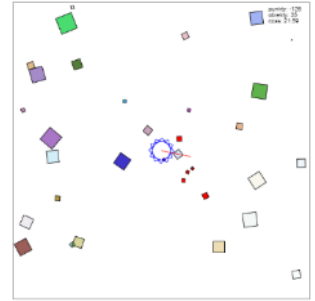


JS 06 – Obiekty – Gra Laser (18)

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.



Canvas i obiekt (2)

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.

- 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>
<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 Wacław",2,10);
</script>
</body>
</html>
```

Kwadratowy obszar canvas.

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

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

  }
  this.init();
  this.rysuj=function() {

  }
  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

po poleceniu *zmienna=new OKWA()* obiekt będzie automatycznie tworzony i uruchamiany

- Wpisz do polecenia **C.fillText()** swoje **nazwisko i imię**
- Zapisz dokumenty i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu HTML.

Tworzenie i rysowanie obiektów (2)

- 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(){`

- Dokument **JS**, obiekt **OKWA()** metoda **this.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 początkowe parametry kwadratu

`this.rysuj=function(){`

- Dokument **JS**, obiekt **OKWA()** metoda **this.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
kwadrat ma zawsze czarną ramkę*

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

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

TK[] tablica na kwadratowe obiekty

w pętli FOR – „zakręci” 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

- W dokumencie **HTML** zmień ilość rysowanych kwadratów na **10**
- Zapisz dokumenty i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu HTML.



Poruszanie obiektów (2)

`this.ruch=function(){`

- Dokument JS, obiekt OKWA(), metoda `this.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}
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 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

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

- Do dokumentu HTML, przed znacznik `</script>` wpisz tekst `C.fillText("Libront Wacław",2,10);`
Zmienne CZAS i SKOK służą do wykonywania rekurencyjnej pętli animacyjnej
- Przenieś instrukcję rysującą na ekranie Twoje **nazwisko i imię** do wnętrza funkcji LASER, za pętlę `for (var i=0;i<ileK;i++){TK[i].ruch();}`
nazwisko i imię cały czas widoczne na stronie
- Zapisz dokumenty i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu HTML.



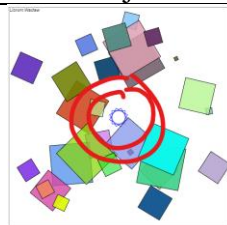
Gwiazda (2)

- Do dokumentu **JS** wklej nowy obiekt z ramki

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

- Dokument **JS**, funkcja **LASER()**, za pętlą **for (...){...}** wpisz instrukcję `GWI.ruch();`
poruszanie w trakcie animacji gwiazdy
- Dokument **HTML**, przed instrukcją **LASER()**, wpisz instrukcję `var GWI=new OGWI();`
tworzenie obiektu gwiazdy
- Zmień kolor gwiazdy na **niebieski**
- Zapisz dokumenty i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu HTML.



Laserowy strzał (2)

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 myszy - linia nie jest rysowana.

Zdarzenia związane z ruchem myszki, naciskaniem i zwalnianiem przycisku opisują zdarzenia przypisane do canvas: `onmousemove`, `onmousedown`, `onmouseup`.

- Do dokumentu **HTML**, pod poleceniem `var C = CAN.getContext("2d");`, wklej tekst z ramki

```
var MYSZKA = {x:0, y:0};
var MyszPrzycisk = false;
C.canvas.onmousemove = function(e) {
    MYSZKA.x = e.clientX - 8;
    MYSZKA.y = e.clientY - 8;
};
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

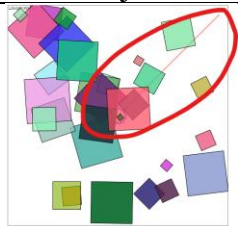
- Dokument **JS**, obiekt **OGWI()** wklej nową metodę `this.strzel()`

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

- Dokument **JS**, metoda **ruch** obiektu **OGWI** wpisz instrukcję `this.strzel();`
- strzelanie, gdy gwiazda się obraca
- Zapisz dokumenty i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu HTML, gdy widać laserowy strzał



Trafienie gwiazdy (2)

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.

```
C.stroke();
}
```

- 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 prawdą

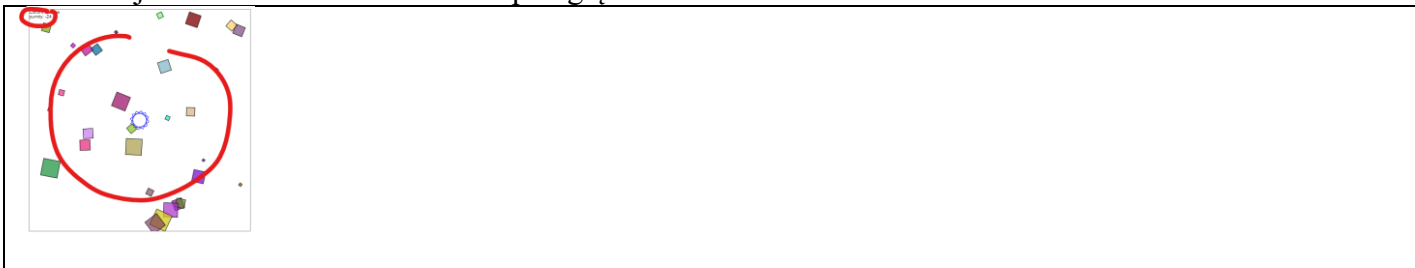
- Dokument **JS**, funkcja **LASER()**

- usuń pętlę `for (var i=0;i<ileK;i++){TK[i].ruch();}`
- wklej 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;
    }
    if (TK[i].bok<=0) {
        TK.splice(i, 1)
    };
}
C.fillText("punkty: " + PUNKTY, 2, 20);
```

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 `ruch` sprawdzamy, czy gwiazda została trafiona danym obiektem, przekazywanym jako parametr. Jeżeli tak, to zmniejszane są punkty 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`

- Do dokumentu **HTML**, przed funkcją **LASER()**, wpisz `var PUNKTY=0;`
- W dokumencie **JS**, w funkcji **OKWA()**, zmniejsz losowaną długość boków z (5,100) na (5,40)
- Zapisz dokumenty i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu HTML



Strzelanie (2)

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.kat=this.kat-360}
this.rysuj();
}
```

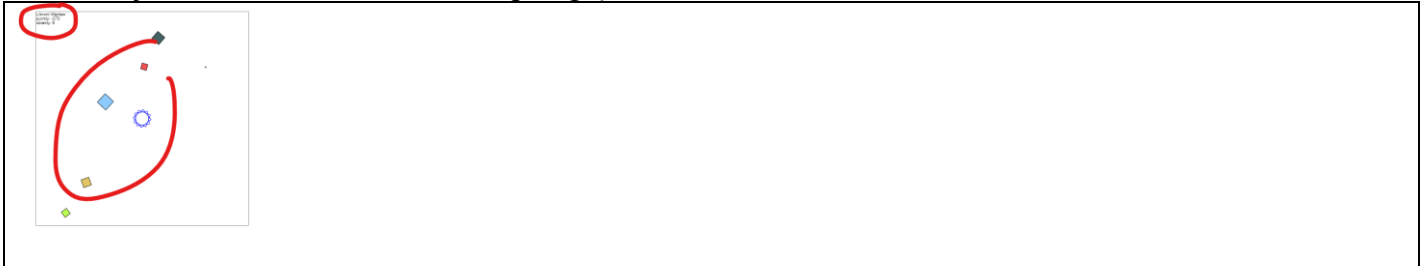
- Dokument **JS**, obiekt **OKWA()**, metoda `this.ruch(){..}` , wklej tekst z ramki

```
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

```
C.fillText("punkty: " + PUNKTY, 2, 20);
GWI.ruch();
```

- Do dokumentu JS, do funkcji LASER, wpisz instrukcję
`C.fillText("obiekty: " + TK.length, 2, 30);`
na stronę liczbą obiektów do zestrzelenia
- Zapisz dokumenty i odśwież przeglądarkę
- Spróbuj „zestrzelić” przynajmniej połowę kwadratów
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu HTML



Iskry (2)

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.

- Dokument JS, metoda OKWA(), obiekt `this.ruch(){...}`

```
PUNKTY=PUNKTY+1;
if (this.bok>0){
  this.bok=this.bok-3;
}
```

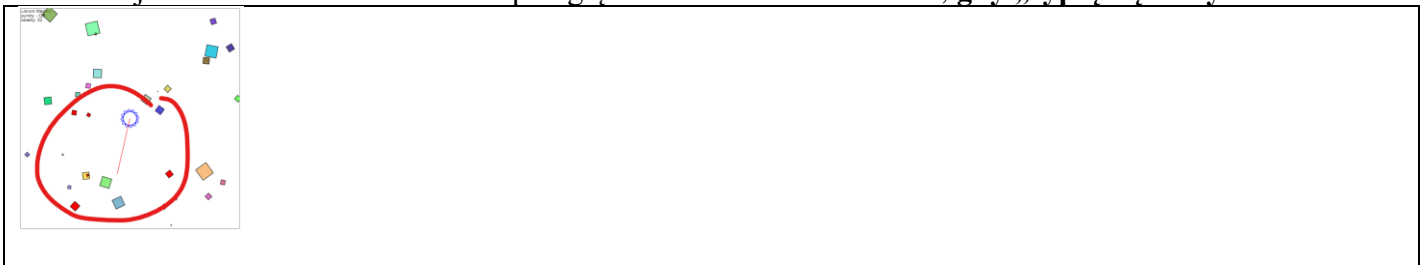
po instrukcji zmniejszania obiektu } wklej tekst z ramki

```
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

```
if (this.kat>360){this.kat=this.kat-360}
this.rysuj();
if (MyszPrzycisk==true) {
```

- Dokument JS, obiekt OKWA(), metoda `this.ruch(){...}`,
wpisz instrukcję `if (this.ISKRA==true && this.bok>0){this.bok=this.bok-0.2}`
gdy obiekt jest iskry, to zmniejszaj go niezależnie czy trafiony
- Zapisz dokumenty i odśwież przeglądarkę
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu HTML, gdy „sypią się iskry”



Czas (2)

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 `var data0=new Date();`
- 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**, we wnętrzu funkcji **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
- **Zestrzeli wszystkie kwadraty**
- Wklej do ramki zrzut ekranu okna przeglądarki i dokumentu HTML.

