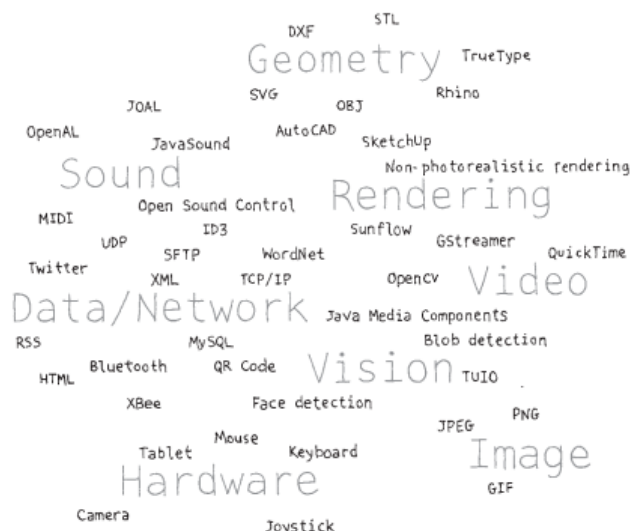


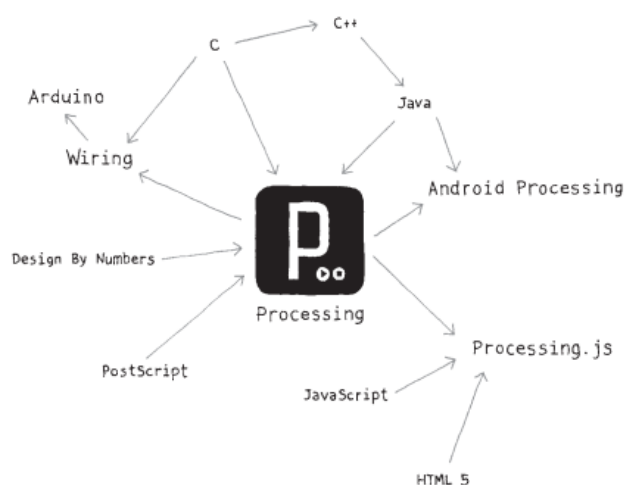
Čas br.1 – Uvod u Processing

“Processing is an open source programming language and environment for people who want to program images, animation, and sound. It is used by students, artists, designers, architects, researchers, and hobbyists for learning, prototyping, and production. It is created to teach fundamentals of computer programming within a visual context and to serve as a software sketchbook and professional production tool. Processing is developed by artists and designers as an alternative to proprietary software tools in the same domain.” — www.processing.org



1 Mnogi tipovi podataka mogu biti ulaz ili izlaz u/iz Processing-a

Kao i ljudski jezici, programski jezici pripadaju određenim familijama srodnih jezika. Processing je dijalekt programskog jezika Java; sintaksa koju koristi je skoro pa identična, ali Processing posjeduje još i dodatke vezane za grafiku i interakciju. Grafički elementi Processing-avezani su za PostScript i OpenGL.



Instalacija i pokretanje

Processing je besplatan za download. Odlaskom na URL <http://www.processing.org/> pronađite sekciju za download.

Preuzmite Processing softver s Processing web stranice, biti će to kompresovani file koji treba raspakovati u folder.

Processing razvojno okruženje je podjednostavljeno za pisanje kompjuterskog koda. Vrlo je jednostavan za korištenje (poput nekog Text editora skupa s media player-om). Processing programi se naziva *sketch*, uz ime file-a posjeduje mjesto za kucanje koda, neke dugmiće za sačuvati program (Save), otvoriti program (Open), te pokrenuti ga (Run).

Vježba br.1.

Kako bismo provjerili da sve radi ispravo pokrenite neki od Processing primjera. Otiđite na FILE -> EXAMPLES, te odaberite neki primer (npr.: Topcs -> Drawing -> ContinuousLines).

Kada otvorite primjer pokrenite ga klikom na Run dugme.

Kodiranje u Processing-u

Kako bismo započeli s kodiranjem bitno je naučiti nekoliko osnovnih pravila sintakse. Postoje tri vrste izjava koje možemo pisati:

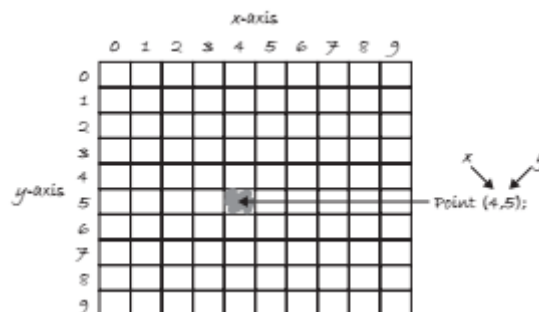
- Pozive funkcija
- Operacije dodjeljivanja
- Kontrolne strukture

Za sada, svaka linija koda biti će poziv funkcije, kao npr:

```
line(0, 0 ,200,00);
```

Draw

Kompjuterski ekran sastavljen je od elemenata koji se nazivaju pikseli. Svaki piksel ima poziciju koja je definirana koordinatama unutar mreže.



3 Tačka

Primjetite gdje je postavljeno koordinatno ishodište.

Koristiti ćemo pozive funkcija koje su ugrađene u Processing za kreiranje različitih slika. Funkcije su jedni od osnovnih gradivnih blokova Processing programa. Ponašanje funkcije definisano je parametrima koje joj proslijedimo.

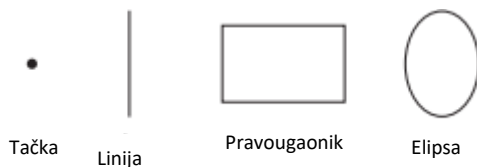
Vježba br.2

Nacrtajte prozor korištenjem `size()` funkcije. `size` funkcija uzima dva parametra: širinu i visinu prozora u pikselima: `size(širina, visina);`

Primjetite da poslije poziva funkcije uvijek stavljamo „;“!!

Pokušajte nacrtati prozore dimenzija: 200x200, 400x400, 600x800.

Jednostavni oblici



Za svaki oblik potrebno je znati šta je potrebno za specifikaciju lokacije i veličine (kasnije i boje) tog oblika.

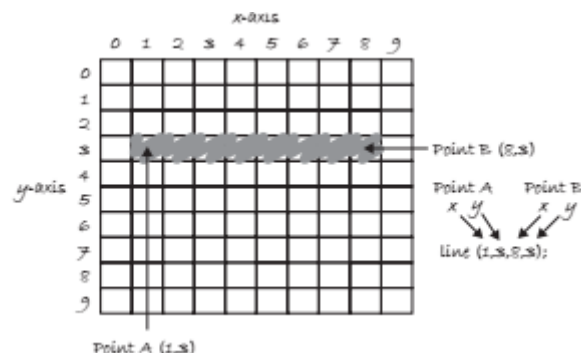
Tačka

Na slici 3 već smo vidjeli prikaz tačke u koordinatnoj ravni. Na istoj slici prikazana je i funkcija za crtanje tačke u Processingu: `point(x, y);`

Linija

Linija je zapravo samo produženje na koncept tačke, ogradaena je s početnom i krajnjom tačkom, dakle potrebno je da znamo njihove koordinate. Funkcija koju ćemo koristiti je:

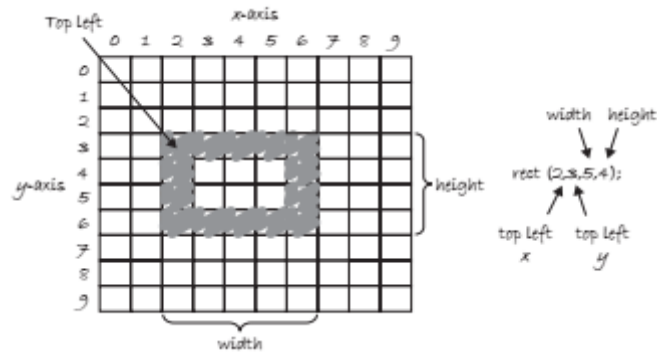
`line(x1, y1, x2, y2);`



4 Linija

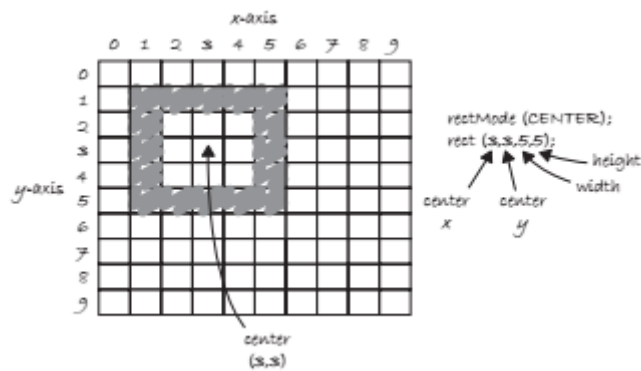
Pravougaonik

Postoji nekoliko načina crtanja pravougaonika no uvijek koristimo funkciju `rect()`; .
 Osnovni je onaj u kojem poznajemo koordinate gornjeg lijevog ugla te dužine stranica pravougaonika.



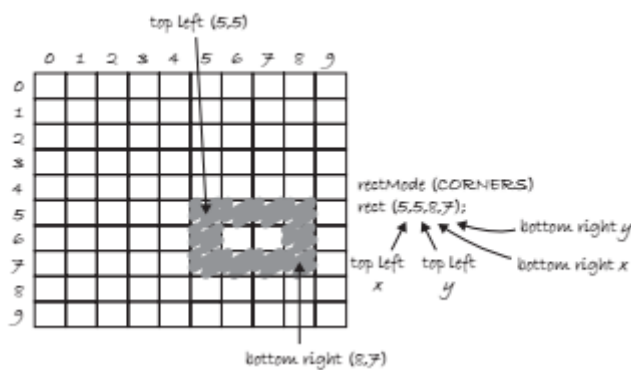
5 Pravougaonik - CORNER mode

Ukoliko znamo gdje želimo da bude centar pravougaonika:



6 Pravougaonik - CENTER mode

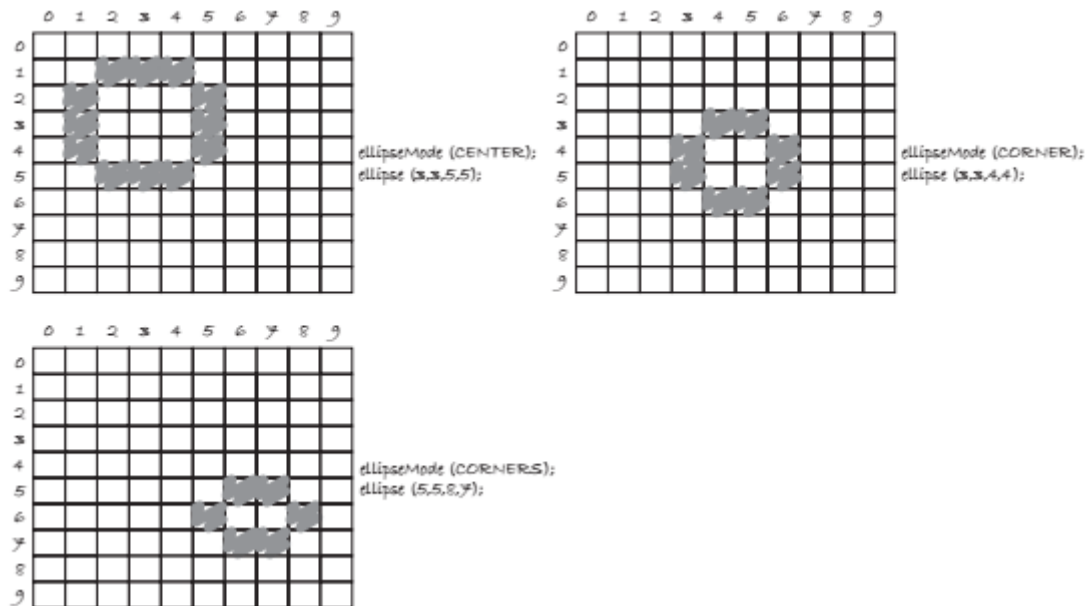
Treći način je ako znamo koje su koordinate gornjeg lijevog i donjeg desnog kuta:



7 Pravougaonik - CORNERS mode

Elipsa

Elipsa kao i pravougaonik ima nekoliko načina crtanja, mi ćemo ipak koristiti za početak samo jedan – onaj u kojem znamo koordinate centra.



8 Elipsa

Vježba br.2.

Testirajte gore navedene funkcije kroz primjer i spasite program pod nazivom **primitivni_oblici**.

Debljina poteza

Inicijalna debljina poteza oblika je 1 piksel. Upotrebom funkcije `strokeWeight()` možemo promijeniti ovu vrijednost.

Vježba br.3.

Otvorite novi program i upišite sljedeći kod:

```
size(480, 120);
smooth();
ellipse(75, 60, 90, 90);
strokeWeight(8); // postavi debljinu poteza na 8 piksela
ellipse(175, 60, 90, 90);
ellipse(279, 60, 90, 90);
strokeWeight(20); // postavi debljinu poteza na 20 piksela
ellipse(389, 60, 90, 90);
```

Boja

Nijanse sive

Svi oblici do sada bili su bijeli s crnim okvirom/potezom, a boja pozadinskog ekrana siva. Kako bi promijenili te vrijednosti koristiti ćemo funkcije `background()`, `fill()` i `stroke()`.

Vrijednosti parametara ovih funkcija su u rasponu od 0 do 255, gdje 255 predstavlja bijelu a 0 crnu, sve između nijanse su sive.

Šta bi radile funkcije `noFill()`, `noStroke()`?

Vježba br.4.

U prozoru dimenzija 400x400 potrebno je prikazati krug pravougaonik i elipsu. Boja pozadinskog ekrana treba da je bijela.

- Krug treba da bude centriran oko tačke (120,120) prečnika 100. Potez treba da je debljine 8px, a sam krug treba da bude ispunjen nekom nijansom tamno sive.
- Pravougaonik treba da počne (gornji lijevi ugao) u tački (220,20) a stranice trebaju da su: širina 80 i visina 120 piksela. Pravougaonik ne bi trebao imati potez.
- Elipsa treba da bude centrirana oko tačke (220,320), horizontalna osa 200 a vertikalna 50px. Elipsa bi trebala imati potez u crnoj boji debljine 2px ali sama elipsa da ne bude ispunjena nikakvom bojom.

RGB

Osim sivih nijansi moguće je ispuniti oblike i nekom drugom bojom. Svaka boja ima crvenu (red), zelenu (green) i plavu (blue) komponentu pa s tim pozivamo sve gore navedene funkcije za bojanje s ova tri parametra.

Primjer:

```
size(480, 120);  
noStroke();  
smooth();  
  
fill(255, 0, 0); // crvena  
ellipse(132, 82, 200, 200); // crveni krug  
fill(0, 255, 0); // zelena  
ellipse(228, -16, 200, 200); // zeleni krug  
fill(0, 0, 255); // plava  
ellipse(268, 118, 200, 200); // plavi krug
```

Vježba br.5.

Nacrtajte nekoliko oblika i ispunite ih različitim nijansama crvene, zelene i plave. Pozadina bi trebala biti u nijansi sive.

Domaća zadaća

Instalirati Processing na ličnom računaru te kreirati program pod nazivom Moog. U ovom programu trebate da kreirate lika koji će biti vaš pratioc tokom ovog kursa. Koristite funkcije koje smo učili.

Kao inspiracija, evo Zoog-a 😊

