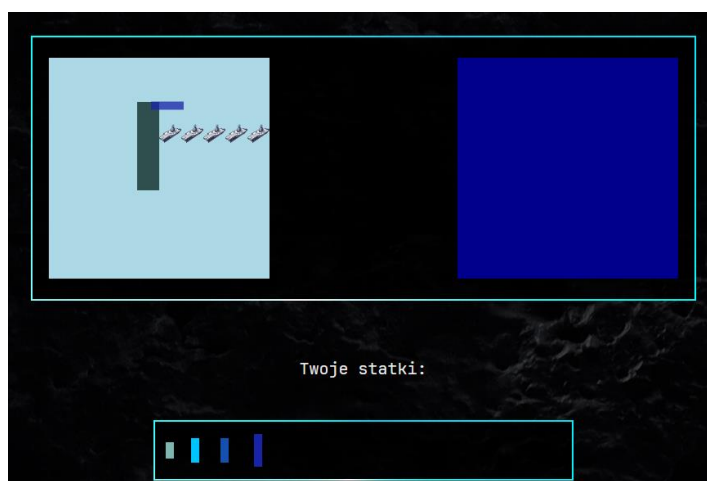
Rys. 1.9

Taki statek nie rzuca cienia, czyli nie pokazuje miejsca w którym zostanie umieszczony. Gdy taki statek zostanie opuszczony, wróci do kontenera ze statkami.

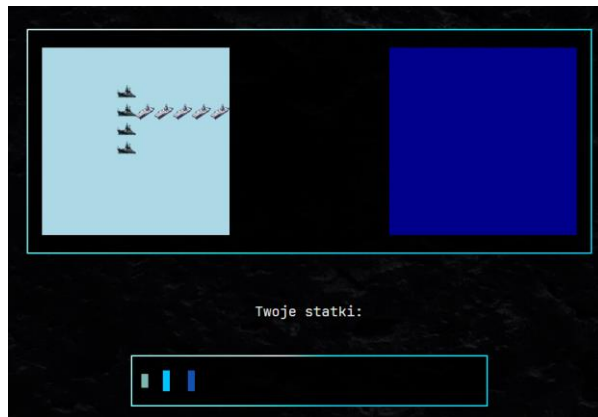


Rys. 2.0

Tyczy się to również braku odpowiedniej odległości pomiędzy statkami aby je umieścić. Tak wygląda poprawne miejsce na umieszczenie statku:

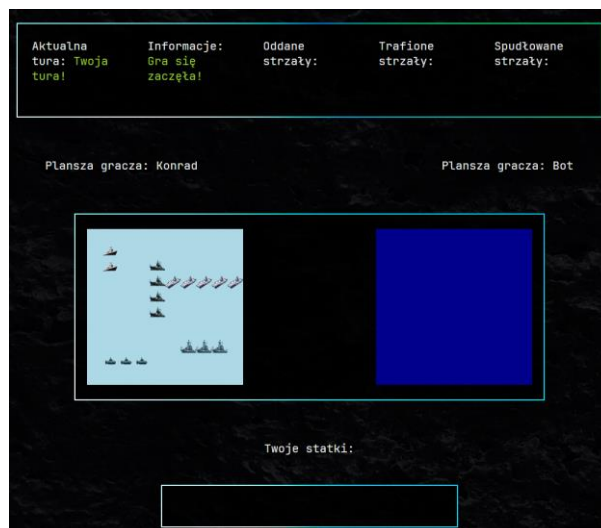


Rys. 2.1



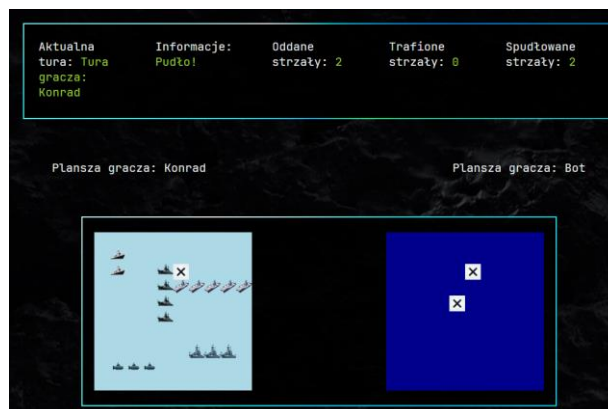
Rys. 2.2

Po rozłożeniu statków i rozpoczęciu gry, gracz zostanie poinformowany o wykonaniu ruchu.



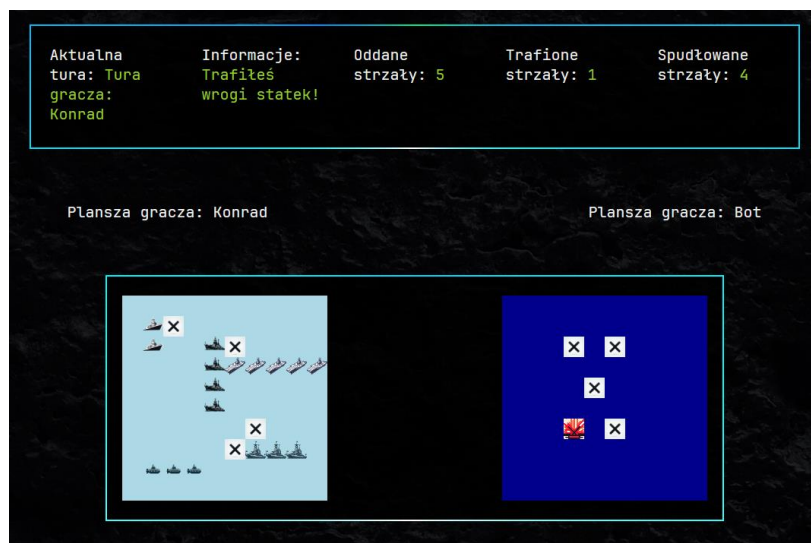
Rys. 2.3

Po wykonaniu ruchu, zostanie on zaznaczony na tablicy informacyjnej oraz na planszy za pomocą ikony „X”.



Rys. 2.4

Trafienie wrogiego statku zostaje podliczone oraz odpowiednio zaznaczone na planszy.



Rys. 2.5

Po zatopieniu całego statku zostanie wyświetlona informacja o typie zatopionego statku:



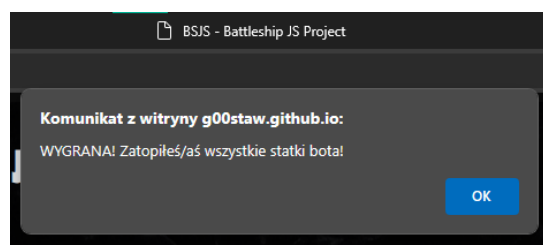
Rys. 2.6

W momencie chęci strzelenia w trafiony już statek, zostanie wyświetlona stosowna informacja o utracie tury oraz podliczenia strzału jako pudło.



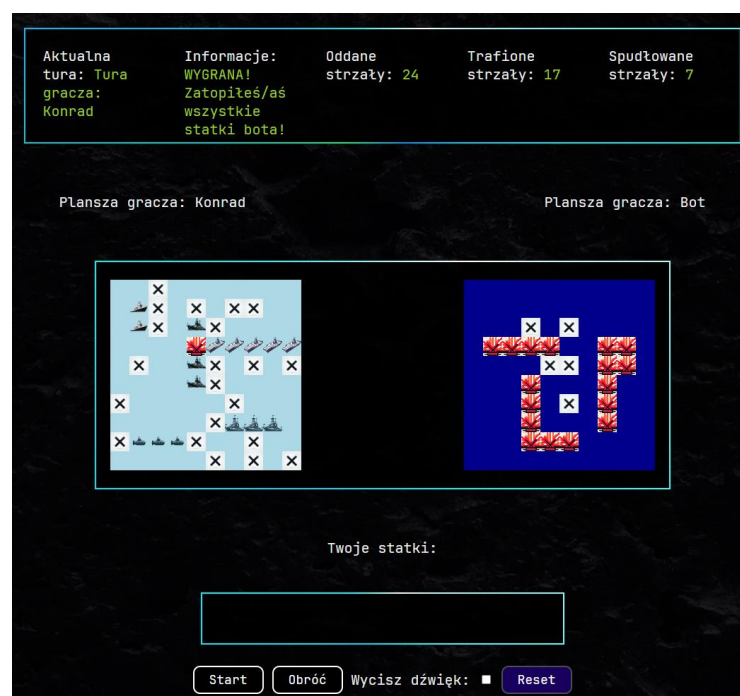
Rys. 2.7

Po zatopieniu wszystkich statków bota lub gracza zostanie wyświetlony stosowny komunikat:



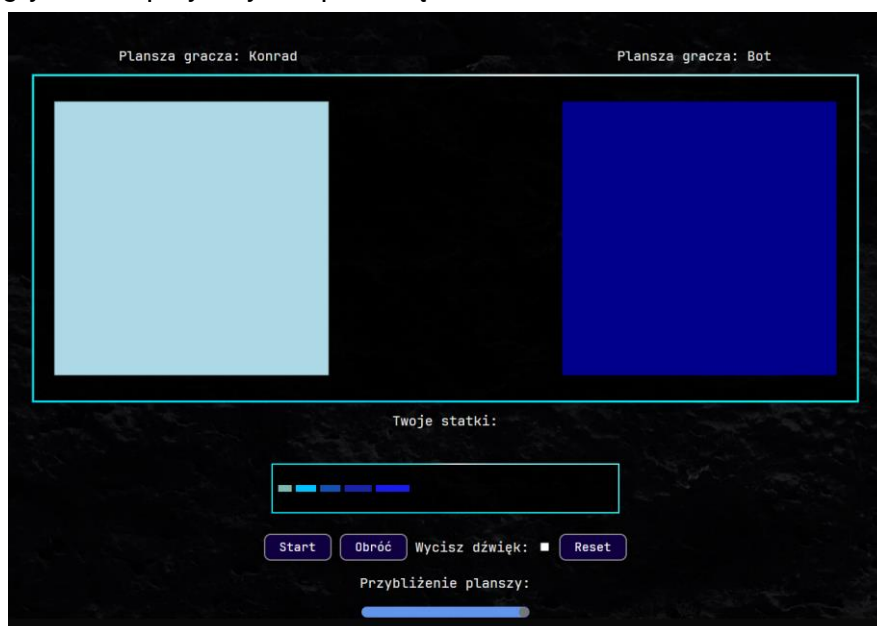
Rys. 2.8

Po zakończeniu rozgrywki można zobaczyć planszę oraz statystyki lub zresetować grę i ponownie rozegrać partię z komputerem.



Rys. 2.9

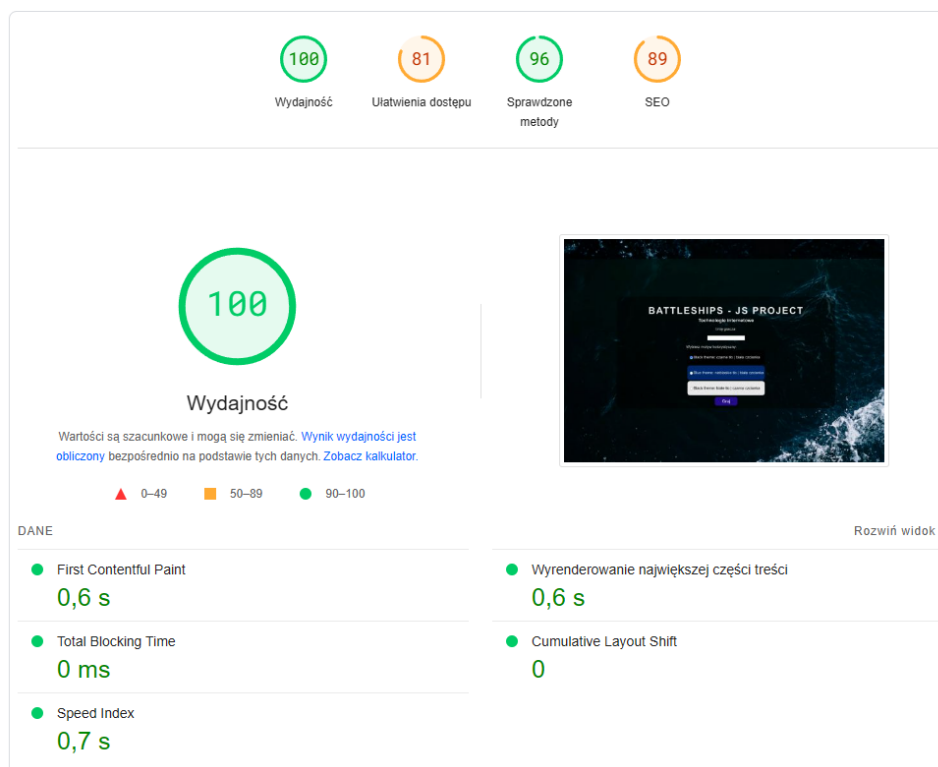
Planszę gry można przybliżyć za pomocą suwaka.



Rys. 3.0

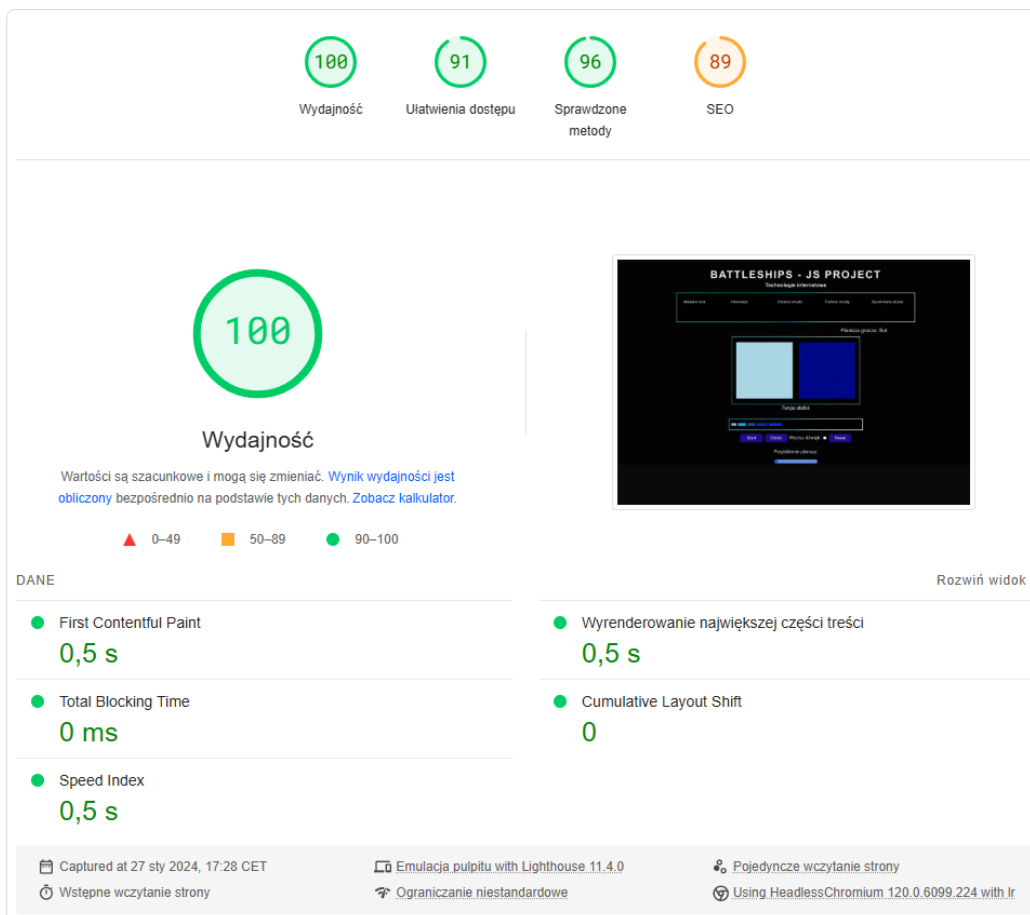
3. Testy wydajności

Menu główne:



Rys. 3.1

Gra:



Rys. 3.2

1. **Wydajność (100):** wynik wydajności jest doskonały. Oznacza to, że strona ładuje się bardzo szybko i oferuje płynne doświadczenie użytkownika.
2. **Ułatwienia dostępu (91):** wynik w tej kategorii jest bardzo wysoki, co wskazuje, że gra jest łatwo dostępna dla użytkowników z różnymi potrzebami.
3. **Sprawdzone metody (96):** wynik w tej kategorii jest również imponujący, co wskazuje, że zostało zastosowane wiele najlepszych praktyk w zakresie tworzenia stron internetowych.
4. **SEO (89):** wynik SEO jest dość wysoki, co wskazuje, że strona jest dobrze zoptymalizowana pod kątem wyszukiwarek. Można jednak dalej pracować nad SEO, aby poprawić widoczność swojej strony w wynikach wyszukiwania.

Podsumowując, stworzona gra w statki jest bardzo dobrze zoptymalizowana pod względem wydajności, dostępności, stosowania sprawdzonych metod i SEO. Nadal jednak istnieje pewien margines do ulepszeń, zwłaszcza w zakresie dostępności i SEO.

5. Dokumentacja kodu źródłowego

JSDoc:

- JavaScript wykorzystywany w projekcie jest dokumentowany za pomocą JSDoc, standardu dla komentarzy w kodzie JavaScript, który pozwala na generowanie atrakcyjnej i funkcjonalnej dokumentacji. Obejmuje to:
 - Opisy Funkcji i Metod: Jasne wyjaśnienia dotyczące tego, co robi każda funkcja lub metoda, jakie argumenty przyjmuje i co zwraca.

Home

Classes

Ship

Global

addShipPiece

checkScore

computerGo

createBoard

dragOver

dragStart

dropShip

flip

getValidity

handleClick

highlightArea

myFunction

startGame

Rys. 3.3

Methods

addShipPiece(user, ship, startId)

Ta funkcja jest odpowiedzialna za dodawanie statków do planszy gry. Działa zarówno dla gracza, jak i dla komputera. Dla komputera, rozmieszczenie statków jest losowe, natomiast dla gracza, statki są rozmieszczane na podstawie interakcji użytkownika.

Parameters:

Name	Type	Description
user	string	Określa, czy funkcja jest wywoływana dla gracza czy dla komputera.
ship	Object	Obiekt reprezentujący statek, który ma być rozmieszczony.
startId	number	Opcjonalny parametr określający indeks początkowy, od którego zaczyna się statek.

Source: [app.js, line 195](#)

checkScore(user, userHits, userSunkShips)

Ta funkcja sprawdza, czy którykolwiek z graczy zatopił wszystkie statki przeciwnika. Dla każdego rodzaju statku, sprawdza, czy wszystkie jego części zostały trafione. Jeżeli tak, statek jest zatopiony. Jeżeli gracz zatopił wszystkie statki komputera, gracz wygrywa grę. Jeżeli komputer zatopił wszystkie statki gracza, komputer wygrywa grę.

Parameters:

Name	Type	Description
user	string	Określa, czy funkcja jest wywoływana dla gracza czy dla komputera.
userHits	Array	Tablica zawierająca wszystkie trafienia danego użytkownika.
userSunkShips	Array	Tablica zawierająca wszystkie zatopione statki danego użytkownika.

Source: [app.js, line 512](#)

Rys. 3.4

Całość dokumentacji znajduje się pod linkiem:

<https://g00staw.github.io/BSJS/jsdoc/>

KSS:

Komentarze są używane, aby opisać, do czego służy każdy blok kodu oraz jakie są jego główne cele i funkcje. To pomaga zrozumieć strukturę i działanie kodu oraz ułatwia utrzymanie i ewentualne modyfikacje w przyszłości.

KSS Style Guide

- 0 Overview
- 2 Globalne ustawienia HTML
- 3 3
- 4 Stylizacja .title-box
- 5 Stylizacja .container
- 6 Stylizacja formularza
- 8 Animacja scroll
- 9 Zapytanie medialne dla min-width: 670px
- 10 Stylizacja .theme
- 11 Stylizacja .playButton
- 12 Animacja backgroundColor
- 13 Import czcionki
- 14 Globalne ustawienia HTML
- 15 Stylizacja nagłówka h1
- 17 Stylizacja .title-box
- 18 Stylizacja .info-container
- 20 Stylizacja .sub
- 21 Stylizacja .cru
- 22 Stylizacja .batt
- 23 Stylizacja .carr
- 24 Stylizacja .sub dla #player
- 25 Stylizacja .dest dla #player
- 26 Stylizacja .cru dla #player
- 26 Stylizacja .batt dla #player
- 27 Stylizacja .carr dla #player
- 28 Stylizacja .dest-prev
- 29 Stylizacja .sub-prev
- 30 Stylizacja .cru-prev
- 31 Stylizacja .batt-prev
- 32 Stylizacja .carr-prev
- 33 Stylizacja .dest, .sub, .carr, .cru, .batt dla #computer
- 34 Stylizacja .hover
- 36 Stylizacja .empty
- 37 Stylizacja .body
- 38 Animacja scroll
- 39 Zapytanie medialne dla prefers-reduced-motion
- 40 Zapytanie medialne dla min-width: 670px
- 41 Stylizacja .main-box
- 42 Stylizacja .slidecontainer
- 43 Stylizacja .info-container
- 44 Animacja rotate
- 45 Właściwość --angle
- 46 Stylizacja suwaka
- 47 Animacja backgroundColor
- 48 Stylizacja .startButton
- 49 Stylizacja .flipButton
- 50 Stylizacja .resetButton
- 51 Zapytanie medialne dla max-height: 1300px

KSS Style Guide

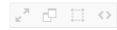
- 0 Overview
- 2 Globalne ustawienia HTML
- 3 3
- 4 Stylizacja .title-box
- 5 Stylizacja .container
- 6 Stylizacja formularza
- 8 Animacja scroll
- 9 Zapytanie medialne dla min-width: 670px
- 10 Stylizacja .theme
- 11 Stylizacja .playButton
- 12 Animacja backgroundColor
- 13 Import czcionki
- 14 Globalne ustawienia HTML
- 15 Stylizacja nagłówka h1
- 17 Stylizacja .title-box
- 18 Stylizacja .info-container
- 20 Stylizacja .sub
- 21 Stylizacja .cru
- 22 Stylizacja .batt
- 23 Stylizacja .carr
- 24 Stylizacja .sub dla #player
- 25 Stylizacja .dest dla #player
- 26 Stylizacja .cru dla #player
- 26 Stylizacja .batt dla #player
- 27 Stylizacja .carr dla #player
- 28 Stylizacja .dest-prev
- 29 Stylizacja .sub-prev
- 30 Stylizacja .cru-prev
- 31 Stylizacja .batt-prev
- 32 Stylizacja .carr-prev
- 33 Stylizacja .dest, .sub, .carr, .cru, .batt dla #computer
- 34 Stylizacja .hover
- 36 Stylizacja .empty
- 37 Stylizacja .body
- 38 Animacja scroll
- 39 Zapytanie medialne dla prefers-reduced-motion
- 40 Zapytanie medialne dla min-width: 670px
- 41 Stylizacja .main-box
- 42 Stylizacja .slidecontainer
- 43 Stylizacja .info-container
- 44 Animacja rotate
- 45 Właściwość --angle
- 46 Stylizacja suwaka
- 47 Animacja backgroundColor
- 48 Stylizacja .startButton
- 49 Stylizacja .flipButton
- 50 Stylizacja .resetButton
- 51 Zapytanie medialne dla max-height: 1300px

Dokumentacja KSS projektu "BSJS - Battleships JavaScript"

Witaj w dokumentacji. Ta strona zawiera szczegóły dotyczące wszystkich dostępnych stylów oraz ich zastosowań w projekcie. Z powodzeniem, Konrad Piatek.

Section 10.1

Stylizacja .black



Definiuje styl dla elementu o klasie 'black'. Ustawia tło na czarne i kolor tekstu na biały.

Example

Przykładowy tekst

► Markup

Source: menu.css, line 140

Section 10.2

Stylizacja .blue



Definiuje styl dla elementu o klasie 'blue'. Ustawia tło na ciemnoniebieskie i kolor tekstu na prawie biały.

Example

Przykładowy tekst

► Markup

Source: menu.css, line 151

Całość dokumentacji znajduje się pod linkiem:

<https://g00staw.github.io/BSJS/kss/styleguide/index.html>

6. Źródła.

- Wszelkie zdjęcia pochodzą ze strony <https://unsplash.com/> lub zostały wygenerowane za pomocą Bing AI Image Creator.
- Całość projektu znajduje się również w repozytorium publicznym na moim GitHubie: g00staw
- Historia commitów:

```

commit 8e306b94317af5656b86efb01baff87adecdf3 (HEAD -> master, origin/master)
Author: Konrad Pluta <konrad.k.j.pluta@gmail.com>
Date: Sun Jan 28 01:03:35 2024 +0100

    Dodanie dokumentacji KSS

commit 2c2e053b82fcd2eb503cc23cd7058e6b0922298
Author: Konrad Pluta <konrad.k.j.pluta@gmail.com>
Date: Sat Jan 27 20:35:22 2024 +0100

    Dodanie dok JSOC

commit 082290e9d314fdd752bd938e81decdad3765246
Author: Konrad Pluta <konrad.k.j.pluta@gmail.com>
Date: Fri Jan 26 18:30:01 2024 +0100

    xd

commit a8c5ff33b83deceecdd83eeed4cb733b03c1e
Author: Konrad Pluta <konrad.k.j.pluta@gmail.com>
Date: Fri Jan 26 18:23:59 2024 +0100

    ogarniecie wygladu dla mniejsz 27call

commit 43c7df498e4d4e8e6e319d81834bd2d01e4fd9
Author: Konrad Pluta <konrad.k.j.pluta@gmail.com>
Date: Fri Jan 26 17:51:35 2024 +0100

    dodanie czcionki

commit d4183ba08873cf777168cde2083aa51e0e26
Author: Konrad Pluta <konrad.k.j.pluta@gmail.com>
Date: Fri Jan 26 17:36:56 2024 +0100

    koniec

commit 7ddfcad6f5eefaec5861581063c7a4035b19487
Author: Konrad Pluta <konrad.k.j.pluta@gmail.com>
Date: Fri Jan 26 17:02:51 2024 +0100

    ukonczenie gry prawie

commit 25bf5c64d3de85093d6e972027364b39bb6e
Author: Konrad Pluta <konrad.k.j.pluta@gmail.com>
Date: Fri Jan 26 01:09:36 2024 +0100

    dodanie fajnych funkcji bratku tak tu gotuje

commit f9a86636e37bd9ab5186cd31d76c92199c64f
Author: g00staw <konrad.k.j.pluta@gmail.com>
Date: Wed Jan 24 23:48:19 2024 +0100

    UKONCZENIE PODSTAWOWEJ MECHANIKI GRY

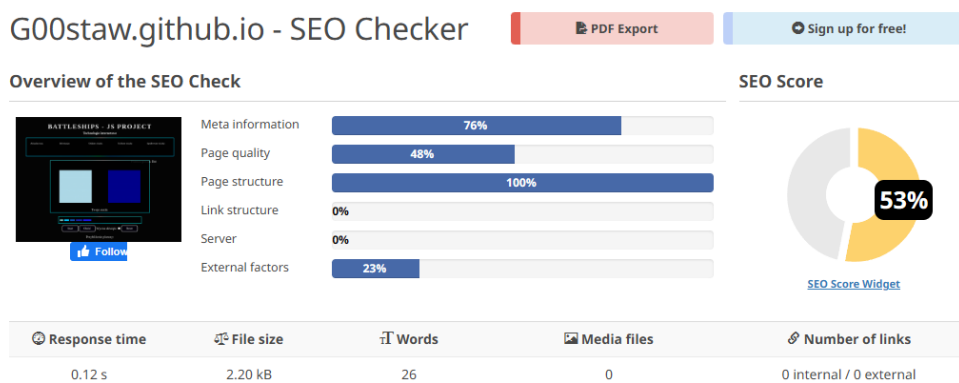
commit cba95992d1d66dbecf9d0a3999f93817cad1a2a
Author: g00staw <konrad.k.j.pluta@gmail.com>

```

Pełna historia znajduje się na ww. profilu github.

7. Raport SEO

a) <https://www.seobility.net/en/seocheck/>



Ocena SEO: Ogólna ocena SEO wynosi 53%, co jest przedstawione na wykresie kołowym. Ta ocena jest podsumowaniem wydajności strony we wszystkich mierzonych aspektach.

Media information: dobra ocena na poziomie 76%, co wskazuje, że tagi meta, takie jak tytuł i opis, są dobrze zoptymalizowane.

Page quality: Ocena wynosi 48%, co sugeruje, że istnieje sporo miejsca na poprawę jakości stron internetowych, możliwe obszary do poprawy to m.in. treść, czytelność lub wydajność strony.

Page structure: Strona otrzymała ocenę 100%, co wskazuje, że struktura strony, która może obejmować nagłówki i wykorzystanie elementów strukturalnych HTML5, jest idealnie zoptymalizowana.

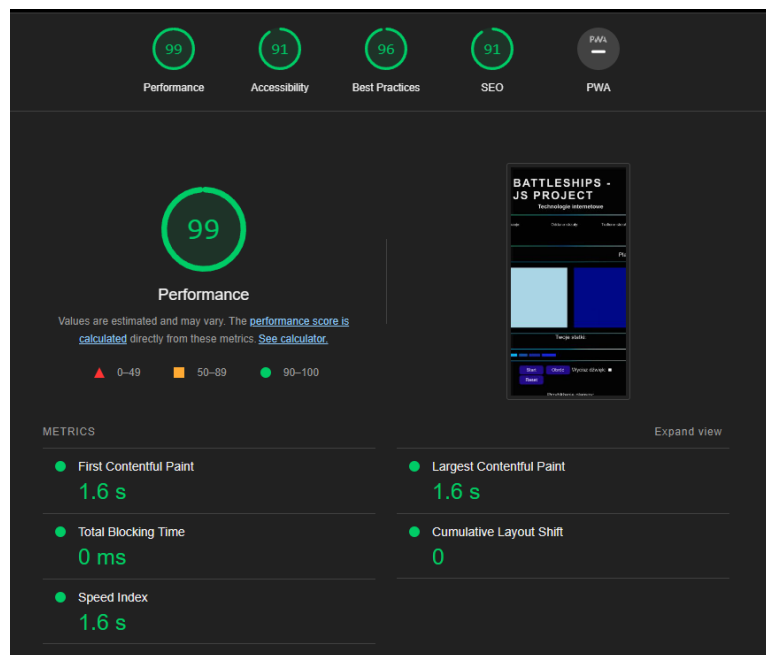
Czynniki Zewnętrzne: Ocena wynosi 23%, co jest dość niską wartością. Może to odnosić się do takich czynników jak linki zwrotne, obecność w mediach społecznościowych i inne czynniki zewnętrzne, które leżą poza stroną.

Czas Reakcji: Czas odpowiedzi serwera jest bardzo dobry, wynosi 0,12 sekundy.

Rozmiar Pliku: Całkowity rozmiar pliku strony wynosi 2.20 kB co jest bardzo lekkie i korzystne dla czasów ładowania.

Słowa: Na stronie znajduje się 26 słów. W zależności od kontekstu, może to być uznane za bardzo małą ilość pod kątem SEO, ponieważ dłuższe treści czasami mogą być lepiej pozycjonowane.

b) <https://developer.chrome.com/docs/lighthouse/>



Wydajność (Performance): 99/100 - Wynik wskazuje, że strona jest bardzo dobrze zoptymalizowana pod kątem szybkości i wydajności. Wysoka ocena w tej kategorii sugeruje krótki czas ładowania, efektywne wykorzystanie zasobów i dobrą reakcję na interakcje użytkowników.

Dostępność (Accessibility): 91/100 - Wynik wskazuje, że strona w ogólnym rozrachunku jest dostępna, ale mogą występować pewne problemy, które można by poprawić, aby strona była bardziej dostępna dla osób z niepełnosprawnościami, takie jak potencjalne problemy z kontrastem, etykietami formularzy lub strukturą nagłówków.

Najlepsze Praktyki (Best Practices): 96/100 - Ten wynik wskazuje na stosowanie prawie samych najlepszych praktyk w zakresie tworzenia stron internetowych, z bardzo małym marginesem na drobne poprawki.

SEO: 91/100 - Ocena SEO jest bardzo dobra, wskazuje to, że strona jest dobrze zoptymalizowana pod kątem wyszukiwarek internetowych.