Projekt semestralny

Gra „Saper”

Wykonawca:

Jakub Kłódkowski 2p1T

Poznań 2024

Spis treści

[1. Założenia projektu 3](#_Toc168696356)

[2. Opis funkcjonalności 4](#_Toc168696357)

[3. Instrukcja obsługi gry 9](#_Toc168696358)

[4. Kod źródłowy gry 14](#_Toc168696359)

[5. Wnioski końcowe 27](#_Toc168696360)

[6. Spis rysunków: 27](#_Toc168696361)

[7. Pola tekstowe: 28](#_Toc168696362)

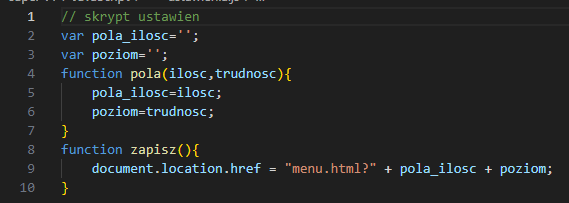
# Założenia projektu

1. Projekt semestralny polegał na stworzeniu gry „Saper”, która będzie zawierać odpowiednio 4 podstrony:
   1. Menu (*menu.html*) – zawierającą przyciski odpowiedzialne za działanie gry
   2. Ustawienia gry (*ustawienia.html*) – zawierającą przyciski odpowiedzialne za wybór poziomów trudności
   3. Nowa gra (*gra.html*) – zawierająca wygenerowaną planszę oraz obsługującą działanie gry
   4. Podziękowania (*koniec.html*) – zawierającą podziękowania dla wygranego
2. Podstrona z menu zawiera:
   1. Przycisk „Nowa gra” uruchamia grę z domyślnymi lub wybranymi przez użytkownika parametrami ustawień. Domyślnie ustawiony jest poziom łatwy i ilość pól równa 5 (plansza 5x5 pól).
   2. Przycisk „Ustawienia gry” uruchamia podstronę zawierającą możliwość wybrania ustawień gry wg poziomu zaawansowania gracza.
      1. Poziomy gry:
         1. Łatwy – pozwala na wybranie planszy 5x5, 8x8, 12x12 oraz 15x15 pól, zawiera ilość bomb wyrażoną równaniem: ilość\_bomb=0.1\*ilość\_pól zaokrągloną w dół do całości, czyli dla ilości pól równej 25 (plansza 5x5) zostanie wczytane 7 bomb (ilość\_bomb=0.1\*25=2,5, jako iż zaokrąglamy w dół zostaną 2 bomby)
         2. Średni – pozwala na wybranie planszy 4x4, 8x8, 10x10 oraz 13x13 pól, zawiera ilość bomb wyrażoną równaniem: ilość bomb=0.2\*ilość\_pól zaokrągloną w dół do całości, czyli dla ilości pól równej 64 (plansza 8x8) zostanie wczytane 12 bomb.
         3. Trudny - pozwala na wybranie planszy 4x4, 9x9, 12x12 oraz 15x15 pól, zawiera ilość bomb wyrażoną równaniem: ilość bomb=0.35\*ilość\_pól zaokrągloną w dół do całości, czyli dla ilości pól równej 81 (plansza 9x9) zostanie wczytane 28 bomb.
         4. Niemożliwy - pozwala na wybranie planszy 4x4, 9x9, 12x12 oraz 14x14 pól, zawiera ilość bomb wyrażoną równaniem: ilość bomb=0.4\*ilość\_pól zaokrągloną w dół do całości, czyli dla ilości pól równej 144 (plansza 12x12) zostanie wczytane 57 bomb.
   3. Przycisk „Zakończ”, który zamyka kartę *menu.html.*
3. Podstrona z ustawieniami zawiera:
   1. Nagłówki informujące użytkownika o poziomie trudności („Łatwy”, „Średni”, „Trudny”, „Niemożliwy”) oraz przyciski zawierające ilości pól (np. 4x4 ,8x8, 12x12 itd.)
   2. Przycisk „Zapisz”– zapisuje on ustawienia wybrane przez użytkownika i przenosi do menu
   3. Przycisk „Powrót” – powraca do menu, bez zmiany ustawień
4. Podstrona z grą zawiera:
   1. Planszę składającą się z pól typu div, które generują się automatycznie w ilości domyślnej lub wybranej przez użytkownika w ustawieniach
      1. Plansza zawiera określoną ilość bomb, które są losowo ustawiane na planszy.
   2. Stoper – liczący czas rozgrywki.
   3. Licznik rozbrojonych bomb – powiększający się w momencie kiedy gracz prawidłowo rozbroi bombę.
5. Podstrona z podziękowaniami zawiera:
   1. Tekst, informujący użytkownika o zakończonej rozgrywce. Zawiera podziękowania za grę.
   2. Przycisk „Kliknij, aby powrócić do menu!”, który po kliknięciu przenosi użytkownika do podstrony *menu.html*

# Opis funkcjonalności

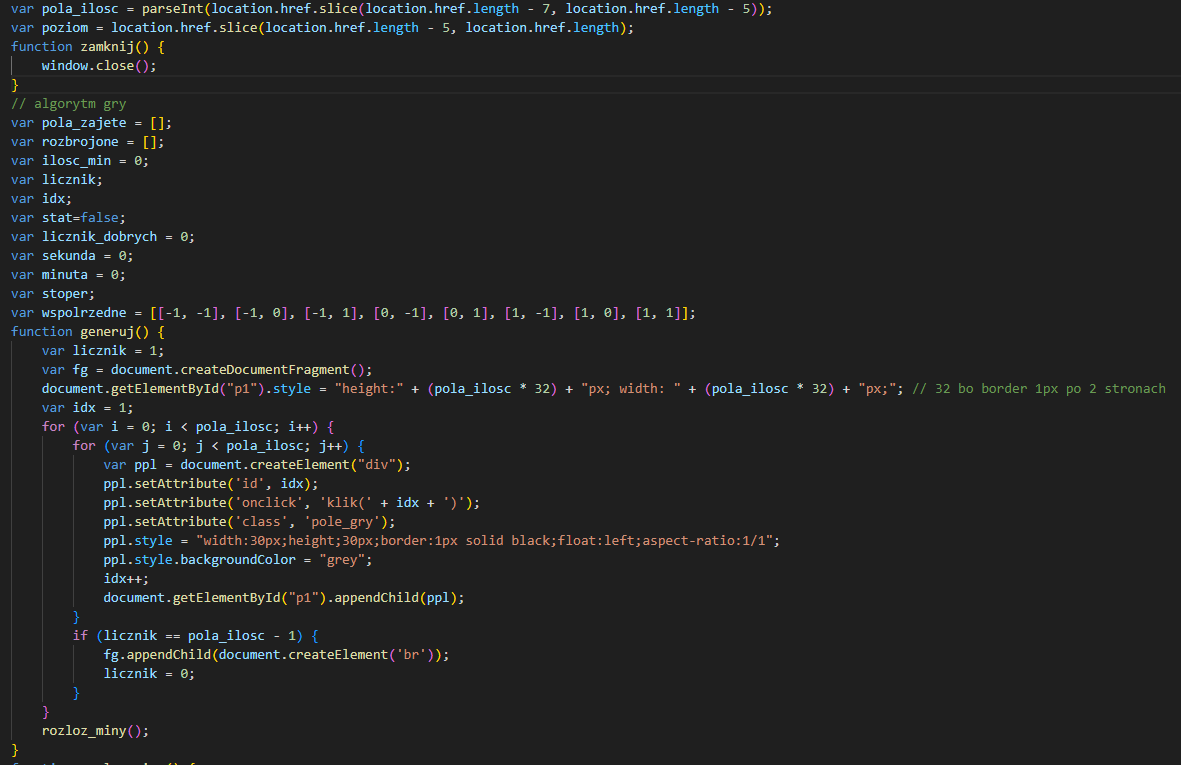
Rysunek 1

Plik „menu.js”, w którym przekazywane są zmienne „pola\_ilosc” oraz „poziom” jeśli gracz zmienił ustawienia, jeżeli nie zostały poczynione zmiany ustawiana jest domyślna wartość dla ilości pól=5 a poziom na łatwy. Funkcja przeslij() przekazuje zmienne „pola\_ilosc” oraz „poziom” do pliku „gra.html”. Funkcja zamknij() zamyka okno „menu.html”



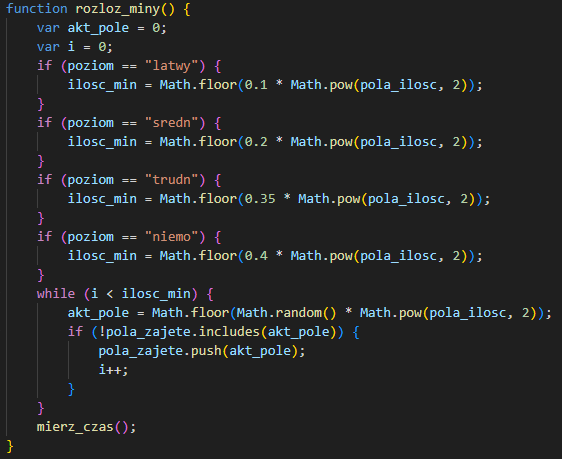
Rysunek 2

Plik „ustawienia.js”. W funkcji pola(ilość,trudność) nadpisywana jest ilość pol i poziom do zmiennych globalnych. Funkcja zapisz odpowiada za przekazanie zmiennych „pola\_ilosc” oraz „poziom” z powrotem do menu głównego gry.



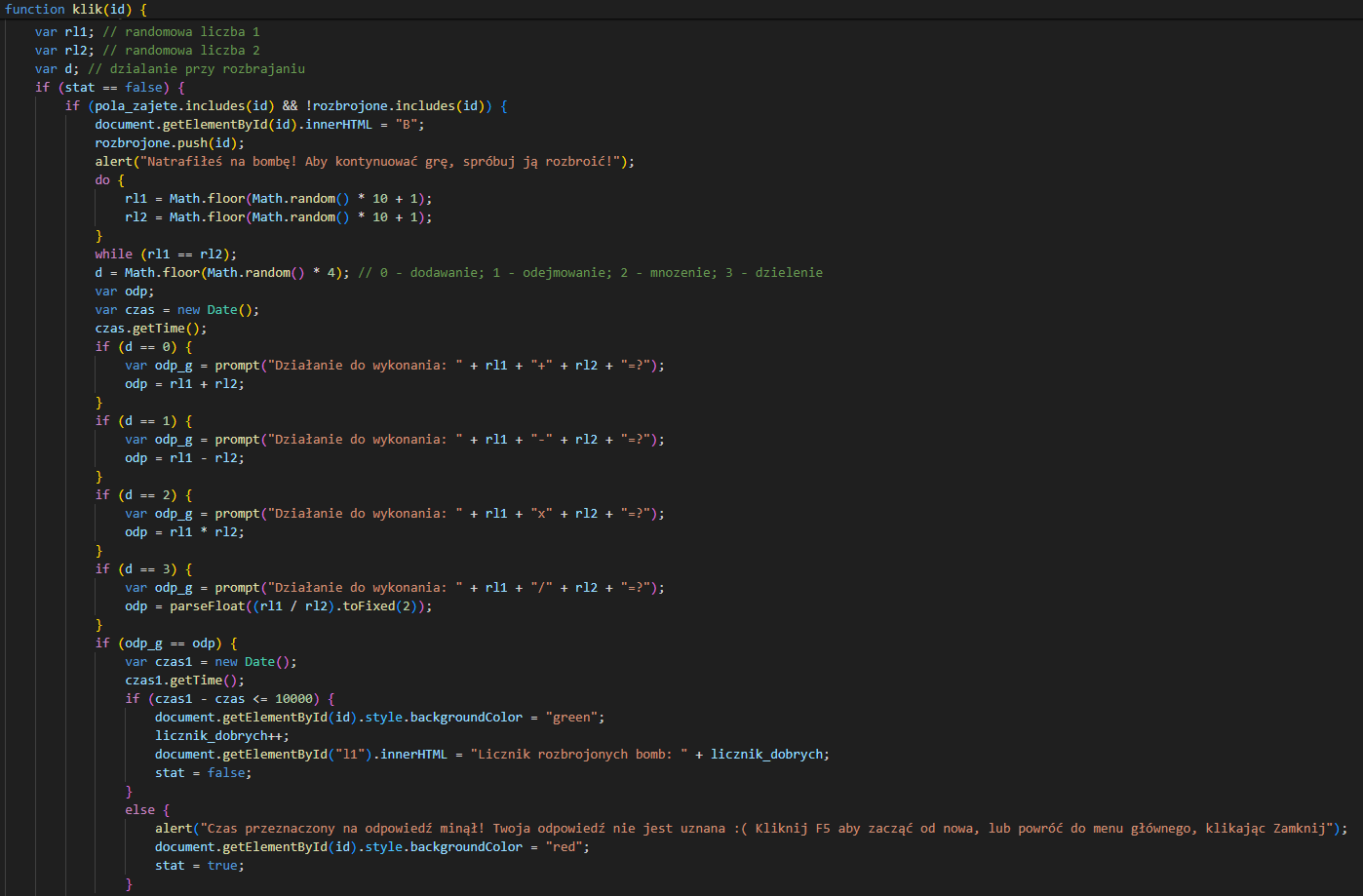
Rysunek 3

Plik „gra.js” zawiera zmienne globalne tj. „pola\_ilosc”,”poziom”, „pola\_zajete”, „rozbrojone”, „ilość\_min”, „licznik”, „idx”, „stat”, „licznik\_dobrych”, „sekunda”, „minuta”, „stoper”, „wsporzedne”. Funkcja generuj() generuje plansze o ilości pól wybranych przez użytkownika.



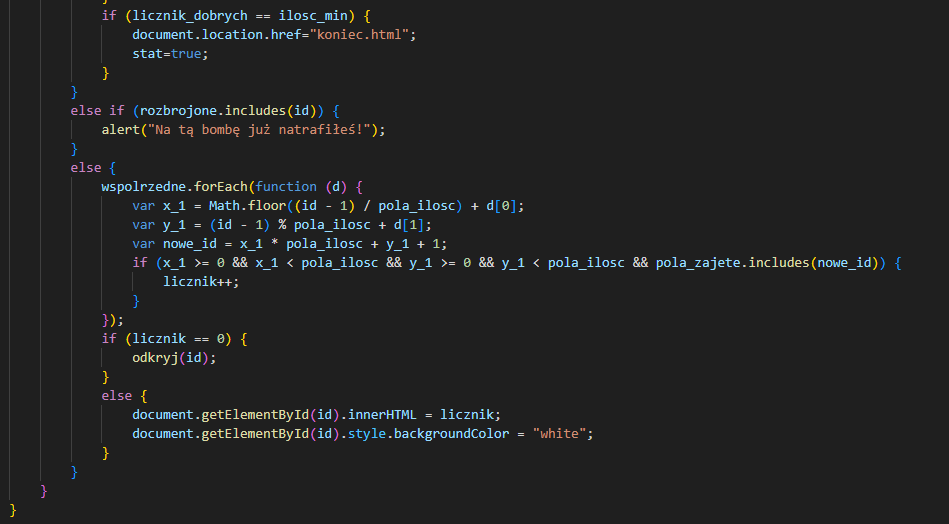
Rysunek 4

Plik „gra.js” zawiera funkcję rozloz\_miny(), która zaleznie od poziomu trudności wybranego przez gracza wykonuje działanie, którego wynik jest równy ilości min na planszy, następnie je rozmieszcza. Funkcja ta wywoływana jest przez funkcję generuj().



Rysunek 5

Plik „gra.js”, funkcja klik((id) 1/2



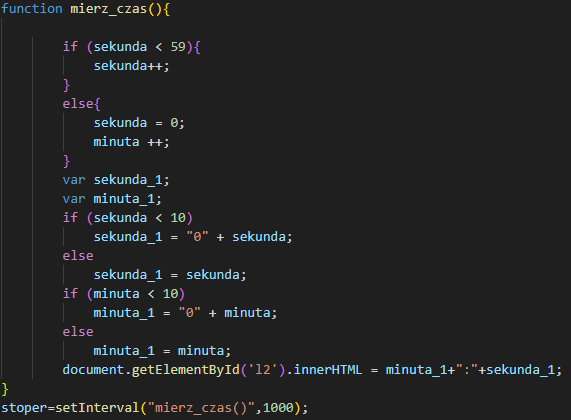
Rysunek 6

Plik „gra.js”, funkcja klik(id) 2/2. Funkcja ta obsługuje moment natrafienia przez gracza na bombę i możliwość jej rozbrojenia, lub jej brak w sąsiednich polach. Gdy nie ma bomb w sąsiednich polach funkcja wywołuje funkcję odkryj(). Jeżeli gracz nie zdoła rozbroić bomby po natrafieniu na nią, algorytm ten blokuje mu możliwość dalszej rozgrywki.



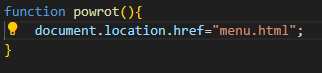
Rysunek 7

Plik „gra.js”, funkcja odkryj(id), która wywoływana jest przez funkcję klik(id), obsługuje ona odsłanianie pól dopóki algorytm nie natrafi na pole, w sąsiedztwie którego znajduję się jedna lub więcej bomb.



Rysunek 8

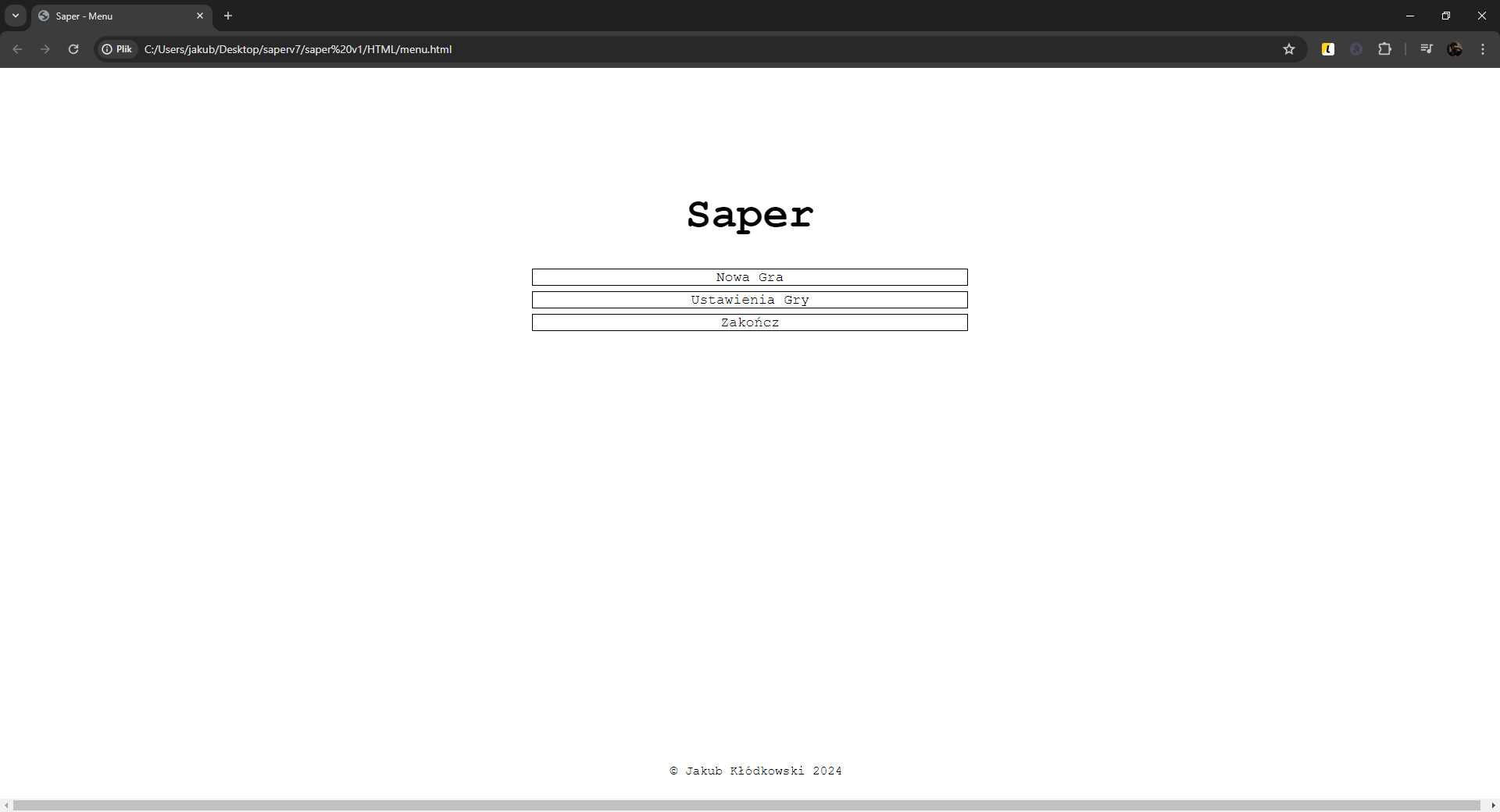
Plik „gra.js”, funkcja mierz\_czas(), która mierzy czas gry.



Rysunek 9

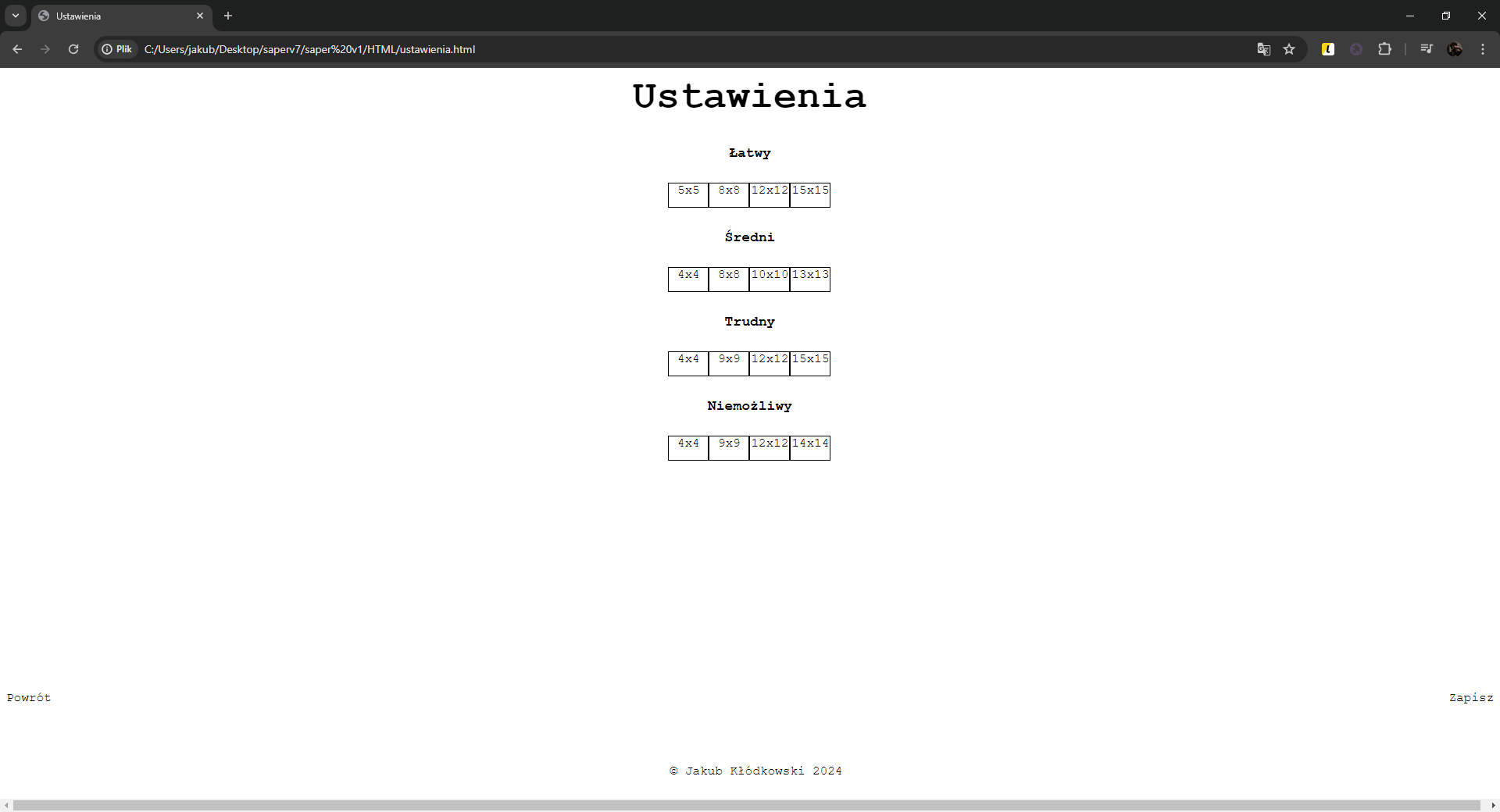
Plik „koniec.js”, funkcja powrot(), która obsługuje przeniesienie gracza z powrotem do podstrony menu.html

# Instrukcja obsługi gry



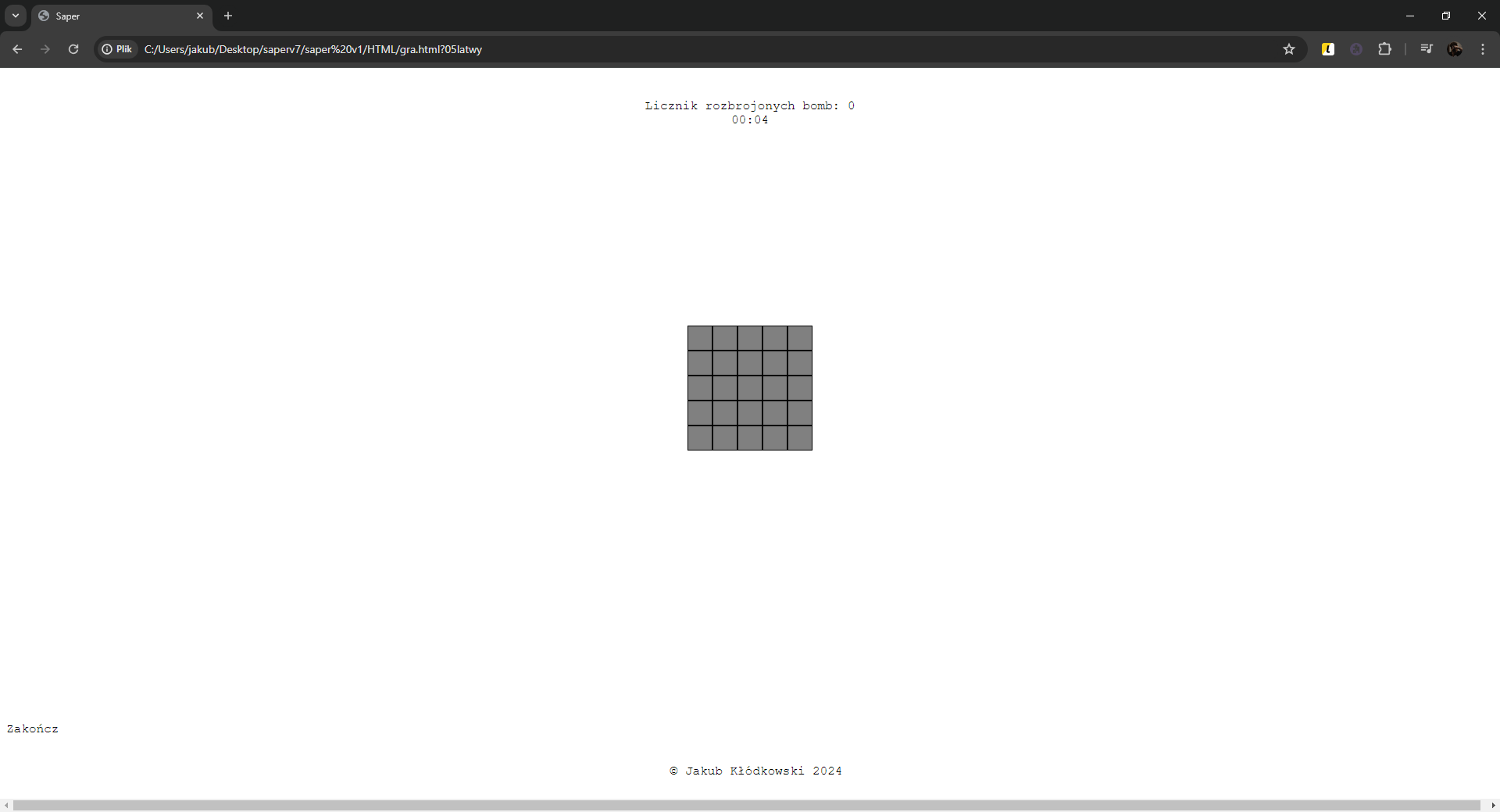
Rysunek 10

Jesteś w menu głównym gry. Po naciśnięciu przycisku „Nowa gra” uruchomi się nowa gra z domyślnymi ustawieniami (plansza: 5x5, poziom trudności: łatwy). Po kliknięciu przycisku „Ustawienia Gry” będziesz miał możliwość zmiany ustawień wedle własnych preferencji. Przycisk „Zakończ” zamyka okno przeglądarki z grą.



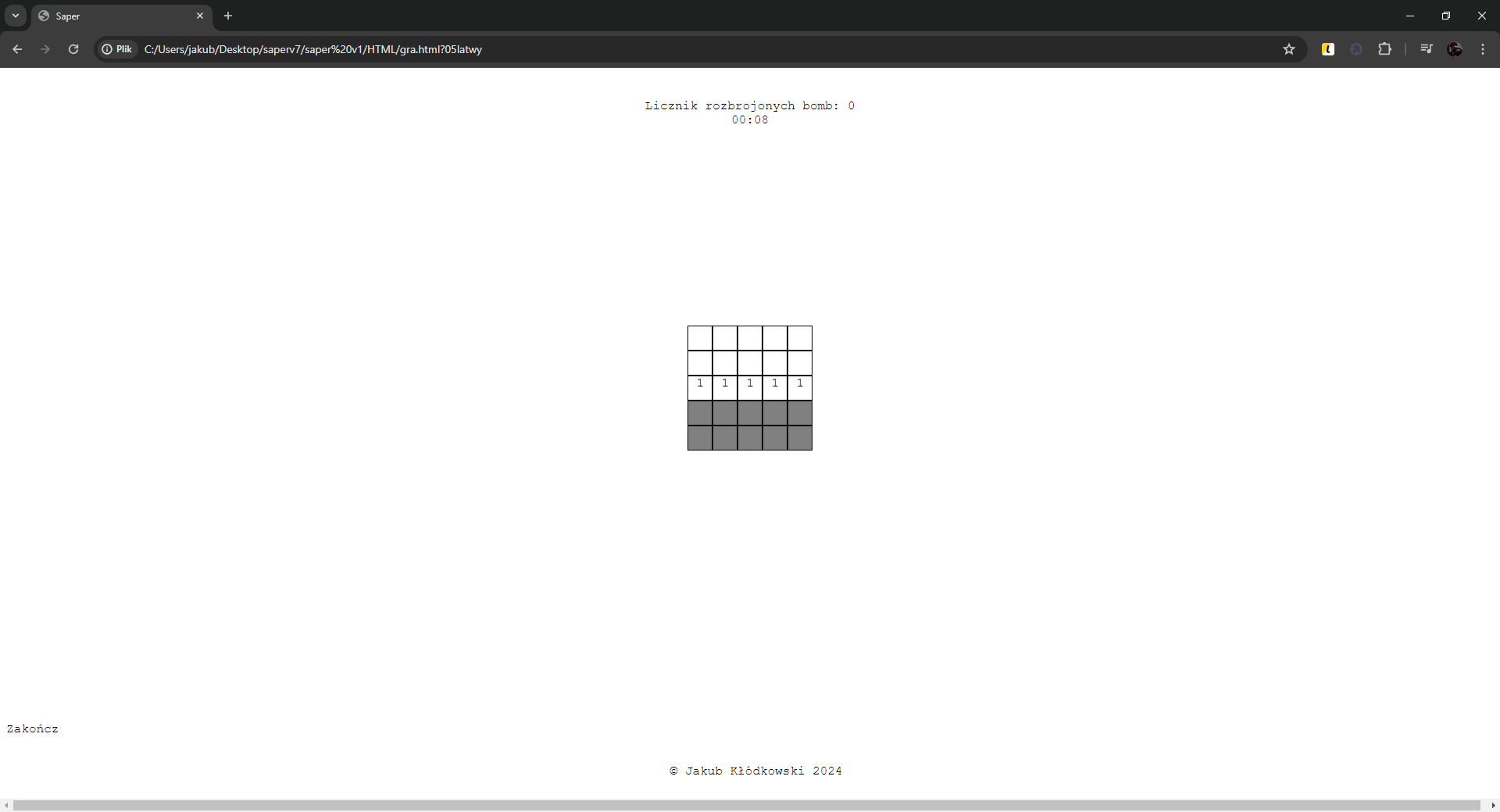
Rysunek 11

Znajdujesz się w ustawieniach gry. W tym miejscu masz możliwość zmiany ustawień, czyli wybór poziomu spośród łatwego, średniego, trudnego oraz niemożliwego. Dla danego poziomu trudności wybierasz ilość pól wedle własnych preferencji. Jeśli nie chcesz zmieniać ustawień po kliknięciu przycisku „Powrót” zostaniesz przeniesiony z powrotem do menu. Natomiast po kliknięciu przycisku „Zapisz” zostaną zapisane ustawienia wybrane przez Ciebie.



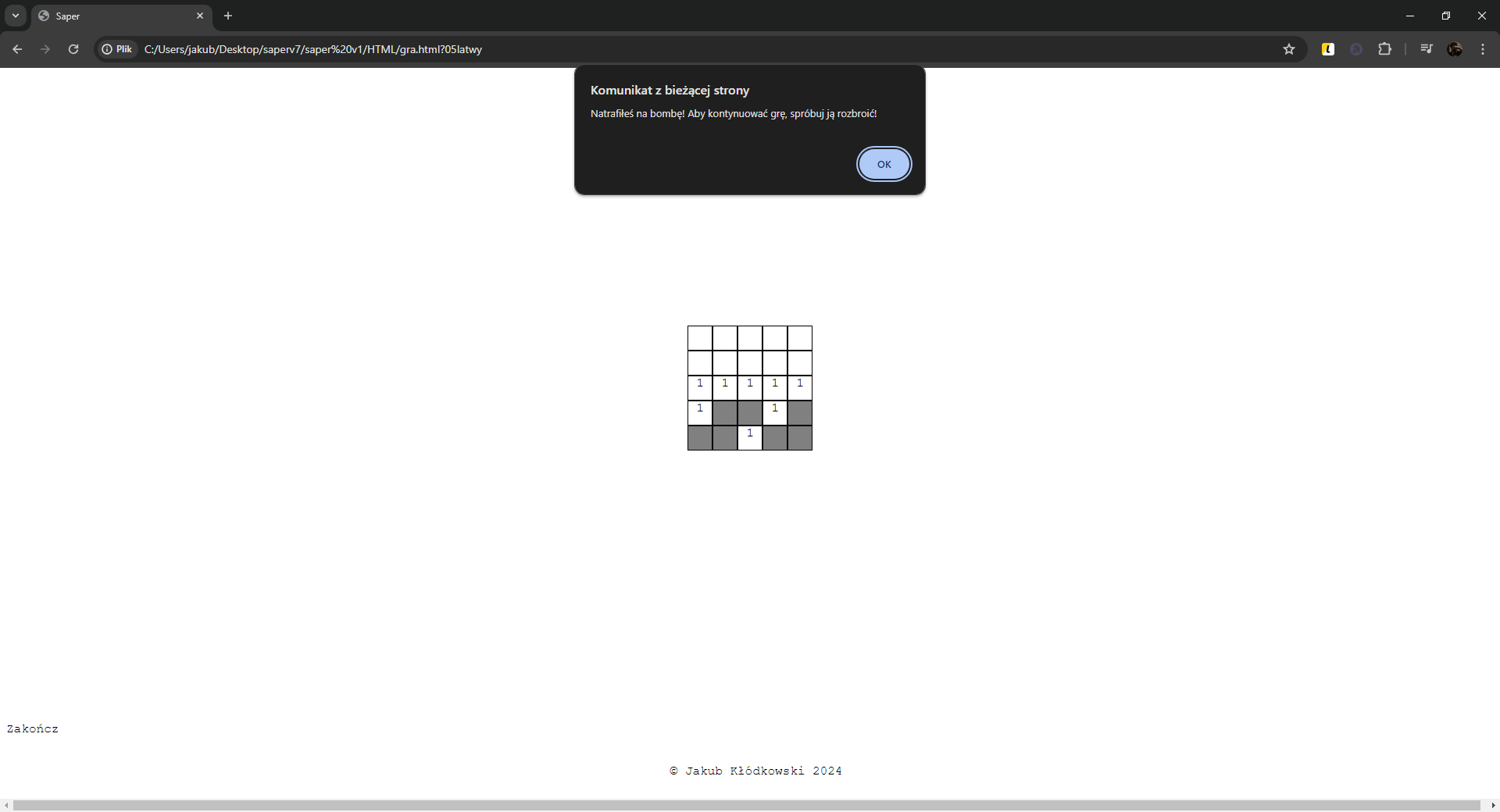
Rysunek 12

Po kliknięciu w menu głównym przycisku „Nowa Gra” zostaniesz przekierowany na stronę z grą, gdzie zostanie załadowana plansza wedle preferencji i poziomu trudności. Rozpocznie się również liczenie czasu rozgrywki oraz uruchomi się licznik rozbrojonych bomb. W każdej chwili możesz zakończyć rozgrywkę klikając przycisk „Zakończ”.



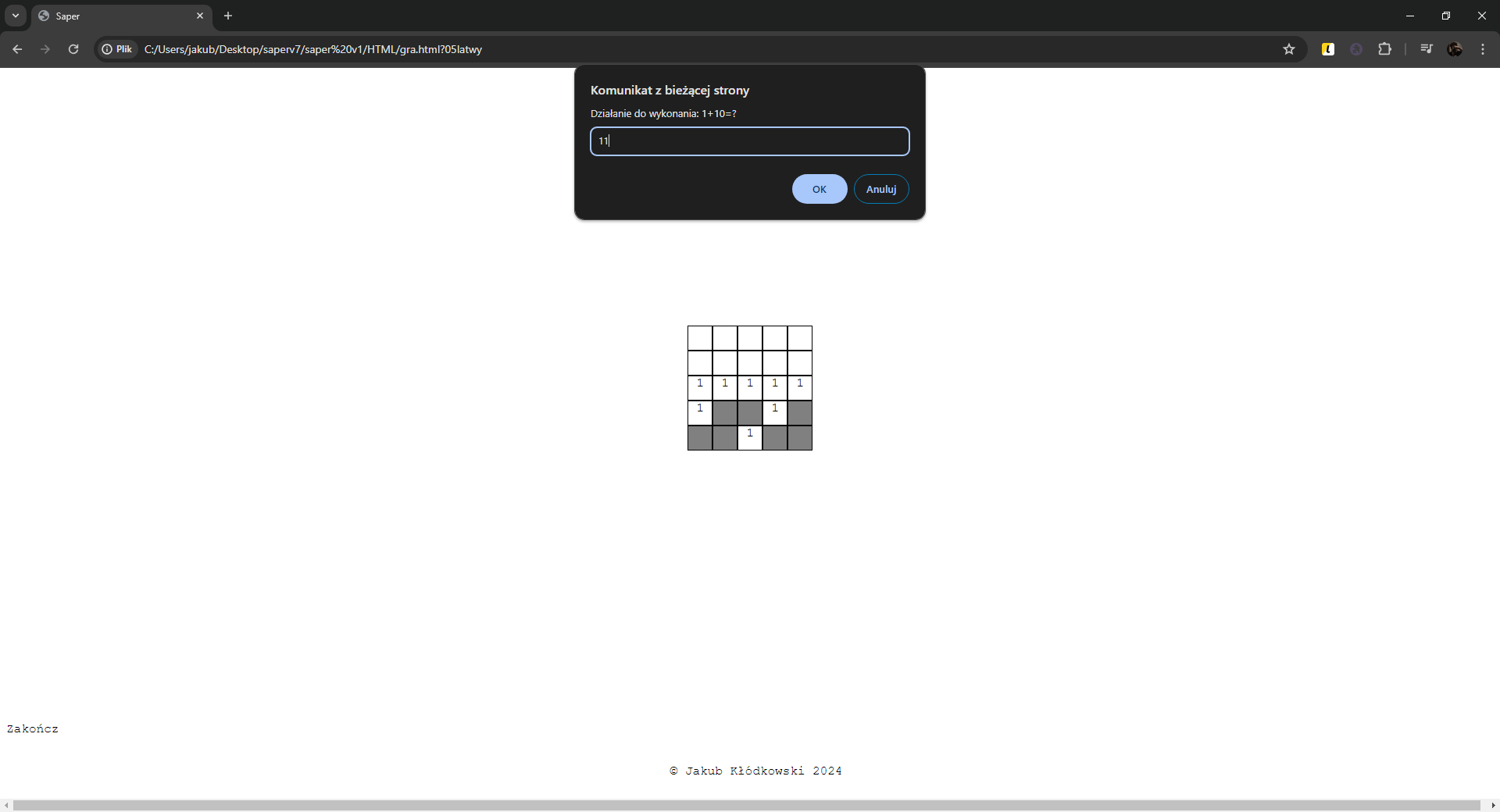
Rysunek 13

Po kliknięciu na pole, w obrębie którego nie znajduję się bomba zostanie odsłonięta reszta takich pól.



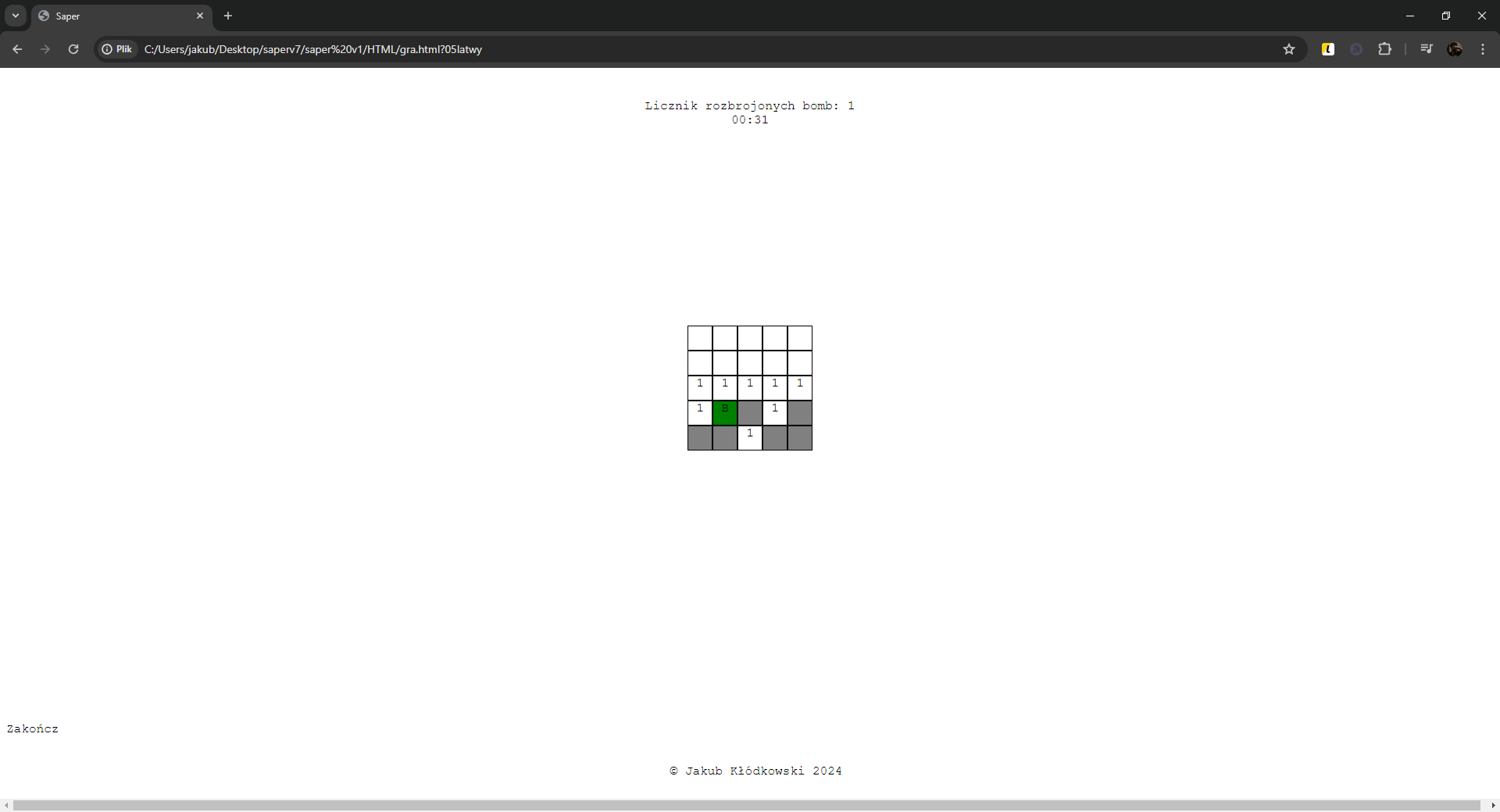
Rysunek 14

Po kliknięciu na pole, na którym znajduje się bomba pojawi się komunikat o natrafieniu na bombę i możliwości jej rozbrojenia.



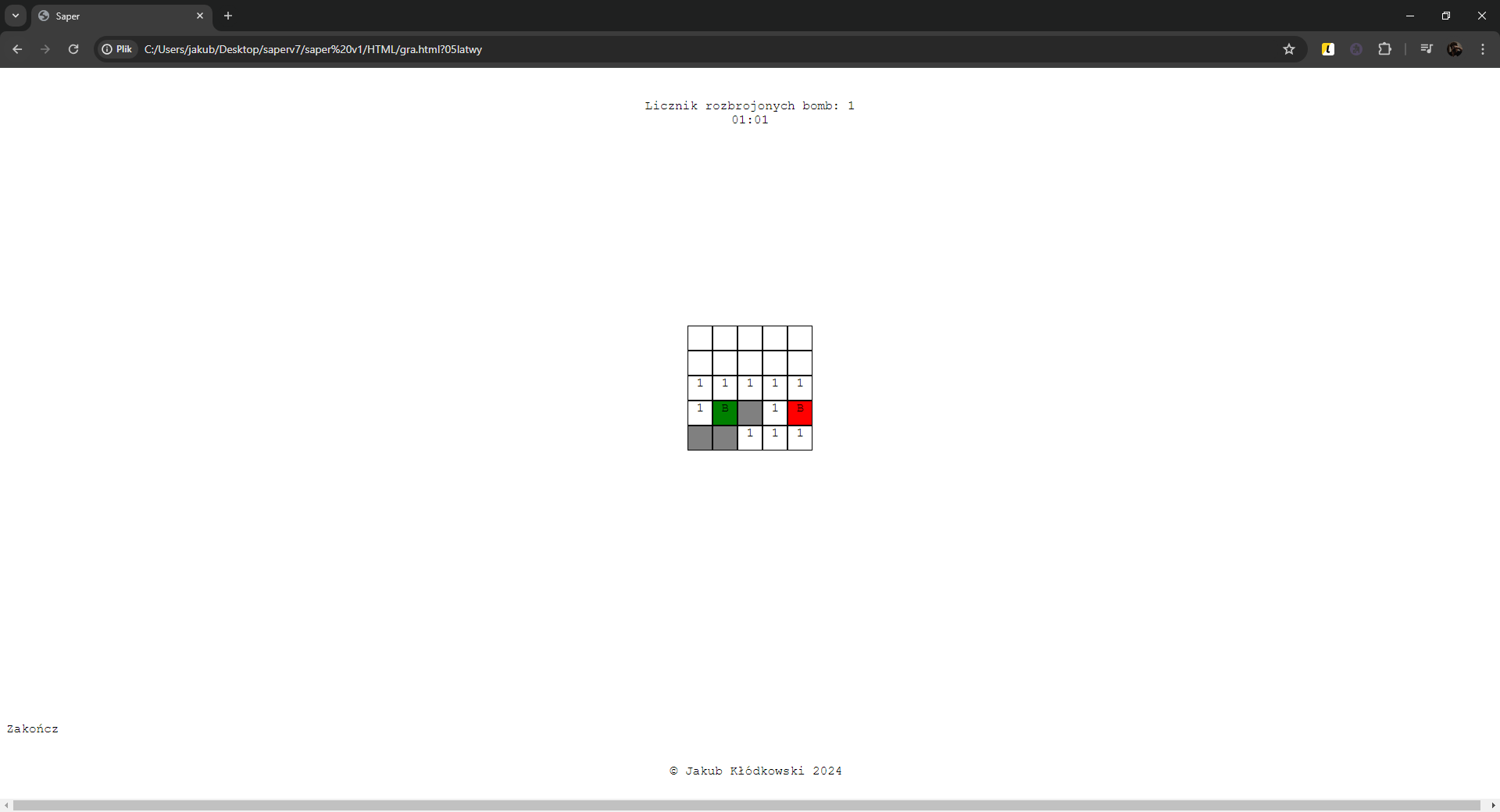
Rysunek 15

Jeśli podejmujesz próbę rozbrojenia bomby, pamiętaj masz na to tylko 10 sekund! To stosunkowo mało czasu. Jeśli znasz prawidłową odpowiedź wpisz ją w pole u góry ekranu.



Rysunek 16

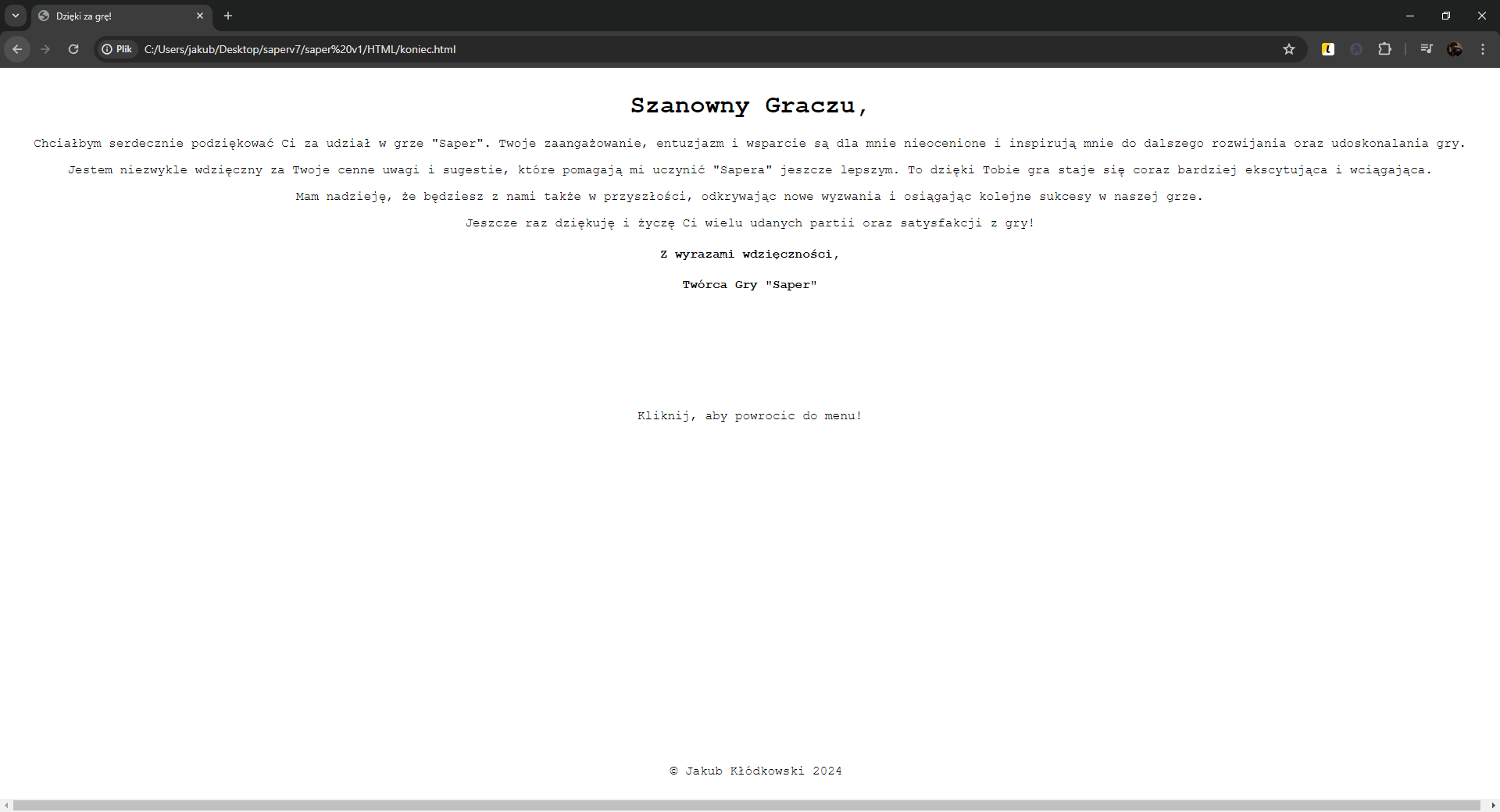
Jeśli twoja odpowiedź okaże się prawidłowa bomba została rozbrojona i jej kolor to zielony.



Rysunek 17

Jeśli natomiast nie uda Ci się rozbroić bomby, zmieni ona kolor na czerwony i spowoduje przegranie przez Ciebie rozgrywki o czym poinformuje Cię komunikat.

Co, jeśli prawidłowo rozbroisz wszystkie bomby na planszy?



Rysunek 18

Jeśli wszystkie bomby zostaną prawidłowo rozbrojone przez Ciebie oznacza to że nie ma lepszego od Ciebie :). Po wygraniu gry w Sapera zostaną wyświetlone podziękowania z możliwością powrotu do menu głównego i rozpoczęcia kolejnej rozgrywki przepełnionej emocjami.

# Kod źródłowy gry

Pole 1

// menu - js

var pola\_ilosc='';

var poziom='';

var myWindow = window.open('', '\_self');

if(location.href.slice(location.href.length - 8, location.href.length-5)!='enu' && location.href.slice(location.href.length - 5, location.href.length)!='.html'){

pola\_ilosc=location.href.slice(location.href.length - 8, location.href.length-5);

poziom= location.href.slice(location.href.length - 5, location.href.length);

}

else{

pola\_ilosc='05';

poziom='latwy';

}

function przeslij(){

document.location.href = "gra.html?"+pola\_ilosc+poziom;

}

function zamknij(){

myWindow.close();

}

Pole 3

Pole 3

// skrypt ustawien

var pola\_ilosc='';

var poziom='';

function pola(ilosc,trudnosc){

pola\_ilosc=ilosc;

poziom=trudnosc;

}

function zapisz(){

document.location.href = "menu.html?" + pola\_ilosc + poziom;

}

Pole 2

Pole 2

Pole 2

// style - menu

$("body").css({"font-family":"Monaco, Courier New, monospace","text-align":"center"});

$(".menu-napis").css({"font-size":"55px"});

$(".menu").css({"height":"300px","width":"250px",});

$(".menu").css({"text-align":"center","display":"block","margin":"auto","margin-top":"110px","width":"30%","border":"1px","padding":"10px"});

$(".menu-przycisk").css({"font-size":"18px", "cursor":"pointer"});

$(".menu-przycisk").css({"border":"1px solid black","margin":"7px"});

$("a").css({"text-decoration":"none","color":"black"});

$(".zamknij").css({"cursor":"pointer"});

$(".foot").css({"position":"absolute","bottom":"10px","text-align":"center","width":"100%"});

Pole 3

//style - ustawienia

$(".ustawienia-napis").css({"font-size":"50px","text-align":"center","margin-top":"2px"});

$(".napis-trudnosc").css({"text-align":"center","font-size":"18px"});

$(".ustaw\_pole\_wyboru").css({"display":"flex","flex-direction":"row","margin":"auto","width":"11%","padding":"10px"})

$(".wybor\_pol").css({"width":"50px","height":"30px","border":"1px solid black","text-align":"center","justify-content":"space-around", "cursor":"pointer"});

$(".zapisz").css({"float":"right","height":"100px", "margin-top":"30vh", "cursor":"pointer"});

$(".powrot").css({"float":"left","height":"100px", "margin-top":"30vh", "cursor":"pointer"});

// style - gra

$(".zakoncz").css({"position":"absolute","float":"left",'bottom':"80px"});

$(".pole\_gry").css({"height":"30px","width":"30px","border":"1px solid black","float":"left"});

$(".srodek").css({"height":"70vh","width":"100%","display":"flex","align-items":"center","justify-content":"center"})

$(".liczniki").css({"margin-top":"40px"});

// style - podziekowania

$(".podziekowania").css({"margin-top":"30px"});

$(".menu-powrot").css({"margin-top":"150px","cursor":"pointer"});

Pole 4

var pola\_ilosc = parseInt(location.href.slice(location.href.length - 7, location.href.length - 5));

var poziom = location.href.slice(location.href.length - 5, location.href.length);

function zamknij() {

window.close();

}

// algorytm gry

var pola\_zajete = [];

var rozbrojone = [];

var ilosc\_min = 0;

var licznik;

var idx;

var stat=false;

var licznik\_dobrych = 0;

var sekunda = 0;

var minuta = 0;

var stoper;

var wspolrzedne = [[-1, -1], [-1, 0], [-1, 1], [0, -1], [0, 1], [1, -1], [1, 0], [1, 1]];

Pole 5

function generuj() {

var licznik = 1;

var fg = document.createDocumentFragment();

document.getElementById("p1").style = "height:" + (pola\_ilosc \* 32) + "px; width: " + (pola\_ilosc \* 32) + "px;"; // 32 bo border 1px po 2 stronach

var idx = 1;

for (var i = 0; i < pola\_ilosc; i++) {

for (var j = 0; j < pola\_ilosc; j++) {

var ppl = document.createElement("div");

ppl.setAttribute('id', idx);

ppl.setAttribute('onclick', 'klik(' + idx + ')');

ppl.setAttribute('class', 'pole\_gry');

ppl.style = "width:30px;height;30px;border:1px solid black;float:left;aspect-ratio:1/1";

ppl.style.backgroundColor = "grey";

idx++;

document.getElementById("p1").appendChild(ppl);

}

if (licznik == pola\_ilosc - 1) {

fg.appendChild(document.createElement('br'));

licznik = 0;

}

}

rozloz\_miny();

}

Pole 6

function rozloz\_miny() {

var akt\_pole = 0;

var i = 0;

if (poziom == "latwy") {

ilosc\_min = Math.floor(0.1 \* Math.pow(pola\_ilosc, 2));

}

if (poziom == "sredn") {

ilosc\_min = Math.floor(0.2 \* Math.pow(pola\_ilosc, 2));

}

if (poziom == "trudn") {

ilosc\_min = Math.floor(0.35 \* Math.pow(pola\_ilosc, 2));

}

if (poziom == "niemo") {

ilosc\_min = Math.floor(0.4 \* Math.pow(pola\_ilosc, 2));

}

while (i < ilosc\_min) {

akt\_pole = Math.floor(Math.random() \* Math.pow(pola\_ilosc, 2));

if (!pola\_zajete.includes(akt\_pole)) {

pola\_zajete.push(akt\_pole);

i++;

}

}

mierz\_czas();

}

Pole 7

function klik(id) {

idx = id;

licznik = 0;

var rl1; // randomowa liczba 1

var rl2; // randomowa liczba 2

var d; // dzialanie przy rozbrajaniu

if (stat == false) {

if (pola\_zajete.includes(id) && !rozbrojone.includes(id)) {

document.getElementById(id).innerHTML = "B";

rozbrojone.push(id);

alert("Natrafiłeś na bombę! Aby kontynuować grę, spróbuj ją rozbroić!");

do {

rl1 = Math.floor(Math.random() \* 10 + 1);

rl2 = Math.floor(Math.random() \* 10 + 1);

}

while (rl1 == rl2);

d = Math.floor(Math.random() \* 4); // 0 - dodawanie; 1 - odejmowanie; 2 - mnozenie; 3 - dzielenie

var odp;

var czas = new Date();

czas.getTime();

if (d == 0) {

var odp\_g = prompt("Działanie do wykonania: " + rl1 + "+" + rl2 + "=?");

odp = rl1 + rl2;

}

if (d == 1) {

var odp\_g = prompt("Działanie do wykonania: " + rl1 + "-" + rl2 + "=?");

odp = rl1 - rl2;

}

Pole 8

if (d == 2) {

var odp\_g = prompt("Działanie do wykonania: " + rl1 + "x" + rl2 + "=?");

odp = rl1 \* rl2;

}

if (d == 3) {

var odp\_g = prompt("Działanie do wykonania: " + rl1 + "/" + rl2 + "=?");

odp = parseFloat((rl1 / rl2).toFixed(2));

}

if (odp\_g == odp) {

var czas1 = new Date();

czas1.getTime();

if (czas1 - czas <= 10000) {

document.getElementById(id).style.backgroundColor = "green";

licznik\_dobrych++;

document.getElementById("l1").innerHTML = "Licznik rozbrojonych bomb: " + licznik\_dobrych;

stat = false;

}

else {

alert("Czas przeznaczony na odpowiedź minął! Twoja odpowiedź nie jest uznana :( Kliknij F5 aby zacząć od nowa, lub powróć do menu głównego, klikając Zamknij");

document.getElementById(id).style.backgroundColor = "red";

stat = true;

}

}

else {

document.getElementById(id).style.backgroundColor = "red";

alert("Twoja odpowiedź jest błędna:( Kliknij F5 aby zacząć od nowa, lub powróć do menu głównego, klikając Zamknij");

stat = true;

}

Pole 9

if (licznik\_dobrych == ilosc\_min) {

document.location.href="koniec.html";

stat=true;

}

}

else if (rozbrojone.includes(id)) {

alert("Na tą bombę już natrafiłeś!");

}

else {

wspolrzedne.forEach(function (d) {

var x\_1 = Math.floor((id - 1) / pola\_ilosc) + d[0];

var y\_1 = (id - 1) % pola\_ilosc + d[1];

var nowe\_id = x\_1 \* pola\_ilosc + y\_1 + 1;

if (x\_1 >= 0 && x\_1 < pola\_ilosc && y\_1 >= 0 && y\_1 < pola\_ilosc && pola\_zajete.includes(nowe\_id)) {

licznik++;

}

});

if (licznik == 0) {

odkryj(id);

}

else {

document.getElementById(id).innerHTML = licznik;

document.getElementById(id).style.backgroundColor = "white";

}

}

}

}

Pole 10

function odkryj(id) {

var pola = [id];

while (pola.length > 0) {

var akt\_id = pola.pop();

var licznik\_1 = 0;

wspolrzedne.forEach(function (d) {

var x\_1 = Math.floor((akt\_id - 1) / pola\_ilosc) + d[0];

var y\_1 = (akt\_id - 1) % pola\_ilosc + d[1];

var nowe\_id = x\_1 \* pola\_ilosc + y\_1 + 1;

if (x\_1 >= 0 && x\_1 < pola\_ilosc && y\_1 >= 0 && y\_1 < pola\_ilosc) {

if (pola\_zajete.includes(nowe\_id)) {

licznik\_1++;

}

}

});

document.getElementById(akt\_id).style.backgroundColor = "white";

document.getElementById(akt\_id).innerHTML = licznik\_1 > 0 ? licznik\_1 : '';

Pole 11

if (licznik\_1 == 0) {

wspolrzedne.forEach(function (d) {

var x\_1 = Math.floor((akt\_id - 1) / pola\_ilosc) + d[0];

var y\_1 = (akt\_id - 1) % pola\_ilosc + d[1];

var nowe\_id = x\_1 \* pola\_ilosc + y\_1 + 1;

if (x\_1 >= 0 && x\_1 < pola\_ilosc && y\_1 >= 0 && y\_1 < pola\_ilosc && !document.getElementById(nowe\_id).style.backgroundColor.includes("white")) {

pola.push(nowe\_id);

}

});

}

}

}

Pole 12

function mierz\_czas(){

if (sekunda < 59){

sekunda++;

}

else{

sekunda = 0;

minuta ++;

}

var sekunda\_1;

var minuta\_1;

if (sekunda < 10)

sekunda\_1 = "0" + sekunda;

else

sekunda\_1 = sekunda;

if (minuta < 10)

minuta\_1 = "0" + minuta;

else

minuta\_1 = minuta;

document.getElementById('l2').innerHTML = minuta\_1+":"+sekunda\_1;

}

stoper=setInterval("mierz\_czas()",1000);

Pole 13

function powrot(){

document.location.href="menu.html";

}

# Wnioski końcowe

Projekt semestralny „Saper” był średniozaawansowanym zadaniem. Największy problem sprawiło mi stworzenie algorytmu, który będzie odsłaniał puste pola. Na szczęście, po odrobinie poszukiwań w Internecie znalazłem rozwiązanie mojego problemu, które skrupulatnie przeanalizowałem. Jednym z przewijających się problemów było zamykanie strony z menu, gdyż przeglądarki ograniczają możliwość zamykania okien przez skrypty. Zostałem przy rozwiązaniu, które od razu po wejściu działa, lecz gdy np. zmieni się ustawienia już niestety nie. Jest to element, który wymaga dopracowania. Reszta funkcjonalności gry nie sprawiła mi większych problemów.

# Spis rysunków:

[Rysunek 1 4](#_Toc168695259)

[Rysunek 2 4](#_Toc168695260)

[Rysunek 3 5](#_Toc168695261)

[Rysunek 4 6](#_Toc168695262)

[Rysunek 5 6](#_Toc168695263)

[Rysunek 6 7](#_Toc168695264)

[Rysunek 7 7](#_Toc168695265)

[Rysunek 8 8](#_Toc168695266)

[Rysunek 9 8](#_Toc168695267)

[Rysunek 10 9](#_Toc168695268)

[Rysunek 11 9](#_Toc168695269)

[Rysunek 12 10](#_Toc168695270)

[Rysunek 13 10](#_Toc168695271)

[Rysunek 14 11](#_Toc168695272)

[Rysunek 15 11](#_Toc168695273)

[Rysunek 16 12](#_Toc168695274)

[Rysunek 17 12](#_Toc168695275)

[Rysunek 18 13](#_Toc168695276)

# Pola tekstowe:

[Pole 1 14](#_Toc168696290)

[Pole 2 15](#_Toc168696291)

[Pole 3 16](#_Toc168696292)

[Pole 4 17](#_Toc168696293)

[Pole 5 18](#_Toc168696294)

[Pole 6 19](#_Toc168696295)

[Pole 7 20](#_Toc168696296)

[Pole 8 21](#_Toc168696297)

[Pole 9 22](#_Toc168696298)

[Pole 10 23](#_Toc168696299)

[Pole 11 24](#_Toc168696300)

[Pole 12 25](#_Toc168696301)

[Pole 13 26](#_Toc168696302)