

Projekat

Napisati program na programskom jeziku Scala za realizaciju jedne varijante igre Bloxorz. Igra se zasniva na kretanju bloka dimenzija 1x1x2 po terenu kompleksnog oblika. Teren za igru se sastoji iz ploča po kojima se blok kreće kotrljanjem pomoću četiri komande (gore, dole, levo, desno). Cilj igre je dovesti blok od startne do ciljne pozicije. Igra se može završiti na dva načina:

- pobedom - kada blok dospe u ciljnu poziciju tako da je na ciljnoj ploči postavljen uspravno;
- porazom - kada se blok pomeri izvan granica terena ili postavi uspravno na neku od specijalnih ploča.

Za maksimalan broj poena, program treba da ispuni sve uslove projektnog zadatka i da radi pouzdano.

0. [10p] Korisnički interfejs treba da bude intuitivan i dovoljno detaljan. Korisnik sa programom intereaguje putem jednostavnog menija. Program treba da ispiše sadržaj menija, a zatim čeka da korisnik izabere redni broj neke od ponuđenih stavki, nakon čega, pre izvršenja, od korisnika očekuje da po potrebi unese dodatne parametre. Program treba da korisniku omogući sledeće manipulacije:

1. [20p] Osnovne funkcionalnosti

- Učitavanje mape terena iz fajla. Format fajla definisati prema primeru fajla sa zapisom mape terena iz Dodatka 1. Mapa terena je matrica koja može da sadrži sledeće simbole:
 - simbol “o” koji predstavlja običnu ploču;
 - simbol “-” koji označava da na datoj poziciji nije postavljena ploča;
 - simbol “S” koji označava startnu poziciju;
 - simbol “T” koji označava krajnju poziciju;
 - simbol “.” koji označava specijalnu ploču - ukoliko se blok postavi na ovu ploču u uspravnom položaju, ploča nestaje i blok propada van terena.
- Započinjanje nove igre izborom neke od raspoloživih mapa.
- Odigravanje poteza. Odigravanje predstavlja pomeranje bloka u zavisnosti od poteza koji korisnik zada (dole, gore, levo, desno), uz prikaz stanja igre nakon odigravanja poteza. Ukoliko se nakon odigravanja blok dovede u ciljnu poziciju, igra se završava pobedom. Ukoliko se nakon odigravanja blok izbaci van terena, igra se završava porazom.
- Odigravanje sekvence poteza učitavanjem iz fajla. Sekvenca poteza sadrži redom oznake poteza, svaki u zasebnom redu. Oznake poteza su:
 - dole: “d”
 - gore: “u”
 - levo: “l”
 - desno: “r”

2. [30p] Kreiranje novih mapa na osnovu postojećih. Operacije za kreiranje mapa su sledeće:

- Uklanjanje zadate ploče sa ivice terena.
- Dodavanje ploče na zadatu poziciju na ivici terena.
- Zamena obične ploče na zadatoj poziciji specijalnom.
- Zamena specijalne ploče na zadatoj poziciji običnom.
- Postavljanje startne pozicije na zadato polje, pri čemu se originalna startna pozicija menja običnom pločom.
- Postavljanje ciljne pozicije na zadato polje, pri čemu se originalna ciljna pozicija menja običnom pločom.
- Pravljenje proizvoljne imenovane kompozitne operacije ulančavanjem osnovnih (prostih ili kompozitnih).
- Predvideti sledeće predefinisane operacije:
 - Inverzija: rezultujuća mapa dobija se tako što se na originalnoj mapi zamene startna i ciljna pozicija.
 - Zamena: rezultujuća mapa dobija se tako što se na originalnoj mapi sve specijalne ploče pretvaraju u obične ploče.
- Filtriranje: ukoliko je makar jedan od suseda date ploče na vertikalnoj ili horizontalnoj razdaljini manjoj ili jednakoj N specijalna ploča, zameniti datu ploču običnom pločom.
- Formiranje imenovane sekvence operacija, koja se definiše kao operacija koja se sastoji od liste operacija.

3. [20p] Ispis rešenja igre u vidu sekvence poteza koji dovode blok iz početne u krajnju poziciju. Sekvencu rezultujućih poteza upisati u fajl. Sekvenca poteza sadrži redom oznake poteza, svaki u zasebnom redu, po istom formatu kao u okviru jedne od prethodnih stavki. Dovoljno je prikazati jedno od mogućih rešenja. Za maksimalni broj poena, implementirani rešavač mora poštovati principe funkcionalnog programiranja, dakle biti implementiran kao terminalna rekurzija bez *var* promenljivih.

Za preostalih 20 poena potrebno je implementirati jedan od zahteva 4 i 5

4. [20p] Realizovati grafički korisnički interfejs za igru. Studentu je dozvoljeno korišćenje biblioteka za pravljenje grafičkog korisničkog interfejsa po izboru.
5. [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.

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

Dodatak 1: primer mape terena

```
-----  
-ooo.-----  
-oSoooo-----  
-oooooooo-----  
--oooooooo-----  
-----ooToo--  
-----ooo-----  
-----
```