

Izdelava simulacij z Easy Java Simulations

Creating Simulations with Easy Java Simulations

Sebastjan Zamuda
Gimnazija Bežigrad, Ljubljana
sebastjan.zamuda@gimb.org

Povzetek

Easy Java Simulations, skrajšano EJS, je odprtokodni program za izdelavo simulacij v Javi. Z EJS lahko preproste izdelamo preproste simulacije brez znanja programiranja. Ob dodatnem programiranju v Javi pa lahko izdelamo kompleksnejše simulacije. V prispevku je na primeru enakomernega gibanja in poševnega meta prikazana izdelava preprostih simulacij po korakih. Obe simulaciji lahko ob besedilu in slikah samostojno izdela vsak učitelj. Prikazano je, kako z majhnimi spremembami spremenimo simulacijo. Simulacije, izdelane z EJS, lahko uporabimo pri pouku za boljšo nazornost razlag, pri domačih nalogah ali v kombinaciji z laboratorijskimi vajami.

Ključne besede: simulacije, Easy Java Simulations, EJS, Java, programiranje, fizika, gibanje

Abstract

Easy Java Simulations, abbreviated EJS, is an open source program for creating Java simulations. EJS can be used to create simple simulations without any knowledge of programming. Using additional programming in Java, more complex simulations can be created. Step by step tutorial is used to show how simple simulations of uniform motion and projectile motion can be created. Any teacher can create both simulations following the descriptions and pictures. It is also demonstrated how simulations can be changed by applying small changes. Simulations created with EJS can be used to better present the topic in class, for homework or in combination with laboratory work.

Keywords: simulations, Easy Java Simulations, EJS, Java, programming, physics, motion

1 Uvod

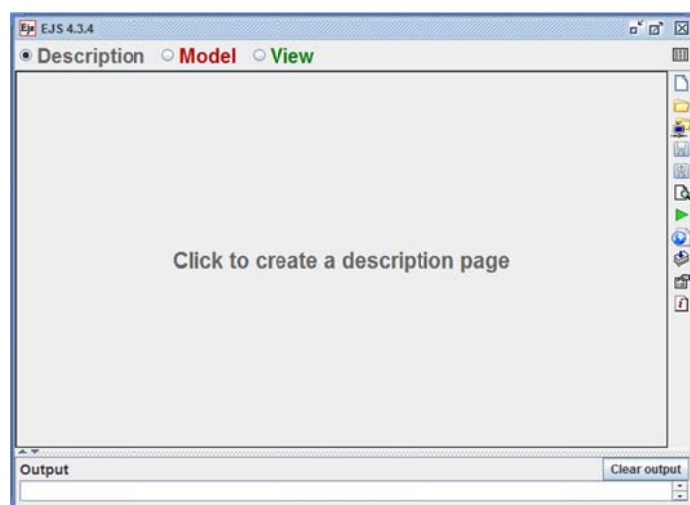
Pri pouku večkrat za boljšo nazornost razlage prikažemo simulacije različnih pojavov. Na ta način lahko obravnavamo pojave, ki jih ne moremo prikazati s poskusom ali posebej poudarimo določene dele prikaza. Na spletu je na voljo veliko brezplačnih simulacij za različne predmete. Kljub velikemu številu simulacij, pa se nam zgodi, da si pri vsaki želimo kakšne spremembe. Ena od možnih rešitev tega izziva je uporaba programa Easy Java Simulations ali skrajšano EJS (Easy Java Simulations, 2012). Za veliko učiteljev je ta program privlačen, ker za izdelavo osnovnih simulacij ni potrebno znanje programiranja. V nadaljevanju je na primeru prikazana izdelava preproste simulacije, ki jo lahko izdelava vsak učitelj sam.

2 Predstavitev Easy Java Simulations (EJS)

Easy Java Simulations (EJS) je odprtokodni program, napisan v Javi, s katerim lahko izdelamo interaktivne simulacije v Javi, predvsem v izobraževalne namene. EJS je razvit neodvisno, a je tudi del projekta Open Source Physics (Open Source Physics, 2012), ki združuje različne odprtokodne programe oziroma orodja v digitalni knjižnici ComPADRE (ComPADRE, 2012). Glede na znanje programiranja lahko uporabljamo program EJS na različnih nivojih. Brez poznavanja programiranja v Javi lahko izdelamo preproste simulacije. Nekaj takih primerov si bomo ogledali v nadaljevanju. Če smo veščji programiranja v Javi, lahko svoje delo nadgradimo in izdelamo tudi kompleksnejše simulacije. Odpremo in spremenimo lahko tudi simulacije, ki jih najdemo na spletu, če so bile te narejene z EJS. Dober vir takih simulacij je prav prej omenjeni projekt Open Source Physics.

Za uporabo EJS moramo imeti na računalniku nameščeno Javo. Zatem EJS v obliki ZIP datoteke brezplačno snamemo domače spletne strani EJS (Easy Java Simulations, 2012). Shranjeno datoteko razpakiramo in zaženemo datoteko *EjsConsole.jar*. Navodila za namestitve in podrobnosti glede različnih verzij so dosegljiva na domači spletni strani EJS. Tam najdemo tudi navodila za uporabo, ki pa so za začetnika relativno dolga in zapletena.

Po zagonu se odpreta dve okni. Manjše okno je konzola, ki je pri osnovni uporabi ne bomo uporabljali. V drugem oknu pa pripravimo simulacijo (slika 1).



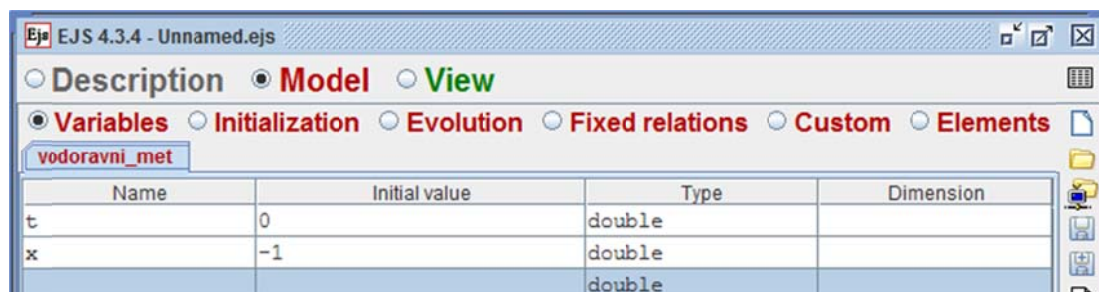
Slika 1: Začetni pogled v programu

3 Primer izdelave simulacij z EJS

V nadaljevanju si bomo ogledali predvsem korake, ki so nujno potrebni za izdelavo preprostih simulacij in na nekaterih mestih le omenili dodatne možnosti, nikakor pa ne bomo spoznali vseh možnosti, ki jih program ponuja. Prikazana bosta primera iz fizike, uporaba pa je možna tudi na drugih področjih. Namenoma sta izbrana zelo preprosta primera, tako v fizikalnem smislu kot v smislu enostavnosti uporabe EJS. Na ta način smo se sicer odrekli zanimivejšim primerom, vendar bo tudi nov uporabnik programa lahko sledil vsem korakom.

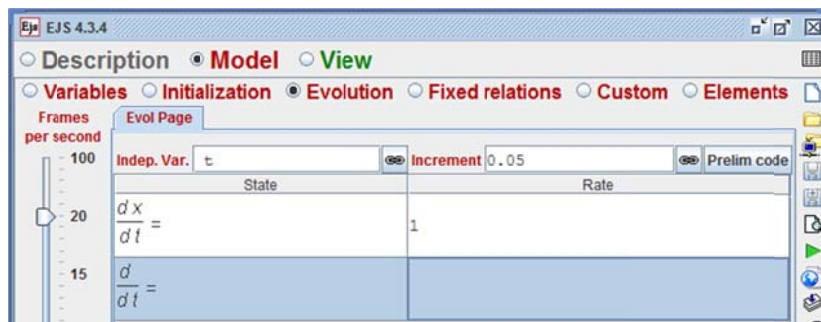
Na več korakih bo potrebno vnesti ime okna, ki je lahko poljubno, zato je ta korak je v nadaljnjem opisu izpuščen. Na vrhu okna na sliki 1 lahko izberemo med opisom simulacije (Description), izdelavo modela (Model) in nastavitvami prikaza simulacije (View). Ko izberemo opis simulacije (Description), lahko kliknemo na sredino okna, vnesemo poljubno ime strani in v urejevalniku opišemo simulacijo. Na ta del se lahko vrnemo tudi kasneje, ko bo simulacija deloma ali v celoti dokončana. Opis lahko tudi preskočimo, saj je lahko uporabniku le v pomoč, ni pa nujen za izvajanje simulacije.

Za boljšo ilustracijo si oglejmo primer izdelave simulacije enakomerne gibanja, pri katerem se bo telo gibalo od leve na desno stran okna. Za izvajanje simulacije je ključen drugi del (Model na sliki 1). Tu določimo spremenljivke, ki jih bomo uporabili, ter nastavimo njihove začetne vrednosti in kasnejše spreminjanje. Po kliku na "Click to create a page of variables" in vnosu poljubnega imena za to stran definiramo spremenljivke. V našem primeru bosta to čas t in lega x . Za spremenljivko t določimo začetno vrednost nič in za x začetno vrednost -1 (to pomeni levi rob okna simulacije), kot je prikazano na sliki 2. Decimalna števila moramo vnesti z decimalno piko, kot je običajno v angleščini.



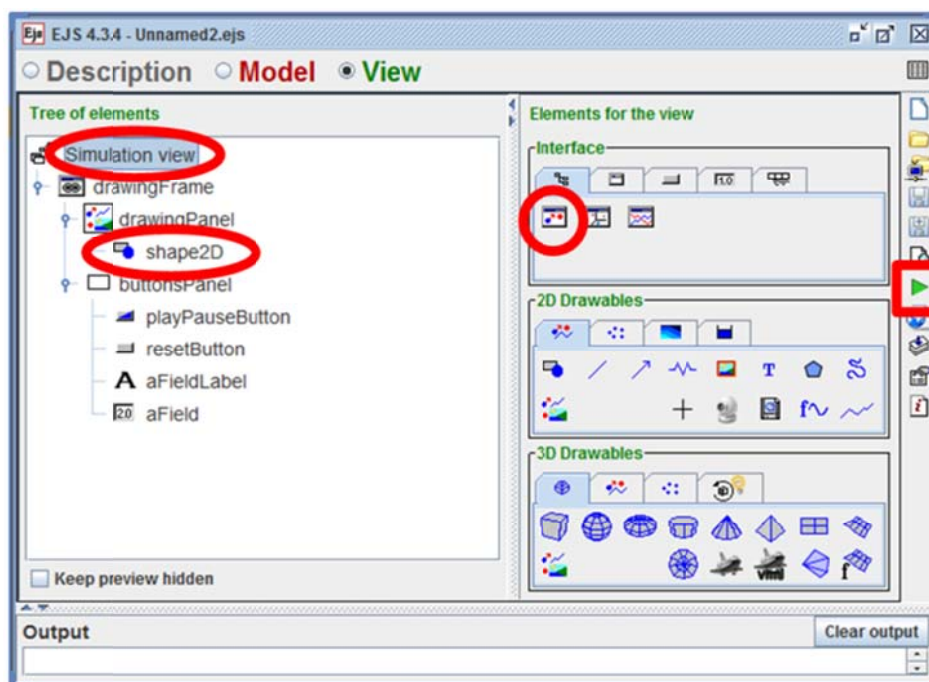
Slika 2: Definiranje spremenljivk

V naslednjem koraku določimo, kako se spremenljivke spreminjajo (slika 3). Kliknemo na Evolution in zatem na Click to create a page of ODEs. V polje Indep. Var. vnesemo neodvisno spremenljivko, kar je v našem primeru čas (t). V naslednjem polju Increment je že določena vrednost 0,05, ki vpliva na hitrost izvajanja simulacije in jo lahko po želji spremenimo. V polje State vnesemo x in v polje Rate vnesemo vrednost 1. Na ta način smo zapisali diferencialno enačbo $dx/dt = 1$ ali preprosteje povedano, določili smo, da je vrednosti hitrosti 1 (v relativnih enotah). Primernost izbranih vrednosti konstant bomo določili ob izvajanju simulacije na koncu.



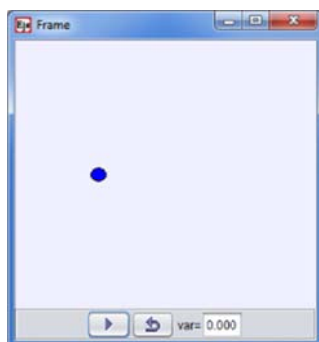
Slika 3: Določanje formul za spreminjanje uporabljenih spremenljivk

V prejšnjem koraku smo določili, kako se bodo spreminjale spremenljivke, poskrbeti moramo le še za prikaz gibanja. Kliknemo na View (slika 4), v desnem delu okna kliknemo na ikono DrawingFrame (na sliki je označena s krogom) in nato na Simulation view v levem delu okna, kjer se prikažejo dodatne lastnosti. Kliknemo na krogec pred drawingFrame, da razširimo drevesno strukturo. Po desnem kliku na shape2D in izbiri Properties se odpre novo okno. V polje Pos X vnesemo x. S tem smo določili, da se bo objekt (moder krogec) premikal tako, kot določa spremenljivka x. Okno zapremo s klikom na križec v zgornjem desnem kotu.



Slika 4: Nastavitve za prikaz okna s simulacijo

Preostane nam le še zagon simulacije s klikom na ikono Run simulation. To je zelen trikotnik na desni strani, ki je na sliki 4 označen z rdečim kvadratom. Po kliku na ta gumb najprej shranimo simulacijo, zatem se odpre novo okno (slika 5), v katerem se izvede simulacija, ki smo jo pripravili. V tem oknu sta samodejno prikazana gumba za zaustavitev in ponovni zagon simulacije. Simulacijo prekinemo in zapremo s ponovnim klikom na ikono Run simulation, na kateri je med izvajanjem simulacije trikotnik obarvan rdeče.



Slika 5: Okno z zagnano simulacijo

Ko poznamo do sedaj opisane osnovne korake, lahko preprosto naredimo še druge simulacije. Za poševni met uporabimo zgoraj opisani postopek, pri čemer v ustrezna polja vnesemo vrednosti, prikazane na spodnjih slikah.

Name	Initial value	Type	Dimension
t	0	double	
x	-0.9	double	
y	-0.9	double	
		double	

Slika 6: Spremenljivke za poševni met

Evol Page	
State	Rate
$\frac{dx}{dt} =$	0.4
$\frac{dy}{dt} =$	1.9-t
$\frac{d}{dt} =$	

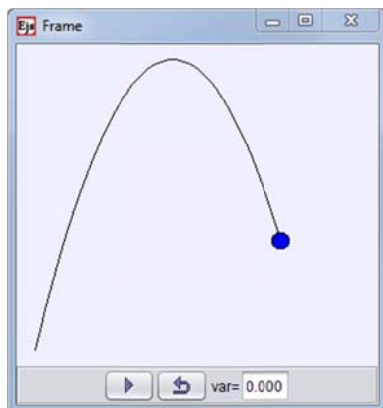
FPS: 20
 SPD: 1
 Solver: Cash-Karp 5(4)
 Tol: 0.00001
 Advanced: 0
 Events: 0

Slika 7: Formule za poševni met

Za vodoravno komponento hitrosti, ki jo običajno označimo z v_x , smo tokrat določili manjšo vrednost ($dx/dt = 0,4$). Za spreminjanje koordinate y smo uporabili formulo za navpično komponento hitrosti $v_y = v_{y,0} - gt$, kjer je $v_{y,0} = 1,9$, za g pa smo nastavili kar vrednost 1, saj gre v vseh primerih za relativne enote, ki niso bistvene, če nas zanima le vrsta gibanja in tir

telesa, ne pa tudi vrednosti hitrosti in drugih količin. Če ne želimo, da se simulacija zažene samodejno po prikazu okna, lahko na sliki 7 odstranimo kljukico pri Autoplay.

Kot smo že omenili, omogoča EJS veliko več, kot smo prikazali. Za prikaz gibanja lahko dodamo tudi sled (slika 8), lahko bi uporabili 3D objekte, ki se vrtijo okrog različnih osi, vektorje, gumbe za interakcijo uporabnika med izvajanjem simulacije in drugo. Uporabimo lahko tudi druge primere gibanja, na primer gibanje teles ob upoštevanju zračnega upora. Z dodatnim programiranjem lahko dodamo trke ob steno ali med telesi. Poleg zgoraj opisane izdelave in izvajanja simulacije lahko simulacijo tudi shranimo kot datoteko JAR in zaženemo samostojno. Simulacijo lahko vključimo tudi v svojo spletno stran.



Slika 8: Prikaz sledi gibanja

4 Uporaba EJS pri pouku

Opisan postopek lahko uporabimo za izdelavo preprostejših ali zahtevnejših animacij. Za primere osnovnih gibanj je na razpolago veliko kvalitetnih prosto dostopnih simulacij, zato bomo običajno uporabili te. Kadar so bile izdelane z EJS, jih lahko spremenimo po svojih potrebah. EJS je posebej uporaben za izdelavo simulacij, pri katerih želim izpostaviti določen pojav ali omogočiti prav določen način interakcije. Že pri nadgradnji prej uporabljenega poševnega meta zlahka najdemo tak primer. Če želimo upoštevati zračni upor in morda ročno spreminjati hitrost vetra, bomo na spletu težko našli prav tako simulacijo.

Po želji prirejene simulacije lahko uporabimo tudi za domače naloge. Namesto da dijaki doma rešujejo naloge iz učbenikov, lahko to v nekaterih primerih nadomestimo z interaktivno simulacijo, s katero dijaki rešijo zastavljeno nalogo.

Zanimiv primer uporabe EJS je prikazal Marko Rožič (Rožič, 2012). Izdelal je simulacijo, ki posnema eksperiment, ki ga dijaki izvajajo pri laboratorijskih vajah. Take simulacije lahko uporabimo za predhodno pripravo dijakov na vaje ali kot njihovo nadomestilo, kadar za izvajanje eksperimenta nimamo časa ali opreme.

5 Zaključek

Obravnavo pojavov pri pouku fizike in pri drugih predmetih lahko popestrimo s simulacijami. S programom Easy Java Simulations (EJS) lahko izdelamo simulacije, ki bodo ustrezale našim potrebam. Kot je prikazano v prispevku, je izdelava osnovnih simulacij preprosta. EJS

ponuja tudi veliko možnosti za nadgradnjo, tudi s programiranjem v Javi. Simulacije lahko uporabimo za boljšo predstavitev učne snovi, za domače naloge ali kot pripravo na laboratorijsko delo.

Viri in literatura

ComPADRE: <http://www.compadre.org/> (8. 6. 2012)

Easy Java Simulations: <http://fem.um.es/Ejs> (8. 6. 2012)

Open Source Physics: <http://www.compadre.org/osp> (8. 6. 2012)

Rožič, M. (2012), Interaktivna simulacija z EJS - Easy Java Simulation. Mednarodna multikonferenca Splet izobraževanja in raziskovanja z IKT – SIRikt 2012 (zbornik), 1233–1239.