Sah in C

Ferencsik-Moldovan Lukas

1.Ideea din spate

Jocul foloseste o logica destul de complicata de implementat, dar relative usoara de inteles intrucat am dorit ca miscarile sa fie facute in scriere algebrica ex:Qe6.

2.Structura si librarii

A screen shot of a computer program

AI-generated content may be incorrect.

Pe langa librariile uzuale, apar ncurses,unistd si arpa/inet. Ncurses este folosit pentru a desena table si a avea o interfata mai prietenoasa, iar arpa/inet si unistd pentru a facilita partea de multiplayer, facuta prin socketuri.

Jocul este alcatuit din doua programe, cel de client si cel de server.

3.Clientul si serverul

Pe partea de client se regaseste codul care determina tot jocul, precum miscarile pieselor,creeare de table, mutarea pieselor etc., iar serverul se ocupa de comunicarea dintre cei doi clienti.

3.1 Conectarea client-serverA computer screen shot of a program code

AI-generated content may be incorrect.A computer code on a black background

AI-generated content may be incorrect.

Acesta este codul de conectare din partea serverului si a clientului care foloseste functiile uzuale de socket, bind si listen pentru a creea iar apoi accepta clienti pe socket-ul creat.

A computer screen shot of a program code

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

Mai apoi intra intr-un while in care asteapta sa se conecteze doi client. Dupa conectare le transmite START urmat de un numar pentru a le indica ce culoare sunt, 1 pentru alb si 2 pentru negru.

3.2 Inceputul Jocului

A screen shot of a computer program

AI-generated content may be incorrect.

Prima data se afiseaza pentru fiecare tabla. Se folosesc doua afisari pentru a printa tabla cu piesele albe jos pentru jucatorul cu alb si invers pentru cel cu piesele negre.

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

Mai apoi intra in doua while-uri, unul care continua pana este sah mat sau remiza si unul care continua la infinit pana se da un break. La inceput se verifica daca ce culoare esti si a cui runda este. Daca este runda persoanei cu piesele negre atunci pesoanei cu piesele albe este facut sa astepte pana ce oponentul isi face mutare si vice versa. Dupa ce face mutarea si trece toate testele de validare se trimite mutarea catre server iar acesta o trimite catre oponent, care pe propria tabla face mutarea oponentului. Asa arata aceasta “convorbire” intre cei doi clienti, si totodata acesta este si singurul rol al serverului:

A screen shot of a computer program

AI-generated content may be incorrect.

3.3 Mutarile

Dupa cum am spus mai sus, mutarile se fac in notatie algebrica, astfel codul preia un string de 3 carctere astfel:

A screen shot of a computer code

AI-generated content may be incorrect.

Dupa ce este primit, se verifica daca este o mutare valida:

A screen shot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer screen

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

In cazul in care nu este o miscare valida, se reia tura persoanei cu mesajul “MISCARE INVALIDA”.

Miscarea pieselor se face aproximativ la fel la toate.

A screen shot of a computer screen

AI-generated content may be incorrect.

Acest switch case verifica in primul rand ce tip de piesa este, iar apoi o introduce intr-o functie care verifica daca se poate muta unde a scris jucatorul ca vrea sa o mute. Aceasta verifcare este cam asa: incepe de la coordonatele introduse de jucator, spre exemplu a5 care este pozitia 04 in matrice, verifica daca pe acea pozitie este o piesa de acelasi tip cu cea a jucatorului si daca este returneaza 1 adica miscare invalida, daca nu, continua si verifica daca exista o piesa de acel tip care poate face mutarea, spre exemplu daca piesa dorita este R ,adica tura, codul va cauta pe intreaga coloana si intreaga linie o tura ,iar daca o gaseste va verifica daca nu este nimic in calea ei si poate ajunge cu succes acolo, daca da va returna 0 si va pune piesa pe pozitia dorita, in caz contrar returneaza 1.

3.4 Disambiguation

Poate v-ati pus problema ce se intampla daca exista doua ture care pot ajunge in locul dorit, oare cum stie codul care este piesa dorita? Raspunsul este ca nu stie, dar daca cumva gaseste doua piese de acelasi tip care pot ajunge pe aceleasi coordonate va aparea un mesaj cu "Coord Piesa dorita:" care mai apoi citeste doua caractere care reprezinta coordonatele piesei dorite, mai apoi intra in functia de disambiguation si si pune eventualele piese care mai pot ajunge pe pozitia dorita pe 0 si face mutarea asa iar apoi o pune inapoi pe ce caracter era.

3.5 Verificarea de sah si sah mat

Verificarea de sah se face la final si cauta fiecare rege, iar apoi incearca fiecare mutare posibila si verifica daca poate ajunge la pozitia regelui.

Verificarea de sah mat se face dupa verificarea de sah in cazul in care functia de sah a returnat ca este sah si incearca fiecare mutare a regelui iar apoi intra din nou in functia de sah si tot asa. Daca nu gaseste o miscare valida a regelui returneaza faptul ca este sah mat si jocul se termina.

LINK GITHUB:https://github.com/qLukas9/PROIECTTP