

`Canvas` HTML element ima svoj koordinatni sustav s cjelobrojnim koordinatama. Međutim, ukoliko želimo nacrtati krivulje zadane u prirodnim koordinatama, moramo prirodni koordinatni sustav preslikati u koordinatni sustav `canvas` elementa.

Zadatak 1

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

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 koordinatnom sustavu s jediničnim dužinama u koordinatnom sustavu `canvas` elementa.

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.

Zadatak 4

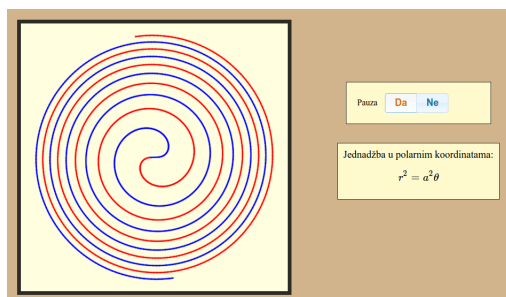
U `canvas` elementu nacrtajte Fermatovu spiralu koja je zadana jednačinom $r^2 = \theta$ u polarnim koordinatama. Pronađite pripadne parametarske jednačine i pomoću ranije implementirane klase nacrtajte zadanu krivulju. Dio krivulje za negativne vrijednosti parametara crtajte crvenom bojom, a dio krivulje za pozitivne vrijednosti parametara crtajte plavom bojom. Također, pomoću tipki omogućite animaciju crtanja te krivulje od početne vrijednosti parametra do završne vrijednosti korak po korak.

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.

Zadatak 6

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



Slika 1: Fermatova spirala

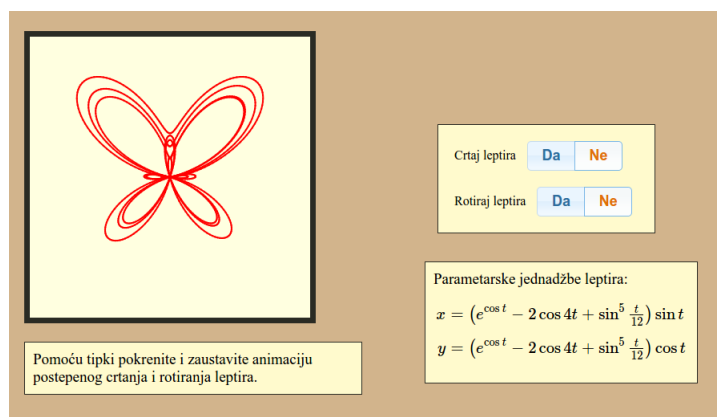
Zadatak 7

U `canvas` elementu nacrtajte leptira koji je zadan parametarskim jednađbama

$$x = \left(e^{\cos t} - 2 \cos 4t + \sin^5 \frac{t}{12} \right) \sin t$$

$$y = \left(e^{\cos t} - 2 \cos 4t + \sin^5 \frac{t}{12} \right) \cos t$$

pri čemu je $t \in [0, 20.5]$. Također, 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.



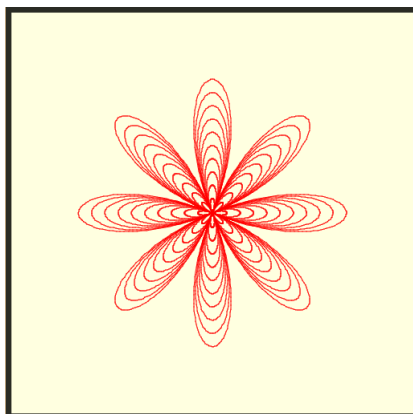
Slika 2: Leptir

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.

Zadatak 9

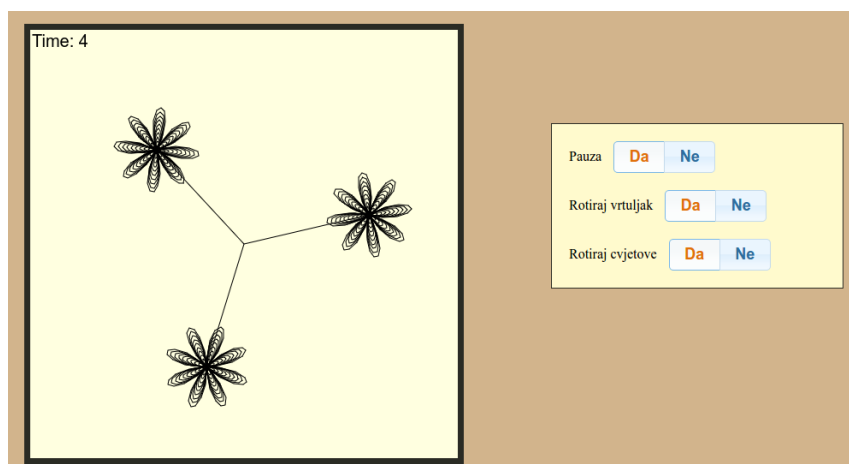
Omogućite klasi *PKS* preuzimanje matrice transformacije koju je stvorila klasa *MAT2D*, a prije crtanja samog objekta ta transformacija se primjenjuje na zadani objekt pa se zapravo crta pripadni transformirani objekt. Ideja je da se klasa *PKS* brine za crtanje objekata zadanih u prirodnim koordinatama, 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. Međutim, 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 3: Cvijet

Zadatak 10

U **canvas** elementu nacrtajte cvijet kako je prikazano na slici 3. Cvijet je zadan svojom jednadžbom u polarnim koordinatama $r = a \cos 4\theta$. Pritom je $\theta \in [0, 2\pi]$, 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 4: Vrtuljak

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. Također, 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.