

SVEUČILIŠTE U ZAGREBU  
PRIRODOSLOVNO-MATEMATIČKI FAKULTET  
FIZIČKI ODSJEK

SMJER: PROFESOR FIZIKE

**Gordan Tomić**

Diplomski rad

**METODIČKO OBLIKOVANJE  
INTERAKTIVNIH SIMULACIJA  
PROGRAMSKIM PAKETOM EJS**

Voditelj diplomskog rada: prof. dr. sc. Darko Androić

Ocjena diplomskog rada: \_\_\_\_\_

Povjerenstvo: 1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

Datum polaganja: \_\_\_\_\_

Zagreb, 2011.

# Sadržaj

---

1. Uvod .....	3
2. Simulacija .....	4
2.1. Povijest .....	4
2.2. Definicija i upotreba .....	4
3. Alat .....	7
3.1. Općenito o EJS-u .....	7
3.2. Instalacija i pokretanje programa .....	7
3.3. Pregled simulacije .....	11
3.3.1. <i>Description</i> panel .....	13
3.3.2. <i>Model</i> panel .....	14
3.3.3. Definicija varijabli .....	14
3.3.3.1. Inicijalizacija modela .....	15
3.3.3.2. Evolucija modela .....	16
3.3.3.3. Veze među varijablama .....	18
3.3.3.4. <i>Custom</i> panel .....	18
3.3.4. <i>View</i> panel .....	19
3.4. Pokretanje simulacije .....	22
3.5. Distribucija simulacije .....	22
3.6. Ostale mogućnosti EJS-a .....	22
4. Upotreba simulacije u nastavi .....	25
5. Zaključak .....	30
6. Literatura .....	31

# 1. Uvod

---

Tehnološkom evolucijom čovječanstvu se otvorila mogućnost simuliranja vrlo kompleksnih sistema. Nekim simulacijama moguće je prikazati sisteme koje ljudi još stoljećima neće biti u mogućnosti uistinu ostvariti, no ambijent je uspješno simuliran. Simulacija se koristi u gotovo svim tehnološkim granama koje čovjek može zamisliti i služi za vježbu u kontrolirano stvorenim uvjetima bez potrebe za rizikom gubitaka ljudskih života i financijskih sredstava.

Ovaj diplomski namijenjen je onima koji su došli u doticaj s programom Easy Java Simulation i žele saznati kako taj program funkcionira. Diplomski rad zamišljen je kao uputa za izradu simulacije. Oni koji pročitaju diplomski rad trebali bi dobiti ideju kako program funkcionira i, ono što je još bitnije, dobiti ideju za izradu svoje simulacije. Ovaj program, koliko god primitivno izgledao na prvi pogled, vrlo je moćan alat za stvaranje kompleksnih simulacija.

U drugom poglavlju dan je općenit uvod o simulacijama: povijest simulacija, definicija i upotrebe simulacija.

U trećem poglavlju opisuje se funkcioniranje alata (EJS) preko izrađene simulacije horizontalnog hica. Ovdje su prikazani dijelovi programa koji su ključni pri stvaranju simulacije.

U četvrtom poglavlju opisana je upotreba simulacije horizontalnog hica u nastavi fizike. Ovdje se govori o pozitivnim i negativnim stranama upotrebe simulacije u nastavi te gdje bi se simulacija trebala, a gdje se ne bi trebala koristiti.

Peto poglavlje je zaključak u kojem su opisane mogućnosti alata te problemi na koje sam naišao tokom izrade ovog diplomskog rada.

Uz diplomski rad dolazi CD na kojem je kratak video s uputama o izradi simulacije.

## 2. Simulacija

---

### 2.1. Povijest

Jedan od pionira koncepta simulacije bio je John von Neumann. U kasnim 40-im godinama prošlog stoljeća koncipirao je ideje repetitivnog ponavljanja modela i sakupljanja statističkih podataka te na osnovu ponašanja tih modela proučavao je ponašanje realnih sustava. Ta metoda je danas poznata kao Monte Carlo metoda<sup>1</sup> zbog uporabe nasumce generiranih slučajnih varijabli za pokazivanje ponašanja koja se nisu mogla egzaktno modelirati, ali su se mogla karakterizirati statistički. Von Neumann je koristio tu metodu za proučavanje neutrona koji se nasumično gibaju i efektivnosti avionskih bombardiranja. Kasnije su se definirali različiti koncepti simulacija među kojima se nalaze DES (Discrete Event Simulations) i GSP (General Simulation Program), razvijeni u General Electricu.

### 2.2. Definicija i upotreba

Simulacija je proces dizajniranja stvarnog ili imaginarnog modela nekog sustava koji nam omogućava izvođenje eksperimenata. Svrha simulacijskih eksperimenata je upoznavanje ponašanja nekog sustava te razvijanje strategije za izvođenje operacija na tom sustavu. Pomoću pretpostavki o sustavu generiraju se matematički algoritmi i veze koji stvaraju model za otkrivanje funkcioniranja sustava.

Ukoliko je sustav jednostavan, model se može opisati analitički: npr. jednostavna jednažba kao  $\text{put} = \text{brzina} \cdot \text{vrijeme}$  je analitičko rješenje koje predstavlja udaljenost koju objekt prevali pri konstantnoj brzini u nekom vremenskom intervalu.

Međutim, zanimljivi problemi stvarnog svijeta inače su puno kompleksniji od gore navedenog problema. Zapravo, problemi mogu biti toliko kompleksni da ih jednostavni matematički modeli ne mogu opisati. U tom slučaju ponašanje sustava se mora utvrditi kroz simulaciju. Točna prezentacija sustava je rijetko moguća, što znači

---

<sup>1</sup> ime je dobila po casinu

da u obzir dolazi samo dovoljno dobra aproksimacija. Postoje razni modeli za gotovo sve sustave: tvornice, komunikacijske i računalne mreže, integrirane krugove, mreže autocesta, dinamiku leta, nacionalnu ekonomiju, socijalne interakcije, izmišljene svijetove, itd. U svim navedenim okolinama model sustava je smanjio rizik i omogućio isplativost. Drugim riječima, puno je praktičnije eksperimentiranje na modelu nego na pravom sustavu.

Kao primjer navest ćemo izradu nove tvornice koja bi zamijenila staru. Ukoliko nije sigurno da li će povećanje produktivnosti pokriti ulaganje, potrebno je izraditi simulaciju koja pokazuje model nove tvornice. Model treba sadržavati potrebnu kvadraturu, broj strojeva, broj zaposlenika, položaj opreme, kapacitet produktivnosti svakog stroja i vrijeme čekanja između strojeva. Simulacije u tom slučaju omogućuju procjenu vrijednosti sustava, procjenu kapaciteta produktivnosti i cijenu nove tvornice. Te su informacije neprocijenjive pri donošenju odluka i nema potrebe za gradnjom stvarne tvornice da bi se dobili potrebni odgovori.

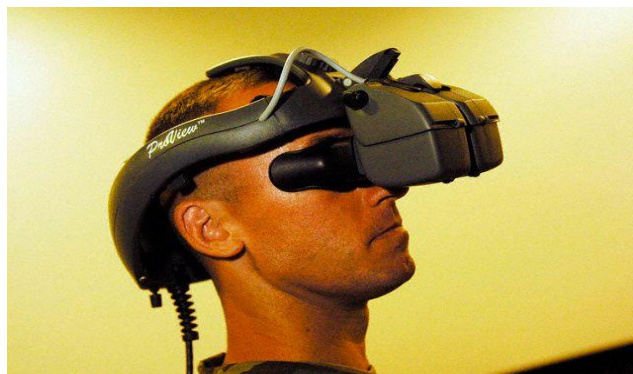
Simulacije se obično određuju prema događajima koji mogu biti diskretni ili kontinuirani, ovisno o načinu mijenjanja varijabli stanja. Diskretni događaji su određeni trenutnim mijenjanjem varijabli stanja u određenoj točki vremena. U kontinuiranoj simulaciji varijable se mijenjaju kontinuirano, obično kroz funkciju u kojoj je varijabla vrijeme.

Simulacija omogućava analizu sustava, tj. pokazuje njegove mogućnosti, kapacitete i ponašanja bez potrebe za gradnjom ili eksperimentiranjem s pravim sistemom. Današnje tehnike koriste se za dizajniranje novih sistema, analizu već postojećih sistema, treninge za sve vrste aktivnosti te za interaktivnu zabavu. Simulacija se koristi u gotovo svim inženjerskim, znanstvenim i tehnološkim disciplinama.

Simulacija se također može koristiti za predviđanje budućih događaja koji nisu predvidivi, kao npr. simulacija vremenskih prilika.



**Slika 2.1.** Simulator leta



**Slika 2.2.** Upotreba simulacije za zabavu



**Slika 2.3.** Simulacija crne rupe



**Slika 2.4.** Simulacija zgrade

## 3. Alat

---

### 3.1. Općenito o EJS-u

EJS (Easy Java Simulation) je alat za modeliranje koji omogućuje znanstvenicima (i to ne nužno računalnim stručnjacima) laku kreaciju simulacija. S tehničkog gledišta, EJS olakšava modeliranje te omogućuje jednostavnu, ali i moćnu konceptualnu strukturu za izradu simulacija.

Alat pruža red radnih zaslona koji se koriste za implementaciju modela i stvaranje grafičkog sučelja. EJS automatizira zadatke kao što su numeričko računanje uobičajenih diferencijalnih jednadžbi i animacije. Komunikacija između programa i krajnjeg korisnika je omogućena bez potrebe za *low-level* programiranjem.

Očito, dio zadatka ovisi o krajnjem korisniku. Krajnji je korisnik odgovoran za nalaženje određenog modela nekog fenomena te dizajniranje i odabir kranjeg izgleda modela. EJS pokazuje svoje dobro lice upravo kod zadataka visokog stupnja koji su više vezani uz samu znanost nego uz programiranje. Korisnik se potiče da svoje vrijeme i energiju posveti učenju znanosti, tj. na nešto što samo računalo ne može.

### 3.2. Instalacija i pokretanje programa

EJS je Java program koji se može koristiti na svim operativnim sustavima koji podržavaju Java Virtual Machine (VM). Pošto je Java dizajnirana da bude neovisna o platformama, EJS sučelje na Mac OS X, Unix-u i Linux-u identično je onom na Windows operativnom sustavu. Ovdje je dan opis za Windows.

#### 1. Instalacija Java Runtime Environment.

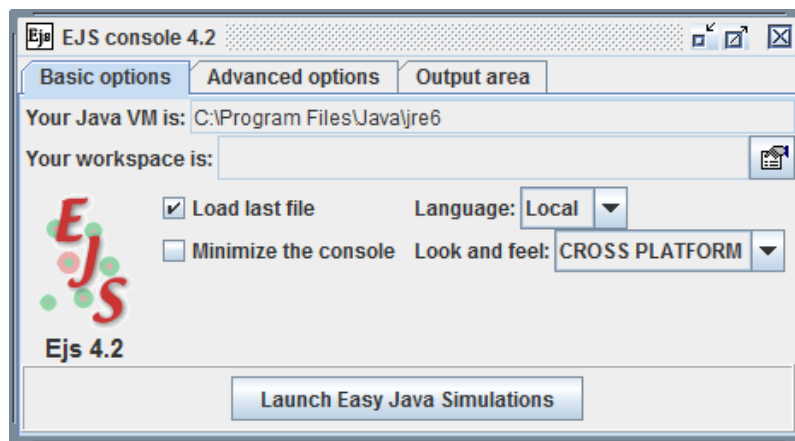
EJS zahtjeva Java Runtime Environment (JRE), 1.5 ili kasniju verziju. Postoji mogućnost da je JRE već instaliran na računalu, a u slučaju da nije, potrebno je posjetiti stranicu <http://java.sun.com> i pratiti instrukcije za download i instalaciju.

## 2. Kopiranje EJS-a na tvrdi disk.

EJS se nalazi u ZIP formatu (EJS\_X.x\_ggmmdd.zip), gdje X.x označava broj verzije EJS-a, ggmmdd dan, mjesec i godinu verzije (npr. moguće je naići na nešto ovako EJS\_4.2\_090901). Za novije verzije potrebno je posjetiti stranicu: <http://www.um.es/fem/Ejs>. Kada se raspakira ZIP format na računalo (na tvrdi disk), dobiva se direktorij EJS\_X.x (npr. EJS\_4.2). Ovaj direktorij sadrži sve što je potrebno za pokretanje EJS-a.

## 3. Pokretanje EJS konzole.

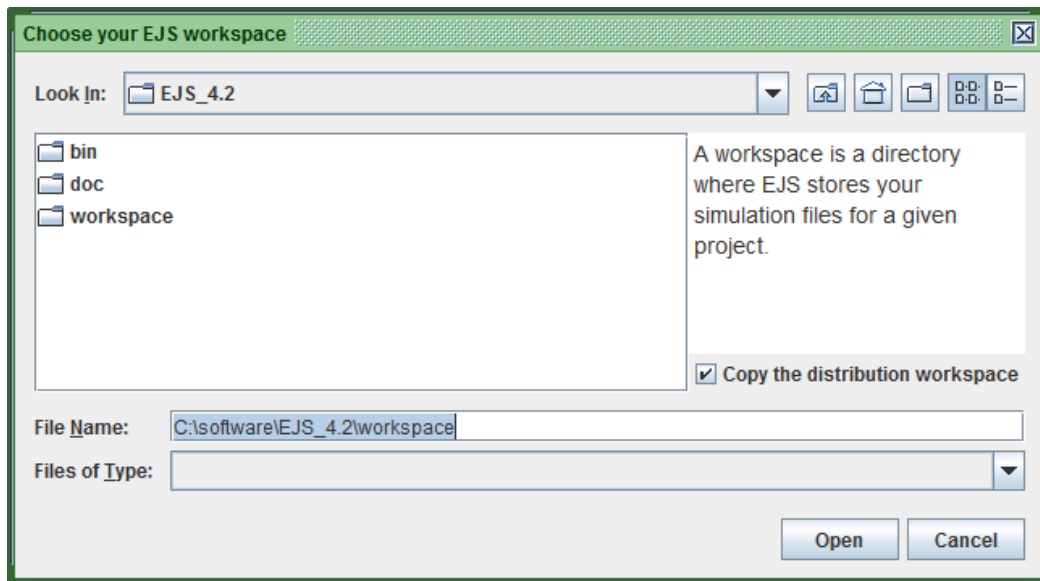
Unutar novonapravljenog direktorija nalazi se file koji se zove *EjsConsole.jar*. Duplim klikom miša pokreće se EJS konzola (Slika 3.1.) i *file chooser* dijalog (Slika 3.2.) koji će biti opisan kasnije.



Slika 3.1. EJS konzola

EJS konzola služi za otvaranje jednog ili više kopija EJS-a te za obavljanje zadataka vezanih uz EJS. Konzola se može koristiti za prilagodbu izgleda i ponašanja pri pokretanju EJS-a (npr. na *look and feel* se mogu mijenjati razni izgledi). Konzola također pokazuje informacije vezane uz EJS program i poruke na *Output area* području.



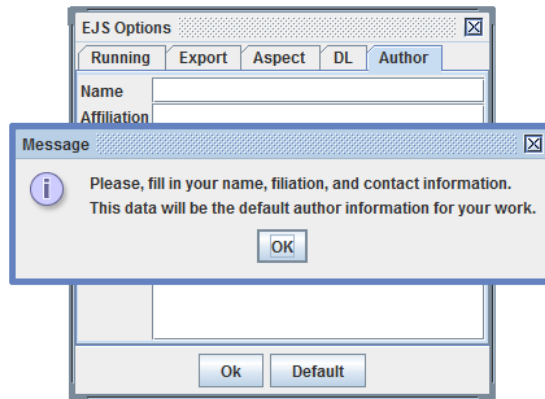


**Slika 3.2.** *File chooser* dijalog. Kada se prvi put pokrene EJS, konzola traži da se izabere *workspace* direktorij.

Prije pokretanja EJS-a, a nakon instalacije, pojavit će se *file chooser* (Slika 3.2.) koji omogućuje odabir direktorija na tvrdom disku koji se koristi kao radni prostor (*workspace*). *Workspace* je direktorij na tvrdom disku gdje EJS sprema podatke o simulaciji. *Workspace* može sadržavati neograničen broj simulacija. Unutar *workspace* direktorija EJS stvara četiri potdirektorija:

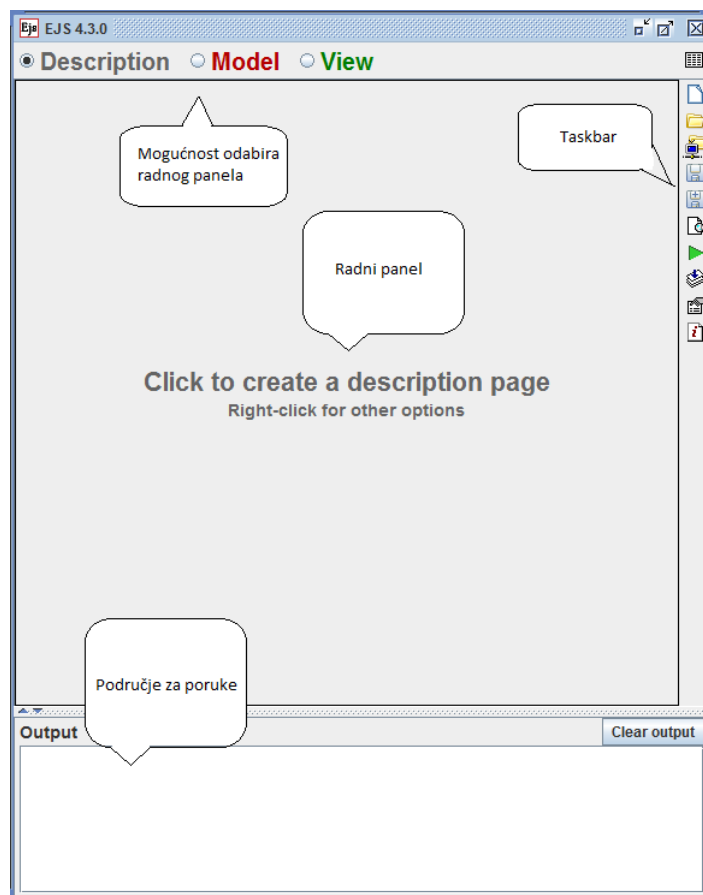
- **config** je direktorij za podatke i opcije koji su definirani od korisnika,
- **export** je direktorij gdje se spremaju sve gotove simulacije,
- **output** je direktorij koji EJS koristi za spremanje trenutno generiranih podataka pri izradi simulacije
- **source** je direktorij u kojem se nalaze svi bitni podaci vezani uz simulacije koje su izrađene.

Također, kada se prvi put pokrene, EJS program traži da napišete svoje ime i grupaciju (Slika 3.3.).



**Slika 3.3.** Prozor za upis podataka o korisniku

Nakon toga se otvara prozor za modeliranje prikazan na slici 3.4. koji, unatoč svom jednostavnom izgledu, ima vrlo velike mogućnosti.



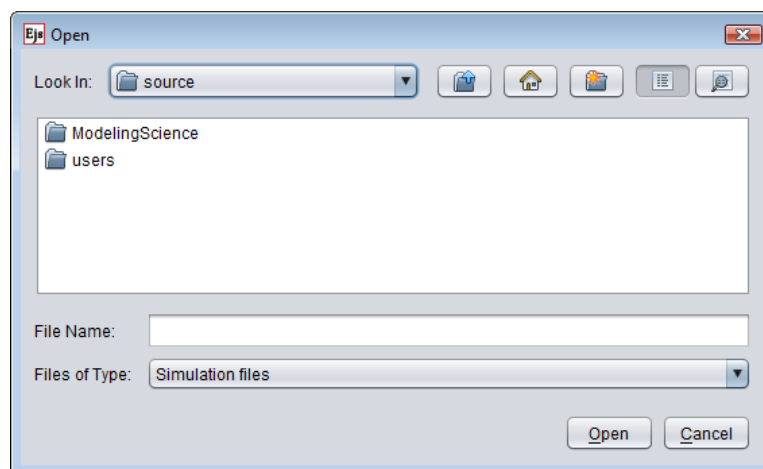
**Slika 3.4.** Prozor za modeliranje

*Taskbar* omogućuje niz ikona koje služe za brisanje, otvaranje, pretraživanje, spremanje podataka, konfiguriranje EJS-a, informacije i pomoć oko EJS-a, što je vrlo korisno. Desnim klikom na neku od ikonica otvara se prozor s mogućim alternativnim, ali povezanim akcijama. Donji dio sučelja sadrži prostor za izlazne poruke koje program tijekom modeliranja izbacuje. Centralni dio sučelja sadrži radne panele gdje se vrši modeliranje.

EJS omogućuje tri radna panela za modeliranje. Prvi panel naziva *Description*, omogućava korisniku stvaranje i editiranje opisa modela baziranog na HTML skripti (npr. neki zadatak za učenike, opis problema, jednadžbe gibanja, itd.). Drugi panel naziva *Model*, posvećen je procesu modeliranja. Ovaj model se koristi za stvaranje varijabli koje opisuju model, početnih uvjeta tih varijabli i upis algoritama koji opisuju kako se model mijenja u vremenu. Treći panel, naziva *View*, posvećen je zadatku građenja grafičkog sučelja te omogućuje korisniku da kontrolira simulaciju i grafički je prikazuje. Izabirom elemenata s palete i dodavanjem na *Tree of elements*, koji se nalazi unutar panela *View*, gradi se grafičko sučelje. Paleta sadrži gumbe, klizače, itd...

### 3.3. Pregled simulacije

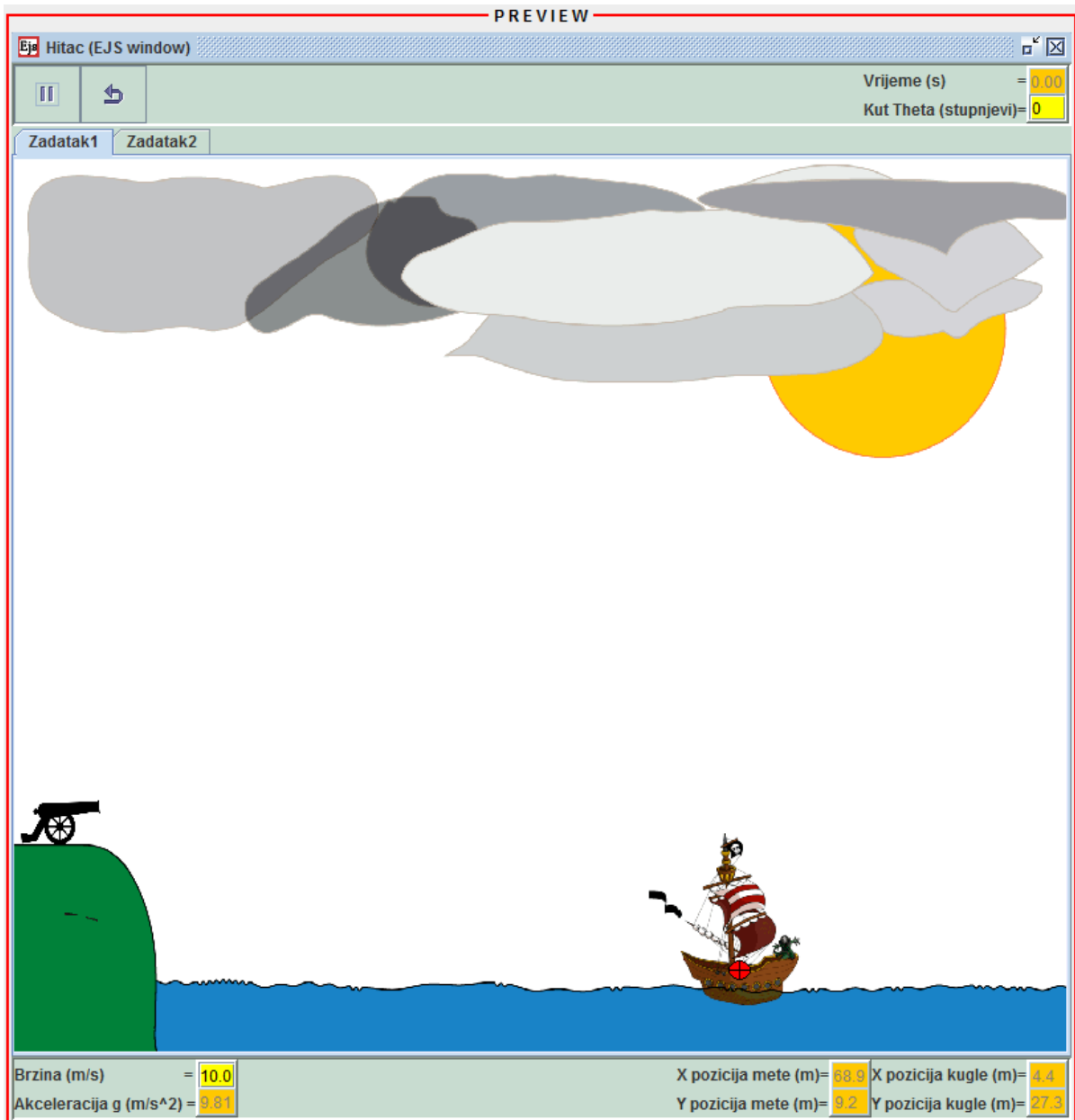
Za bolje razumijevanje rada pojedinih panela, nabolje je prolaziti kroz program za već gotovu simulaciju. Klikom na *Open* ikonu, koja se nalazi na taskbaru u EJS-u, otvara se prozor prikazan na slici 3.5.



**Slika 3.5.** *Open file* dijalog omogućuje pregled tvrdog diska i učitavanje postojeće simulacije

U ovom slučaju otvaranjem direktorija *users* otvara se izbor potdirektorija. Unutar ovog direktorija može se naći file *horizontalni hitac.ejs*. Potrebno je označiti taj file i stisnuti gumb *Open*.

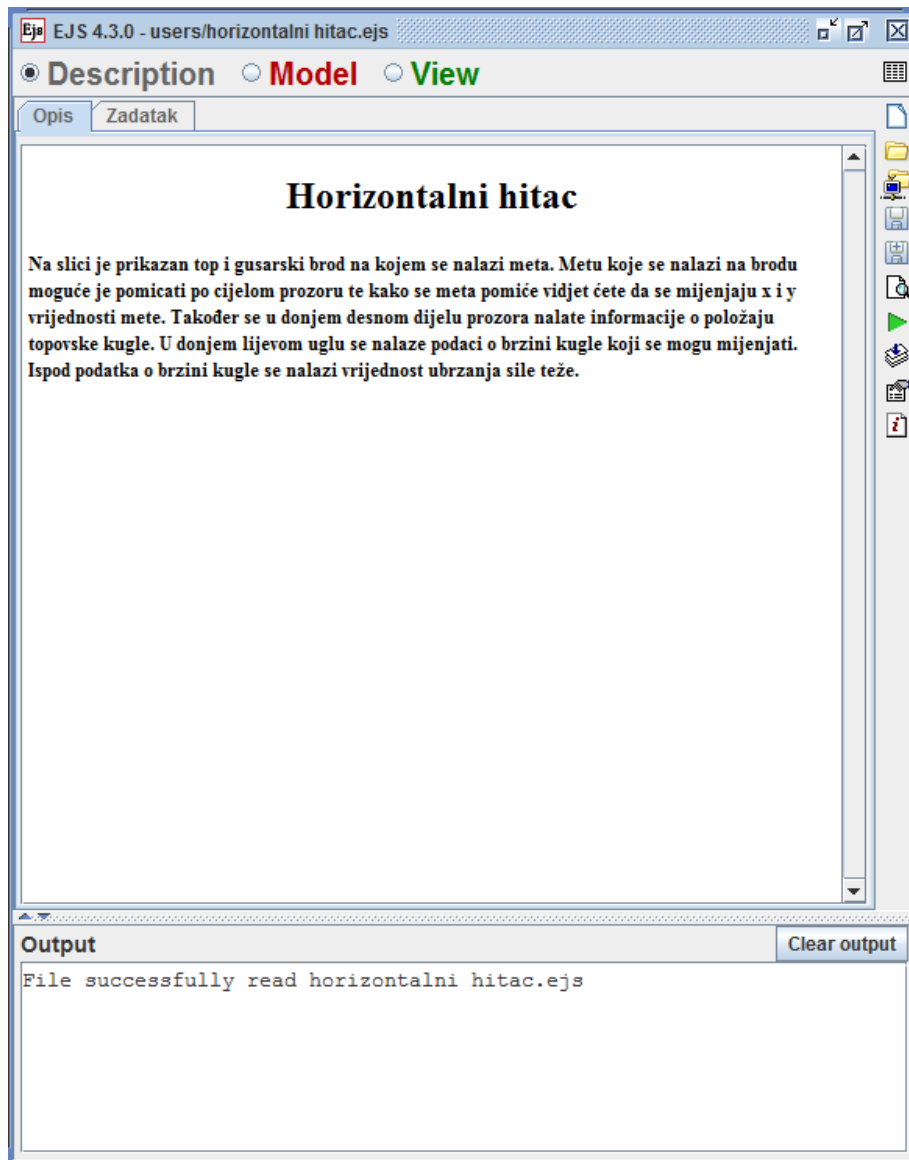
Sada EJS učitava taj dokument te se otvara novi EJS prozor prikazan na slici 3.6.



Slika 3.6. EJS prozor

### 3.3.1. Description panel

Klikom na *Description panel* (ili u slobodnom prijevodu: panel za opis modela) prikazuju se dvije stranice opisa ovog modela. Prva stanica je prikazana na slici 3.7.



**Slika 3.7.** Prva stranica sadrži kratku diskusiju modela - u ovom slučaju model horizontalnog hica

Klikom na *Activities* tab prikazuje se druga stranica. Ovaj panel je HTML multimedijски tekst koji daje informacije i instrukcije o simulaciji. HTML znači Hyper Text Markup Language i najčešće je korišten protokol za formatiranje i prikazivanje dokumenata na Web-u. EJS sadrži jednostavni HTML editor koji omogućuje

stvaranje i razvijanje stranica unutar EJS-a. Također, HTML stranice mogu se unositi u EJS.

### 3.3.2. *Model panel*

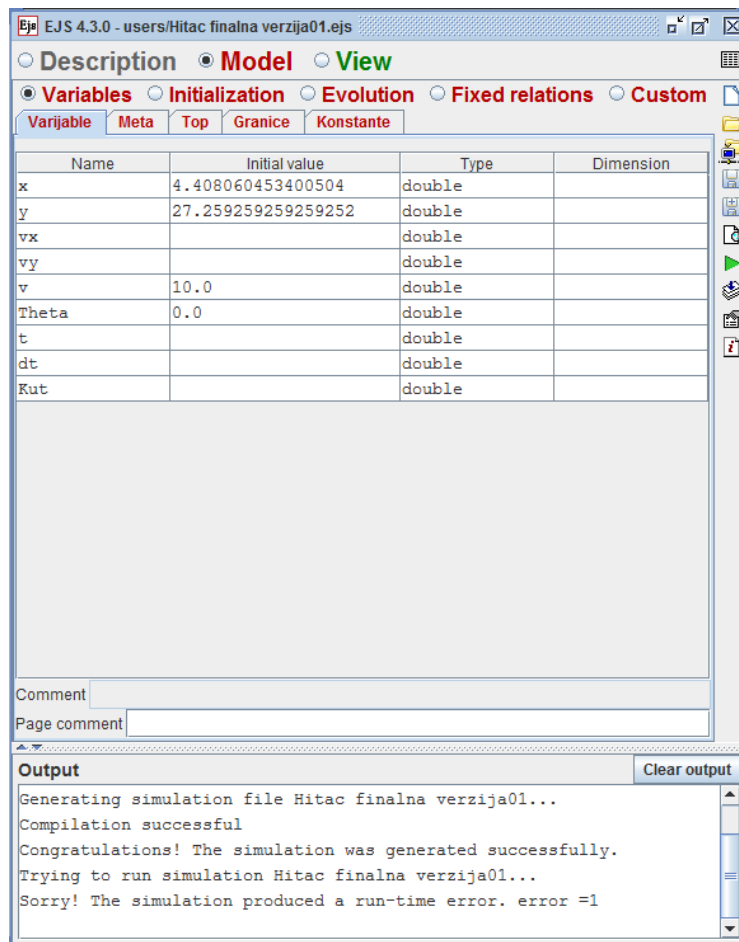
Model se definira u panelu naziva *Model*. U ovoj simulaciji se proučava gibanje kugle mase  $m$  u gravitacijskom polju te znamo da na kuglu djeluje sila  $F=mg$ . Znamo da se kugla giba u  $x$  i u  $y$  smjeru. U  $x$  smjeru kugla ima brzinu  $v_x$  koja je konstantna i ne mijenja se ukoliko zanemarimo otpor zraka, dok u  $-y$  smjeru kugla ubrzava i to akceleracijom  $g$  koja iznosi  $9.81 \text{ m/s}^2$ . U ovom primjeru se koristi koordinatni sustav u  $x$  i  $y$  smjeru, dok ima primjera gdje se koristi samo u jednom smjeru, npr. harmonički oscilator. Kugla je smještena u točki  $(x, y)$  i ima neku početnu brzinu  $v_x$ .

### 3.3.3. *Definicija varijabli*

Kada implementiramo model dobro je kao prvi korak identificirati, definirati i postaviti početne uvjete varijabli koje opisuju sistem za koji želimo napraviti simulaciju. Termin varijabla u ovom slučaju je vrlo općenit i može se odnositi na bilo što čemu se može dati ime, uključujući i fizičke konstante i grafove. Slika 3.8. pokazuje tablicu varijabli u EJS-u. Svaki red definira varijablu modela s imenom, tipom, dimenzijom i početnom vrijednošću varijable.

U računalnim programima postoji nekoliko tipova varijabli koji ovise o podacima koje ta varijabla opisuje. Tipovi koji se najčešće koriste su *boolean* za *true/false* vrijednosti, *int* za cijele brojeve, *double* za visoko precizne brojeve i *string* za text. Horizontalni hitac koristi samo tipove varijabli *double* i *boolean*.

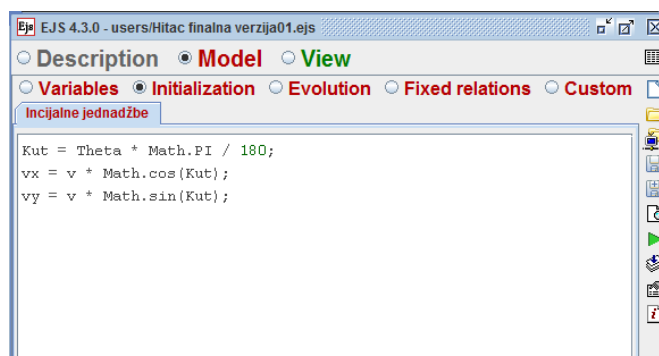
Varijable se mogu koristiti kao parametri, varijable stanja ili kao ulazne i izlazne informacije modela. Na slici 3.8. se vide definirane varijable za vrijeme  $t$ , za poziciju  $x$  i brzinu  $v_x$ ,... Također se u nekim drugim slučajevima mogu definirati varijable koje se ne pojavljuju u jednadžbi, kao npr. kinetička, potencijalna ili ukupna energija. Donji dio panela s varijablama sadrži komentare koji daju opis uloga varijabli.



Slika 3.8. Panel za modeliranje sadrži 5 potpanela; na slici je prikazan panel za definiranje varijabli

### 3.3.3.1. Incijalizacija modela

Važno je točno postaviti početne uvjete kada implementiramo model pošto model mora početi u fizički mogućem stanju. Ovaj model je relativno jednostavan pa incijalizaciju modela radimo uvrštavajući vrijednosti (ili u nekim ostalim primjerima jednostavne Java izraze poput izraza za kinetičku energiju  $0.5 \cdot m \cdot vx \cdot vx$ ) u *Initial value* kolonu. U modelu kosog hica potrebno je definirati kut pod kojim se ispaljuje kugla. EJS sučelje kuteve računa u radijanima pa ukoliko želimo mijenjati kuteve u stupnjevima potrebno je napisati algoritam za pretvorbu radijana u stupnjeve (prikazano na slici 3.9). Također je potrebno definirati brzine  $vx$  i  $vy$  u ovisnosti o kutu. EJS koristi ove vrijednosti kada pokreće simulaciju. Neki napredniji modeli mogu zahtijevati kompliciranije algoritme za incijalizaciju.



**slika 3.9.** Potpanel za inicijalizaciju modela. U ovom panelu se postavljaju uvjeti koje program mora učitati prije nego što pokrene simulaciju. Sve matematičke funkcije u EJS-u počinju sa *Math.* (željena funkcija).

### 3.3.3.2. Evolucija modela

Panel naziva *Evolution* omogućuje zapis Java koda koji defnira kako kuglica u gravitacijskom polju evoluira u vremenu i ova opcija se često koristi na modelima koji nisu bazirani na običnim diferencijalnim jednačbama (ODE-ordinary differential equation). Tu je i druga opcija koja nam omogućuje upis ODE bez programiranja. EJS pruža mogućnost da preko editora definiramo diferencijalne jednačbe u obliku koji odgovara matematičkoj notaciji. EJS tada automatski generira točni Java kod.

Sada ćemo vidjeti kako taj editor funkcionira na ovom modelu. Zbog toga što ODE algoritmi rješavaju sisteme diferencijalnih jednačba prvog reda, jednačbe viših redova moraju biti napisane u obliku jednačbi prvog reda. To se može učiniti tako da brzinu definiramo kao neovisnu varijablu koja odgovara svojoj jednačbi:

$$\frac{dx}{dt} = v_x \quad (1)$$

$$\frac{dy}{dt} = v_y \quad (2)$$

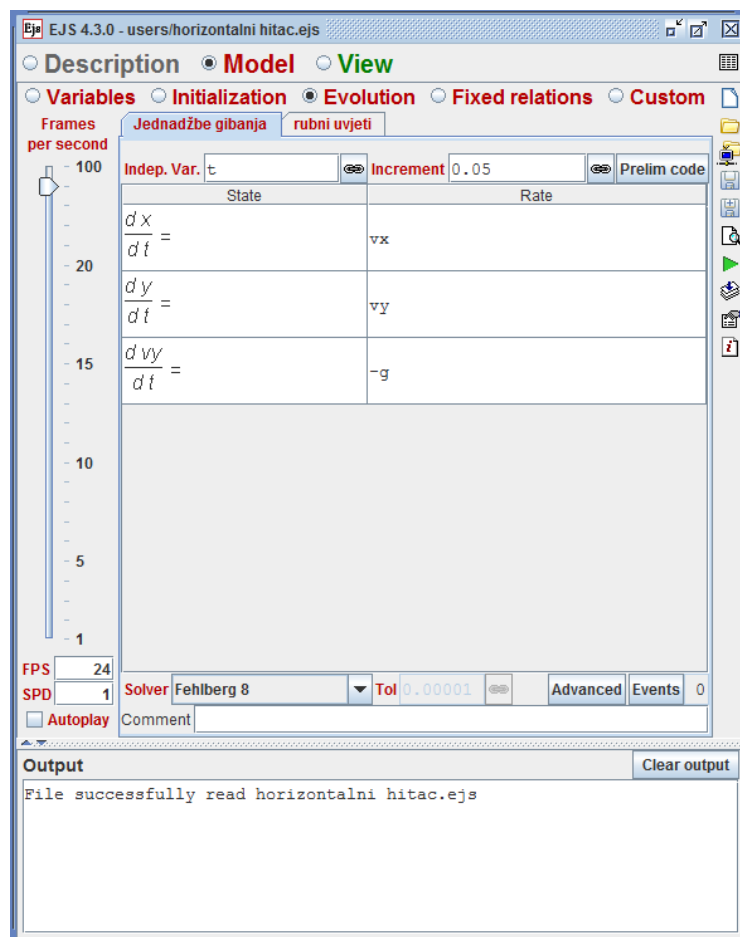
$$\frac{dv_y}{dt} = -g \quad (3)$$



Potreba za dodatnom diferencijalnom jednažbom je razlog zbog kojeg se definira varijabla  $v_x$  u tablici varijabli.

Ako kliknete na *Evolution* panel, on pokazuje ODE editor koji je prikazan na slici 3.10.

Polja oko vrha editora specificiraju neovisnu varijablu  $t$  i povećanje varijable  $dt$ . Gumb *Prelim* u gornjem desnom kutu editora služi za upisivanje preliminarnih kodova koji služe za računanje prije samog pokretanja modela (to je potrebno koristiti u nešto kompleksnijim modelima). Lijeva strana panela sadrži polja pomoću kojih se određuje kako glatko i brzo će simulacija raditi. U većini slučajeva vrijednosti koje su zadane nema potrebe mijenjati.



**Slika 3.10.** Panel *Evolution* prikazuje ODE editor u kojem su prikazane jednažbe (1), (2) i (3) za model horizontalnog hica

### 3.3.3.3. Veze među varijablama

Iako bismo *Evolution* panel mogli nazvati srcem programa, ne računaju se sve varijable modela ovdje. Varijable se mogu računati i poslije evolucije. Koristeći varijable koje su izračunate *evolution* algoritmom, moguće je odrediti tzv. vanjske varijable. U ovom modelu ne postoje vanjske varijable, ali to mogu biti varijable kao što su npr. kinetička, potencijalna ili ukupna energija sistema. Te varijable nazivaju se fiksnim relacijama.

Kao primjer, na slici 3.11. prikazan je *fixed relations* panel za model kosog hica. Taj panel se koristi za upis veza među varijablama. Da bi upisali te relacije mora se paziti samo na dva detalja: korištenje znaka \* za množenje i na kraju svake relacije upisati ;:



Slika 3.11. Fiksne relacije za model kosog hica

### 3.3.3.4. Custom panel

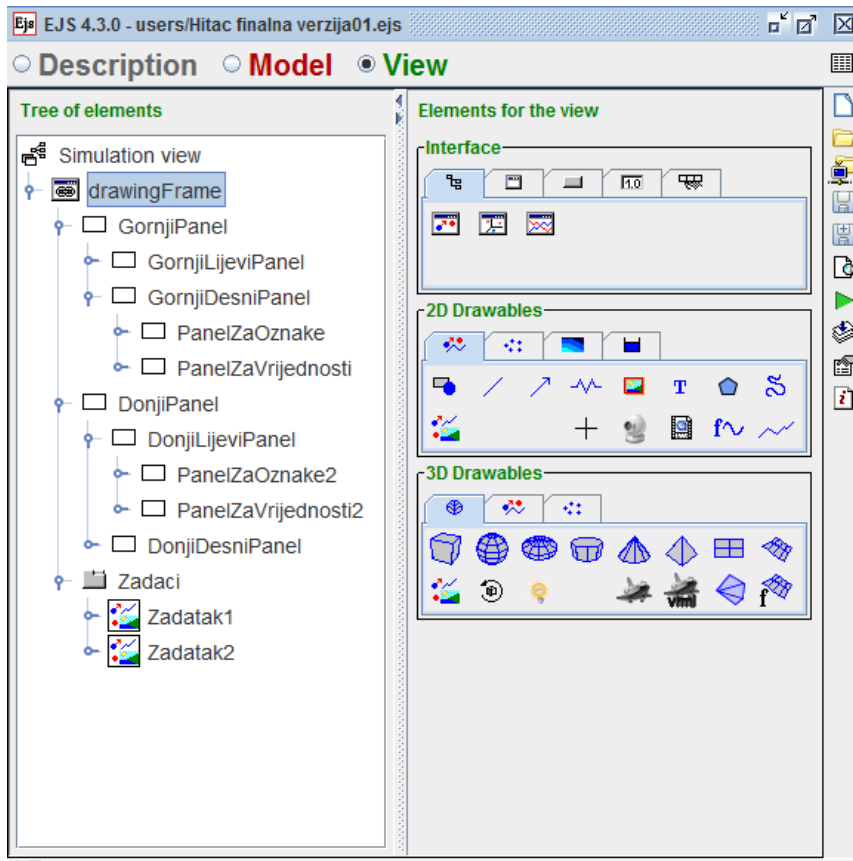
Na kraju panela za modeliranje nalazi se *Custom* podpanel. Ovaj panel se može koristiti za definiranje metoda (funkcija) koje se mogu koristiti kroz model. Ovaj panel je u ovom primjeru prazan jer model ne zahtjeva dodatne metode.

### 3.3.4. View panel

Treći radni panel u EJS-u je panel naziva *View* (izgled). Ovaj panel omogućava stvaranje grafičkog sučelja koje sadrži vizualizaciju same simulacije, interakciju korisnika i kontrolu programa s minimalnim programiranjem. Slika 3.6. prikazuje izgled modela horizontalnog hica.

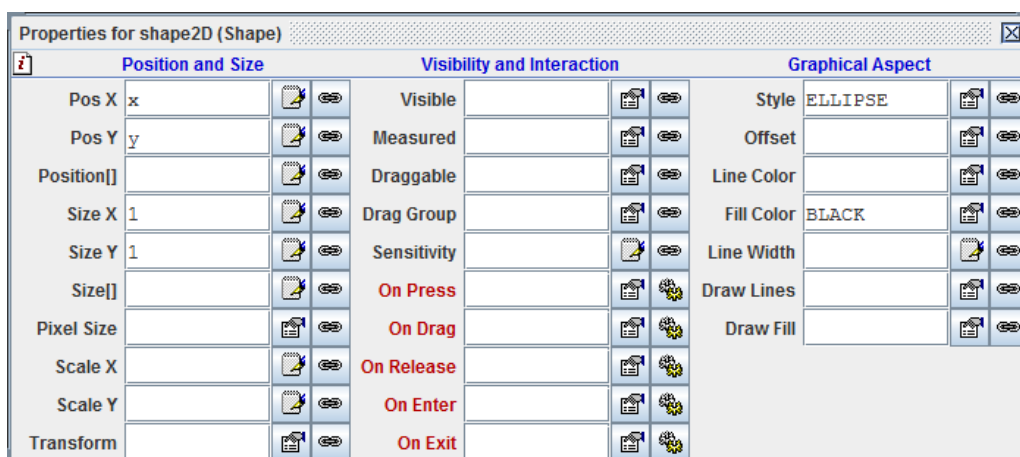
Desni okvir panela prikazan na slici 3.12. sadrži kolekciju elemenata koji nam služe za stvaranje grafičkog sučelja. Svaki od tih elemenata je specijalizirani objekt s *on-screen* prezentacijom. Da bi dobili informacije o pojedinom elementu, potrebno je kliknuti na ikonu i stisnuti *F1* ili desnim klikom odabrati *Help* meni. Pri izradi korisničkog sučelja, radi se okvir i dodaju se elementi (npr. gumbi i grafovi) na „drag and drop“ način.

Lijevi okvir *Tree of elements* (stablo elemenata), također prikazan na slici 3.12., prikazuje strukturu sučelja ovog modela. U ovom primjeru stablo elemenata sadrži prozor, *glavni okvir*, koji sadrži sve elemente od kojih je napravljeno sučelje. Stablo prikazuje njihova imena koja opisuju njihovu namjenu. Desnim klikom na pojedini element otvara se Meni u kojem postoji mogućnost mijenjanja strukture pojedinog elementa. Svaki element ima set vlastitih parametara naziva *properties* koji nam omogućuju uređivanje prikazivanja i ponašanja elementa. Dvostrukim klikom na element u stablu, otvara se tzv. *properties inspector*. Tu su karakteristike, kao npr. boja, obično namještene na konstantnu vrijednost (npr. RED (crvena)). Varijable iz modela se također mogu koristiti za postavljanje osobina elemenata. Mogućnost spajanja varijable i osobina, bez potrebe za programiranjem, ključ je za pretvaranje izgleda u dinamičku i interaktivnu vizualizaciju.



Slika 3.12. Panel View koji nam pokazuje stablo elemenata za model horizontalnog hica

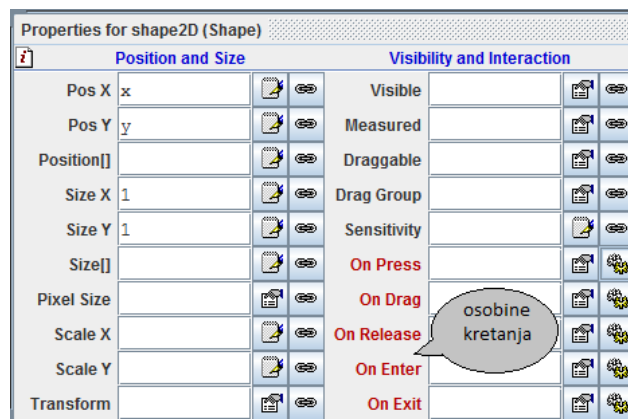
Duplim klikom na *Shape2D* element otvorit će se tzv. *properties inspector* prikazan na slici 3.13.



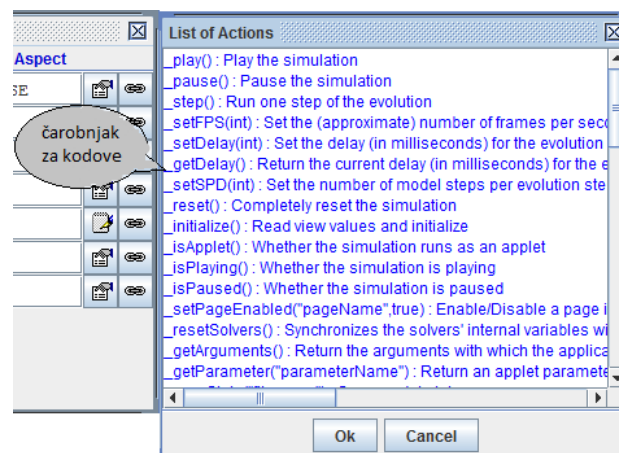
Slika 3.13. Properties inspector

Karakteristike koje imaju zadane konstantne vrijednosti (*Style*, *Size X*, *Size Y* i *Fill Color*) stvaraju elipsu (*Style-Ellipse*) veličine (*Size*) jedinice (1,1). To čini taj element kružnicom obojanom crnom bojom (*Fill color - Black*). *Pos X* i *Pos Y* vezani su za zadane varijable *x* i *y* omogućujući vezu između panela *Model* i panela *View*. Mijenjanjem tih varijabli u vremenu, mijenja se pozicija elementa i to za točne vrijednosti *x* i *y*. Ako korisnik povuče objekt na drugo mjesto, varijable *x* i *y* se mijenjaju. Također, moguće je pomicati objekt samo kada je animacija zaustavljena.

Elementi mogu imati i osobine kretanja (označene crvenom bojom) koje se mogu opisati kodovima. Većinu kodova može se upisati preko čarobnjaka za kodove. To je prikazano na slikama 3.14. i 3.15.



Slika 3.14. Osobine kretanja




Slika 3.15. Čarobnjak za kodove

### 3.4. Pokretanje simulacije

Iz ovog primjera vidi se kako je Easy Java Simulation doista moćan alat koji nam dopušta da iskoristimo znanje o nekom modelu. Postoji već mnogo napravljenih modela koji su kompleksniji - kako fizikalno tako i vizualno.

Da bismo mogli pokrenuti simulaciju moramo kliknuti zelenu ikonu *Run*. Kada kliknemo ikonu, varijable se iniciraju i u mogućnosti smo pokrenuti simulaciju. Kada kliknemo na ikonu, ikona mijenja boju u crvenu i to označava da je simulacija spremna za pokretanje. Sve dok ikona svijetli nismo u mogućnosti mijenjati varijable i ostale opcije u EJS-u. Za zaustavljanje simulaciju kliknite desnim gumbom miša na ikonu i stisnite *Kill current simulation*.

### 3.5. Distribucija simulacije

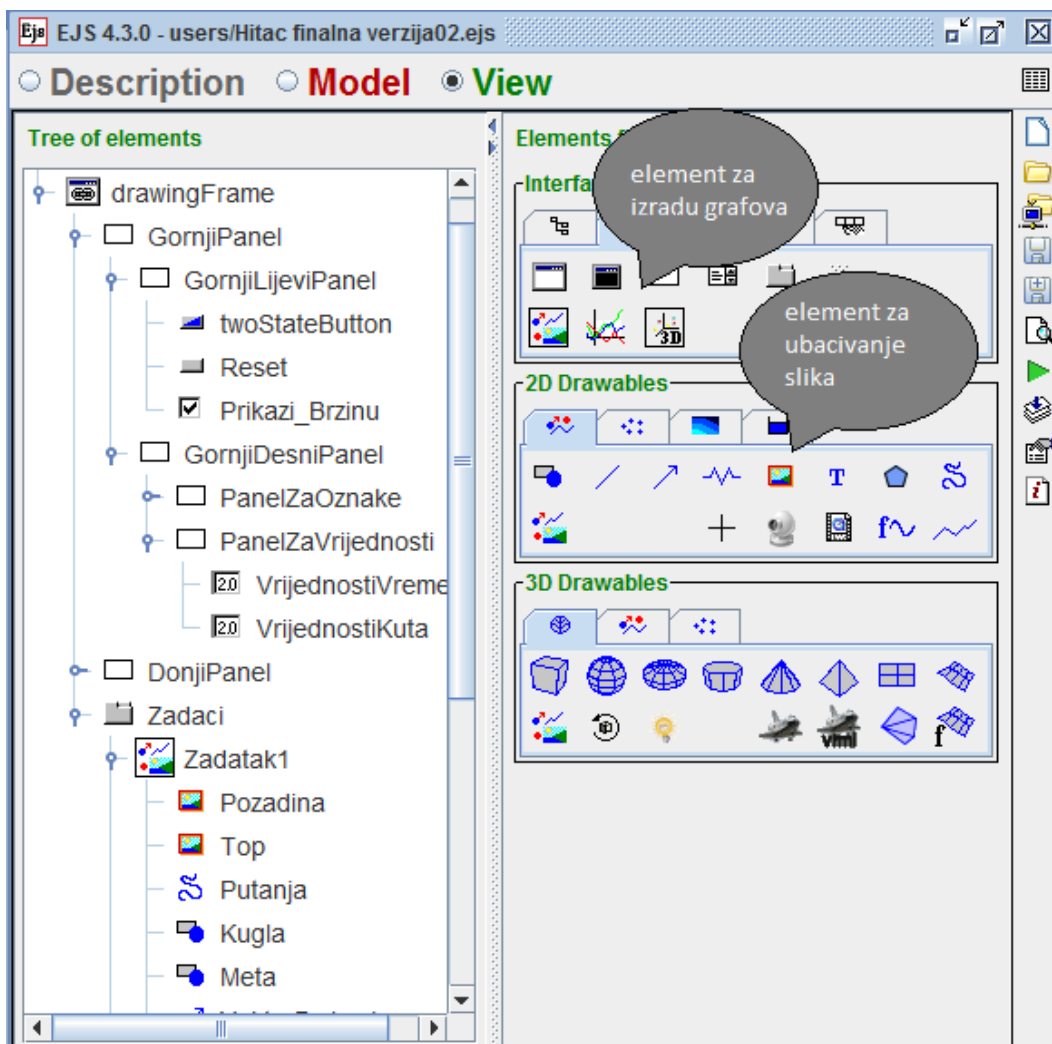
Simulacija napravljena u EJS-u je stand-alone Java program koji mogu koristiti ostali korisnici bez samog EJS-a. Ono što morate napraviti je pakirati simulaciju u jedan tzv. *Jar file* i to tako da kliknete na *Package* ikonu (  ). Tada vam *file browser* omogućava da izaberete ime. Paket se sprema u *export* potdirektorij koji se nalazi u *workspace* direktoriju. Tog trenutka *jar file* je spreman za distribuciju putem interneta, CD-a,...

### 3.6. Ostale mogućnosti EJS-a


U simulaciji horizontalnog hica neće se vidjeti sve mogućnosti ovog programa pa ću stoga nabrojati još neke bitne dijelove u programu koji mogu doprinjeti krajnjem izgledu simulacije.

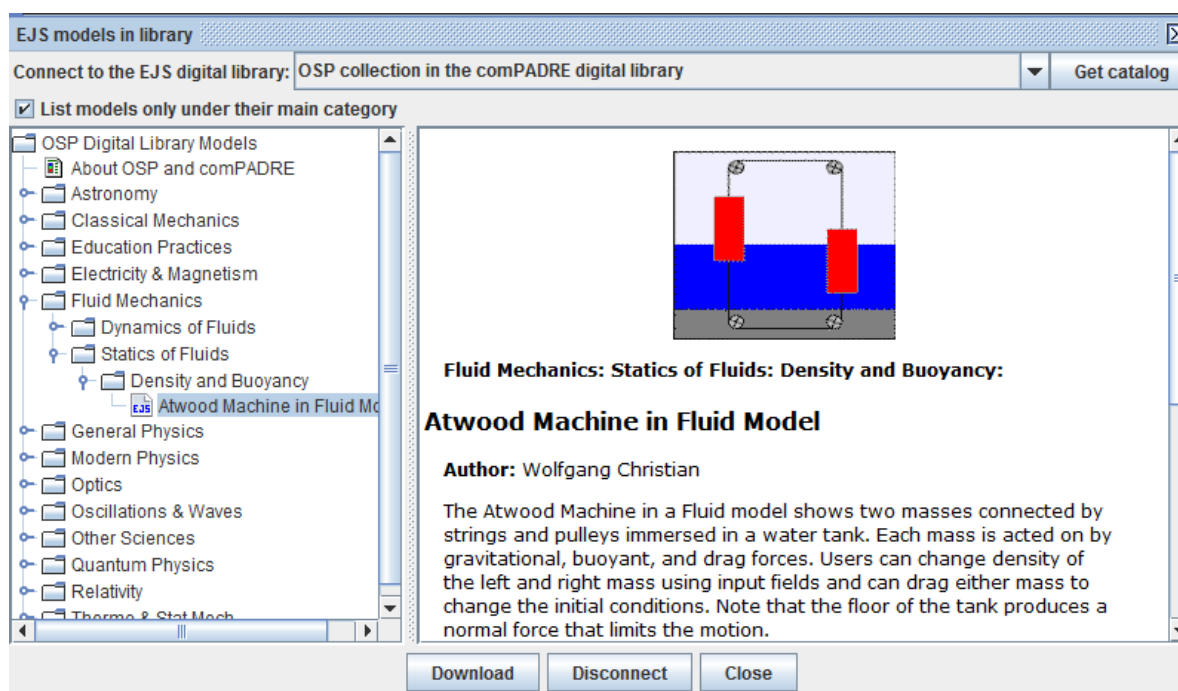
Najbitniji dio programa, odnosno tamo gdje se zapravo stvara vizualni identitet simulacije, je panel *View*. Panel je podijeljen na dva okvira. S lijeve strane je okvir u kojem je prikazano stablo elemenata koje koristimo u našoj simulaciji, a s desne strane je okvir u kojem se nalaze svi mogući elementi koji se mogu koristiti.

*Interface* panel sadrži elemente poput okvira za crtanje, okvira za grafove, okvira za pisanje, raznoraznih gumbića koji čine simulaciju interaktivnom i doprinose stvaranju zanimljive simulacije. Panel koji se nalazi ispod *Interface* panela zove se *2D drawables*. U ovom panelu se nalaze dvodimenzionalni elementi poput strelica (vektori), tragova (prikaz putanje), okvira za ubacivanje vlastitih slika (korišteno na simulaciji horizontalnog hica),... Slijedeći panel zove se *3D drawables*. Kao što mu i samo ime kaže, ovdje se nalaze elementi za stvaranje trodimenzionalne simulacije. Kada se prođe mišem preko elementa, pojavi se prozorčić u kojem piše što pojedini element radi. Svi ti elementi na jednostavan se način ubacuju u stablo elemenata s lijeve strane.



**Slika 3.16.** Prikaz elemenata za izradu grafova i ubacivanje slika

Ukoliko računalo na kojem program radi ima pristup internetu, postoji mogućnost ulaska u tzv. EJS digital library<sup>2</sup>. U toj knjižnici se nalaze modeli korisnika koji žele podijeliti svoje simulacije s ostalim korisnicima. Knjižnica je vrlo korisna stvar jer omogućuje pregled izgleda simulacije „iznutra“, tj. može se vidjeti kako je neki napredniji korisnik riješio probleme vezane uz neki svojevrsni model. Kada se klikne na ikonicu  koja se nalazi na desnoj strani prozora programa, otvara se novi prozor u kojem se nalazi mapa *Models*. U desnom gornjem kutu se nalazi gumb *Get catalog*. Pritiskom na taj gumb prikazuju se modeli razvrstani po granama fizike. Svaku simulaciju unutar pojedine grane moguće je spremiti na računalo. Nakon što je spremljena na računalo moguće ju je otvoriti u EJS-u.



**Slika 3.17.** Digitalna knjižnica iz koje je moguće spremiti gotove modele ostalih korisnika na računalo

<sup>2</sup>u slobodnom prijevodu: digitalna knjižnica



# Simulacija u nastavi

---

Simulacija u nastavi fizike može bit vrlo koristan alat za približavanje nekog problema učenicima. To se posebno odnosi na primjere u kojima je nemoguće praktično pokazati ponašanje modela. Ono što je dobro kod alata poput EJS-a jest da omogućuje izradu simulacija bez potrebe za naprednim znanjem o programiranju te ostavlja mnogo prostora za razmišljanje o samom modelu. EJS se može koristiti u nastavi kao pomoćni alat pri predavanju, tj. kada nismo u mogućnosti izvesti neki pokus može nam koristiti za simulaciju pokusa. Međutim, treba biti vrlo oprezan pri odabiru simulacija koje će se prikazivati. Simulacija, kao što i sama riječ kaže, simulira ponašanje nekog sistema, što znači da neke sisteme, ako je moguće, treba prikazivati u živo.

Simulacija se može koristiti u nastavi fizike u okviru praktikuma gdje bi učenici sami izrađivali simulacije ili kao učenički projekt koji bi učenici izradili kod kuće. Izrađene simulacije učenici bi mogli prezentirati ostatku razreda i opisati problem modela koji su simulirali. Prilikom izrade simulacija u EJS-u učenici imaju mogućnost vježbe analitičkog načina razmišljanja te upoznavanja osnova programiranja. Kroz izradu simulacije učenici bi se trebali upoznati s načinom funkcioniranja programa te detaljnom analizom problema prije upotrebe EJS-a. Kroz jednostavnije primjere učenici bi učili kako program funkcionira, a zatim bi dobili zadatak izrade vlastite simulacije.

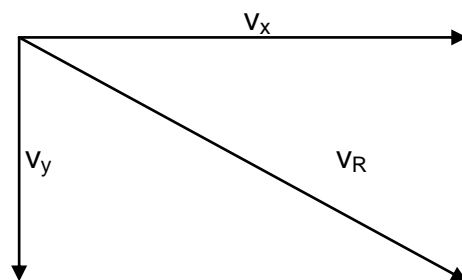
Simulacija horizontalnog hica bi bila korisna za nastavnu jedinicu horizontalni hitac u prvom razredu gimnazije. Pri uvođenju horizontalnog hica, učenici najviše problema imaju s razumijevanjem nezavisnosti gibanja. Najveći problem kod usvajanja nezavisnosti gibanja je taj što je učenicima teško zamisliti da se neko tijelo može neovisno gibati u dva različita smjera. Stoga je vrlo bitno usvojiti taj koncept. Prije uvođenja horizontalnog hica nužno je da su učenici dobro upoznati sa slobodnim padom (tj. gibanjem tijela pod djelovanjem sile teže), jednolikim pravocrtnim gibanjem i vektorskim zbrojem. Također, bilo bi dobro izraditi simulaciju slobodnog pada i paralelno prikazivati simulaciju horizontalnog hica i slobodnog pada te uspoređivati

vremena padanja. Ukoliko se radi samo o prikazivanju već izrađene simulacije, tada bi učenici dobili zadatak koji dolazi sa simulacijom.

### *Složeno gibanje*

Da bi objasnili složeno gibanje potrebno je postaviti nekoliko pitanja kao npr. Kakvo je gibanje topovske kugle u y smjeru? Da li na kuglu djeluje sila u tom smjeru? Koja sila djeluje na kuglu? Da li u x smjeru na kuglu djeluje sila? Kako se kugla giba prema desno? Čemu je jednak put za tijela koja se gibaju jednoliko? Čemu je jednak put za tijela koja se gibaju jednoliko ubrzano?

Nezavisnost gibanja bi se nakon toga mogla grafički objasniti kao zbroj vektora brzine u x smjeru i vektora brzine u y smjeru. Smjer kretanja kugle je određen rezultantnom brzinom.



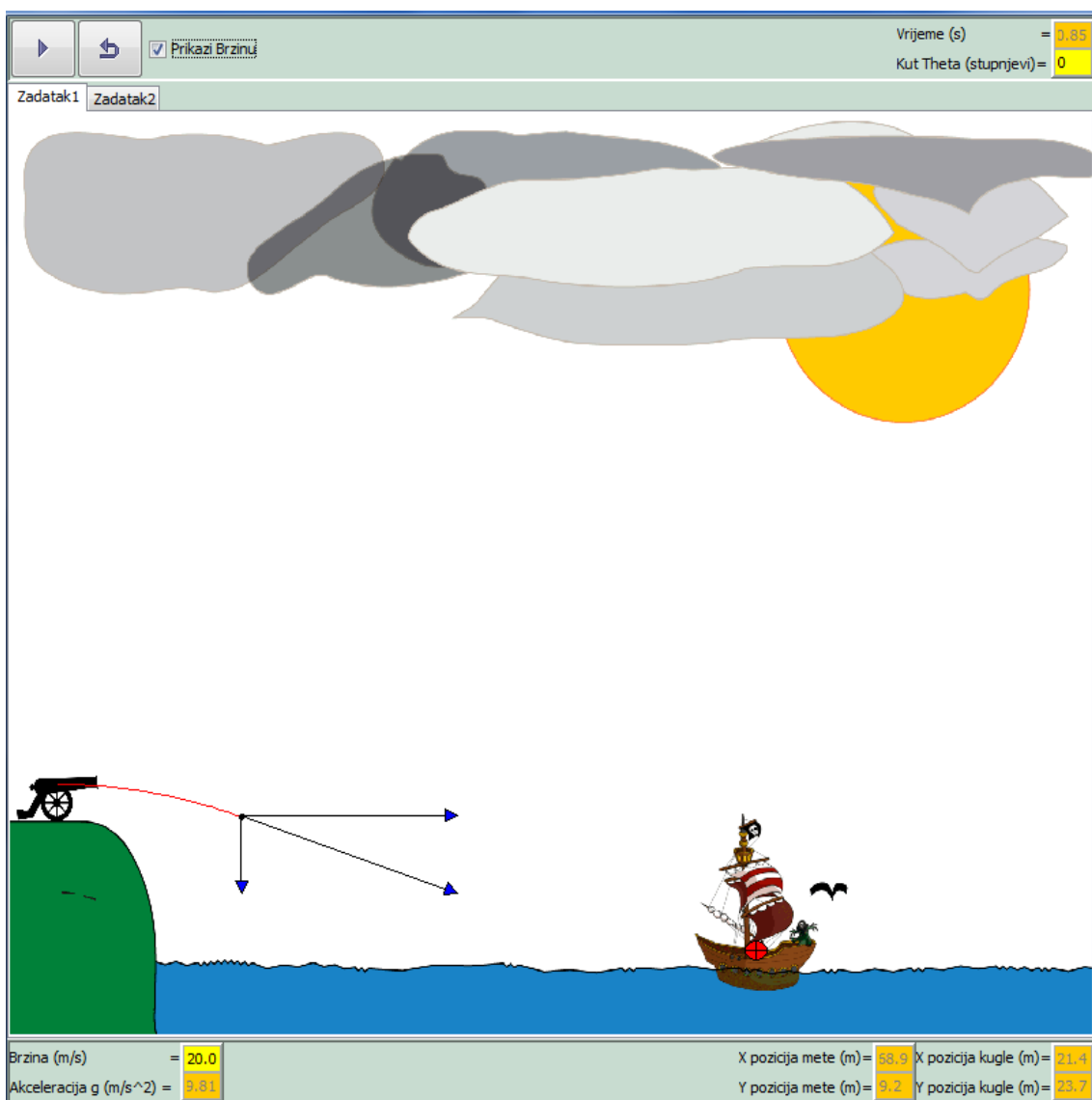
U izrađenoj simulaciji učenicima je zadatak izračunati brzinu koja je potrebna topovskoj kugli da bi uspješno pogodila gusarski brod. U donjem dijelu ekrana su prikazane sve informacije koje su potrebne da bi učenici mogli izračunati brzinu (koordinate mete i kugle, ubrzanje sile teže). Meta je pokretna tako da učenici sami mogu namjestiti metu na željeni položaj. Nakon što izračunaju brzinu, učenici isprobavaju svoje izračune primjeniti na simulaciji.

Da bi učenici izračunali kojom brzinom treba ispaliti topovsku kuglu, moraju znati da domet horizontalnog hica ovisi o početnoj brzini i vremenu slobodnog pada s visine na kojoj se nalazi kugla:

$$D = v_x t \quad \rightarrow \quad v_x = \frac{D}{t}$$

$$h = \frac{1}{2} g t^2 \quad \rightarrow \quad t = \sqrt{\frac{2h}{g}}$$

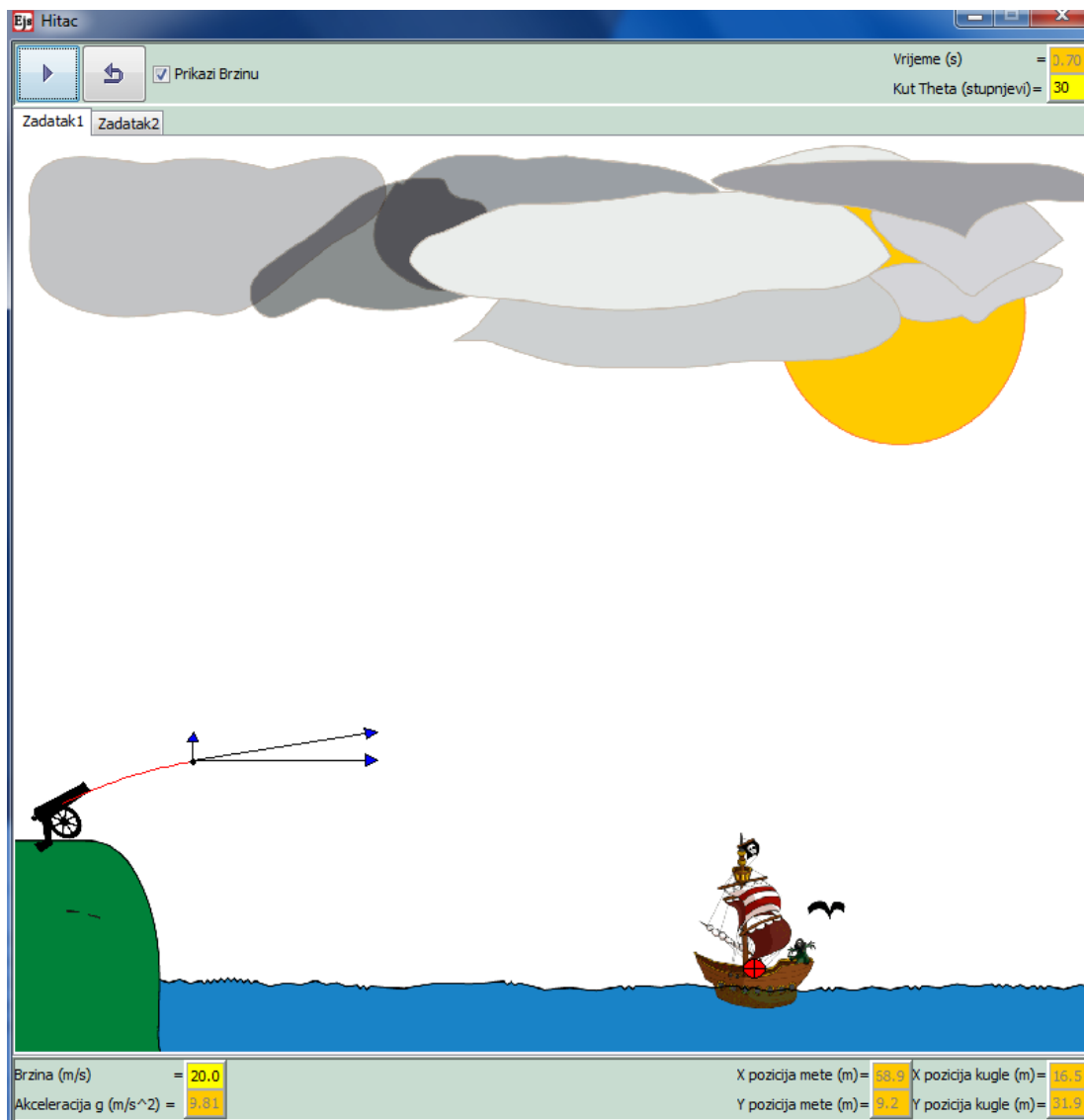
Domet horizontalnog hica će izračunati kao:  $D = (X \text{ pozicija mete} - X \text{ pozicija kugle})$ , što u konkretnom slučaju iznosi 37,5 m. Vrijeme slobodnog pada će izračunati preko visine topovske kugle, odnosno:  $h = (Y \text{ pozicija kugle} - Y \text{ pozicija mete})$ , što u konkretnom slučaju iznosi 14,5 m.



**Slika 4.1.** Prikaz horizontalnog hica

EJS simulacije se mogu koristiti u većini područja fizike koja zahtijevaju zorni prikaz problema. Kao na primjer simulacija horizontalnog hica u električnom polju. Na taj način se mogu povezivati dva ista gibanja s različitim dijelovima fizike i učenici mogu vidjeti kako se gibaju tijela koja imaju masu u gravitacijskom polju, a kako tijela koja imaju naboj u električnom polju.

Smatram da je u školi bolje, za sva područja fizike gdje je to moguće, izvoditi pokuse jer pokus ipak uživo prikazuje fizikalni problem, a puno je lakše doživjeti nešto uživo nego na ekranu računala.



Slika 4.2. Prikaz kosog hica

Drugi zadatak je prikazati kosi hitac. U ovom zadatku cilj je pomoću kuteva prikazati da su horizontalni i vertikalni hitac zapravo granični slučajevi kosog hica. Bitno je da učenici uoče da domet kosog hica ovisi o kutu pod kojim se ispaljuje kugla i brzini.

## 5. Zaključak

---

U ovom diplomskom radu opisan je način funkcioniranja programa Easy Java Simulation, alata za izradu interaktivnih simulacija. Cilj diplomskog rada je upoznavanje s mogućnostima programa preko rada na simulaciji horizontalnog hica. Pri izradi vlastite simulacije koristio sam se već gotovim simulacijama i to tako da sam s tim simulacijama uspoređivao svoju simulaciju. Mislim da je to najbolji način za učenje rukovanja ovim programom. Ono što olakšava čitav taj proces je to što već postoji jako puno izrađenih simulacija. Neke od njih su dobro napravljene, ali neke su nedovršene, što otvara mogućnost za njihovo dovršavanje.

Kada sam krenuo raditi ovaj diplomski rad nisam ništa znao o ovom programu i mislio sam da će biti teško s obzirom da ne znam baš mnogo o programiranju. No, nakon što sam prvi put otvorio program iznenadio me svojom jednostavnošću.

Pri izradi simulacije vrlo je bitno dobro analizirati model koji će se simulirati jer to olakšava posao izrade. Ja sam imao problema s definicijama varijabli jer u ovom programu ima jako puno varijabli koje je potrebno definirati i treba vremena da neki model profunkcionira na željeni način. Najviše problema sam imao s definiranjem kuteva za kosi hitac. Program definira kuteve u radijanima i morao sam naći naredbu za pretvaranje radijana u stupnjeve.

Rad u EJS-u ne zahtjeva programerske vještine te omogućava znanstvenicima da većinu vremena posvete znanosti i problemima vezanim uz modele. Problem kod ovog programa je to što je za neke kompliciranije simulacije nužno poznavati skriptu, tj. napisati dobre naredbe, jer u protivnom nije moguće napraviti dobre modele.

## 6. Literatura

---

[1] <http://www.modelbenders.com/encyclopedia/encyclopedia.html>

[2] <http://www.um.es/fem/EjsWiki/>

[3] Wolfgang Christian, Francisco Esquembre, Modeling Science: From Free Fall to Chaos

[4] <http://en.wikipedia.org/wiki/Simulation>

[5] Rudolf Krsnik, Fizika 1

[6] Young & Freedman, University Physics