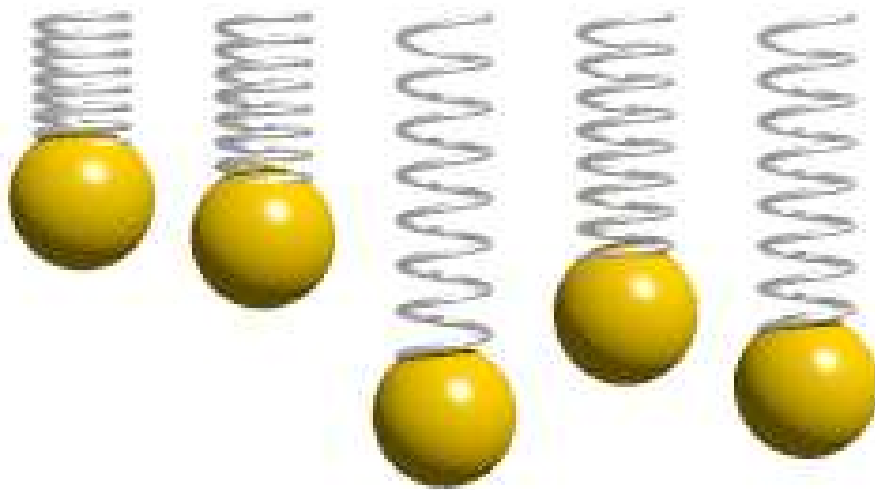


## Scéna 1:

Pro svou první scénu jsem si vybral kuličky zavěšené na pružině, kdy se ve výsledné animaci tyto kuličky nerovnoměrně pohybují. Scéna je volena bez pozadí, které se objeví až se scénou číslo 2.



## Popis objektů:

Jak dle obrázku vidíme, ve scéně se nachází 5 kuliček zavěšených na pružině, které se natahují a smršťují dle pohybu kuliček.

Pozadí je zvoleno bílé, v této scéně se budu věnovat pouze konfiguraci kuliček a jejich pružin.

## Popis souboru pruzinka.pov:

```
#version 3.6;

global_settings{ assumed_gamma 1.0 }

#default{ finish{ ambient 0.1 diffuse 0.9 conserve_energy}}

#include "colors.inc"

#include "textures.inc"

// include příkazem nahráváme knihovny s využitelnými balíčky pro naše scény
// deklarace kamery, v tomto případě máme pouze jednu, kterou touto deklarací
// definujeme

#declare Camera_0 = camera {angle 25

    location <0.0 , 1.0 ,-15>

    right   x*image_width/image_height

    look_at <0.0 , 1.0 , 0.0>}

camera{Camera_0}

// light source nám v naší scéně definuje sluneční svit, jelikož máme pozadí bílé, tak je
// v podstatě bezpředmětné pro tuto danou scénu

light_source{< 1500,2500,-2500> color White}

// sky jako nebe či oblohu, máme nastavenou bílou barvu

sky_sphere { pigment { color rgb <1.0,1.0,1.0>}}

// Níže začínáme se samotnou deklarací objektů pro naši scénu

// hodnotami níže nastavujeme sinusový průběh pohybů pružin a kuliček

#declare Amplitude = 0.60 ;

#declare Minimal_Length = 0.80 ;

#declare Middle_Length = Amplitude + Minimal_Length ;

// ve scéně máme 5 kuliček a pružin, pro každou proto nastavíme i časově sinusový průběh
// budeme-li chtít současný pohyb kuliček, tak bude stačit nastavit pouze jednu časovou
// proměnnou hodnotou za "Time_testX" nastavíme napnutí pružiny a sinusovou polohu
// kuličky v našem případě lze užít pouze hodnotu od 0.25 do 0.75
```

```

#declare Time_test1 = 0.73;
#declare Sp_Length1 = Middle_Length+Amplitude*sin((clock+Time_test1)*2*pi);
#declare Time_test2 = 0.61;
#declare Sp_Length2 = Middle_Length+Amplitude*sin((clock+Time_test2)*2*pi);
#declare Time_test3 = 0.49;
#declare Sp_Length3 = Middle_Length+Amplitude*sin((clock+Time_test3)*2*pi);
#declare Time_test4 = 0.37;
#declare Sp_Length4 = Middle_Length+Amplitude*sin((clock+Time_test4)*2*pi);
#declare Time_test5 = 0.25;
#declare Sp_Length5 = Middle_Length+Amplitude*sin((clock+Time_test5)*2*pi);

// následuje deklarace podoby a základní polohy samotných pružin a jejich kuliček
#declare Spiral1 =
union{
#local N_per_Rev = 500;

// k určení počtu prvků na jednu otáčku
#local N_of_Rev = 8.00;

// k určení počtu spirál, otáček na pružině
#local H_per_Ref = Sp_Length1 / N_of_Rev;

// definice lokálního výpočtu délky jedné otáčky
#local Nr = 0;

// počáteční smyčka pružiny

#while (Nr < N_per_Rev*N_of_Rev)
  sphere{ <0,0,0>,0.025

// poloha a průměr pružiny
  translate<0.25, -Nr*H_per_Ref/N_per_Rev, 0>
  rotate<0, Nr * 360/N_per_Rev,0>

// k určení pokračující polohy a rotace
  texture{ Chrome_Metal
    finish { phong 1}}

// grafická podoba pružinky

```

```

#local Nr = Nr + 2;

// k zajištění sinusového pohybu, výše máme uveden příkaz ``WHILE``, tedy dělej, dokud
// nenastane v našem případě se bude pružinka s kuličkou natahovat či smršťovat tak
// dlouho dokud nenarazí na svou maximální či minimální polohu definovanou
// ``Time_testX``

#end

// konec ``WHILE`` první spirálové pružinky

sphere { <0,0,0>, 0.4

    translate<0,-Nr*H_per_Ref/N_per_Rev-0.2,0>

    texture{ pigment{ color rgb<1,0.65,0>}

        finish { phong 1}}

// k určení umístění, následného pohybu a samotné podoby přidružené kuličky

}}

// konec těla celé definice spirály s navazující pružinou

// jelikož se všechny zobrazené spirály se liší pouze nastavením sinusového pohybu, popis
// níže vypsanych deklarací je totožný s první deklarací spirály, změnou je pouze volání jiné
// deklarace sinusového pohybu

#declare Spiral2 =

union{

#local N_per_Rev = 500;

#local N_of_Rev = 8.00;

#local H_per_Ref = Sp_Length2 / N_of_Rev;

#local Nr = 0;

#while (Nr< N_per_Rev*N_of_Rev)

    sphere{ <0,0,0>,0.025

        translate<0.25, -Nr*H_per_Ref/N_per_Rev, 0>

        rotate<0, Nr * 360/N_per_Rev,0>

        texture{ Chrome_Metal

            finish { phong 1}}

```

```

#local Nr = Nr + 2;

#end

sphere { <0,0,0>, 0.4
    translate<0,-Nr*H_per_Ref/N_per_Rev-0.2,0>
    texture{ pigment{ color rgb<1,0.65,0>}
        finish { phong 1}}
    }}

#declare Spiral3 =
union{

#local N_per_Rev = 500;

#local N_of_Rev = 8.00;

#local H_per_Ref = Sp_Length3 / N_of_Rev;

#local Nr = 0;

#while (Nr< N_per_Rev*N_of_Rev)
    sphere{ <0,0,0>,0.025
        translate<0.25, -Nr*H_per_Ref/N_per_Rev, 0>
        rotate<0, Nr * 360/N_per_Rev,0>
        texture{ Chrome_Metal
            finish { phong 1}}
    }
#local Nr = Nr + 2;

#end

sphere { <0,0,0>, 0.4
    translate<0,-Nr*H_per_Ref/N_per_Rev-0.2,0>
    texture{ pigment{ color rgb<1,0.65,0>}
        finish { phong 1}}
    }}

#declare Spiral4 =
union{

#local N_per_Rev = 500;

#local N_of_Rev = 8.00;

#local H_per_Ref = Sp_Length4 / N_of_Rev;

```

```

#local Nr = 0;
#while (Nr < N_per_Rev*N_of_Rev)
  sphere{ <0,0,0>,0.025
    translate<0.25, -Nr*H_per_Ref/N_per_Rev, 0>
    rotate<0, Nr * 360/N_per_Rev,0>
    texture{ Chrome_Metal
      finish { phong 1}}}
#local Nr = Nr + 2;
#end
sphere { <0,0,0>, 0.4
  translate<0,-Nr*H_per_Ref/N_per_Rev-0.2,0>
  texture{ pigment{ color rgb<1,0.65,0>}
    finish { phong 1}}
}
#declare Spiral5 =
union{
#local N_per_Rev = 500;
#local N_of_Rev = 8.00;
#local H_per_Ref = Sp_Length5 / N_of_Rev;
#local Nr = 0;
#while (Nr < N_per_Rev*N_of_Rev)
sphere{ <0,0,0>,0.025
  translate<0.25, -Nr*H_per_Ref/N_per_Rev, 0>
  rotate<0, Nr * 360/N_per_Rev,0>
  texture{ Chrome_Metal
    finish { phong 1}}}
#local Nr = Nr + 2;
#end

```

```
sphere { <0,0,0>, 0.4
    translate<0,-Nr*H_per_Ref/N_per_Rev-0.2,0>
    texture{ pigment{ color rgb<1,0.65,0>}
        finish { phong 1}}
    }
```

**// konec popisu a definice objektů pro naši scénu**

```
object { Spiral1 translate< -2.0,2.42,0>}
object { Spiral2 translate< -1.0,2.42,0>}
object { Spiral5 translate< 0.0,2.42,0>}
object { Spiral3 translate< 1.0,2.42,0>}
object { Spiral4 translate< 2.0,2.42,0>}
```

**// nakonec si všechny objekty vyvoláme společně s určením jejich pozice na scéně**

### Popis pruzinka.ini:

Antialias=On

Antialias\_Threshold=0.3

Antialias\_Depth=3

**// vyhlazovací technika snímkování**

Input\_File\_Name=pruzinka.pov

**// vstupní soubor k snímkování**

Initial\_Frame=1

Final\_Frame=30

**// zde dáváme programu vědět kolik si přejeme snímků, v našem případě 30 snímků**

Initial\_Clock=0

Final\_Clock=1

**// definice proměnné k ustanovení časového formátu**

Cyclic\_Animation=on

**// požadavek na cyklickou animaci**

Pause\_When\_Done=off

**// ukončit máme-li vyrendrováno**