**Tic Tac Toe Game**

INF212 Project #03

Prepared by

1801022071

Alperen Arslan

INF 212 Algorithms and Programming II

Spring 2020

Electronic Engineering

Date Submitted: 08.06.2020

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 bir Tic Tac Toe oyunu yapmak. Genel mantığı XoX oyunun ikişer kişilik takımlarla oynanması. Takımlar ikişer kişilik olduğu için karakterlerde XxOoXx şeklinde oluyor. Aynı şekilde harita da 4x4 oluyor. Takımlar kendi karakterlerini alt alta , yan yana veya çapraz olarak 4 kareye yazdırmaları durumunda oyunu kazanırlar. Oyunun başında oyunculardan nickleri alınır. Oyunun sonunda ise oyunla alakalı istatistikler gösterilir. |
| *Description of Problem / Problem Tanımlama* |
| Oyunun giriş, çıkış ve oyunun oynandığı ekranları yazdıracağımız konsol ekranlarını etkili olarak kullanabilmek, takımların karakterlerini oyun tahtasının istedikleri yerine yerleştirebilmelerini sağlamak, yerleştirilen karakterleri analiz edip kazanma durumunu belirleyebilme ve ekrana bu durumu yazdırmak. |
| **Description of Method / Metodunun Tanımlaması** |
| C++ dilinde bir sınıf, üye fonksiyonları, operatör overload fonksiyonları, friend fonksiyonları kullanılarak oyunu oluşturdum. Oyun ilk açıldığında oyunculardan isimleri alınır. İsimler ilgili string değişkenlere atanır. Oyun tahtası önümüze çıktığında ise ilk oyuncudan başlayarak sırasıyla yerleştirileceği satır ve sutun alınır. Bu değerler ilgili değişkenlere atanır. Bu arada oluşturduğum char matris e de yazdırılan karakterleri ekledim. Oyunu kazanma, berabere kalma ve oyun sonu istatistiklerini bu matris üzerinden değerlendirdim. Kazanma ve berabere kalma durumları için matrisi önce satır olarak, sonra sütun olarak , sonra çapraz olarak ve en son da matrisin tamamını inceleyerek durumları belirledim. En son olarak da matrisi tekrar inceleyerek istatistiksel rakamları belirledim ve ekrana yazdırdım. |
| *Text of Program/ Programın Kodunu Tanımlaması* |
| Kod çok uzun olduğu için raporun sonuna eklendi. |
| *User’s guide / Kullanıcı Rehberi* |
| Program kısaca class game adında bir class yapısıyla oluşturuldu. Yapıya ait member functionlar, overload functionlar, friend functionlar gibi yapılar kullanıldı. Değişkenler olarak da genel değişkenler class a ait private değişkenler ve fonksiyonların local değişkenleri denebilir. Programda bir global değişken bulunmamakta. Input parametlerinden bahsedersekte gotoxy fonksiyonu koordinat mantığı ile kendisine gönderilen x ve y değişkenlerini kullanarak imleci konsol ekranında istenilen konuma getiriyor. Kodun içinde bu fonksiyon kullanıcıdan herhangi bir değer almıyor. Sadece benim tarafımdan belirlenen değerler ile kullanıldı. SignWrite isimli değişken ise oyuncuların karakterlerini (X, o gibi) istedikleri satır ve sütuna yazdırmalarını sağlıyor. Input parametresi olarak da girilen satır, sutun ve hangi oyuncunun hamle yaptıgına dair bilgi veren int bir değişkenin değerlerini alıyor. Son olarak da oyuncuların ekrana baktıkları zaman sıranın hangi oyuncuda olduğunu anlamalarını sağlayacak işaretçiyi ekrana yazdırmakla görevli fonksiyon input parametresi alıyor. Hangi oyuncunun oynacağını saklayan int bir değişkeni input alıyor ve ona göre oynama sırasındaki oyuncunun isminin yanına işaretçi yazdırıyor. Onun dışındaki diğer fonksiyonlar ise input almıyor. Çıktılardan bahsedersek giriş ve çıkış menüleri ve oyunun oynandığı oyun menusunu soyleyebiliriz. Olası sınırlamalarda oyun, oyuncular satır ve sutun icin 4 değerinden daha büyük bir değer girerlerse sıra atlamadan aynı oyuncudan tekrar değer girmesini bekliyor. Olası hatalarda ise bazen çapraz kazanma durumlarında takımın kazanması sayılmayabiliyor. Ayrıca ilk ekranda oyuncu eğer çok uzun bir nick girerse oyun ekranına gelindiğinde nick in belli bir kısmı gösterilemiyor. |
| *Results of the solution / Programın Sonuçları* C:\Users\hamdi\OneDrive\Masaüstü\alperen\Proje 3\girisEkran.pngC:\Users\hamdi\OneDrive\Masaüstü\alperen\Proje 3\girisEkran2.pngOyun ilk açıldığında aşağıdaki gibi ekranla karşılaşılır. Daha sonra oyuncuların isimleri alınır.  C:\Users\hamdi\OneDrive\Masaüstü\alperen\Proje 3\oyunEkrani.pngOyuncular isimlerini girdikten sonra aşağıdaki gibi bir oyun ekranı açılır.  Kazanma durumlarında aşağıdaki gibi görsellerle karşılaşılır.  C:\Users\hamdi\OneDrive\Masaüstü\alperen\Proje 3\IstatistikSayfasi.pngC:\Users\hamdi\OneDrive\Masaüstü\alperen\Proje 3\caprazKazanma.pngC:\Users\hamdi\OneDrive\Masaüstü\alperen\Proje 3\sutunKazanma.pngC:\Users\hamdi\OneDrive\Masaüstü\alperen\Proje 3\satirKazanma.png |
| *Flowchart of the Program / Programın Akış Diyagramı* |
| C:\Users\hamdi\OneDrive\Masaüstü\alperen\Proje 3\Flowchart.png |
| *Conclusion and Remarks / Sonuç ve Notlar* |
| Bu proje süresince genellikle karşılaştığım sorunlar sırasında çok az internet kullandım. Bazı unuttuğum syntax kullanımları dışında algoritmayı ve kodu inşaa ederken karşılaştığım sorunları düşünerek aştım. Bu da koda fazlaca vakit ayırmama sebep oldu. Ama sonucunda bana daha iyi algoritma ve kod inşaası konusunda yeni yetenekler kazandırdı. Eskiden karşılaşsam belki saatler harcayacağım mantıkları artık daha kısa sürede ve kolayca algılayıp uygulayabiliyorum. Tabi ki zaten bu tarz projelerinde amacı budur. O yüzden bu proje sırasında kazanmam gereken özellikleri kazandım diyebilirim. Program ise %90 oranında sorunsuz çalışıyor. Sadece sağ çaprazdan kazanma durumunda sıkıntı yaşanıyor. Onun dışındaki kombinasyonlarda takımlar kazandıkları veya berabere kaldıkları zaman herhangi bir sorun olmuyor. Program gayet kullanıcı dostu. Oyuncular oyuna girdikleri zaman herhangi bir zorluk yaşamadan oyunu çözüp oynayabilirler. Kodun iyileştirilmesi ise algoritma mantığı üzerine daha uzun süreler düşünülüp ,farklı yöntemler denenerek kodun satır sayısı azaltılarak sağlanabilir. Sağ çapraz kazanma durumu hakkındaki sorun ortadan kaldırılabilir. Onun dışında proje süresince proje için yeterli zaman ayırdım ve sorunları kendim üzerine düşünerek, kendi algoritma mantığımı kurmaya çalışarak gereken özeni gösterdim. Kazandığım yetenekler kısmına gelirsek ise ucu açık, programın sonucunda analiz edilmesi gereken verinin farklılığı ihtimalinin gayet yüksek düzeyde olduğu programlarda, koda nasıl yaklaşılması gerektiği ve nasıl bir algoritma mantığı kurulması gerektiği hakkında en azından giriş seviyesinde bilgi sahibi oldum. Her projede olduğu gibi algoritma inşaası konusunda kendimi ilerlettim ve yeni fonksiyonlar ve kütüphaneler keşfettim. |
| *References / Kaynaklar* |
| * Kodlakafa.com * Alldatasheet.com * Stackoverflow.com * Algoritmaornekleri.com * Arduinolibraries.com * Github.com * Sanatsalbilgi.com * Theengineeringprojects.com * Rapidtables.com * Koddefteri.com * Erhanulman.com * Forum.donanimhaber.com * Cagataycebi.com * Forum.csharpnedir.com * En.wikipedi0.org * Hayaletveyap.com * Grafiboyz.blogcu.com * Aliozcan.org * Gelecegiyazanlar.turkcell.com.tr * Iucoders.com * Sanalkurs.net * Buraksenyurt.com * Microcore.com * Link24in.wordpress.com * Soyunmezoglubaris.blogspot.com * Motorobit.com * Kodumundunyasi.net |

# *KOD:*

***MAİN***

#include"tictactoe.h"

**int** main**()** **{**

game tic**;**

tic**.**entranceCode**();**

tic**.**gameCode**();**

**return** 0**;**

**}**

***HEADER***

#ifndef TICTACTOE\_H

#define TICTACTOE\_H

#include<iostream>

#include<windows.h>

#include<cctype>

#include<conio.h>

#include<iomanip>

**using** **namespace** std**;**

**class** game**{**

**private:**

string oneOneNick**;**

string oneTwoNick**;**

string twoOneNick**;**

string twoTwoNick**;**

**char** signs**[**4**]** **=** **{**'X'**,**'x'**,**'O'**,**'o'**};**

**char** map**[**4**][**4**]** **=** **{{**'n'**,**'n'**,**'n'**,**'n'**},{**'n'**,**'n'**,**'n'**,**'n'**},{**'n'**,**'n'**,**'n'**,**'n'**},{**'n'**,**'n'**,**'n'**,**'n'**}};**

**int** playerAtNow**;**

**int** drawCounter**;**

**int** analysisCounter**;**

**public:**

game**(){**

oneOneNick **=** **" ";**

oneTwoNick **=** **" ";**

twoOneNick **=** **" ";**

twoTwoNick **=** **" ";**

playerAtNow **=** 0**;**

drawCounter **=** 0**;**

analysisCounter **=** 0**;**

**}**

**void** **friend** gotoxy**(int** x**,** **int** y**);**

**void** **friend** nickPointer**(int** playerAtNow**);**

**void** gameTheme**();**

**void** gameCode**();**

**void** entranceTheme**();**

**void** entranceCode**();**

**void** signWrite**(int** line**,** **int** column**,** **int** i**);**

**int** winControl**();**

**void** analysisTheme**();**

**void** analysis**();**

game**&** **operator++();**

**};**

**void** gotoxy**(int** x**,** **int** y**);**

**void** nickPointer**();**

#endif

***CPP***

#include"tictactoe.h"

**void** game**::**entranceTheme**(){**

*//definitions*

**const** **char** teamLine **=** 196**;**

**const** **char** frame **=** 219**;**

*//doing stuff*

*//periphalers*

gotoxy**(**0**,**3**);**

cout **<<** **"TEAM 1";**

**for(int** i **=** 0 **;** i **<=** 5 **;** i**++){**

gotoxy**(**i**,**4**);**

cout **<<** teamLine**;**

**}**

gotoxy**(**0**,**6**);**

cout **<<** oneOneNick**;**

gotoxy**(**0**,**8**);**

cout **<<** oneTwoNick**;**

gotoxy**(**27**,**3**);**

cout **<<** **"TEAM 2";**

**for(int** i **=** 27 **;** i **<=** 32 **;** i**++){**

gotoxy**(**i**,**4**);**

cout **<<** teamLine**;**

**}**

gotoxy**(**27**,**6**);**

cout **<<** twoOneNick**;**

gotoxy**(**27**,**8**);**

cout **<<** twoTwoNick**;**

*//game board*

**for(int** i **=** 8 **;** i **<** 25**;** i**++){**

gotoxy**(**i**,**3**);**

cout **<<** frame**;**

gotoxy**(**i**,**16**);**

cout **<<** frame**;**

**}**

**for(int** j **=** 3 **;** j **<** 17 **;** j**++){**

gotoxy**(**8**,**j**);**

cout **<<** frame**;**

gotoxy**(**24**,**j**);**

cout **<<** frame**;**

**}**

**}**

**void** game**::**entranceCode**(){**

*//definitions*

**int** i **=** 1 **,** j **=** 1**;**

*//doing stuff*

**while(true){**

entranceTheme**();**

gotoxy**(**9**,**7**);**

cout **<<** **"Team "** **<<** i **<<** **" Player "** **<<** j**;**

gotoxy**(**12**,**9**);**

cout **<<** **"Enter Nick";**

gotoxy**(**12**,**11**);**

**if** **(** i **==** 1 **&&** j **==** 1 **){**

j**++;**

cin **>>** oneOneNick**;**

**}**

**else** **if** **(** i **==** 1 **&&** j **==** 2 **){**

j **=** 1**;**

i**++;**

cin **>>** oneTwoNick**;**

**}**

**else** **if** **(** i **==** 2 **&&** j **==** 1 **){**

j**++;**

cin **>>** twoOneNick**;**

**}**

**else{**

cin **>>** twoTwoNick**;**

**break;**

**}**

system**("cls");**

**}**

gotoxy**(**12**,**11**);**

entranceTheme**();**

Sleep**(**1000**);**

**}**

**void** gotoxy**(int** x**,** **int** y**){**

COORD coord**;**

coord**.**X **=** x**;**

coord**.**Y **=** y**;**

SetConsoleCursorPosition**(**GetStdHandle**(**STD\_OUTPUT\_HANDLE**),**coord**);**

**}**

**void** nickPointer**(int** playerAtNow**){**

**const** **char** pointerRight **=** 175**;**

**if(**playerAtNow **==** 0**){**

gotoxy**(**26**,**8**);**

cout **<<** **" ";**

gotoxy**(**0**,**6**);**

cout **<<** pointerRight**;**

**}**

**else** **if(**playerAtNow **==** 1**){**

gotoxy**(**0**,**6**);**

cout **<<** **" ";**

gotoxy**(**0**,**8**);**

cout **<<** pointerRight**;**

**}**

**else** **if(**playerAtNow **==** 2**){**

gotoxy**(**0**,**8**);**

cout **<<** **" ";**

gotoxy**(**26**,**6**);**

cout **<<** pointerRight**;**

**}**

**else{**

gotoxy**(**26**,**6**);**

cout **<<** **" ";**

gotoxy**(**26**,**8**);**

cout **<<** pointerRight**;**

**}**

**}**

**void** game**::**gameTheme**(){**

*//definitions*

**const** **char** teamLine **=** 196**;**

**const** **char** frame **=** 219**;**

**const** **char** verticalLine **=** 179**;**

**const** **char** flatLine **=** 196**;**

*//doing stuff*

*//periphalers*

gotoxy**(**0**,**3**);**

cout **<<** **"TEAM 1";**

**for(int** i **=** 0 **;** i **<=** 5 **;** i**++){**

gotoxy**(**i**,**4**);**

cout **<<** teamLine**;**

**}**

gotoxy**(**2**,**6**);**

cout **<<** oneOneNick**;**

gotoxy**(**2**,**8**);**

cout **<<** oneTwoNick**;**

gotoxy**(**27**,**3**);**

cout **<<** **"TEAM 2";**

**for(int** i **=** 27 **;** i **<=** 32 **;** i**++){**

gotoxy**(**i**,**4**);**

cout **<<** teamLine**;**

**}**

gotoxy**(**28**,**6**);**

cout **<<** twoOneNick**;**

gotoxy**(**28**,**8**);**

cout **<<** twoTwoNick**;**

gotoxy**(**5**,**22**);**

cout **<<** **"L I N E:";**

gotoxy**(**17**,**22**);**

cout **<<** **"C O L U M N:";**

*//game board*

**for(int** i **=** 8 **;** i **<** 25**;** i**++){**

gotoxy**(**i**,**3**);**

cout **<<** frame**;**

gotoxy**(**i**,**19**);**

cout **<<** frame**;**

**}**

**for(int** j **=** 3 **;** j **<** 19 **;** j**++){**

gotoxy**(**8**,**j**);**

cout **<<** frame**;**

gotoxy**(**24**,**j**);**

cout **<<** frame**;**

**}**

**for(int** i **=** 12 **;** i **<=** 20 **;** i **+=** 4**){**

**for(int** j **=** 4 **;** j **<** 19 **;** j**++){**

gotoxy**(**i**,**j**);**

cout **<<** verticalLine**;**

**}**

**}**

**for(int** j **=** 7 **;** j **<** 19 **;** j **+=** 4**){**

**for(int** i **=** 9 **;** i **<** 24 **;** i**++){**

gotoxy**(**i**,**j**);**

cout **<<** flatLine**;**

**}**

**}**

**}**

**void** game**::**signWrite**(int** line**,** **int** column**,** **int** i**){**

**if(**line **==** 1**){**

**switch(**column**){**

**case** 1**:**

gotoxy**(**10**,**5**);**

cout **<<** signs**[**i**];**

**break;**

**case** 2**:**

gotoxy**(**14**,**5**);**

cout **<<** signs**[**i**];**

**break;**

**case** 3**:**

gotoxy**(**18**,**5**);**

cout **<<** signs**[**i**];**

**break;**

**case** 4**:**

gotoxy**(**22**,**5**);**

cout **<<** signs**[**i**];**

**break;**

**}**

**}**

**if(**line **==** 2**){**

**switch(**column**){**

**case** 1**:**

gotoxy**(**10**,**9**);**

cout **<<** signs**[**i**];**

**break;**

**case** 2**:**

gotoxy**(**14**,**9**);**

cout **<<** signs**[**i**];**

**break;**

**case** 3**:**

gotoxy**(**18**,**9**);**

cout **<<** signs**[**i**];**

**break;**

**case** 4**:**

gotoxy**(**22**,**9**);**

cout **<<** signs**[**i**];**

**break;**

**}**

**}**

**if(**line **==** 3**){**

**switch(**column**){**

**case** 1**:**

gotoxy**(**10**,**13**);**

cout **<<** signs**[**i**];**

**break;**

**case** 2**:**

gotoxy**(**14**,**13**);**

cout **<<** signs**[**i**];**

**break;**

**case** 3**:**

gotoxy**(**18**,**13**);**

cout **<<** signs**[**i**];**

**break;**

**case** 4**:**

gotoxy**(**22**,**13**);**

cout **<<** signs**[**i**];**

**break;**

**}**

**}**

**if(**line **==** 4**){**

**switch(**column**){**

**case** 1**:**

gotoxy**(**10**,**17**);**

cout **<<** signs**[**i**];**

**break;**

**case** 2**:**

gotoxy**(**14**,**17**);**

cout **<<** signs**[**i**];**

**break;**

**case** 3**:**

gotoxy**(**18**,**17**);**

cout **<<** signs**[**i**];**

**break;**

**case** 4**:**

gotoxy**(**22**,**17**);**

cout **<<** signs**[**i**];**

**break;**

**}**

**}**

map**[**line**-**1**][**column**-**1**]** **=** signs**[**i**];**

drawCounter**++;**

**}**

**int** game**::**winControl**(){**

*//definitions*

**int** rowWinX **=** 0**,**colWinX **=** 0**,**rowWinO **=** 0**,**colWinO **=** 0**;**

**int** leftCrossWinX **=** 0**,**rightCrossWinX **=** 0**,**leftCrossWinO **=** 0**,**rightCrossWinO **=** 0**;**

**char** sign**;**

*//doing stuff*

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

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

sign **=** tolower**(**map**[**i**][**j**]);**

**if(**sign **==** 'x'**){**

rowWinX**++;**

**}**

**else** **if(**sign **==** 'o'**){**

rowWinO**++;**

**}**

**}**

**if(**rowWinX **==** 4**){**

gotoxy**(**5**,**24**);**

cout **<<** **"ROW WIN FOR X TEAM...";**

**return** 1**;**

**}**

**else** **if(**rowWinO **==** 4**){**

gotoxy**(**5**,**24**);**

cout **<<** **"ROW WIN FOR O TEAM...";**

**return** 1**;**

**}**

**else{**

rowWinX **=** 0**;**

rowWinO **=** 0**;**

**}**

**}**

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

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

sign **=** tolower**(**map**[**i**][**j**]);**

**if(**sign **==** 'x'**){**

colWinX**++;**

**}**

**else** **if(**sign **==** 'o'**){**

colWinO**++;**

**}**

**}**

**if(**colWinX **==** 4**){**

gotoxy**(**5**,**24**);**

cout **<<** **"COLUMN WIN FOR X TEAM...";**

**return** 1**;**

**}**

**else** **if(**colWinO **==** 4**){**

gotoxy**(**5**,**24**);**

cout **<<** **"COLUMN WIN FOR O TEAM...";**

**return** 1**;**

**}**

**else{**

colWinX **=** 0**;**

colWinO **=** 0**;**

**}**

**}**

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

sign **=** tolower**(**map**[**i**][**i**]);**

**if(**sign **==** 'x'**){**

leftCrossWinX**++;**

**}**

**else** **if(**sign **==** 'o'**){**

leftCrossWinO**++;**

**}**

**if(**leftCrossWinX **==** 4**){**

gotoxy**(**5**,**24**);**

cout **<<** **"LEFT CROSS WIN FOR X TEAM...";**

**return** 1**;**

**}**

**else** **if(**leftCrossWinX **==** 4**){**

gotoxy**(**5**,**24**);**

cout **<<** **"LEFT CROSS WIN FOR O TEAM...";**

**return** 1**;**

**}**

**}**

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

**int** j **=** 3**;**

sign **=** tolower**(**map**[**i**][**j**]);**

**if(**sign **==** 'x'**){**

rightCrossWinX**++;**

**}**

**else** **if(**sign **==** 'o'**){**

rightCrossWinO**++;**

**}**

**if(**rightCrossWinX **==** 4**){**

gotoxy**(**5**,**24**);**

cout **<<** **"RIGHT CROSS WIN FOR X TEAM...";**

**return** 1**;**

**}**

**else** **if(**rightCrossWinO **==** 4**){**

gotoxy**(**5**,**24**);**

cout **<<** **"RIGHT CROSS WIN FOR O TEAM...";**

**return** 1**;**

**}**

j**--;**

**}**

**if(**drawCounter **==** 16**){**

gotoxy**(**5**,**24**);**

cout **<<** **"DRAW...";**

**return** 1**;**

**}**

**return** 0**;**

**}**

**void** game**::**gameCode**(){**

*//definitions*

**bool** running **=** **true;**

**int** line **=** 1 **,** column **=** 1**,** returnValue **=** 0**;**

*//doing stuff*

system**("cls");**

**while(**running**){**

gameTheme**();**

nickPointer**(**playerAtNow**);**

labelOne**:**

returnValue **=** winControl**();**

**if(**returnValue **==** 1**){**

Sleep**(**5000**);**

system**("cls");**

analysis**();**

**}**

gotoxy**(**14**,**22**);**

cin **>>** line **;**

labelTwo**:**

gotoxy**(**30**,**22**);**

cin **>>** column**;**

gotoxy**(**14**,**22**);**

cout **<<** **" ";**

gotoxy**(**30**,**22**);**

cout **<<** **" ";**

**if(**line **<** 1 **||** line **>** 4 **){**

gameTheme**();**

**goto** labelOne**;**

**}**

**if(**column **<** 1 **||** column **>** 4 **){**

gameTheme**();**

**goto** labelTwo**;**

**}**

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

**if(**map**[**line**-**1**][**column**-**1**]** **==** signs**[**i**])** **goto** labelOne**;**

**}**

**if(**playerAtNow **==** 0**){**

signWrite**(**line**,**column**,**playerAtNow**);**

playerAtNow**++;**

**}**

**else** **if(**playerAtNow **==** 1**){**

signWrite**(**line**,**column**,**playerAtNow**);**

playerAtNow**++;**

**}**

**else** **if(**playerAtNow **==** 2**){**

signWrite**(**line**,**column**,**playerAtNow**);**

playerAtNow**++;**

**}**

**else{**

signWrite**(**line**,**column**,**playerAtNow**);**

playerAtNow **=** 0**;**

**}**

**}**

**}**

game**&** game**::operator++(){**

analysisCounter**++;**

**return** **\*this;**

**}**

**float** analysisCalculations**(int** x**){**

**return** **(**100 **\*** **(float)**x**)** **/** 16 **;**

**}**

**void** game**::**analysisTheme**(){**

*//definitions*

**const** **char** teamLine **=** 196**;**

**const** **char** frame **=** 219**;**

**const** **char** verticalLine **=** 179**;**

**const** **char** flatLine **=** 196**;**

*//doing stuff*

*//game board*

**for(int** i **=** 8 **;** i **<** 33**;** i**++){**

gotoxy**(**i**,**3**);**

cout **<<** frame**;**

gotoxy**(**i**,**20**);**

cout **<<** frame**;**

**}**

**for(int** j **=** 3 **;** j **<** 20 **;** j**++){**

gotoxy**(**8**,**j**);**

cout **<<** frame**;**

gotoxy**(**32**,**j**);**

cout **<<** frame**;**

**}**

**for(int** i **=** 20 **;** i **<=** 26 **;** i **+=** 6**){**

**for(int** j **=** 4 **;** j **<** 20 **;** j**++){**

gotoxy**(**i**,**j**);**

cout **<<** verticalLine**;**

**}**

**}**

**for(int** j **=** 7 **;** j **<** 19 **;** j **+=** 4**){**

**for(int** i **=** 9 **;** i **<** 32 **;** i**++){**

gotoxy**(**i**,**j**);**

cout **<<** flatLine**;**

**}**

**}**

gotoxy**(**20**,**20**);**

**}**

**void** game**::**analysis**(){**

*//definitons*

game rowNumberX**,** columnNumberX**,** diagonalNumberX**;**

game rowNumberO**,** columnNumberO**,** diagonalNumberO**;**

**char** sign**;**

*//doing stuff*

*//calculations*

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

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

sign **=** tolower**(**map**[**i**][**j**]);**

**if(**sign **==** 'x'**){**

rowNumberX**.**analysisCounter**++;**

**}**

**else** **if(**sign **==** 'o'**){**

rowNumberO**.**analysisCounter**++;**

**}**

**}**

**}**

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

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

sign **=** tolower**(**map**[**i**][**j**]);**

**if(**sign **==** 'x'**){**

columnNumberX**.**analysisCounter**++;**

**}**

**else** **if(**sign **==** 'o'**){**

columnNumberO**.**analysisCounter**++;**

**}**

**}**

**}**

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

sign **=** tolower**(**map**[**i**][**i**]);**

**if(**sign **==** 'x'**){**

diagonalNumberX**.**analysisCounter**++;**

**}**

**else** **if(**sign **==** 'o'**){**

diagonalNumberO**.**analysisCounter**++;**

**}**

**}**

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

**int** j **=** 3**;**

sign **=** tolower**(**map**[**i**][**j**]);**

**if(**sign **==** 'x'**){**

diagonalNumberX**.**analysisCounter**++;**

**}**

**else** **if(**sign **==** 'o'**){**

diagonalNumberO**.**analysisCounter**++;**

**}**

j**--;**

**}**

*//printing the calculations*

analysisTheme**();**

gotoxy**(**10**,**5**);**

cout **<<** **"TEAM";**

gotoxy**(**10**,**9**);**

cout **<<** **"ROW";**

gotoxy**(**10**,**13**);**

cout **<<** **"% COLUMN";**

gotoxy**(**10**,**17**);**

cout **<<** **"% DIAGONAL";**

gotoxy**(**22**,**5**);**

cout **<<** **"Xx";**

gotoxy**(**28**,**5**);**

cout **<<** **"Oo";**

gotoxy**(**22**,**9**);**

cout **<<** fixed **<<** setprecision**(**1**)** **<<** analysisCalculations**(**rowNumberX**.**analysisCounter**);**

gotoxy**(**22**,**13**);**

cout **<<** fixed **<<** setprecision**(**1**)** **<<** analysisCalculations**(**columnNumberX**.**analysisCounter**);**

gotoxy**(**22**,**17**);**

cout **<<** fixed **<<** setprecision**(**1**)** **<<** analysisCalculations**(**diagonalNumberX**.**analysisCounter**);**

gotoxy**(**28**,**9**);**

cout **<<** fixed **<<** setprecision**(**1**)** **<<** analysisCalculations**(**rowNumberO**.**analysisCounter**);**

gotoxy**(**28**,**13**);**

cout **<<** fixed **<<** setprecision**(**1**)** **<<** analysisCalculations**(**columnNumberO**.**analysisCounter**);**

gotoxy**(**28**,**17**);**

cout **<<** fixed **<<** setprecision**(**1**)** **<<** analysisCalculations**(**diagonalNumberO**.**analysisCounter**);**

gotoxy**(**10**,**22**);**

cout **<<** **"...Press Anything...";**

getch**();**

exit**(**1**);**

**}**