using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Drawing;

using System.Windows.Forms;

namespace Tic\_Tac\_Toe\_game\_Petralia

{

class board

{

public int movimento = 0;

public int VittorieO = 0;

public int VittorieX = 0;

private contenitori[,] contenitore = new contenitori[3, 3];

public int turnoGiocatore = X;

public int SelezionaTurnoGiocatore()

{

return turnoGiocatore;

}

public int getVittorieO()

{

return VittorieO;

}

public int getVittorieX()

{

return VittorieX;

}

public const int X = 0;

public const int O = 1;

public const int B = 2;

public void Inizializza\_Board()

{

for (int x = 0; x < 3; x++) //ordinate

{

for (int y = 0; y < 3; y++) //ascisse

{

contenitore[x, y] = new contenitori();

contenitore[x, y].setValore(B);

contenitore[x, y].setPosizione(new Point(x, y));

}

}

}

public void detecthit(Point loc)

{

if (loc.Y <= 500)

{

int x = 0;

int y = 0;

if (loc.X < 167)

{

x = 0;

}

else if (loc.X > 167 && loc.X < 334)

{

x = 1;

}

else if (loc.X > 334)

{

x = 2;

}

if (loc.Y < 167)

{

y = 0;

}

else if (loc.Y > 167 && loc.Y < 334)

{

y = 1;

}

else if (loc.Y > 334 && loc.Y < 500)

{

y = 2;

}

if (movimento % 2 == 0)

{

Grafica.disegnaX(new Point(x, y));

contenitore[x, y].setValore(X);

if (identificaLinea())

{

MessageBox.Show("Hai vinto X, Congratulazioni");

VittorieX++;

reset();

Grafica.setUp();

}

if (form1.richiamaIA() && !identificaLinea() && boardpiena())

{

contenitori mossaIA = InteligenzaArtificiale.determinaEPiazzaSimbolo(contenitore);

Grafica.disegnaO(mossaIA.getPosizione());

contenitore[mossaIA.getPosizione().X,mossaIA.getPosizione().Y].setValore(O);

if (identificaLinea())

{

MessageBox.Show("Il computer ha vinto");

VittorieO++;

reset();

Grafica.setUp();

}

movimento++;

}

turnoGiocatore = O;

}

else

{

Grafica.disegnaO(new Point(x, y));

contenitore[x, y].setValore(O);

if (identificaLinea())

{

MessageBox.Show("Hai vinto O, Congratulazioni");

VittorieO++;

reset();

Grafica.setUp();

}

turnoGiocatore = X;

}

movimento++;

}

}

public bool identificaLinea()

{

bool Vittoria = false;

for (int x = 0; x < 3; x++)

{

if (contenitore[x, 0].getValore() == X && contenitore[x, 1].getValore() == X && contenitore[x, 2].getValore() == X)

{

return true;

}

if (contenitore[x, 0].getValore() == O && contenitore[x, 1].getValore() == O && contenitore[x, 2].getValore() == O)

{

return true;

}

switch (x)

{

//indentifica le diagonali per la vittoria

case 0:

if (contenitore[x, 0].getValore() == X && contenitore[x+1,1].getValore() == X && contenitore[x+2,2].getValore() == X)

{

return true;

}

if (contenitore[x, 0].getValore() == O && contenitore[x + 1, 1].getValore() == O && contenitore[x + 2, 2].getValore() == O)

{

return true;

}

break;

case 2:

if (contenitore[x, 0].getValore() == X && contenitore[x - 1, 1].getValore() == X && contenitore[x - 2, 2].getValore() == X)

{

return true;

}

if (contenitore[x, 0].getValore() == O && contenitore[x - 1, 1].getValore() == O && contenitore[x - 2, 2].getValore() == O)

{

return true;

}

break;

}

}

for (int y = 0; y < 3; y++)

{

if (contenitore[0, y].getValore() == X && contenitore[1, y].getValore() == X && contenitore[2, y].getValore() == X)

{

return true;

}

if (contenitore[0, y].getValore() == O && contenitore[1, y].getValore() == O && contenitore[2, y].getValore() == O)

{

return true;

}

}

return Vittoria;

}

public void reset()

{

contenitore = new contenitori[3, 3];

Inizializza\_Board();

}

public bool boardpiena()

{

bool piena = true;

foreach (contenitori h in contenitore)

{

if (h.getValore() == board.B)

{

piena = true;

}

}

return piena;

}

}

}