

Sveučilište u Zagrebu
Prirodoslovno-matematički fakultet
Matematički odsjek

MMS Paint

Ivka Čačić, Lovorka Matejaš, Anton Rosanda, Brigita Rusan, Ela
Žuvanić

Završni projekt iz kolegija
Multimedijski sustavi



Matematički odsjek Prirodoslovno-matematički fakultet
Sveučilište u Zagrebu
ožujak 2024.

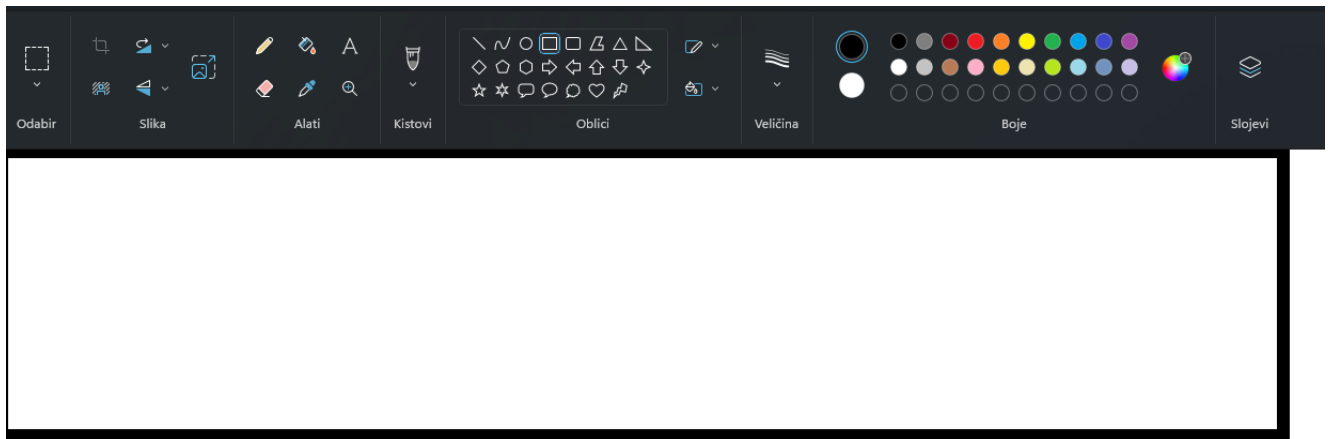
Sadržaj

1	Uvod	2
2	Grafički dio	3
3	Opis funkcija	5
4	Moguća poboljšanja	6
5	Literatura	7

1 Uvod

Odlučili smo primijeniti znanje stečeno na kolegiju Multimedijски sustavi kako bismo stvorili vlastitu inačicu popularnog programa Paint. Cilj nam je bio kreirati našu verziju ovog jednostavnog grafičkog alata koristeći alate i tehnike koje smo naučili na spomenutom kolegiju te ih integrirati u okruženje Processing.

Aplikacija bit će slična MS Paintu, omogućujući korisnicima crtanje, dodavanje oblika, dodavanje teksta, dodavanje slika, mijenjanje njihove veličine, bojanje i slične funkcionalnosti. Uvest ćemo mogućnost učitavanja prethodnih crteža, stvaranje novih, te eksportiranje i spremanje crteža na računalo. Dodatno, korisnicima će biti ponuđeno dodavanje slika koje su izrađene u programima poput Inkscape-a, čime ćemo postići sličnu razinu funkcionalnosti kao u Canvi. Također, korisnicima će biti ponuđene teme koje će ih inspirirati za kreiranje vlastitih crteža. Naši glavni izazovi u ovom projektu su definiranje estetskog izgleda programa i implementacija svih ključnih funkcionalnosti.



Slika 1: Prikaz MS Painta i željenih funkcionalnosti

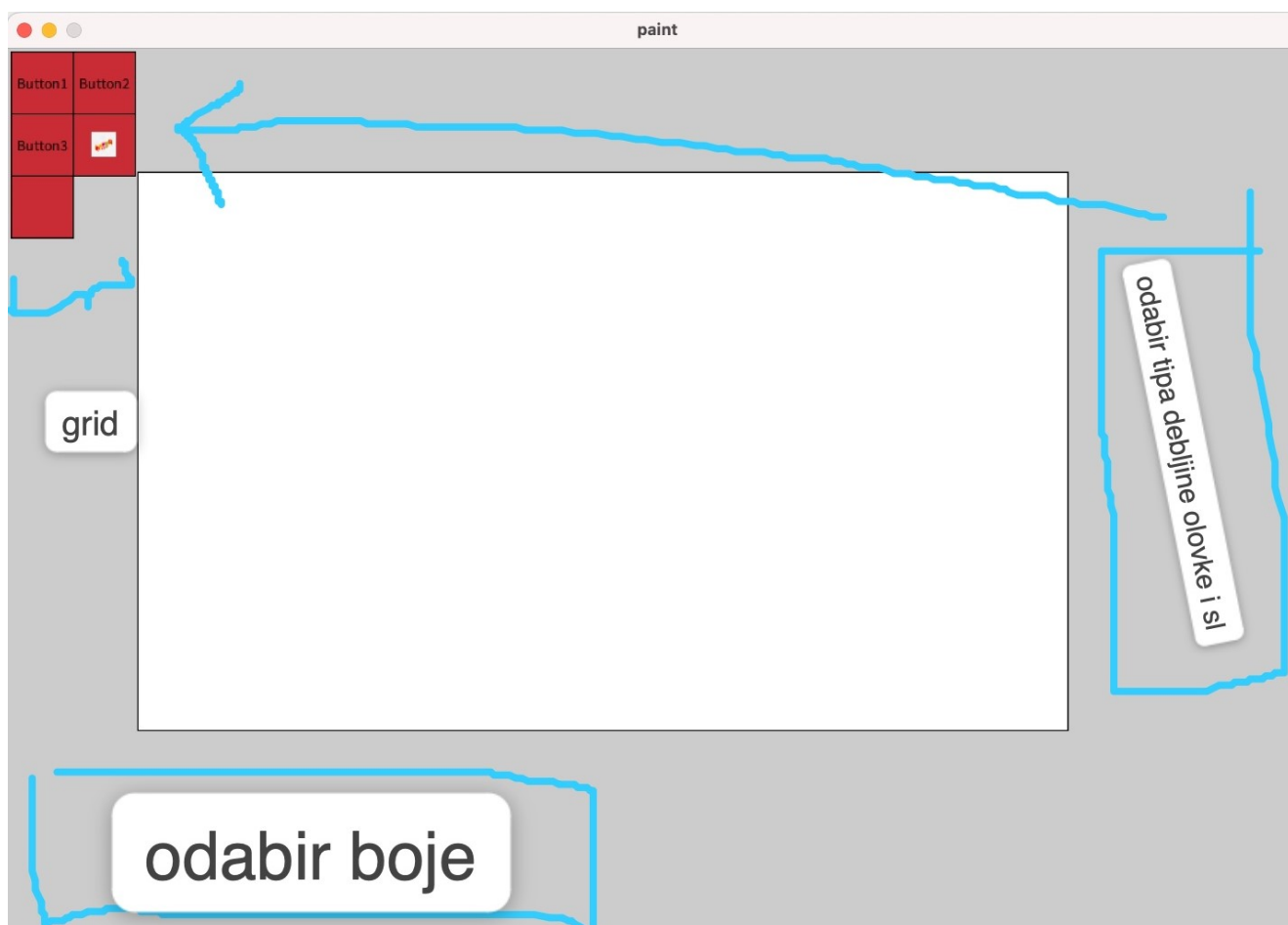
2 Grafički dio

U procesu razvoja grafičke aplikacije slične Paintu, jedan od ključnih izazova bio je definiranje njenog grafičkog sučelja. Morali smo odlučiti kako će točno aplikacija izgledati i kako će korisnici interaktivno koristiti njezine funkcionalnosti. Odlučili smo se za strukturu sučelja koja je intuitivna i funkcionalna, nudeći korisnicima jednostavan pristup svim potrebnim alatima i opcijama.

S lijeve strane ekrana planirali smo postaviti gumbove koji bi, kada ih korisnik pritisne mišem, omogućili aktivaciju određenih funkcionalnosti. Među ovim funkcijama planirali smo uključiti alate poput olovke, gumice, te mogućnost odabira različitih oblika i fotografija. Ideja je bila omogućiti korisnicima brz i jednostavan pristup ovim alatima.

Na sredini ekrana zamislili smo bijelo platno koje predstavlja glavni prostor za korisničko stvaranje. Bijela pozadina osigurala bi optimalnu vidljivost za crtanje i lakše razlikovanje boja.

S desne strane ekrana planirali smo rezervirati prostor za odabir debljine olovke, pregled svih dostupnih oblika te galeriju fotografija. Ideja je bila omogućiti korisnicima fino podešavanje alata za crtanje, kao i jednostavan pristup različitim oblicima, fotografijama koje bi im bile korisne za integraciju u njihove crteže. Ispod platna za crtanje nalazile bi se boje.



Slika 2: Grafički prikaz projekta

Za olakšavanje organizacije i postavljanje gumba unutar grafičkog sučelja, razvili smo klasu nazvanu Grid. Ova klasa omogućuje postavljanje gumba na određene pozicije unutar sučelja, što olakšava korisnicima pronalaženje i korištenje funkcionalnosti aplikacije.

Konkretno, Grid klasa radi na principu definiranja određenog broja redaka i stupaca, tvoreći matricu čiji elementi predstavljaju gumbove s različitim funkcionalnostima. Na taj način, koristeći ovu klasu, možemo organizirati gumbove za odabir boja, gumbove za različite funkcionalnosti, gumbove za odabir oblika, gumbove za umetanje slika unutar aplikacije. Implementacija Grid klase omogućuje nam fleksibilnost u postavljanju i prilagodbi položaja gumba ovisno o potrebama korisničkog sučelja.

Korištenje opisane klase u organizaciji gumba unutar aplikacije omogućuje nam stvaranje intuitivnog i preglednog sučelja koje korisnicima omogućuje brzo pronalaženje i korištenje željenih funkcionalnosti.

Osim klase Grid, za grafički prikaz bitna je klasa Gumb koja će imati ključnu ulogu u stvaranju i upravljanju gumbovima unutar grafičkog sučelja naše aplikacije. Glavni cilj klase Gumb je omogućiti kreiranje gumba željene veličine, te će sadržavati tekstualni opis svoje namjene. Umjesto tekstualnog opisa, također je moguće dodati sliku unutar gumba kako bi se još jasnije prepoznala njegova namjena ili funkcija. Ova mogućnost omogućuje dodatnu fleksibilnost i prilagodbu sučelja aplikacije prema potrebama i preferencijama korisnika.

Gumbovi će biti proizvoljne boje, no za navigaciju korisnika kroz funkcije aplikacije odabrali smo crvenu boju s crnim slovima.

Kada se odredi pozicija za gumbove unutar sučelja, oni će ostati fiksni na toj poziciji, pružajući stabilnost i konzistentnost korisničkog iskustva.

Da bismo omogućili korisnicima da crtaju na bijelom platnu unutar naše aplikacije, razvili smo klasu nazvanu DrawArea. Glavni cilj ove klase je stvoriti područje za crtanje koje će biti bijele boje i fiksne veličine od 750 piksela u širinu i 450 piksela u visinu.

Klasa Pencil je odgovornu za različite alate za crtanje, kao što su olovka, gumica i čarobna olovka. Ova klasa omogućuje korisnicima da crtaju, brišu i primjenjuju različite efekte na platnu unutar naše aplikacije.

Klasa Text služi za stvaranje i prikazivanje tekstualnih objekata unutar aplikacije za crtanje. Sadrži attribute kao što su položaj, veličina teksta, tekstualni sadržaj i boja teksta.

Klasa nazvana Oblici, predstavlja sustav za crtanje različitih geometrijskih oblika unutar aplikacije.

Klasa TextBlox omogućuje korisnicima jednostavno dodavanje teksta na platno aplikacije. Koristeći različite boje, veličine i vrste fontova, korisnici mogu prilagoditi izgled teksta prema svojim potrebama. Također, moguće je stvoriti više polja za unos teksta na različitim pozicijama na platnu.

3 Opis funkcija

- **void setup()** izvršava se odmah pri pokretanju aplikacije. U ovoj metodi obavljam sve potrebne pripreme kako bi naše grafičko sučelje bilo spremno za korištenje. Konkretno, pozivamo različite funkcije koje dodaju gumbove, slidere, slike te postavljaju veličinu zaslona i font.
- **void draw()** je ključna funkcija unutar našeg programa koja se neprekidno izvršava i odgovorna je za crtanje grafičkih elemenata na ekranu. U ovoj funkciji definiramo sve vizualne elemente i interakcije koje se događaju unutar naše aplikacije. Postavljamo sivu boju pozadine te pozivamo funkciju `drawGrid()` iz klase `Grid` koje crta gumb unutar mreže na zaslonu.
- **void mouseClicked()** je funkcija koja reagira na korisnički klik mišem i pokreće odgovarajuće funkcije ovisno o tome gdje je korisnik kliknuo na ekranu. Provjeravamo je li korisnik kliknuo na gumb za alate pomoću funkcije `returnPressedButton()`. Ako je, prvo poništavamo prethodno odabrani gumb za alat (ako postoji) te ažuriramo trenutni odabrani gumb. Nakon toga, ažuriramo vizualni prikaz odabranog gumba te ispisujemo naziv slike povezane s tim gumbom.
- **void useSelectedTool()** je funkcija koja se poziva kako bi se primijenio odabrani alat za crtanje na platnu. Ova funkcija provjerava koji alat je odabran te prema tome izvršava odgovarajuću akciju.
- **void useSelectedShape()** je funkcija koja se koristi za primjenu odabranog oblika na platnu. Ova funkcija provjerava je li odabran neki od oblika te je li korisnik odabrao alat za crtanje oblika, nakon čega primjenjuje odgovarajući oblik na platnu.
- **void pen(int c, int w, DrawArea area)** je funkcija koja omogućuje crtanje olovkom.
- **void eraser(int c, int w, DrawArea area)** je funkcija koja omogućuje brisanje slike gumicom.
- **void magicPen(int w, DrawArea area)** je funkcija koja omogućuje crtanje prilikom kojeg dolazi do promjene boje.
- **void drawLine(PVector point1, PVector point2, DrawArea area)** služi za crtanje linije na platnu u okviru zadane površine.
- **void colorCan(color c, DrawArea area)** je funkcija koja služi za bojanje veće površine.
- **void drawShape(color boja, String oblikCrtaj, DrawArea area)** je funkcija koja služi za crtanje određenog oblika na zaslonu prema parametrima koje prima. Funkcija crta: pravokutnik, krug, srce, zvijezdu, trokut, romb, pravac, peterkut, zvijezdu, strelice.
- **void tBlob(color boja, String velicina, String font, DrawArea area)** provjerava je li je tekstualni okvir stvoren ili nije tj, je li pozicija miša unutar određenog područja na platnu. Ako je, poziva funkciju za stvaranje tekstualnog okvira.
- **void createTextField(color boja, String velicina, String font, int x, int y)** služi za stvaranje tekstualnog okvira na određenoj poziciji na platnu.

4 Moguća poboljšanja

1. Implementacija drag and drop funkcije
2. Dodavanje glazbe
 - pozadinske
 - zvuk određenog alata - npr. zvuk olovke pri odabigu magične olovke
 - napraviti vlastitu glazbu
3. dodavanje mogućnosti rada na više paralelnih crteža
 - klasa Tab čuva "screenshot" odrezanog djela za crtanje
4. Dodavanje mogućnosti biranja dimenzije datoteka
5. Omogućiti korisniku poništavanje posljednje radnje (vraćanje prethodnog stanja)
6. Napisati i implementirati funkciju za sprej u Paintu
7. Dodavanje tema, kao inspiracija za crteže
8. Poboljšanje izgleda oblika
9. Dodavanje mogućnosti premještanja teksta povlačenjem miša
10. Dodavanje više fontova, mogućnost biranja veličine fonta iz padajućeg izbornika
11. Dodavanje undo funkcionalnosti
12. Poboljšati čitljivost koda. Namještanje vrijednosti prebaciti u zasebne funkcije koje čine cjelinu
13. Popraviti greške unutar koda. Npr. import/export gumbi se ne prikazuju ukoliko je odabran neki od mnogo oblika za crtanje, istovremen odabir gumba za učitavanje slike te alata dovodi do toga da se naslika slika, ali i vrijednost alata(npr. pokušati sa sprejem)
14. Skaliranje SVG slika prije drag and drop
15. Omogućiti promjenu radijusa i gustoće alata spreja
16. Dodati ostale mogućnosti unutar Painta - rotaciju, Bezirove krivulje, pincete za boje, povećalo, crop, itd.
17. Dodati meni i mogućnosti mijenjanja tema (malo teže i treba se promijeniti dio koncepta logike)

5 Literatura

1. <https://processing.org/reference/>
2. <https://discourse.processing.org>
3. https://tabreturn.github.io/code/processing/python/2018/06/20/processing.py_in_ten_lessons-2.2-_vertices.html
4. http://pyde.org/Basics___Form___RegularPolygon/
5. <https://www.soundjay.com/aerosol-can-sounds-1.html>
6. <https://github.com/sojamo/controlp5/tree/master/examples/controllers>
7. <https://icons8.com/icons/set/import>