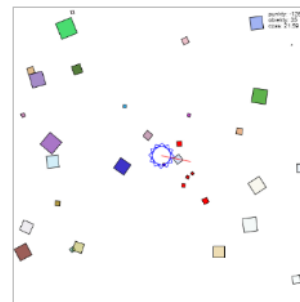


## JS 06 – Obiekty – Gra Laser (12)

Na środku ekranu kręci się laserowa baza (trzy obrócone kwadraty), z której (klikając myszką) możemy wypuszczać laserowe strzały. Z brzegów ekranu „płyną” w kierunku bazy wrogie (kwadratowe) okręty przeciwnika. Jeśli laserowa strzała trafi w kwadratowy okręt, ten zmniejsza się i znika, a my otrzymujemy punkty.



**Pamiętaj o tym, by zrzut ekranu DOKUMENTOWAŁ Twoją pracę**

### Canvas (1)

- W swoim folderze utwórz 2 nowe dokumenty: **js06.html** **js06.js**
- Otwórz oba dokumenty w notatniku, a dokument HTML w przeglądarce
- Do dokumentu **HTML** wklej tekst z ramki

```
<html>
<head>
  <meta charset=utf8>
  <title> LASER </title>
  <script src=js06.js></script>
</head>
<body>
  <canvas width=500 height=500 id=CAN></canvas>
<font size=6>Libront Waclaw</font>
<script>
  var C = CAN.getContext("2d");
  var w=C.canvas.width;
  var h=C.canvas.height;
  C.strokeStyle="black";
  C.strokeRect(0,0,w,h);
  C.fillText("Libront Waclaw",2,10);
</script>
</body>
</html>
```

*Kwadratowy obszar canvas.*

- Wpisz swoje inicjały do tytułu dokumentu
- Wpisz swoje nazwisko i imię (2 razy)
- Zapisz dokument i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu (przeglądarka notatnik)



### Obiekt (1)

*Współczesne programowanie opiera się na obiektach. Obiekty łączą w sobie zmienne (własności) oraz funkcje, które działają na tych zmiennych (metody).*

*W naszej grze zdefiniujemy kilka obiektów. Najbardziej podstawowym będzie OKWA rysujący losowe kwadraty z własnościami: x,y,vx,vy, ... i metodami: init() i rysuj(), ruch(). We wnętrzu obiektu posługujemy się poleceniem this, aby odwołać się do przekazanych parametrów.*

- Do dokumentu **JS** wklej tekst z ramki

```
//obiekt - funkcja OKWA
function OKWA() {
  //metody
  //tworzenie losowego kwadratu
  this.init=function() {
    }
  this.init();
  //rysowanie
```

```
this.rysuj=function() {  
  
}  
//zmiana położenia  
this.ruch=function() {  
  
}  
}
```

**OKWA** szkielet obiektu rysującego kwadraty  
**this.** w ten sposób odwołujemy się do elementów obiektu w jego wnętrzu  
**init()** metoda automatycznie tworzy nowy obiekt po poleceniu `zmienna=new OKWA();`  
**rysuj()** rysowanie obiektu  
**ruch()** obliczanie nowego położenia

```
this.rysuj=function() {
```

- Do dokumentu **JS**, do wnętrza metody **rysuj** }, wklej tekst z ramki

```
C.save();  
C.translate(this.x,this.y);  
C.rotate(Math.PI*this.kat/180);  
C.fillStyle=this.kol;  
C.fillRect(-this.bok/2,-this.bok/2,this.bok,this.bok);  
C.strokeStyle="black";  
C.strokeRect(-this.bok/2,-this.bok/2,this.bok,this.bok);  
C.restore();
```

obrotowy kwadrat rysowany od środka za pomocą transformacji i rotacji canvas własności kwadratu:

**x,y** współrzędne środka  
**bok** bok kwadratu  
**kat** kąt obrotu  
**kol** kolor wypełnienia  
kwadrat ma zawsze czarną ramkę

```
</script>
```

- Do dokumentu **HTML**, przed znacznik **</script>** wpisz instrukcje

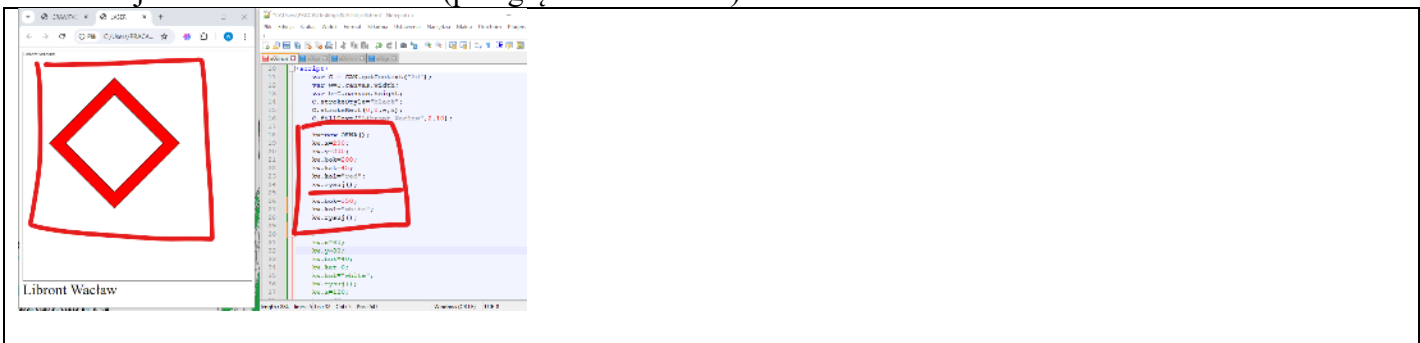
```
kw.x=200;  
kw.y=200;  
kw.bok=200;  
kw.kat=45;  
kw.kol="red";  
kw.rysuj();
```

polecenie `new` tworzy nowy obiekt o nazwie `kw` za pomocą „kropek” ustawiamy własności obiektu `kw` metodą `rysuj()` uruchamia funkcję rysującą

- We wnętrzu czerwonego kwadratu narysuj biały o boku 150

```
kw.bok=150;  
kw.kol="white";  
kw.rysuj();
```

- Zapisz dokumenty i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu (przeglądarka notatnik)



## Losowanie obiektów (1)

Na ekranie zostanie namalowanych mnóstwo losowych (kolor i położenie) obiektów

- W dokumencie **HTML** usuń instrukcje rysowania dwóch kwadratów od `kw=new OKWA();` do `kw.rysuj();`;
- Do dokumentu **JS** wklej nowe funkcje z ramki

```
function losowa(p,k) {
    return Math.floor(Math.random()*(k-p+1)+p);
}
function losRGBA(){
    var r = losowa(0,255);
    var g = losowa(0,255);
    var b = losowa(0,255);
    var a = Math.random()*0.5+0.5;
    return "rgba("+r+","+g+","+b+","+a+")";
}
```

losowanie liczb z przedziału <p..k>

losowanie kolorów RGBA, łącznie z przezroczystością maksymalnie 50%

```
this.init=function(){
```

- Do dokumentu **JS**, do wnętrza metody `init` }, wklej tekst z ramki

```
this.x=losowa(100,400);
this.y=losowa(100,400);
this.bok=losowa(5,100);
this.kat=losowa(0,90);
this.kol=losRGBA();
this.vx=losowa(-4,4);
this.vy=losowa(-4,4);
this.vk=losowa(-4,4);
```

w metodzie `init()` losujemy parametry kwadratu

- Do dokumentu **HTML**, przed znacznik `</script>` wpisz instrukcje

```
var TK=[];
var ileK=100;
var PUNKTY=0;
for (var i=0;i<ileK;i++){
    TK[i]=new OKWA();
    TK[i].rysuj();
}
```

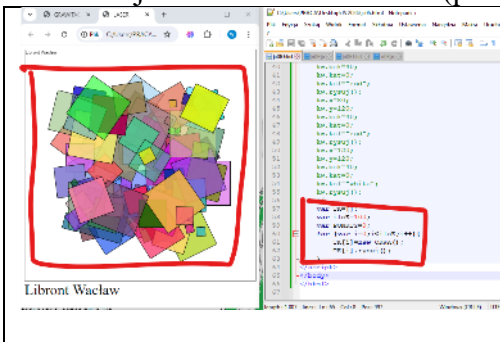
`TK[]` tablica na kwadratowe obiekty

pętla `FOR` – „kręci” się `ileK` razy

`TK[i]=new OKWA();` w komórce tablicy `TK` utworzony zostanie nowy obiekt z losowymi parametrami

`TK[i].rysuj();` rysowanie obiektu znajdującego się w komórce tablicy `TK`

- Zapisz dokumenty i odśwież kilka razy przeglądarkę
- Wklej do ramki zrzut ekranu (przeglądarka notatnik)



## Poruszanie obiektów (1)

```
this.ruch=function(){
```

- Do dokumentu **JS**, do wnętrza metody `ruch` }, wklej tekst z ramki

```
var b2=this.bok/2;
this.x=this.x+this.vx;
this.y=this.y+this.vy;
```

```

this.kat=this.kat+this.vk;
if (this.x-b2 < 0 || this.x+b2 > h){this.vx=-this.vx;}
if (this.y-b2 < 0 || this.y+b2 > w){this.vy=-this.vy;}
if (this.kat>360){this.kat=this.kat-360}
//automatyczne przerysowanie
this.rysuj();

```

*b2* zamiast kilka razy pisać dzielenie tworzymy jedną zmienną z połową boku  
*this.x=this.x+this.vx;* zmiana położenia obiektu  
*this.x-b2 < 0* odbicie od lewego brzegu  
*this.x+b2 > h* odbicie od prawego brzegu  
*this.vx=-this.vx;* zmiana kierunku ruchu  
*this.kat>360* gdy kąt przekroczy wartość 360, to zliczamy kąt od nowa *this.kat=this.kat-360*

- Do dokumentu JS (na koniec) wklej nową funkcję z ramki

```

function LASER() {
    C.clearRect(0,0,w,h);
    C.strokeStyle="black";
    C.strokeRect(0,0,w,h);
    for (var i=0;i<ileK;i++){TK[i].ruch();}
    clearTimeout(CZAS);
    CZAS=setTimeout(LASER,SKOK);
}

```

*główna pętla animacji*  
*czyścimy canvas*  
*rysujemy ramkę*  
*w pętli FOR poruszamy wszystkimi kwadratami*  
*rekurencyjnie wykonujemy kolejne pętle*

- Do dokumentu HTML, przed znacznik `</script>` wpisz tekst

```

var SKOK=20;
var CZAS;
LASER();

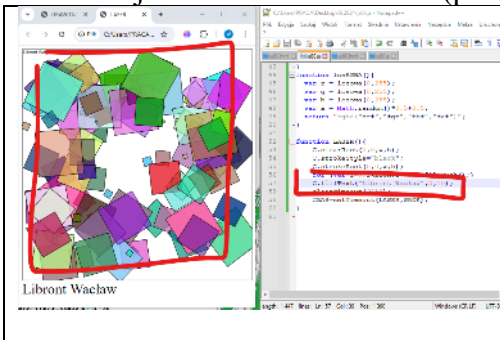
```

*Zmienne CZAS i SKOK służą do wykonywania rekurencyjnej pętli animacyjnej*

- W dokumencie HTML przenieś polecenie `C.fillText("Libront Waclaw",2,10);`

do wnętrza funkcji LASER `clearTimeout(CZAS);`  
*aby nazwisko i imię było cały czas widoczne*

- Zapisz dokumenty i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu (przeglądarka notatnik)



## Gwiazda (1)

*Na środku pola gry obraca się gwiazda złożona z trzech kwadratów*

- Do dokumentu JS (na koniec) wklej definicję nowego obiektu

```

//kręcąca się gwiazda złożona
function OGWI() {
    this.init=function() {
        this.x=w/2;
        this.y=h/2;
        this.bok=30;
        this.kat=0;
        this.kol="red";
        this.vk=3;
    }
    this.init();
}

```

```
// obliczanie i przerysowanie
this.ruch=function () {
  this.kat=this.kat+this.vk;
  this.rysuj ();
}
// rysowanie gwiazdy
this.rysuj=function () {
  var b2=this.bok/2;
  var b=this.bok;
  C.save ();
  C.fillStyle="rgba (0,0,255,0.5)";
  C.strokeStyle=this.kol;
  C.translate (this.x,this.y);
  C.rotate (Math.PI*this.kat/180);
  C.strokeRect (-b2,-b2,b,b);
  C.rotate (Math.PI*30/180);
  C.strokeRect (-b2,-b2,b,b);
  C.rotate (Math.PI*30/180);
  C.strokeRect (-b2,-b2,b,b);
  C.restore ();
}
}
```

obiekt OGWI to trzy kwadraty w kolorze niebieskim, kręcące się na środku canvas  
 metoda init() ustawia położenie, obroty i kolor  
 metoda ruch() obraca obiektem  
 metoda rysuj() rysuje trzy niebieskie kwadraty obrócone wokół siebie o kąt 30°

- Do dokumentu JS, do wnętrza funkcji LASER() `clearTimeout (CZAS);` wpisz `GWI.ruch ();`  
 poruszanie w trakcie animacji gwiazdy

```
C.fillText ("Libront Waclaw",2,10);
```

- Do dokumentu HTML, przed funkcję LASER() `var CZAS;` wpisz `var GWI=new OGWI ();`  
 tworzenie obiektu gwiazda

```
var CZAS;
```

- W dokumencie HTML, zmniejsz liczbę kolorowych kwadratów do 10
- W dokumencie JS, zmień kolor gwiazdy na niebieski
- Zapisz dokumenty i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu (przeglądarka notatnik)



## Laserowy strzał (1)

Laserowy strzał, to linia biegnąca od środka obracającej się gwiazdy do kursora myszki. Rysowana linia jest nową metodą strzel w obiekcie GWIAZDA. Rysujemy wtedy, gdy przycisk myszki jest wciśnięty `MyszPrzycisk=true`, gdy puścimy przycisk myszki - linia nie jest rysowana.  
 Zdarzenia związane z ruchem myszki, naciskaniem i zwalnianiem przycisku opisują zdarzenia przypisane do canvas: `onmousemove`, `onmousedown`, `onmouseup`.

- Do dokumentu HTML, przed funkcję LASER() `var GWI=new OGWI ();` wpisz `LASER ();` wklej tekst z ramki

```
var GWI=new OGWI ();
```

```
var MYSZKA = {x:0, y:0};
var MyszPrzycisk = false;
C.canvas.onmousemove = function(e) {
  MYSZKA.x = e.offsetX;
  MYSZKA.y = e.offsetY;
};
```

```
C.canvas.onmousedown = function(e) {
    MyszPrzycisk = true;
};
C.canvas.onmouseup = function(e) {
    MyszPrzycisk = false;
};
```

Położenie myszki zapisywane jest w zmiennej  $MYSZKA=\{x:0, y:0\}$  – dwa elementy  
 Canvas jest przesunięty względem tego, co pokazuje myszka o 8 pikseli w prawy, dolny róg.  
 C.canvas.onmousemove - pobieranie położenia wskaźnika myszki na obszarze canvas  
 C.canvas.onmousedown - lewy przycisk myszki wciśnięty  
 canvas.onmouseup - lewy przycisk myszki zwolniony

```
C.restore();
```

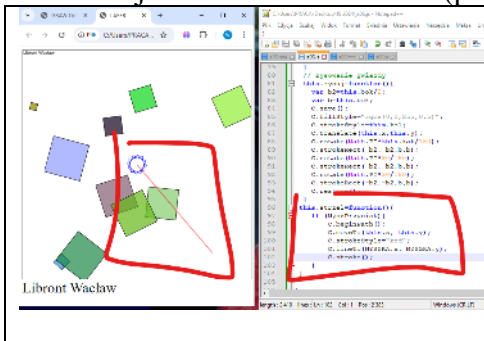
- Do dokumentu JS, do obiektu OGWI } wklej nową metodę **strzel**

```
this.strzel=function() {
    if (MyszPrzycisk) {
        C.beginPath();
        C.moveTo(this.x, this.y);
        C.strokeStyle="red";
        C.lineTo(MYSZKA.x, MYSZKA.y);
        C.stroke();
    }
}
```

strzel() metoda rysuje czerwoną linię gdy wciśnięty przycisk myszki

```
this.rysuj();
```

- Do dokumentu JS, do obiektu OGWI do metody **ruch** } wpisz instrukcję **this.strzel();**  
 strzelanie, gdy gwiazda się obraca
- Zapisz dokumenty i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu (przeglądarka notatnik), **gdy widać laserowy strzał**



## Trafienie gwiazdy (1)

Jak sprawdzić, czy gwiazda została trafiona przez jeden z kwadratów? W tym celu obliczymy odległość pomiędzy kwadratem a gwiazdą. Jeśli jest mniejsza niż suma połowy kwadratu i połowy gwiazdy, to znaczy, że gwiazda została trafiona.

- Do dokumentu JS, jako nową metodę obiektu OGWI } wklej tekst z ramki

```
this.CzyTrafiona=function(obiekt) {
    var sx=this.x-obiekt.x;
    var sy=this.y-obiekt.y;
    var dl=Math.sqrt(sx*sx+sy*sy);
    if (dl < this.bok/2 + obiekt.bok/2){return true}
    return false;
}
```

CzyTrafiona metoda sprawdza kwadratowy obiekt (podany jako parametr) i oblicza odległość kwadratu od gwiazdy gdy odległość jest mniejsza, to metoda zwraca wartość prawda

- Dokument JS, funkcja **LASER**
  - usuń pętlę `for (var i=0;i<ileK;i++){TK[i].ruch();}`
  - wpisz w miejsce pętli instrukcje z ramki

```
for (var i=0;i<TK.length;i++) {
    TK[i].ruch();
    if (GWI.CzyTrafiona(TK[i])) {
```

```
PUNKTY=PUNKTY-1;
```

```
}
```

```
}
```

w pętli FOR która sprawdza całą tablicę jej długość może ulec zmianie, gdy w programie ustawimy tworzenie nowych obiektów lub ich zestrzelenie po narysowaniu kwadratów – za pomocą metody `move` sprawdzamy, czy gwiazda została trafiona danym obiektem, przekazywanym jako parametr. Jeżeli tak, to zmniejszane są punkty

- Do dokumentu **HTML**, przed funkcję **LASER()**, wpisz  
`var PUNKTY=0;`
- Zmniejsz **bok** losowanych kwadratów w obiekcie **OKWA** pomiędzy 5..40  
`this.bok=losowa(5,40);`
- Zapisz dokumenty i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu (przeglądarka notatnik)



## Strzelanie (1)

Jak sprawdzić, czy koniec laserowego strzału (położenie myszki) trafiło w środek któregoś kwadratowego wroga? Za każdym razem, gdy strzelamy obliczamy odległość pomiędzy myszką, a środkiem kwadratu. Jeśli ta odległość jest mniejsza niż połowa boku, to znaczy, że strzał się udał. Zwiększamy **PUNKTY**, zmniejszamy bok trafionego kwadratowego wroga, a gdy zniknie, to rodzi się nowy..

```
if (this.kat>360)
```

```
this.rysuj();
```

- Do dokumentu **JS**, obiekt **OGWI**, metoda **ruch**, `this.rysuj();` wklej tekst z ramki

```
if (MyszPrzycisk==true) {  
    var dx=MYSZKA.x-this.x;  
    var dy=MYSZKA.y-this.y;  
    var d=Math.sqrt(dx*dx+dy*dy);  
    if (d<b2 || d<10) {  
        PUNKTY=PUNKTY+1;  
        if (this.bok>0){this.bok=this.bok-3;}  
    }  
}
```

gdy wciśnięty przycisk myszki, to obliczamy odległość wskaźnika do kwadratu gdy odległość mniejsza niż połowa boku to dodajemy punkty i zmniejszamy bok kwadratu, aż do zera

- Do dokumentu **JS**, funkcja **LASER**, na koniec pętli **FOR** } wpisz tekst

```
if (TK[i].bok<=0) TK.splice(i, 1)
```

podczas przerysowywania wszystkich obiektów i strzelania sprawdzamy, czy długość boku obiektu nie jest mniejsza niż zero i wtedy usuwamy go z tablicy TK

`TK.splice(i,1)` z tablicy TK usuwamy jeden obiekt o indeksie i

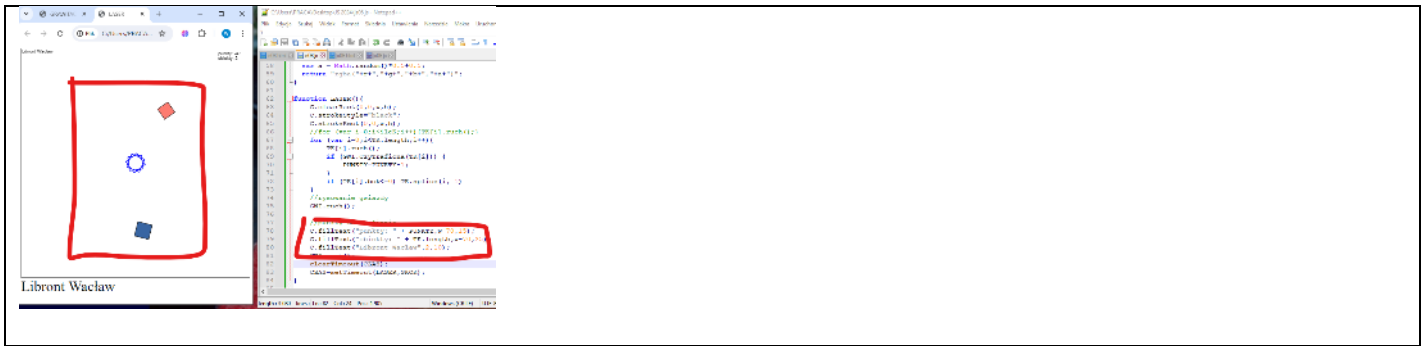
```
C.fillText("punkty: " + PUNKTY,w-70,15);
```

- Do dokumentu **JS**, funkcja **LASER** `C.fillText("Libront Wacław",2,10);` wpisz tekst

```
C.fillText("obiekty: " + TK.length,w-70,25);
```

na stronę liczba obiektów do zestrzelenia

- Zapisz dokumenty i odśwież przeglądarkę
- Spróbuj zestrzelić przynajmniej **połowę kwadratów**
- Wklej do ramki zrzut ekranu (przeglądarka notatnik)



## Błążąca gwiazda (1)

```

this.ruch=function() {
    this.kat=this.kat+this.vk;
}

```

- Dokument JS, obiekt OGWI, metoda ruch

```

this.x=this.x+(MYSZKA.x-this.x)/20;
this.y=this.y+(MYSZKA.y-this.y)/20;

```

*zmiana położenie gwiazdy w kierunku lasera*

```

C.lineTo(MYSZKA.x, MYSZKA.y);
C.stroke();

```

- Dokument JS, obiekt OGWI, metoda strzel

```

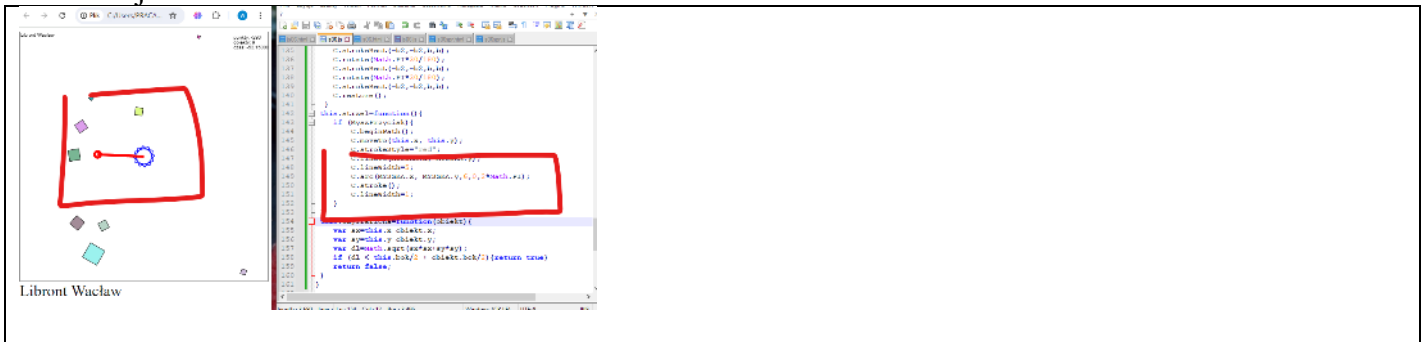
C.lineWidth=5;
C.arc(MYSZKA.x, MYSZKA.y, 6, 0, 2*Math.PI);
C.stroke();
C.lineWidth=1;
}

```

wpisz

*zwiększamy grubość laserowego promienia, na końcu kółko  
dwie instrukcje przed zielonym i jedna po zielonym*

- Zapisz dokumenty i odśwież przeglądarkę
- Strzel laserem i w tym momencie wykonaj zrzut ekranu
- Wklej do ramki zrzut ekranu



## Iskry (1)

*A gdyby tak po trafieniu wroga sypały się z niego mniejsze iskry?*

*Sprawdzamy zestrzelenie obiektu i tworzymy nowy obiekt OKWA. Do tablicy TK dodajemy go za pomocą polecenia push.*

*Aby odróżnić typowy kwadratowy obiekt od iskry nadajemy każdemu dodatkową własność ISKRA.*

```

if (this.bok > 0) {this.bok=this.bok-3;}

```

- Do dokumentu JS, obiekt OKWA, metoda ruch

wklej

```

if (this.bok > 0) {
    var kw = new OKWA();
    kw.x = this.x + kw.vx * 5;
    kw.y = this.y + kw.vy * 5;
    kw.bok = this.bok;
    kw.kol="red";
    kw.ISKRA = true;
    TK.push(kw);
}

```

*strzelamy i trafiamy, obiekt się zmniejsza i sypią się z niego czerwone iskry  
iskra to nowy obiekt OKWA, który*

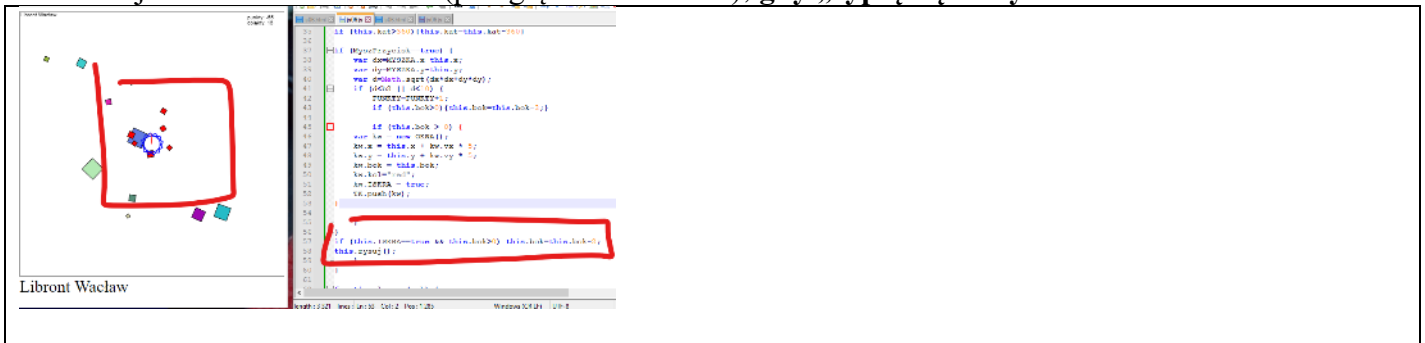


porusza się 5 razy szybciej  
 początkowo ma taki sam bok jak trafiony obiekt  
 kolor czerwony

kw.ISKRA.true - nowa własność obiektu TK, aby odróżnić normalny kwadrat od iskry – te będą same się zmniejszać  
 TK.push(kw) - dołożenie iskry do tablicy TK

```
}
this.rysuj();
```

- Do dokumentu JS, obiekt OKWA, metoda ruch `this.rysuj();` wpisz tekst `if (this.ISKRA==true && this.bok>0) this.bok=this.bok-2;`  
 gdy obiekt jest iskry, to zmniejszaj go niezależnie czy trafiony
- Zapisz dokumenty i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu (przeglądarka notatnik), gdy „sypią się iskry”



## Czas (1)

Po uruchomieniu gry zaczynamy liczyć czas. Masz 60 sekund na zestrzelenie wszystkich obiektów.  
 Gdy zestrzelisz wszystkie obiekty otrzymasz dodatkowe punkty za czas  $wynik = PUNKTY + (60 - \text{sekundy}) * 10$ .

- Do dokumentu HTML, przed funkcję LASER() wpisz tekst

```
var data0=new Date();
```

```
GWI.ruch();
clearTimeout(CZAS);
```

- Do dokumentu JS, do wnętrza funkcji LASER() wklej tekst z ramki

```
var data1=new Date();
var czas=60-Number(Math.round((data1-data0)/1000 + 'e+2') + 'e-2');
C.fillText("czas: " + czas,w-70,35);
```

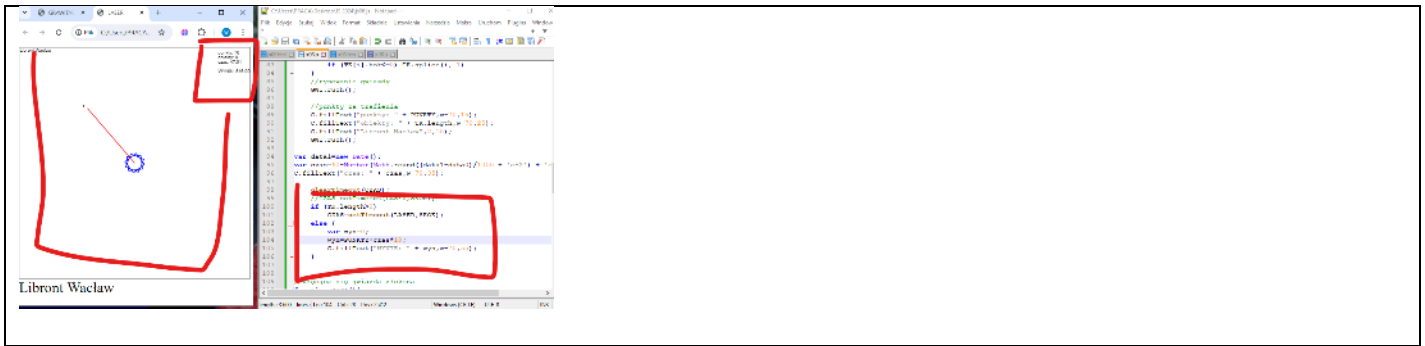
w rekurencyjnej pętli tworzymy kolejny obiekt Date() i wyświetlamy na ekranie liczbę sekund  
 zaokrąglenie za pomocą notacji naukowej (JS nie potrafi normalnie zaokrąglić do określonej liczby miejsc po przecinku)  
 sekundy wyświetlamy na ekranie

- Do dokumentu JS, funkcja LASER(), zastąp polecenie `CZAS=setTimeout(LASER, SKOK);` instrukcjami z ramki

```
if (TK.length>0)
    CZAS=setTimeout(LASER, SKOK);
else {
    var wyn=0;
    wyn=PUNKTY+czas*10;
    C.fillText("WYNIK: " + wyn,w-70,55);
}
```

pętla rekurencyjna działa dotąd aż liczba kwadratów większa od zera  
 gdy zestrzelono wszystkie kwadraty  
 wyświetlamy na ekranie wynik  $PUNKTY + (60 - \text{czas}) * 10$   
 za pomocą dodatkowej zmiennej, bo „+” skleja a nie dodaje

- Zapisz dokumenty i odśwież przeglądarkę
- Zestrzel wszystkie kwadraty
- Wklej do ramki zrzut ekranu (przeglądarka notatnik)



## Przycisk OdNowa (1)

- Dokument **HTML**, przed znacznikiem `<script>` `<font size=6>Libront Waclaw</font>` `<script>` wpisz  
`<br><input type=button value="OD NOWA" onclick=OdNowa ()>`  
 przycisk, do którego przypisano funkcję `OdNowa()`
- Dokument **JS**, na końcu wklej funkcję

```
function OdNowa () {
    PUNKTY=0;
    data0=new Date ();
    STRZAL=false;
    for (var i=0;i<TK.length;i++){
        TK.splice(i, 1);
    }
    for (var i=0;i<ileK;i++){
        TK[i]=new OKWA ();
    }
    GWI=new OGWI ();
    LASER ();
}
```

zerujemy punkty, czas i strzał  
 usuwamy wszystkie kwadraty z tablicy TK  
 tworzymy nowy zestaw kwadratów  
 tworzymy gwiazdę  
 uruchamiamy główną pętlę

- Zapisz dokument i odśwież przeglądarkę
- Wciśnij przycisk i rozpocznij program od nowa
- Wklej do ramki zrzut ekranu i okna JS z funkcją

