

Kreiranje GUI

Miloš Mojović, v.prof.

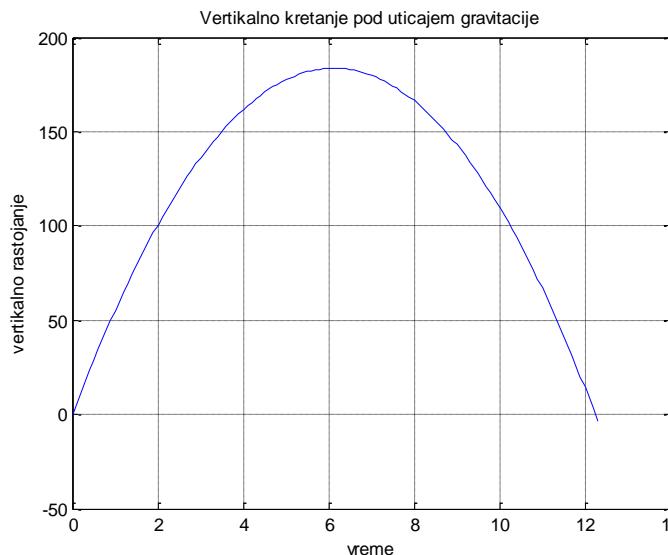
GUI - Graphical User Interface

- U MATLAB-u imamo mogućnost da kreiramo programe bazirane na GUI.
- Takvi programi su podjednako funkcionalni kao i CLI (Command Line Interface) programi, ali su mnogo jednostavniji za korišćenje tzv. "User friendly".
- Pokrećemo ih sa: Home-New-Graphical User Interface-Blank GUI
- Tada na radnu površinu prevlačimo potrebne elemente programa koji se nalaze sa leve strane prozora.
- Hajde da napravimo jedan program koji traži unos promenljivih i pritiskom na "dugme" crta grafik funkcije koja zavisi od njihovih vrednosti.

Grafički prikaz jednačine kretanja

- Ako se malo prisetimo, program koji grafički prikazuje jednačinu kretanja izgledao je ovako:

```
% Vertikalno kretanje pod uticajem gravitacije  
g = 9.8; % gravitaciono ubrzanje  
u = 60; % pocetna brzina (m/s)  
t = 0 : 0.1 : 12.3; % vreme u sekundama  
s = u * t - g / 2 * t.^2; % predjeno rastojanje u metrima  
plot(t, s), title( 'Vertikalno kretanje pod uticajem gravitacije' ), xlabel( 'vreme' ), ylabel( 'vertikalno rastojanje' ), grid
```



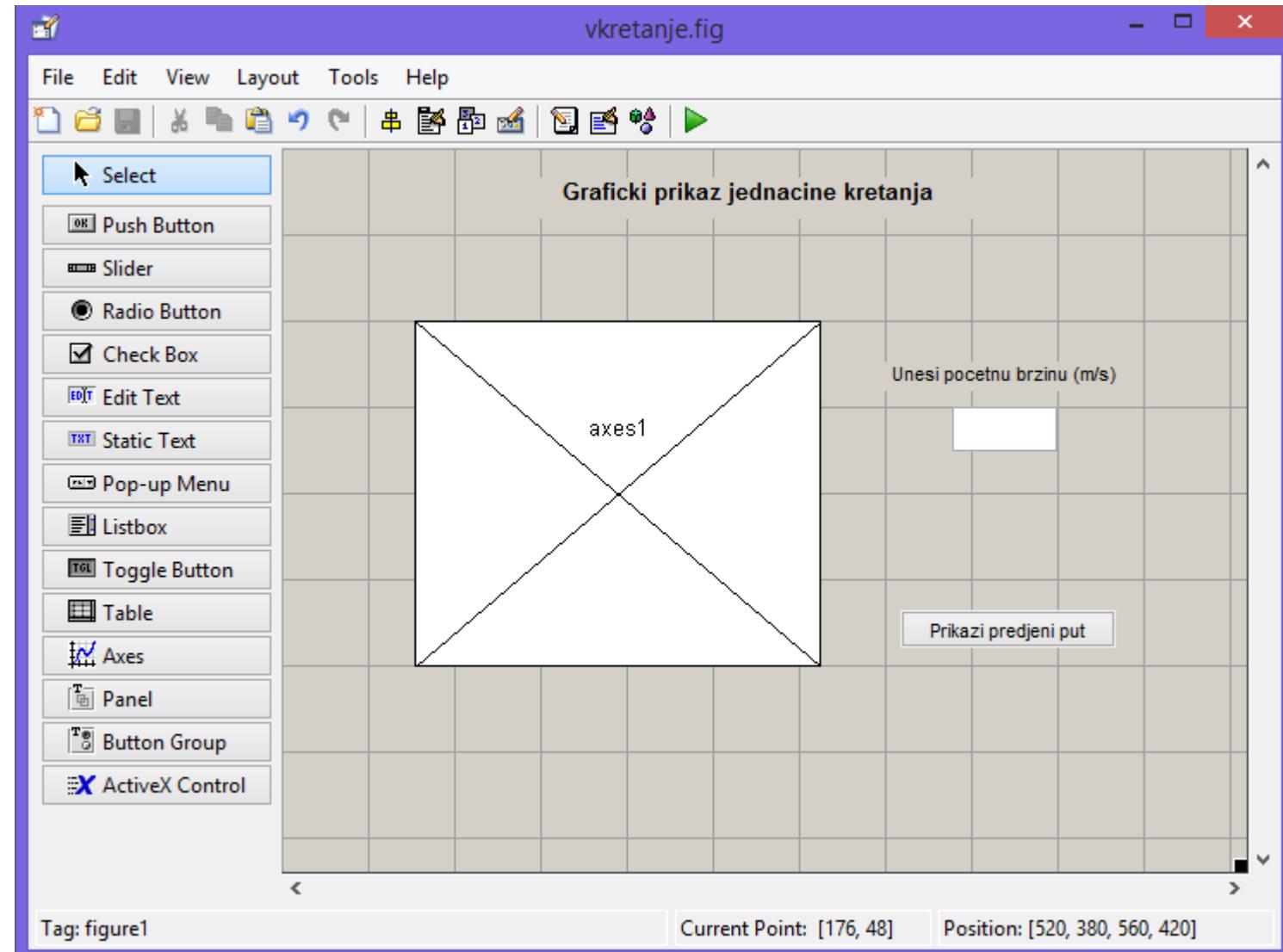
Program:"vkretanje.m

- Ovaj program smo morali da unesemo u "Editor" prozor, a zatim da ga pokrenemo sa "Run".
- Ukoliko bi želeli da promenimo vrednost početne brzine, morali bi da u "Editor" prozoru (u liniji koja se odnosi na vrednost promenljive "u") upišemo novu vrednost.
- Probajmo da napravimo ovaj isti program koristeći GUI.

Grafički prikaz jednačine kretanja - GUI

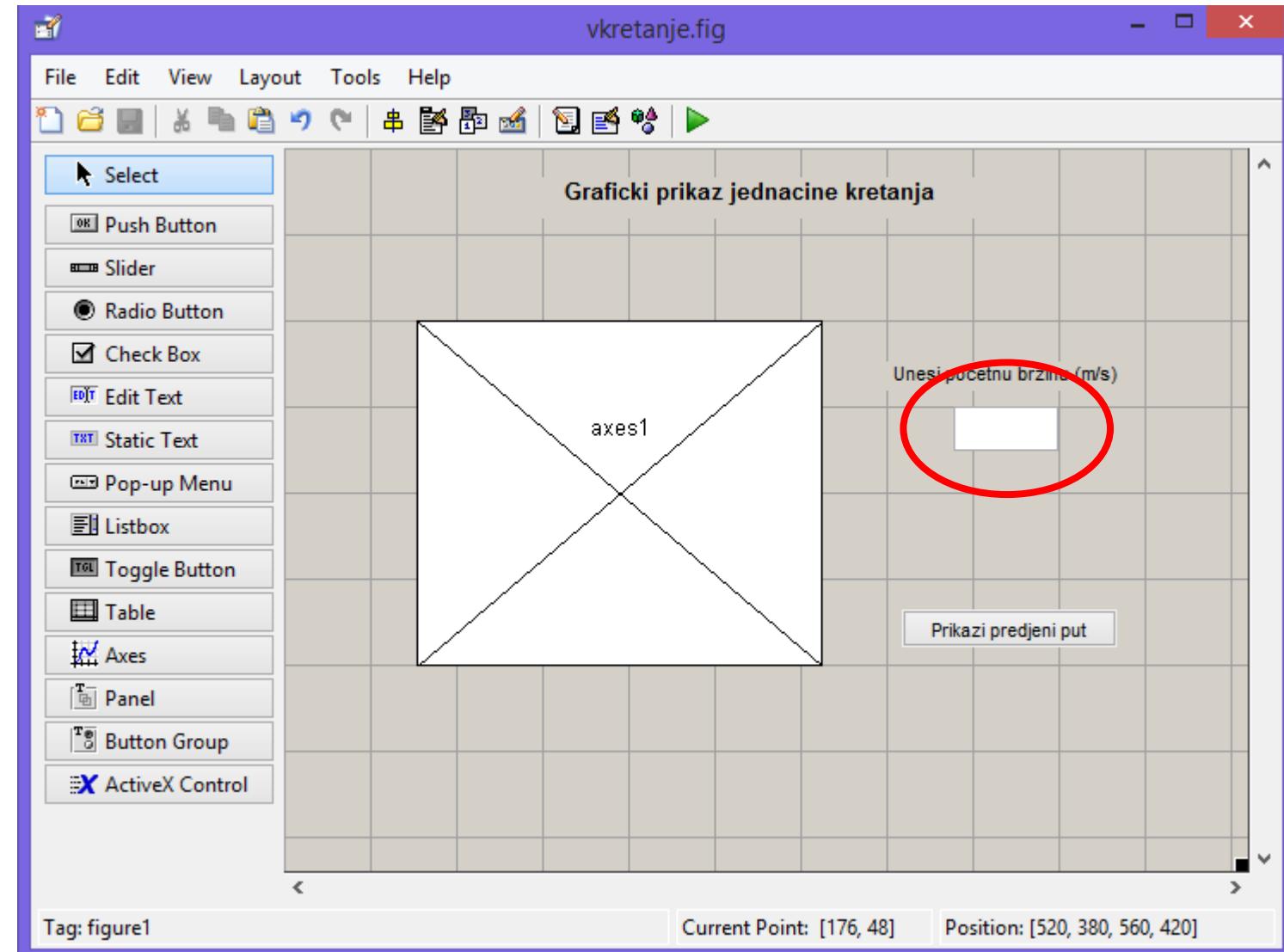
- Napravimo ovakvo grafičko okruženje prevlačenjem elemenata sa leve strane:

- Axes % za prikaz grafika
- Push Button % za izvršenje fje
- Edit text % za unos početne brzine
- Static text (2 kom) % za prateći tekst
- Pokrenimo program pritiskom na dugme "Play".
- Sada MATLAB kreira 2 fajla. Jedan je klasičan .m-fajl u kome se nalazi programski kod, a drugi je .fig-fajl u kome se nalazi GUI.
- Nazovimo ove fajlove "vkretanje.m" i "vkretanje.fig".
- Kreirani GIU trenutno ne radi ništa.
- Zatvorimo kreirane fajlove.



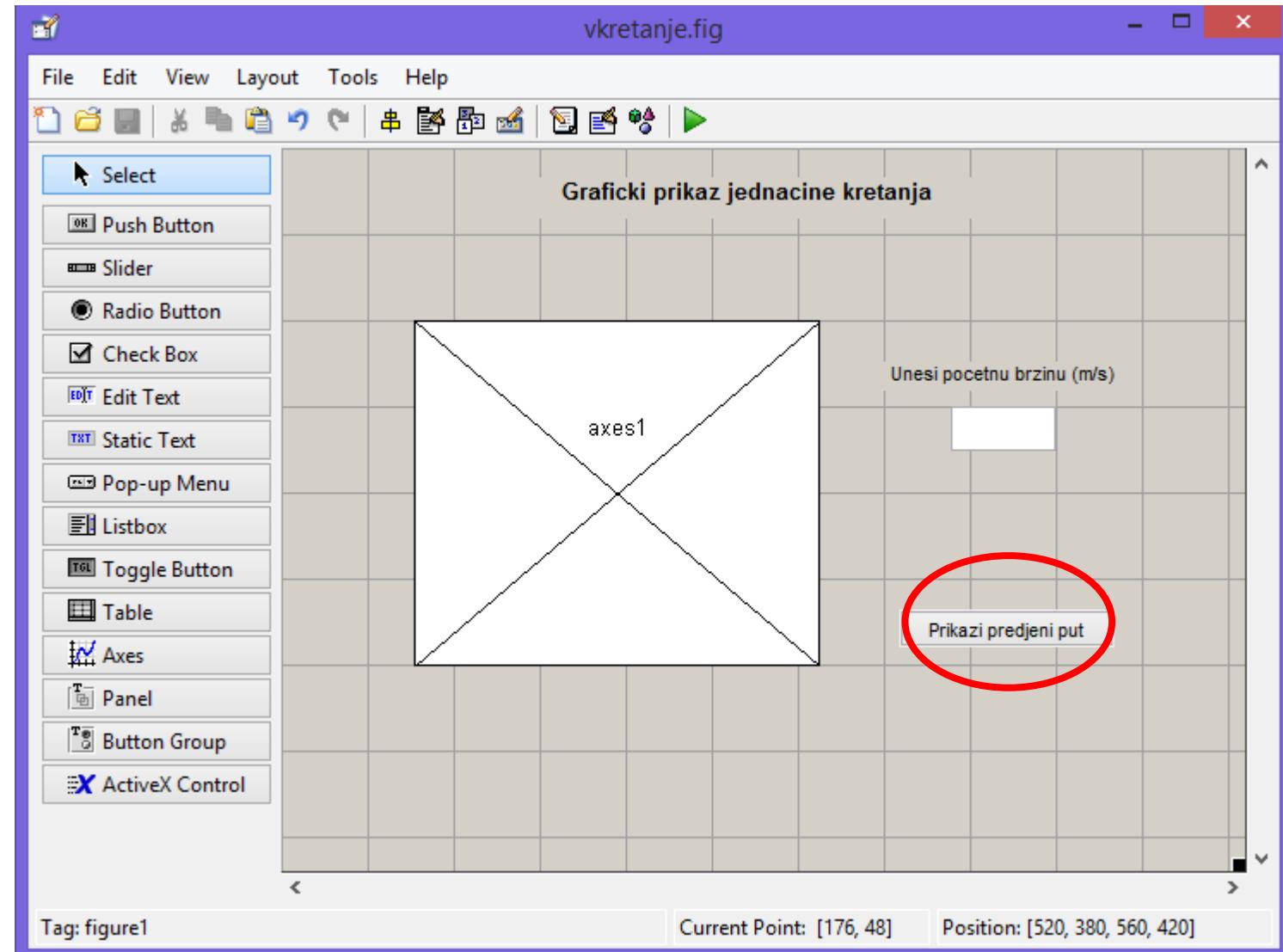
Grafički prikaz jednačine kretanja - GUI

- Ukoliko želimo dalje da editujemo naše fajlove uradićemo sledeće:
 - ❑ Dvoklikom ćemo otvoriti fajl vkretanje.m da bi mogli da editujemo programski kod.
 - ❑ Da bi mogli da editujemo GUI fajl, u "Command" prozoru upišimo komandu:
`>>guide`
 - ❑ Idemo na: Open Existing GUI - Browse - vkretanje.fig
- Sada možemo početi sa editovanjem našeg GUI.
- Prva stvar je da prilagodimo "Tag" za "Edit Text" polje u koje ćemo unositi početnu brzinu.
- Tag je u stvari naziv promenljive koja dobija onu vrednost koja se upisuje u to polje.
- ❑ Dvoklik na polje - Tag - upisati "edit1".
- ❑ Izbrisati "String" da bi polje bilo prazno.



Grafički prikaz jednačine kretanja - GUI

- Zatim treba editovati naše izvršno polje tj. "Push Button".
- Međutim, u GUI prozoru možemo samo promeniti naziv za "Tag" ovog dugmeta, kao i njegov naziv:
 - Dvoklik na dugme - Tag - upisati "s".
 - Kao "String" upisati: "Prikazi predjeni put".
- Da bi ovom dugmetu pripisali neku funkciju moramo nastavati editovanje u .m-fajlu.



Grafički prikaz jednačine kretanja - GUI

- U okviru ovog .m-fajla, MATLAB je već sam kreirao kod koji sada treba malo dopuniti. Najpre moramo naći deo koda koji se odnosi na izvršenje programa koji se pokreće pritiskanjem na dugme (čiji smo Tag nazvali "s").

```
% --- Executes on button press in s.
```

```
function s_Callback(hObject, eventdata, handles)
```

```
% hObject handle to s (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
g = 9.8; % gravitaciono ubrzanje
```

Upisujemo vrednost
promenjive "g"

```
poc = get(handles.edit1, 'String'); % uzimanje pocetne brzine iz prozora
```

```
poc = str2double (poc); % konverzija u doubles matricu
```

```
t = 0 : 0.1 : 12.3; % vreme u sekundama
```

```
s = poc * t - g / 2 * t.^2; % predjeno rastojanje u metrima
```

```
plot(t, s), xlabel( 'Vreme (s)' ), ylabel( 'Vertikalno rastojanje (m)' ), grid
```

Osnovni kod

Kreiramo promenljivu
"poc" koja uzima vrednost
Taga "edit1", i prevodi ga
u doubles matricu

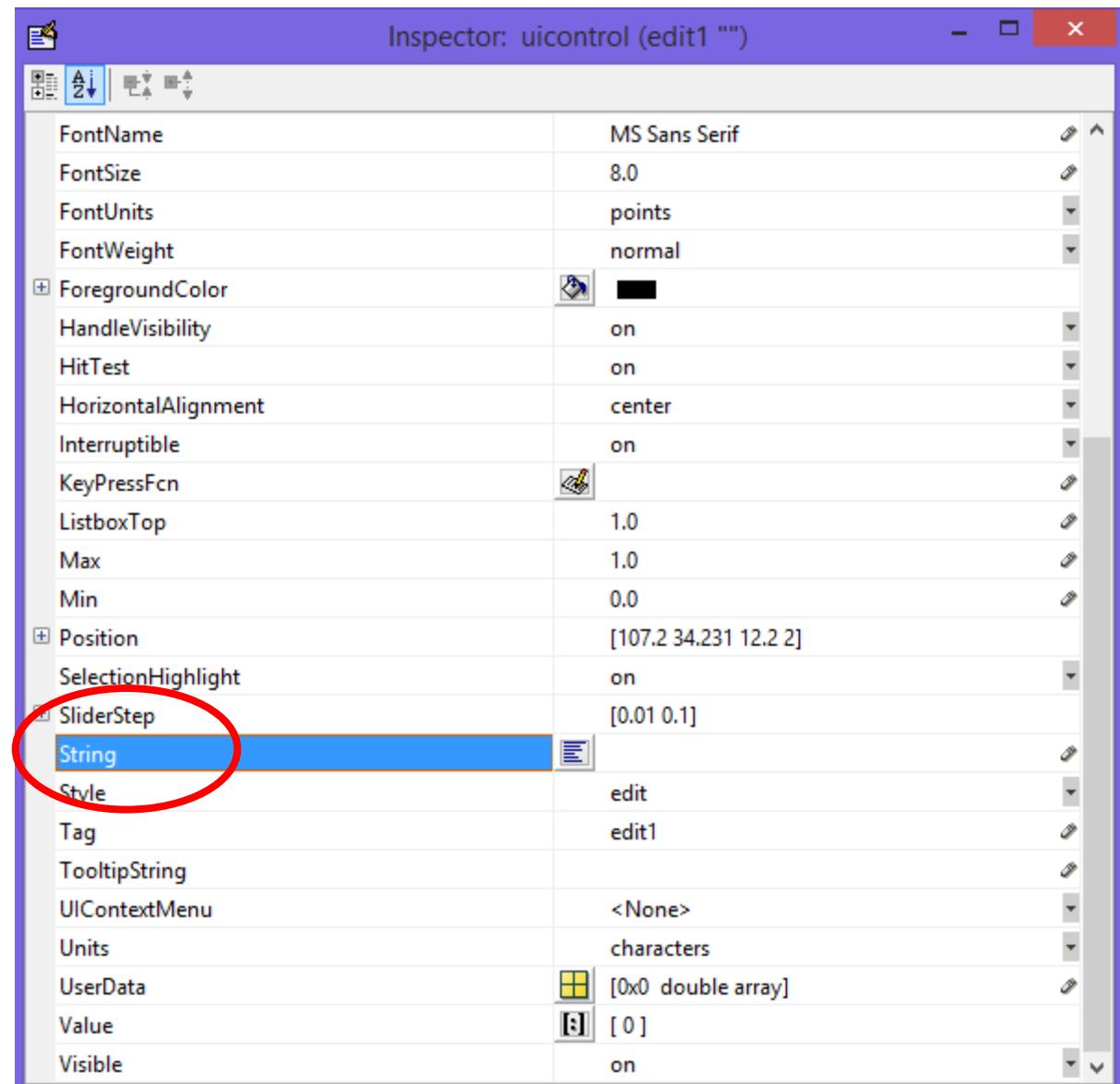
Kreiramo matricu za
vreme "t"

Zadajemo jednačinu za
računanje "s"

Crtamo grafik, sa
oznakama i mrežom

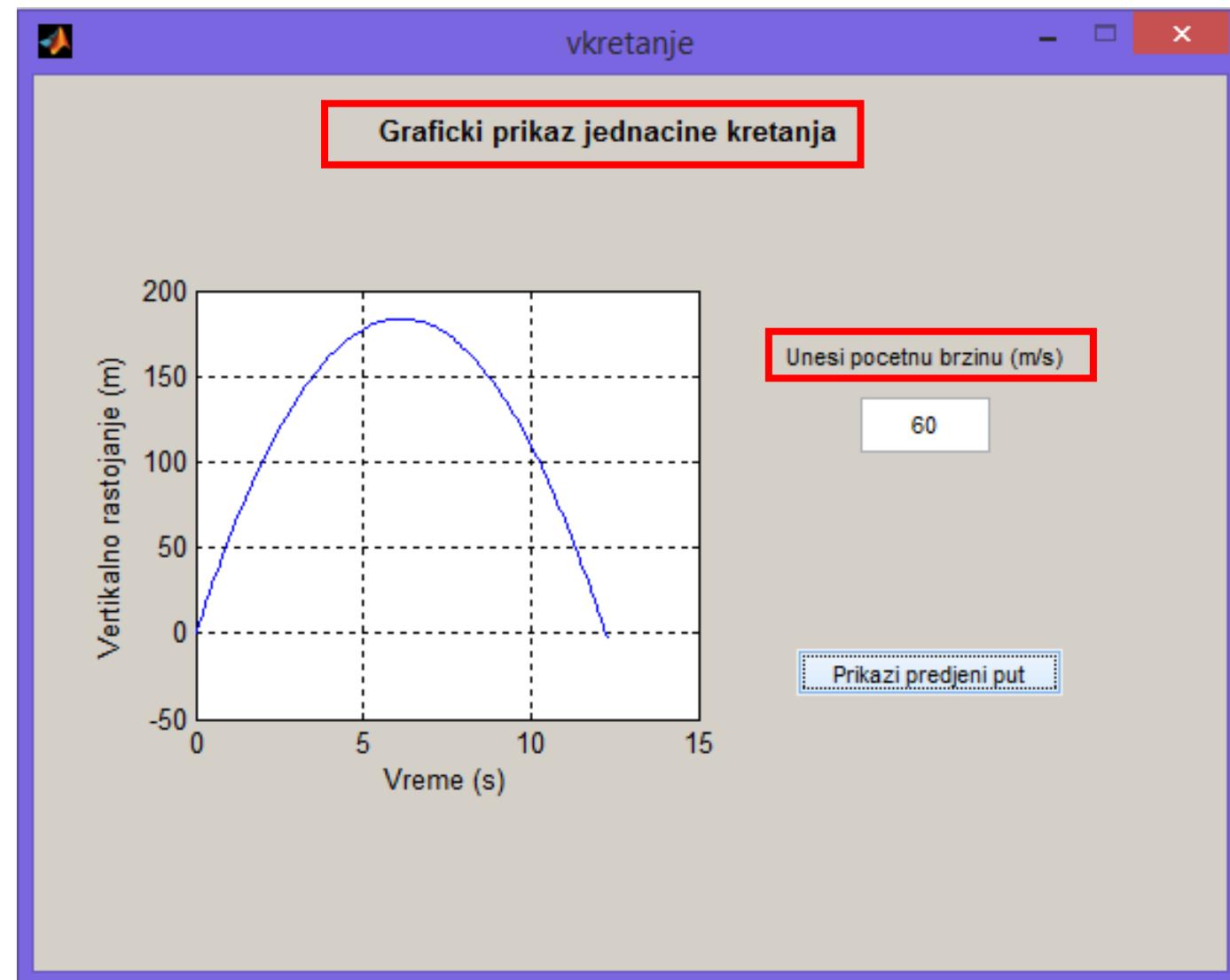
Function handles

- Uočimo sintaksu koja se odnosi na određeni objekat.
- Sintaksa: `get(handles.edit1, 'String')`.
- Najpre imamo komandu "get", zatim u zagradi komandu "handles", zatim "." pa naziv Taga tog objekta "edit1" i na kraju pod apostrofima osobinu tog objekta (npr. `'String'`).
- Ova osobina se takođe može videti kao deo menija koji dobijemo ako dvokliknemo na taj objekat.
- Objašnjenje novih komandi:
 - Komanda "get" uzima trenutnu vrednost osobine objekta (String)
 - Komanda "num2str" prevodi vrednost objekta u broj
 - Komanda "set" postavlja novu brojevnu vrednost objekta (String) iz edit1 prozora, prema nekoj novoj vrednosti



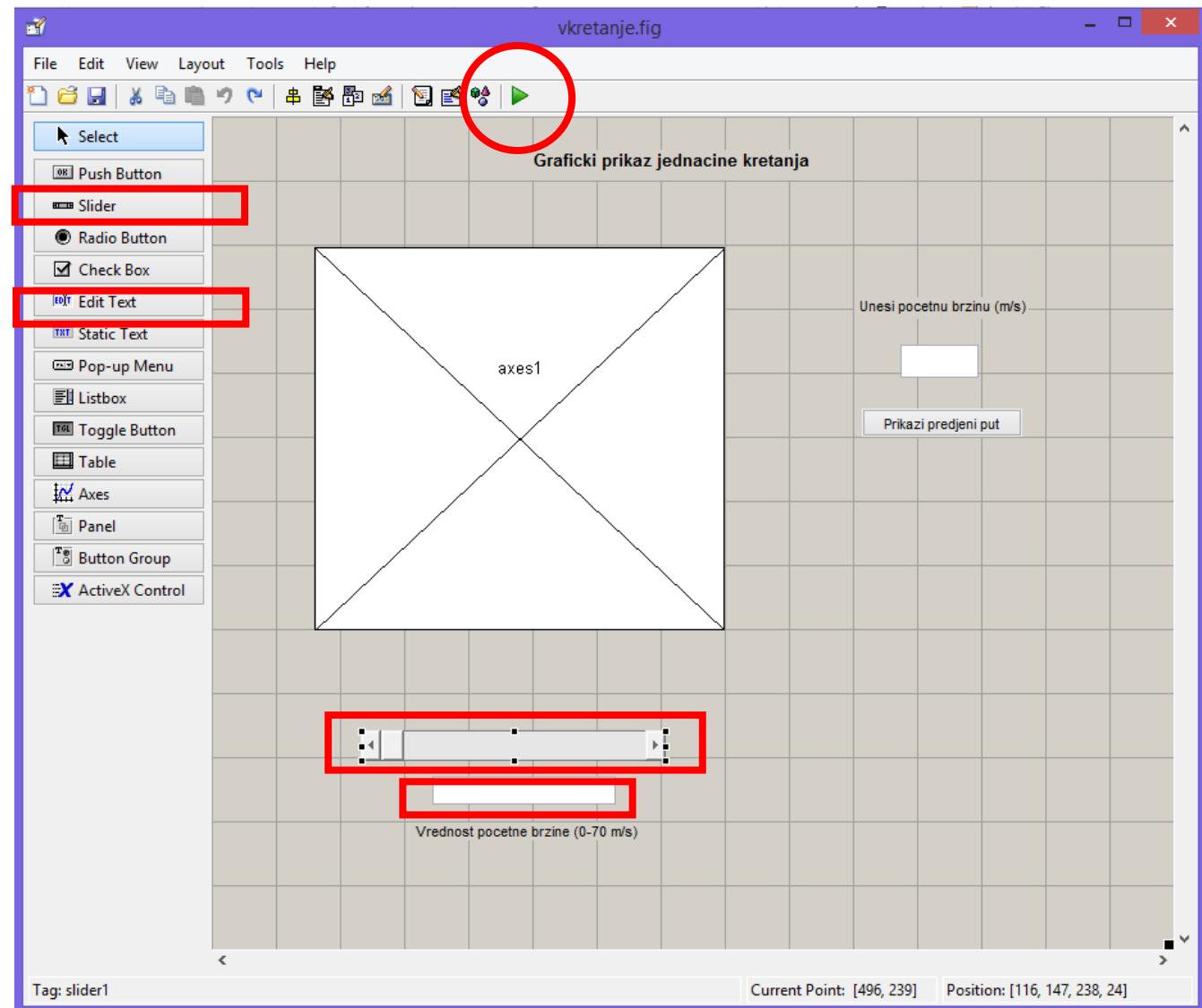
Grafički prikaz jednačine kretanja - GUI

- Konačno, program može da funkcioniše.
- U "edit1" polje možemo uneti željenu vrednost početne brzine, i ta vrednost će biti uzeta u obzir prilikom računanja pređenog puta "s".
- Automatski, crta se grafik, sa adekvatno skaliranim osama x i y (što je osnovna postavka MATLAB programa).
- Da bismo malo ulepšali program, prevucimo još 2 "Static text" panela koje ćemo preimenovati u naslov grafika i uputstvo "Unesite početnu brzinu m/s".
- Možemo i promeniti boju pozadine:
 - Dvoklik na pozadinu - Color
- Iako ovaj program lepo funkcioniše, pokušajmo da ga obogatimo slajderom (klizačem).
- Tada bi bili u mogućnosti da brzo menjamo vrednosti početne brzine bez ukucavanja u edit1 prozor.



Grafički prikaz jednačine kretanja - GUI

- Slajder unosimo u program prevlačenjem "Slider" panela na radnu površinu.
- Tagujmo naš slajder nazivom "slider1" (dvoklik na slajder - Tag -slider1).
- Pomoću miša namestimo da slajder bude odgovarajuće visine i širine (na ovaj način možemo izabrati da li ćemo imati vertikalni ili horizontalni slajder).
- Da bismo uvek znali koju vrednost promenljive zauzima slajder, prevućemo ispod njega još jedan "Edit Text" panel koji ćemo tagovati sa nazivom "edit6".
- Sada, kada imamo sve GUI elemente, kliknimo na "Play" dugme da bi se novi elementi implementirali u .m-fajl.
- Sada možemo pristupiti pisanju koda koji će omogućiti funkcionisanje slajdera "slider1" i novog "edit6" panela.
- Pređumo u .m-fajl (vkretanje.m). Možemo uočiti da se .m-fajl povećao za nekoliko novih programske redova koji se odnose na funkcije "slider1" i editor6" elemenata.
- Najpre povežimo "slajder 1" sa "editor6" tagom, tako da svaka promena jednog elementa izaziva promenu drugog.



Povezivanje elemenata GUI

- Neka nam zadatak bude sledeći:
 - Zadajmo da slajder može uzimati vrednosti od 0 do 70.
 - Napravimo da nam, ukoliko neko u "edit6" displej unese vrednosti koje nisu od 0 do 70, program napiše da unete vrednosti nisu dobre i ...
 - Napravimo da nam program izbroji koliko puta smo napravili grešku prilikom unosa.
- Prvo da ispišemo kod za slider1:
 - Najpre postavimo brojač grašaka na 0.
 - Zatim, povežemo vrednost položaja slajdera sa brojevnom vrednošću na displeju.
 - Unesemo sve elemente za crtanje funkcije, koja će se prikazati odmah nakon bilo kakvog pomerenja slajdera.
 - Minimalnu i maksimalnu brojevnu vrednost koju može da ima "slider1" nameštamo dvoklikom na slajder pa zatim nađemo: "Max" gde unesemo 70 i "Min" gde unesemo 0.
- Pomenimo sada da ćemo ove vrednosti sada moći da pozivamo korišćenjem "handles" funkcije, kao na primer: `get(handles.slider1,'Min')` ili `get(handles.slider1,'Max')`

Osnovni kod

```
% --- Executes on slider movement.  
function slider1_Callback(hObject, eventdata, handles)  
% hObject handle to slider1 (see GCBO)  
% eventdata reserved - to be defined in a future version of MATLAB  
% handles structure with handles and user data (see GUIDATA)  
  
% Hints: get(hObject,'Value') returns position of slider  
% get(hObject,'Min') and get(hObject,'Max') to determine range of slider  
  
handles.number_errors = 0; % Postavlja brojac pogresnih pokusaja na 0  
  
guidata(hObject,handles); % Sacuvava novu handles strukturu u GUI  
% Inace, da bi sacuvali bilo koje promene u strukturi handlesa, mora se  
% koristiti komanda guidata(hObject, handles), a zatim treba ubaciti kod  
% koji implementira tu promenu. Evo ovako ...  
  
set(handles.edit6,'String',num2str(get(handles.slider1,'Value'))); %Ovde se  
% uzima (get) brojevna vrednost (Value) slajdera (slider1), prevodi njegova vrednost  
% (Value) u broj (num2str) koji se dodeljuje (set) broju (String) koji vidimo  
% na displeju(edit6).  
% Sada ide kod koji crta grafik  
g = 9.8; % gravitaciono ubrzanje - konstanta  
vo = get(handles.edit6,'String'); % Uzimanje pocetne brzine iz edit6 prozora  
vo = str2double(vo); % Konverzija u doubles matricu  
t = 0 : 0.1 : 12.3; % Pravimo matricu za vreme u sekundama  
s = vo * t - g / 2 * t.^2; % Racunamo predjeno rastojanje u metrima  
plot(t,s), xlabel( 'Vreme (s)' ), ylabel( 'Vertikalno rastojanje (m)' ), grid % Crtanje grafika
```

Povezivanje elemenata GUI

- Sada prelazimo na prikaz ispisa na "edit6" panelu:
- Pravimo promenljivu (val) koja predstavlja doubles matricu broja koji se nalazi u edit6 kućici.
- Slajder postavljamo na poziciju koja se nalazi u kućici "edit6" i određuje se da li je u edit6 kućici upisan broj izmedju 0 i 70 (i ako ne, prijavljuje grešku i broj neuspešnih pokušaja).
- Postavljamo uslov da li je upisan broj (komandom "isnumeric(val)" koja je 1 ako jeste broj a 0 ako nije i "length(val)" koja daje dimenzije vektora koji je za bilo koji broj =1).
- Proveravamo da li je broj \geq od postavljenog minimuma
- Proveravamo da li je broj \leq od postavljenog maksimuma
- Ako je sve ok, postavlja vrednost slajdera na vrednost "val".
- Ali, u slučaju da ono gore nije ispunjeno aktivira se brojanje broja grešaka preko brojača i prikazuje se vrednost tog broja.

Osnovni kod

```
function edit6_Callback(hObject, eventdata, handles)
% hObject handle to edit6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
% str2double(get(hObject,'String')) returns contents of edit6 as a double

val = str2double(get(handles.edit6,'String')); % Pravi promenljivu (val) koja predstavlja doubles matricu broja (String) koji se nalazi u edit6 kucici

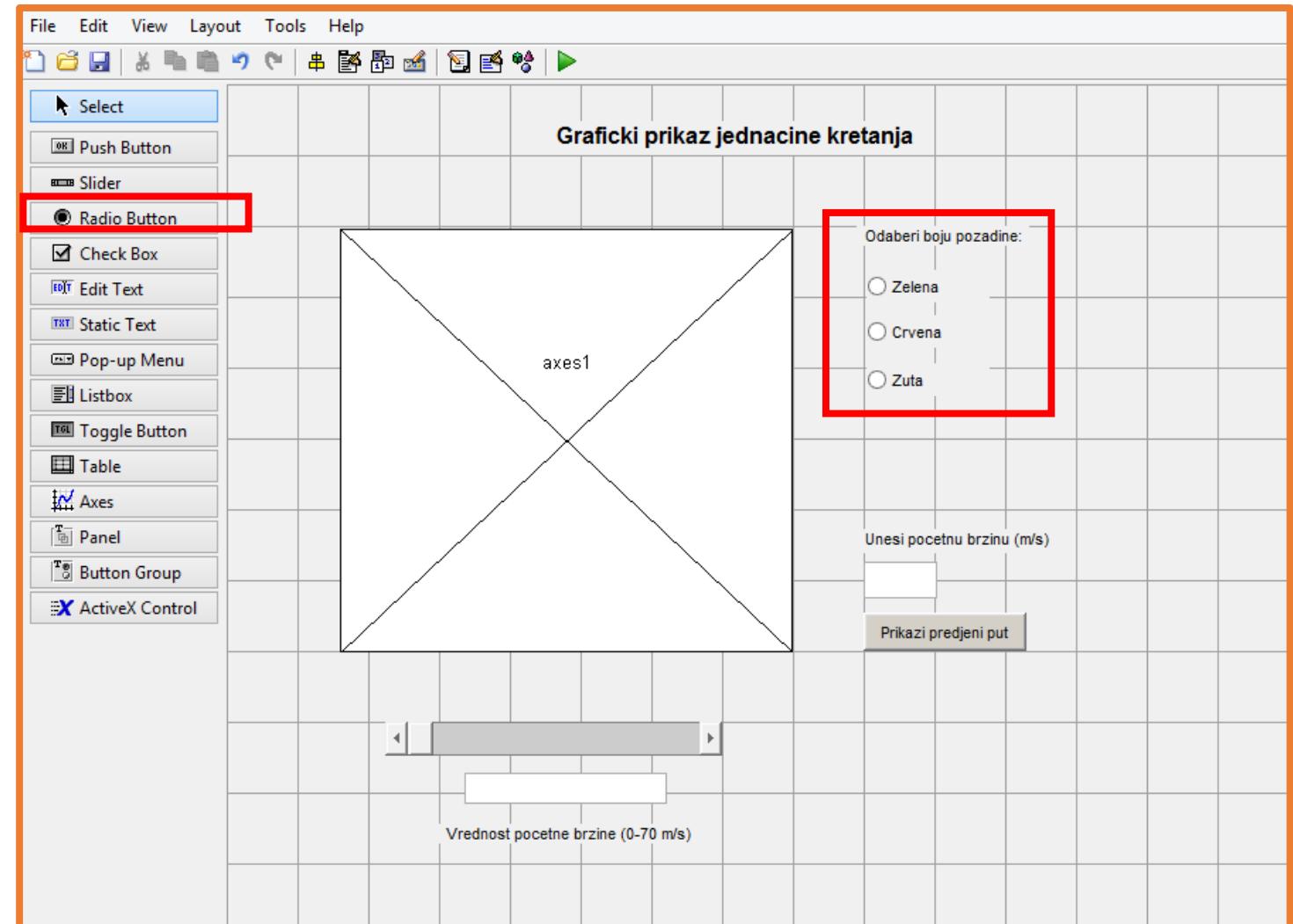
% Sada se slajder postavlja na poziciju koja se nalazi u kućici "edit6" i određuje se da li je u edit6 kucici upisan broj izmedju 0 i 70 (i ako ne, prijavljuje grešku i broj neuspešnih pokušaja)

if isnumeric(val) && length(val)==1 && ... %Početak uslova da li je upisan broj
    val >= get(handles.slider1,'Min') && ... % Proveravamo da li je  $\geq$  od minimuma
    val <= get(handles.slider1,'Max') % Proveravamo da li je  $\leq$  od maksimuma
    set(handles.slider1,'Value',val); % Ako je sve ok, postavlja vrednost slajdera na vrednost "val"
else % U slučaju da ono gore nije ispunjeno

    % Brojanje broja grešaka preko brojača i prikaz tog broja
    handles.number_errors = handles.number_errors+1;
    guidata(hObject,handles); % Čuva promene (bilo ranije)
    set(handles.edit6,'String',... % Ispis broja napravljenih grešaka
        ['Pogresan opseg: ',...
        num2str(handles.number_errors),'. put']);
end % Kraj uslova
```

Dopuna GUI

- Ako želimo da nam program izgleda još lepše, možemo ubaciti i opciju da nam se pomoću "Radio button" elementa menja boja pozadine.
- ❑ To se lako radi prevlačenjem 3 takva elementa i ubacivanjem odgovarajućih uslova.
- ❑ Evo kako izgleda kod koji bi trebali da kreiramo u okviru .m-fajla:



Dopuna GUI

- Ako želimo da nam program izgleda još lepše, možemo ubaciti i opciju da nam se pomoću "Radio button" elementa menja boja pozadine.
- To se lako radi prevlačenjem 3 takva elementa i ubacivanjem odgovarajućih uslova.
- Evo kako izgleda kod koji bi trebali da kreiramo u okviru .m-fajla:
- Napravimo da nam se pozadina menja iz sive u zelenu, crvenu ili žutu boju.
- Najpre moramo napraviti uslov da se boja pozadine radnog panela promeni zajedno sa bojom pozadine "radio-dugmeta" i pratećeg teksta, ukoliko je dugme uključeno.
- Zatim treba napisati kod koji sve vraća na osnovnu (sivu) boju, ukoliko je dugme isključeno.
- Ovaj kod treba napisati za svako (od tri) radio-dugmeta, uz promenu teksta koji se odnosi na boju (green, red, yellow).
- Napravimo još jednu dopunu programa, naime postavimo jedan "popupmeny" koji će u svom listingu imati opciju da nam se prikaže maksimalna visina i vreme leta.

```
% --- Executes on button press in radiobutton1.  
function radiobutton1_Callback(hObject, eventdata, handles)  
% hObject handle to radiobutton1 (see GCBO)  
% eventdata reserved - to be defined in a future version of MATLAB  
% handles structure with handles and user data (see GUIDATA)
```

```
c=get(handles.radiobutton1,'Value');  
if c==1;  
    set(handles.figure1,'Color','green')  
    set(handles.radiobutton1,'BackgroundColor','green')  
    set(handles.radiobutton2,'BackgroundColor','green')  
    set(handles.radiobutton3,'BackgroundColor','green')  
    set(handles.text4,'BackgroundColor','green')  
    set(handles.text5,'BackgroundColor','green')  
    set(handles.text6,'BackgroundColor','green')  
    set(handles.text7,'BackgroundColor','green')  
    set(handles.text8,'BackgroundColor','green')  
else  
    set(handles.figure1,'Color',[0.94 0.94 0.94])  
    set(handles.radiobutton1,'BackgroundColor',[0.94 0.94 0.94])  
    set(handles.radiobutton2,'BackgroundColor',[0.94 0.94 0.94])  
    set(handles.radiobutton3,'BackgroundColor',[0.94 0.94 0.94])  
    set(handles.text4,'BackgroundColor',[0.94 0.94 0.94])  
    set(handles.text5,'BackgroundColor',[0.94 0.94 0.94])  
    set(handles.text6,'BackgroundColor',[0.94 0.94 0.94])  
    set(handles.text7,'BackgroundColor',[0.94 0.94 0.94])  
    set(handles.text8,'BackgroundColor',[0.94 0.94 0.94])  
end
```

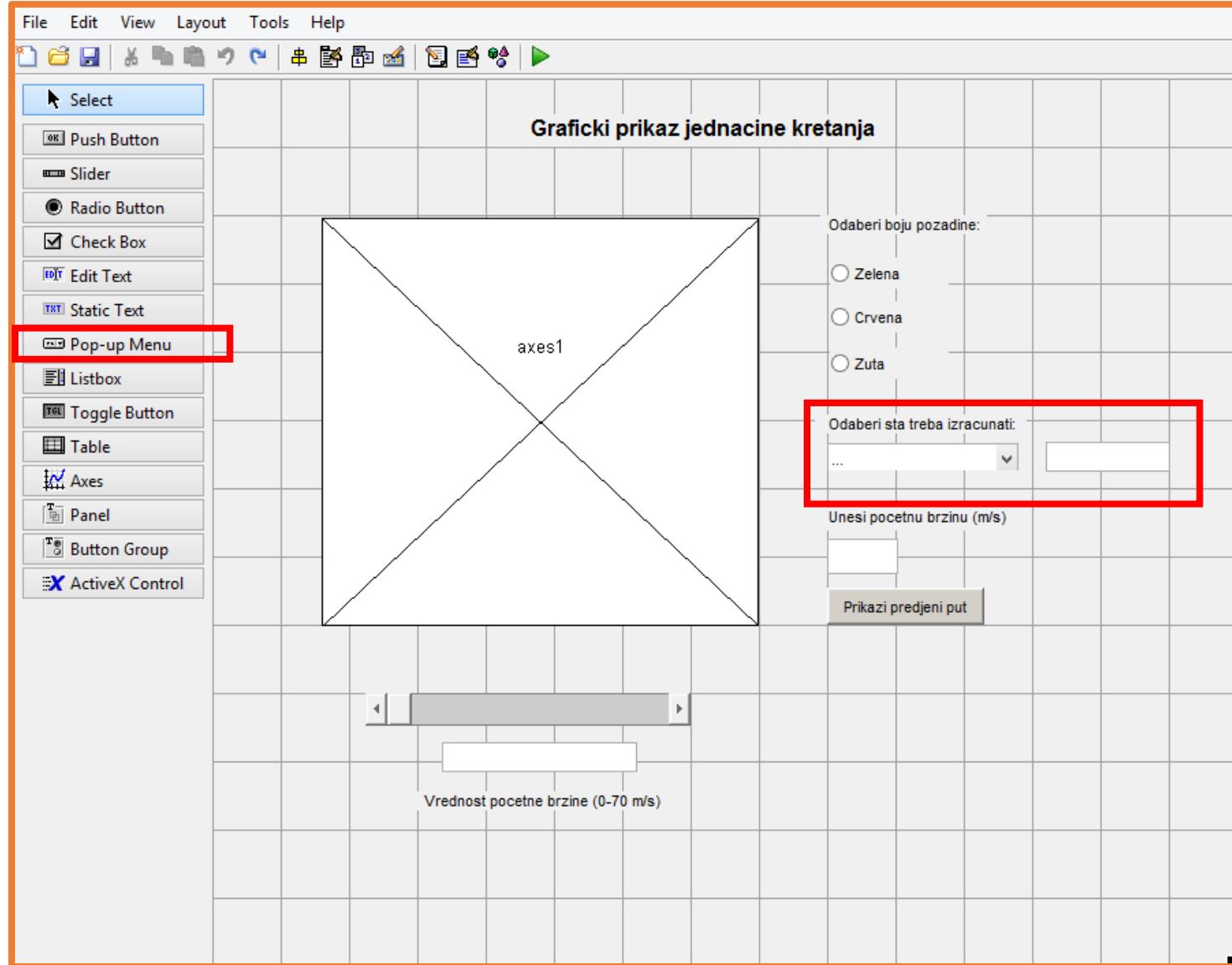
Osnovni kod

Postavljanje uslova zelene pozadine radnog panela "Color" i boje pozadine dugmeta i teksta "BackgroundColor", samo ukoliko je "uključeno" radiobutton1 tj. ukoliko mu je "Value"=1

Postavljanje drugog uslova, tj. vraćanje boje pozadine radnog panela "Color" i boje pozadine dugmeta i teksta "BackgroundColor", na sivu boju, ukoliko je "isključeno" radiobutton1

Dopuna GUI

- Za tu svihu prevlačimo na radni pane element "Pop-up Menu".
- U okviru njega, dvoklikom na "String" upisujemo: "...", "Maksimalna visina" i "Vreme leta".
- Prevućićemo još dva "Edit text" bloka (edit 7 i 8), u kome će se prikazivati izračunati parametri.
- Dopunimo sada .m-fajl kodom koji će se odnositi na:
 - Izračunavanje željenih parametara
 - Povezivanje našeg izbora u meniju sa računom
 - Povezivanje računa sa ispisom u edit 7 i 8 bloku
- Evo kako bi taj kod trebao da izgleda:



Dopuna GUI

- Za tu svhu prevlačimo na radni pane element "Pop-up Menu".
- U okviru njega, dvoklikom na "String" upisujemo: "...", "Maksimalna visina" i "Vreme leta".
- Prevućićemo još dva "Edit text" bloka (edit 7 i 8), u kome će se prikazivati izračunati parametri.
- Dopunimo sada .m-fajl kodom koji će se odnositi na:
 - Izračunavanje željenih parametra
 - Povezivanje našeg izbora u meniju sa računom
 - Povezivanje računa sa ispisom u edit 7 i 8 bloku
- Evo kako bi taj kod trebao da izgleda:
- Kao što se može videti iz komentara u okviru programa, radi potpune funkcionalnosti programa, kod za račun ubacujemo i u okvиру "slider" i u okviru "button" objekta.
- Na ovaj način, račun će biti izvršen na osnovnu vrednosti početne brzine koja se unosi ili pomoću slajdera ili upisom u kućicu i pokretanjem dugmetom.

```
% --- Executes on slider movement.  
% --- Executes on button press in s.
```

Pravljenje novih promenljivih "maksv" i "tleta" prema formulama. Naravno, u ispisu rezultata ne smemo zaboraviti na jedinice :)

```
% Racunanje maksimalne visine i vremena leta  
maksv = (poc.^2/g)-g/2*(poc.^2/g^2); %formula za maksimalnu visinu  
tleta = poc/g; %formula za vreme leta  
set(handles.edit7,'String',[num2str(maksv) ' m']); %ispis maksimalne visine u edit7 prozoru  
set(handles.edit8,'String',[num2str(tleta) ' s']); %ispis vremena leta u edit8 prozoru
```

```
% --- Executes on selection change in popupmenu1.
```

```
function popupmenu1_Callback(hObject, eventdata, handles)  
% hObject handle to popupmenu1 (see GCBO)  
% eventdata reserved - to be defined in a future version of MATLAB  
% handles structure with handles and user data (see GUIDATA)
```

```
% Hints: contents = cellstr(get(hObject,'String')) returns popupmenu1 contents as cell array  
% contents{get(hObject,'Value')} returns selected item from popupmenu1
```

```
switch get(hObject,'Value');  
  
case 1  
    set(handles.edit7,'Visible','off');  
    set(handles.edit8,'Visible','off');  
case 2  
    set(handles.edit7,'Visible','on');  
    set(handles.edit8,'Visible','off');  
case 3  
    set(handles.edit8,'Visible','on');  
    set(handles.edit7,'Visible','off');  
end
```

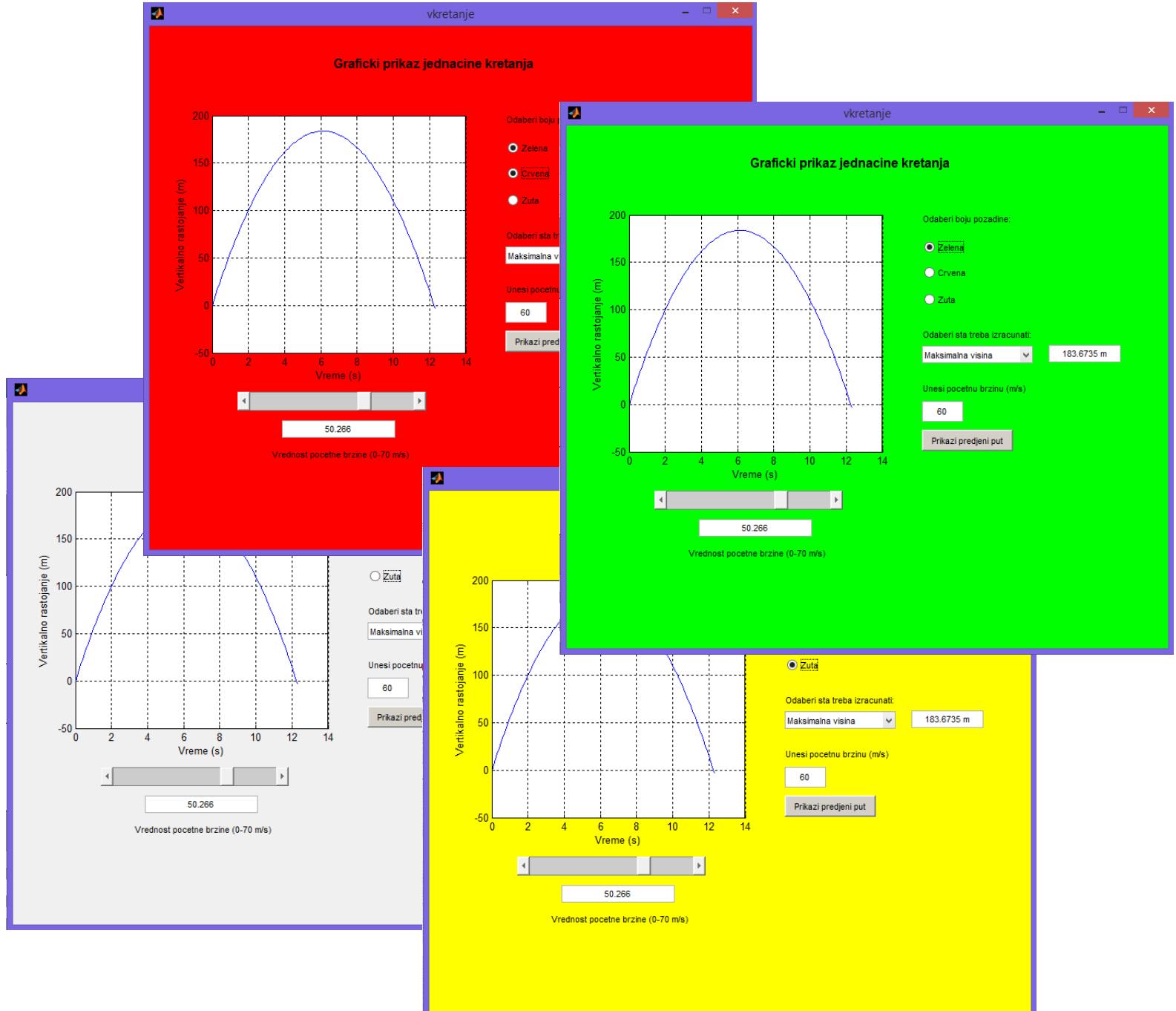
Kreirani "popupmenu" je najbolje kodirati pomoću "switch" komande koja u zavisnosti od "case" tj. izbora prikazuje da li je određeni (edit7 ili edit8) prozor vidljiv.

Prvi slučaj ne prikazuje ništa i tu je da ne bi na početku "popupmenu" bio prazan (ili i našem slučaju, prikazao tri tačke).

Ne zaboravite "end" kojim morate završiti "switch" komandu.

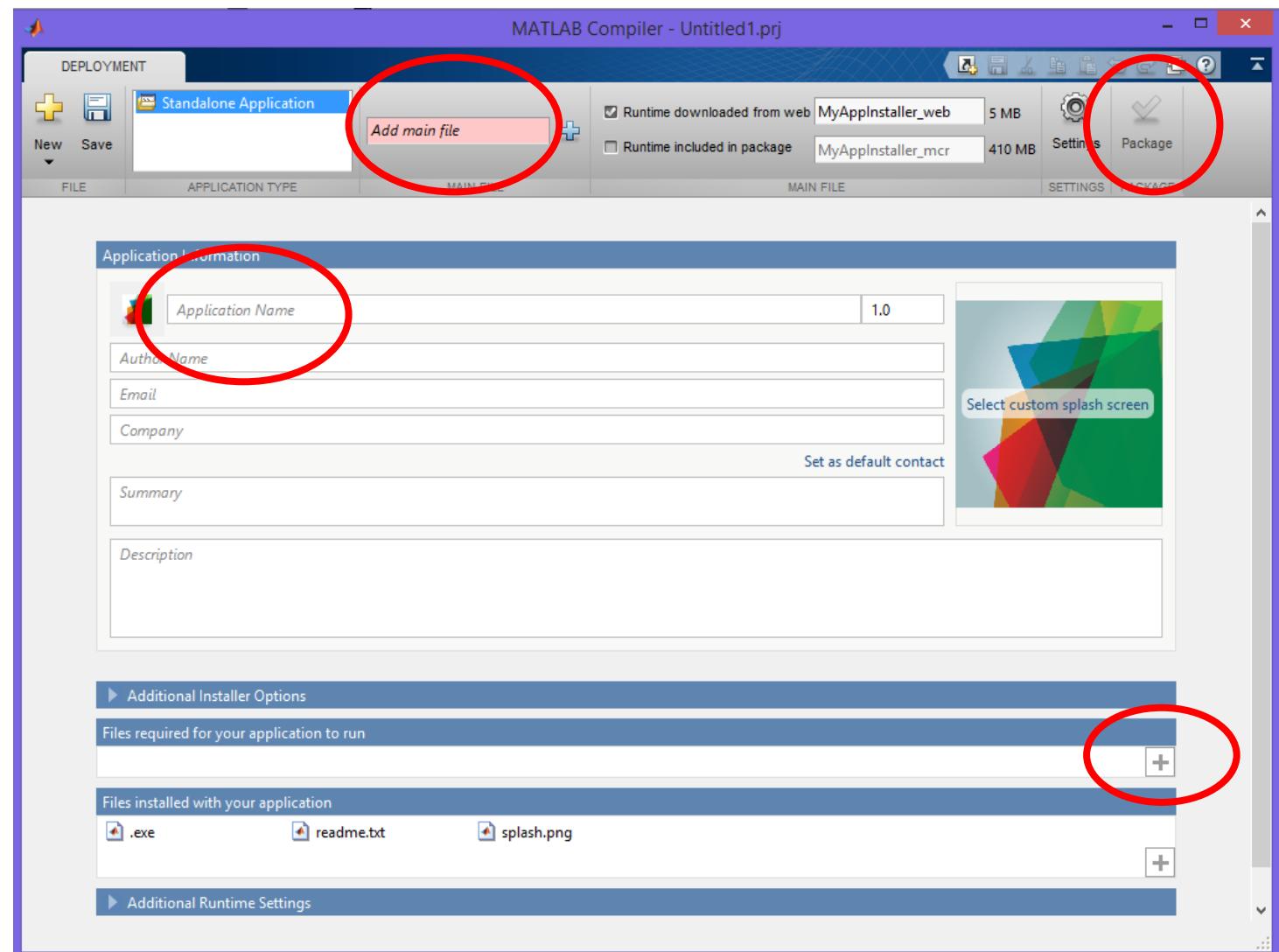
Finalni GUI

- Konačno, imamo finalnu verziju programa.
- Naravno, mogućnosti za "doterivanje" su neograničene.
- Kao zadatak na vežbama probajte da uradite i sledeće:
 - ❑ Napravite još neki program koji iscrtava zadatu funkciju na osnovu ubačenih poznatih parametara (iz oblasti radiohemije, kinetike, termodinamike ...)
 - ❑ Napraviti još neki dodatni vizuelni efekat (npr. ukoliko su uključena oba dugmeta za crvenu i žutu boju, da nam pozadina bude narandžasta). Dajte mašti na volju.
- Naučimo sada kako možete da napravljeni program objavite kao .exe fajl, tako da bi mogli da ga koriste i kolege koje nemaju instalisan MATLAB.



Eksportovanje MATLAB programa kao .exe fajla

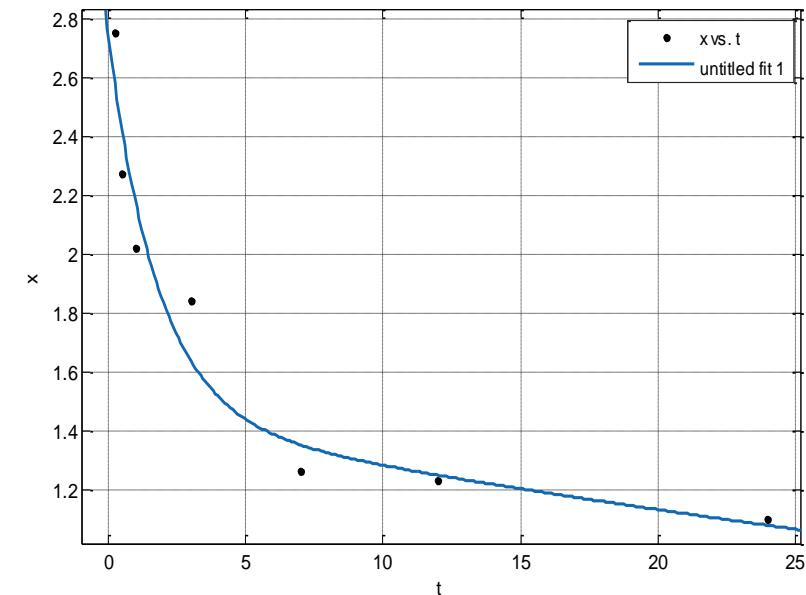
- Napravimo tzv. "standalone" aplikaciju od našeg programa.
- Naime, MATLAB nam omogućava da eksportujemo napravljene programe u vidu .exe-fajla koji možemo pokrenuti i na računarima koji nemaju instalisan MATLAB.
- To se radi pozivanjem komande: >>deploytool
- ❑ Izaberemo: Application Compiler i upišemo:
- ❑ Naziv .exe-fajla (željeni naziv aplikacije).
- ❑ Dodamo glavni .m-fajl (vkretanje.m).
- ❑ Po želji, upišemo svoje lične podatke.
- ❑ Dodamo dopunske fajlove koji su potrebni za pokretanje programa (vkretanje.fig).
- ❑ Pokrenemo "Package" u gornjem desnom uglu.
- MATLAB je kreirao naš .exe-fajl koji možemo pokrenuti u bilo kom Windows okruženju (ili Linux uz adekvatni emulator).
- Gotovo!



Pravljenje samostalne GUI aplikacije za fitovanje

- Za ovaj zadatak, uzećemo vrednosti koncentracije leka koji je praćen u nekom organu u određenom vremenskom periodu.
- Vrednosti koncentracija date su sa nekom greškom (SD).
- Ovo se inače zove farmakokinetička analiza i sastavni je deo istraživanja koji se bave ispitivanjem novih lekova.
- Zadatak je da koristeći MATLAB napravite samostalnu aplikaciju koja:
 - Uvozi tačke iz MS Excel fajla
 - Fituje ih dvostruko eksponencijalnom funkcijom oblika:
$$f(t) = a_1 \cdot \exp(-k_1 \cdot t) + a_2 \cdot \exp(-k_2 \cdot t)$$
 - Prikazuje u posebnoj tabeli odgovarajuće koeficijente fitovanja
 - Prikazuje u prozoru funkciju po kojoj je fitovano
 - Crta grafik (tačke sa greškama i fitovanu fju)
 - Određuje i obeležava površinu ispod krive
 - Određuje vrednost funkcije ekstrapolisanu na nultu vrednost nezavisno promenljive
 - Određuje poluvreme eliminacije leka
 - Eksportuje GUI kao samostalnu (*.exe) aplikaciju

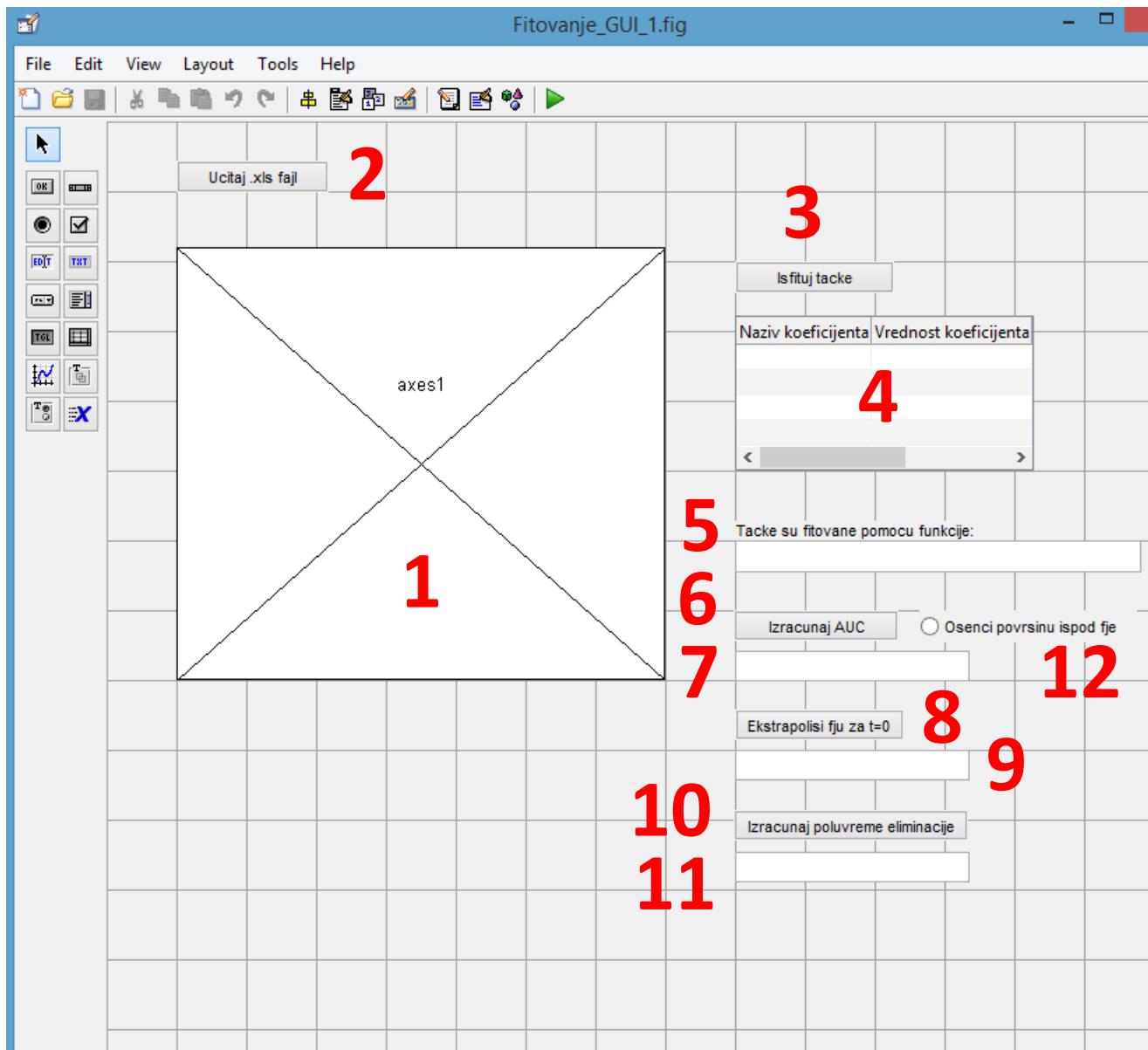
Vreme (h)	Koncentracija (ng/ml)	SD, ng/ml
0.25	2.75	0.15
0.5	2.27	0.22
1	2.02	0.11
3	1.84	0.2
7	1.26	0.16
12	1.23	0.12
24	1.1	0.13



Počećemo formiranjem grafičkog okruženja za GUI

- Kreiramo nov GUI i prevlačimo potrebne elemente:

- Panel za crtanje grafika (Tag: Axes1)
- Dugme za učitavanje .xls fajla (Tag: pushbutton1)
- Dugme za fitovanje (Tag: pushbutton2)
- Tabelu za ispis izračunatih koeficijenata fje (Tag: uitable1)
- „Edit text“ prozor za ispis fje po kojoj je fitovano (Tag: edit4)
- Dugme za računanje površine ispod krive (Tag: pushbutton3)
- „Edit text“ prozor za ispis računa AUC (Tag: edit3)
- Dugme za računanje ekstrapolisane vrednosti fje za t=0 (Tag: pushbutton4)
- „Edit text“ prozor za ispis ekstrapolisane vrednosti fje za t=0 (Tag: edit5)
- Dugme za računanje poluvremena eliminacije (Tag: pushbutton5)
- „Edit text“ prozor za ispis poluvremena eliminacije (Tag: edit6)
- Radio-dugme pomoću kojeg možemo osenčiti površinu ispod krive (Tag: radiobutton1)
- Dodamo adekvatni „Static text“ da bi objasnili šta je šta.



Kako izgleda naš Excel fajl sa tačkama koje želimo da fitujemo?

- Imamo tri kolone: A - Vreme (nezavisno promenljiva), B - Koncentracija (zavisno promenljiva), C- Greške
 - Zapazimo da su prvi redovi naših kolina u stvari nazivi kolona a ne njihove vrednosti

Pravimo kod za uvoženje tačaka iz Excel fajla (pod Tag: pushbutton1)

```
% --- Executes on button press in pushbutton1.  
function pushbutton1_Callback(hObject, eventdata, handles)  
% hObject    handle to pushbutton1 (see GCBO)  
% eventdata   reserved - to be defined in a future version of MATLAB  
% handles     structure with handles and user data (see GUIDATA)  
  
% Uvoz tacaka iz Excel fajla  
  
tacke=uigetfile('* xls'); % Ucitava .xls fajl u kome se nalaze sve tacke  
matrica=xlsread(tacke); % Pravi "double" matricu od svih ucitanih tacaka  
% Promenljiva "matrica" sadrzi samo brojeve, ne i nazine kolona  
  
% Razbijamo ucitane podatke na nazine kolona i tacke  
  
global vrednosti; % Pravimo da je promenljiva "vrednosti" globalna (trebace nam posle)  
  
[vrednosti,nazivi]=xlsread(tacke); % Ova zanimljiva komanda pravi dve  
% odvojene promenljive.  
% Prva je matrica "vrednosti" u kojoj su  
% sve vrednosti tacaka, a druga je matrica  
% "nazivi" koja daje samo nazine kolona.  
% Crtanje grafika
```

Osnovni
kod

Pravimo kod za crtanje grafika (pod Tag: pushbutton1)

```
%Crtanje grafika

global xdata; % Pravi da su vrednosti za podatke x-ose globalni.
               % Inace, promenljivu mozemo pozvati ukucavanjem "global xdata" u Command Window.
global ydata; % Pravi da su vrednosti za podatke y-ose globalni.
               % Takodje, promenljivu mozemo pozvati ukucavanjem "global ydata" u Command Window.
global error; % Pravi da su vrednosti za greske globalni.
               % Takodje, i ovu promenljivu mozemo pozvati ukucavanjem "global error" u Command Window.

xdata=matrica(:,1); % Kao x-vrednosti unosi tacke koje se nalaze u svim redovima
                      % ":" i prvoj koloni "1"
ydata=matrica(:,2); % Kao y-vrednosti unosi tacke koje se nalaze u svim redovima
                      % ":" i drugoj koloni "2"
error=matrica(:,3); % Kao greske unosi tacke koje se nalaze u svim redovima
                      % ":" i trecoj koloni "3"

errorbar(xdata,ydata,error,'-bo'); % Crtanje grafik sa greskama (slicno kao plot)
                                   % Povezuje tacke "o" punom linijom "-", koja su plave "b"
set(gca,'YLim',[0 vrednosti(1,2)]); % Postavlja granice za x i y osu.
                                         % "gca" komanda daje osobine osa, a YLim opseg za Y-osu
                                         % Y ide od 0 do broja u prvom redu matrice "vrednosti" u drugoj koloni

hold on; % Zadrzava osobine plota, tj. vezu izmedju tacaka
xlabel('Vreme(h)'); % Oznaka x-ose
ylabel('Koncentracija(ng/ml)'); % Oznaka y-ose
```

Pravimo kod za fitovanje zadatom funkcijom (pod Tag: pushbutton2)

```
% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Fitovanje

global xdata;    % Pravi da su vrednosti za podatke x-ose globalni.
                  % Inace, promenljivu mozemo pozvati ukucavanjem "global xadata" u Command Window.
global ydata;    % Pravi da su vrednosti za podatke y-ose globalni.
                  % Takodje, promenljivu mozemo pozvati ukucavanjem "global yadata" u Command Window.
global error;    % Pravi da su vrednosti za greske globalni.
                  % Takodje, i ovu promenljivu mozemo pozvati ukucavanjem "global error" u Command Window.

ft=fittype(@(a1,a2,k1,k2,x) a1*exp(-k1*x)+a2*exp(-k2*x), 'independent', {'x'}, 'dependent', {'y'}); %
Priprema fju za fitovanje
f=fit(xdata,ydata,ft);                      % Fituje po gornjoj funkciji, po zadatim tackama, generisuci
koeficijente a1,a2,k1,k2
plot(f,xdata,ydata,'r');                     % Crtanje isfitovanju funkciju crvenom linijom
hold all                                       % Zadrzava isfitovanu liniju na grafiku
errorbar(xdata,ydata,error,'-bo');            % Crtanje tacke sa greskama (slicno kao plot)
                                              % Povezuje tacke "o" punom linijom "-", koja su plave "b"
xlabel('Vreme(h)');                           % Oznaka x-ose
ylabel('Koncentracija(ng/ml)');               % Oznaka y-ose
```

Osnovni
kod

Pravimo kod za "izvlačenje" koeficijenata fita (Tag: pushbutton2) i njihovo prikazivanje u tabeli (Tag: uitable1)

```
% Sada da vidimo te koeficijente i fju koje smo dobili tokom fitovanja

cfnames=coeffnames(f); % Pravi promenljivu "cfnames" koja predstavlja nazive izracunatih koeficijenata:
a1,a2,k1,k2
cfval=coeffvalues(f); % Pravi promenljivu "cfval" koja predstavlja vrednosti izracunatih koeficijenata:
a1,a2,k1,k2
resultt=table(cfnames,cfval'); % Od dobijenih rezultata tj. izracunatih koeficijenata pravi tabelu sa
% nazivima i vrednostima koeficijenata. Ono
% """ je moralo da se stavi kod "cfval" jer
% MATLAB ovde pravu matricu sa 1 redom a 4
% kolone dok je kod "cfnames" to obrnuto.
resultc=table2cell(resultt); % Od gornje tabele pravi celiju da bi moglo da se to prikaze u GUI Table
elementu (uitable1)
set(handles.uitable1,'Data',resultc); % Ubacujemo u GUI tabelu nazive i vrednosti koeficijenata

koefal=cfval(1); % Iz matrice vrednosti koeficijenata ekstrahujemo prvi tj. a1
koefa2=cfval(2); % Iz matrice vrednosti koeficijenata ekstrahujemo drugi tj. a2
koefk1=cfval(3); % Iz matrice vrednosti koeficijenata ekstrahujemo treći tj. k1
koefk2=cfval(4); % Iz matrice vrednosti koeficijenata ekstrahujemo četvrti tj. k2

% Mali "hint". Ako niste sigurni kpoji koeficijet je koji, najbolje je
% pozvati matricu vrednosti koeficijenata sa nazivima (ovde je to "resultt")
% komandom "global resultt", pa videti koji je koji.
```

Pravimo funkciju od izračunatih koeficijenata da bi je zatim ispisali u posebnom prozoru (Tag: edit4)

```
% Pravimo funkciju od izracunatih koeficijenata  
fun=@(x) (koefal*exp(-koefk1*x)+koefa2*exp(-koefk2*x));  
  
% Prikazujemo fuknciju po kojoj je fitovano  
  
set(handles.edit4, 'String', [ 'y=' num2str(koefal) '*exp(' num2str(koefk1) '*x)+'  
num2str(koefa2) '*exp(-' num2str(koefk2) '*x) ']);
```

Računamo površinu ispod fitovane krive pritiskom na dugme (Tag: pushbutton3) i prikazujemo je u posebnom prozoru (Tag: edit3)

```
% --- Executes on button press in pushbutton3.  
function pushbutton3_Callback(hObject, eventdata, handles)  
% hObject    handle to pushbutton3 (see GCBO)  
% eventdata   reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Racunanje povrsine ispod fitovane krive nakon pritiska na dugme  
  
global fun;          % Mora da stoji, da bi upotrebio kreiranu funkciju  
global vrednosti; % Uzima onu matricu svih brojevnih vrednosti iz .xls fajla (za granice integrala)  
global donjagranica;  
global gornjagranica;  
  
donjagranica=vrednosti(1,1); % Uzima broj iz prvog reda u prvoj koloni (donja granica intergrala)  
gornjagranica=vrednosti(end,1); % Uzima broj iz poslednjeg reda u prvoj koloni (gornja granica  
integrala)  
  
auc=double(integral(fun,donjagranica,gornjagranica)); % Racuna integral u zadatim granicama  
set(handles.edit3,'String',[num2str(auc) ' ngh/ml']); % Ispisuje AUC u edit3 prozoru i dodaje  
jedinice
```

Osnovni
kod

Računamo maksimalnu vrednost funkcije ekstrapolacijom za t=0 pritiskom na dugme (Tag: pushbutton4) i to ispisuje u prozoru (Tag: edit5)

```
% --- Executes on button press in pushbutton4.  
function pushbutton4_Callback(hObject, eventdata, handles)  
% hObject handle to pushbutton4 (see GCBO)  
% eventdata reserved - to be defined in a future version of MATLAB  
% handles structure with handles and user data (see GUIDATA)
```

Osnovni
kod

```
% Racunanje maksimalne vrednosti fje ekstrapolacijom za t=0 pritiskom na  
% dugme  
  
global fun; % Mora da stoji da bi video podatke od ranije  
global maks; % Mora da stoji da bi videli ovu promenljivu kasnije  
  
maks=feval(fun,0);  
set(handles.edit5,'String',[num2str(maks) ' ng/ml']);
```

Računamo vreme za koje konc. leka padne na 1/2 od maksimalne vrednosti pritiskom na dugme (Tag: pushbutton5) i to ispisuje u prozoru (Tag: edit6)

```
% --- Executes on button press in pushbutton5.  
function pushbutton5_Callback(hObject, eventdata, handles)  
% hObject    handle to pushbutton5 (see GCBO)  
% eventdata   reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Racunanje vremena za koje koncentracija leka padne na 1/2 od maksimalne  
% vrednosti kada se pritisne dugme  
  
global maks;    % Mora da stoji da bi video podatke od ranije  
global koefal; % Mora da stoji da bi video podatke od ranije  
global koefa2; % Mora da stoji da bi video podatke od ranije  
global koefk1; % Mora da stoji da bi video podatke od ranije  
global koefk2; % Mora da stoji da bi video podatke od ranije  
  
syms x; % Pravi da je x simbolicka promenljiva da bi radila komanda "vpasolve"  
  
pve=vpasolve(koefal*exp(-koefk1*x)+koefa2*exp(-koefk2*x) == maks/2,x); % Racuna vrednost x  
za zadatu vrednost funkcije (koja je polovima od maksimalne vrednosti)  
set(handles.edit6,'String',[num2str(double(pve)) ' h']); % Ispisuje rezultat u prozoru
```

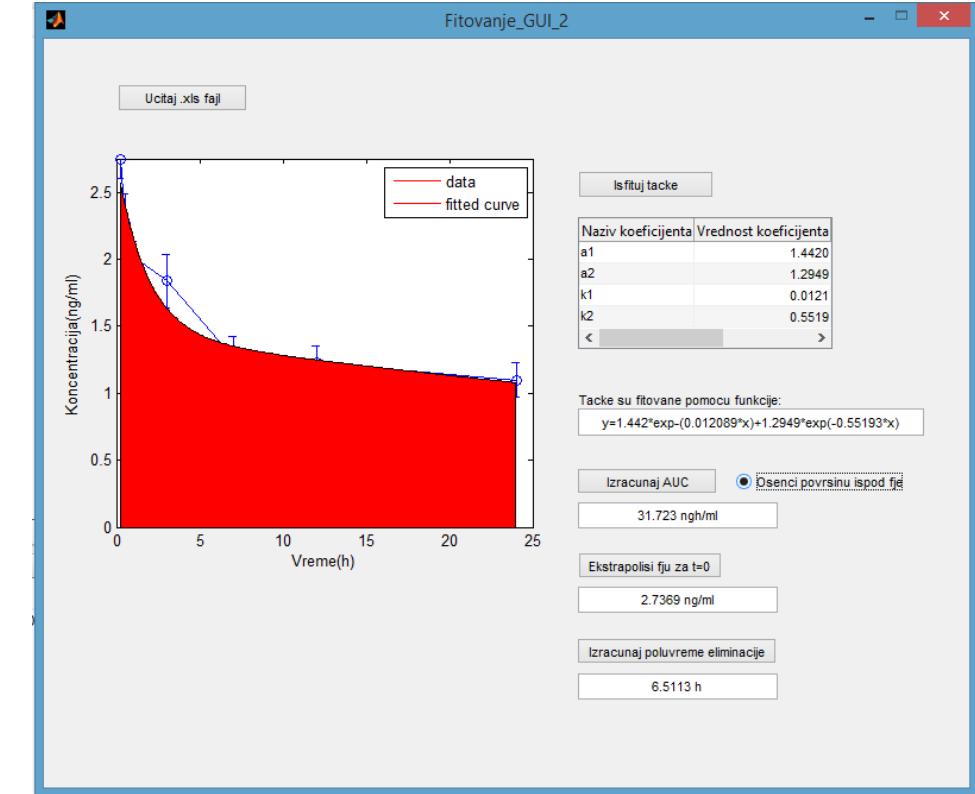
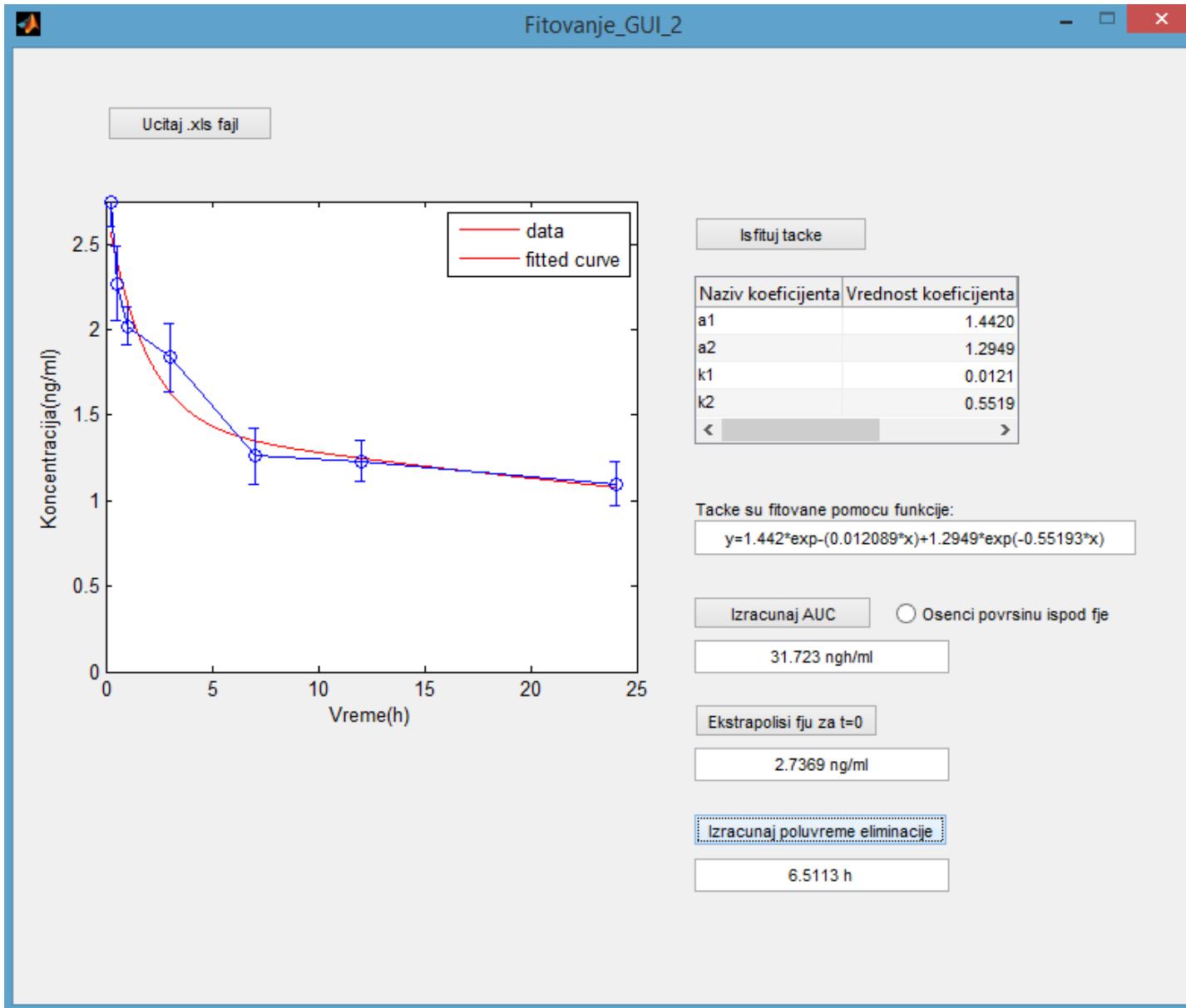
Osnovni
kod

Senčimo oblast integraljene površine pomoću radio-dugmeta (Tag: radiobutton1)

```
% Sencenje integraljena povrsine ispod isfitovane fje pomocu radio-dugmeta  
  
% --- Executes on button press in radiobutton1.  
function radiobutton1_Callback(hObject, eventdata, handles)  
% hObject    handle to radiobutton1 (see GCBO)  
% eventdata   reserved - to be defined in a future version of MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% Hint: get(hObject,'Value') returns toggle state of radiobutton1  
  
global fun; % Mora da stoji, da bi upotrebio kreiranu funkciju  
global donjagranica; % Mora da stoji, da bi znao donju granicu  
global gornjagranica; % Mora da stoji, da bi znao gornju granicu  
  
c=get(handles.radiobutton1,'Value'); % Pravi promenljivu za vredost "Value" radio-dugmeta  
if c==1; % Postavlja uslov da li je radio-dugme ukljuceno (br. 1 ako jeste)  
xg = donjagranica:0.1:gornjagranica; % Zadaje vrednosti za x koje ce bojiti sa korakom 0.1  
yg = fun(xg); % Zadaje fju do koje ce bojiti  
h=area(xg,yg,0); % Pravi promenljivu za bojeni prostor od y=0 do fje  
set(h,'FaceColor','r'); % Postavlja crvenu boju za povrsinu ispod fje  
else  
xg = donjagranica:0.1:gornjagranica; % Zadaje vrednosti za x koje ce bojiti sa korakom 0.1  
yg = fun(xg); % Zadaje fju do koje ce bojiti  
h=area(xg,yg,0); % Pravi promenljivu za bojeni prostor od y=0 do fje  
set(h,'FaceColor','w'); % Postavlja belu boju tj. obezbojava povrsinu ispod fje  
end
```

Osnovni
kod

Dodamo "Static text" gde treba, i program je završen



Sada program možemo eksportovati kao samostalnu *.exe aplikaciju korišćenjem komande: >>deploytool

Zadatak: Napraviti nekoliko različitih setova tačaka i fitovati ih različitim funkcijama