

Projekat

Minesweeper je klasična računarska igra u kojoj je cilj otkriti sva polja na kvadratnoj tabli bez detoniranja mina. Tabla je podeljena u polja, a neka od tih polja sadrže mine, dok druga polja sadrže brojeve koji označavaju koliko mina se nalazi u neposrednom susedstvu tog polja.

Igrač započinje igru tako što odabere polje na tabli. Ako odabere polje s minom, igra se završava i mine se otkrivaju. Ako odabere polje bez mine, prikazuje se broj koji označava koliko mina se nalazi u susedstvu tog polja. Igrač koristi ove informacije da bi otkrio sigurna polja i izbegao polja s minama. Igra se završava kada su sva polja bez mina uspešno otkrivena.

Napisati program na programskom jeziku Scala za realizaciju jedne varijante igre Minesweeper. Implementirana igra može se pogledati na <https://minesweeperonline.com>¹. Za maksimalan broj poena, program treba da ispuni sve uslove projektnog zadatka i da radi pouzdano.

0. [15p] Grafički korisnički interfejs igre

Studentu je dozvoljeno korišćenje biblioteka za pravljenje grafičkog korisničkog interfejsa po izboru. Korisnički interfejs treba da bude intuitivan i dovoljno detaljan. Korisnik sa programom intereaguje putem jednostavnog menija kojim se bira da li se započinje nova igra, učitava sačuvana, nezavršena igra, kreira nivo ili pregledaju najbolji ostvareni rezultati.

1. [20p] Osnovne funkcionalnosti – učitavanje, odigravanje i čuvanje igre, računanje poena

Program treba da korisniku omogući sledeće manipulacije:

- Učitavanje nivoa iz fajla. Mapa nivoa je pravougaona matrica koja može da sadrži sledeće simbole:
 - simbol “-” koji predstavlja obično polje;
 - simbol “#” koji predstavlja polje sa minom.
- Zapocinjanje nove igre izborom težine ("Početnik", "Srednji" i "Ekspert"). Težina igre utiče na granice za veličinu table i dozvoljeni opseg broja mina na tabli. Na primer, na Početničkom nivou može biti manja tabla sa manje mina, dok na Ekspertskom nivou može biti veća tabla sa više mina.
- Nakon izbora težine, igraču se prikazuje spisak dostupnih nivoa zadate težine. Igrač bira konketan nivo ili se odlučuje za slučajno odabran nivo zadate težine.
- Pri odigravanju poteza u igri Minesweeper, igrač koristi miš kako bi interagovao sa igrom.
- Levim klikom, igrač otkriva polje na tabli. Ako je polje mina, igra se završava i mine se otkrivaju. Ako polje nema minu, prikazuje se broj koji označava koliko mina se nalazi u susedstvu tog polja u slučaju da na barem jednom susednom polju postoji mina. U slučaju da se u susedstvu polja ne nalazi nijedna mina, razotkriva se grupa polja koja su međusobno povezana i u susedstvu nemaju mine i granica formirana od polja koja su susedi mina ili ivica mape. Radi lakšeg razumevanja, predlažemo da odigrate nekoliko nivoa na

¹ Pristupano 14.5.2024. godine.

<https://minesweeperonline.com> i obratite pažnju na način na koji se otkrivanje polja radi pri prvom levom kliku.

- Desnim klikom, igrač označava polje koje pretpostavlja da sadrži minu. Ovo mu pomaže da prati gde misli da su mine. Desnim klikom na označeno polje, oznaka se uklanja i na polje se može ponovo kliknuti levim klikom. Levi klik na označeno polje nema efekta.
- Meri se vreme trajanja nivoa i broj klikova i na osnovu toga dodeljuje rezultat ako se nivo uspešno reši. Najbolji rezultati se prate i prikazuju izborom odgovarajuće opcije iz glavnog menija.
- Obezbediti dugme za pomoć kojim se korisniku predlaže potez. Svaka upotreba ove funkcionalnosti smanjuje ostvarene poene.
- Može se sačuvati i učitati započeti nivo iz tekstualnog fajla. Logično osmisliti format za čuvanje nivoa.
- Obezbediti odigravanje sekvence poteza učitavanjem iz fajla. Sekvenca poteza sadrži redom oznake poteza, svaki u zasebnom redu. Oznake poteza su:
 - Levi klik na polje u redu x i koloni y (x i y numerisani od 1): "L(x,y)"
 - Desni klik na polje u redu x i koloni y (x i y numerisani od 1): "D(x,y)"

2. [15+30p] Kreiranje novih nivoa na osnovu postojećih

[15p] Proste operacije za kreiranje nivoa su sledeće:

- *Proširivanje mape nivoa redom ili kolonom* (dodaju se kao prvi ili kao poslednji red/kolona).
- *Uklanjanje prve ili poslednje kolone ili reda* iz mape nivoa.
- *Zamena tipa polja* na zadatoj poziciji drugim tipom polja.
- *Čišćenje zadatog pravougaonog sektora mape* koje pretvara sva polja u njemu u polja bez mina.

Obezbediti i proveru validnosti nivoa, koja podrazumeva proveru da li su dimenzije nivoa i broj polja sa minama u dozvoljenim granicama za zadata težinu nivoa. Ova funkcionalnost ne proverava da li je nivo rešiv.

[30p] *Izometrije* su operacije koje se primenjuju na zadati pravougaoni sektor mape (original) i određuju njegovu sliku. Može se desiti da neko od polja slike nije u okvirima mape, pa svaka od izometrija može biti *proširujuća* (proširuje mapu potrebnim brojem kolona i/ili vrsta tako da sva polja slike budu u okviru mape) ili *neproširujuća* (polja slike koja su van okvira mape se izostavljaju). Po načinu na koji interaguju sa poljima unutar originala i slike, izometrije mogu da budu transparentne (slika se ponaša kao providna folija na kojoj su polja sa minama tačke, tako da se nova mapa dobija superpozicijom polja unutar ciljnog sektora i slike) ili netransparentne (slika se ponaša kao parče neprovidnog papira na kojem su polja sa minama tačke, tako da se ciljani sektor "prekriva" slikom). Nezavisno od transparentnosti, polja u originalu koja nisu u slici se čiste od mina. Za maksimalan broj poena, proširivost i transparentnost treba implementirati u skladu sa principima funkcionalnog programiranja u programskom jeziku Scala, recimo korišćenjem crta.

- *Rotacija*: rezultujuća mapa dobija se tako što se zadati pravougaoni sektor originalne mape zarotira za 90 stepeni u zatom smeru oko zadatog polja. Polja unutar originalnog sektora postaju obična polja.
- *Osna refleksija*: rezultujuća mapa dobija se tako što se zadati pravougaoni sektor originalne mape osnom simetrijom preslika u odnosu na zadata kolonu, vrstu ili dijagonalu.

Izometrijama može da se napravi kvazi-inverz i mogu se komponovati putem odgovarajućih

operatora.

- Obezbediti pravljenje proizvoljne imenovane izometrije ulančavanjem tj. komponovanjem izometrija (prostih ili kompozitnih). Rezultat je funkcija, dakle voditi računa da se eventualne provere domena i ulaznih podataka svih koraka koji se komponuju moraju obaviti na početku tela kompozicije, pri svakom pozivanju kompozicije.
- *Translacija* i *centralna simetrija* su dostupne izometrije koje se dobijaju komponovanjem odgovarajućih rotacija i osnih simetrija.
- Kvazi-inverz proizvoljne izometrije I je izometrija J koja ima suprotan efekat.

Za preostalih 20 poena potrebno je implementirati jedan od zahteva 3 i 4.

3. [20p] Implementirati pametno, adaptivno generisanje nivoa koje obezbeđuje:

- da prvo otvoreno polje ne sme sadržati minu, već biti polje čiji nijedan sused nije mina i deo je “ostrva” koje sadrži slučajan broj polja bez mina;
- da igrač koji igra optimalno (koristi sve dostupne informacije) što manje mora da rizikuje, već da se neaktiveni deo mape adaptivno menja njegovim igranjem tako da se teži tome da postoji barem jedno polje za koje se optimalnim logičkim zaključivanjem može reći da li je mina ili čisto, odnosno odigrati naredni potez bez rizikovanja.

Ovo podrazumeva da se nivo generiše ili prilagođava u zavisnosti od napravljenog poteza.

4. [20p] Sprovesti testiranje korišćenjem biblioteke ScalaTest. Potrebno je implementirati svite testova koje ispituju i demonstriraju ispravnost rada svake od implementiranih funkcionalnosti. Svaka svita testova treba da sadrži barem 3 jedinična (Unit) testa.

Program napisati tako da se koriste programski konstrukti svojstveni funkcionalnim jezicima, koji su obrađeni tokom nastave na predmetu. Gde je primereno, koristiti funkcije višeg reda, parcijalno primenjene funkcije, odnosno funkcije u kerifikovanom obliku, crte, implicitne funkcije i klase itd.

Za sve nedovoljno precizne zahteve od studenata se očekuje da usvoje razumne pretpostavke i da ih dosledno primenjuju u rešenju.