SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
U VARAŽDINU

Projekt iz kolegija Odabrana poglavlja matematike

Prirodni koordinatni sustav u canvas elementu

U Varaždinu, svibanj 2017.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
U VARAŽDINU

Filip Gatarić, 42676/13-R
Petra Lončar, 42140/13-R
Ivan Belec, 41940/13-R

Projekt iz kolegija Odabrana poglavlja matematike

Prirodni koordinatni sustav u canvas elementu

Nositelj kolegija: Prof. dr. sc. Blaženka Divjak

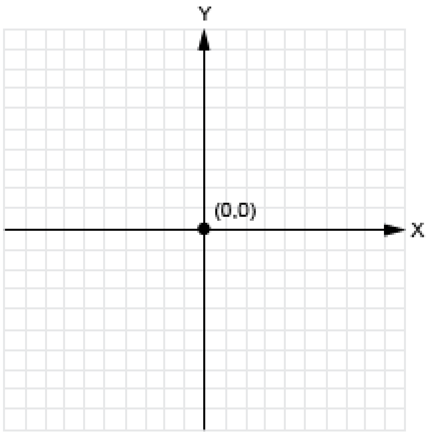
U Varaždinu, svibanj 2017.

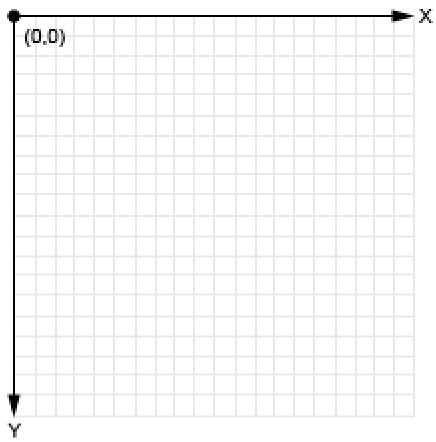
Sadržaj

Zadatak 1.

Opišite ukratko canvas element. Objasnite njegov koordinatni sustav i neke mogućnosti koje nudi. Posebno obratite pažnju na crtanje linija.

Canvas je element (tag) unutar HTML-a 5 sa oznakom <canvas>. Predstavlja "rastersku površinu za crtanje ovisno o rezoluciji koja se može koristiti za iscrtavanje grafova, grafike igara i drugih slika tokom izvršavanja"[1]. Za iscrtavanje grafičkih elemenata, fotografskih kompozicija i pravljenje jednostavnih animacija potrebno je koristiti neki skriptni jezik, a najčešće se koristi JavaScript. Canvas je pravokutnog oblika, te mu možemo zadati širinu i visinu. Pomoću canvasa možemo prikazati podatke pomoću grafova, izraditi animacije i interaktivne igre. Canvas element sam po sebi nije vidljiv, već mu je potrebno dodati određena svojstva, a najmanje što možemo nacrtati unutar canvasa su: točka, pravac, pravokutnik, krug i slično. Canvas koordinatni sustav se razlikuje od nama dobro znanog Kartezijevog koordinatnog sustava. U Kartezijevom sustavu pozicije se određuju pomoću X i Y koordianta, X-os se prostire horizontalno, a Y-os vertikalno. Dok centar ima poziciju X=0 i Y=0 , što kraće zapisujemo kao (0,0). Canvasov koordinatni sustav postavlja točku u gornjem lijevom uglu canvas-a sa X koordinatom koja se prostire desno i Y koordinatom koja se prostire prema dolje do samog dna canvas-a. Canvas prostor nema vidljivih negativnih točaka već će objekt nestati sa strane.





Slika 1: Usporedba Kartezijevog i Canvas koordinatnog sustava

Kako bi crtali linije potrebna nam je canvasova metoda za crtanje linija:

```
var c=document.getElementById("myCanvas");
var ctx=c.getContext("2d");
ctx.beginPath();
ctx.moveTo(x1,y1);
ctx.lineTo(x2,y2);
```

```
ctx.stroke();
```
[3]. Da bi nacrtali neku liniju potrebno je započeti put, prebaciti se na početnu poziciju (x1, y1), a canvas će prema poziciji (x2, y2) nacrtati liniju.

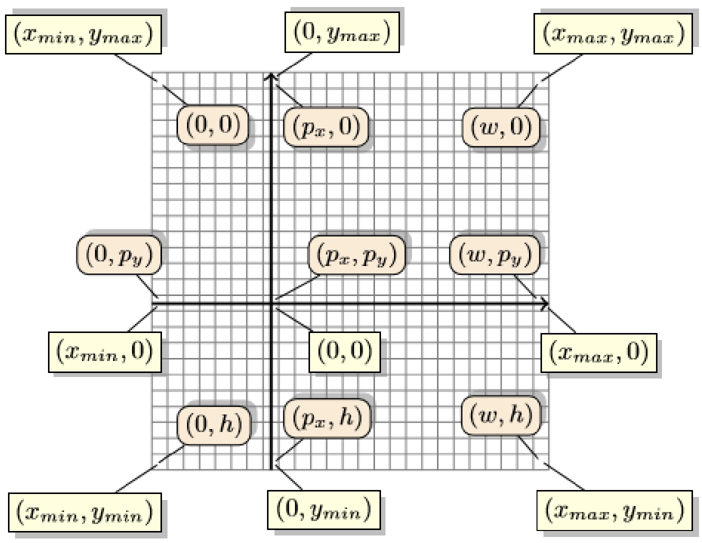
```
moveTo(x,y)- pomiče „olovku“ na zadanu početnu točku

lineTo(x,y)- povlači crtu do zadane završne točke
```

Zadatak 2.

Opišite preslikavanje koje prirodni koordinatni sustav preslikava na koordinatni sustav canvas elementa. Objasnite od kojih dijelova se sastoji to preslikavanje i povežite ga s linearnim operatorima. Objasnite povezanost jediničnih dužina u prirodnom kooordinatnom sustavu s jediničnim dužinama u koordinatnom sustavu canvas elementa.

Neka je V_1 realni dvodimenzionalni vektorski prostor koji predstavlja list papira. Prostor V_1 osim vektorske strukture ima i točkovnu strukturu pa ga možemo gledati i kao realni afini prostor. Veza izmedu točkovne i afine strukture je intuitivno jasna u smislu da je orijentirana dužina uredeni par dvije točke pa je na taj način jasna klasična veza izmedu vektora i točaka. Isto tako, na V_1 imamo standardni skalarni produkt pa na V_1 imamo metriku, tj. možemo preko skalarnog produkta definirati duljinu vektora, kut izmedu vektora, okomitost vektora. Neka je V_2 realni dvodimenzionalni vektorski prostor koji predstavlja površinu monitora. Sve što smo rekli o vektorskoj i točkovnoj strukturi prostora V_1 vrijedi takoder i za prostor V_2. Štoviše, prostori V_1 i V_2 su izomorfni kao vektorski prostori, također i kao afini prostori, te isto tako i kao unitarni prostori.



Slika 2: Preslikavanje koordinatnog sustava

Neka su $A_1 = \{\,(x_{max} - x_{min}\,, 0), (0,\ y_{max} - y_{min})\,\}$, $A_2 = \{\,(1,0), (0,1)\,\}$ dvije baze za vektorski prostor V_1. Isto tako, neka su $B_1 = \{\,(w,0), (0,-h)\,\}$, $B_2 = \{\,(1,0), (0,1)\,\}$ dvije baze za vektorski prostor V_2 .

Najprije tražimo linearni operator $K : V_1 \rightarrow V_2$ koji vektore iz baze A_1 preslikava u bazu B_1 .
Matrica tog operatora u paru baza (A_1, B_2) je: $K_{(A_1,B_2)} = \begin{bmatrix} w & 0 \\ 0 & -h \end{bmatrix}$.

Nas zanima matrica operatora K u paru kanonskih baza (A_2, B_2). Iz teorije znamo da je

$$K_{(A_2,\,B_2)} = T^{-1} K_{(A_1,\,B_2)} S$$

gdje su S i T matrice prijelaza između odgovarajućih baza, tj.

$$A_1 \xrightarrow{S} A_2,\ B_2 \xrightarrow{T} B_2$$

Iz toga je jasno da je $T = T^{-1} = I$. Nadalje odmah se dobije

$$S^{-1} = \begin{bmatrix} x_{max} - x_{min} & 0 \\ 0 & y_{max} - y_{min} \end{bmatrix}$$

pa je

$$S = \begin{bmatrix} \frac{1}{x_{max} - x_{min}} & 0 \\ 0 & \frac{1}{y_{max} - y_{min}} \end{bmatrix}.$$

Sada iz $K_{(A_2,B_2)} = T^{-1} K_{(A_1,\,B_2)} S$ slijedi

$$K_{(A_2,B_2)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} w & 0 \\ 0 & -h \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{x_{max} - x_{min}} & 0 \\ 0 & \frac{1}{y_{max} - y_{min}} \end{bmatrix}$$

odnosno nakon množenja

$$K_{(A_2,B_2)} = \begin{bmatrix} \frac{w}{x_{max} - x_{min}} & 0 \\ 0 & \frac{h}{y_{min} - y_{max}} \end{bmatrix}.$$

Uvedemo li oznake

$$s_x = \frac{w}{x_{max} - x_{min}}\,,\ s_y = \frac{h}{y_{min} - y_{max}}$$

Možemo kratko pisati

$$K=\begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}.$$

Sada još želimo da se točka (x_{min},y_{max}) iz prirodnog koordinatnog sustava preslika u točku na ekranu s koordinatama (0,0). Zapravo tražimo afino preslikavanje u obliku

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

pri čemu treba odrediti vektor translacije (p_x, p_y). Iz

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x_{min} \\ y_{max} \end{bmatrix} + \begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

dobivamo

$$p_x = -s_x \cdot x_{min}, \;\; p_y = -s_y \cdot y_{max}$$

Jedinična dužina na x-osi iz prirodnog koordinatnog sustava ima duljinu $\lfloor s_x \rfloor$ piksela na monitoru.

Jedinična dužina na y-osi iz prirodnog koordinatnog sustava ima duljinu piksela $\lfloor -s_y \rfloor$ na monitoru.

Ishodište prirodnog koordinatnog sustava preslika se u točku (piksel) na monitoru koja ima koordinate ($\lfloor p_x \rfloor$, $\lfloor p_y \rfloor$)

Točka (x,y) iz prirodnog koordinatnog sustava se preko

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} p_x \\ p_y \end{bmatrix}$$

preslikava u piksel s koordinatama ($\lfloor x' \rfloor$, $\lfloor y' \rfloor$). [4.]

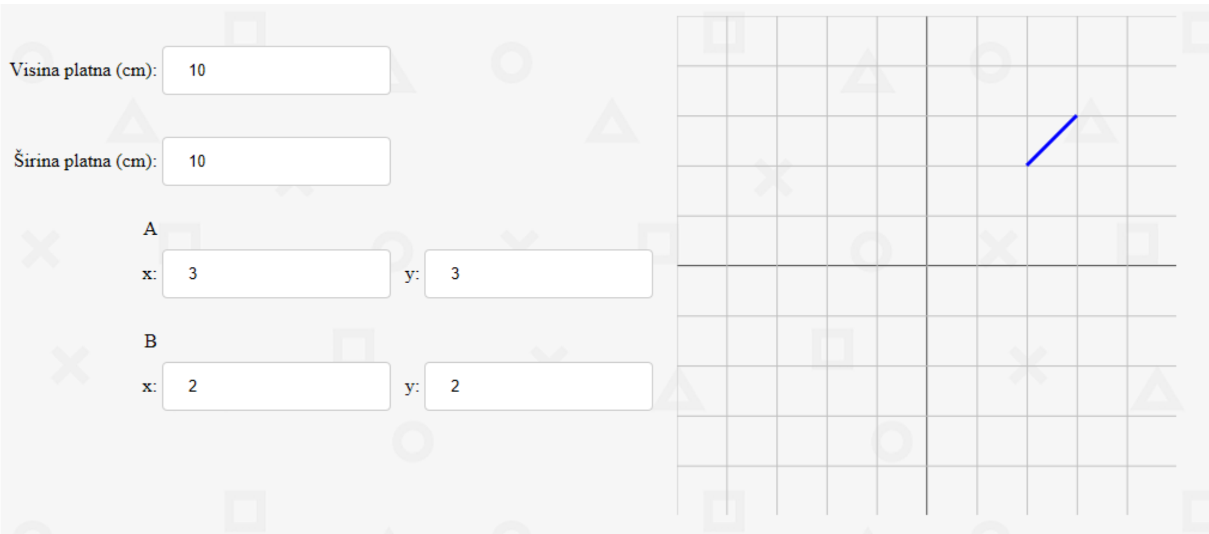
Zadatak 3.

U programskom jeziku javascript implementirajte klasu PKS (prirodni koordinatni sustav) koja će omogućiti crtanje ravnih linija koje su zadane u prirodnim koordinatama. Drugim riječima, dužinu zadanu koordinatama krajeva u prirodnim koordinatama treba nacrtati u canvas elementu. Konstruktoru morate zadati sljedeće parametre: raspon koordinata u prirodnom koordinatnom sustavu i dimenzije canvas elementa. Na temelju tih parametara treba prirodni koordinatni sustav preslikati na zadani canvas element.

Ako želimo nacrtati neku krivulju, tada moramo u domeni uzeti dovoljno točaka i susjedne točke samo spajamo ravnim linijama.

Kada zadamo našem canvas elementu visinu i dužinu, crtamo koordinatne osi. X-os crtamo od (širina/2, 0) do (širina/2, visina). Y-os crtamo od (0, visina/2) do (širina, visina/2). Pošto je u canvas elementu točka (0,0) smještena u gornjem lijevom kutu, mi ju moramo pomaknuti (virtualno) u sredinu canvas elementa (širina/2, visina/2). Dužine se crtaju tako da se iz html input elementa učitaju vrijednosti točke A (ax, ay) i B (bx, by), pretvori iz cm u px (1cm je odprilike 41.79 px) i povuče ravna crta između te dvije točke.

Slika rješenja



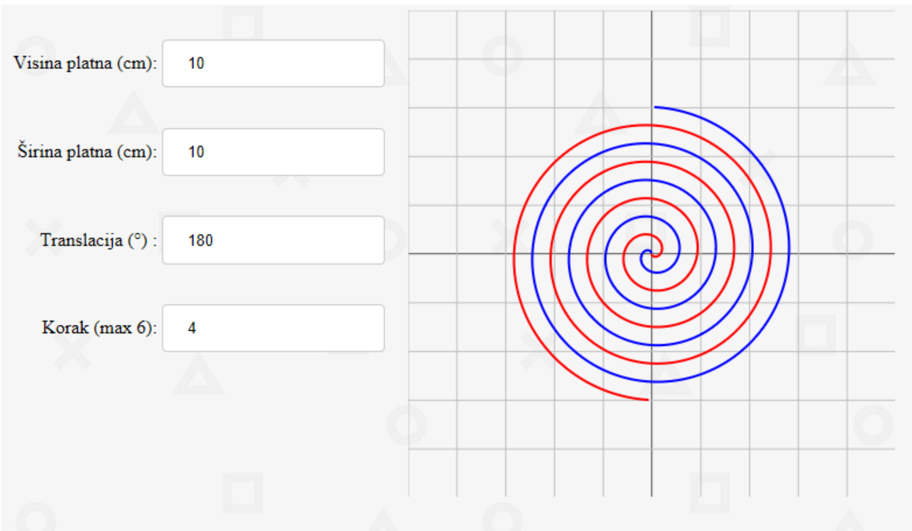
Slika 3: Crtanje linija

Zadatak 4.

U canvas elementu nacrtajte Fermatovu spiralu koja je zadana jednadžbom r2 = θ u polarnim koordinatama. Pronadite pripadne parametarske jednadžbe i pomoću ranije implementirane klase nacrtajte zadanu krivulju. Dio krivulje za negativne vrijednosti parametara crtajte crvenom bojom, a dio krivulje za pozitivne vrijednosti parametara crtajte plavom bojom. Takoder, pomoću tipki omogućite animaciju crtanja te krivulje od početne vrijednosti parametra do završne vrijednosti korak po korak.

Kontekst canvas elementa (spirala) prebacuje se u središte canvas elementa. Spirala se crta na sljedeći način: zadamo koliko krugova želimo da spirala napravi, postavimo da je kut između krakova spirale=0 i počinjemo put crtanja. Prema formuli za x i y crtamo liniju na poziciju (x,y) i to radimo tako dugo dok nam je kut < broja krugova. Dio krivulje koji je pozitivan bojamo plavom bojom, a dio za negativne crvenom.

Slika rješenja



Slika 4: Fermatova spirala

Zadatak 5.

Opišite kako izgledaju matrice sljedećih transformacija u homogenim koordinatama: translacija, rotacija oko ishodišta i skaliranje. Objasnite kako se određuju koordinate transformirane točke ukoliko je transformacija zadana svojom matricom u homogenim koordinatama.

Homogene koordinate pojavile su se početkom 19.stoljeća, a definiraju točku u ravnini koja koristi tri koordinate umjesto dvije. Primjerice za točku P s koordinatama (x,y) postoji homogena točka (x, y, t) tako da X = x / t i Y = y / t. Na primjer, točka (3, 4) ima homogene koordinate (6, 8, 2), jer 3 = 6/2 i 4 = 8/2. Ali homogena točka (6, 8, 2) nije jedinstvena za (3,4) zato što su: (12, 16, 4), (15, 20, 5) i (300, 400, 100) također moguće homogene koordinate za (3, 4). Razlog zbog kojeg se koordinatni sustav naziva "homogenim" je to što je moguće transformirati funkcije kao što su f (x, y) u formu f (x / t, y / t) bez ometanja stupnja krivulje.

TRANSLACIJA

Algebarska i matrična notacija za 2D translaciju je

$$x' = x + t_x$$

$$y' = y + t_y$$

ili koristeći matrice

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

SKALIRANJE

Algebarska i matrična notacija za 2D skaliranje je

$$x' = s_x x$$

$$y' = s_y y$$

ili koristeći matrice

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Skaliranje je relativno u odnosu na ishodište, točka (0,0) ostaje (0,0) dok se sve ostale točke pomiču od ishodišta. Algebarski to je

$$x' = s_x(x - p_x) + p_x$$

$$y' = s_y(y - p_y) + p_y$$

pojednostavljeno

$$x' = s_x x + p_x(1 - s_x)$$

$$y^{'} = s_y y + p_y(1 - s_y)$$

ili u homogenom obliku matrice

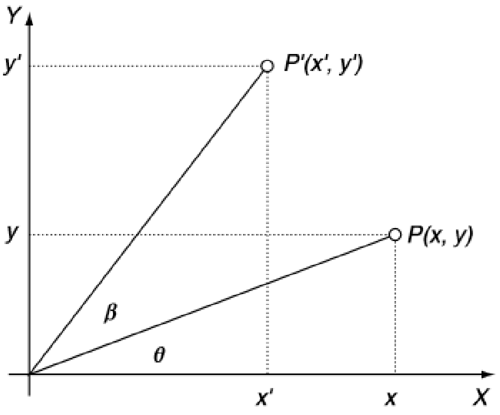
$$\begin{bmatrix} x^{'} \\ y^{'} \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & p_x(1 - s_x) \\ 0 & s_y & p_y(1 - s_y) \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Negativan predznak $-s_x$ (ili $-s_y$) daje zrcaljenje oko koordinatnih osi x (ili y).

ROTACIJA OKO ISHODIŠTA

Slika 5. prikazuje točku P(x,y) koju treba rotirati za kut β od ishodišta do P´(x´,y´). Što možemo zapisti pomoću jednadžbi kao: $x' = Rcos(\theta + \beta)$

$$y' = Rsin(\theta + \beta)$$



Slika 5: Rotacija

stoga

$$x' = R(\cos(\theta)\cos(\beta) - \sin(\theta)\sin(\beta))$$

$$y' = R(\sin(\theta)\cos(\beta) - \cos(\theta)\sin(\beta))$$

$$x' = R(\frac{x}{R}\cos(\beta) - \frac{y}{R}\sin(\beta))$$

$$y' = R(\frac{y}{R}\cos(\beta) + \frac{x}{R}\sin(\beta))$$

$$x' = xcos(\beta) - ysin(\beta)$$

$$y' = xsin(\beta) + ycos(\beta)$$

ili u matričnom zapisu

$$\begin{bmatrix} x^{'} \\ y^{'} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 \\ \sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Da bi rotirali točku (x,y) oko proizvoljne točke (p_x, p_y), prvo je potrebno oduzeti koordinate (p_x, p_y) od (x,y). To nam omogućuje da izvršimo rotaciju oko ishodišta. Drugo je da izvršimo rotaciju i treći korak je da dodamo (p_x, p_y).

1) Oduzimamo (p_x, p_y):

$$x_1 = (x - p_x)$$
$$y_1 = (y - p_y)$$

2) Rotiramo β oko ishodišta:

$$x_2 = (x - p_x)\cos(\beta) - (y - p_y)\sin(\beta)$$
$$y_2 = (x - p_x)\sin(\beta) + (y - p_y)\cos(\beta)$$

3) Dodamo (p_x, p_y):

$$x' = (x - p_x)\cos(\beta) - \left(y - p_y\right)\sin(\beta) + p_x)$$
$$y_2 = (x - p_x)\sin(\beta) + \left(y - p_y\right)\cos(\beta) + p_y)$$

pojednostavljeno,

$$x' = xcos(\beta) - ysin(\beta) + p_x(1 - \cos(\beta)) + p_y\sin(\beta)$$
$$y' = x\sin(\beta) + ycos(\beta) + p_y(1 - \cos(\beta)) - p_x\sin(\beta)$$

i zapis u matričnoj formi:

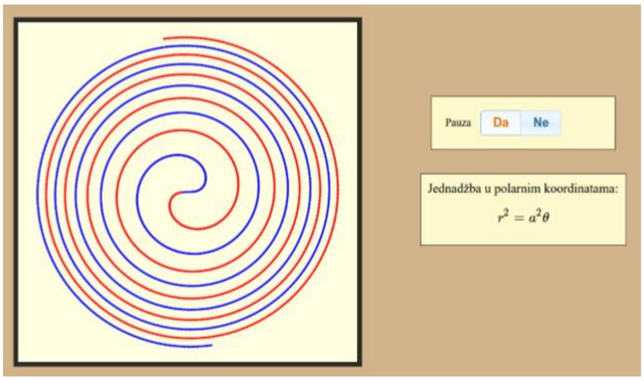
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & p_x(1 - \cos(\beta)) + p_y\sin(\beta) \\ \sin(\beta) & \cos(\beta) & p_y(1 - \cos(\beta)) - p_x\sin(\beta) \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

[2] (57.-65.str.)

Zadatak 6.

Ranije implementiranoj klasi PKS u zadatku 3 omogućite da se prije crtanja objekta na njega primijeni neka transformacija, a tek nakon toga nacrta transformirani objekt. Ukoliko na objekt ne želimo primjenjivati transformaciju, stavimo da je transformacija jednaka identiteti, tj. jediničnoj matrici u homogenim koordinatama.

Riješenje ovog zadatka nalazi se u klasi PKS.



Slika 6: Fermatova spirala

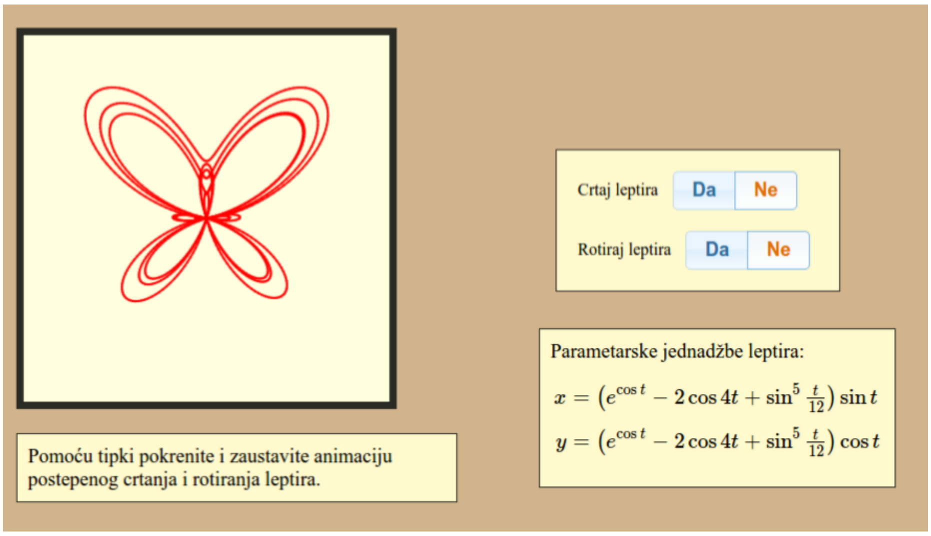
Zadatak 7.

U canvas elementu nacrtajte leptira koji je zadan parametarskim jednadžbama

$$x = \left(e^{\cos t} - 2cos4t + sin^5 \frac{t}{12}\right) \sin t$$

$$y = (e^{\cos t} - 2cos4t + sin^5 \frac{t}{12}) \cos t$$

pri čemu je t ∈ [0,20.5].Takoder, pomoću tipki omogućite postepeno crtanje leptira. Isto tako, omogućite pomoću tipki rotiranje leptira oko ishodišta. Korisnik u svakom trenutku može zaustaviti ili pokrenuti rotiranje leptira kao i postepeno crtanje leptira. Obje radnje se mogu i istovremeno obavljati ili neka od njih može biti isključena. Sve kombinacije moraju biti omogućene na izbor.

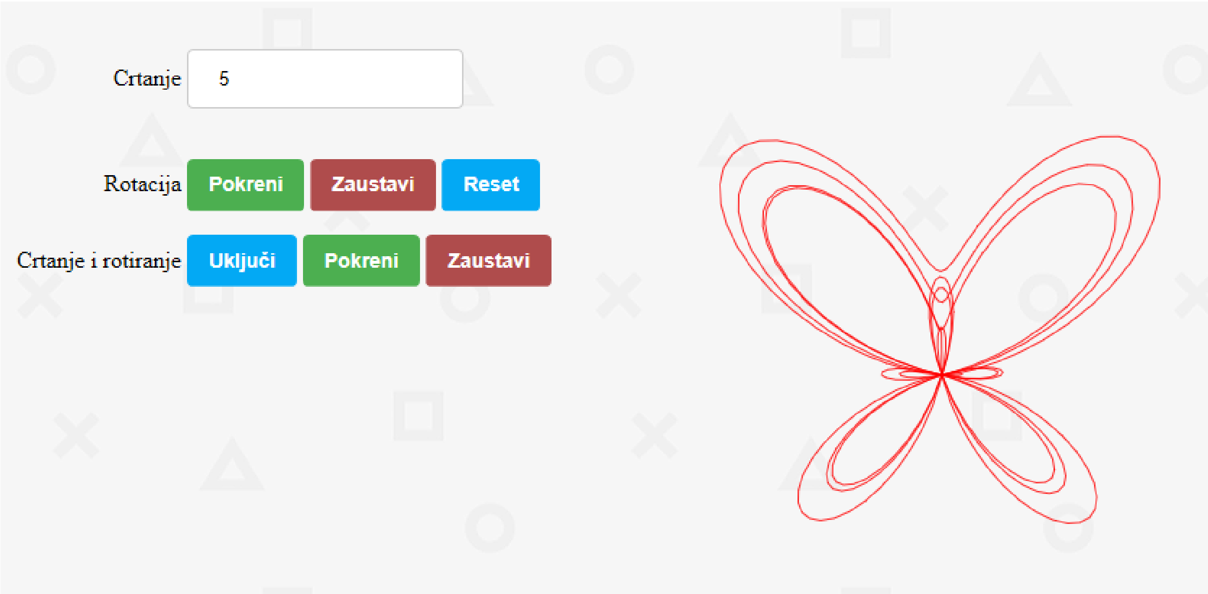


Slika 7: Leptir

Leptir se crta na sličini način kao i spirala. Počinjemo put crtanja na način da postavljamo t=0 i kod svakog završetka petlje ga povećavamo za 0.05, sve dok nam je t < 20.5 crtamo novu liniju na poziciju (x, y).

Leptira rotiramo tako da prvo spremimo kontekst za canvas element, obrišemo sve što imamo u canvas elementu, rotiramo kontekst koji smo spremili i ponovno nacrtamo spremljeni kontekst.

Slika rješenja



Slika 8: Leptir- naše rješeje

Zadatak 8.

Objasnite na koji način je povezana kompozicija linearnih operatora s matričnim množenjem. Implementirajte u programskom jeziku javascript klasu MAT2D koja će generirati matrice sljedećih transformacija ravnine u homogenim koordinatama: translacija, rotacija oko ishodišta, rotacija oko proizvoljne točke, skaliranje. Nadalje, toj klasi omogućite da radi kompoziciju tih transformacija. Objasnite na koji način rotaciju oko proizvoljne točke možete realizirati pomoću "jednostavnijih" transformacija.

Sve transformacije su opisane u zataku 5, ali postoji mala razlika kod translacije za računalnu grafiku. Matrica translacije izgleda ovako:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ d_x & d_y & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Razlog takve matrice je taj da množenjem matrica pomoću pomnoži dobimo stvarne vrijednosti i ispravne vrijednosti množenja matrica.

Rotacija oko ishodišta zadana je

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 \\ \sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

koristeći matrice možemo razviti rotaciju oko proizvoljne točke (p_x, p_y):

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & p_x \\ 0 & 1 & p_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 \\ \sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & -p_x \\ 0 & 1 & -p_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Sada je potrebno matrice međusobno pomnožiti, prvo ćemo pomnožiti matricu koja se odnosi na rotaciju β s matricom ($-p_x$, $-p_y$):

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & p_x \\ 0 & 1 & p_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos(\beta) & -\sin(\beta) & -p_x\cos(\beta) + p_y\sin(\beta) \\ \sin(\beta) & \cos(\beta) & -p_x\sin(\beta) + p_y\cos(\beta) \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

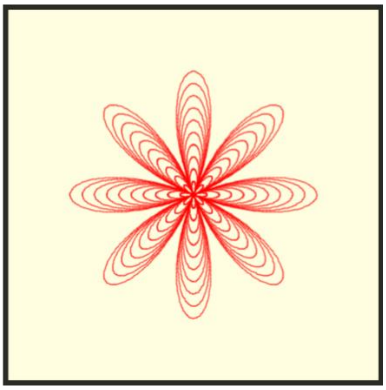
i konačno

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\beta) & -\sin(\beta) & p_x(1-\cos(\beta)) + p_y\sin(\beta) \\ \sin(\beta) & \cos(\beta) & p_y(1-\cos(\beta)) - p_x\sin(\beta) \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

što je isto kao i rotacija oko ishodišta. Što nas dovodi do zaključka da se sve vrste transformacija mogu izvesti ili algebarski ili pomoću matrica.

Zadatak 9.

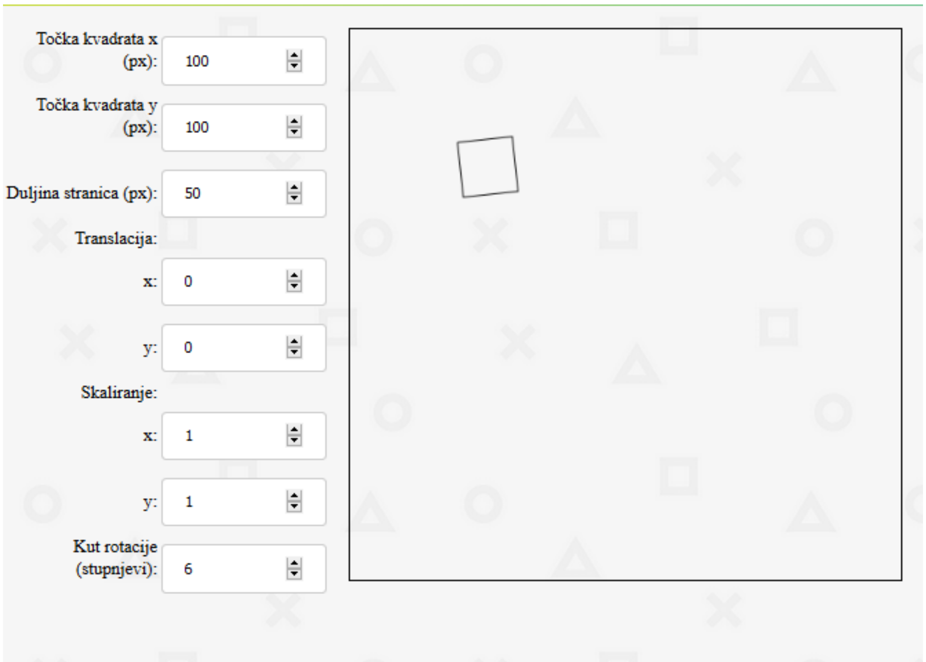
Omogućite klasi PKS preuzimanje matrice transformacije koju je stvorila klasa MAT2D, a prije crtanja samog objekta ta transformacija se primjenjuje na zadani objekt pa se zapravo crta pripadni transformirani objekt. Ideja je da se klasa PKS brine za crtanje objekata zadanih u prirodnim koordinatama, a klasa MAT2D pomaže klasi PKS ukoliko treba transformirati zadani objekt prije samog crtanja. Pretpostavimo da je zadan kvadrat sa svojim vrhovima u prirodnim koordinatama. Medutim, prije crtanja tog kvadrata želimo ga zarotirati oko ishodišta za neki kut i nakon toga pomaknuti za neki vektor. Tada će klasa MAT2D najprije kreirati matrice tih transformacija i napraviti njihovu kompoziciju, a klasa PKS će preuzeti tu gotovu matricu i primijeniti ju na zadani kvadrat nakon čega će nacrtati transformirani kvadrat.



Slika 9: Cvijet

Za realizaciju crtanja kvadrata koristimo funkciju kvadrat iz „zadaci.js“ koja kao ulazne parametre ima točku(x, y) od koje želimo crtati i parametar a kojim određujemo koliko će biti duge stranice. Kad želimo objekt rotirati pozivamo funkciju rotacija koja se nalazi u klasi MAT2D.js. Kad se pozove rotacija ona za ulazne parametre ima sljedeće parametre (kut, centar, točke[]), kut nam je zadan u stupnjevima pa prije same rotacije preračunavamo kut u radiane, centar nam predstavlja središnju točku kvadrata i točke je matrica točke (x,y). Funkcija rotacija prosljeđuje matrice rotacije i točke oko koje se treba izvršiti rotacija. Ako je kut negativni brojevi rotacija se izvršava u smjeru kazaljke na satu, a ako je kut pozitivni brojevi rotacija se izvršava obrnuto od kazaljke na satu.

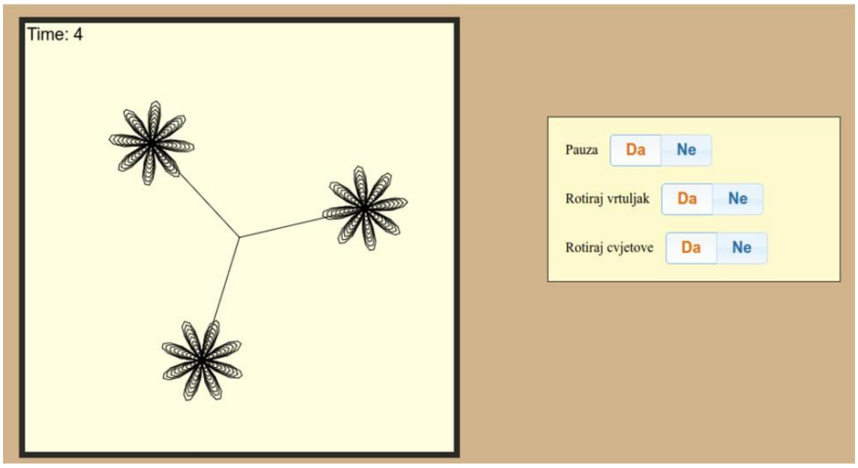
Slika rješenja



Slika 10: Naše riješenje zadatka 9

Zadatak 10.

U canvas elementu nacrtajte cvijet kako je prikazano na slici 9. Cvijet je zadan svojom jednadžbom u polarnim koordinatama r = a cos 4θ. Pritom je θ ∈ [0,2π], dok je a > 0 neki zadani realni broj. Na taj način dobit ćete jedan obris cvijeta. Više obrisa dobit ćete mijenjanjem vrijednosti parametra a.



Slika 11: Vrtuljak

Za realizaciju ovog zadatka potrebno je bilo prvo pretvorit jednadžbu u polarnim kordinatama u jednadžbe u kartezijevim kordinatama. Pretvorba iz polarne jednadžbe u kordinatnu jednadžbu ide prema sljedećim formulama:

$$x = r * cos\theta$$

$$y = r * sin\theta$$

Nakon što uvrstimo u formulu dobijemo sljedeće formule:

$$k = 4$$

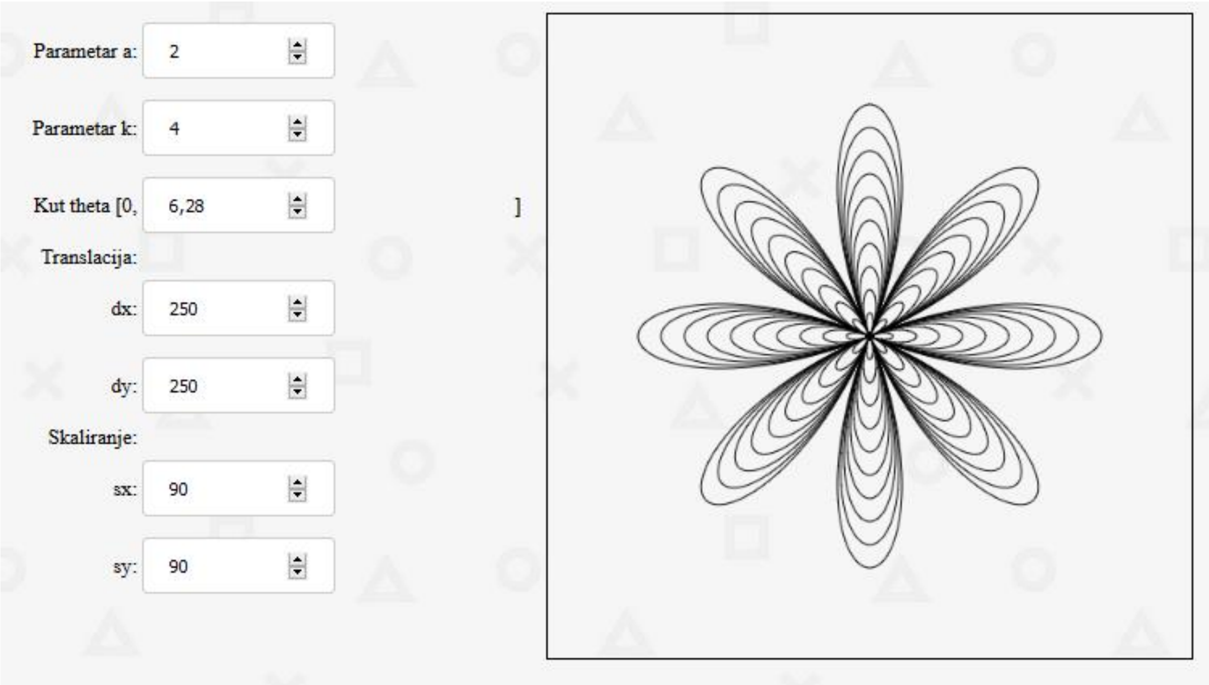
$$r = a * cosk\theta$$

$$x = a * (\cos(k\theta) * \cos(\theta))$$

$$y = a * (\cos(k\theta) * \sin(\theta))$$

Nakon što dobijemo točku nadalje pozivamo funkcije iz MAT2D.js i na temelju dobivenih točaka funkcija nacrtaj iz klase PKS1.js crta dobivene točke. Parametar a se mjenja od nulte vrijednosti pa sve do vrijednosti koju smo mi odredili za 0,2.

Slika rješenja

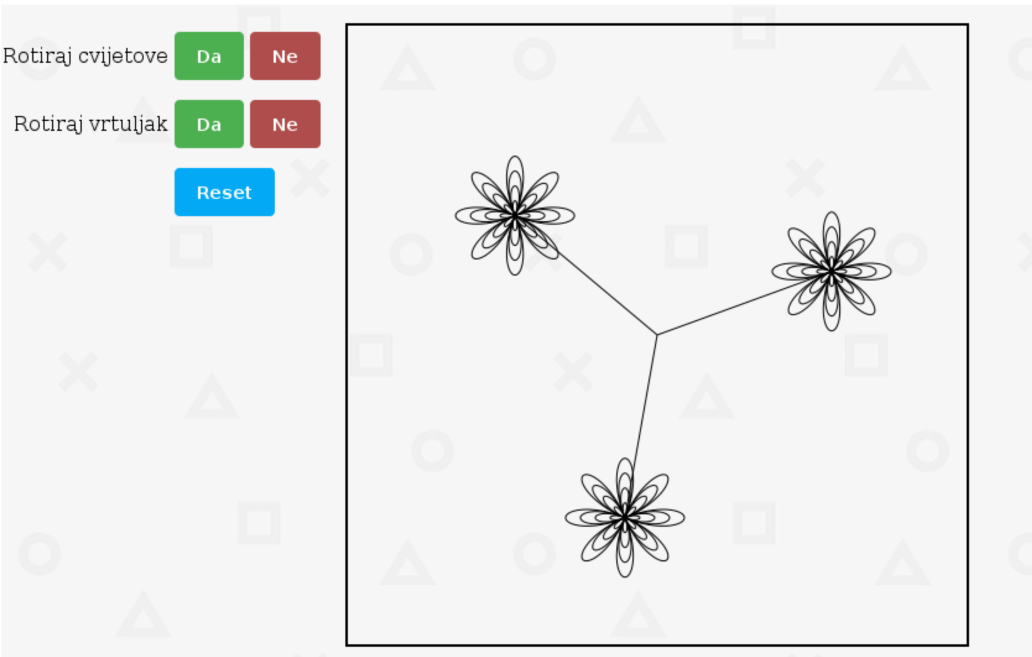


Slika 12: Riješenje zadatka 10.

Zadatak 11.

Napravite vrtuljak koji se sastoji od cvijetova iz prošlog zadatka kako je prikazano na slici 4. Vrtuljak se sastoji od tri cvijeta i tri dužine. Omogućite rotaciju vrtuljka oko točke u kojoj su spojene tri dužine. Takoder, omogućite rotaciju svakog cvijeta oko njegovog središta. Rotaciju vrtuljka i rotacije cvjetova korisnik mora moći uključiti ili isključiti pomoću odgovarajuće tipke. Sve kombinacije su moguće: rotira se samo vrtuljak, rotiraju se samo cvjetovi, rotiraju se istovremeno vrtuljak i cvjetovi. U ovom primjeru vidjet ćete svu snagu i moć implementiranih klasa PKS i MAT2D. Svaka od njih radi svoj posao, a zajedno čine čuda na ekranu.

Slika rješenja



Slika 13: Riješenje zadatka 11.

Literatura

1. Pilgrim M. (2010). HTML5 Spreman za upotrebu. O'Reilly.
2. John A. Vince (2010.) Mathematics for Computer Graphics
3. https://www.w3schools.com/tags/canvas_lineto.asp [Dostupno: 30.05.2017]
4. https://drive.google.com/open?id=0BxMHVDriBmrnbVNOcjd1WjVXdEU [Dostupno: 30.05.2017]

Literatura koja nam je pomogla kod razumijevanja teme

a) http://matt.might.net/articles/rendering-mathematical-functions-in-javascript-with-canvas-html/ [Dostupno: 30.05.2017]

b) https://stackoverflow.com/questions/6824391/drawing-a-spiral-on-an-html-canvas-using-javascript [Dostupno: 30.05.2017]

c) http://jwilson.coe.uga.edu/EMAT6680Su12/Carreras/HW_11/HW_11.html [Dostupno: 29.05.2017]

d) https://socratic.org/questions/what-is-the-graph-of-r-a-cos-4theta [Dostupno: 30.05.2017]

e) https://math.stackexchange.com/questions/1930325/polar-to-cartesian-form-of-r-sin4%CE%B8 [Dostupno: 31.05.2017]

f) http://blog.wolframalpha.com/2011/08/03/generating-polar-and-parametric-plots-in-wolframalpha/ [Dostupno: 29.05.2017]

g) https://krazydad.com/tutorials/circles/ [Dostupno: 31.05.2017]