

Korisnička sučelja

KORISNIČKA SUČELJA

Aleksandar Maksimović
IRB

EJS

➤ Easy Java Simulations

- freeware, open source
- alat napravljen u Javi
- generira html stranice
- interaktivni virtualni laboratorij

➤ Autor

- Prof. Dr. Francisco Esquembre

➤ Prednosti

- Jednostavna instalacija i uporaba
- slabo znanje programiranja
- navodi korisnika pri stvaranju virtualnog lab-a.
- generira Java programe kao applete, mogu se uključiti u HTML stranice

Dizajn

- ➔ model
 - ➔ opisuje fenomene pomoću jednadžbi i/ili algoritama
- ➔ kontrola
 - ➔ interakcija korisnika s modelom, promjena vrijednosti varijabli
- ➔ view
 - ➔ GUI od modela

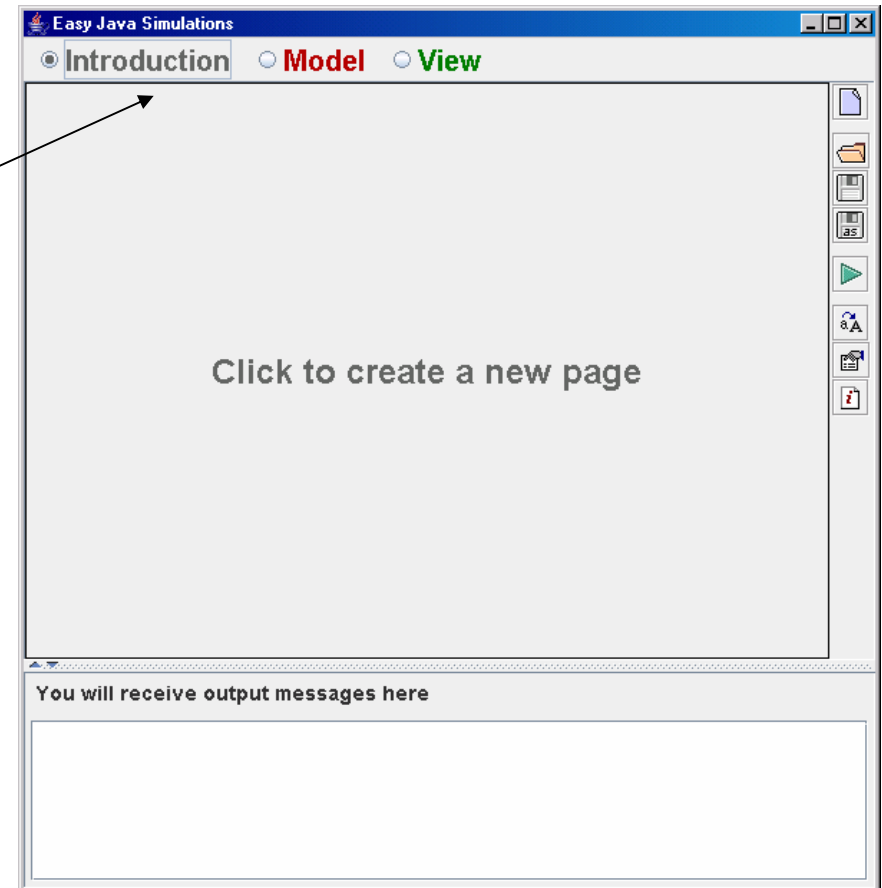
EJS ima kontrolu i GUI integrirane. Interakcija korisnika i modela ostvarena preko GUI-a

ejs

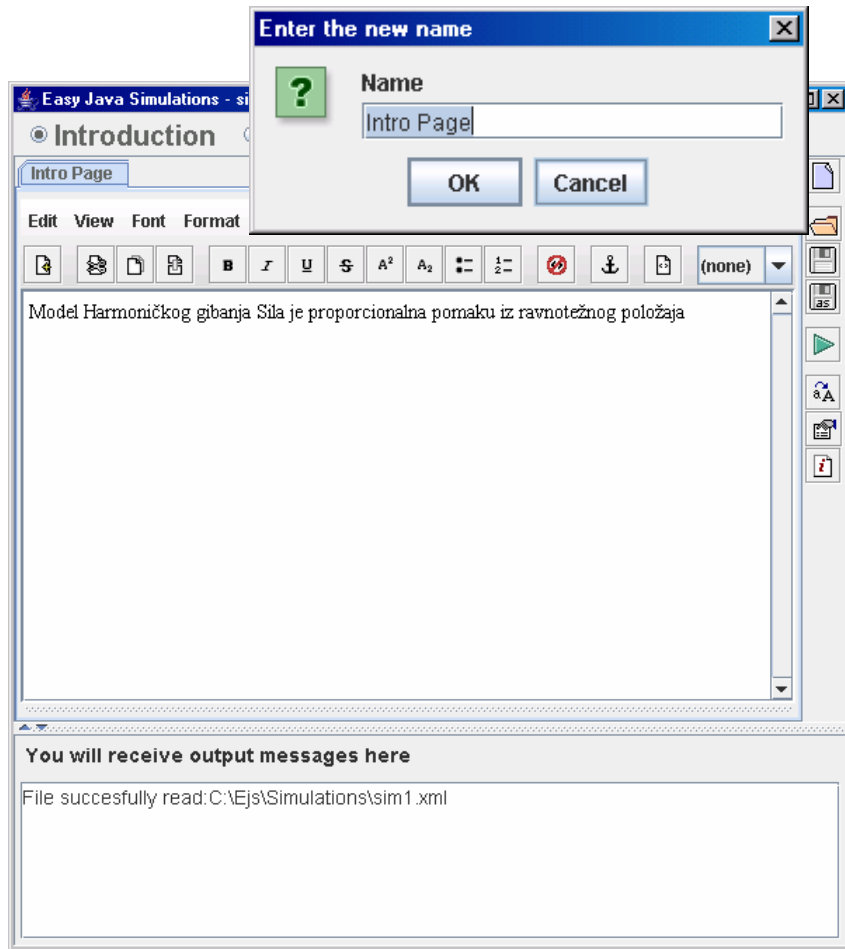
➤ virtualni lab

ostvaruje se pomoću tri panela
odabiru se pomoću tri radio
gumba

- Introduction (opis modela)
- Model - definiamo model, varijable, jednažbe
- View - konstruiramo GUI pomoću definiranih elemenata

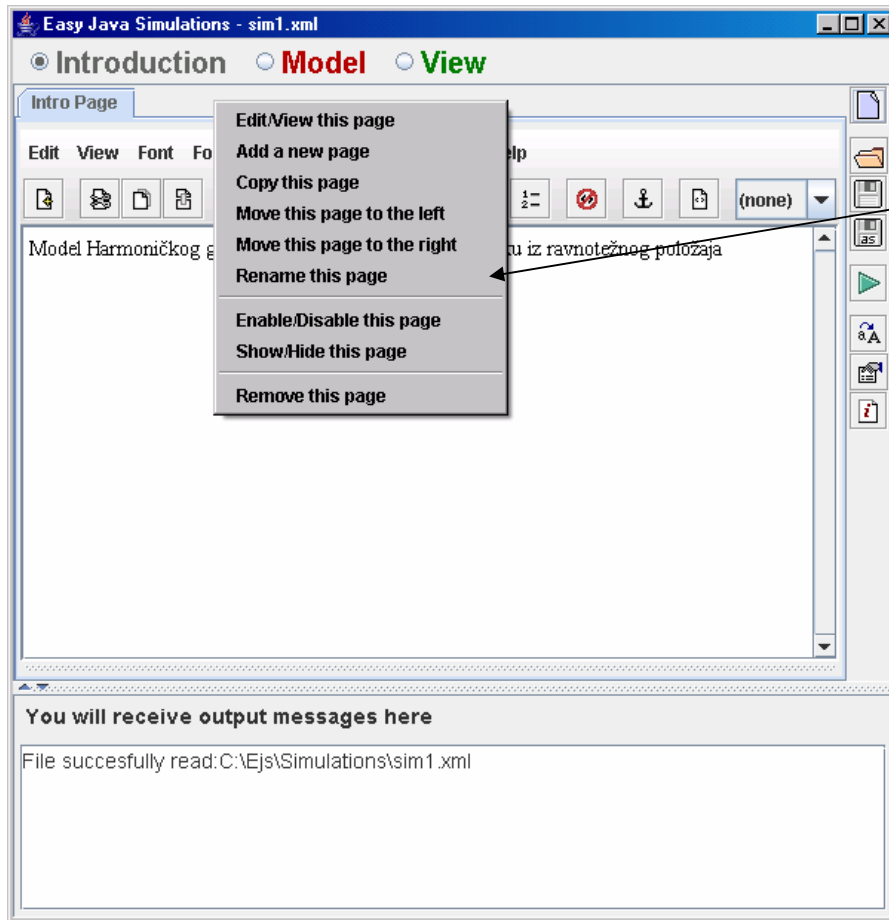


uvod



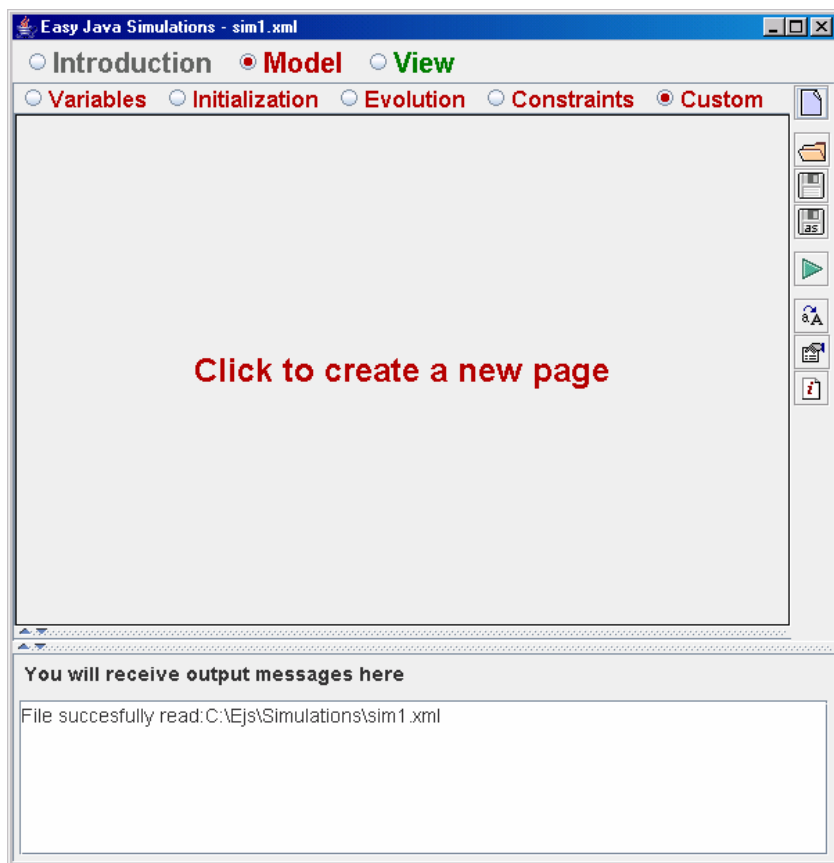
- jednostavan editor
- rezultat je HTML stranica
- možemo napraviti proizvoljan broj stranica, npr. s opisom modela, zadatka i informacija o autoru.
- nove stranice dobivamo pritiskom desnog gumba na mišu dok je kursor na naslovu

izbornik uvodne stranice



Izbornik s opcijama za
manipuliranje uvodne stranice
Edit, Copy, Move, promjena imena
Prikazivanje i skrivanje stranice

Model

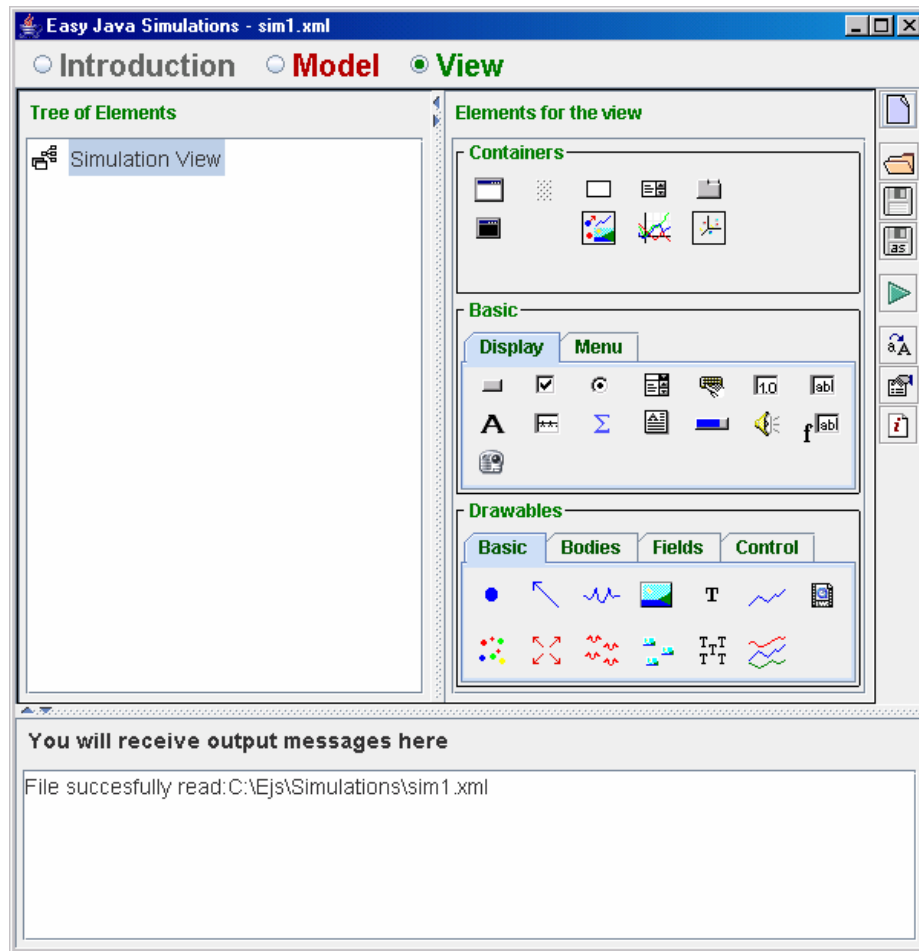


5 panela za opisivanje modela

- Variables
- Initialization
- Evolution
- Constraints
- Custom

Postoji mogućnost povezivanja s Matlab/Simulink programom

GUI



➤ View

➤ konstrukcija GUI-a

➤ Elements for the view

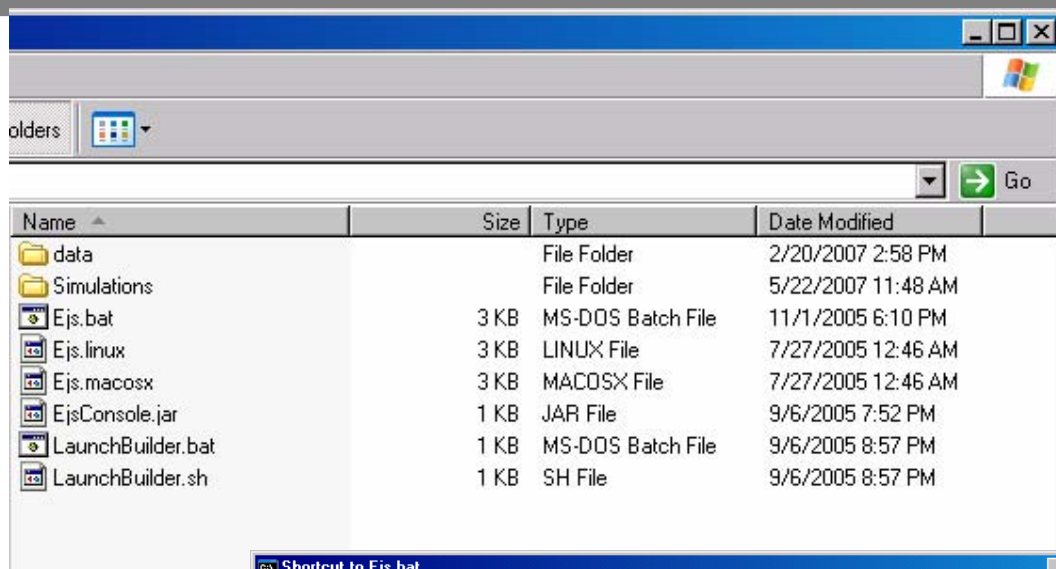
➤ sadrži gotove vizualne elemente (u wxpythonu widgeti)

➤ Tree of Elements

➤ pokazuje odabrane elemente i hijerarhiju

Odaberemo element pomoću miša, nakon toga možemo ga staviti u lijevi prozor (TREE)

Pokretanje



Windows - Ejs.bat

Linux - Ejs.linux

MacOsx - Ejs.macosx

```
Shortcut to Ejs.bat
C:\Ejs>"C:\Program files\Java\jdk1.5.0_04\bin\java" -classpath "C:\Program files\Java\jdk1.5.0_04\lib\tools.jar;C:\Program files\Java\jdk1.5.0_04\jre\lib\rt.jar;data\osejs.jar;Simulations\library\HotEgn.jar;Simulations\library\EjsBasic.jar;Simulations\library\external\matlab.jar;Simulations\library\HTTPClient.jar;Simulations\library\EjsContrib.jar;Simulations\library\EjsDisplay2d.jar;Simulations\library\EjsMedia.jar;Simulations\library\QTJava.zip;Simulations\library\EjsWebCam.jar;data\alllocales.jar" -Dcodebase=. -Duser.home="Simulations" org.colos.ejs.osejs.Osejs -iconsAt East -elements elementsOrdered.txt
```

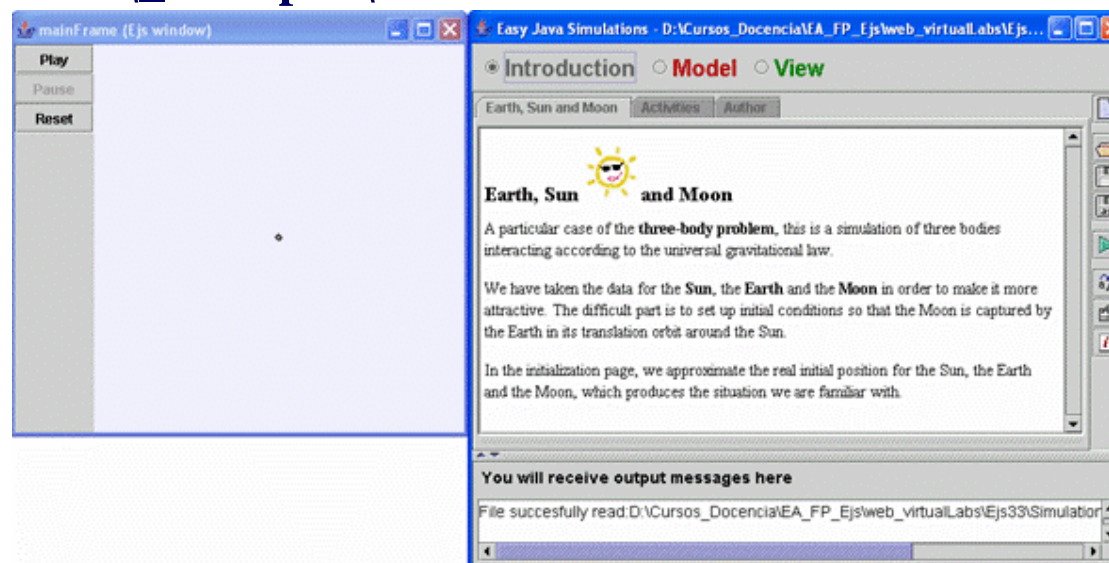
Primjer


Otvori file "Open an existing file" u Ejs' GUI,



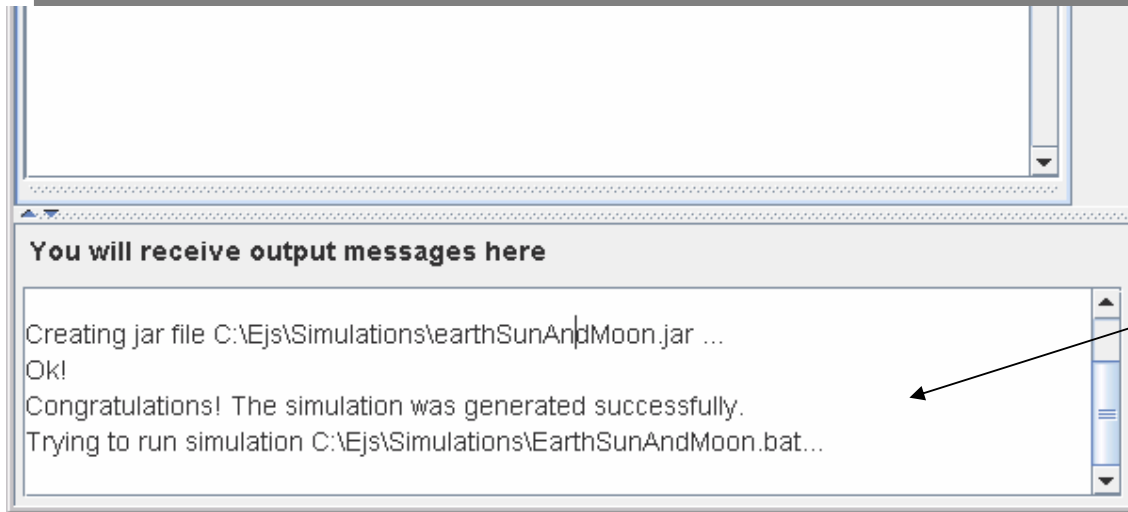
nakon toga odaberi file **Simulations_examples\EarthSunAndMoon.xml**.

Virtualni-lab je otvoren



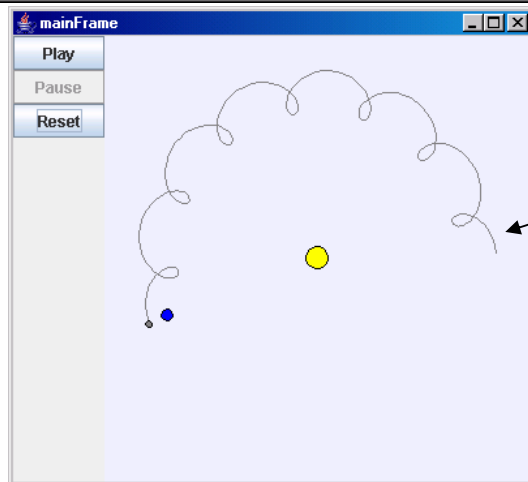
Pritisni gumb "Run Simulation", , program stvori jar file sa simulacijom.

Primjer



Kompajlira java program u jar file, rezultat se prikazuje u prozoru.

Ako je sve u redu, otvara se simulacija, to je još jedan prozor istog oblika kao onaj koji je konstruiran u View panelu.



Gumb zaustavlja simulaciju.

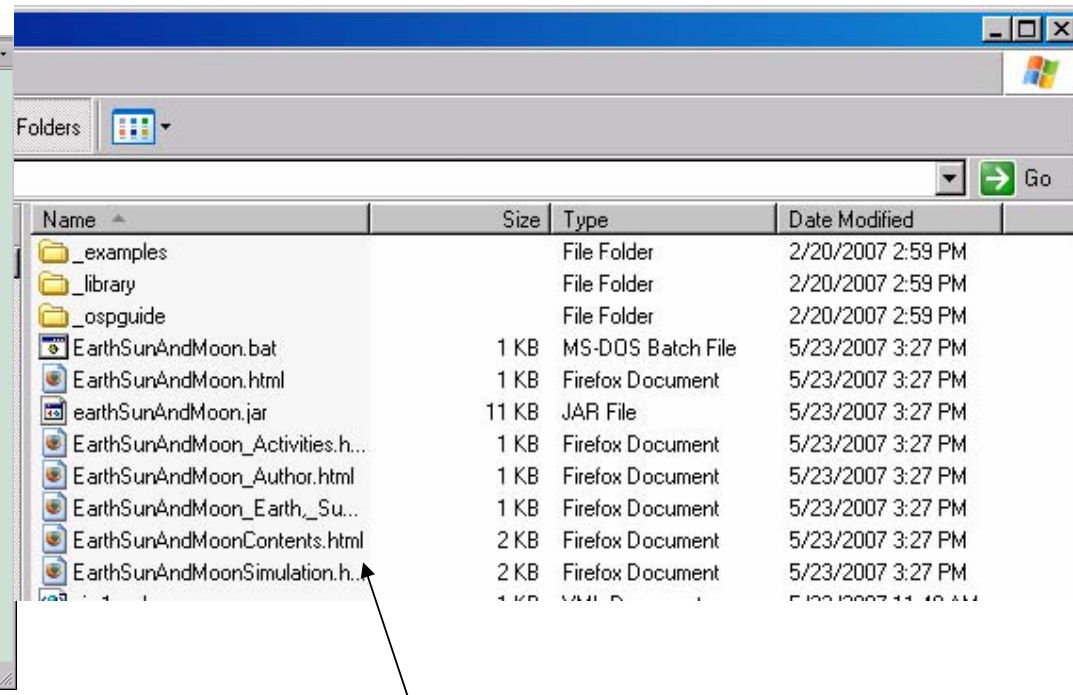
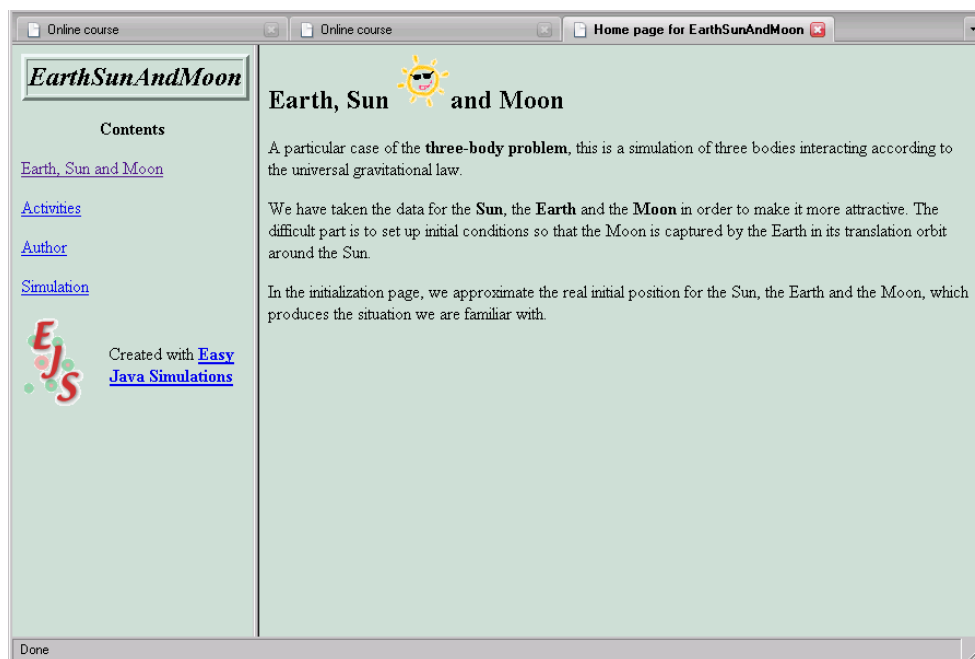
Primjer

- Nakon simulacije, Ejs generira fileove u radnom direktoriju (Simulations)
 - može se promjeniti pomoću argumenata (vidi .bat ili .linux)

➤ Rezultat

- **earthSunAndMoon.jar**: svi Java fileovi u jar-u
- **earthSunAndMoon.html**: html dokument
- **earthSunAndMoon.bat**: batch file simulacije

html stranice



Generirane stranice, batch i jar file

html simulacija

EarthSunAndMoon


Contents

[Earth, Sun and Moon](#)

[Activities](#)

[Author](#)

[Simulation](#)



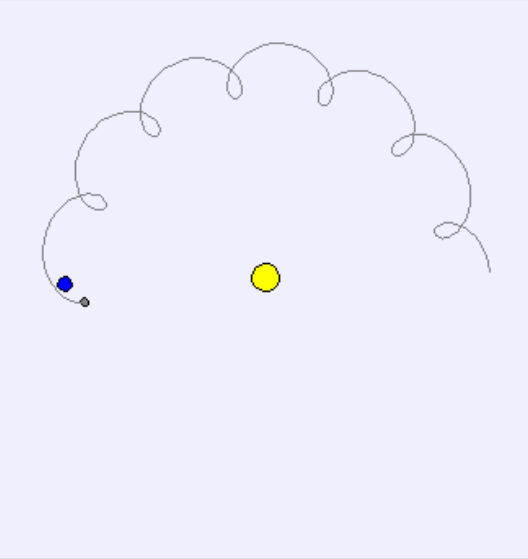
Created with [Easy Java Simulations](#)

The simulation's view should appear right under this line.

Play

Pause

Reset



You can control it using JavaScript. For example, using buttons:

Play Pause Reset Step

Slow Fast Faster

Sadržaj direktorija Ejs

- batch file **Ejs.bat**, koji pokreće Ejs.
- Direktorij **data**, sadrži Ejs' source code. Ne mijenjajte ako ne znate.
- Direktorij **Simulations**, "default" radni direktorij. Sadrži dva pod-direktorija: **_examples** (primjeri distribuirani s Ejs) i **_library** (fileovi potrebni za rad Ejs-a, ne modificiraj ovaj direktorij).

dijelovi modela

- Model opisujemo nizom informacija, potrebno je
 - Deklarirati varijable modela.
 - Napisati algoritam koji računa evoluciju i varijable modela.
 - Definirati promjene u varijablama modela koje su rezultat interakcije s korisnikom.

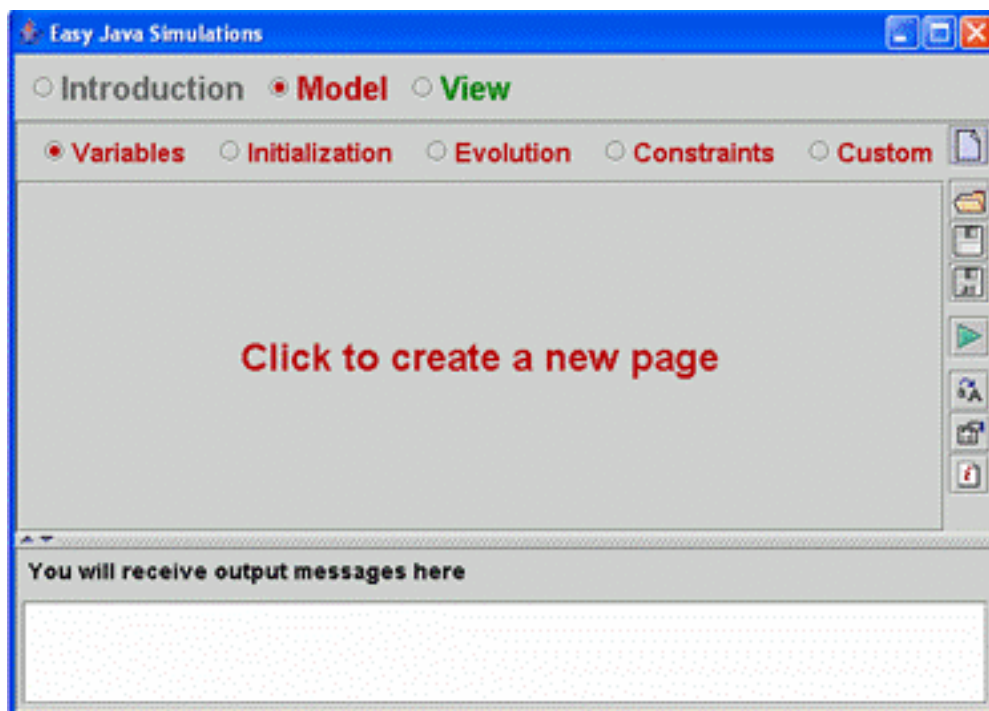
Ejs ima jednostavnu proceduru upisivanja neophodnih informacija pomoću pet panela.

model

- Varijable inicijaliziramo prilikom deklaracije. Početne vrijednosti napisane su u koloni “Value” panela “Variables”.
- Algoritmi za postavljanje početnih vrijednosti upisujemo u panel “Initialization”.
- Evoluciju opisujemo u panelu “Evolution”.
- U panel “Constraints”, upisujemo potrebna ograničenja na varijable.
- Dodatne metode upisujemo u panel “Custom”. One se mogu pozivati iz bilo kojeg dijela modela i “View” dijela.

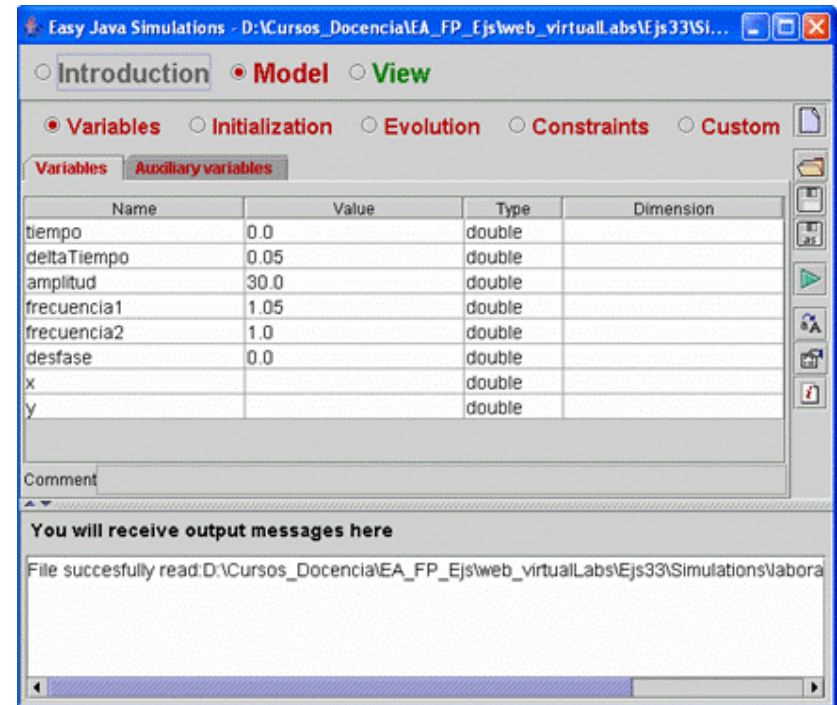
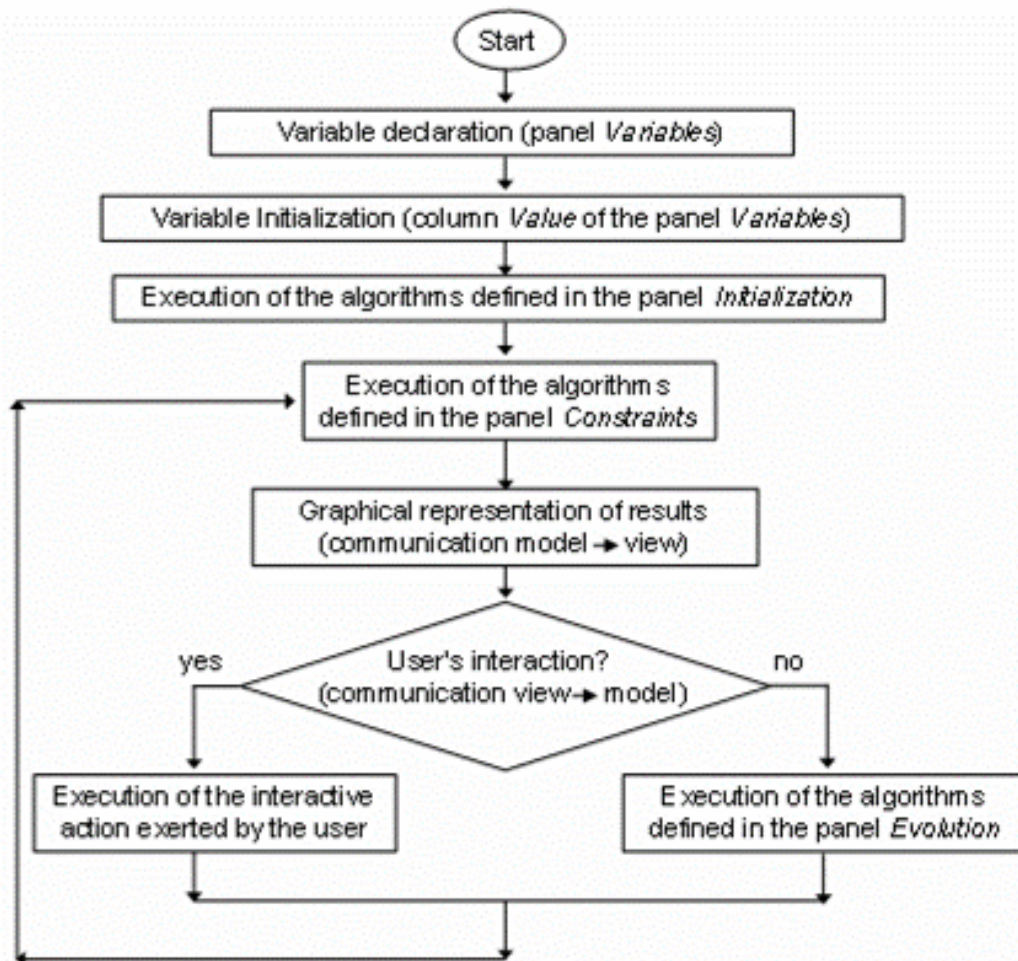
model

Custom metode obično se odnose na interakciju korisnika i modela.



5 panela za opisivanje modela

algoritam simulacija



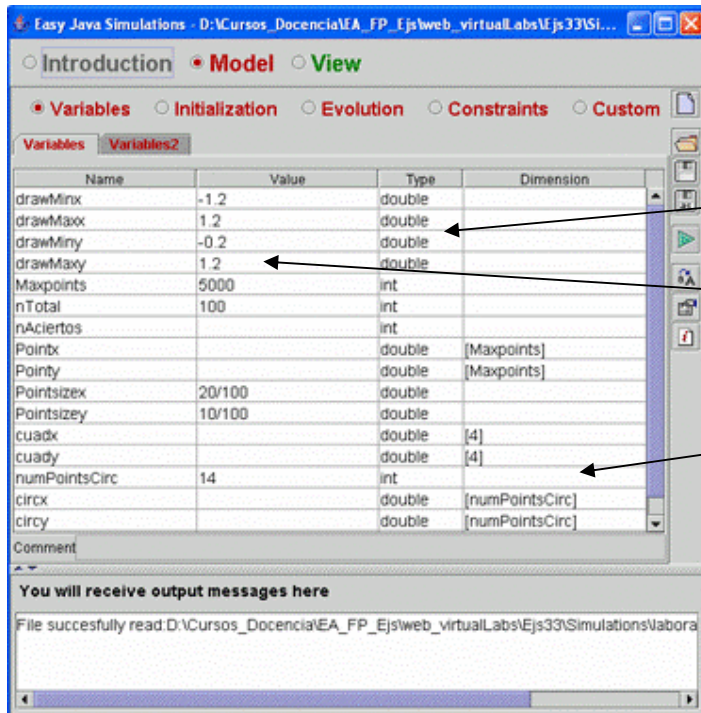
Definiranje varijabli i vrijednosti

Paneli se izvršavaju s lijeva na desno.

Varijable

- Ime varijable upisujemo u kolonu "Name".
- Tip varijable specificiramo u koloni "Type". Ejs zna 4 tipa varijabli: Boolean, int, double i string.
- Kolona "Dimension" definira dimenziju varijabli. Npr. "Dimension" kolona "cuadx" varijable definira vektor s 4 komponente: cuadx[0], cuadx[1], cuadx[2] i cuadx[3].
- Početne vrijednosti varijable upisujemo u kolonu "Value".

varijable

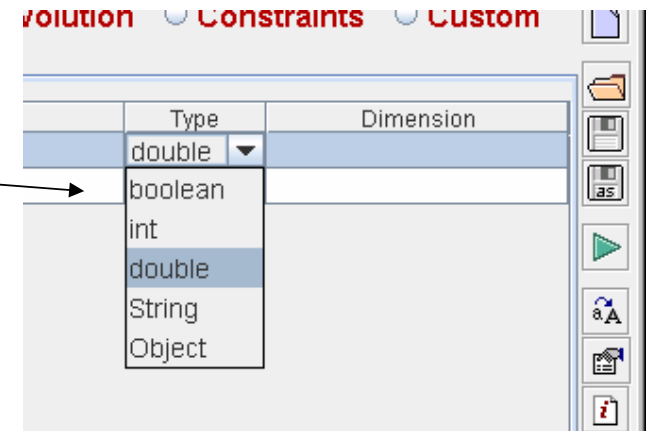


Definiramo:

Tip

vrijednost

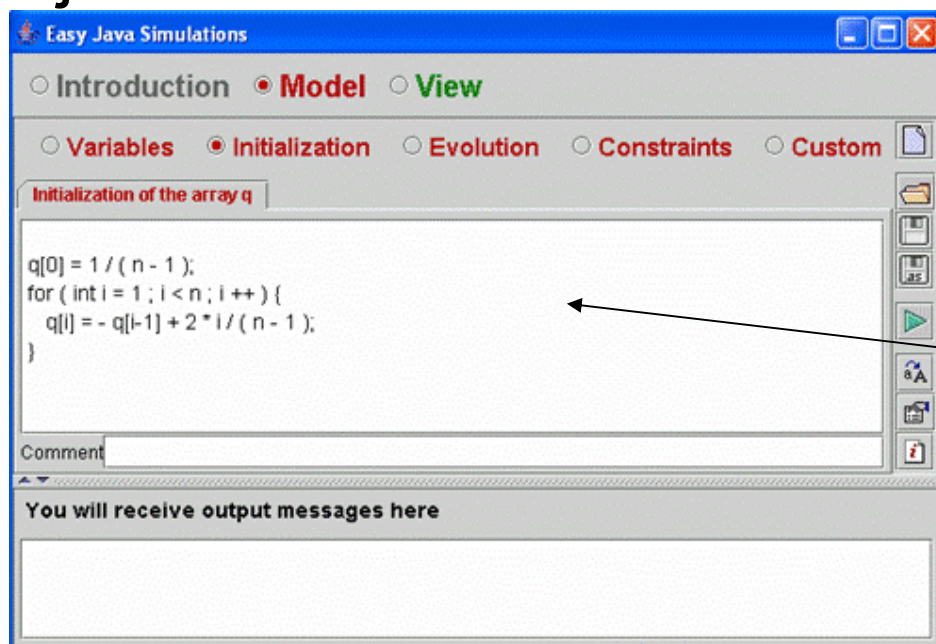
dimenzija



Inicijalizacija

- Ako je potrebno inicijalizirati vektor onda algoritam pišemo u panelu "Initialization". Npr. vektor q mora imati vrijednosti dane formulama

$$q[0] = \frac{1}{n-1}$$
$$q[i] = -q[i-1] + \frac{2 \cdot i}{n-1}$$



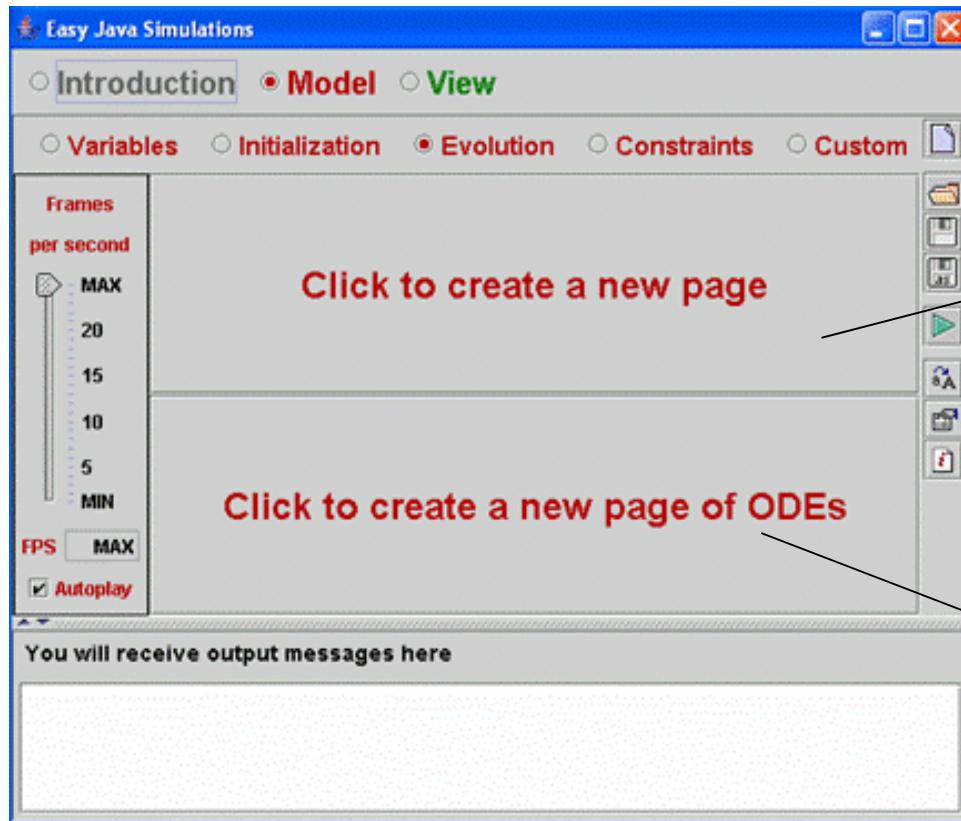
The screenshot shows the 'Easy Java Simulations' window with the 'Initialization' tab selected. The code in the text area is:

```
q[0] = 1 / (n - 1);  
for ( int i = 1; i < n; i++ ) {  
    q[i] = - q[i-1] + 2 * i / (n - 1);  
}
```

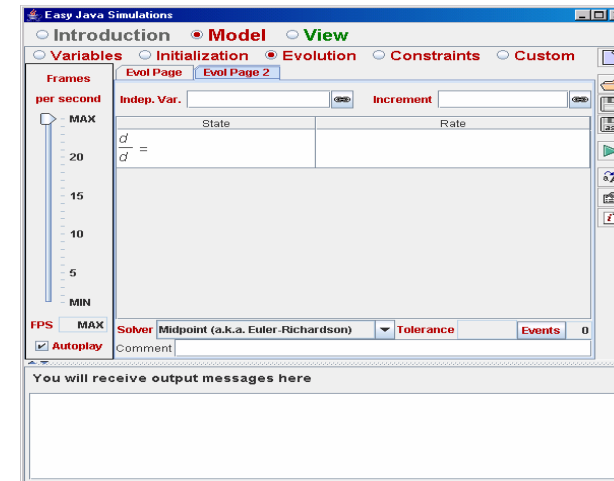
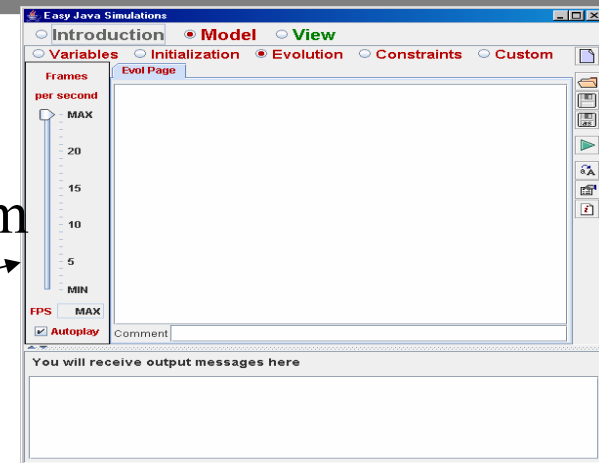
An arrow points from the text 'Java code' to this code block.

Java code

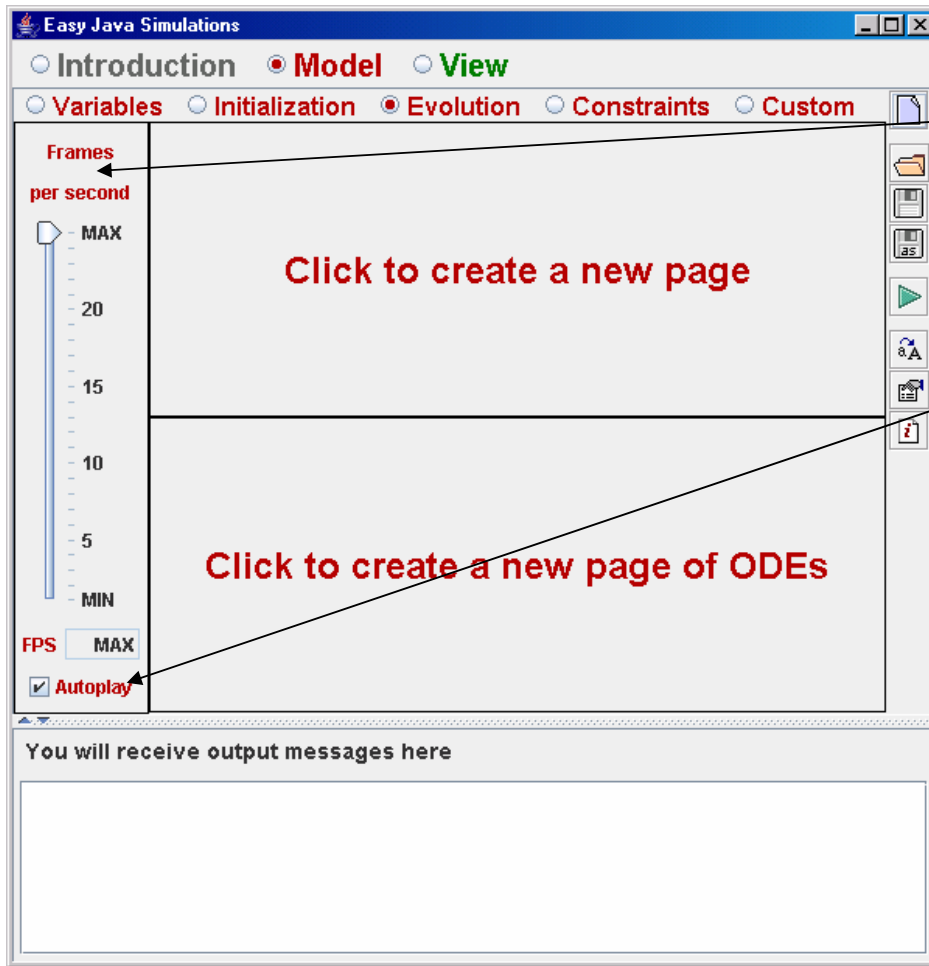
Evolucija



Java
algoritam



Evolucija



Brzina simulacije

Simulacija počinje odmah, inače moramo uključiti gumb s funkcijom `_play()`.

Algoritmi za ODE

Euler, Euler-Richardson, Runge-Kutta 4
i Runge-Kutta-Fehlberg 4-5

algoritmi

Euler

$$x_{i+1} = x_i + f(x_i, t_i) \cdot \Delta t$$

Euler Richardson

$$x_{med} = x_i + f(x_i, t_i) \cdot \frac{\Delta t}{2}$$
$$x_{i+1} = x_i + f\left(x_{med}, t_i + \frac{\Delta t}{2}\right) \cdot \Delta t$$

Runge Kutta 4-ti red

$$k_1 = \Delta t \cdot f(x_i, t_i)$$
$$k_2 = \Delta t \cdot f\left(x_i + \frac{k_1}{2}, t_i + \frac{\Delta t}{2}\right)$$
$$k_3 = \Delta t \cdot f\left(x_i + \frac{k_2}{2}, t_i + \frac{\Delta t}{2}\right)$$
$$k_4 = \Delta t \cdot f(x_i + k_3, t_i + \Delta t)$$
$$x_{i+1} = x_i + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6}$$

algoritmi

Runge-Kutta-Fehlberg (4 - 5 red)

$$k_1 = \Delta t \cdot f(x_i, t_i)$$

$$k_2 = \Delta t \cdot f\left(x_i + \frac{k_1}{4}, t_i + \frac{\Delta t}{4}\right)$$

$$k_3 = \Delta t \cdot f\left(x_i + \frac{3}{32} \cdot k_1 + \frac{9}{32} \cdot k_2, t_i + \frac{3}{8} \cdot \Delta t\right)$$

$$k_4 = \Delta t \cdot f\left(x_i + \frac{1932}{2197} \cdot k_1 - \frac{7200}{2197} \cdot k_2 + \frac{7296}{2197} \cdot k_3, t_i + \frac{12}{13} \cdot \Delta t\right)$$

$$k_5 = \Delta t \cdot f\left(x_i + \frac{439}{216} \cdot k_1 - 8 \cdot k_2 + \frac{3680}{513} \cdot k_3 - \frac{845}{4104} \cdot k_4, t_i + \Delta t\right)$$

$$k_6 = \Delta t \cdot f\left(x_i - \frac{8}{27} \cdot k_1 + 2 \cdot k_2 - \frac{3544}{2565} \cdot k_3 + \frac{1859}{4104} \cdot k_4 - \frac{11}{40} \cdot k_5, t_i + \frac{\Delta t}{2}\right)$$

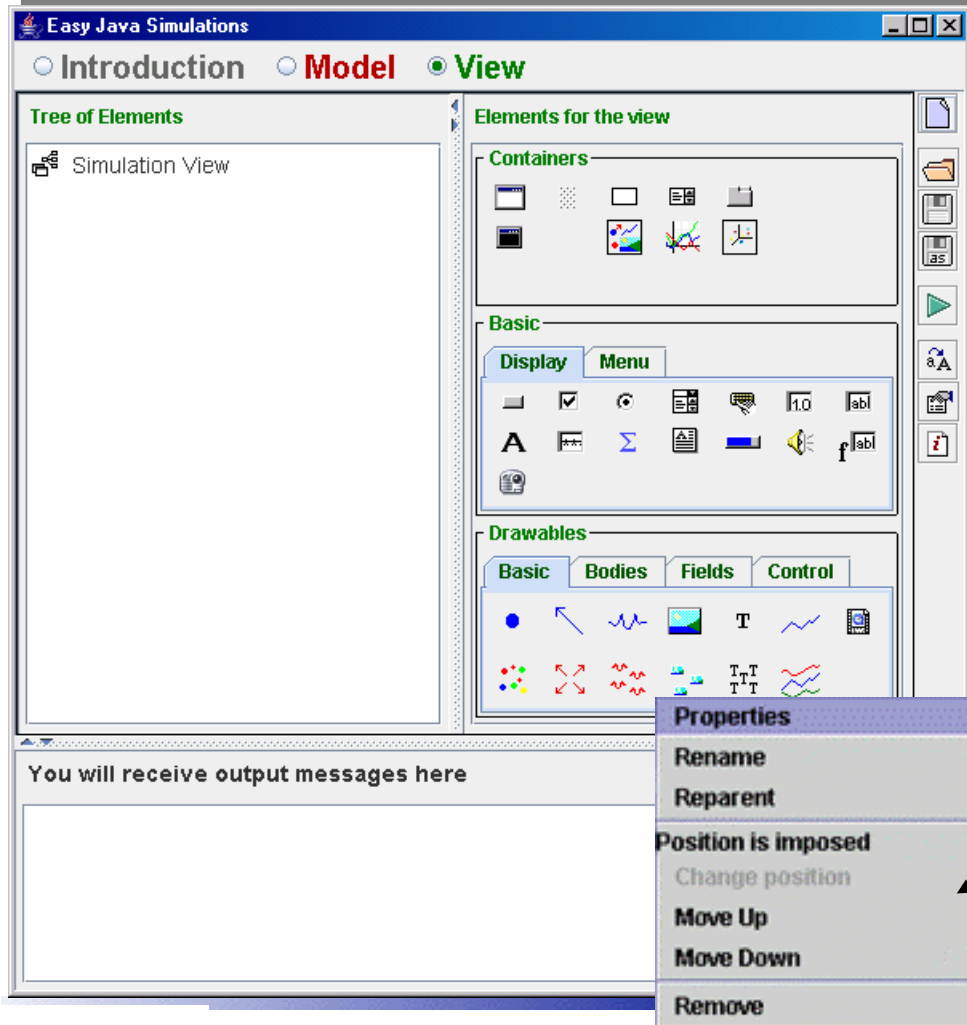
View

- Kontrolira simulaciju, tj. start, pauzu i ponovno inicijaliziranje (reinitialize).
- Interaktivna promjena vrijednosti varijabli.
- Vizualizacija modela u 2D i 3D, mogućnost crtanja grafa (x,y) ili (nekoliko varijabli, y)
- Prikazuje ili skriva prozore. Obično se prikazuju prozori s kontrolama, a graf se skriva.

View se definiira pomoću već definiranih kontrola koje su podijeljene u 3 kategorije

- Containers
- Basic
- Drawables
 - Basic
 - Graphs
 - Bodies, and Fields.

View

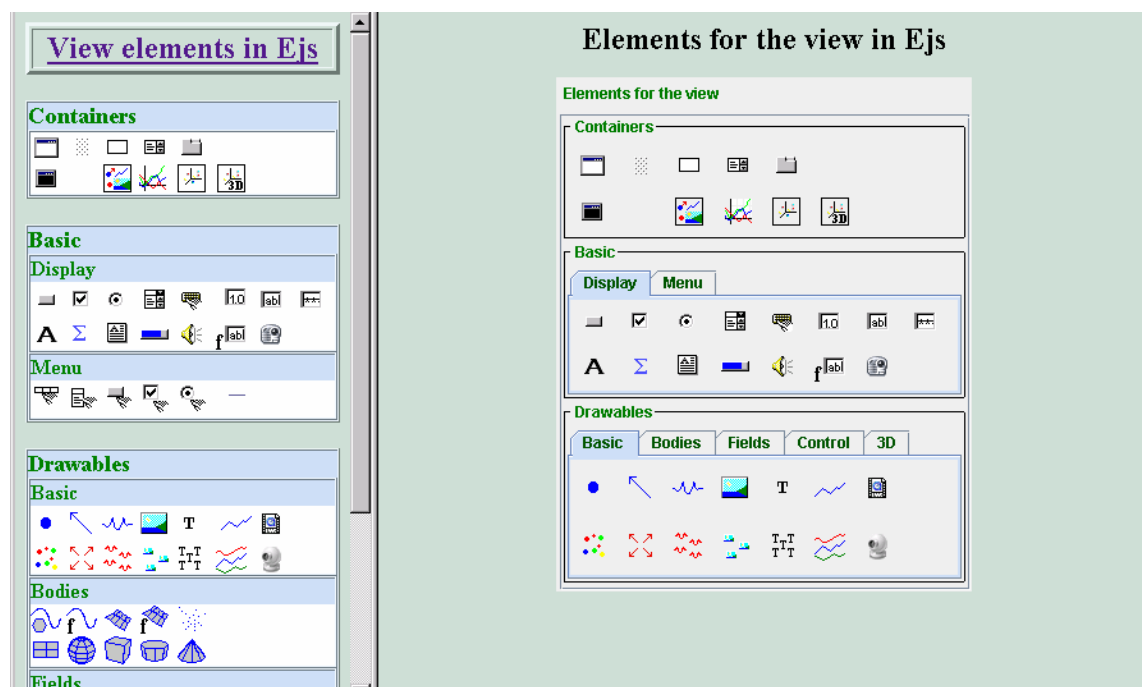


- Tree of Elements - sadrži elemente koje smo dodali
- redosljed je važan
- miš+klik odabire element iz desnog panela “Elements of the view” klikom u lijevi panel dodajemo element.
- grafičke osobine GUIa mjenjamo tako da kliknemo desnim mišem

Elementi



Kompletan opis elemenata dan je u uputama: Easy Java Simulations
The Manual, i na adresi

http://fem.um.es/Ejs/LibroEjs_en/CD/Reference/Reference.html



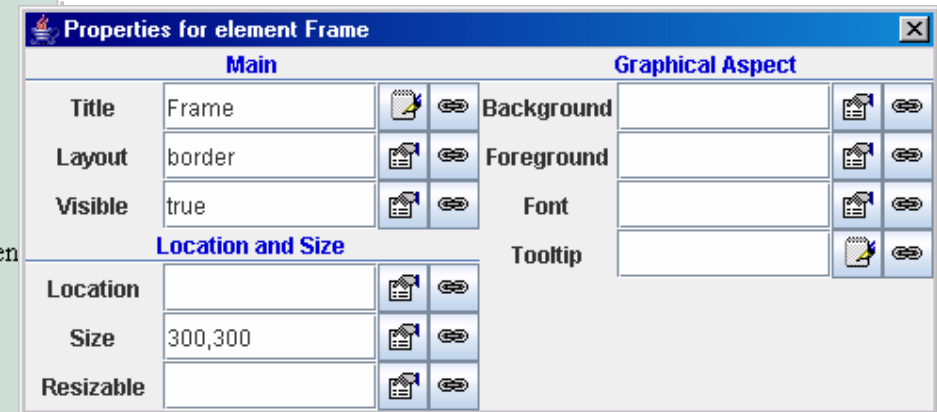
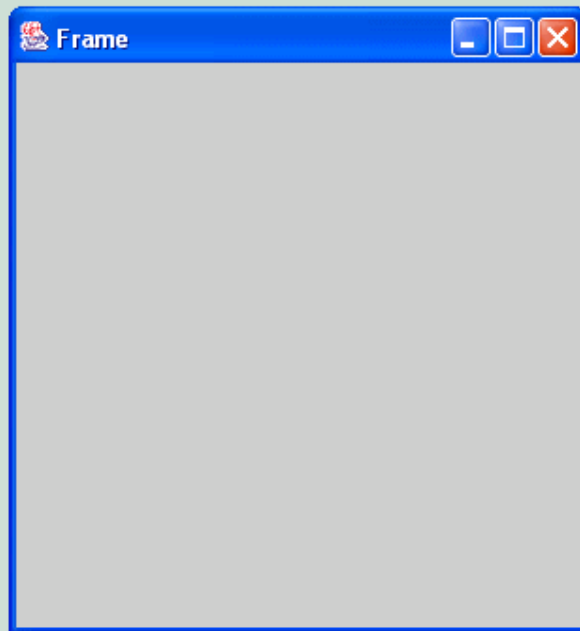
Frame

Element: Frame.

Icon:  , when it is the main window.

Text: A top-level window.


Description: A *Frame* is a container element that displays in an independent window. When this window acts as the main window, closing it (not minimizing it) exits the simulation.



Osobine prozora- dobivamo
duplim klikom u panelu Tree ...

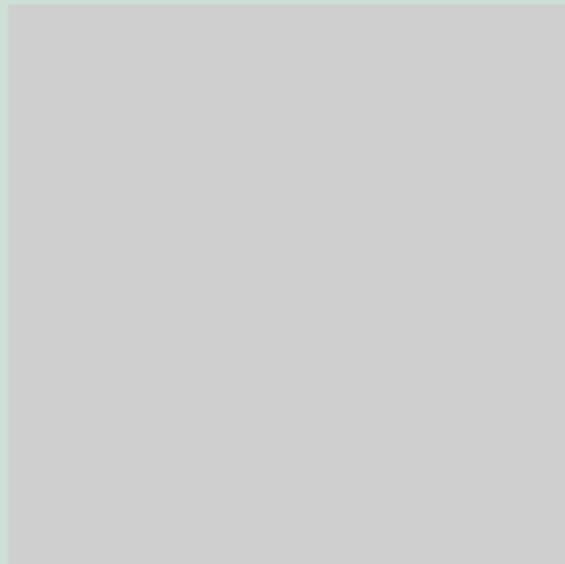
panel

Element: Panel.

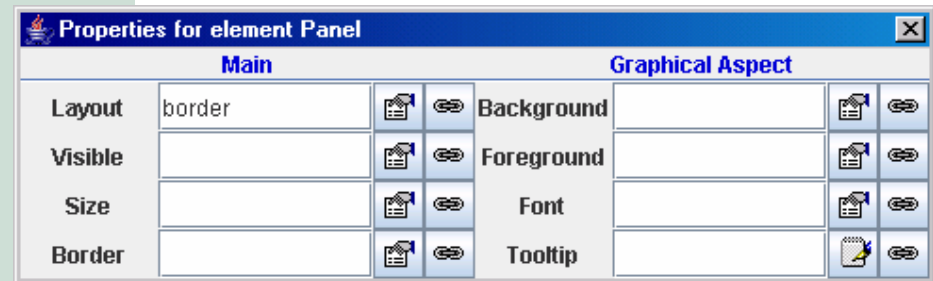
Icon: 

Text: A basic container panel.

Descripción: A *Panel* is the most basic container. It can be used just to host one or more children, according to its layout property. A panel can be hidden using the *visible* property.



Osobine panela



Drawing panel

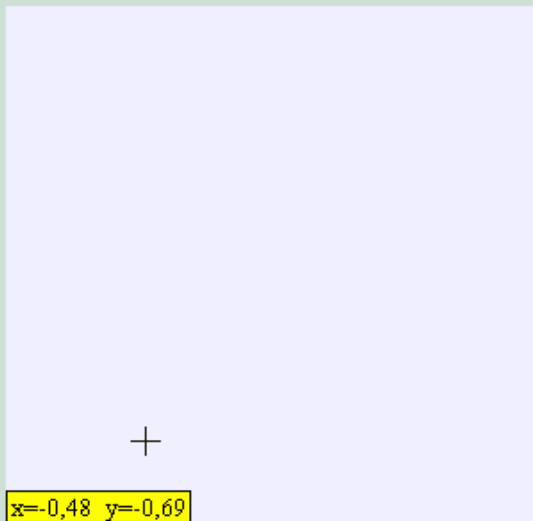
Element: DrawingPanel.



Text: A 2D container for Drawables.

Description: A *DrawingPanel* is a special container to host drawable children. Even when it is considered a container, it cannot be used to host other type of elements.

Drawing panels represent a region of the two-dimensional plane and provide their own coordinate system, from the point (Minimum X, Minimum Y) to (Maximum X, Maximum Y). (Even though the scales can also be automatically adjusted.)



Properties for element DrawingPanel

Scales		Configuration	
Autoscale X	false	Square	
Autoscale Y	false	Gutters	
Minimum X	-1.0	Coordinates	
Maximum X	1.0	X Format	
Minimum Y	-1.0	Y Format	
Maximum Y	1.0	The format for the X coordinate graphical aspect	
Interaction		Visible	
X		Size	
Y		Background	
On Press		Foreground	
On Drag		Font	
On Release		Tooltip	
Mouse Enter			
Mouse Exit			
Key Action			
Key Pressed			

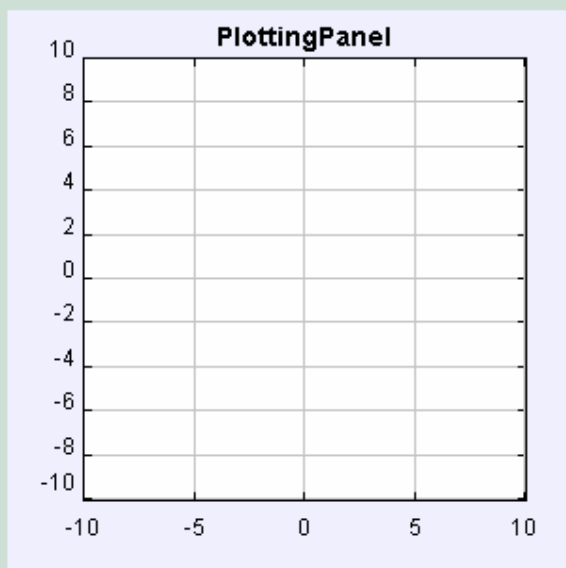
Plotting panel

Element: PlottingPanel.

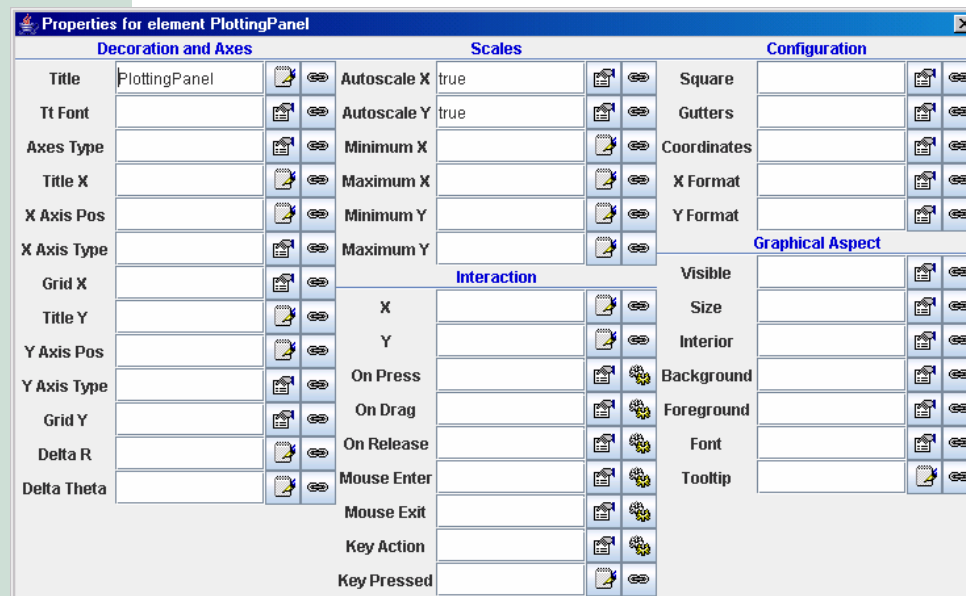
Icon: 

Text: A 2D drawing panel with a system of axes.

Description: A *PlottingPanel* is a 2D drawing panel that provides, by default, a system of coordinate axes.



osobine




Decoration and Axes		Scales		Configuration	
Title	PlottingPanel	Autoscale X	true	Square	
Tt Font		Autoscale Y	true	Gutters	
Axes Type		Minimum X		Coordinates	
Title X		Maximum X		X Format	
X Axis Pos		Minimum Y		Y Format	
X Axis Type		Maximum Y		Graphical Aspect	
Grid X		Interaction		Visible	
Title Y		X		Size	
Y Axis Pos		Y		Interior	
Y Axis Type		On Press		Background	
Grid Y		On Drag		Foreground	
Delta R		On Release		Font	
Delta Theta		Mouse Enter		Tooltip	
		Mouse Exit			
		Key Action			
		Key Pressed			

Plotting panels are interactive and respond to user interaction in the same way as drawing panels.

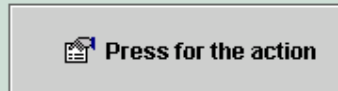
gumb

Element: Button.

Icon: 

Text: A button for actions.

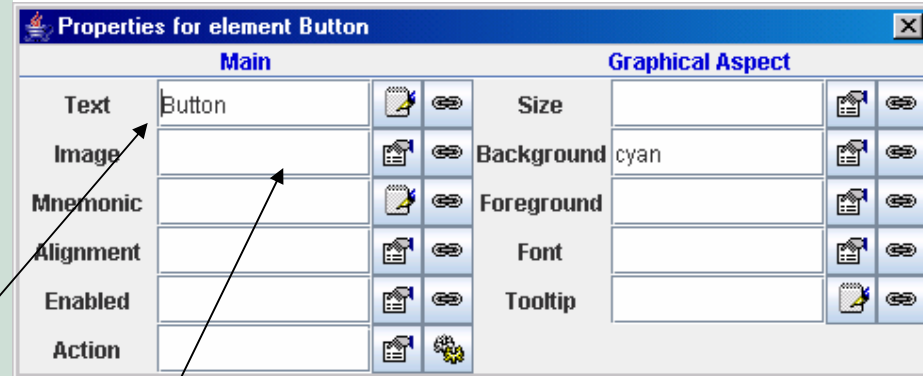
Description: A *Button* is a basic element used to invoke actions. It can display a text, an image, or both. The action is invoked when the button is clicked (i.e. pressed and released).



Buttons can be disabled using the "Enabled" property. In this case, the interface is grayed out.

Table of properties

Name	Description	Values accepted
Main		
Text	The text displayed by the element.	Any constant or variable of type String .
Image	The image displayed by the element.	Any constant or variable of type String , corresponding to a GIF or animated GIF image. The string indicates the path to the corresponding image file. The path can be relative to the working directory or an Internet URL.



checkbox

Element: CheckBox.

Icon:

Text: A check box for boolean variables.

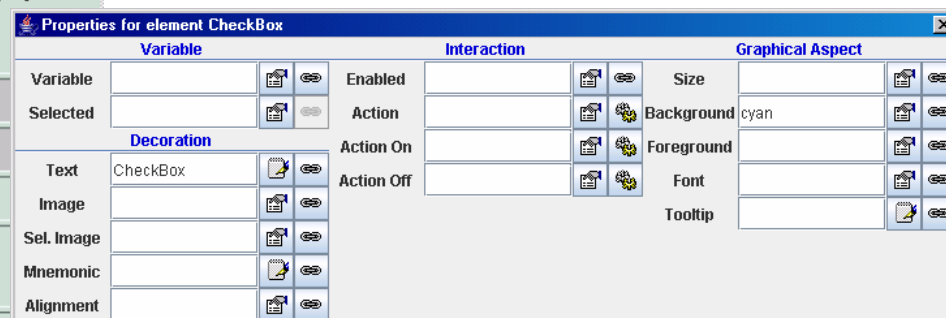
Description: A *CheckBox* is a basic element used to display and modify a boolean value. It can display a text, an image, or both. Check boxes can invoke an action when the value is changed. Moreover, individual actions can also be specified for the cases in which the value is selected to be **true** or **false**. If both are set, this second action is always invoked after the first one.

Option on

Check boxes can be disabled using the "Enabled" property. In this case, the interface is greyed out.


Table of properties

Name	Description	Values accepted
Variable		
Variable	The variable to edit using this element.	A boolean variable.
Selected	Initial state of the variable.	A boolean variable or one of the constants true or false .



particle

Element: Particle.

Icon: 

Text: An interactive particle.


Description: A *Particle* is a drawing element that displays a simple geometrical shape, a rectangle or an ellipse, at given coordinates of the parent drawing panel. The shape is drawn at the given coordinates with the size specified in each direction (in 3D the largest of the X and Y sizes is used for the horizontal dimension). The element can also be applied a rotation and a zoom factor.



Particles respond to user interaction on their position coordinates (changing their position), invoking up to three different actions. The precise drawing position of the geometrical shape with respect to the given coordinates can be chosen among several standard values.

spring

Element: Spring.

Icon: 

Text: An interactive spring.

Description: A *Spring* is a drawing element that displays a spring at given coordinates of the parent drawing panel, with a given size. The element can also be applied a zoom factor.




Springs respond to user interaction on their head (changing size) and on their tail (changing position), invoking up to three different actions.

Table of properties

Name	Description	Values accepted
Position and Size		
X	The X coordinate of the element.	Any constant or variable of type double or int .
Y	The Y coordinate of the element.	Any constant or variable of type double or int .
Z	The Z coordinate of the element.	Any constant or variable of type double or int .

image

Element: Image.

Icon: 

Text: An interactive image.

Description: An *Image* is a drawing element that displays a GIF or animated GIF image. The image is displayed at given coordinates with the size indicated in each direction (in 3D display modes, the largest of the sizes in X and Y is used for the horizontal dimension). The element can also be applied a rotation and a zoom factor.

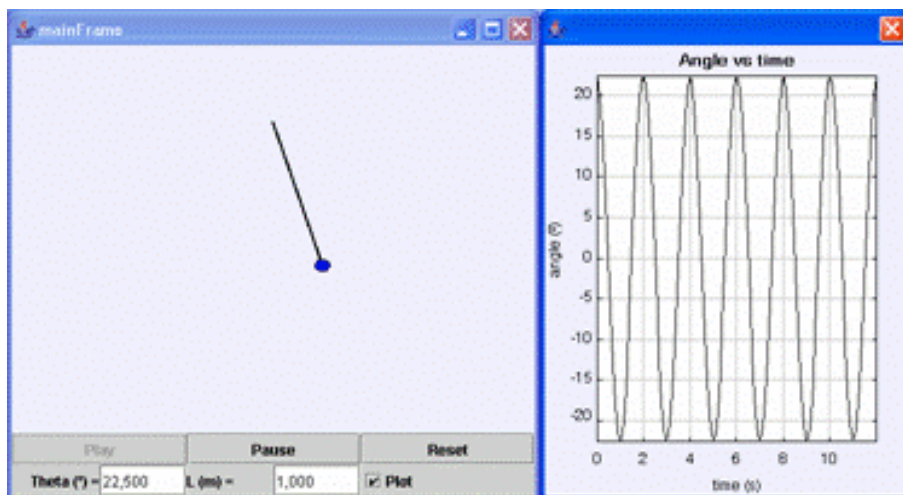


Images respond to user interaction on their position coordinates (changing their position), invoking up to three different actions. The precise drawing position of the image with respect to the given coordinates can be chosen among several standard values.

primjeri

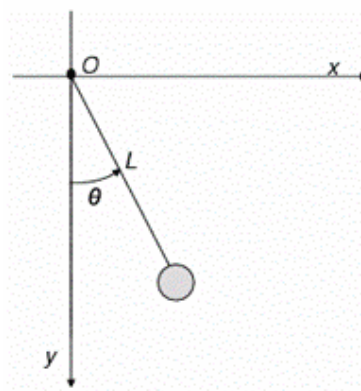
- EJS
 - niz primjera u `_examples` direktoriju
- Dodatni primjeri
 - Kod "virtualnih laboratorija" (t.j. `.xml` fileovi) i slike nalaze se u jednom fileu: *laboratoriosTexto.zip*. File se nalazi na adresi http://www.simulab-pfp.dia.uned.es/curso_online/laboratoriosTexto.zip
 - Instalacija se sastoji od raspakiranja file-a *laboratoriosTexto.zip* u direktoriju *Simulations*

njihalo



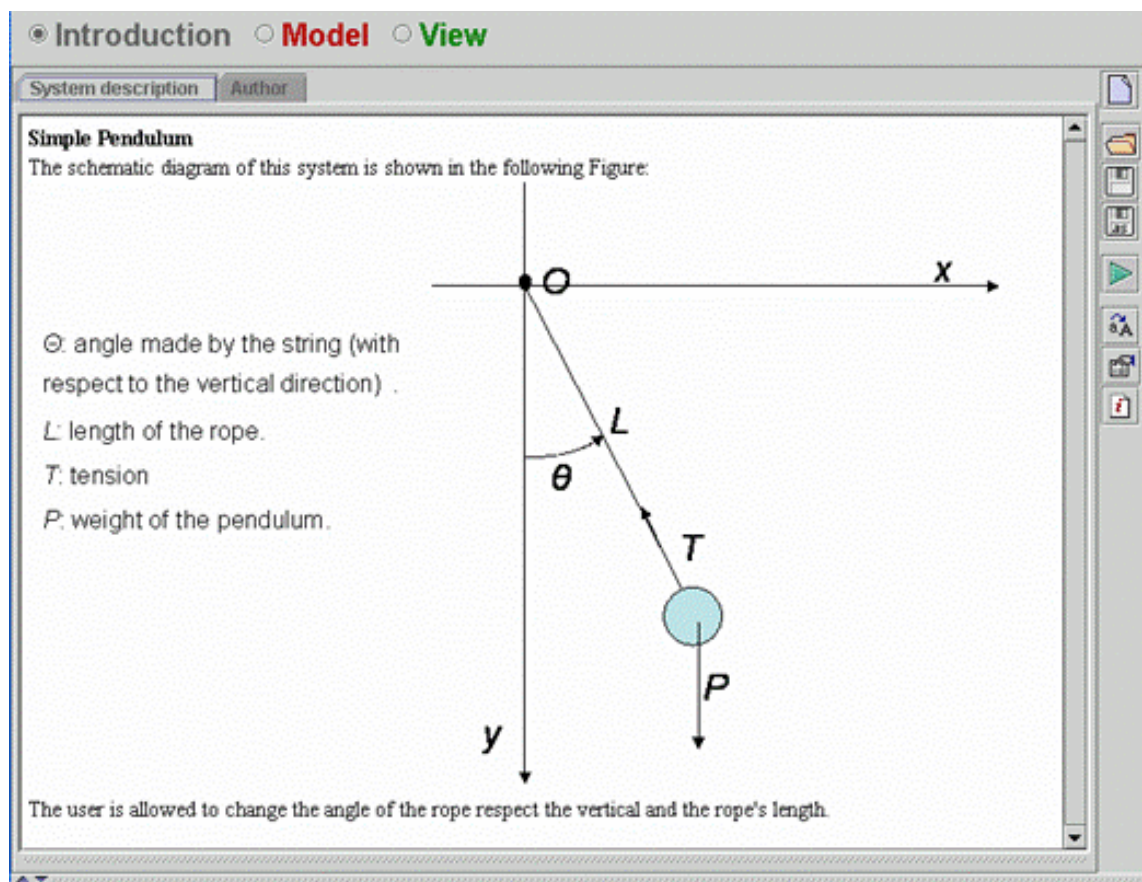
[PenduloSimple_En.xml](#)

model



$$\frac{d\theta}{dt} = \omega$$
$$\frac{d\omega}{dt} = -K^2 \cdot \theta$$

njihalo uvod

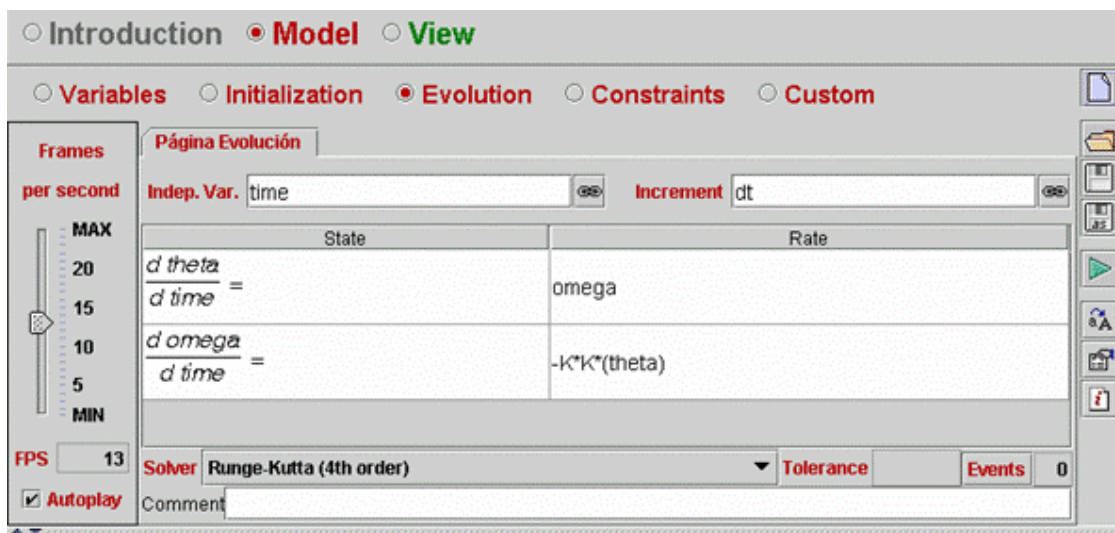


njihalo evolucija

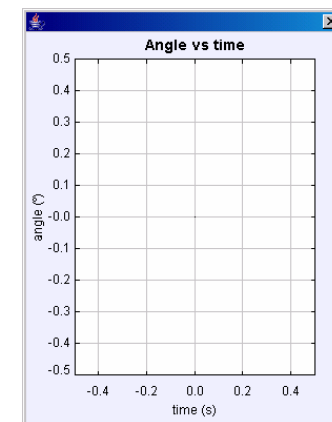
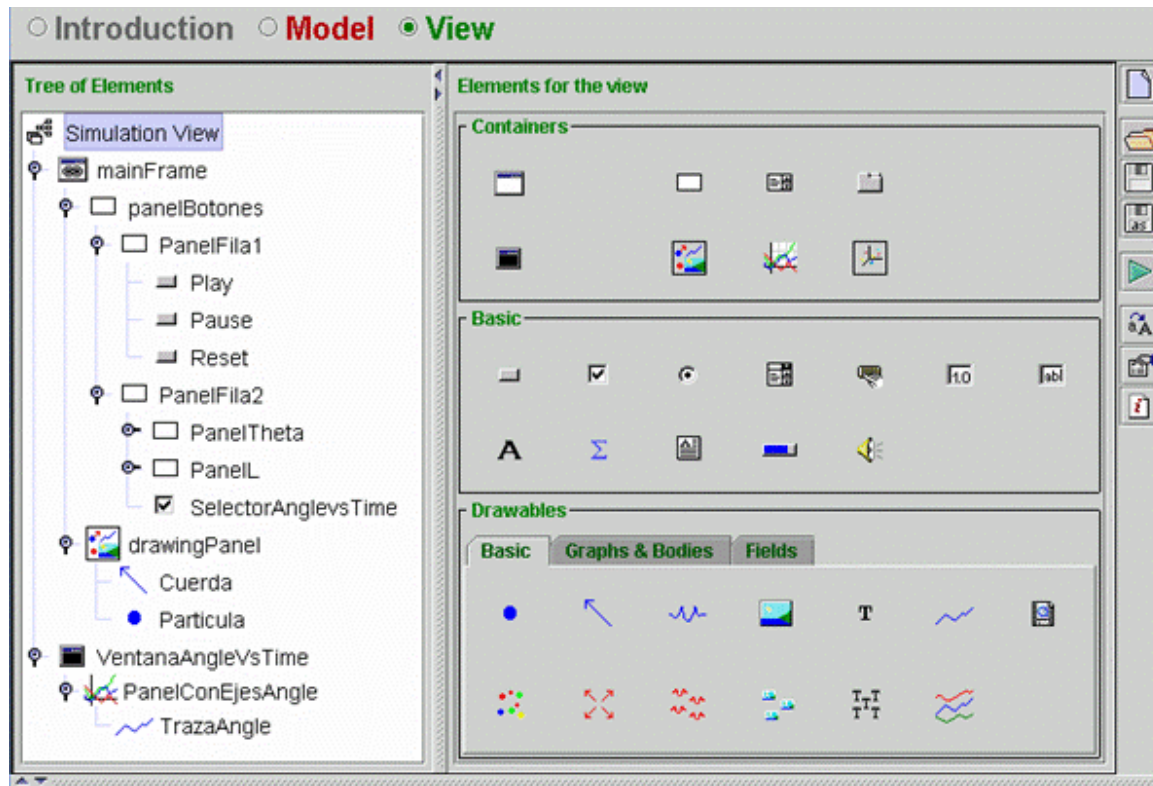
ODE

$$\frac{d\theta}{dt} = \omega$$
$$\frac{d\omega}{dt} = -K^2 \cdot \theta$$

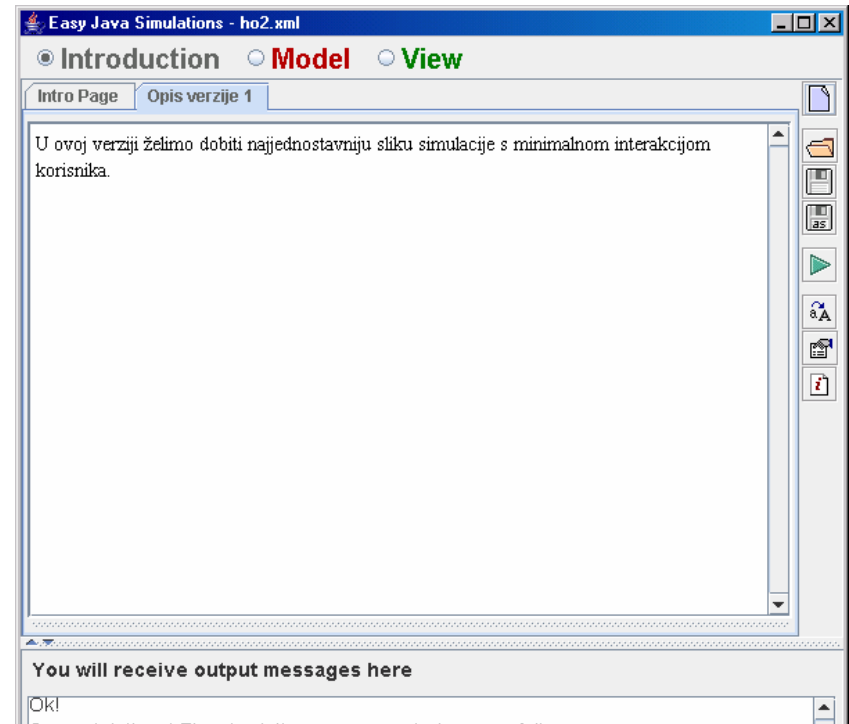
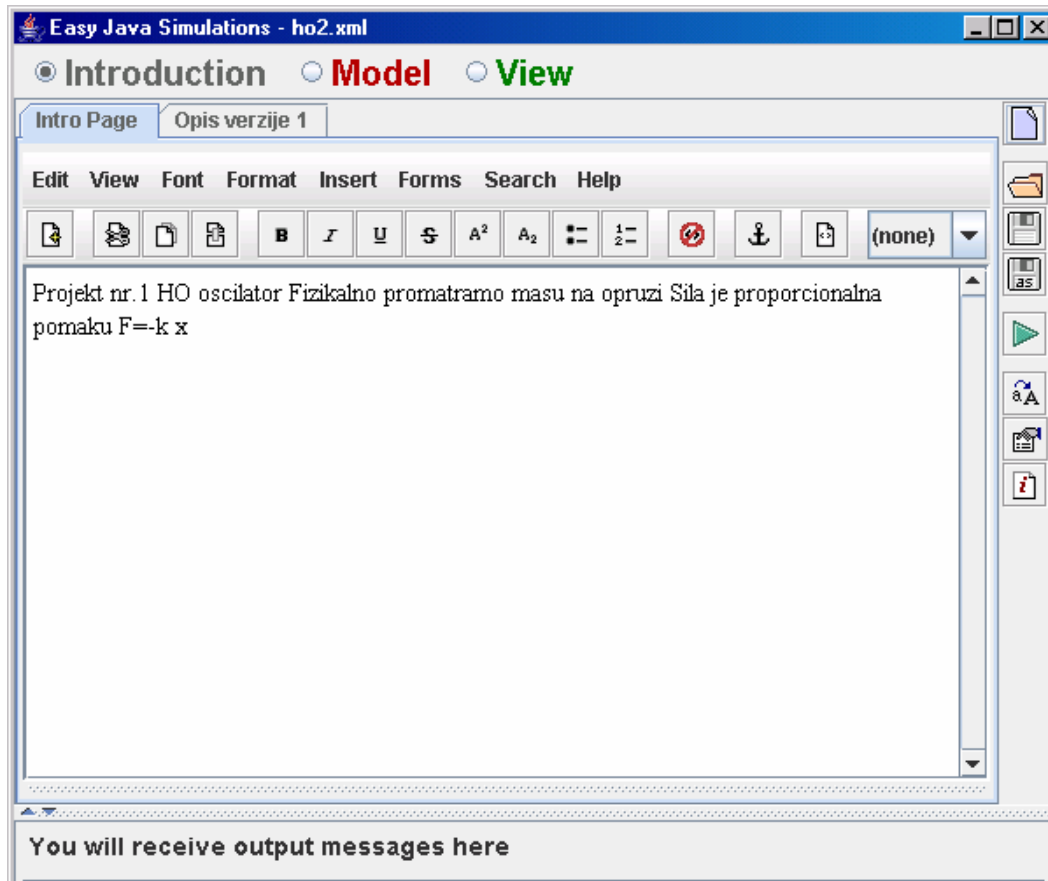
K - parametar, ovisi o duljini niti i akceleraciji gravitacije g.



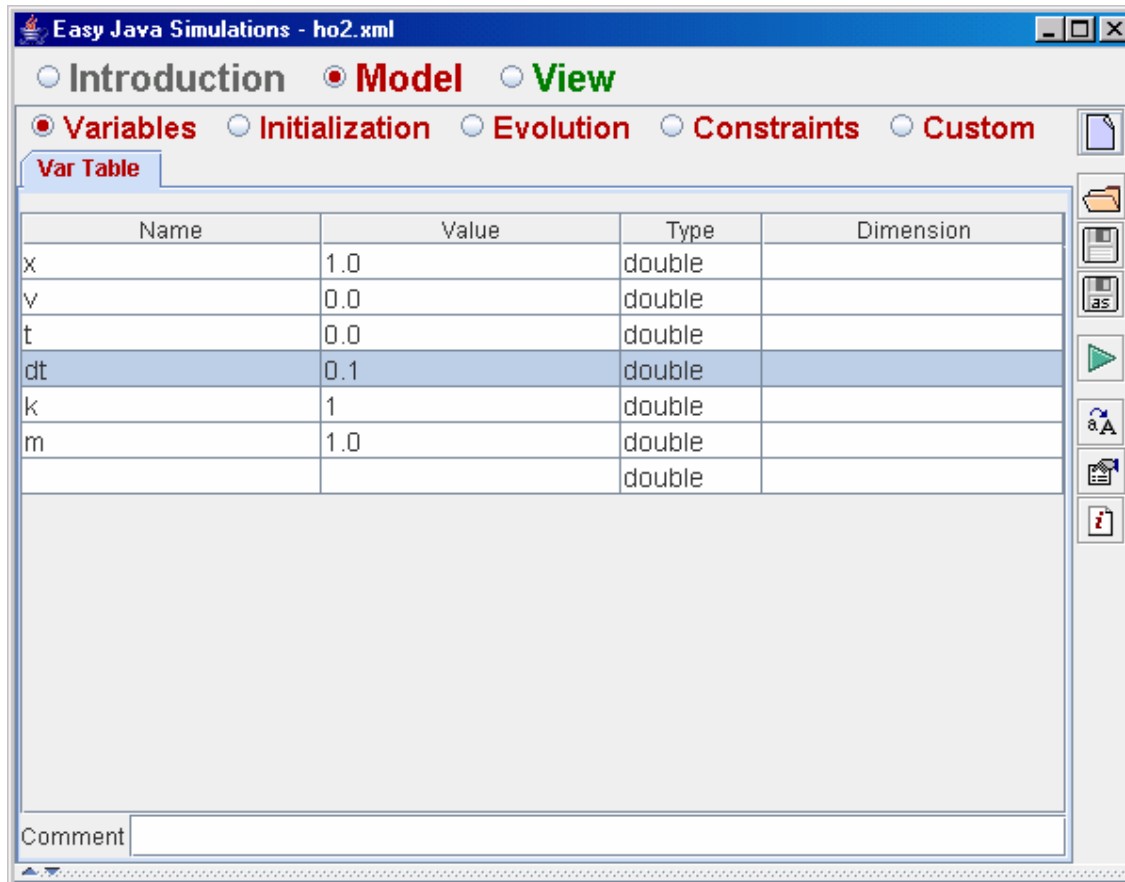
njihalo view



HO uvod



HO model/varijable

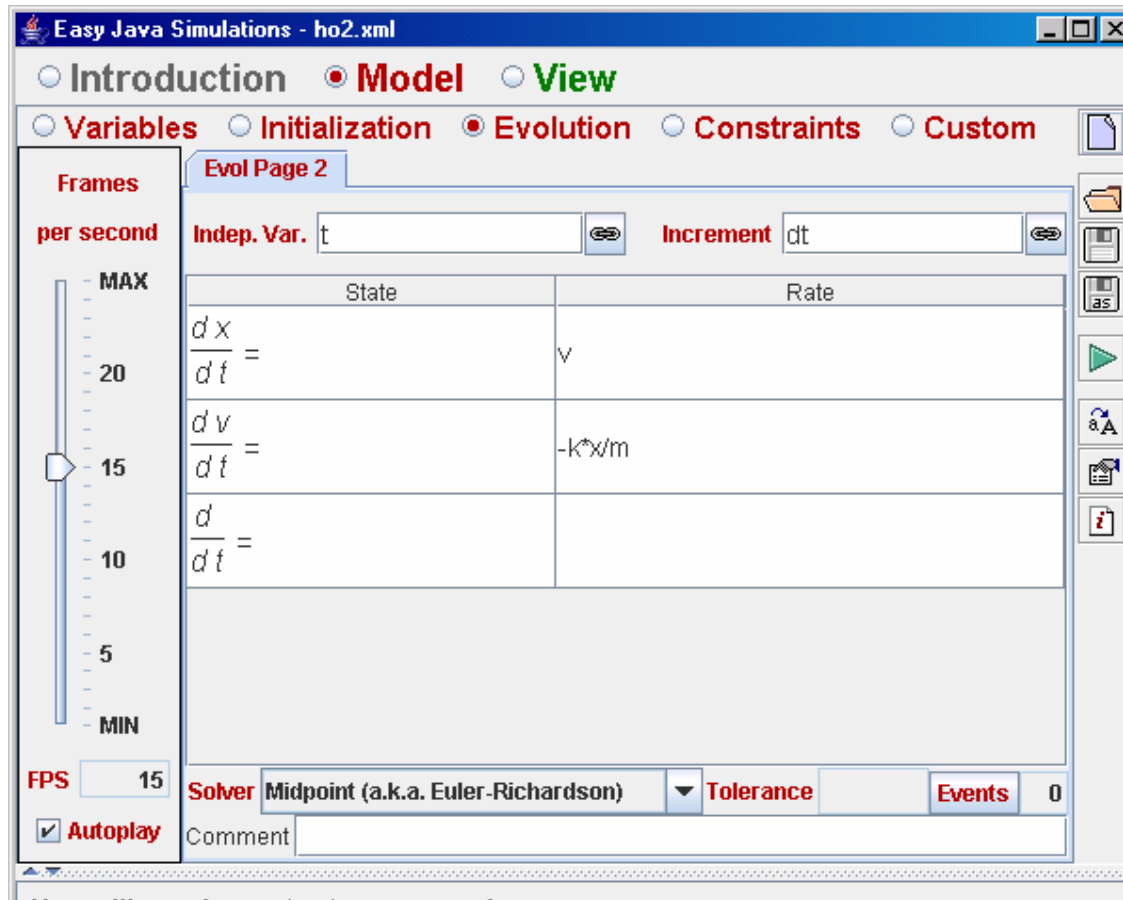


The screenshot shows the 'Easy Java Simulations - ho2.xml' window. The 'Model' tab is selected, and the 'Variables' sub-tab is active. A 'Var Table' is displayed with the following data:

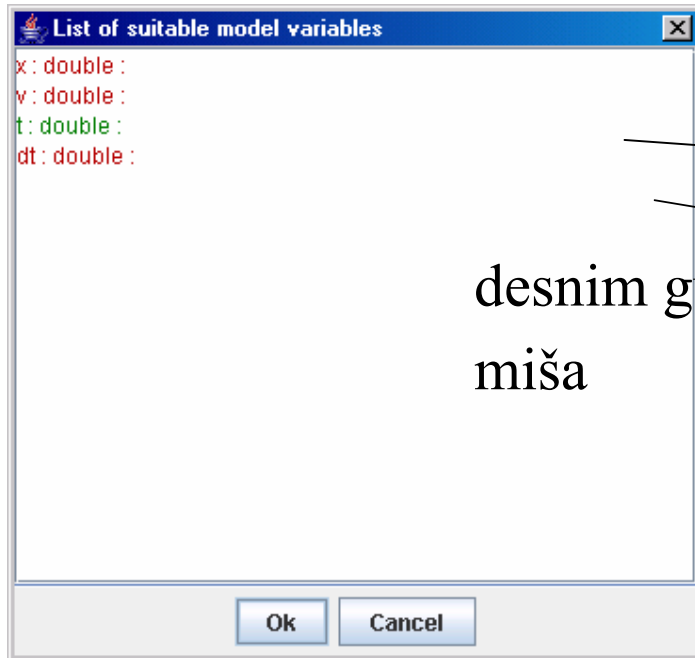
Name	Value	Type	Dimension
x	1.0	double	
y	0.0	double	
t	0.0	double	
dt	0.1	double	
k	1	double	
m	1.0	double	
		double	

Below the table is a 'Comment' text area.

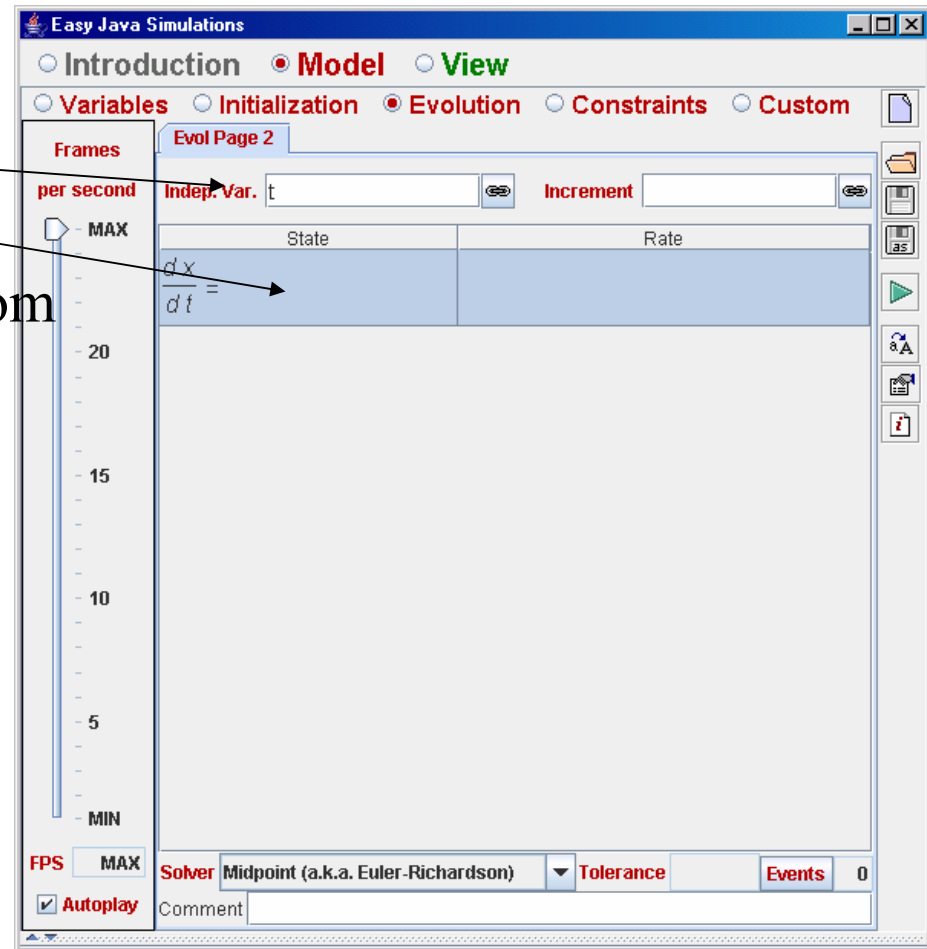
HO model/evolucija



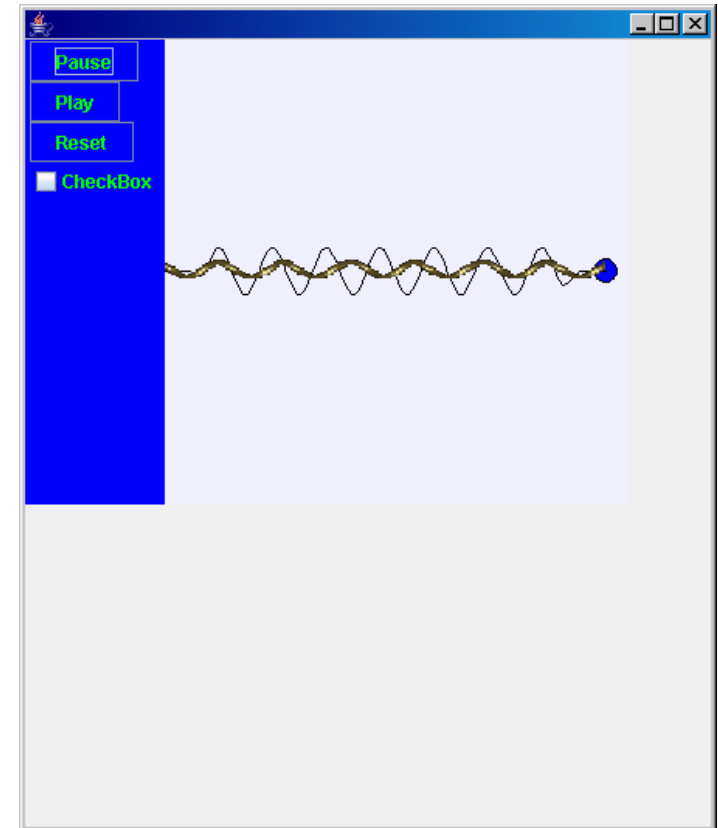
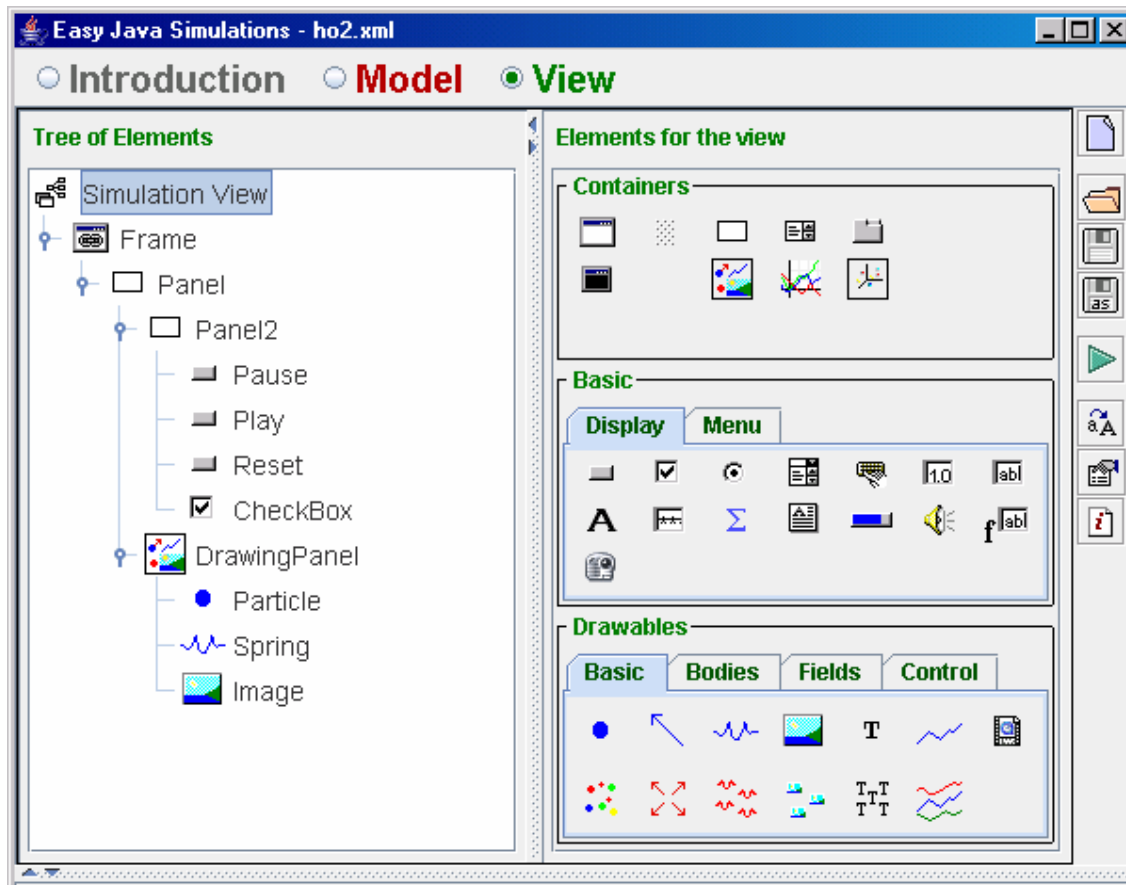
HO model/evolucija



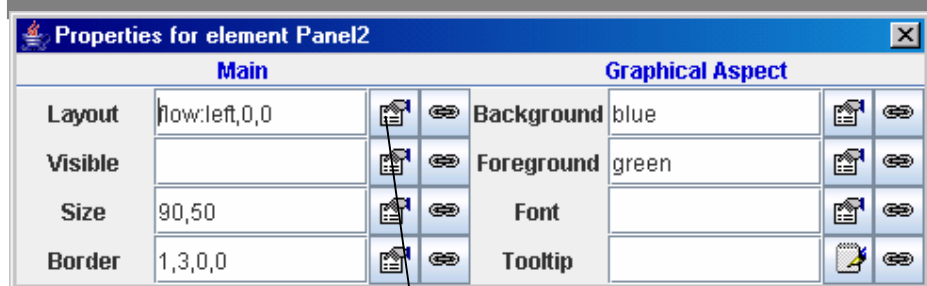
desnim gumbom
miša



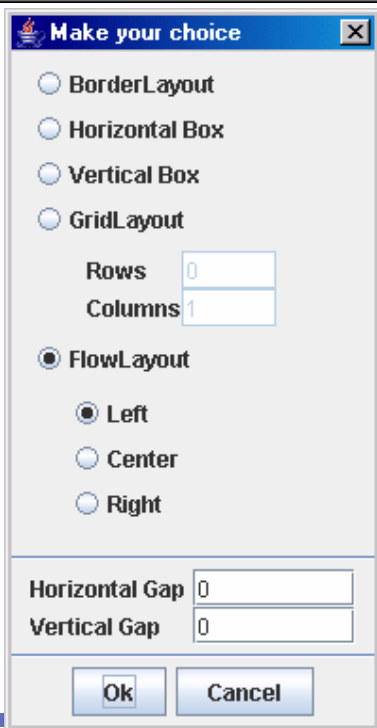
HO view



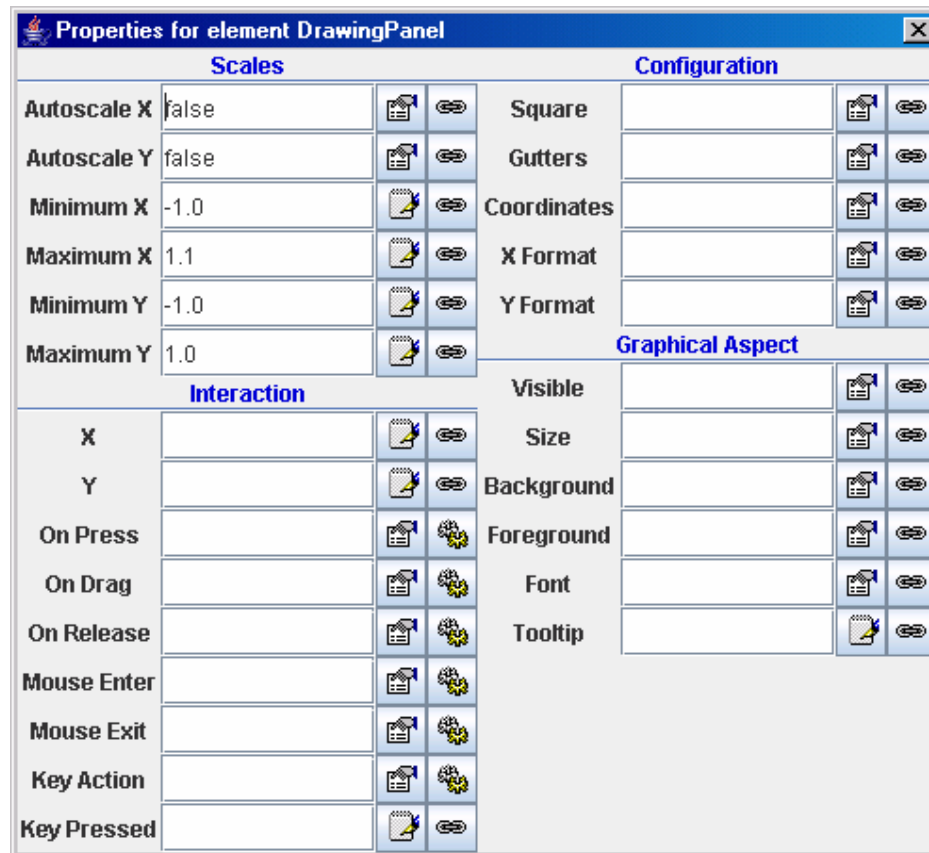
HO paneli



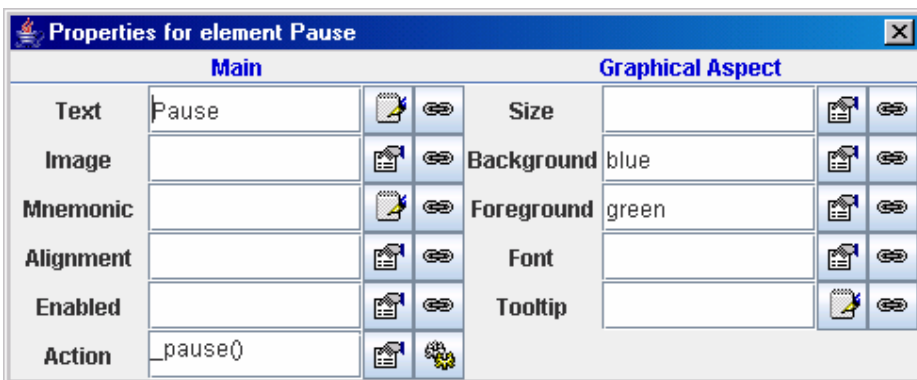
Panel s gumbima



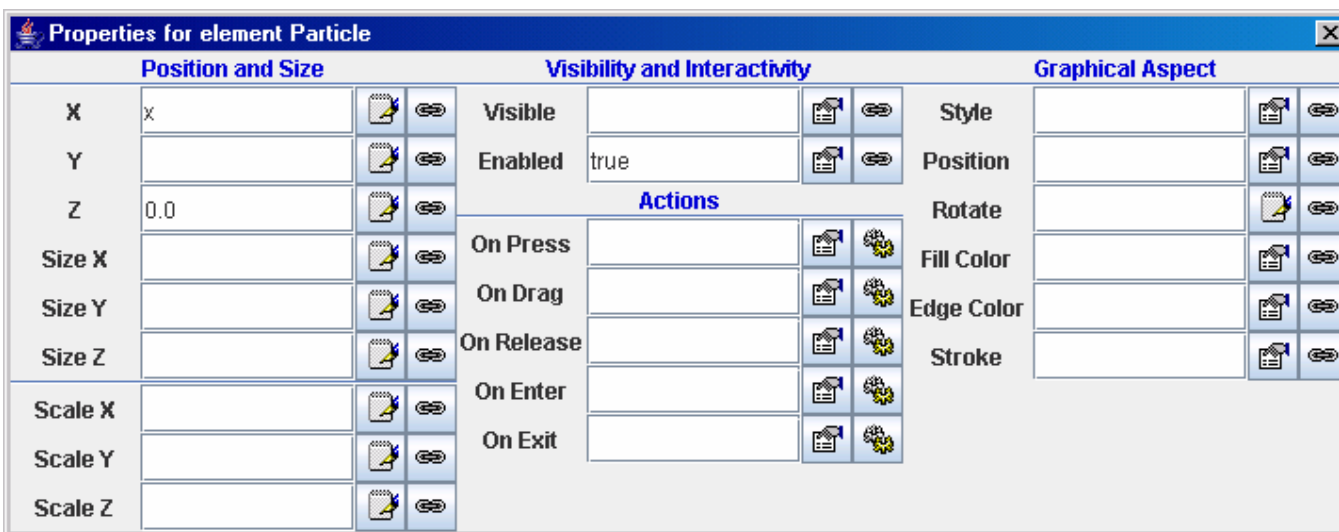
panel simulacije



elementi

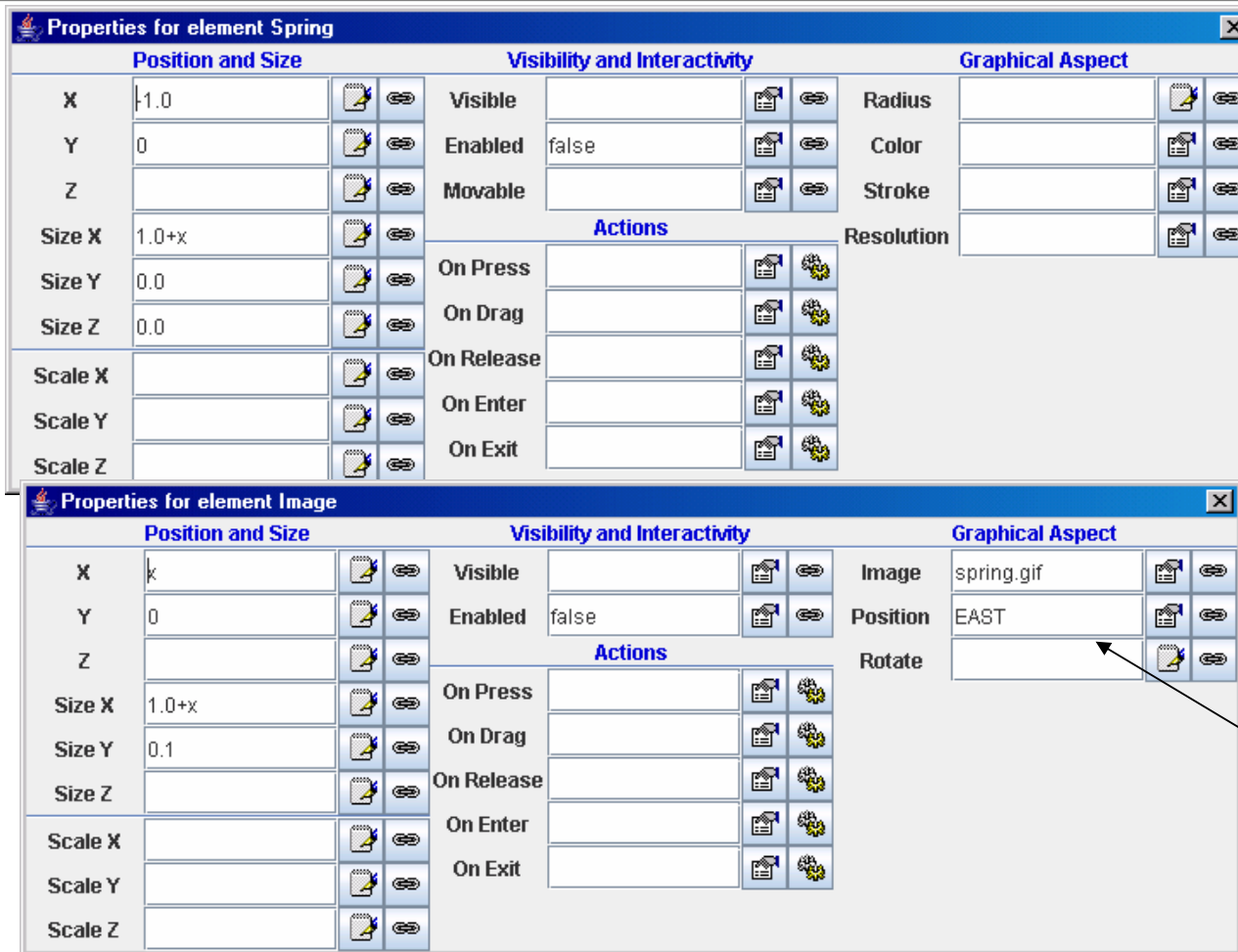


gumb pauza



Particle

elementi



opruga

U oba slučaja mijenjamo veličinu elementa u X smjeru.

slika opruge

X ovisi o načinu na koji se pozicionira element

EJS u mathematici

■ Java Inicijalizacija

```
Off[General::"spell1"]
```

```
Needs["JLink`"]
```

```
JLink`InstallJava`Private`$UILinkProtocol = "TCP";
```

```
InstallJava[]
```

```
LinkObject[
```

```
  C:\Program Files\Wolfram Research\Mathematica\5.2\SystemFiles\Java\Windows\bin\javaw, 2, 2]
```

```
If[$OperatingSystem == "Windows",
```

```
  apppath = "c:/sertic/applets/", apppath = "/home/maks/sertic/applets/"]
```

```
c:/sertic/applets/
```

```
loadApplet[appletName_, apLpath_] := Module[{l1, apath}, apath = apppath <> apLpath;
```

```
  AddToClassPath[apath]; AppletViewer[appletName, {"width=600", "height=440"}];]
```

podrška za Javu

direktorij s appletima

funkcija koja učitava applet

appleti

```
loadAppletN[appletName_, apLpath_] := Module[{l1, apath},
  apath = apppath <> apLpath; AddToClassPath[apath]; AppletViewer[appletName];

AppletParabollic :=
  loadApplet["parabollicThrow.ParabollicThrowApplet", "Parabollic"]

AppletBall :=
  loadApplet["throwingABall.ThrowingABallApplet", "Mechanics/ThrowingABall.app"]

AppletKosina := loadApplet["kosina.kosinaApplet", "Mehanika/kosina.app"]

AppletTrenje := loadApplet["friction.frictionApplet", "Mehanika/friction.app"]

AppletKola := loadAppletN[
  "cartPulledByString.CartPulledByStringApplet", "Mehanika/CartPulledByString.app"]

AppletKolotura := loadAppletN["kolotura.koloturaApplet", "Mehanika/kolotura.app"]
```

Korištenje appleta

■ Interaktivni dio u javi

U simulaciji se promatra gibanje tijela na kosini bez trenja. Pritiskom na **Start** počinje simulacija. Možemo odabrati nagib kosine i prikazivanje komponentata vektora brzine (prikazan je žutim strelicama) i akceleracije (prikazano crvenim strelicama). U svakom trenutku se može zaustaviti simulacija (**Pauza**), i ponovo pokrenuti (**Start**). Ispocetku se prikazuje graf x i y komponente brzina.

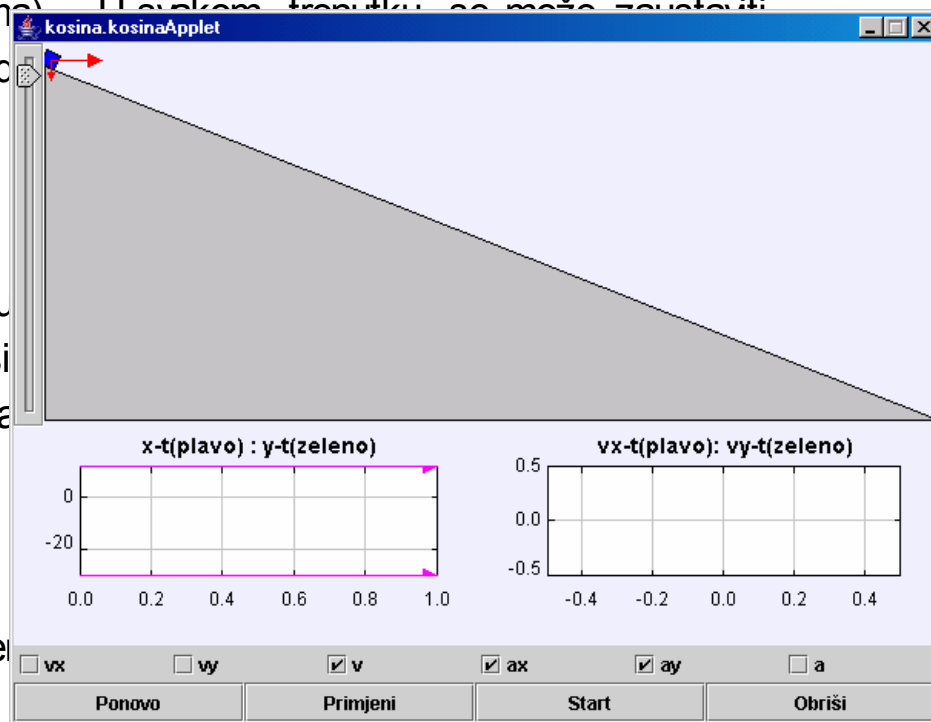
[AppletKosina](#)

Ponovo promatramo gibanje tijela na kosini, među prethodnoj simulaciji možemo promijeniti nagib kosine i koeficijent trenja. Na početku se tijelo nalazi u ravnoteži. Smanjivanjem koeficijenta trenja tijelo više nije u ravnoteži.

[AppletTrenje](#)

Simulacija u kojoj uteg povlaci kolica, položaj kolica može se mijenjati.

[AppletKola](#)



applet u HTML stranici

```
<applet code="spring.SpringApplet.class"
        codebase="." archive="_library/ejsBasic.jar,spring.jar"
        name="Spring" id="Spring"
        width="315" height="248">
</applet>
```

Fileovi potrebni za simulaciju

Javascript

moramo imati _library direktorij

name, id=Spring

_play() - metode

```
<!-- Finally the JavaScript buttons -->
<br><hr width="100%" size="2"><br>
<p>You can control it using JavaScript. For example, using buttons:</p>
<p>
<input type="BUTTON" value="Play" onclick="document.Spring._play();";>
<input type="BUTTON" value="Pause" onclick="document.Spring._pause();";>
```

metode

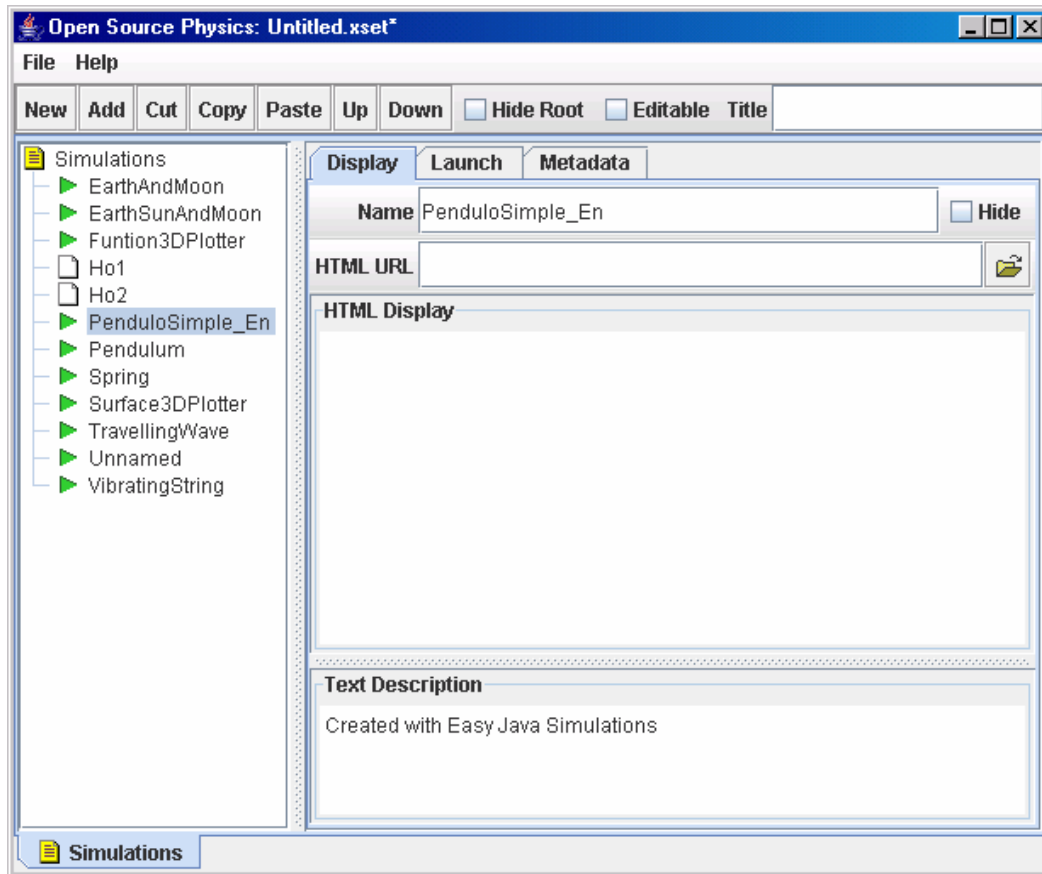
- Javascript može u simulaciji
 - pozvati predefinirane metode (kao `_play()`, `_pause()`)
 - pozvati javne metode koje korisnik definira
 - pročitati ili postaviti varijable u modelu

Metode možemo pozvati javascriptom ili preko hiperlinka:

```
<input type="BUTTON" value="Invoke my method"  
      onclick="document.Spring._model.myMethod();" >
```

```
<a href="JavaScript:document.Spring._model.myMethod();" >  
  click here  
</a>
```


Pregled simulacija



skripta LaunchBuilder

pregled napravljenih simulacija

Instalacija simulacije

- Potrebno je iskopirati
 - sve fileove generirane simulacije
 - `_library` direktorij
- napisati učitavanje appleta
 - HTML - javascript, hiperlink
 - java
 - EJS ili LaunchBuilder
 - Mathematica ili bilo koji program koji poziva applete

Literatura

- Francisco Esquembre, Easy Java Simulations, The Manual.
- EJS web stranica: <http://fem.um.es/Ejs>
- Interaktivne help stranice
http://fem.um.es/Ejs/LibroEjs_en/CD/Reference/Reference.html