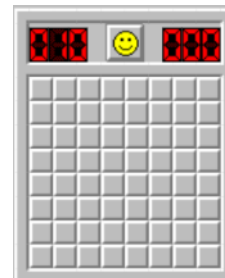


JS 11 - Saper (16)

Saper (tytuł oryginalny Minesweeper) – klasyczna jednoosobowa gra komputerowa napisana w 1981 roku przez Roberta Donnera, dostępna jako akcesorium w każdym systemie Microsoft Windows (istnieją też wersje dla innych systemów operacyjnych). Gra polega na odkrywaniu na planszy poszczególnych pól w taki sposób, aby nie natrafić na minę. Na każdym z odkrytych pól napisana jest liczba min, które bezpośrednio stykają się z danym polem (od zera do ośmiu). Jeśli gracz oznaczy dane pole flagą, jest ono zabezpieczone przed odsłonięciem, dzięki czemu przez przypadek nie odsłoni miny.



Pamiętaj o tym, by zrzut ekranu DOKUMENTOWAŁ Twoją pracę

Plansza (1)

- W swoim folderze utwórz 2 nowe dokumenty: **js11.html** i **js11.js**
- Otwórz oba dokumenty w notatniku, a dokument HTML w przeglądarce
- Ustaw oba okna na dwóch połowach monitora
- Do dokumentu **HTML** wklej tekst z ramki

```
<html>
<head>
  <meta charset=utf8>
  <title> SAPER </title>
  <script src=js11.js></script>
</head>

<body>
<center>
<font size=6>Libront Waclaw</font>
<br>
<canvas width=320 height=320 id=SAPER></canvas>
</center>

<script>
  var S=SAPER.getContext("2d");
  var SZE=S.canvas.width;
  var WYS=S.canvas.height;
  S.fillStyle="black";
  S.fillRect(0,0,SZE,WYS);
</script>

</body>
</html>
```

obszar canvas ma rozmiar 320x320 pikseli
dostęp do canvas za pomocą „uchwyty” o nazwie S
zadeklarowane dwie zmienne SZE i WYS, w których zapamiętujemy szerokość i wysokość obszaru canvas
wypełniamy canvas czarny kolorem

- Zmień tytuł strony **SAPER** na swoje **inicjały**
- Wpisz swoje **nazwisko i imię**
- Zapisz dokumenty i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)



Grafika (1)

Niezbędne grafiki znajdują się w pliku *saper.png*. Plik zawiera połączone 14 obrazków, każdy o wymiarach 40x40 pikseli. Malowanie obrazkami na obszarze canvas realizujemy za pomocą instrukcji `drawImage()`.



- Pobierz plik **saper.png** i zapisz w swoim folderze <http://zsobobowa.eu/pliki/program/js/saper.png>
- Dokument **HTML**, przed `<script>` `</script>`

```
var IMG=new Image();
IMG.src='saper.png';
IMG.onload=function () {
    S.drawImage(IMG,5,5);
}
```

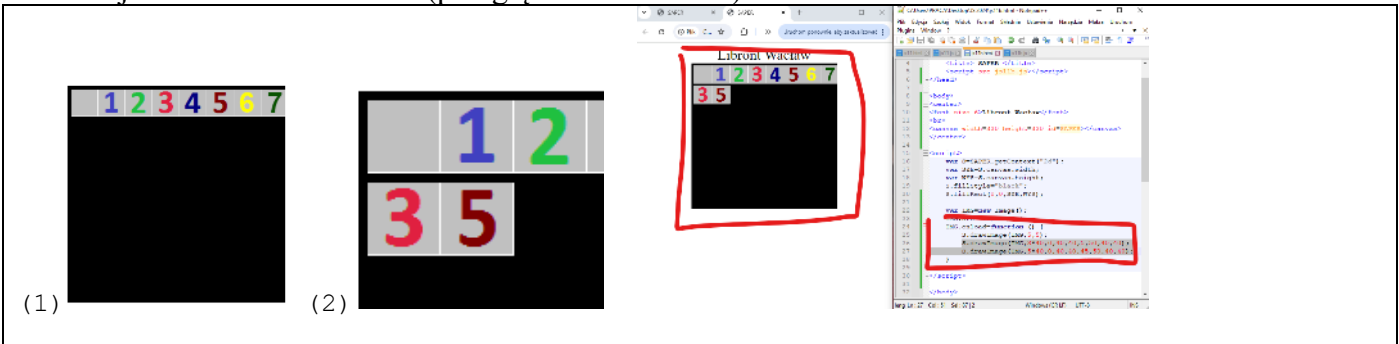
*IMG zmienna na nowy obraz
do własności src zmiennej IMG wstawiamy nazwę ładowanego obrazka
ładujemy obraz za pomocą funkcji onload
w obszarze canvas (S) wyświetlamy obraz, lewy górny róg w punkcie (5,5)*

- (1) Zapisz dokumenty i odśwież przeglądarkę
w obszarze canvas 320x320 pikseli mieści się tylko fragment

```
S.drawImage(IMG,5,5);
```

- (2) Dokument **HTML**, funkcja **onload** }
`S.drawImage(IMG,3*40,0,40,40,5,50,40,40);`
`S.drawImage(IMG,5*40,0,40,40,45,50,40,40);`
*wersja rozszerzona drawImage
z całego obrazka wycinamy fragment od lewej 3*40=120 pikseli i od góry 0 pikseli o wielkości 40x40 pikseli
i wyświetlamy na canvas w miejscu (5,50) o wielkości 40x40 pikseli
drugi obrazek - pomijamy od 200 pikseli z lewej, na canvas od 45 pikseli*

- **ZADANIE** Zmień parametry 2 ostatnich poleceń i wyświetl swój wiek w latach
- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)



Plansza (1)

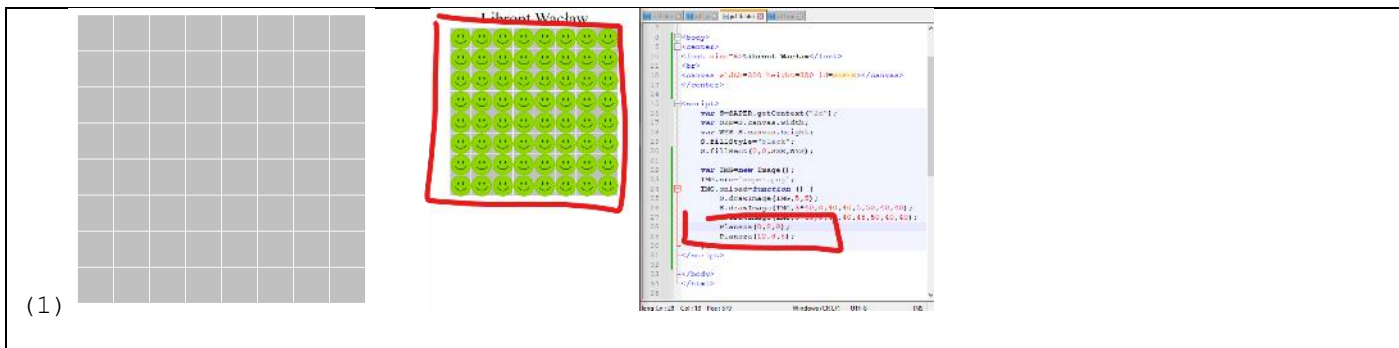
- Dokument **JS**

```
function Plansza(nr,wie,kol){
    for (var w=0;w<wie;w++)
        for (var k=0;k<kol;k++)
            S.drawImage(IMG,nr*40,0,40,40,k*40,w*40,40,40);
}
```

*wypełnianie obszaru canvas obrazkami
dwie pętle ustawiają zmienne w i k
z ich pomocą wybieramy obrazek nr i rysujemy na canvas w miejscu (w*40, k*40)*

```
S.drawImage(IMG,5*40,0,40,40,45,50,40,40);
```

- Dokument **HTML**, funkcja **onload** }
`Plansza(0,8,8);`
wypełnianie obszaru canvas obrazkiem 0 - 8 wierszy i 8 kolumn
- (1) Zapisz dokumenty i odśwież przeglądarkę
- **ZADANIE** Dopisz kolejną funkcję Plansza `Plansza([redacted]);`,
aby wydrukowane zostały zielone uśmiechy - **nr 12**
- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)



Myszka i klawiatura (1)

Grę obsługujemy głównie za pomocą myszki. Obsługa klawiatury dotyczy klawisza SHIFT, który pozwoli ustawiać „flagi”

- Dokument HTML, przed `</center>`

```
<canvas width=320 />
</center>
```

```
<br>
<label id=idMYSZK</label>
<label id=idKLAW</label>
```

pola, w których będą pojawiły się informacje o położeniu myszki i wciśniętych klawiszach

- Dokument HTML, przed `</script>`

```
var MYSZKA={x:0,y:0};
var SHIFT=false;
Zdarzenia();
```

zmienna MYSZKA zapamiętuje dwie współrzędne kliknięcia myszką w canvas
zmienna SHIFT zapamiętuje wciśnięcie klawisza SHIFT
funkcja Zdarzenia() będzie obsługiwać zdarzenia związane z myszką i klawiaturą

- Dokument JS

```
function KlawiszON(e){
  if (e.key=="Shift") SHIFT=true;
  idKLAW.innerHTML=e.key;
}
function KlawiszOFF(e){
  SHIFT=false;
  idKLAW.innerHTML="...";
}
function Myszka(){
  idMYSZK.innerHTML="("+MYSZKA.x+" "+MYSZKA.y+")";
}
function Zdarzenia(){
  S.canvas.onmousemove=function(e){
    MYSZKA.x=e.offsetX;
    MYSZKA.y=e.offsetY;
    Myszka();
  };
  S.canvas.onclick=function(e){
    MYSZKA.x=e.offsetX;
    MYSZKA.y=e.offsetY;
  }
  document.addEventListener("keydown", KlawiszON, false);
  document.addEventListener("keyup", KlawiszOFF, false);
}
```

funkcja KlawiszON - wciskamy klawisze, jeżeli to SHIFT - ustawiamy zmienną i wyświetlamy w polu idKLAW
funkcja KlawiszOFF - puszczamy klawisz, - ustawiamy zmienną SHIFT i wyświetlamy kropki w polu idKLAW
funkcja Myszka - współrzędne kliknięcia zapisane w MYSZKA wyświetlamy w polu id MYSZ
funkcja Zdarzenia

- pobieramy współrzędne położenia onmousemove myszki i zapisujemy w zmiennej MYSZKA
- pobieramy współrzędne kliknięcia onclick
- ustawiamy funkcje odpowiedzialne za obsługę klawiatury

- Zapisz dokumenty i odśwież przeglądarkę
- Wciskaj klawisze
- (1) Ustaw wskaźnik myszki w punkcie (100,100) myszką

- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)



Tablice (1)

Nasz saper będzie miał możliwość ustawiania wersji rozgrywki w zależności od stopnia opanowania - liczba losowanych min, wielkość planszy na trzech poziomach: podstawowy, zaawansowany, ekspert.

- Dokument **HTML**, przed `</script>` `Zdarzenia();`

```
var ILE_MIN=10;
var KOLUMNY=8;
var WIERSZE=8;
var MINY=[];
var POLA=[];
```

ILE_MIN liczba losowanych min

KOLUMNY liczba kolumn

WIERSZE liczba wierszy

MINY tablica, w której zapamiętamy położenie min i liczbę min wokół pustego pola

POLA tablica, w której zapamiętamy numer wyświetlanego obrazka na ekranie

- Dokument **JS**

```
function ZerowanieTablic(){
    var m=[];
    var p=[];
    for (var w=0;w<WIERSZE;w++){
        m[w]=[];
        p[w]=[];
        for (var k=0;k<KOLUMNY;k++){
            m[w][k]=0;
            p[w][k]=9;
        }
    }
    MINY=m;
    POLA=p;
}
```

automatyczne ustawianie tablic dynamicznych (nie ma początkowo określonej wielkości) przebiega w trzech etapach:

- tworzymy pustą tablicę `var m=[];`
- tworzymy pusty wiersz w tablicy `m[w]=[];`
- wypełniamy komórkę tablicy `m[w][k]=0;`

wszystko realizujemy w dwóch pętłach

wielkość tworzonych tablic zależy od zmiennych *WIERSZE* i *KOLUMNY*

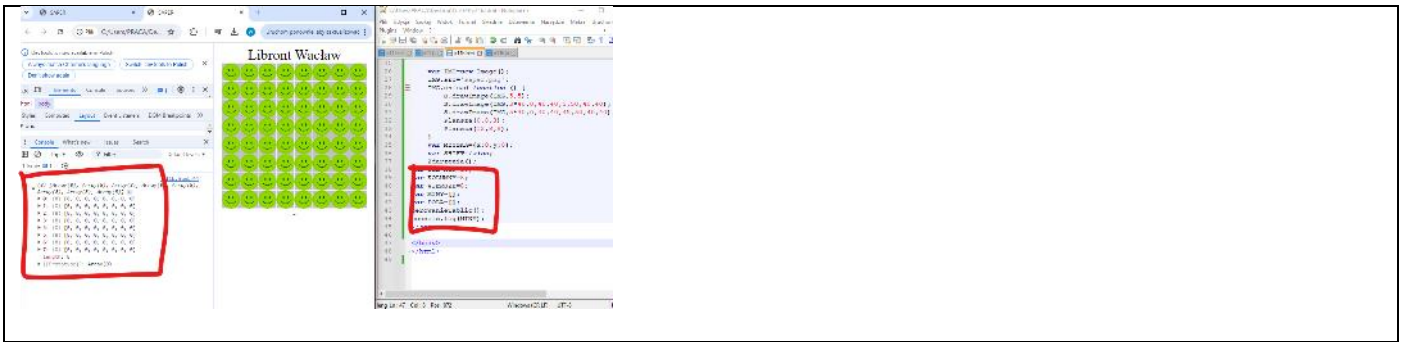
tablica *MINY* wypełniona jest zerami (brak min)

tablica *POLA* wypełniona jest dziewiątkami (nieodkryte pole 0 obrazek nr 9)

- Dokument **HTML**, przed `</script>` `Zdarzenia();`
`ZerowanieTablic();`
`console.log(MINY);`

- Kliknij prawym przyciskiem myszki w okno przeglądarki i wybierz **Zbadaj**
- Zapisz dokumenty i odśwież przeglądarkę

- W oknie konsoli „otwórz” zmienną *MINY* `(8) [Array]`
- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)



Losowanie min (1)

Zmienna `ILE_MIN` określa liczbę losowanych min.

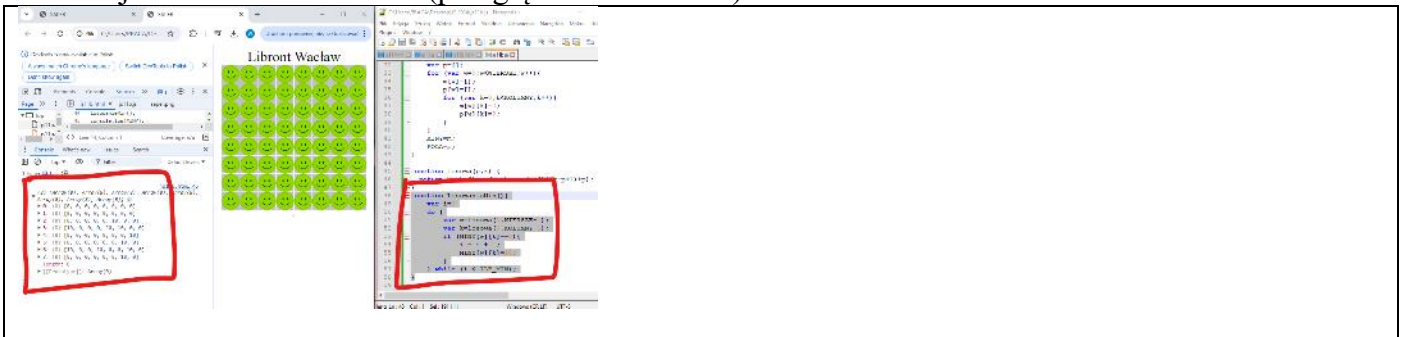
Zerowanie Tablicy :
`console.log(MINY);`

- Dokument **HTML** `LosowanieMin()` ;
funkcja losująca miny - w tablicy MINY wstawia 10 (obrazek nr 10 to mina)
- Dokument **JS**

```
function losowa(p,k) {
    return Math.floor(Math.random() * (k-p+1)+p);
}
function LosowanieMin() {
    var i=0
    do {
        var w=losowa(0,WIERSZE-1);
        var k=losowa(0,KOLUMNY-1);
        if (MINY[w][k]==0){
            i = i + 1;
            MINY[w][k]=10;
        }
    } while (i < ILE_MIN);
}
```

*funkcja losowa(p,k) podaje liczbę losową (całkowitą) z przedziały od p do k
 funkcja LosowanieMin działa w pętli dopóki nie zostanie wylosowana odpowiednia ILE_MIN liczba min
 losowane są numer wiersza i numer kolumny i do tej komórki tablicy wpisywana jest liczba 10
 jeżeli wylosowana komórka już jest zajęta przez 10 pętla kontynuuje działanie
 program zawiesi się, gdy będziemy próbować wylosować np. 100 min do tablicy 8x8 - braknie pustych pól*

- Zapisz dokumenty i odśwież przeglądarkę
- W oknie konsoli „otwórz” zmienną MINY
powinno być 10 pól wypełnionych liczbą 10
- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)



Plansza (1)

Po rozlosowaniu min ustawiamy tablicę MINY - wszystkie puste pola wypełniamy liczbami, które pokazują liczbę min wokół tego pola

LosowanieMin() :
`console.log(MINY);`

- Dokument **HTML** `console.log(MINY);` ;
`const sk=[1,1,1,0,-1,-1,-1,0];`
`const sw=[1,0,-1,-1,-1,0,1,1];`

```
LiczenieMin();
console.log(POLA);
```

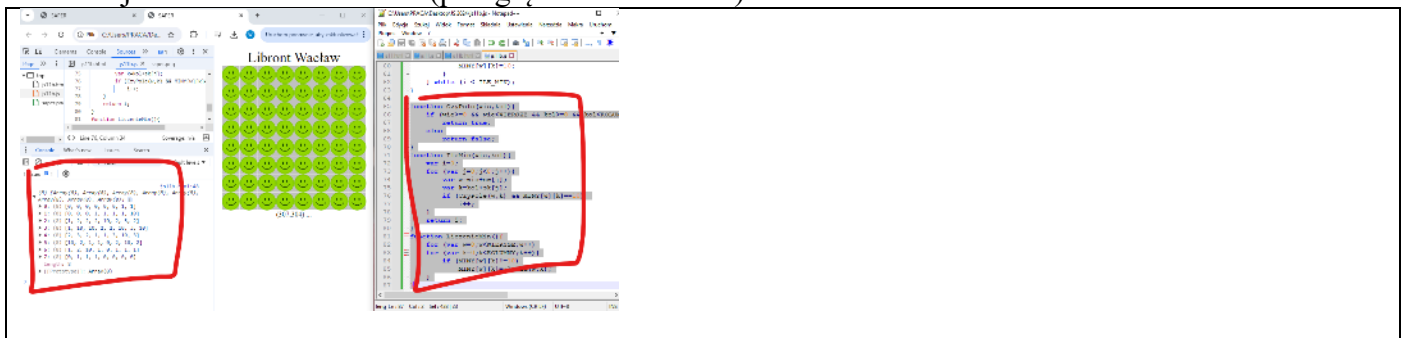
zmienna *sk* i *sw* są tablicami z kierunkami - do wybranego pola dodajemy *sk* i *sw* aby uzyskać współrzędne pól dookoła
funkcja *LiczenieMin* sprawdza ile min jest wokół danego pola w całej tablicy *MINY*

• Dokument JS

```
function CzyPole(wie, kol) {
    if (wie >= 0 && wie < WIERSZE && kol >= 0 && kol < KOLUMNY)
        return true;
    else
        return false;
}
function IleMin(wie, kol) {
    var i = 0;
    for (var j = 0; j < 8; j++) {
        var w = wie + sw[j];
        var k = kol + sk[j];
        if (CzyPole(w, k) && MINY[w][k] == 10)
            i++;
    }
    return i;
}
function LiczenieMin() {
    for (var w = 0; w < WIERSZE; w++)
        for (var k = 0; k < KOLUMNY; k++) {
            if (MINY[w][k] != 10)
                MINY[w][k] = IleMin(w, k);
        }
}
```

sprawdzanie pól generuje mnóstwo błędów - nie można sprawdzać pól (tablic), których komórek nie ma (np. [-1, -1])
funkcja *CzyPole* sprawdza czy podane jako parametr *wie* i *kol* mieszczą się w zakresie
funkcja *IleMin* zlicza ilość min wokół pola (*wie, kol*), posługuje się wektorami *sw* i *sk* do wyliczenia współrzędnych wokół
funkcja *LiczenieMin* sprawdza za pomocą funkcji *IleMin* całą tablicę

- Zapisz dokumenty i odśwież przeglądarkę
- W oknie konsoli „otwórz” zmienną **MINY**
wszystkie pola bez min powinny mieć wpisane liczby min wokół (chyba, że pole nie ma żadnej miny dookoła - jest puste)
- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)



Testowanie planszy (1)

Myszka wskazuje współrzędne na canvas, które należy przeliczyć na współrzędne pól-komórek tablicy, co umożliwi sprawdzenie i wyświetlenie na ekranie podpowiedzi

• Dokument JS

```
function div(a, b) {
    return (Math.round(a/b - 0.5));
}
function PrzeliczXYKW() {
    var w = div(MYSZKA.y, 40);
    var k = div(MYSZKA.x, 40);
    return {w, k};
}
```

funkcja *PrzeliczXYKW* przelicza współrzędne myszki na komórki tablicy - dzieli całkowicie przez szerokość pola w pikselach
`idMYSZK.innerHTML=`

• Dokument JS, funkcja *Myszka()* }

```
var wk = PrzeliczXYKW();
var wie = wk.w;
var kol = wk.k;
```

```
idMYSZ.innerHTML=idMYSZ.innerHTML+
"("+wie+", "+kol+") "+
"<br>MINY: "+MINY[wie][kol]+
" POLA: "+POLA[wie][kol]+
" ILE: "+IleMin(wie, kol);
```

przeliczamy współrzędne myszki na

dopisujemy do pola idMYSZ: zawartość tablicy MINY, POLA i liczbę min wokół pola, które wskazuje myszka

- Zapisz dokumenty i odśwież przeglądarkę
- (1) Wskaż myszką pole, w którym znajduje się mina **MINY=10 ILE=0**
- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)

(1)

Przerysowanie planszy (1)

Tablica POLA zawiera numery obrazków z pliku saper.png, które mają być wyświetlone na ekranie

- Dokument JS

```
function RysujPola() {
  for (var w=0;w<WIERSZE;w++)
    for (var k=0;k<KOLUMNY;k++){
      var nr=POLA[w][k];
      if (nr==0) nr=MINY[w][k];
      S.drawImage(IMG,nr*40,0,40,40,k*40,w*40,40,40);
    }
}
```

funkcja RysujPola „zagłada” do tablicy POLA

- gdy pole jest odkryte = 0, to wstawi obrazek z tablicy MINY

- gdy pole jest zakryte to wstawi obrazek z tablicy POLA zakryte 9 lub flaga 11

- Dokument HTML, przed `</script>`

```
LiczenieMin();
console.log(POLA);
IMG.onload=function () {
  RysujPola();
}
```

- (1) Zapisz dokumenty i odśwież przeglądarkę

ponieważ wszystkie pola w tablicy POLA mają numer 9 - zakryte więc wyświetlane są te obrazki chcemy jednak zobaczyć, co znajduje się w tablicy - na chwilę dlatego wyzerujemy tablicę POLA

- Dokument HTML, przed rysowaniem


```
LiczenieMin();
IMG.onload=function ()
for (var w=0;w<WIERSZE;w++)
for (var k=0;k<KOLUMNY;k++)
POLA[w][k]=0;
```

- Zapisz dokumenty i odśwież przeglądarkę (kilka razy)

sprawdź czy ilość min wokół pól jest dobrze policzona
- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)

```
(8) [Array(8), Array(8), Array(8),
Array(8), Array(8), Array(8)]
▶ 0: (8) [9, 9, 9, 9, 9, 9, 9, 9]
▶ 1: (8) [9, 9, 9, 9, 9, 9, 9, 9]
▶ 2: (8) [9, 9, 9, 9, 9, 9, 9, 9]
▶ 3: (8) [9, 9, 9, 9, 9, 9, 9, 9]
▶ 4: (8) [9, 9, 9, 9, 9, 9, 9, 9]
▶ 5: (8) [9, 9, 9, 9, 9, 9, 9, 9]
▶ 6: (8) [9, 9, 9, 9, 9, 9, 9, 9]
▶ 7: (8) [9, 9, 9, 9, 9, 9, 9, 9]
(1) length: 8
```

Libront Wacław

Odkrywanie pól (1)

Klikamy w pole i je odkrywamy. Jeżeli klikniemy z SHIFT - ustawiamy flagę

- Dokument **HTML** - usuń 3 instrukcje odkrywania tablicy

```
for (var w=0;w<WIERSZE;w++)
for (var k=0;k<KOLUMNY;k++)
    POLA[w][k]=0;
```

~~MYSZKA.y=e.offsetY;~~

- Dokument **JS**, funkcja **Zdarzenie**, zdarzenie **onclick** }

```
var wk=PrzeliczXYKW();
var wie=wk.w;
var kol=wk.k;
if (CzyPole(wie,kol)){
    var p=POLA[wie][kol];
    if (SHIFT)
        if (p==9)
            POLA[wie][kol]=11;
        else
            POLA[wie][kol]=9;
    else {
        POLA[wie][kol]=0;
    }
}
RysujPola();
```

kliknięto w pole
 wyliczamy wiersz i kolumnę
 sprawdzamy, czy kliknięto w obszar saper
 jeżeli wciśnięty SHIFT to ustawiamy 11 lub likwidujemy 9 flagę
 w pozostałych przypadkach ustawiamy tablicę PLA na 0 - będzie odkryte
 przerysowujemy całą planszę

- Zapisz dokumenty i odśwież przeglądarkę
- Odkryj wszystkie pola z minami
- Ustaw flagi na kilku pustych polach
- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)

Rekurencyjne odkrywanie (1)

Prawdziwy saper wspomaga odkrywanie pól. Gdy obok odkrytego właśnie pustego pola znajdują się inne puste, to zostają automatycznie odkryte. Wykonuje to funkcja rekurencyjna - wywołuje samą siebie na kolejnych poziomach.

- Dokument **JS**

```
function OdkryjPuste(wie,kol){
    for (var j=0;j<8;j++){
```



```

var w=wie+sw[j];
var k=kol+sk[j];
if (CzyPole(w,k)) {
    var m=MINY[w][k];
    var p=POLA[w][k];
    if (m==0 && p!=0) {
        POLA[w][k]=0;
        OdkryjPuste(w,k);
    }
    if (m!=10) {
        POLA[w][k]=0;
    }
}
} //czypole
} //for
}

```

sprawdzamy, co się dzieje w okolo pola (wie,kol) za pomocą wektorów sw i sk
 jeżeli w nowej komórce nie ma min wokół MINY=0, to rekurencyjnie szukaj kolejnego
 jeżeli w nowej komórce są liczby 1..8, to tylko odkryj to pole POLA=0

```

else {
    POLA[wie][kol]=0;
}

```

- Dokument JS, funkcja Zdarzenie, zdarzenie onclick }

```

var m=MINY[wie][kol];
var p=POLA[wie][kol];
if (m==0) OdkryjPuste(wie,kol);

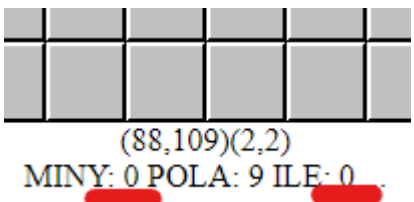
```

kliknięto w pole

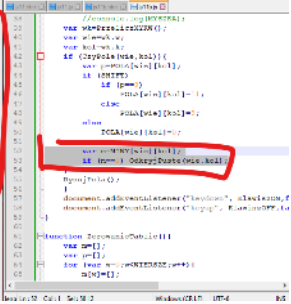
sprawdzamy stan tablicy MINY

jeżeli w okolo pola nie ma min, to uruchamiamy rekurencyjne sprawdzanie pozostałych pól

- Zapisz dokumenty i odśwież przeglądarkę
- (1) Kliknij pole, które nie ma wokół min: MINY=0 ILE=0
powinny odsonić się kolejne pola, wokół których też nie ma min
- Odkryj w ten sposób inne pola na planszy
- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)



(1)



Koniec gry -porażka (1)

„Wdepnąłeś” w minę - koniec gry

```

if (m==0) OdkryjPuste(wie,kol);
}

```

- Dokument JS, funkcja Zdarzenie, zdarzenie onclick

```

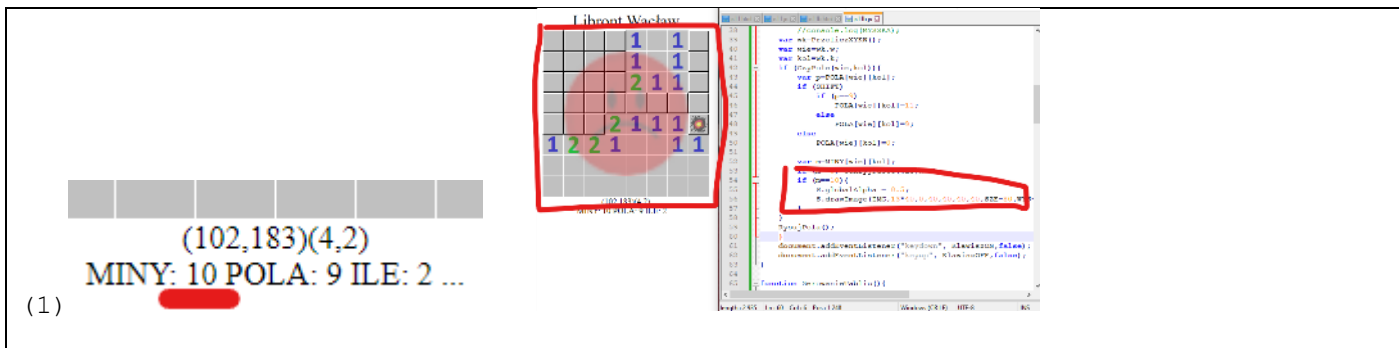
if (m==10) {
    S.globalAlpha = 0.5;
    S.drawImage (IMG, 13*40, 0, 40, 40, 40, 40, SZE-80, WYS-80) ;
}

```

kliknięto w pole z miną

rysujemy wielką czerwoną półprzezroczystą buźkę

- Zapisz dokumenty i odśwież przeglądarkę
- (1) Kliknij w pole z miną: MINY=10
- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)



Koniec gry - sukces (1)

Odkryto wszystkie pola bez min. Pola z minami nieodkryte lub oznaczone flagą

- Dokument JS

```
function CzyKoniec(){
    var ile=0;
    var koniec=false;
    for (var w=0;w<WIERSZE;w++){
        for (var k=0;k<KOLUMNY;k++){
            var p=POLA[w][k];
            if (p==9 || p==11) ile++;
        }
    }
    if (ile==ILE_MIN) koniec=true;
    return koniec;
}
```

sprawdzamy wszystkie pola
jeżeli jest nieodkryte lub z flagą to zwiększamy ILE
jeżeli ILE jest równe liczbie min na planszy to koniec

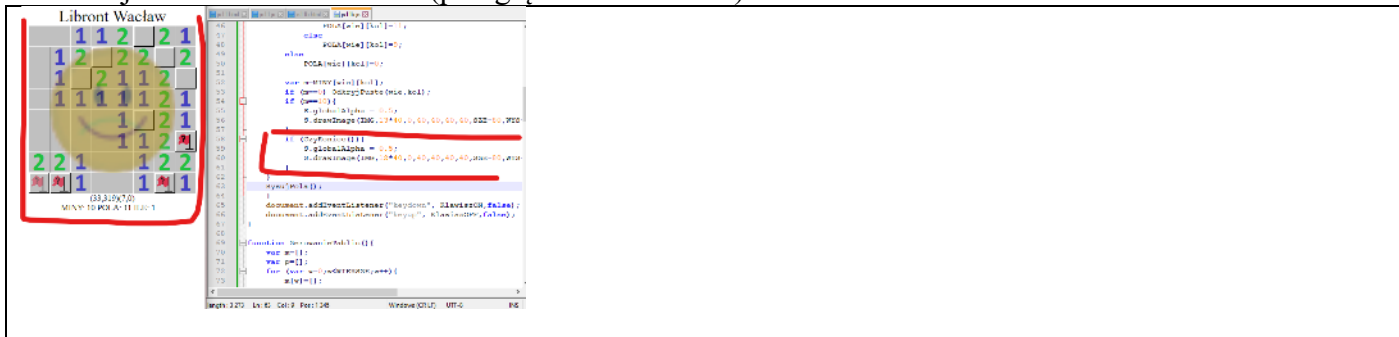
```
S.drawImage(IMG,13*40,
```

- Dokument JS, funkcja Zdarzenie, zdarzenie onclick

```
if (CzyKoniec()){
    S.globalAlpha = 0.5;
    S.drawImage(IMG,12*40,0,40,40,40,40,SZE-80,WYS-80);
}
```

odkryto wszystkie pola bez min
rysujemy wielką zieloną półprzezroczystą buźkę

- Zapisz dokumenty i odśwież przeglądarkę
będziesz musiał odkryć wszystkie pola bez min lub zaznaczyć flagą pola z minami
- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)



Czas (1)

W momencie pierwszego kliknięcia zaczynamy liczyć czas gry. Po zakończeniu wypisujemy na ekranie.

```
<label id=idKLIKAW></label>
</center>
```

- Dokument HTML

```
<br>
<label id=idCZAS></label>
```

do tego pola wpisujemy czas zakończenia gry

```
console.log(POLA);
```

- Dokument **HTML**

```
var CzasStart=0
```

w zmiennej przechowujemy czas startu gry

```
S.canvas.onclick=function(e) {
```

- Dokument **JS**, funkcja **Zdarzenia**

```
MYSZKA.x=e.offsetX;
```

```
if (CzasStart==0) CzasStart=new Date();
```

przy pierwszym kliknięciu startujemy czas

```
RysujPola();
```

- Dokument **JS**, funkcja **Zdarzenia**

```
JakiCzas();
```

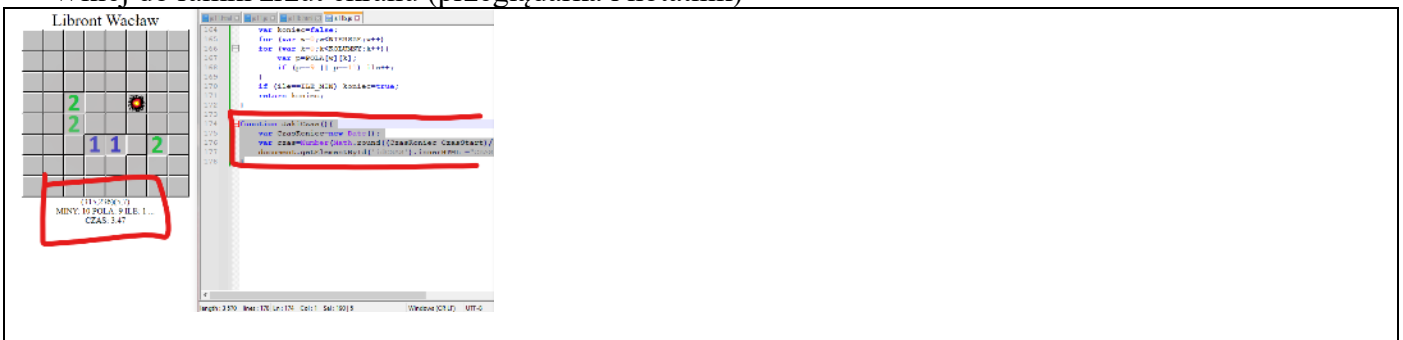
po każdym kliknięciu wpisujemy czas gry

- Dokument **JS**

```
function JakiCzas(){
    var CzasKoniec=new Date();
    var czas=Number(Math.round((CzasKoniec-CzasStart)/1000 + 'e+2') + 'e-2');
    document.getElementById('idCZAS').innerHTML ="CZAS: "+czas;
}
```

przy pierwszym kliknięciu startujemy czas

- Zapisz dokumenty i odśwież przeglądarkę
- Poklikaj w pola
po każdym kliknięciu pojawia czas gry
- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)



Od nowa (1)

```
<label id=idCZAS></label>
```

- Dokument **HTML**

```
</center>
```

```
<br>
```

```
<input type=button value="OD NOWA" onclick=OdNowa()>
```

przycisk, do którego przypisano funkcję *OdNowa()*

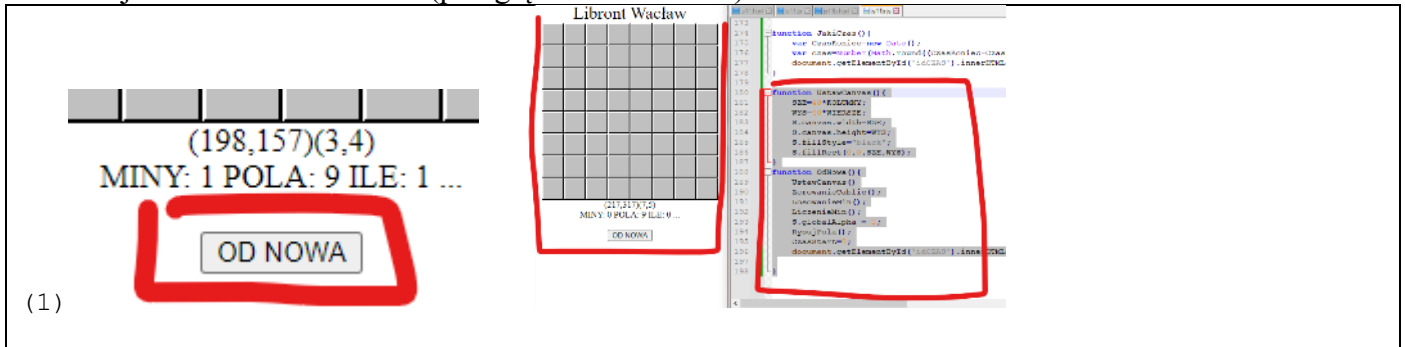
- (1) Zapisz dokumenty i odśwież przeglądarkę
- Dokument **JS**

```
function UstawCanvas(){
    SZE=40*KOLUMNY;
    WYS=40*WIERSZE;
    S.canvas.width=SZE;
    S.canvas.height=WYS;
    S.fillStyle="black";
    S.fillRect(0,0,SZE,WYS);
}
function OdNowa(){
    UstawCanvas();
    ZerowanieTablic();
    LosowanieMin();
    LiczenieMin();
    S.globalAlpha = 1;
    RysujPola();
    CzasStart=0;
    document.getElementById('idCZAS').innerHTML ="";
}
```

funkcja *UstawCanvas* wylicza nowe wymiary canvas i rysuje go od nowa - czarny kolor

funkcja OdNowa: rysuje canvas, zeruje tablice, losuje miny, zlicza miny wokół pól i rysuje zakrytą planszę
S.globalAlpha = 1 - przezroczystość zlikwidowana - była ustawiona, gdy rysowaliśmy buźki na zakończenie

- Rozpocznij grę i kliknij w przycisk *gra powinna się zresetować*
- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)



Stopnie trudności (1)

- Dokument HTML `<input type=button`

```
<select name=WERSJA>
  <option id=idPO selected=true>początkujący 8x8 10
  <option id=idZA>zaawansowany 16x16 40
  <option id=idEK>ekspert 32x16 100
</select>
```

pole combi, z którego możemy wybrać stopnie trudności

- Dokument JS

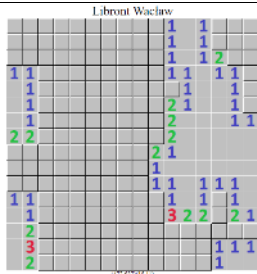
```
function WersjaSapera() {
  if (document.getElementById("idPO").selected) {
    ILE_MIN=10;
    WIERSZE=8;
    KOLUMNY=8;
  }
  if (document.getElementById("idZA").selected) {
    ILE_MIN=40;
    WIERSZE=16;
    KOLUMNY=16;
  }
  if (document.getElementById("idEK").selected) {
    ILE_MIN=100;
    WIERSZE=16;
    KOLUMNY=32;
  }
}
```

pole combi, z którego możemy wybrać stopnie trudności

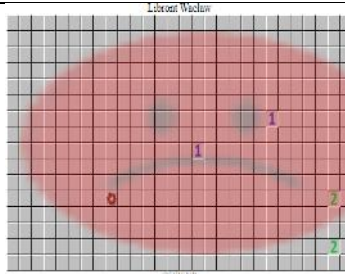
- Dokument JS, funkcja `OdNowa` `UstawCanvas`
`WersjaSapera();`

gramy od nowa - na samym początku pobieramy wartość z pola combi i ustawiamy podstawowe zmienne planszy

- Zapisz dokumenty i odśwież przeglądarkę
- (1) Ustaw wersję **zaawansowany 16x16** i wciśnij przycisk **OD NOWA**
- Pomniejsz okno przeglądarki, aby było widać wszystko
- Kliknij w pola planszy
- (2) Ustaw wersję **ekspert 32x16** i wciśnij przycisk **OD NOWA**
- Pomniejsz okno przeglądarki, aby było widać wszystko
- Kliknij w pola planszy
- Spróbuj rozwiązać Sapera na poziomie **ekspert** 😊
- Wklej do ramki zrzut ekranu (przeglądarka i notatnik)



(1)



(2)