VEŽBA 11

Pregled

InkScape – program otvorenog izvornog koda namenjen vektorskom načinu crtanja koristi SVG format kao primarni način skladištenja crteža. SVG formatzasnovan je na XML jeziku za opis podataka u strukturi stabla. U prethodnoj vežbi, implementirane su (softverske) ekstenzije za InkScape primenom neposredne interpretacije XML dokumenta, izmene sadržaja i ponovnog izdavanja izmenjenih podataka u vidu XML (odnosno SVG) dokumenta primenom Python modula ElementTree.

Autori InkScape-a predvideli su i nešto sistematičniji pristup pisanju ekstenzija u čijem centru se nalazi modul inkex, a pored i preko njega dostupni su i drugi. U ovoj vežbi biće proučen ovaj pristup.

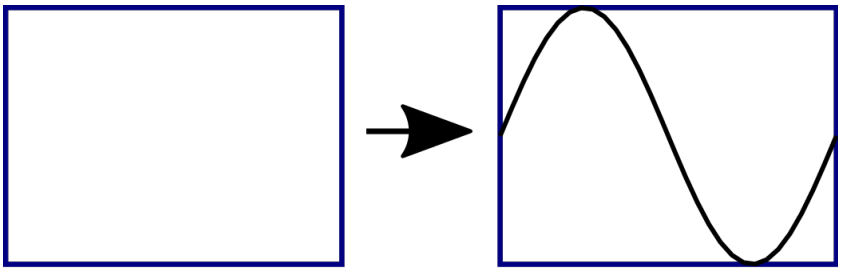
SVG primitive

Izlomljena linija

```
<polyline points="x1 y1, x2 y2,  x3 y3, x4 y4, ... , xn yn"
style="stroke:black; stroke-width:3.0; fill:none; " />
```

Zadaci

1. Korišćenjem modula *inkex* napisati ekstenziju koja crta pravougaonik (*rect* element) oko celog crteža (koristiti *width* i *height* atribute svg dokumenta), a na pravougaonik dodati i obe dijagonale (*line* elementi). Dijalog za zadavanje parametara nije potreban.
2. Prethodni zadatak uraditi tako što će se pre poziva ekstenzije unutar dijaloga zadavati string parametar koji će se konvertovati u *float* vrednost koja će potom biti korišćena za određivanje debljine linija. Koristiti *try*/*except* konstrukciju za detekciju ValueError izuzetka koji se dobija ako string nije moguće konvertovati u *float*. Debljina mora biti veća od 0. U slučaju bilo kakve greške debljina linije treba da bude 1.0
3. Promeniti boju linije svim izabranim elementima. Koristiti modul *inkex*. Za izbor boje dijalog preurediti tako da daje izbor crne, crvene, zelene i plave boje pomoću radio dugmića. Za manipulaciju stilovima elemenata koristiti modul *simplestyle*.
4. Napisati ekstenziju za InkScape korišćenjem modula *inkex*. koja iscrtava jednu periodu sinusoide crnom bojom debljine linije 1.0 mm unutar izabranog pravougaonika onako kako ilustruje sledeća slika:



selektovani element pre poziva ekstenzije *izgled elementa nakon poziva ekstenzije*

Obavestiti korisnika porukom o grešci ako nema izabranih elemenata, izabrano je više od jednog ili izabrani element nije pravougaonik. Za crtanje sinusoide koristiti primitivu `polyline`. Korisnik zadaje jednu `int` vrednost između 10 i 60 unutar dijaloga. Za poziv funkcije `sin` biće neophodno importovanje modula `math`.

Dodaci

Razni tipovi podataka u dijalogu za podešavanje pre poziva ekstenzije

Izgled dijaloga za zadavanje parametara zadaje se u `inx` fajlu ekstenzije. Tipovi relevantni za ovu vežbu navedeni su ispod. Imena atributa bi trebalo da govore sami za sebe. U slučaju `string` i `int` parametara tekst sadržaj elementa predstavlja početnu vrednost u trenutku pojave dijaloga, dok kod radio dugmića tekst deo predstavlja natpis pored odgovarajućeg dugmića. Takođe, kod radio dugmića atribut `value` određuje vrednosti od kojih će jedna biti vraćena kao vrednost parametra u zavisnosti od izabranog dugmića.

String

```
<param name="width" type="string" gui-text="Sirina
linije">2.0</param>
```

Integer

```
<param name="segs" type="int" min="10" max="60" gui-text="Br.
segmenata">20</param>
```

Višestruki, međusobno isključiv izbor – radio dugmići

```
<param name="color" type="optiongroup" gui-text="Izaberite
boju">
  <option value="black">black</option>
  <option value="red">red</option>
  <option value="green">green</option>
  <option value="blue">blue</option>
</param>
```

Obaveštenja korisniku o nastalog grešci

U posebnom dijalogu koji se prikazuje korisniku može se pojaviti proizvoljna poruka o grešci koja treba da obavesti o pogrešnom izboru elemenata ili kombinaciji parametara koji nije promenljiv na izabrane elemente. Za prikazivanje ove poruke treba iz modula inkex pozvati funkciju:

```
inkex.errormsg('string sa porukom o gresci')
```

String koji je argument funkcije će bez izmene biti prikazan korisniku.

Pozivanje funkcija za rad sa XML stablom – modul ElementTree

Modul *inkex* za rad sa XML dokumentima koristi ekvivalent modulu *ElementTree* korišćenom na prethodnim vežbama. Zbog toga pri pisanju InkScape ekstenzije korišćenjem *inkex* modula nije potrebno importovati modul *ElementTree*, to je već učinio modul *inkex*. Jedino što se neposredni pozivi funkcijama modula, kao i kreiranje objekata iz tog modula obavlja na nešto izmenjen način. Sledi primer kreiranja novog elementa (sa tagom *svg:line*) ovom metodom:

```
inkex.etree.Element(inkex.addNS('line','svg'))
```

Pomoć pri radu sa *namespece*-ovima XML-a

Savremeni SVG dokumenti koriste mnoštvo namespace-ova, a modul ElementTree očekuje da namespace bude pridružen svakom imenu (kvalifikovano ime) ako se u datom XML dokumentu koriste namespace-ovi i to u formi

```
'{http://www.w3.org/2000/svg}line'
```

što u okviru SVG fajla odgovara tag-u oblika *<svg:line>*. Gornja notacija je dosta nepregledna, a pošto se i pri najjednostavnijim intervenicjama na SVG fajlu zahteva puno rada sa kvalifikovanim imenima, *inkex* nudi rešenje za kraće i preglednije zapisivanje u vidu *addNS* funkcije:

```
inkex.addNS('line','svg'))
```

Ova funkcija vraća string prikazan u prethodnom primeru nekoliko redova iznad. Korišćenjem ovakvog metoda kvalifikacije imena zapis postaje sličniji onom koji se koristi i u samim XML dokumentima (*<namespace:ime>*). Sam modul *inkex* sadrži rečnik sa URL-ovima koji odgovaraju svakom namespace-u, pa o tome nije neophodno dodatno voditi računa.

Tipična forma ekstenzije pisane pomoću *inkex* modula

Suština rada sa *inkex* modulom je sledeća: kreira se nova klasa koja nasleđuje klasu *inkex.Effect*. U konstruktoru nove klase poziva se konstruktor nasleđene klase, a ako se ekstenziji prosleđuju i parametri, potrebno je pozvati i metodu add_option objekta OptionParser koja je ugrađena u objekat, potreban broj puta – za svaku opciju po jednom.

Metoda *effect* je nasleđena prazna. Treba je ponovo definisati za novi objekat jer će ona biti ta koja zapravo radi modifikaciju SVG fajla u skladu sa poslom koji ekstenzija treba da obavi.

```
import inkex

class NEx(inkex.Effect):
  def __init__(self):
    inkex.Effect.__init__(self)
    # za svaki parametar treba dodati po jednu opciju
    # ako ekstenzija nema parametre, moze se isostaviti
    self.OptionParser.add_option('-p', '--par',
      action='store', dest='par', type='string',
      default='2.0', help='Prosledjeni parametar')

  def effect(self):
```

```
        # u ovoj funkciji treba napisati kompletnu
        # funkcionalnost ekstenzije (tipa effect).
        # U ovom primeru funkcija je prazna
        pass

# glavni program: kreiranje objekta i poziv aktivacione funkcije
ne = NEx()
ne.affect()
```

Primer dodavanja jedne linije u crtež

Ovaj primer ilustrovaće način dodavanja novog elemenata u crtež – linije, a istovremeno pokazaće i način korišćenja parametara – debljina linije – koje korisnik unosi u dijalog pre poziva ekstenzije. Ovi parametri na raspolaganju su u funkciji effect kao promenljiva *self.options.par* gde se poslednje ime u nizu poklapa sa stringom prosleđenim kao parametar dest u pozivu metode *add_option* u konstruktoru klase *NEx* (prethodni primer).

```
def effect(self):
  # preuzimanje i provera parametara (parametri su uvek string)
  try:
    sw = float(self.options.par)
  except ValueError:
    sw = 1.0  # ako je broj neispravan, default je 1.0
  if sw < 0.0:
    sw = 1.0  # i negativne vrednosti su neregularne
  # koreni element crteza - svg
  svg = self.document.getroot()
  # koreni element sadrzi atribute koji definisu sirinu i visinu
  w = float(svg,get('width'))
  h = float(svg,get('height'))
  # pronalazenje default layer-a u crtezu – element g
  g = svg.find('./'+inkex.addNS('g','svg'))
  # kreiranje novog elementa - line
  nel = inkex.etree.Element(inkex.addNS('line','svg'))
  nel.set('x1','0.0')
  nel.set('y1','0.0')
  nel.set('x2',str(w))
  nel.set('y2',str(h))
  nel.set('style','stroke:black; stroke-width:'+str(sw)+'; ')
  # dodavanje novog elementa roditelju – elementu g
  g.append(nel)
```

Obratiti pažnju na tretman parametra *par* – širina linije. Treba imati u vidu da najčešće ništa ne sprečava korisnika da kao parametar unese besmisleni parametar preko dijaloga, pa u takvom slučaju treba obezbediti razuman default – 1.0 u ovom primeru.

Obrada elemenata selektovanih pre poziva ekstenzije

U trenutku izvršavanja funkcije effect, promenljiva *self.selected* sadrži rečnik sa svim selektovanim elementima crteža. Ključevi su InkScape identifikatori elementa oblika *elementnnnn* gde je *nnnn* slučajno genericani broj, a *element* tag odgovarajućeg tipa

elementa, npr. *line3541*, a element pod tim ključem je sam selektovani element (tj. ElementTree objekat koji ga predstavlja). Sledi primer promene boje svakog elementa u crvenu.

```
def effect(self):
  for k in self.selected.keys():
    self.selected[k].set('style',
                         'stroke: red; stroke-width: 2.0; ')
```

Nedostatak ovog rešenja je to što iako će svakom elementu biti promenjena boja linije, biće promenjena i širina iste te linije, što često nije ono što bi korisnik želeo. Problem je u tome što je style jedan jedinstven string koji u sebi sadrži sve podatke. Promena samo pojedinih elemenata bez promene drugih nije trivijalna operacija. Rešenje nudi modul opisan u nastavku.

Sistem za obradu stilova elemenata

U SVG formatu svaki element slike je opisan jednim XML elementom i tag tog elementa određuje njegov tip. Atributi pak određuju pojedinosti vezane za element, poput koordinata, boje, tipa linije itd. Jedan to tih atributa – *style* – iako je samo jedan, određuje veći broj osobina objekta posredstvom samo jednog stringa. Modul *simplestyle* omogućuje da se svim elementima koje atribut style određuje manipuliše odvojeno.

```
# na pocetku modula za ekstenziju
import simplestyle
...
# kasnije unutar funkcije effect
xstyle = simplestyle.parseStyle(svgelem.get('style'))
# xstyle je sada recnik ciji su
# kljucevi elementi atributa style
xstyle['stroke'] = 'blue'
# kada se izmene urade, treba ponovo formirati string
# i upisati ga na mesto atributa style SVG elementa
svgelem.set('style',simplestyle.formatStyle(xstyle))
```

Funkcija *parseStyle* prevodi string sa elementima stila (oblika *'element1: vrednost1; element2: vrednost2; ... elementN: vrednostN;'*) objekta u rečnik. U obliku rečnika, svakom elementu se može pristupiti direktno. Nakon što je intervencija na vrednostima elemenata završena poziva se funkcija *formatStyle* koja rečnik opet prevodi u string koji treba da bude deo SVG fajla kao atribut *style*.

Generisanje koordinata sinusoide unutar zadatog proavougaonika

Generisani element će biti izlomljena linija (polyline). Pretpostavka je da su sledeće promenljive tipa float i da su postavljene na odgovarajuće početne vrednosti: *x* – x-pozicija leve ivice; *y* – y-pozicija gornje ivice; *h* – visina pravougaonika; *w* – širina pravougaonika.

sgs je celobrojna promenljiva i sadrži broj linijskih segmenata sinusoide.

```
new = inkex.etree.Element(inkex.addNS('polyline','svg'))
sincoords = ''
dx = (2*math.pi)/(sgs-1)
rdx = w/(sgs-1)
A = h/2
M = y+A
```

```
for i in range(sgs):
  sincoords += '%.2f %.2f, ' % (x+rdx*i,M-A*math.sin(dx*i))
# zarez i prazno mesto iza poslednjeg para treba ukloniti
new.set('points',sincoords[:-2])
```