**Takım ile Tic Tac Toe (XxOoXx)**

INF212 Project #3

Prepared by

1801022028

Furkan Küçük

INF 212 Algorithms and Programming II

Spring 2020

Electronic Engineering

Date Submitted: 8 Haziran 2020

Evaluation/Değerlendirme

|  |  |
| --- | --- |
| Report layout /Rapor düzeni | 10 -- |
| Project Flowchart and functions flow chart /Projenin ve fonksiyonların Akış Diyagramı | 25 -- |
| Programming /Programlama Struct, Functions | 45 -- |
| Questions / Sorular | 20 -- |

|  |
| --- |
| Project Objective /Projenin Amacı |
| Projenin amacı 2 takım ve 4 oyuncu ile hepimizin bildiği xox oyununu geliştirerek bir oyun yazmak. Bu sayede algoritma oluşturma yeteneğimizi geliştirmek. |
| Description of Problem / Problem Tanımlama |
| Projedeki problem class yapısını kullanarak hocamızın verdiği kazanma şartlarına uygun oyun geliştirebilmek ve ardından hamlelerin analizlerini hesaplayabilmektir. |
| **Description of Method / Metodunun Tanımlaması** |
| Projede class yapısı kullanıldı hamlelerin analizi yapılarak ekrana bastırıldı. |
| Text of Program/ Programın Kodunu Tanımlaması |
| Program kodu raporun en altında ve proje raporunun yanında mevcuttur. (.cpp halinde) |
| User’s guide / Kullanıcı Rehberi |
| Programın kullanımı oldukça basittir. Kodu çalıştırdıktan sonra ekrandaki yazıları takip ederek hamlelerinizi yapabilirsiniz. Sırası gelen kullanıcı hamle yapmak istediği kutucugun numarasını girerek hamlesini tamamlayabilir. Oyun bittiğinde uyarılacaksınız. |
| Results of the solution / Programın Sonuçları Give the program inputs and outputs for test cases.  Programın giriş ve çıkışlarını örnekler ile gösterin. |
| Conclusion and Remarks / Sonuç ve Notlar |
| Proje eksiksiz bir şekilde çalışıyor. Proje çalışmaları sırasında hocamızın verdiği kazanma şartlarını uzun uzadıya yazmamak için bir algoritma geliştirmem gerekti ve en çok burada zorlandım. Kodun geri kalan kısımlarını internetteki xox projelerine bakarak gayet basit bir şekilde yazabildim. Güzel bir projeydi. |
| References / Kaynaklar |
| <https://www.delimuhendis.com/tic-tac-toe-xox/>  <https://stackoverflow.com/questions/26088549/c-tic-tac-toe-using-classes>  <http://hakangokce06.blogspot.com/2013/04/c-x-o-x-oyun-kodu.html>  <https://gist.github.com/ardamavi/54a95f549063793140715597b63ba4d5>  <https://github.com/Pipping/Xox-for-2-players/blob/master/XOX%20CONSOLE.cpp> |

Oyun.h

#ifndef OYUN\_h

#define OYUN\_h

class XxOoXx {

private:

int board[4][4];

int tHamle;

int sira;

public:

XxOoXx();

void oyunGiris();

void drawBoard();

void getMove(int);

void kazananiBul();

void printGameResult(int);

void analiz();

int control();

};

#endif

Oyun.cpp

#include <iostream>

using namespace std;

#include "oyun.h"

#define SIZE 4

#define TEAMO 1

#define TEAMX 2

#define EMPTY -1

XxOoXx::XxOoXx(){

int temp[SIZE][SIZE]={{EMPTY,EMPTY,EMPTY,EMPTY},{EMPTY,EMPTY,EMPTY,EMPTY},{EMPTY,EMPTY,EMPTY,EMPTY},{EMPTY,EMPTY,EMPTY,EMPTY}};

int i,j;

for(i=0;i<SIZE;i++){

for(j=0;j<SIZE;j++)

board[i][j]=temp[i][j];

}

tHamle = 0;

}

void XxOoXx::drawBoard(){

int i,j;

for(i=0;i<SIZE;i++){

for (j=0;j<SIZE;j++){

if((j%SIZE)==0) cout << "\n";

if(board[i][j]==EMPTY) cout << (i\*SIZE)+(j+1) << " ";

else{

if(board[i][j]==1) cout << "o ";

else if (board[i][j]==2) cout << "X ";

else if (board[i][j]==3) cout << "O ";

else if (board[i][j]==4) cout << "x ";

else cout << "Hata" ;

}

}

}

}

void XxOoXx::oyunGiris (){

cout << "Programda numarali alanlar bos oldugunu gostermektedir.\n"

<< "Oyun ikiser kisilik iki takim ile oynanir.\n"

<< "Xx Takimi: 1. kisi -> X karakterini, 2. kisi -> x karakterini kullanabilir.\n"

<< "Oo Takimi: 1. kisi -> O karakterini, 2. kisi -> o karakterini kullanabilir.\n";

}

void XxOoXx::getMove(int hamle){

tHamle++;

sira=tHamle%SIZE;

// Geçersiz hamleler önleniyor.

if (((hamle<0)||(hamle>17))){

cout << "Gecersiz hamle yaptiniz.\n"

<< "Yeni hamle yapiniz.\n";

while((hamle<0)||(hamle>17)){

cout << "Yeni hamleniz bekleniyor.";

cin >> hamle;

}

}

// Hamlenin yeri belirleniyor.

if (hamle<=SIZE){

if(board[0][hamle-1]==EMPTY){

cout << "Hamleniz gerceklestirildi." << endl;

board[0][hamle-1]=sira+1;

}

else{

cout << "\nHamle yaptiginiz alan doludur.\n"

<< "Hamleniz gecersiz sayilacaktir.\n"

<< "Tahta goruntulendikten sonra yeni hamlenizi yapiniz.\n"

<< endl;

tHamle--;

}

}

else if (hamle <=8){

if(board[1][(hamle%5)]==EMPTY){

cout << "Hamleniz gerceklestirildi." << endl;

board[1][(hamle%5)]=sira+1;

}

else{

cout << "\nHamle yaptiginiz alan doludur.\n"

<< "Hamleniz gecersiz sayilacaktir.\n"

<< "Tahta goruntulendikten sonra yeni hamlenizi yapiniz.\n"

<< endl;

tHamle--;

}

}

else if (hamle<=12){

if(board[2][((hamle%8)-1)]==EMPTY){

cout << "Hamleniz gerceklestirildi." << endl;

board[2][((hamle%8)-1)]=sira+1;

}

else{

cout << "\nHamle yaptiginiz alan doludur.\n"

<< "Hamleniz gecersiz sayilacaktir.\n"

<< "Tahta goruntulendikten sonra yeni hamlenizi yapiniz.\n"

<< endl;

tHamle--;

}

}

else{

if(board[3][((hamle%12)-1)]==EMPTY){

cout << "Hamleniz gerceklestirildi." << endl;

board[3][((hamle%12)-1)]=sira+1;

}

else{

cout << "\nHamle yaptiginiz alan doludur.\n"

<< "Hamleniz gecersiz sayilacaktir.\n"

<< "Tahta goruntulendikten sonra yeni hamlenizi yapiniz.\n"

<< endl;

tHamle--;

}

}

}

void XxOoXx::printGameResult(int team){

if(team==TEAMX) cout << "\n\nOYUN 1. TAKIM KAZANMISTIR." << endl;

else if(team==TEAMO) cout << "\n\nOYUN 2. TAKIM KAZANMISTIR." << endl;

else cout << "\n\n\*\*OYUN BERABERE BITMISTIR.\*\*" << endl;

tHamle = SIZE\*SIZE+1;

}

void XxOoXx::kazananiBul(){

bool winStatus=true,loopStatus=true;

int i=0,j=0,x=0,teamStatus, x1count = 0, x2count = 0, o1count = 0, o2count = 0;

if(tHamle==SIZE\*SIZE+1) printGameResult(0);

else{

while(i<SIZE && loopStatus){ /\*YATAY KAZANMA DURUMU KONTROLÜ\*/

while(j<SIZE && loopStatus){

if(j==0){

if(board[i][j]==EMPTY) winStatus=false;

if(board[i][j]==1 || board[i][j]==3) teamStatus=TEAMO;

else if(board[i][j]==2 || board[i][j]==4) teamStatus=TEAMX;

}

if(teamStatus==TEAMO && winStatus==true){

if(board[i][j]%2!=1) winStatus=false;

}

if(teamStatus==TEAMX && winStatus==true){

if(board[i][j]%2!=0) winStatus=false;

}

j++;

}

if(winStatus)

{

if(board[i][0]!=board[i][3] && board[i][1]!=board[i][2] && board[i][0]!=board[i][1] && board[i][2]!=board[i][3]) winStatus=false;

while(x<SIZE){

if (board[i][x] == 1) o1count++;

else if (board[i][x] == 2) x2count++;

else if (board[i][x] == 3) o2count++;

else if (board[i][x] == 4) x1count++;

x++;

}

if (teamStatus == TEAMO && o1count != o2count) winStatus = false;

else if(teamStatus == TEAMX && x1count != x2count) winStatus = false;

x = 0;

o1count = 0; o2count = 0; x1count = 0; x2count = 0;

}

if (winStatus) loopStatus=false;

j=0;

i++;

winStatus=true;

}

if(winStatus && !loopStatus){

printGameResult(teamStatus);

return;

}

i=0; j=0; winStatus=true; loopStatus=true;

while(j<SIZE && loopStatus){ /\*DIKEY KAZANMA DURUMU KONTROLÜ\*/

while(i<SIZE && loopStatus){

if(i==0){

if(board[i][j]==EMPTY) winStatus=false;

if(board[i][j]==1 || board[i][j]==3) teamStatus=TEAMO;

else if(board[i][j]==2 || board[i][j]==4) teamStatus=TEAMX;

}

if(teamStatus==TEAMO && winStatus==true){

if(board[i][j]%2!=TEAMO) winStatus=false;

}

if(teamStatus==TEAMX && winStatus==true){

if(board[i][j]%2!=0) winStatus=false;

}

i++;

}

if(winStatus)

{

if(board[0][j]!=board[3][j] && board[1][j]!=board[2][j] && board[0][j]!=board[1][j] && board[2][j]!=board[3][j]) winStatus=false;

while(x<SIZE){

if (board[x][j] == 1) o1count++;

else if (board[x][j] == 2) x2count++;

else if (board[x][j] == 3) o2count++;

else if (board[x][j] == 4) x1count++;

x++;

}

if (teamStatus == TEAMO && o1count != o2count) winStatus = false;

else if(teamStatus == TEAMX && x1count != x2count) winStatus = false;

x = 0;

o1count = 0; o2count = 0; x1count = 0; x2count = 0;

}

if (winStatus) loopStatus=false;

i=0;

j++;

winStatus=true;

}

if(winStatus && !loopStatus){

printGameResult(teamStatus);

return;

}

if((board[0][0]==2 && board[1][1]==4 && board[2][2]==4 && board[3][3]==2)

||(board[0][0]==4 && board[1][1]==2 && board[2][2]==2 && board[3][3]==4)

||(board[0][0]==2 && board[1][1]==2 && board[2][2]==4 && board[3][3]==4)

||(board[0][0]==4 && board[1][1]==4 && board[2][2]==2 && board[3][3]==2)

||(board[0][3]==2 && board[1][2]==4 && board[2][1]==4 && board[3][0]==2)

||(board[0][3]==4 && board[1][2]==2 && board[2][1]==2 && board[3][0]==4)

||(board[0][3]==2 && board[1][2]==2 && board[2][1]==4 && board[3][0]==4)

||(board[0][3]==4 && board[1][2]==4 && board[2][1]==2 && board[3][0]==2))

{

printGameResult(TEAMX);

}

else if((board[0][0]==3 && board[1][1]==1 && board[2][2]==1 && board[3][3]==3)

||(board[0][0]==1 && board[1][1]==3 && board[2][2]==3 && board[3][3]==1)

||(board[0][0]==3 && board[1][1]==3 && board[2][2]==1 && board[3][3]==1)

||(board[0][0]==1 && board[1][1]==1 && board[2][2]==3 && board[3][3]==3)

||(board[0][3]==3 && board[1][2]==1 && board[2][1]==1 && board[3][0]==3)

||(board[0][3]==1 && board[1][2]==3 && board[2][1]==3 && board[3][0]==1)

||(board[0][3]==3 && board[1][2]==3 && board[2][1]==1 && board[3][0]==1)

||(board[0][3]==1 && board[1][2]==1 && board[2][1]==3 && board[3][0]==3))

{

printGameResult(TEAMO);

}

}

}

int XxOoXx::control(){

if(tHamle == SIZE\*SIZE+1) return 2020;

return 2019;

}

void XxOoXx::analiz(){

int i,j;

float satirx=0.0,satiro=0.0,kosegenx=0.0,kosegeno=0.0;

for(i=0;i<4;i++){

for(j=0;j<4;j++){

if(board[i][j]==2 || board[i][j]==4)

satirx++;

if(board[i][j]==1 || board[i][j]==3)

satiro++;

}

}

if(board[0][0]==1 || board[0][0]==3)

kosegeno++;

if(board[1][1]==1 || board[1][1]==3)

kosegeno++;

if(board[2][2]==1 || board[2][2]==3)

kosegeno++;

if(board[3][3]==1 || board[3][3]==3)

kosegeno++;

if(board[0][0]==2 || board[0][0]==4)

kosegenx++;

if(board[1][1]==2 || board[1][1]==4)

kosegenx++;

if(board[2][2]==2 || board[2][2]==4)

kosegenx++;

if(board[3][3]==2 || board[3][3]==4)

kosegenx++;

if(board[0][3]==1 || board[0][3]==3)

kosegeno++;

if(board[1][2]==1 || board[1][2]==3)

kosegeno++;

if(board[2][1]==1 || board[2][1]==3)

kosegeno++;

if(board[3][0]==1 || board[3][0]==3)

kosegeno++;

if(board[0][3]==2 || board[0][3]==4)

kosegenx++;

if(board[1][2]==2 || board[1][2]==4)

kosegenx++;

if(board[2][1]==2 || board[2][1]==4)

kosegenx++;

if(board[3][0]==2 || board[3][0]==4)

kosegenx++;

kosegenx= (kosegenx\*100)/8;

kosegeno= (kosegeno\*100)/8;

satirx = (satirx\*100)/16;

satiro = (satiro\*100)/16;

cout << "Xx | Satir orani: %" << satirx << " Sutun orani %" << satirx << " Kosegen orani %" << kosegenx << endl;

cout << "Oo | Satir orani: %" << satiro << " Sutun orani %" << satiro << " Kosegen orani %" << kosegeno << endl;

}

main.cpp

#include <iostream>

#include "oyun.h"

using namespace std;

int main(){

XxOoXx game;

int hamleyap;

game.oyunGiris();

while((game.control())!=2020){

game.drawBoard();

cout << "\nHamlenizi giriniz: " ;

cin >> hamleyap;

game.getMove(hamleyap);

game.kazananiBul();

}

game.analiz();

system("PAUSE");

return 0;

}