

FTN Terms

ncurses battleship type board game spinoff

02.06.2014.

SADRŽAJ:

1. UVOD
2. AUTORSKA PRAVA I NAPOMENE
3. SADRŽAJ DATOTEKE I POKRETANJE PROGRAMA
4. VEŠTAČKA INTELIGENCIJA
5. PROBLEMI

1. UVOD

FTN Terms je kompjuterska igra zasnovana na ideji društvene igre Battleship koja je nastala u prvoj polovini 20. veka.

Pravila igre Battleship su sledeća:

- Igraju je dva igrača
- Svaki igrač ima tablu veličine 10x10 polja
- Svaki igrač na raspolaganju ima 5 brodova, tako da postoji:
 - jedan brod veličine 5 polja
 - jedan broj veličine 4 polja
 - dva broda veličine 3 polja
 - jedan brod veličine 2 polja
- Na početku igre se 5 brodova raspoređuju na polja
- Nakon postavljanja table naizmenično se prozivaju koordinate table
- Ukoliko je koordinata označena brodom, polje je potopljeno
- Igra se nastavlja sve dok jedan od igrača ne pogodi sve pozicije brodova sa druge table

Kako je cilj igre jasan – pogoditi sve pozicije na ograničenoj tabli, koncept o pravljenju kompjuterskog modela je nastao. Algoritamski jasna igra je bio prvi izbor za prvi veći i samostalni projekat, kroz koji bih upoznao osnove pravljenja veštačke inteligencije i poboljšao veštine programiranja.

Osnovna proširenja igre Battleship koja su implementirana:

- Mogućnost izbora veličine table
- Implementacija osnovne veštačke inteligencije različitih nivoa težine
- Interfejs svojstven kompjuterskim igrama

Da bi igra dobila na originalnosti, pored proširenja izabrana je i tematika koju je svaki student doživeo – prvi semestar.

Kod programa pisan je u programskom jeziku C. Za grafički prikaz upotrebljena je biblioteka ncurses.

2. AUTORSKA PRAVA I NAPOMENE

Naziv programa: **FTN Terms**

Verzija: 1.0.0

Autor programa: Viktor Šanca

Kontakt: viktor.sanca@gmail.com

Za izradu projekta korišćen je IDE Code Blocks, kompajler gcc i biblioteka ncurses. Dokumentacija koda generisana je pomoću alata Doxygen. ASCII art tekst u fajlu logo.txt koji se upotrebljava prilikom pokretanja glavnog menija generisan je na web-sajtu: <http://patorjk.com/software/taag/> .

Program podleže GNU General Public Licence.

Ovaj program je besplatan softver: moguće ga je redistribuirati i/ili modifikovati pod uslovima GNU General Public Licenci objavljenoj od strane Free Software Fondacije, bilo verzije 3 Licence, ili kasnije verzije. Za pun tekst licence posetite website: www.gnu.org/licenses.

Copyright © 2014. Viktor Šanca

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

3. SADRŽAJ DATOTEKE I POKRETANJE PROGRAMA

Sve priložene datoteke neophodne su za pravilno funkcionisanje programa.

Izvorni kod:

- FTNTerms1.0.0.c
- menu.h

Lokalizacija i uputstvo:

- engleski jezik:
 - eng.txt
 - gameeng.txt
 - manualeng.txt
- srpski jezik:
 - srb.txt
 - gamesrb.txt
 - manualsrp.txt

Dokumentacija:

- Documentation.pdf
- folder html
 - dokumentaciji se pristupa otvaranjem fajla index.html

Datoteke:

- logo.txt
- hiscore.txt

Makefile:

- makefile

Pokretanje programa:

Program se pokreće preko terminala. Pomoću terminala otvoriti lokaciju na kojoj je otpakovana arhiva FTNTerms1.0.0. U terminal ukucati komandu *make*. Program pokrenuti u terminalu pomoću komande *./FTNTerms*.

NAPOMENA: Da bi se program uspešno kompajlirao potrebna je 32-bitna ncurses biblioteka. (sudo apt-get install lib32ncurses5-dev)

4. VEŠTAČKA INTELIGENCIJA (AI)

Program je zasnovan na poteznoj igri protiv računara, odnosno napisanog algoritma koji omogućava nekoliko stepena težine.

AI segmenti:

- kreiranje table
- potezi u igri
 - težina AI:
 - lako
 - srednje
 - teško
 - veoma teško

Kreiranje table je zasnovano na raspoređivanju "brodova" na tabli, odnosno algoritmu za raspored oznaka na matrici veličine korisničke table koja je varijabilne veličine (minimalno 5x5, maksimalno 26x26). Sam raspored jedinica na tabli izveden je nasumičnom raspodelom po strogo određenom pravilu, kako bi se prilikom svakog pokretanja igre ostvario element nepoznate table. Problem prilikom generisanja ovakve table jesu slučajevi prilikom kojih nije moguće završiti kreiranje table po definisanim parametrima. Takvo ponašanje rešeno je uvođenjem brojača koji nakon određenog maksimalnog broja iteracija prilikom kojih se pokušava pronaći odgovarajuće polje poništava dati pokušaj postavke "broda" i ponovo pokreće funkciju za raspored. Da bi se utvrdila valjanost algoritma raspodela na tabli i eventualne bagove upotrebljena je direktna metoda posmatranja generisanih tabli. Testiranje je vršeno na tablama veličine 5x5 i 10x10 i izvršena je serija od 200 000 testiranja prilikom kojih je algoritam unapređen. Za "nasumičnost" upotrebljena je funkcija standardne biblioteke srand i randomize. Parametre navedenih funkcija i implementaciju moguće je videti u izvornom kodu.

Potezi u igri predstavljaju segment prilikom kojeg računar "pogađa" polje na tabli korisnika. Svako pogađanje zasnovano je na nasumičnom izboru polja na tabli korisnika uz modifikaciju ponašanja u zavisnosti od izbora težine. Postoje četiri implementirana nivoa težine: lak, srednji, težak i veoma težak.

Lak nivo težine predstavlja osnovni koncept nasumičnog pogađanja polja na tabli korisnika uz uslov da se tabla pretražuje sve dok se ne pronađe polje koje nije još uvek gađano. Ovaj mod predstavlja čist slučajan izbor.

Srednji nivo težine predstavlja proširenje lakog nivoa težine pri kojem postoji teorijska verovatnoća od $0.25 \cdot 0.25 = 0.0625$ da se izabere polje koje je sigurno označeno i nije još uvek pogođeno. U praksi ova verovatnoća je mnogo veća jer se na slučajan proces dodaje sigurno pogađanje.

Težak nivo predstavlja proširenje srednjeg nivoa gde verovatnoća da se pored slučajnog pogađanja sigurno pogodi označeno polje iznosi $0.5 \cdot 0.25 = 0.125$.

Najteži nivo predstavlja verovatnoću od 0.5 za pogađanje polja na tabli.

Radi provere ispravnosti napisanog algoritma igra je privremeno modifikovana tako da se postave potrebni parametri veličine table i nivoa težine i pokrene simulacija igre između dva AI igrača. Jedini parametar koji figuriše u ovoj proverbi jeste broj koraka koji je potreban da bi se stiglo do kraja igre, odnosno da su sva označena polja bilo kog od igrača pogođena. Simulacija je puštana u serijama od po 100 000 partija na tablama veličine 5x5 i 10x10, pri čemu je tražena srednja vrednost broja potrebnih poteza.

5. PROBLEMI

Prilikom programiranja brojni problemi su se pojavili. Prvi problem predstavljalo je pravilno rešavanje postavljanja AI table. Mogućnost slučajeva pri kojima nije moguće dalje postavljanje po navedenom algoritmu uzrokuje prekid programa. Navedeni problem rešen je uvođenjem funkcije za poništavanje poteza koja se primenjuje nakon što broj iteracija pređe izračunatu vrednost (kako bi se izbegla beskonačna petlja koja je uzrokovana samim problemom rasporeda na tabli). Funkcija generisanja nasumične vrednosti `srand(time(NULL))` proširena je iz tih razloga na `srand(time(NULL)+nasumična_vrednost)` kako se dobijena vrednost ne bi dobila dva puta, što bi uzrokovalo prekid programa.

Identičan problem se dogodio prilikom kodiranja nasumičnog AI pogađanja, koji je rešen modifikovanjem `srand` funkcije na već naveden način.

Front end igre realizovan je pomoću `ncurses` biblioteke. Kako je sama biblioteka stara i ograničenih mogućnosti manipulacije sistemom (može da modifikuje samo izgled terminala), pojavilo se nekoliko problema. Prvi problem predstavlja nemogućnost interne promene veličine terminala na ekranu. Dati problem rešen je tako što se korisnik odgovarajućom porukom na ekranu obavesti da je za nastavak igre potrebno povećati veličinu terminala. Drugi problem predstavlja pravilan ispis na ekranu, kao i olakšan unos korisnika. Iz navedenog razloga uvedeno je ograničenje table od maksimalno 26x26 polja. Iz internih razloga i pravila igre tabla mora biti veća od 5x5 kako bi uopšte bilo moguće označiti makar neka polja radi igre.

Da bi se ostvarila šira portabilnost programa, izvorni kod se kompajlira kao 32-bitni.