

SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET



Dokumentacija za završni projekt iz kolegija Multimedijски sustavi

MINOLOVAC

Ivan Knezić
Jakov Krunić

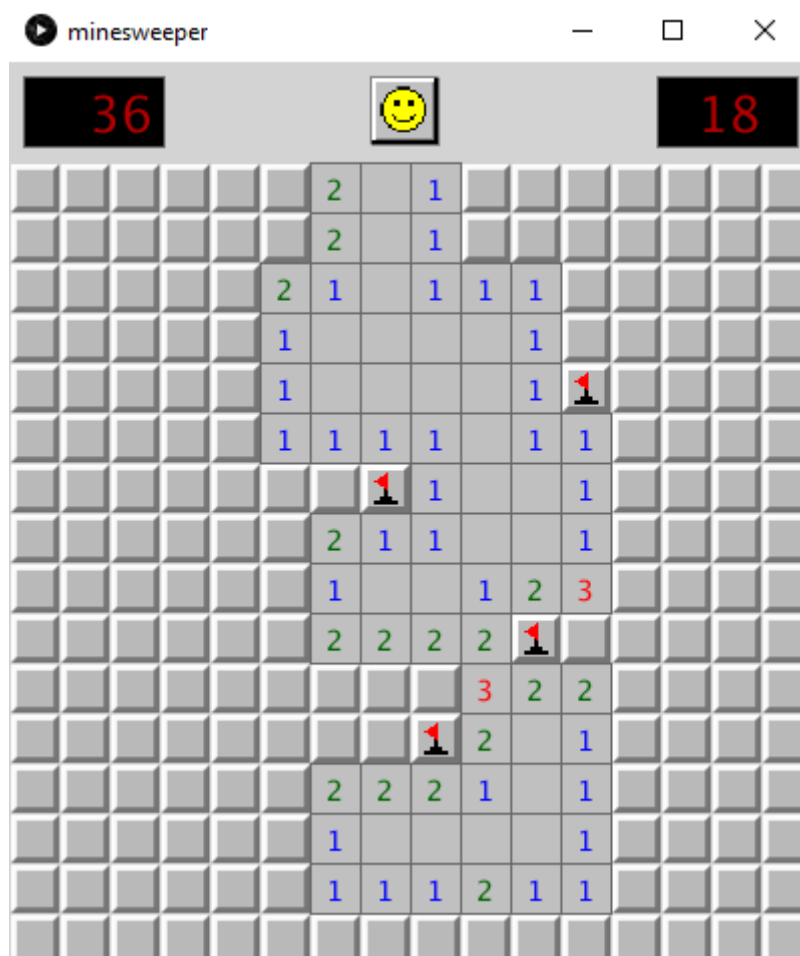
Zagreb, siječanj 2020. godine

Sadržaj

1. Opis programa.....	1
2. Grafički dio.....	2
3. Opis varijabli i funkcija.....	3
4. Moguća poboljšanja.....	6
5. Literatura.....	6

1. Opis programa

Napravili smo poznatu klasičnu igru Minolovac (Minesweeper) u Processingu. Minolovac je igra koju igra jedan igrač. Nakon pokretanja programa, prikaže se početni izbornik. U početnom izborniku, igrač može izabrati koju težinu igre (beginner, intermediate, expert) želi igrati. Osim toga, igrač može odabrati želi li igrati sa zvučnim efektima ili bez njih te može odabrati želi li igrati sa svjetlima ili bez (tj. dnevni ili noćni „mode“). Odabirom opcije New game, prikazuje se pravokutna ploča i započinje igra. Cilj igre je otvoriti sva polja koja ne sadrže mine. Ako je igrač uspio otvoriti sva polja koja ne sadrže mine, smatra se da je igra uspješno riješena. Svako polje (koje nije mina) sadrži informaciju o tome koliko ima susjednih mina. Lijevim klikom na polje koje nije mina, to polje se otvara i prikaže se broj u tom polju. Taj broj pomaže igraču tako što govori koliko ima mina među susjednih osam polja. Ako nema mina među susjednim poljima, otvara se polje u kojem ne piše broj te se automatski otvaraju sva susjedna polja. Ako igrač klikne na polje koje sadrži mina, igra je gotova i igrač automatski gubi te se prikaže efekt eksplozije mine. Igrač može označiti sa zastavicom polje za koje misli da sadrži minu.



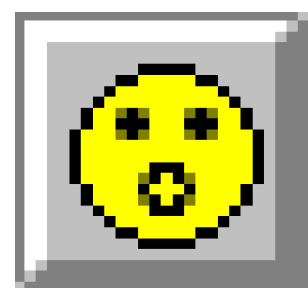
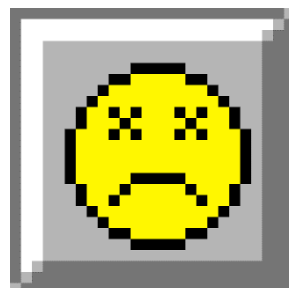
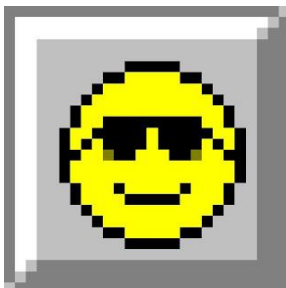
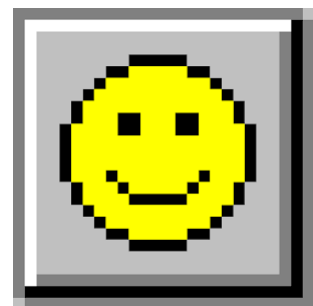
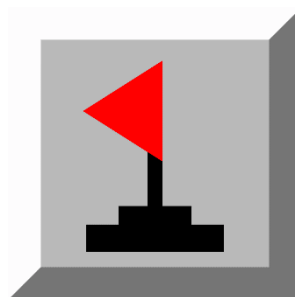
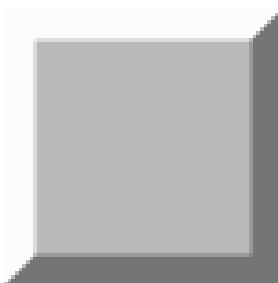
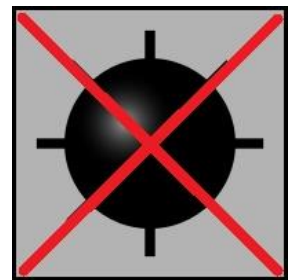
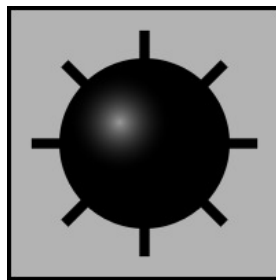
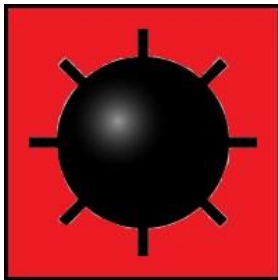
U svakom trenutku, igrač može isprazniti ploču i započeti novu igru (iste težine) klikom na žuti smajličić. U gornjem lijevom kutu se nalazi brojač mina, a u gornjem desnom kutu štoperica (koja se pokreće nakon prvog klika).

2. Grafički dio

Odlučili smo se za izgled igre kao u klasičnoj verziji (koja je bila dostupna za Windows XP). Ploču smo realizirali koristeći slike koje smo skinuli s interneta.

Slike koje smo koristili su :

- Neotvoreno polje - polje koje još nije kliknuto
- Neotvoreno polje sa zastavicom – polje na koje je odabran desni klik miša
- Crvena mina – otvara se prilikom „eksplozije“ tj. otvaranja polja koje sadrži minu
- Mina – u slučaju poraza, prikazuju se sve preostale mine
- Prekrižena mina – u slučaju poraza, prikazuje se na poljima gdje je bila krivo postavljena zastavica
- Smajlići – ovisno o fazi igre, pojavljuje se pojedini smajlic



Crvenu i prekriženu minu smo ručno uredili pomoću GIMP-a koristeći sliku obične mine.

3. Opis varijabli i funkcija

Program smo razdvojili u dva dijela: cell i minesweeper. U dijelu cell smo implementirali klasu *Cell* koja nam predstavlja jedno polje. Klasa *Cell* sadrži varijable koje opisuju položaj, veličinu, status ćelije (je li otvorena, sadrži li minu, je li postavljena zastavica) te broj susjednih mina (ako je to polje mina, postavimo taj broj na -1). Osim konstruktora, klasa *Cell* sadrži i sljedeće funkcije:

void drawCell()

Funkcija koja iscrtava polje ovisno o statusu.

boolean isChosen(int x, int y)

Funkcija koja provjerava nalazi li se kursor miša iznad tog polja.

boolean isNeighbour (int nx, int ny)

Funkcija koja provjerava je li polje s koordinatama nx i ny susjed tog polja.

ArrayList<Cell> getNeighbours ()

Funkcija koja vraća niz ćelija koje predstavljaju susjede tog polja.

Opišimo sada što se događa u glavnom dijelu programa. Koristili smo int varijablu *gameState* koja ovisno o stanju igre poprima sljedeće vrijednosti od kojih je svaka predefinirana:

1. *gameState* = izbornik – prvi ekran koji vidimo kada pokrenemo program. On predstavlja početni izbornik igre te sadrži opcije New Game, Options i Difficulty.
2. *gameState* = options – ekran u kojem se prikazuju opcije Sound on/off, Lights on/off te Back. Klikom na Sound on/off se pali i gasi zvuk. Klikom na Lights on/off se bira između dnevnog i noćnog načina igranja. Klikom na Back, vraćamo se na početni izbornik.
3. *gameState* = tezina – ekran u kojem se prikazuju opcije Begginer, Intermediate, Expert i Back. Klikom na jednu od opcija se podešava težina igre (broj mina te dimenzije ploče).
4. *gameState* = playing – prikazuje se ploča na kojoj se odvija igra
5. *gameState* = gameLost – prikazuje se ploča koja označava gubitak igre
6. *gameState* = gameWon – prikazuje se ploča koja označava pobjedu igrača

Nadalje, varijabla *velicinaPolja* predstavlja veličinu jednog polja (svako polje je kvadrat dimenzije *velicinaPolja*). *brojMina* predstavlja broj mina u igri, ovisno o težini. Defaultna vrijednost je 40 (kao u intermediate težini). *brojPreostalihMina* predstavlja broj mina umanjeno za broj postavljenih zastavica. *firstClick* pamti jesmo li odigrali prvi potez, po

defaultu je false. Varijable s prefiksom *clock* nam služe za mjerenje vremena od početka igre. Varijabla *difficulty* poprima vrijednosti od 1 do 3, ovisno o težini koja je odabrana, defaultna vrijednost je 2 (intermediate). Varijabla *grid* je dvodimenzionalno polje *Cell*-ova.

Opišimo sada ukratko sve funkcije korištene u glavnom dijelu programa.

void setup()

Postavljaju se dimenzije ekrana i inicijalizira se nova varijabla *grid* tipa *Cell[][]* koja predstavlja ploču.

void draw()

Ovisno o ekranu koji želimo prikazati, izvršava različite stvari. Varijabla kojom biramo što će se prikazati je *gameState*, koja je opisana već ranije.

void mousePressed()

Funkcija koja reagira na klik miša. Također ovisi o varijabli *gameState* na sljedeći način:

1. *gameState* = izbornik – ovisno o pravokutniku na koji se klikne, funkcija mijenja *gameState*
2. *gameState* = playing – ovisno gdje se klikne, funkcija izvršava različite stvari. Klikom na smajlič, igra se resetira. Ako je *firstclick*=false i kliknuta je neka ćelija lijevim klikom, ta ćelija se otvara i pozivaju se funkcije *set_mines(mouseX, mouseY)* i *set_numbers()*. Inače, provjeramo za svaku ćeliju je li kliknuta nekim klikom miša. Ako je kliknuta zatvorena ćelija (koja nije označena zastavicom) lijevim klikom, otvaramo je. Ako je to bila mina, igra je izgubljena. Ako je broj susjednih mina te ćelije jednak 0, automatski se otvaraju sva susjedna polja. Ako je kliknuta zatvorena ćelija (koja nije označena zastavicom) desnim klikom, postavlja se zastavica. Ako je kliknuta ćelija već označena zastavicom, zastavica se briše. Ako je kliknuta otvorena ćelija srednjim klikom, imamo 2 mogućnosti. Ako je broj susjednih zastavica jednak broju susjednih mina te ćelije (taj broj je vidljiv u igri jer je ćelija otvorena), onda se otvaraju svi zatvoreni susjedi (ako je neka zastavica bila netočno postavljena, igra se gubi). Inače, onda se zatvoreni susjedi istaknu čime se pomaže igraču.
3. *gameState* = gameLost ili gameWon – ako se klikne na smajlič, igra se resetira
4. *gameState* = options ili tezina – mijenjaju se postavke igre ovisno na koji je pravokutnik kliknuto

Također, ako se u nekom trenutku otvori polje bez broja (polje koje nema susjednih mina), automatski se otvaraju sva njegova susjedna polja (koja nisu označena zastavicom).

void open_neighbours(int x, int y)

Funkcija koja otvara sve susjede ćelije *grid[x][y]* koji nisu označeni zastavicom.

void highlight_closed_neighbours(int x, int y)

Funkcija koja istakne sve zatvorene susjede ćelije *grid[x][y]* koji nisu označeni zastavicom.

void highlight_back_closed_neighbours(int x, int y)

Funkcija koja vraća istaknute susjede ćelije *grid[x][y]* koji nisu označeni zastavicom na „normalno“ stanje, tj. obično zatvoreno polje.

void open_closed_neighbours(int x, int y)

Funkcija koja otvara sve susjede ćelije *grid[x][y]* koji nisu označeni zastavicom. Za razliku od funkcije *open_neighbours*, ova funkcija može otvoriti i minu. Ova funkcija se koristi prilikom klika srednje tipke miša.

void keyPressed()

Funkcija koja reagira na pritisak tipke 'q' na tipkovnici. U tom slučaju, vraćamo se na glavni izbornik i cijela igra se resetira.

int neighbour_flags(int x, int y)

Funkcija koja broji koliko susjednih ćelija od *grid[x][y]* ima postavljenu zastavicu.

void set_mines(int x, int y)

Funkcija koja nasumično postavlja mine po ploči. Također, mine ne mogu biti postavljene na polje *grid[x][y]* ni na njegove susjede. Ova funkcija se koristi prilikom prvog lijevog klika igrača.

void set_numbers ()

Funkcija koja svakoj ćeliji pridružuje broj susjednih mina.

int neighbour_mines(int i, int j, int r, int c)

Pomoćna funkcija koja broji koliko susjednih ćelija ima *grid[i][j]*. Ako je postavljena mina na *grid[i][j]*, funkcija vraća -1.

boolean win()

Funkcija koja vraća *true* ako su otvorena sva polja koja nisu mine, inače vraća *false*.

4. Moguća poboljšanja

- Pozadinska glazba umjesto trenutnog zvuka.
- Omogućiti igraču da odabere „custom mode“, tj. da sam može odabrati dimenzije ploče te broj mina.
- Dodati opciju da igrač može postaviti znak ? na zatvoreno polje (npr. ako sumnja da bi tu mogla biti mina, ali nije sasvim siguran).
- Dodati još posebne efekte nakon pobjede (npr. padanje konfeta).
- Dodati još posebne efekte nakon eksplozije mine.
- Omogućiti igranje pomoću tipkovnice.
- Dodati „highscore“ za svaku težinu koja ostane spremljena i nakon zatvaranja programa (što uključuje i unos nadimka igrača prije početka igre).
- Na ekranu, kada je *gameState*=playing, dodati i gumb za izlazak iz igre i povratak na glavni izbornik ili napisati negdje da se pritiskom na 'q' izlazi iz igre i vraća na glavni izbornik.
- Omogućiti da srednji klik miša radi i na zatvorenim poljima (da istakne zatvorene susjede).
- Dodati podizbornik koji opisuje pravila igre.
- Dodati podizbornik „controls“ koji opisuje što koja tipka radi.

5. Literatura

<https://processing.org/>

<http://minesweeperonline.com/>

[https://en.wikipedia.org/wiki/Minesweeper_\(video_game\)](https://en.wikipedia.org/wiki/Minesweeper_(video_game))