

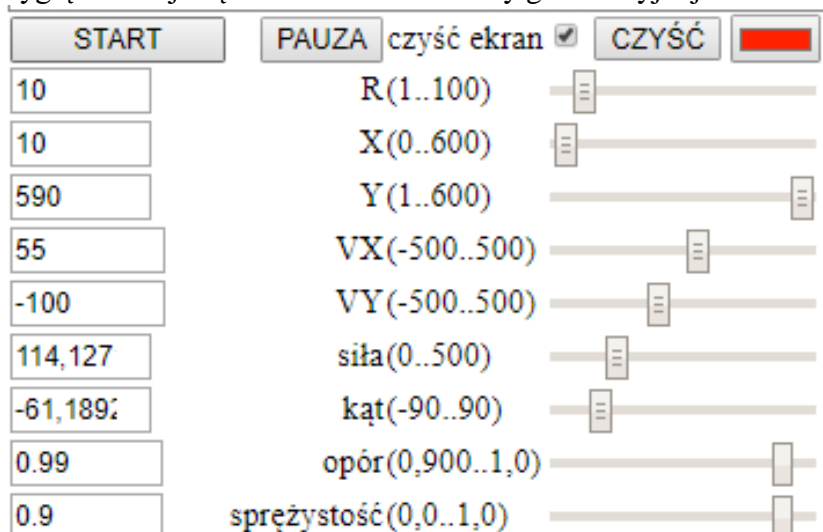
Jak zasymulować grawitację przy użyciu komputera? Niedawno używaliśmy arkusza kalkulacyjnego, a tym razem animowane kulki pojawią się na stronie internetowej. Na czym polega grawitacja w przypadku Ziemi? Wszystkie ziemskie obiekty są przez nią przyciągane i poruszają się z przyspieszeniem, w kierunku jej środka. Równania, które opisują ten ruch można wyprowadzić z najprostszych fizycznych zależności na prędkość i przyspieszenie i opisane zostały w artykule „Jak zrobić grawitację?”, a podstawy w artykule „JavaScript 07 – Grawitacja”. Nieco poszerzona wersja jest uboga w graficzne detale i animowane „wodotryski” – program ma być prosty i zrozumiały dla chętnych do „zabawy” w programowanie i przyjazny dla chcących przeprowadzić eksperyment fizyczny.

Opis programu

Obiekt **canvas** w HTML – szerokość i wysokość 600 pikseli

```
<canvas id="c1" width="600" height="600"></canvas>
```

Wygląd dolnej części – sterownia armaty grawitacyjnej



Przyciski, pola i suwaki w HTML oraz przypisane do nich funkcje

```
<table border=0>
<tr>
<td> <input id="OdNowa" type="button" value="      START      " onclick=OdNowa()> </td>
<td align=right> <input id="StartStop" type="button" value="PAUZA" onclick=StartStop()> </td>
<td align=right>czyść ekran</td>
<td>
<input id="rys0" type="checkbox">
<input id="CzyscEkran" type="button" value="CZYŚĆ" onclick=CzyscEkran()>
<input id="kol0" type="color" value="#ff0000" onchange=ZmianaKoloru()>
</td>
</tr>
</tr>
<tr>
<td> <input id="r0" type="number" width=100 min=1 max=100 onchange=PolaSuwaki()> </td>
<td align=right>R</td>
<td>(1..100)</td>
<td> <input type="range" id="Rr0" min=1 max=100 onchange=SuwakiPola()> </td>
</tr>
</tr>
<tr>
<td> <input id="x0" type="number" min=0 max=600 onchange=PolaSuwaki()> </td>
<td align=right>X</td>
<td>(0..600)</td>
<td> <input type="range" id="Rx0" min=0 max=600 onchange=SuwakiPola()> </td>
</tr>
</tr>
<tr>
<td> <input id="y0" type="number" min=0 max=600 onchange=PolaSuwaki()> </td>
<td align=right>Y</td>
<td>(1..600)</td>
<td> <input type="range" id="Ry0" min=0 max=600 onchange=SuwakiPola()> </td>
</tr>
</tr>
<tr>
<td><input id="vx0" type="number" min=-500 max=500 onchange=PolaSuwaki()></td>
<td align=right>VX</td>
<td> (-500..500)</td>
<td><input type="range" id="Rvx0" min=-500 max=500 onchange=SuwakiPola()></td>
</tr>
</tr>
<tr>
<td><input id="vy0" type="number" min=-500 max=500 onchange=PolaSuwaki()></td>
<td align=right>VY</td>
<td>(-500..500)</td>
<td><input type="range" id="Rvy0" min=-500 max=500 onchange=SuwakiPola()> </td>
</tr>
</tr>
<tr>
<td><input id="Fw0" type="number" min=0 max=500 onchange=PolaSila()></td>
<td align=right>siła</td>
<td>(0..500)</td>
<td><input type="range" id="RFw0" min=0 max=500 onchange=SilaPola()> </td>
</tr>
</tr>
<tr>
<td><input id="Fk0" type="number" min=-90 max=90 onchange=PolaSila()></td>
<td align=right>kąt</td>
<td>(-90..90)</td>
<td><input type="range" id="RFk0" min=-90 max=90 onchange=SilaPola()> </td>
</tr>
</tr>
<tr>
<td><input id="op0" type="text" size=6 onchange=PolaSuwaki()></td>
<td align=right>opór</td>
<td> (0,900..1,0)</td>
<td><input type="range" id="Rop0" min=900 max=1000 onchange=SuwakiPola()></td>
</tr>
</tr>
<tr>
<td><input id="sp0" type="text" size=6 onchange=PolaSuwaki()></td>
<td align=right>sprężystość</td>
<td>(0,0..1,0)</td>
<td><input type="range" id="Rsp0" min=0 max=1000 onchange=SuwakiPola()> </td>
</tr>
</tr>
</table>>
```

Cała reszta programu to skrypt w JS

Zmienna **c** symbolizuje **canvas** i poprzez nią będziemy się odwoływali do wszystkich graficznych elementów
Zmienne **w** i **h** – szerokość i wysokość zdefiniowanego w HTML obszaru rysowania
Zmienne **skok** i **TIM** są niezbędne podczas animacji

```
<script>
var c = c1.getContext('2d')
var w = c.canvas.width;
var h = c.canvas.height;
var skok = 10;
var TIM;>
```

Kulka – pocisk armatni – którą będziemy animować. Zmienne **x** i **y** oznaczają środek kulki, **r** – promień i ostatni parametr to kolor kulki.

```
function pilka(x,y,r,kolor){
c.beginPath();
c.strokeStyle=kolor;
c.fillStyle=kolor;
c.arc(x, y, r,0, Math.PI * 2);
c.stroke();
c.fill();
}
```

Parametry początkowe pierwszego strzału: **x**, **y**, **r**, **k** oraz początkowa szybkość: **vx** i **vy**. Zmienna **dt** to przyrost czasu, którym możemy regulować szybkość przemieszczania się kulki. **g** – przyspieszenie ziemskie. **opor** – opór powietrza (1.0 – ruch bez oporu). **spr** – Sprężystość odbić (1.0 – odbicia sprężyste). **animuj** ustawione na **true** (prawda) – animacja jest kontynuowana, **false** (fałsz) - pauza

```
var x=r;
var y=h-r;
var r=10;
var k="red";
var vx=55;
var vy=-100;
var dt=0.1;
var g=9.81;
var opor=0.99;
var spr=0.9;
var animuj=true;
```

Dwie funkcje (i zmienne) pomocnicze, które obliczają długość przeciwprostokątnej i kąt nachylenia przeciwprostokątnej w trójkącie prostokątnym utworzonym przez składowe **fx** i **fy**. Z ich pomocą wyliczamy szybkość wystrzeliwania kulki z armaty i kąt pochylenia armaty.

```
function Fsil(fx,fy){
return Math.pow(parseFloat(Math.pow(fx,2)) + parseFloat(Math.pow(fy,2)),0.5);
}
var Fw=Fsil(vx,vy);

function Fkat(fx,fy){
return (Math.atan(fy/fx)*180)/Math.PI;
}
var Fk=Fkat(vx,vy);
```

Nasza grawitacyjna armata, to dwa kółka i odcinek je łączący. Położenie, długość i kąt nachylenia armaty wyliczany jest na podstawie wartości pobieranych z pól tekstowych **x0**, **y0**, **vx0**, **vy0**.

```
function RysujArmate(){
var wx0=parseFloat(x0.value);
var wy0=parseFloat(y0.value);
var wvx0=parseFloat(vx0.value);
var wvy0=parseFloat(vy0.value);
var wx=wx0+wvx0;
var wy=wy0+wvy0;
pilka(wx0,wy0,2,"black");
c.beginPath();
c.strokeStyle="black";
c.moveTo(wx0,wy0);
c.lineTo(wx,wy);
c.stroke();
pilka(wx,wy,1,"black");
}
```

Obsługa przycisków START,
PAUZA i CZYŚĆ

```
function OdNowa(){
    animuj=true;
    PolaZmienne();
    czas=setTimeout(grawitacja,skok);
}
function StartStop(){
    animuj=!animuj;
    if (animuj) {czas=setTimeout(grawitacja,skok)}
    clearTimeout(TIM);
}
function CzyscEkran(){
    c.clearRect(0, 0, w, h);
    c.strokeStyle="black";
    c.strokeRect(0, 0, w, h);
    RysujWektory();
};
```

Jeśli zmianie ulegną się podstawowe parametry strzelającej armaty, kulki lub środowiska (r,x,y,vx,vy,opor, spr), to ich wartości wpisujemy do pól tekstowych i jednocześnie ustawiamy suwaki. Po każdej zmianie szybkości wyliczamy na nowo siłę wyrzutu i kąt armaty.

Jeśli zmienimy parametry programu wpisując je ręcznie do pól tekstowych, to zmieniamy odpowiadające im zmienne i wyliczamy siłę i kąt armaty.

```
function PolaZmienne(){
    r=parseFloat(r0.value);
    x=parseFloat(x0.value);
    y=parseFloat(y0.value);
    vx=parseFloat(vx0.value);
    vy=parseFloat(vy0.value);
    Fw=Fsila(vx,vy);
    Fk=Fkat(vx,vy);
    Fw0.value=Fw;
    Rfw0.value=Fw;
    Fk0.value=Fk;
    Rfk0.value=Fk;
    opor=parseFloat(op0.value);
    spr=parseFloat(sp0.value);
}
```

```
function ZmiennePola(){
    r0.value=r;
    Rr0.value=r;
    x0.value=x;
    Rx0.value=x;
    y0.value=y;
    Ry0.value=y;
    vx0.value=vx;
    Rvx0.value=vx;
    vy0.value=vy;
    Rvy0.value=vy;
    Fw=Fsila(vx,vy);
    Fk=Fkat(vx,vy);
    Fw0.value=Fw;
    Rfw0.value=Fw;
    Fk0.value=Fk;
    Rfk0.value=Fk;
    op0.value=opor;
    Rop0.value=opor*1000;
    sp0.value=spr;
    Rsp0.value=spr*1000;
}
```

Korzystamy z suwaków – wartości na nich ustawione zapisujemy w polach tekstowych oraz wyliczamy siłę i kąt armaty.

Ręczne wpisanie parametrów do pól tekstowych powinno wymusić zmianę ustawień na suwakach.

```
function PolaSuwaki(){
    Rr0.value=r0.value;
    r=r0.value;
    Rx0.value=x0.value;
    Ry0.value=y0.value;
    Rvx0.value=vx0.value;
    Rvy0.value=vy0.value;
    Rop0.value=op0.value*1000;
    Rsp0.value=sp0.value*1000;
}
```

```
function SuwakiPola(){
    r0.value=Rr0.value;
    r=r0.value;
    x0.value=Rx0.value;
    y0.value=Ry0.value;
    vx0.value=Rvx0.value;
    vy0.value=Rvy0.value;
    Fw=Fsila(vx0.value,vy0.value);
    Fk=Fkat(vx0.value,vy0.value);
    Fw0.value=Fw;
    Fk0.value=Fk;
    Rfw0.value=Fw0.value;
    Rfk0.value=Fk0.value;
    op0.value=Rop0.value/1000;
    sp0.value=Rsp0.value/1000;
}
```

Zmieniając siłę i kąt armaty musimy wyliczyć z funkcji trygonometrycznych składowe szybkości i wstawić je do pól i na suwaki **vx** i **vy**.

Wpisujemy wartości ręcznie do pól tekstowych **siła** i **kąt** – konieczne należy wyliczyć na nowo składowe szybkości.

```
function PolaSila(){
  var sila=Fw0.value;
  var kat=Fk0.value;
  Rfw0.value=sila;
  Rfk0.value=kat;
  vx=Math.cos(kat*Math.PI/180)*sila;
  vy=Math.sin(kat*Math.PI/180)*sila;
  Rvx0.value=vx;
  Rvy0.value=vy;
  vx0.value=vx;
  vy0.value=vy;
}
```

```
function SilaPola(){
  var sila=Rfw0.value;
  var kat=Rfk0.value;
  Fw0.value=sila;
  Fk0.value=kat;
  vx=Math.cos(kat*Math.PI/180)*sila;
  vy=Math.sin(kat*Math.PI/180)*sila;
  vx0.value=vx;
  vy0.value=vy;
  Rvx0.value=vx;
  Rvy0.value=vy;
}
```

Funkcja realizująca animację (rekurencyjnie).

rys0.checked – jeśli zaznaczone pole **czyść ekran**, to kulka nie jest wymazywana.

Poniżej rysowana jest czarna ramka, armata i kulka.

x=x+vx*dt; y=y+vy*dt;
vy=vy+g*dt; vx=vx*opor;
vy=vy*opor;

Równania ruchu opisują w kolejności zmianę położenia, zmianę szybkości związaną z grawitacją i zmianę szybkości związaną z oporem powietrza.

if (y>h-r-vy*dt){vy=-(vy-g*dt)*spr}

Odbicie od dolnej krawędzi – uwzględniamy promień kulki i

szybkość z jaką się zbliża. Po odbiciu kulka zmienia kierunek na przeciwny – uwzględniamy sprężystość odbić oraz pomniejszamy o (niepotrzebny) dodatkowy wzrost szybkości.

```
function grawitacja() {
  if (rys0.checked) { c.clearRect(0, 0, w, h);}
  c.strokeStyle="black";
  c.strokeRect(0, 0, w, h);
  RysujArmate();
  pilka(x,y,r,k);

  x=x+vx*dt;
  y=y+vy*dt;
  vy=vy+g*dt;
  vx=vx*opor;
  vy=vy*opor;

  if (y>h-r-vy*dt){vy=-(vy-g*dt)*spr}

  clearTimeout(TIM);
  TIM = setTimeout(grawitacja, skok);
}
grawitacja();
</script>
```

Ostatnia instrukcja programu, to właściwe uruchomienie animacji i zamknięcie skryptu.

Cały program

```
<canvas id="c1" width="600" height="600"></canvas>
<table border=0>
<tr>
    <td>
        <input id="OdNowa" type="button" value="START" onclick=OdNowa()>
    </td>
    <td align=right>
        <input id="StartStop" type="button" value="PAUZA" onclick=StartStop()>
    </td>
    <td align=right>czyść ekran</td>
</tr>
<tr>
    <td>
        <input id="rys0" type="checkbox">
        <input id="CzyscEkran" type="button" value="CZYŚĆ" onclick=CzyscEkran()>
        <input id="kol0" type="color" value="#ff0000" onchange=ZmianaKoloru()>
    </td>
</tr>
</table>
<tr>
    <td><input id="r0" type="number" width=100 min=1 max=100 onchange=PolaSuwaki()></td>
    <td align=right>R</td>
    <td>(1..100)</td>
    <td><input type="range" id="Rr0" min=1 max=100 onchange=SuwakiPola()> </td>
</tr>
<tr>
    <td><input id="x0" type="number" min=0 max=600 onchange=PolaSuwaki()></td>
    <td align=right>X</td>
    <td>(0..600)</td>
    <td><input type="range" id="Rx0" min=0 max=600 onchange=SuwakiPola()> </td>
</tr>
<tr>
    <td><input id="y0" type="number" min=0 max=600 onchange=PolaSuwaki()></td>
    <td align=right>Y</td>
    <td>(1..600)</td>
    <td><input type="range" id="Ry0" min=0 max=600 onchange=SuwakiPola()> </td>
</tr>
<tr>
    <td><input id="vx0" type="number" min=-500 max=500 onchange=PolaSuwaki()></td>
    <td align=right>VX</td>
    <td>(-500..500)</td>
    <td><input type="range" id="Rvx0" min=-500 max=500 onchange=SuwakiPola()></td>
</tr>
<tr>
    <td><input id="vy0" type="number" min=-500 max=500 onchange=PolaSuwaki()></td>
    <td align=right>VY</td>
    <td>(-500..500)</td>
    <td><input type="range" id="Rvy0" min=-500 max=500 onchange=SuwakiPola()> </td>
</tr>
<tr>
    <td><input id="Fw0" type="number" min=0 max=500 onchange=PolaSila()></td>
    <td align=right>siła</td>
    <td>(0..500)</td>
    <td><input type="range" id="RFw0" min=0 max=500 onchange=SilaPola()> </td>
</tr>
<tr>
    <td><input id="Fk0" type="number" min=-90 max=90 onchange=PolaSila()></td>
    <td align=right>kąt</td>
    <td>(-90..90)</td>
    <td><input type="range" id="RFk0" min=-90 max=90 onchange=SilaPola()> </td>
</tr>
<tr>
    <td><input id="op0" type="text" size=6 onchange=PolaSuwaki()></td>
    <td align=right>opór</td>
    <td>(0,900..1,0)</td>
    <td><input type="range" id="Rop0" min=900 max=1000 onchange=SuwakiPola()></td>
</tr>
<tr>
    <td><input id="sp0" type="text" size=6 onchange=PolaSuwaki()></td>
    <td align=right>sprężystość</td>
    <td>(0,0..1,0)</td>
    <td><input type="range" id="Rsp0" min=0 max=1000 onchange=SuwakiPola()> </td>
</tr>
</table>
<script>
var c = c1.getContext('2d')
var w = c.canvas.width;
var h = c.canvas.height;
var skok = 10;
var TIM;

function pilka(x,y,r,kolor){
    c.beginPath();
    c.strokeStyle=kolor;
```

```

        c.fillStyle=kolor;
        c.arc(x, y, r,0, Math.PI * 2);
        c.stroke();
        c.fill();
    }
    var r=10;
    var k="red";

    var x=r;
    var y=h-r;
    var vx=55;
    var vy=-100;

    var dt=0.1; //przyrost czasu
    var g=9.81;
    var opor=0.99;
    var spr=0.9;

    var animuj=true;
    rys0.checked=true;

    function Fsila(fx,fy){
        return Math.pow(parseFloat(Math.pow(fx,2))
            + parseFloat(Math.pow(fy,2)),0.5);
    }
    var Fw=Fsila(vx,vy);

    function Fkat(fx,fy){
        return (Math.atan(fy/fx)*180)/Math.PI;
    }
    var Fk=Fkat(vx,vy);

    function ZmianaKoloru(){
        k=kol0.value;
    }

    //położenie początkowe i wektor prędkości
    function RysujArmatae(){
        var Wx0=parseFloat(x0.value);
        var Wy0=parseFloat(y0.value);
        var Wvx0=parseFloat(vx0.value);
        var Wvy0=parseFloat(vy0.value);
        var Wx=Wx0+Wvx0;
        var Wy=Wy0+Wvy0;
        pilka(Wx0,Wy0,2,"black");
        c.beginPath();
        c.strokeStyle="black";
        c.moveTo(Wx0,Wy0);
        c.lineTo(Wx,Wy);
        c.stroke();
        pilka(Wx,Wy,1,"black");
    }

    function StartStop(){
        animuj=!animuj;
        if (animuj)
            {czas=setTimeout(grawitacja,skok)}
        clearTimeout(TIM);
    }

    function OdNowa(){
        animuj=true;
        PolaZmienne();
        czas=setTimeout(grawitacja,skok);
    }

    function CzyscEkran(){
        c.clearRect(0, 0, w, h);
        c.strokeStyle="black";
        c.strokeRect(0, 0, w, h);
        RysujWektory();
    }

    //zmienne przenosimy do pól tekstowych i na suwaki
    function ZmiennePola(){
        r0.value=r;          Rr0.value=r;
        x0.value=x;         Rx0.value=x;
        y0.value=y;         Ry0.value=y;
        vx0.value=vx;       Rvx0.value=vx;
        vy0.value=vy;       Rvy0.value=vy;
        //zmieniamy suwaki prędkości - zmiana na sile i kącie
        Fw=Fsila(vx,vy);
        Fk=Fkat(vx,vy);
        Fw0.value=Fw;       RfW0.value=Fw;
        Fk0.value=Fk;       RfK0.value=Fk;
        op0.value=opor;     Rop0.value=opor*1000;
    }

```

```

        sp0.value=spr; Rsp0.value=spr*1000;
    }
    ZmiennePola();//za pierwszym razem
    //z pól tekstowych na zmienne - zamiana na liczby
    function PolaZmienne(){
        r=parseFloat(r0.value);
        x=parseFloat(x0.value);
        y=parseFloat(y0.value);
        vx=parseFloat(vx0.value);
        vy=parseFloat(vy0.value);
        //wpisano w pola to też obliczamy siły od nowa
        Fw=Fsila(vx,vy);
        Fk=Fkat(vx,vy);
        Fw0.value=Fw;   Rfw0.value=Fw;
        Fk0.value=Fk;   Rfk0.value=Fk;
        opor=parseFloat(op0.value);
        spr=parseFloat(sp0.value);
    }
    //z suwaków na pola tekstowe - do zmiennych nie trzeba bo automat w programie
    function SuwakiPola(){
        r0.value=Rr0.value;
        r=r0.value;
        x0.value=Rx0.value;
        y0.value=Ry0.value;
        vx0.value=Rvx0.value;
        vy0.value=Rvy0.value;
        //zmiana vx lub vy to zmiana siły i kąta
        Fw=Fsila(vx0.value,vy0.value);
        Fk=Fkat(vx0.value,vy0.value);
        Fw0.value=Fw;
        Fk0.value=Fk;
        Rfw0.value=Fw0.value;
        Rfk0.value=Fk0.value;
        op0.value=Rop0.value/1000;
        sp0.value=Rsp0.value/1000;
    }
    //zmiana pól powoduje zmianę na suwakach
    //opór i sprężystość na zwykłych polach tekstowych
    //bez siły i kąta - osobno
    function PolaSuwaki(){
        Rr0.value=r0.value;
        r=r0.value;
        Rx0.value=x0.value;
        Ry0.value=y0.value;
        Rvx0.value=vx0.value;
        Rvy0.value=vy0.value;
        Rop0.value=op0.value*1000;
        Rsp0.value=sp0.value*1000;
    }
    //zmiana siły i kąta na suwakach - obliczyć i wstawić do pól vx i vy
    function SilaPola(){
        var sila=Rfw0.value;
        var kat=Rfk0.value;
        Fw0.value=sila;
        Fk0.value=kat;
        //obliczyć vx i vy
        vx=Math.cos(kat*Math.PI/180)*sila;
        vy=Math.sin(kat*Math.PI/180)*sila;
        //do pól i na suwaki vx i vy
        vx0.value=vx;
        vy0.value=vy;
        Rvx0.value=vx;
        Rvy0.value=vy;
    }
    //zmiana w polach - powoduje zmianę na suwakach vx i vy
    function PolaSila(){
        var sila=Fw0.value;
        var kat=Fk0.value;
        Rfw0.value=sila;
        Rfk0.value=kat;
        vx=Math.cos(kat*Math.PI/180)*sila;
        vy=Math.sin(kat*Math.PI/180)*sila;
        //do pól i na suwaki vx i vy
        Rvx0.value=vx;
        Rvy0.value=vy;
        vx0.value=vx;
        vy0.value=vy;
    }
    //pętla główna
    function grawitacja() {
        if (rys0.checked) {
            c.clearRect(0, 0, w, h);
        }
    }

```



```
c.strokeStyle="black";
c.strokeRect(0, 0, w, h);
RysujArmata();
pilka(x,y,r,k);

x=x+vx*dt;
y=y+vy*dt;
vy=vy+g*dt;
vx=vx*opor;
vy=vy*opor;

    //przy odbiciu od ziemi uwzględniamy
    //promień kulki i prędkość kulki
    //gdy się odbija to nie zwiększamy prędkości
    //dlatego jest za każdym razem ujmowana -g*dt
if (y>h-r-vy*dt){vy=-(vy-g*dt)*spr}

clearTimeout(TIM);
TIM = setTimeout(grawitacja, skok);
}
grawitacja();
</script>
```