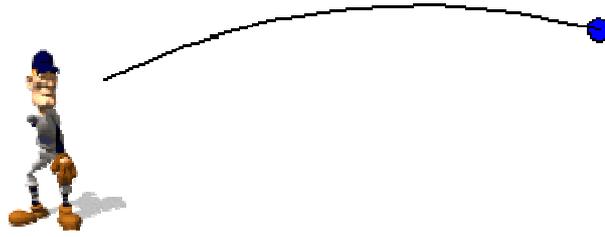


Lemparan bola (lemparan parabolik)



penjelasan

kita mensimulaikan gerak partikel, katakan saja sebuah bola basket, bergerak dalam bidang XY. Bola di lempar oleh seorang pelempar dengan berbagai kecepatan dalam sudut tertentu terhadap arah horizontal. Gesekan udara diabaikan. Gerak kemudian diurai menjadi gerak dalam arah X (arah horizontal) dan gerakan dipercepat pada arah Y.

kita gunakan solusi persamaan yang mendekati:

$$x(t) = x(0) + v_x \cdot t$$

$$y(t) = y(0) + v_y(0) \cdot t - \frac{1}{2} g \cdot t^2$$

$$v_y(t) = v_y(0) - g \cdot t$$

dan berhenti mensimulaikan ketika ball mencapai tanah. Kita masih menggunakan persamaan :

$$x(t + \Delta t) = x(t) + v_x \cdot \Delta t$$

$$y(t + \Delta t) = y(t) + v_y(t) \cdot \Delta t - \frac{1}{2} g \cdot \Delta t^2$$

$$v_y(t + \Delta t) = v_y(t) - g \cdot \Delta t$$

formula ini telah dirubah sehingga memudahkan pengguna menentukan posisi ball secara langsung ketika simulai sedang berjalan, jika diinginkan.

Model

Variabel

Kita membuat variabel dasar seperti berikut:

The screenshot shows the Ejs software interface for a simulation titled 'ThrowingABall.xml'. The 'Model' tab is selected, and the 'Variables' sub-tab is active. A table lists the variables used in the simulation:

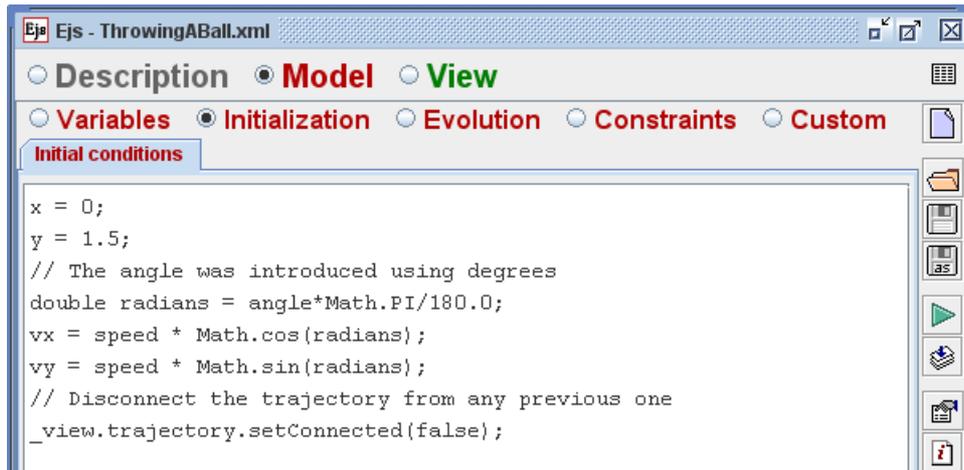
Name	Value	Type	Dimension
g	9.8	double	
speed	8	double	
angle	20	double	
dt	0.1	double	
x	0	double	
y	1.5	double	
vx		double	
vy		double	

Easy Java Simulations step-by-step series of examples

- g adalah percepatan gravitasi, yang diasumsikan konstan
- $speed$ adalah inisial untuk kecepatan yang diberikan pelempar kepada ball
- $angle$ adalah sudut (dalam derajat), yang merupakan sudut antara bidang horizontal dengan bola ketika meninggalkan tangan pelempar.
- dt adalah perubahan waktu tiap posisi bola
- x dan y adalah (horizontal dan vertikal) koordinat dari posisi bola
- v_x dan v_y adalah koordinat dari kecepatan bola.

Inisialisasi

Untuk inisialisasi kita menuliskan code berikut:



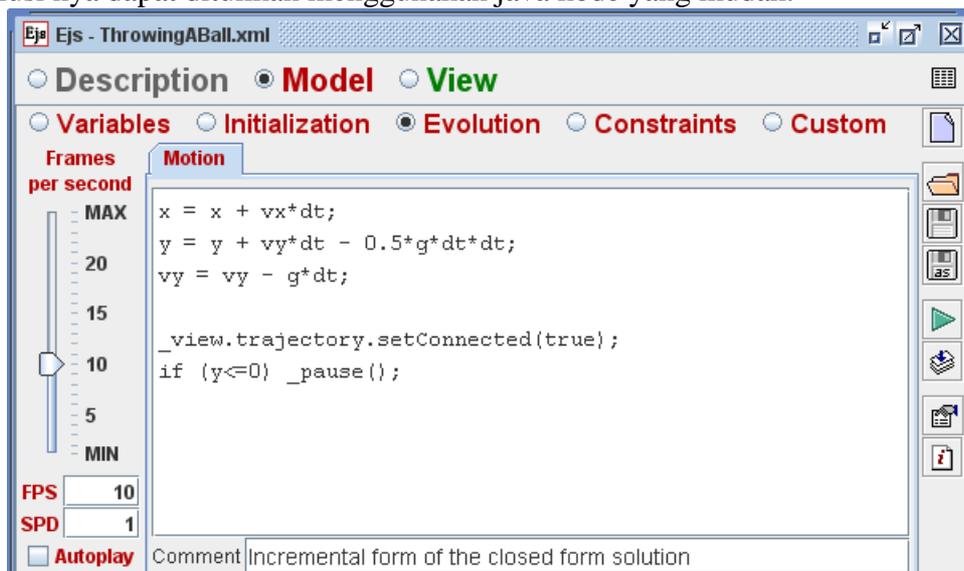
```
x = 0;
y = 1.5;
// The angle was introduced using degrees
double radians = angle*Math.PI/180.0;
vx = speed * Math.cos(radians);
vy = speed * Math.sin(radians);
// Disconnect the trajectory from any previous one
_view.trajectory.setConnected(false);
```

code ini menset kondisi dari koordinat posisi dan kecepatan. Perintah yang khusus: `_view.trajectory.setConnected(false);`

digunakan untuk memutus hubungan lintasan dari yang sebelumnya. Kita memerlukan ini agar pengguna bisa membandingkan perbedaan lintasan. Bagian view trajectory adalah bagian berupa tipe *Trace* dan akan menampilkan lintasan dari bola.

Evolusi

Evolusi nya dapat dituliskan menggunakan java kode yang mudah.



```
x = x + vx*dt;
y = y + vy*dt - 0.5*g*dt*dt;
vy = vy - g*dt;

_view.trajectory.setConnected(true);
if (y<=0) _pause();
```

Comment: Incremental form of the closed form solution

sekarang, perintah:

```
_view.trajectory.setConnected(false);
```

Easy Java Simulations step-by-step series of examples

digunakan untuk membuat lintasan terlihat sebagai garis yang terhubung. 10 frame per second cocok untuk dinal 0.1 yang diberikan terhadap dt .

Constraints

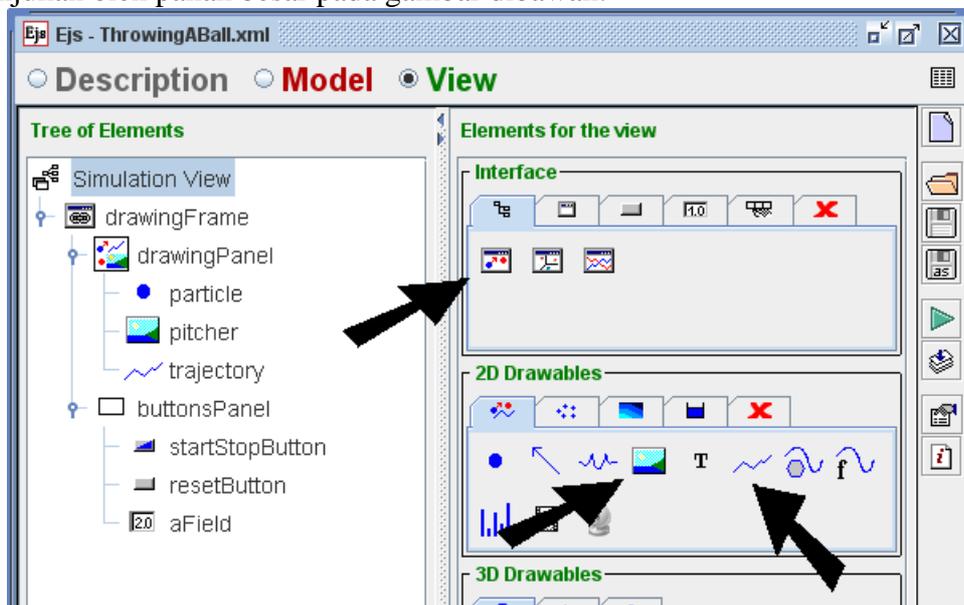
Tidak ada constraints diperlukan

Custom code

Tidak ada custom code required.

View

View dimulai dengan bagian tergabung pada drawing panel dengan partikel yang ada. Kita masukan pada drawing panel bagian *image* dan *trace 2D*, yang disebut *pitcher* dan *trajectory*, untuk keperluan kita, elemen gabungan dan gambar ditunjukkan oleh panah besar pada gambar dibawah.



main properties yang perlu dirubah ditunjukkan pada kolom yang diwarnai pada property panels dibawah:

Properties for drawingPanel (DrawingPanel)

Scales		Configuration	
Autoscale X	false	Square	
Autoscale Y	false	Gutters	
Minimum X	-1	Coordinates	
Maximum X	10	X Format	
Minimum Y	0	Y Format	
Maximum Y	5.5	Messages	
X Margin (%)		Expression	
Y Margin (%)		Expr Format	
Interaction		TL Message	
X		TR Message	
Y		BL Message	
On Press		BR Message	
On Drag		Graphical Aspect	
On Release		Visible	
On Enter		Size	
On Exit		Background	WHITE
Key Action		Foreground	
Key Pressed		Font	
Print Target		Tooltip	

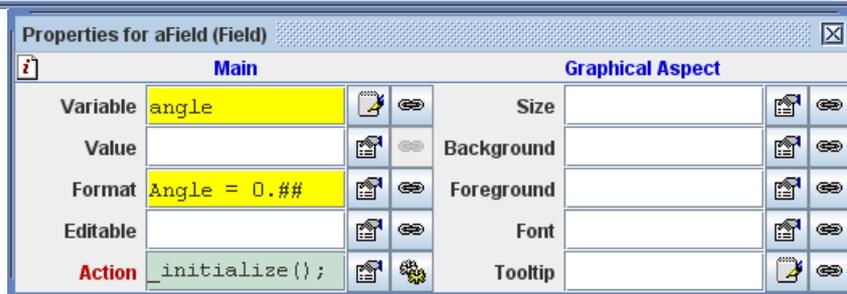
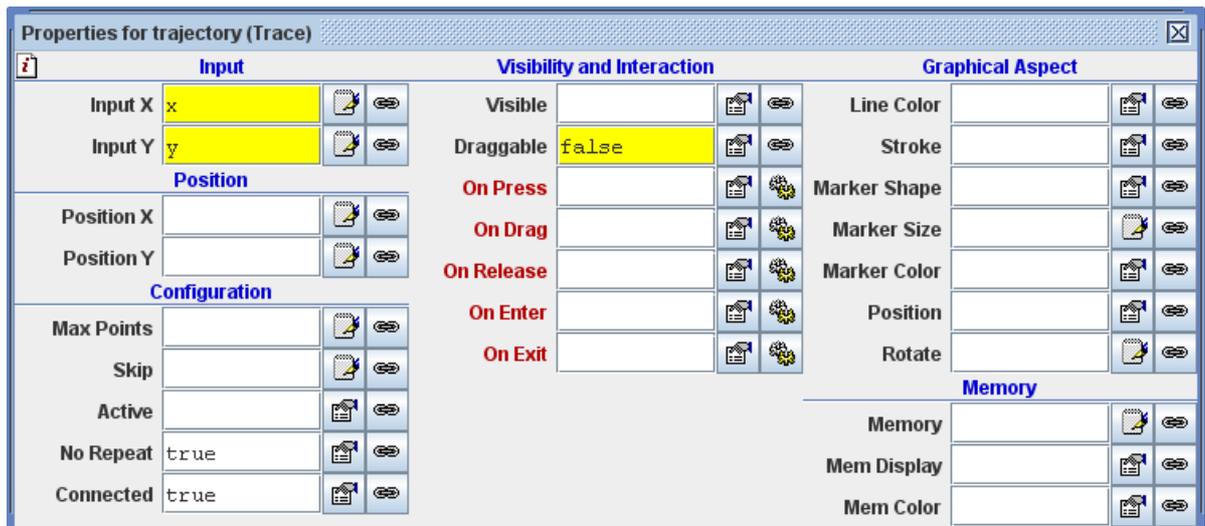
Properties for particle (Particle)

Position and Size		Visibility and Interaction		Graphical Aspect	
X	x	Visible		Style	
Y	y	Draggable	true	Position	
Size X	0.2	Sensitivity		Rotate	
Size Y	0.2	On Press		Line Color	
Pixel Size		On Drag		Fill Color	
Scale X		On Release		Stroke	
Scale Y		On Enter			
		On Exit			

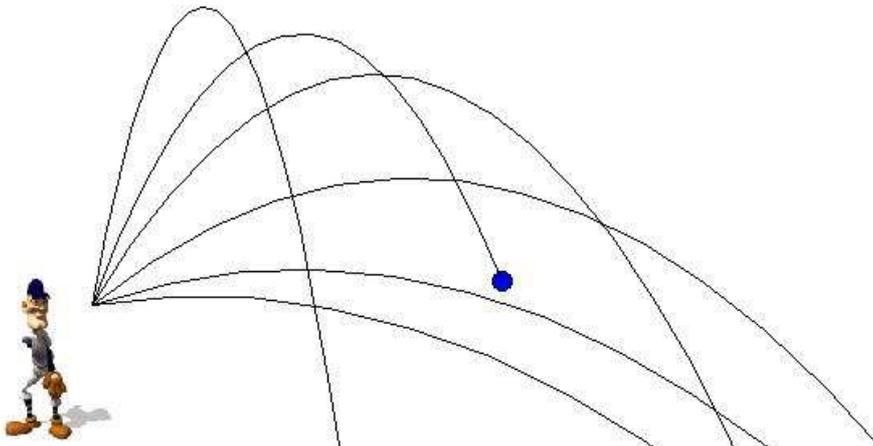
Properties for pitcher (Image)

Position and Size		Visibility and Interaction		Graphical Aspect	
X	-0.5	Visible		Image	./Pitcher.gif
Y	0	Draggable	false	Position	SOUTH
True Size		Sensitivity		Rotate	
Size X	1.6	On Press			
Size Y	1.8	On Drag			
Scale X		On Release			
Scale Y		On Enter			
		On Exit			

gambar *pitcher.gif* dapat ditemukan dalam direktori simulasi ini.



menjalankan simulasi
berikut adalah contoh hasil eksekusi dengan berbagai lintasan:



author
rizal
universitas pendidikan indonesia, bandung
april 2008