ОГБПОУ "Томский техникум информационных технологий"

**Шахматы**

Подготовил студент  
группы 682  
Лебедюк Егор

Проверила  
Павлов В.С

Томск 2020

**Задание:**

Разработать код игры «Шахматы»  
**Код программы:**

class Program

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Drawing;

namespace ZMEIKKA

{

class Program

{

static void Main(string[] args)

{

Console.OutputEncoding = Encoding.Unicode;

Osnova XYI = new Osnova();

// Console.CursorVisible = false;

do

{

XYI.Figura\_vibor();

XYI.AI\_Figura\_vibor();

} while (true);

}

}

}

class Osnova

using System;

using System.Collections.Generic;

using System.Text;

namespace ZMEIKKA

{

public class Osnova

{

public bool end = true;

private string[,] mass = new string[,]

{

{ "\u265c ","\u265e ","\u265d ", "\u265b ", "\u265a ", "\u265d ", "\u265e ", "\u265c " },

{ "\u265F ","\u265F ","\u265F ", "\u265F ", "\u265F ", "\u265F ", "\u265F ", "\u265F " },

{ " "," "," ", " ", " ", " ", " ", " " },

{ " "," "," ", " ", " ", " ", " ", " " },

{ " "," "," ", " ", " ", " ", " ", " " },

{ " "," "," ", " ", " ", " ", " ", " " },

{ "\u2659 ", "\u2659 ","\u2659 ", "\u2659 ", "\u2659 ", "\u2659 ", "\u2659 ", "\u2659 " },

{ "\u2656 ", "\u2658 ","\u2657 ", "\u2655 ", "\u2654 ", "\u2657 ", "\u2658 ", "\u2656 " }

};

ConsoleKeyInfo mKey;

private int x = 0, y = 0;

int i = 0;

int Z = 0;

Random rand = new Random();

public Osnova()

{

this.BandW();

}

private void BandW()

{

for (int i = 0; i < mass.GetLength(0); i++)

{

for (int j = 0; j < mass.GetLength(1); j++)

{

if ((j + i) % 2 == 0)

{

Console.BackgroundColor = ConsoleColor.White;

Console.Write(mass[i, j]);

Console.ResetColor();

Console.BackgroundColor = ConsoleColor.Black;

}

else

{

Console.BackgroundColor = ConsoleColor.Black;

Console.Write(mass[i, j]);

Console.ResetColor();

}

}

Console.WriteLine();

}

}

public void Figura\_vibor()

{

Console.SetCursorPosition(x, y);

i = 0;

do

{

do

{

mKey = Console.ReadKey(true);

switch (mKey.Key)

{

case ConsoleKey.W:

if (y > 0)

{

y--;

Console.SetCursorPosition(x, y);

}

break;

case ConsoleKey.A:

if (x > 0)

{

x = x - 2;

Console.SetCursorPosition(x, y);

Z--;

}

break;

case ConsoleKey.S:

if (y < 7)

{

y++;

Console.SetCursorPosition(x, y);

}

break;

case ConsoleKey.D:

if (x < 7 \* 2)

{

x = x + 2;

Console.SetCursorPosition(x, y);

Z++;

}

break;

default:

break;

}

} while (mKey.Key != ConsoleKey.Enter);

if (mass[y, (x - Z)] == "\u2659 ")//Пешка

{

Pehuia\_i();

i++;

}

else if (mass[y, (x - Z)] == "\u2658 ")//конь

{

Kanina\_i();

i++;

}

else if (mass[y, (x - Z)] == "\u2657 ")//Слон

{

Slon\_i();

i++;

}

else if (mass[y, (x - Z)] == "\u2656 ")//Ладья

{

Ladia\_i();

i++;

}

else if (mass[y, (x - Z)] == "\u2655 ")//Королева

{

COVID\_19();

i++;

}

else if (mass[y, (x - Z)] == "\u2654 ")//Король

{

COVID\_20();

i++;

}

} while (i != 1);//==

}

private void Pehuia\_i()

{

int j = 0;

mKey = Console.ReadKey(true);

int add\_UD = 0, add\_LD = 0, x\_p = x, y\_p = y;

do

{

do

{

mKey = Console.ReadKey(true);

switch (mKey.Key)

{

case ConsoleKey.W:

if (y > 0)

{

y--;

Console.SetCursorPosition(x, y);

add\_UD--;

}

break;

case ConsoleKey.A:

if (x > 0)

{

x = x - 2;

Console.SetCursorPosition(x, y);

Z--;

add\_LD = add\_LD - 2;

}

break;

case ConsoleKey.S:

if (y < 7)

{

y++;

Console.SetCursorPosition(x, y);

add\_UD++;

}

break;

case ConsoleKey.D:

if (x < 7 \* 2)

{

x = x + 2;

Console.SetCursorPosition(x, y);

Z++;

add\_LD = add\_LD + 2;

}

break;

default:

break;

}

} while (mKey.Key != ConsoleKey.Enter);

if (Proverka\_i() == true)

{

if (mass[x\_p + 1, y\_p + 1] == "\u265b " || mass[x\_p + 1, y\_p + 1] == "\u265a " || mass[x\_p + 1, y\_p + 1] == "\u265d " || mass[x\_p + 1, y\_p + 1] == "\u265e " || mass[x\_p + 1, y\_p + 1] == "\u265c " || mass[x\_p + 1, y\_p + 1] == "\u265F " || mass[x\_p - 1, y\_p + 1] == "\u265b " || mass[x\_p - 1, y\_p + 1] == "\u265a " || mass[x\_p - 1, y\_p + 1] == "\u265d " || mass[x\_p - 1, y\_p+1] == "\u265e " || mass[x\_p - 1, y\_p + 1] == "\u265c " || mass[x\_p - 1, y\_p + 1] == "\u265F ")

{

j = 1;

j = 1;

if ((((x - Z) + y) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

mass[y, x - Z] = " ";

Console.SetCursorPosition(x, y);

Console.Write(mass[y, x - Z]);

y += add\_UD;

x\_p = x;

x += add\_LD;

if (x\_p > x)

{

Z--;

}

else

{

Z++;

}

Console.SetCursorPosition(x, y);

mass[y, x - Z] = "\u2659 ";

if ((((x - Z) + y) % 2) == 0)

{

Console.BackgroundColor = ConsoleColor.White;

Console.Write(mass[y, x - Z]);

}

else

{

Console.BackgroundColor = ConsoleColor.Black;

Console.Write(mass[y, x - Z]);

}

}

else if (x\_p == 0 && y\_p == 6 || x\_p == 2 && y\_p == 6 || x\_p == 4 && y\_p == 6 || x\_p == 6 && y\_p == 6 || x\_p == 8 && y\_p == 6 || x\_p == 10 && y\_p == 6 || x\_p == 12 && y\_p == 6 || x\_p == 14 && y\_p == 6)

{

j = 1;

if ((((x - Z) + y) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

mass[y, x - Z] = " ";

Console.SetCursorPosition(x, y);

Console.Write(mass[y, x - Z]);

y += add\_UD;

Console.SetCursorPosition(x, y);

mass[y, x - Z] = "\u2659 ";

if ((((x - Z) + y) % 2) == 0)

{

Console.BackgroundColor = ConsoleColor.White;

Console.Write(mass[y, x - Z]);

}

else

{

Console.BackgroundColor = ConsoleColor.Black;

Console.Write(mass[y, x - Z]);

}

}

else if (add\_UD == 1)

{

j = 1;

if ((((x - Z) + y) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

mass[y, x - Z] = " ";

Console.SetCursorPosition(x, y);

Console.Write(mass[y, x - Z]);

y += add\_UD;

Console.SetCursorPosition(x, y);

mass[y, x - Z] = "\u2659 ";

if ((((x - Z) + y) % 2) == 0)

{

Console.BackgroundColor = ConsoleColor.White;

Console.Write(mass[y, x - Z]);

}

else

{

Console.BackgroundColor = ConsoleColor.Black;

Console.Write(mass[y, x - Z]);

}

}

}

} while (j != 1);

}

private void Kanina\_i()

{

int lineHorse, columHorse, linePos, columPos;//lineHorse-строка columHorse2-столбец

lineHorse = y;

columHorse = x;

int Z\_old = Z;

int i = 0;

do

{

do

{

mKey = Console.ReadKey(true);

switch (mKey.Key)

{

case ConsoleKey.W:

if (y > 0)

{

y--;

Console.SetCursorPosition(x, y);

}

break;

case ConsoleKey.A:

if (x > 0)

{

x = x - 2;

Console.SetCursorPosition(x, y);

Z--;

}

break;

case ConsoleKey.S:

if (y < 7)

{

y++;

Console.SetCursorPosition(x, y);

}

break;

case ConsoleKey.D:

if (x < 7 \* 2)

{

x = x + 2;

Console.SetCursorPosition(x, y);

Z++;

}

break;

default:

break;

}

} while (mKey.Key != ConsoleKey.Enter);

linePos = y;

columPos = x;

if (Proverka\_i() == true)

{

if (((Math.Abs(linePos - lineHorse) == 2) && (Math.Abs((columPos - Z) - (columHorse - Z\_old)) == 1)) || ((Math.Abs(linePos - lineHorse) == 1) && (Math.Abs((columPos - Z) - (columHorse - Z\_old)) == 2)))

{

mass[lineHorse, columHorse - Z\_old] = " ";

Console.SetCursorPosition(columHorse, lineHorse);

if ((((columHorse - Z\_old) + lineHorse) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[lineHorse, columHorse - Z\_old]);

mass[linePos, columPos - Z] = "\u2658 ";

Console.SetCursorPosition(columPos, linePos);

if ((((columPos - Z) + linePos) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[linePos, columPos - Z]);

i++;

}

}

} while (i == 0);

}

private void Slon\_i()

{

int lineelEphant, columelEphant, linePos, columPos;//lineelEphant-строка columelEphant2-столбец

lineelEphant = y;

columelEphant = x;

int Z\_old = Z;

int i = 0;

do

{

do

{

mKey = Console.ReadKey(true);

switch (mKey.Key)

{

case ConsoleKey.W:

if (y > 0)

{

y--;

Console.SetCursorPosition(x, y);

}

break;

case ConsoleKey.A:

if (x > 0)

{

x = x - 2;

Console.SetCursorPosition(x, y);

Z--;

}

break;

case ConsoleKey.S:

if (y < 7)

{

y++;

Console.SetCursorPosition(x, y);

}

break;

case ConsoleKey.D:

if (x < 7 \* 2)

{

x = x + 2;

Console.SetCursorPosition(x, y);

Z++;

}

break;

default:

break;

}

} while (mKey.Key != ConsoleKey.Enter);

linePos = y;

columPos = x;

if (Proverka\_i() == true)

{

if (Math.Abs(lineelEphant - linePos) == Math.Abs((columelEphant - Z\_old) - (columPos - Z)))

{

if (Slon(Old\_y: lineelEphant, Old\_x: columelEphant, Old\_Z: Z\_old, New\_x: columPos, New\_y: linePos, New\_Z: Z))//Поменять местави х и у

{

mass[lineelEphant, columelEphant - Z\_old] = " ";

Console.SetCursorPosition(columelEphant, lineelEphant);

if ((((columelEphant - Z\_old) + lineelEphant) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[lineelEphant, columelEphant - Z\_old]);

mass[linePos, columPos - Z] = "\u2657 ";

Console.SetCursorPosition(columPos, linePos);

if ((((columPos - Z) + linePos) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[linePos, columPos - Z]);

i++;

}

}

}

} while (i == 0);

}

private void Ladia\_i()

{

int lineRook, columRook, linePos, columPos;//lineRook-строка columRook2-столбец

lineRook = y;

columRook = x;

int Z\_old = Z;

int i = 0;

do

{

do

{

mKey = Console.ReadKey(true);

switch (mKey.Key)

{

case ConsoleKey.W:

if (y > 0)

{

y--;

Console.SetCursorPosition(x, y);

}

break;

case ConsoleKey.A:

if (x > 0)

{

x = x - 2;

Console.SetCursorPosition(x, y);

Z--;

}

break;

case ConsoleKey.S:

if (y < 7)

{

y++;

Console.SetCursorPosition(x, y);

}

break;

case ConsoleKey.D:

if (x < 7 \* 2)

{

x = x + 2;

Console.SetCursorPosition(x, y);

Z++;

}

break;

default:

break;

}

} while (mKey.Key != ConsoleKey.Enter);

linePos = y;

columPos = x;

if (Proverka\_i() == true)

{

if ((lineRook - linePos == 0) || ((columRook - Z\_old) - (columPos - Z) == 0))

{

if (Ladia(Old\_y: lineRook, Old\_x: columRook, Old\_Z: Z\_old, New\_x: columPos, New\_y: linePos, New\_Z: Z))//Поменять местави х и у

{

mass[lineRook, columRook - Z\_old] = " ";

Console.SetCursorPosition(columRook, lineRook);

if ((((columRook - Z\_old) + lineRook) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[lineRook, columRook - Z\_old]);

mass[linePos, columPos - Z] = "\u2656 ";

Console.SetCursorPosition(columPos, linePos);

if ((((columPos - Z) + linePos) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[linePos, columPos - Z]);

i++;

}

}

}

} while (i == 0);

}

private void COVID\_19()

{

int lineQueen, columQueen, linePos, columPos;//lineQueen-строка columQueen2-столбец

lineQueen = y;

columQueen = x;

int Z\_old = Z;

int i = 0;

do

{

do

{

mKey = Console.ReadKey(true);

switch (mKey.Key)

{

case ConsoleKey.W:

if (y > 0)

{

y--;

Console.SetCursorPosition(x, y);

}

break;

case ConsoleKey.A:

if (x > 0)

{

x = x - 2;

Console.SetCursorPosition(x, y);

Z--;

}

break;

case ConsoleKey.S:

if (y < 7)

{

y++;

Console.SetCursorPosition(x, y);

}

break;

case ConsoleKey.D:

if (x < 7 \* 2)

{

x = x + 2;

Console.SetCursorPosition(x, y);

Z++;

}

break;

default:

break;

}

} while (mKey.Key != ConsoleKey.Enter);

linePos = y;

columPos = x;

if (Proverka\_i() == true)

{

if ((Math.Abs(lineQueen - linePos) == Math.Abs((columQueen - Z\_old) - (columPos - Z))) || (lineQueen == linePos) || (columQueen - Z\_old == columPos - Z))

{

if ( Slon(Old\_y: lineQueen, Old\_x: columQueen, Old\_Z: Z\_old, New\_x: columPos, New\_y: linePos, New\_Z: Z) && Ladia(Old\_y: lineQueen, Old\_x: columQueen, Old\_Z: Z\_old, New\_x: columPos, New\_y: linePos, New\_Z: Z))//Поменять местави х и у

{

mass[lineQueen, columQueen - Z\_old] = " ";

Console.SetCursorPosition(columQueen, lineQueen);

if ((((columQueen - Z\_old) + lineQueen) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[lineQueen, columQueen - Z\_old]);

mass[linePos, columPos - Z] = "\u2655 ";

Console.SetCursorPosition(columPos, linePos);

if ((((columPos - Z) + linePos) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[linePos, columPos - Z]);

i++;

}

}

}

} while (i != 1);

}

private void COVID\_20()

{

int line, colum;

int lineKing, columKing, linePos, columPos;//lineKing-строка columKing-столбец

lineKing = y;

columKing = x;

int Z\_old = Z;

int i = 0;

do

{

do

{

mKey = Console.ReadKey(true);

switch (mKey.Key)

{

case ConsoleKey.W:

if (y > 0)

{

y--;

Console.SetCursorPosition(x, y);

}

break;

case ConsoleKey.A:

if (x > 0)

{

x = x - 2;

Console.SetCursorPosition(x, y);

Z--;

}

break;

case ConsoleKey.S:

if (y < 7)

{

y++;

Console.SetCursorPosition(x, y);

}

break;

case ConsoleKey.D:

if (x < 7 \* 2)

{

x = x + 2;

Console.SetCursorPosition(x, y);

Z++;

}

break;

default:

break;

}

} while (mKey.Key != ConsoleKey.Enter);

linePos = y;

columPos = x;

if (Proverka\_i() == true && Proverka\_AI\_COVID\_20() == true)

{

linePos = y;

columPos = x;

line = linePos - lineKing;

colum = columPos - columKing;

if (((Math.Abs(line) == 1) && (Math.Abs(colum) == 0)) || ((Math.Abs(line) == 1) && (Math.Abs(colum) == 1)) || ((Math.Abs(line) == 0) && (Math.Abs(colum) == 1)))

{

mass[lineKing, columKing - Z\_old] = " ";

Console.SetCursorPosition(columKing, lineKing);

if ((((columKing - Z\_old) + lineKing) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[lineKing, columKing - Z\_old]);

mass[linePos, columPos - Z] = "\u2654 ";

Console.SetCursorPosition(columPos, linePos);

if ((((columPos - Z) + linePos) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[linePos, columPos - Z]);

i++;

}

}

} while (i == 0);

}

private bool Proverka\_AI\_COVID\_20()

{

bool Vilat = false;

if (mass[y, (x - Z)] != "\u265a " && mass[y, (x - Z)] != "\u265b " && mass[y, (x - Z)] != "\u265c " && mass[y, (x - Z)] != "\u265d " && mass[y, (x - Z)] != "\u265e " && mass[y, (x - Z)] != "\u265F ")

{

Vilat = true;

}

return Vilat;

}

private bool Proverka\_i()

{

bool Virsuta = false;

if (mass[y, (x - Z)] != "\u2659 " && mass[y, (x - Z)] != "\u2658 " && mass[y, (x - Z)] != "\u2657 " && mass[y, (x - Z)] != "\u2656 " && mass[y, (x - Z)] != "\u2655 " && mass[y, (x - Z)] != "\u2654 ")

{

Virsuta = true;

}

return Virsuta;

}

public void AI\_Figura\_vibor()

{

int i = 0;

do

{

int AI\_x = rand.Next(0, 8);

int AI\_y = rand.Next(0, 8);

int AI\_reserve = AI\_x;

if (mass[AI\_y, AI\_x] == "\u265F ")//Пешка

{

AI\_x = AI\_x \* 2;

Pehuia\_AI(AI\_x, AI\_y, AI\_reserve);

i++;

}

else if (mass[AI\_y, AI\_x] == "\u265e ")//конь

{

AI\_x = AI\_x \* 2;

Kanina\_AI(AI\_x, AI\_y, AI\_reserve);

i++;

}

else if (mass[AI\_y, AI\_x] == "\u265d ")//Слон

{

AI\_x = AI\_x \* 2;

Slon\_AI(AI\_x, AI\_y, AI\_reserve);

i++;

}

else if (mass[AI\_y, AI\_x] == "\u265c ")//Ладья

{

AI\_x = AI\_x \* 2;

Ladia\_AI(AI\_x, AI\_y, AI\_reserve);

i++;

}

else if (mass[AI\_y, AI\_x] == "\u265b ")//Королева

{

AI\_x = AI\_x \* 2;

COVID\_19\_AI(AI\_x, AI\_y, AI\_reserve);

i++;

}

else if (mass[AI\_y, AI\_x] == "\u265a ")//Король

{

AI\_x = AI\_x \* 2;

COVID\_20\_AI(AI\_x, AI\_y, AI\_reserve);

i++;

}

}

while (i != 1);

}

private void Pehuia\_AI(int AI\_x, int AI\_y, int AI\_reserve)

{

int add\_UD = 0, add\_LD = 0;

if (mass[AI\_x + 1, AI\_y + 1] == "\u2654 " || mass[AI\_x + 1, AI\_y + 1] == "\u2655 " || mass[AI\_x + 1, AI\_y + 1] == "\u2656 " || mass[AI\_x + 1, AI\_y + 1] == "\u2657 " || mass[AI\_x + 1, AI\_y + 1] == "\u2658 " || mass[AI\_x + 1, AI\_y + 1] == "\u2659 " || mass[AI\_x - 1, AI\_y + 1] == "\u2654 " || mass[AI\_x - 1, AI\_y + 1] == "\u2655 " || mass[AI\_x - 1, AI\_y + 1] == "\u2656 " || mass[AI\_x - 1, AI\_y + 1] == "\u2657 " || mass[AI\_x - 1, AI\_y + 1] == "\u2658 " || mass[AI\_x - 1, AI\_y + 1] == "\u2659 ")

{

switch (mass[AI\_x + 1, AI\_y + 1])

{

case "\u2655 ":add\_LD = 1;add\_UD = 1;break;

case "\u2654 ": add\_LD = 1; add\_UD = 1;break;

case "\u2657 ": add\_LD = 1; add\_UD = 1;break;

case "\u2658 ": add\_LD = 1; add\_UD = 1;break;

case "\u2656 ": add\_LD = 1; add\_UD = 1;break;

case "\u2659 ": add\_LD = 1; add\_UD = 1;break;

}

switch (mass[AI\_x - 1, AI\_y + 1])

{

case "\u2655 ": add\_LD = -1; add\_UD = 1; break;

case "\u2654 ": add\_LD = -1; add\_UD = 1; break;

case "\u2657 ": add\_LD = -1; add\_UD = 1; break;

case "\u2658 ": add\_LD = -1; add\_UD = 1; break;

case "\u2656 ": add\_LD = -1; add\_UD = 1; break;

case "\u2659 ": add\_LD = -1; add\_UD = 1; break;

}

}

else if (AI\_x == 0 && AI\_y == 1 || AI\_x == 2 && AI\_y == 1 || AI\_x == 4 && AI\_y == 1 || AI\_x == 6 && AI\_y == 1 || AI\_x == 8 && AI\_y == 1 || AI\_x == 10 && AI\_y == 1 || AI\_x == 12 && AI\_y == 1 || AI\_x == 14 && AI\_y == 1)

{

add\_UD = rand.Next(1, 3);

}

else

{

add\_UD = 1;

}

if (Proverka\_AI(AI\_x, AI\_y, AI\_reserve) == true)

{

if (((AI\_reserve + AI\_y) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

mass[AI\_y, AI\_reserve] = " ";

Console.SetCursorPosition(AI\_x, AI\_y);

Console.Write(mass[AI\_y, AI\_reserve]);

AI\_y += add\_UD;

AI\_x += (add\_LD \* 2);

AI\_reserve += add\_LD;

Console.SetCursorPosition(AI\_x, AI\_y);

mass[AI\_y, AI\_reserve] = "\u265F ";

if (((AI\_reserve + AI\_y) % 2) == 0)

{

Console.BackgroundColor = ConsoleColor.White;

Console.Write(mass[AI\_y, AI\_reserve]);

}

else

{

Console.BackgroundColor = ConsoleColor.Black;

Console.Write(mass[AI\_y, AI\_reserve]);

}

}

}

private void Kanina\_AI(int AI\_x, int AI\_y, int AI\_reserve)

{

int lineHorse, columHorse, linePos, columPos;//lineHorse-строка columHorse2-столбец

lineHorse = AI\_y;

columHorse = AI\_x;

int Z\_old = AI\_reserve;

int i = 0;

int Z\_New;

do

{

AI\_x = rand.Next(0, 8);

AI\_y = rand.Next(0, 8);

AI\_reserve = AI\_x;

AI\_x = AI\_x \* 2;

linePos = AI\_y;

columPos = AI\_x;

Z\_New = AI\_reserve;

if (Proverka\_AI(AI\_x, AI\_y, AI\_reserve) == true)

{

if (((Math.Abs(linePos - lineHorse) == 2) && (Math.Abs((columPos - Z\_New) - (columHorse - Z\_old)) == 1)) || ((Math.Abs(linePos - lineHorse) == 1) && (Math.Abs((columPos - Z\_New) - (columHorse - Z\_old)) == 2)))

{

mass[lineHorse, columHorse - Z\_old] = " ";

Console.SetCursorPosition(columHorse, lineHorse);

if ((((columHorse - Z\_old) + lineHorse) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[lineHorse, columHorse - Z\_old]);

mass[linePos, columPos - Z\_New] = "\u265e ";

Console.SetCursorPosition(columPos, linePos);

if ((((columPos - Z\_New) + linePos) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[linePos, columPos - Z\_New]);

i++;

}

}

} while (i == 0);

}

private void Slon\_AI(int AI\_x, int AI\_y, int AI\_reserve)

{

int lineelEphant, columelEphant, linePos, columPos;//lineelEphant-строка columelEphant2-столбец

lineelEphant = AI\_y;

columelEphant = AI\_x;

int Z\_old = AI\_reserve;

int i = 0;

int Z\_New;

do

{

AI\_x = rand.Next(0, 8);

AI\_y = rand.Next(0, 8);

AI\_reserve = AI\_x;

AI\_x = AI\_x \* 2;

linePos = AI\_y;

columPos = AI\_x;

Z\_New = AI\_reserve;

if (Proverka\_AI(AI\_x, AI\_y, AI\_reserve) == true)

{

if (Math.Abs(lineelEphant - linePos) == Math.Abs((columelEphant - Z\_old) - (columPos - Z\_New)))

{

if (Slon(Old\_y: lineelEphant, Old\_x: columelEphant, Old\_Z: Z\_old, New\_x: columPos, New\_y: linePos, New\_Z: Z\_New))//Поменять местави х и у

{

mass[lineelEphant, columelEphant - Z\_old] = " ";

Console.SetCursorPosition(columelEphant, lineelEphant);

if ((((columelEphant - Z\_old) + lineelEphant) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[lineelEphant, columelEphant - Z\_old]);

mass[linePos, columPos - Z\_New] = "\u265d ";

Console.SetCursorPosition(columPos, linePos);

if ((((columPos - Z\_New) + linePos) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[linePos, columPos - Z\_New]);

i++;

}

}

}

} while (i == 0);

}

private void Ladia\_AI(int AI\_x, int AI\_y, int AI\_reserve)

{

int lineRook, columRook, linePos, columPos;//lineRook-строка columRook2-столбец

lineRook = AI\_y;

columRook = AI\_x;

int Z\_old = AI\_reserve;

int i = 0;

int Z\_New;

do

{

AI\_x = rand.Next(0, 8);

AI\_y = rand.Next(0, 8);

AI\_reserve = AI\_x;

AI\_x = AI\_x \* 2;

linePos = AI\_y;

columPos = AI\_x;

Z\_New = AI\_reserve;

if (Proverka\_AI(AI\_x, AI\_y, AI\_reserve) == true)

{

if ((lineRook - linePos == 0) || ((columRook - Z\_old) - (columPos - Z) == 0))

{

if (Ladia(Old\_y: lineRook, Old\_x: columRook, Old\_Z: Z\_old, New\_x: columPos, New\_y: linePos, New\_Z: Z\_New))//Поменять местави х и у

{

mass[lineRook, columRook - Z\_old] = " ";

Console.SetCursorPosition(columRook, lineRook);

if ((((columRook - Z\_old) + lineRook) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[lineRook, columRook - Z\_old]);

mass[linePos, columPos - Z\_New] = "\u265c ";

Console.SetCursorPosition(columPos, linePos);

if ((((columPos - Z\_New) + linePos) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[linePos, columPos - Z\_New]);

i++;

}

}

}

} while (i == 0);

}

private void COVID\_19\_AI(int AI\_x, int AI\_y, int AI\_reserve)

{

int lineQueen, columQueen, linePos, columPos;//lineQueen-строка columQueen2-столбец

lineQueen = AI\_y;

columQueen = AI\_x;

int Z\_old = AI\_reserve;

int i = 0;

int Z\_New;

do

{

AI\_x = rand.Next(0, 8);

AI\_y = rand.Next(0, 8);

AI\_reserve = AI\_x;

AI\_x = AI\_x \* 2;

linePos = AI\_y;

columPos = AI\_x;

Z\_New = AI\_reserve;

if (Proverka\_AI(AI\_x, AI\_y, AI\_reserve) == true)

{

if ((Math.Abs(lineQueen - linePos) == Math.Abs((columQueen - Z\_old) - (columPos - Z\_New))) || (lineQueen == linePos) || (columQueen - Z\_old == columPos - Z\_New))

{

if (Slon(Old\_y: lineQueen, Old\_x: columQueen, Old\_Z: Z\_old, New\_x: columPos, New\_y: linePos, New\_Z: Z\_New) && Ladia(Old\_y: lineQueen, Old\_x: columQueen, Old\_Z: Z\_old, New\_x: columPos, New\_y: linePos, New\_Z: Z\_New))//Поменять местави х и у

{

mass[lineQueen, columQueen - Z\_old] = " ";

Console.SetCursorPosition(columQueen, lineQueen);

if ((((columQueen - Z\_old) + lineQueen) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[lineQueen, columQueen - Z\_old]);

mass[linePos, columPos - Z\_New] = "\u265b ";

Console.SetCursorPosition(columPos, linePos);

if ((((columPos - Z\_New) + linePos) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[linePos, columPos - Z\_New]);

i++;

}

}

}

} while (i != 1);

}

private void COVID\_20\_AI(int AI\_x, int AI\_y, int AI\_reserve)

{

int line, colum;

int lineKing, columKing, linePos, columPos;//lineKing-строка columKing-столбец

lineKing = AI\_y;

columKing = AI\_x;

int Z\_old = AI\_reserve;

int i = 0;

int Z\_New;

do

{

AI\_x = rand.Next(0, 8);

AI\_y = rand.Next(0, 8);

AI\_reserve = AI\_x;

AI\_x = AI\_x \* 2;

linePos = AI\_y;

columPos = AI\_x;

Z\_New = AI\_reserve;

if (Proverka\_AI(AI\_x, AI\_y, AI\_reserve) == true && Proverka\_COVID\_20(AI\_x, AI\_y, AI\_reserve))

{

line = linePos - lineKing;

colum = columPos - columKing;

if (((Math.Abs(line) == 1) && (Math.Abs(colum) == 0)) || ((Math.Abs(line) == 1) && (Math.Abs(colum) == 1)) || ((Math.Abs(line) == 0) && (Math.Abs(colum) == 1)))

{

mass[lineKing, columKing - Z\_old] = " ";

Console.SetCursorPosition(columKing, lineKing);

if ((((columKing - Z\_old) + lineKing) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[lineKing, columKing - Z\_old]);

mass[linePos, columPos - Z\_New] = "\u265a ";

Console.SetCursorPosition(columPos, linePos);

if ((((columPos - Z\_New) + linePos) % 2) != 0)

{ Console.BackgroundColor = ConsoleColor.Black; }

else

{ Console.BackgroundColor = ConsoleColor.White; }

Console.Write(mass[linePos, columPos - Z\_New]);

i++;

}

}

} while (i == 0);

}

private bool Proverka\_AI(int AI\_x, int AI\_y, int AI\_Z)

{

bool Vilat = false;

if (mass[AI\_y, (AI\_x - AI\_Z)] != "\u265a " && mass[AI\_y, (AI\_x - AI\_Z)] != "\u265b " && mass[AI\_y, (AI\_x - AI\_Z)] != "\u265c " && mass[AI\_y, (AI\_x - AI\_Z)] != "\u265d " && mass[AI\_y, (AI\_x - AI\_Z)] != "\u265e " && mass[AI\_y, (AI\_x - AI\_Z)] != "\u265F ")

{

Vilat = true;

}

return Vilat;

}

private bool Proverka\_COVID\_20(int AI\_x, int AI\_y, int AI\_Z)

{

bool Virsuta = false;

if (mass[AI\_y, (AI\_x - AI\_Z)] != "\u2659 " && mass[AI\_y, (AI\_x - AI\_Z)] != "\u2658 " && mass[AI\_y, (AI\_x - AI\_Z)] != "\u2657 " && mass[AI\_y, (AI\_x - AI\_Z)] != "\u2656 " && mass[AI\_y, (AI\_x - AI\_Z)] != "\u2655 " && mass[AI\_y, (AI\_x - AI\_Z)] != "\u2654 ")

{

Virsuta = true;

}

return Virsuta;

}

public void Ender\_mir()

{

int X = 0, Y = 0, Z\_OLD = 0;

for (int i = 0; i < 8; i++)

{

int beck = 0;

for (int f = 0; f < 8; f++)

{

if (mass[i, f] == "\u2654 ")

{

X = i;

Y = f;

beck = 1;

break;

}

}

if (beck == 1)

{

break;

}

}

Z\_OLD = X;

X = X \* 2;

if (Vacsina(X, Y, Z\_OLD) == true)

{

Console.SetCursorPosition(0, 8);

Console.WriteLine("Конец игры вы проиграли");

}

for (int i = 0; i < 8; i++)

{

int beck = 0;

for (int f = 0; f < 8; f++)

{

if (mass[i, f] == "\u265a ")

{

X = i;

Y = f;

beck = 1;

break;

}

}

if (beck == 1)

{

break;

}

}

Z\_OLD = X;

X = X \* 2;

if (Vacsina(X, Y, Z\_OLD) == true)

{

Console.SetCursorPosition(0, 8);

Console.WriteLine("Конец игры вы победили");

}

}

private bool Vacsina(int AI\_x, int AI\_y, int AI\_reserve)

{

bool Game = false;

if (Game == false)

{

if (mass[AI\_x, AI\_y] == "\u2654 ")

{

var block\_1 = true;

var block\_2 = true;

var block\_3 = true;

var block\_4 = true;

var block\_5 = true;

var block\_6 = true;

var block\_7 = true;

var block\_8 = true;

for (int i = 0; i < 8; i++)

{

for (int g = 0; g < 8; g++)

{

switch (mass[i, g])

{

case "\u2659 ":

for (int p = 0; i < 8; i++)

{

int long\_x = i, long\_y = g,long\_z;

if (p == 0 && block\_1 == true)

{

AI\_reserve = AI\_x;

long\_y = AI\_y - 1;

AI\_x = AI\_x \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else if (p == 1 && block\_2 == true)

{

AI\_reserve = AI\_x + 1;

long\_y = AI\_y - 1;

AI\_x = (AI\_x + 1) \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else if (p == 2 && block\_3 == true)

{

AI\_reserve = AI\_x + 1;

long\_y = AI\_y;

AI\_x = (AI\_x + 1) \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else if (p == 3 && block\_4 == true)

{

AI\_reserve = AI\_x + 1;

long\_y = AI\_y + 1;

AI\_x = (AI\_x + 1) \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else if (p == 4 && block\_5 == true)

{

AI\_reserve = AI\_x;

long\_y = AI\_y + 1;

AI\_x = AI\_x \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else if (p == 5 && block\_6 == true)

{

AI\_reserve = AI\_x - 1;

long\_y = AI\_y + 1;

AI\_x = (AI\_x - 1) \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else if (p == 6 && block\_7 == true)

{

AI\_reserve = AI\_x - 1;

long\_y = AI\_y;

AI\_x = (AI\_x - 1) \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else if (p == 7 && block\_8 == true)

{

AI\_reserve = AI\_x - 1;

long\_y = AI\_y - 1;

AI\_x = (AI\_x - 1) \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else

{

AI\_reserve = 0;

long\_y = 0;

AI\_x = 0;

long\_x = 0;

long\_z = 0;

}

if (mass[long\_x + 1, long\_y + 1] == "\u2654 " || mass[long\_x - 1, long\_y + 1] == "\u2654 ")

{

if (p == 0)

{

block\_1 = false;

}

else if (p == 1)

{

block\_2 = false;

}

else if (p == 2)

{

block\_3 = false;

}

else if (p == 3)

{

block\_4 = false;

}

else if (p == 4)

{

block\_5 = false;

}

else if (p == 5)

{

block\_6 = false;

}

else if (p == 6)

{

block\_7 = false;

}

else if (p == 7)

{

block\_8 = false;

}

}

};

break;//Пешка

case "\u2658 ":

int lineHorse, columHorse, linePos, columPos;//lineHorse-строка columHorse2-столбец

lineHorse = g;//y

int Z\_old = i;

columHorse = i \* 2;//x

int Z\_New;

for (int p = 0; i < 8; i++)

{

if (p == 0 && block\_1 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y - 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 1 && block\_2 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 2 && block\_3 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 3 && block\_4 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 4 && block\_5 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y + 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 5 && block\_6 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 6 && block\_7 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 7 && block\_8 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else

{

AI\_reserve = 0;

linePos = 0;

AI\_x = 0;

columPos = 0;

Z\_New = 0;

}

if (((Math.Abs(linePos - lineHorse) == 2) && (Math.Abs((columPos - Z\_New) - (columHorse - Z\_old)) == 1)) || ((Math.Abs(linePos - lineHorse) == 1) && (Math.Abs((columPos - Z\_New) - (columHorse - Z\_old)) == 2)))

{

if (p == 0)

{

block\_1 = false;

}

else if (p == 1)

{

block\_2 = false;

}

else if (p == 2)

{

block\_3 = false;

}

else if (p == 3)

{

block\_4 = false;

}

else if (p == 4)

{

block\_5 = false;

}

else if (p == 5)

{

block\_6 = false;

}

else if (p == 6)

{

block\_7 = false;

}

else if (p == 7)

{

block\_8 = false;

}

}

};

break;//конь

case "\u2657 ":

int lineelEphant, columelEphant;//lineHorse-строка columHorse2-столбец

lineHorse = g;//y

columHorse = i \* 2;//x

lineelEphant = g;

columelEphant = i;

Z\_old = i;

for (int p = 0; i < 8; i++)

{

if (p == 0 && block\_1 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y - 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 1 && block\_2 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 2 && block\_3 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 3 && block\_4 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 4 && block\_5 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y + 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 5 && block\_6 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 6 && block\_7 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 7 && block\_8 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else

{

AI\_reserve = 0;

linePos = 0;

AI\_x = 0;

columPos = 0;

Z\_New = 0;

}

if (Math.Abs(lineelEphant - linePos) == Math.Abs((columelEphant - Z\_old) - (columPos - Z\_New)))

{

if (p == 0)

{

block\_1 = false;

}

else if (p == 1)

{

block\_2 = false;

}

else if (p == 2)

{

block\_3 = false;

}

else if (p == 3)

{

block\_4 = false;

}

else if (p == 4)

{

block\_5 = false;

}

else if (p == 5)

{

block\_6 = false;

}

else if (p == 6)

{

block\_7 = false;

}

else if (p == 7)

{

block\_8 = false;

}

}

};

break;//Слон

case "\u2656 ":

int lineRook, columRook;//lineHorse-строка columHorse2-столбец

lineRook = g;//y

columRook = i \* 2;//x

Z\_old = i;

for (int p = 0; i < 8; i++)

{

if (p == 0 && block\_1 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y - 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 1 && block\_2 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 2 && block\_3 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 3 && block\_4 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 4 && block\_5 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y + 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 5 && block\_6 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 6 && block\_7 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 7 && block\_8 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else

{

AI\_reserve = 0;

linePos = 0;

AI\_x = 0;

columPos = 0;

Z\_New = 0;

}

if ((lineRook - linePos == 0) || ((columRook - Z\_old) - (columPos - Z) == 0))

{

if (p == 0)

{

block\_1 = false;

}

else if (p == 1)

{

block\_2 = false;

}

else if (p == 2)

{

block\_3 = false;

}

else if (p == 3)

{

block\_4 = false;

}

else if (p == 4)

{

block\_5 = false;

}

else if (p == 5)

{

block\_6 = false;

}

else if (p == 6)

{

block\_7 = false;

}

else if (p == 7)

{

block\_8 = false;

}

}

}; break;//Ладья

case "\u2655 ":

int lineQueen, columQueen;//lineHorse-строка columHorse2-столбец

lineQueen = g;//y

columQueen = i \* 2;//x

Z\_old = i;

for (int p = 0; i < 8; i++)

{

if (p == 0 && block\_1 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y - 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 1 && block\_2 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 2 && block\_3 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 3 && block\_4 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 4 && block\_5 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y + 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 5 && block\_6 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 6 && block\_7 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 7 && block\_8 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else

{

AI\_reserve = 0;

linePos = 0;

AI\_x = 0;

columPos = 0;

Z\_New = 0;

}

if ((Math.Abs(lineQueen - linePos) == Math.Abs((columQueen - Z\_old) - (columPos - Z))) || (lineQueen == linePos) || (columQueen - Z\_old == columPos - Z))

{

if (p == 0)

{

block\_1 = false;

}

else if (p == 1)

{

block\_2 = false;

}

else if (p == 2)

{

block\_3 = false;

}

else if (p == 3)

{

block\_4 = false;

}

else if (p == 4)

{

block\_5 = false;

}

else if (p == 5)

{

block\_6 = false;

}

else if (p == 6)

{

block\_7 = false;

}

else if (p == 7)

{

block\_8 = false;

}

}

}; break;//Королева

}

}

}

}

else if (mass[AI\_x, AI\_y] == "\u265a")

{

var block\_1 = true;

var block\_2 = true;

var block\_3 = true;

var block\_4 = true;

var block\_5 = true;

var block\_6 = true;

var block\_7 = true;

var block\_8 = true;

for (int i = 0; i < 8; i++)

{

for (int g = 0; g < 8; g++)

{

switch (mass[i, g])

{

case "\u265F ":

for (int p = 0; i < 8; i++)

{

int long\_x = i, long\_y = g, long\_z;

if (p == 0 && block\_1 == true)

{

AI\_reserve = AI\_x;

long\_y = AI\_y - 1;

AI\_x = AI\_x \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else if (p == 1 && block\_2 == true)

{

AI\_reserve = AI\_x + 1;

long\_y = AI\_y - 1;

AI\_x = (AI\_x + 1) \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else if (p == 2 && block\_3 == true)

{

AI\_reserve = AI\_x + 1;

long\_y = AI\_y;

AI\_x = (AI\_x + 1) \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else if (p == 3 && block\_4 == true)

{

AI\_reserve = AI\_x + 1;

long\_y = AI\_y + 1;

AI\_x = (AI\_x + 1) \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else if (p == 4 && block\_5 == true)

{

AI\_reserve = AI\_x;

long\_y = AI\_y + 1;

AI\_x = AI\_x \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else if (p == 5 && block\_6 == true)

{

AI\_reserve = AI\_x - 1;

long\_y = AI\_y + 1;

AI\_x = (AI\_x - 1) \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else if (p == 6 && block\_7 == true)

{

AI\_reserve = AI\_x - 1;

long\_y = AI\_y;

AI\_x = (AI\_x - 1) \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else if (p == 7 && block\_8 == true)

{

AI\_reserve = AI\_x - 1;

long\_y = AI\_y - 1;

AI\_x = (AI\_x - 1) \* 2;

long\_x = AI\_x;

long\_z = AI\_reserve;

}

else

{

AI\_reserve = 0;

long\_y = 0;

AI\_x = 0;

long\_x = 0;

long\_z = 0;

}

if (mass[long\_x + 1, long\_y + 1] == "\u265F " || mass[long\_x - 1, long\_y + 1] == "\u265F ")

{

if (p == 0)

{

block\_1 = false;

}

else if (p == 1)

{

block\_2 = false;

}

else if (p == 2)

{

block\_3 = false;

}

else if (p == 3)

{

block\_4 = false;

}

else if (p == 4)

{

block\_5 = false;

}

else if (p == 5)

{

block\_6 = false;

}

else if (p == 6)

{

block\_7 = false;

}

else if (p == 7)

{

block\_8 = false;

}

}

}; ; break;//Пешка

case "\u265b ":

int lineQueen, columQueen, linePos, columPos, Z\_old, Z\_New;//lineHorse-строка columHorse2-столбец

lineQueen = g;//y

columQueen = i \* 2;//x

Z\_old = i;

for (int p = 0; i < 8; i++)

{

if (p == 0 && block\_1 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y - 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 1 && block\_2 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 2 && block\_3 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 3 && block\_4 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 4 && block\_5 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y + 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 5 && block\_6 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 6 && block\_7 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 7 && block\_8 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else

{

AI\_reserve = 0;

linePos = 0;

AI\_x = 0;

columPos = 0;

Z\_New = 0;

}

if ((Math.Abs(lineQueen - linePos) == Math.Abs((columQueen - Z\_old) - (columPos - Z))) || (lineQueen == linePos) || (columQueen - Z\_old == columPos - Z))

{

if (p == 0)

{

block\_1 = false;

}

else if (p == 1)

{

block\_2 = false;

}

else if (p == 2)

{

block\_3 = false;

}

else if (p == 3)

{

block\_4 = false;

}

else if (p == 4)

{

block\_5 = false;

}

else if (p == 5)

{

block\_6 = false;

}

else if (p == 6)

{

block\_7 = false;

}

else if (p == 7)

{

block\_8 = false;

}

}

}; ; break;//Королева

case "\u265c ":

int lineRook, columRook;//lineHorse-строка columHorse2-столбец

lineRook = g;//y

columRook = i \* 2;//x

Z\_old = i;

for (int p = 0; i < 8; i++)

{

if (p == 0 && block\_1 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y - 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 1 && block\_2 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 2 && block\_3 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 3 && block\_4 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 4 && block\_5 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y + 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 5 && block\_6 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 6 && block\_7 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 7 && block\_8 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else

{

AI\_reserve = 0;

linePos = 0;

AI\_x = 0;

columPos = 0;

Z\_New = 0;

}

if ((lineRook - linePos == 0) || ((columRook - Z\_old) - (columPos - Z) == 0))

{

if (p == 0)

{

block\_1 = false;

}

else if (p == 1)

{

block\_2 = false;

}

else if (p == 2)

{

block\_3 = false;

}

else if (p == 3)

{

block\_4 = false;

}

else if (p == 4)

{

block\_5 = false;

}

else if (p == 5)

{

block\_6 = false;

}

else if (p == 6)

{

block\_7 = false;

}

else if (p == 7)

{

block\_8 = false;

}

}

}

; break;//Ладья

case "\u265d ":

int lineelEphant, columelEphant;//lineHorse-строка columHorse2-столбец

lineelEphant = g;

columelEphant = i;

Z\_old = i;

for (int p = 0; i < 8; i++)

{

if (p == 0 && block\_1 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y - 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 1 && block\_2 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 2 && block\_3 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 3 && block\_4 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 4 && block\_5 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y + 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 5 && block\_6 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 6 && block\_7 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 7 && block\_8 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else

{

AI\_reserve = 0;

linePos = 0;

AI\_x = 0;

columPos = 0;

Z\_New = 0;

}

if (Math.Abs(lineelEphant - linePos) == Math.Abs((columelEphant - Z\_old) - (columPos - Z\_New)))

{

if (p == 0)

{

block\_1 = false;

}

else if (p == 1)

{

block\_2 = false;

}

else if (p == 2)

{

block\_3 = false;

}

else if (p == 3)

{

block\_4 = false;

}

else if (p == 4)

{

block\_5 = false;

}

else if (p == 5)

{

block\_6 = false;

}

else if (p == 6)

{

block\_7 = false;

}

else if (p == 7)

{

block\_8 = false;

}

}

}; break;//Слон

case "\u265e ":

int lineHorse, columHorse; //lineHorse-строка columHorse2-столбец

lineHorse = g;//y

Z\_old = i;

columHorse = i \* 2;//x

for (int p = 0; i < 8; i++)

{

if (p == 0 && block\_1 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y - 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 1 && block\_2 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 2 && block\_3 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 3 && block\_4 == true)

{

AI\_reserve = AI\_x + 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x + 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 4 && block\_5 == true)

{

AI\_reserve = AI\_x;

linePos = AI\_y + 1;

AI\_x = AI\_x \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 5 && block\_6 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y + 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 6 && block\_7 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else if (p == 7 && block\_8 == true)

{

AI\_reserve = AI\_x - 1;

linePos = AI\_y - 1;

AI\_x = (AI\_x - 1) \* 2;

columPos = AI\_x;

Z\_New = AI\_reserve;

}

else

{

AI\_reserve = 0;

linePos = 0;

AI\_x = 0;

columPos = 0;

Z\_New = 0;

}

if (((Math.Abs(linePos - lineHorse) == 2) && (Math.Abs((columPos - Z\_New) - (columHorse - Z\_old)) == 1)) || ((Math.Abs(linePos - lineHorse) == 1) && (Math.Abs((columPos - Z\_New) - (columHorse - Z\_old)) == 2)))

{

if (p == 0)

{

block\_1 = false;

}

else if (p == 1)

{

block\_2 = false;

}

else if (p == 2)

{

block\_3 = false;

}

else if (p == 3)

{

block\_4 = false;

}

else if (p == 4)

{

block\_5 = false;

}

else if (p == 5)

{

block\_6 = false;

}

else if (p == 6)

{

block\_7 = false;

}

else if (p == 7)

{

block\_8 = false;

}

}

}; break;//конь

}

}

}

}

}

return Game;

}

private bool Ladia (int Old\_x, int Old\_y,int Old\_Z, int New\_x, int New\_y, int New\_Z)

{

if (New\_x - Old\_x > 0)

{

for (int l = Old\_x+2; l < New\_x; l+=2)

{

int Neon\_z = Old\_Z;

if (mass[l - Neon\_z, Old\_y] == "\u2654 " || mass[l - Neon\_z, Old\_y] == "\u2655 " || mass[l - Neon\_z, Old\_y] == "\u2656 " || mass[l - Neon\_z, Old\_y] == "\u2657 " || mass[l - Neon\_z, Old\_y] == "\u2658 " || mass[l - Neon\_z, Old\_y] == "\u2659 " || mass[l - Neon\_z, Old\_y] == "\u2654 " || mass[l - Neon\_z, Old\_y] == "\u2655 " || mass[l - Neon\_z, Old\_y] == "\u2656 " || mass[l - Neon\_z, Old\_y] == "\u2657 " || mass[l - Neon\_z, Old\_y] == "\u2658 " || mass[l - Neon\_z, Old\_y] == "\u2659 " || mass[l - Neon\_z, Old\_y] == "\u265b " || mass[l - Neon\_z, Old\_y] == "\u265a " || mass[l - Neon\_z, Old\_y] == "\u265d " || mass[l - Neon\_z, Old\_y] == "\u265e " || mass[l - Neon\_z, Old\_y] == "\u265c " || mass[l - Neon\_z, Old\_y] == "\u265F " || mass[l - Neon\_z, Old\_y] == "\u265b " || mass[l - Neon\_z, Old\_y] == "\u265a " || mass[l - Neon\_z, Old\_y] == "\u265d " || mass[l - Neon\_z, Old\_y] == "\u265e " || mass[l - Neon\_z, Old\_y] == "\u265c " || mass[l - Neon\_z, Old\_y] == "\u265F ")

{

return false;

}

Neon\_z++;

}

}

else if (New\_x - Old\_x < 0)

{

for (int l = Old\_x - 2; l > New\_x; l -= 2)

{

int Neon\_z = Old\_Z;

if (mass[l - Neon\_z, Old\_y] == "\u2654 " || mass[l - Neon\_z, Old\_y] == "\u2655 " || mass[l - Neon\_z, Old\_y] == "\u2656 " || mass[l - Neon\_z, Old\_y] == "\u2657 " || mass[l - Neon\_z, Old\_y] == "\u2658 " || mass[l - Neon\_z, Old\_y] == "\u2659 " || mass[l - Neon\_z, Old\_y] == "\u2654 " || mass[l - Neon\_z, Old\_y] == "\u2655 " || mass[l - Neon\_z, Old\_y] == "\u2656 " || mass[l - Neon\_z, Old\_y] == "\u2657 " || mass[l - Neon\_z, Old\_y] == "\u2658 " || mass[l - Neon\_z, Old\_y] == "\u2659 " || mass[l - Neon\_z, Old\_y] == "\u265b " || mass[l - Neon\_z, Old\_y] == "\u265a " || mass[l - Neon\_z, Old\_y] == "\u265d " || mass[l - Neon\_z, Old\_y] == "\u265e " || mass[l - Neon\_z, Old\_y] == "\u265c " || mass[l - Neon\_z, Old\_y] == "\u265F " || mass[l - Neon\_z, Old\_y] == "\u265b " || mass[l - Neon\_z, Old\_y] == "\u265a " || mass[l - Neon\_z, Old\_y] == "\u265d " || mass[l - Neon\_z, Old\_y] == "\u265e " || mass[l - Neon\_z, Old\_y] == "\u265c " || mass[l - Neon\_z, Old\_y] == "\u265F ")

{

return false;

}

Neon\_z--;

}

}

else if (New\_y - Old\_y > 0)

{

for (int l = Old\_y + 1; l < New\_y; l ++)

{

int Neon\_z = Old\_x - Old\_Z;

if (mass[Neon\_z, l] == "\u2654 " || mass[Neon\_z, l] == "\u2655 " || mass[Neon\_z, l] == "\u2656 " || mass[Neon\_z, l] == "\u2657 " || mass[Neon\_z, l] == "\u2658 " || mass[Neon\_z, l] == "\u2659 " || mass[Neon\_z, l] == "\u2654 " || mass[Neon\_z, l] == "\u2655 " || mass[Neon\_z, l] == "\u2656 " || mass[Neon\_z, l] == "\u2657 " || mass[Neon\_z, l] == "\u2658 " || mass[Neon\_z, l] == "\u2659 " || mass[Neon\_z, l] == "\u265b " || mass[Neon\_z, l] == "\u265a " || mass[Neon\_z, l] == "\u265d " || mass[Neon\_z, l] == "\u265e " || mass[Neon\_z, l] == "\u265c " || mass[Neon\_z, l] == "\u265F " || mass[Neon\_z, l] == "\u265b " || mass[Neon\_z, l] == "\u265a " || mass[Neon\_z, l] == "\u265d " || mass[Neon\_z, l] == "\u265e " || mass[Neon\_z, l] == "\u265c " || mass[Neon\_z, l] == "\u265F ")

{

return false;

}

}

}

else if (New\_y - Old\_y < 0)

{

for (int l = Old\_y - 1; l > New\_y; l--)

{

int Neon\_z = Old\_x - Old\_Z;

if (mass[Neon\_z, l] == "\u2654 " || mass[Neon\_z, l] == "\u2655 " || mass[Neon\_z, l] == "\u2656 " || mass[Neon\_z, l] == "\u2657 " || mass[Neon\_z, l] == "\u2658 " || mass[Neon\_z, l] == "\u2659 " || mass[Neon\_z, l] == "\u2654 " || mass[Neon\_z, l] == "\u2655 " || mass[Neon\_z, l] == "\u2656 " || mass[Neon\_z, l] == "\u2657 " || mass[Neon\_z, l] == "\u2658 " || mass[Neon\_z, l] == "\u2659 " || mass[Neon\_z, l] == "\u265b " || mass[Neon\_z, l] == "\u265a " || mass[Neon\_z, l] == "\u265d " || mass[Neon\_z, l] == "\u265e " || mass[Neon\_z, l] == "\u265c " || mass[Neon\_z, l] == "\u265F " || mass[Neon\_z, l] == "\u265b " || mass[Neon\_z, l] == "\u265a " || mass[Neon\_z, l] == "\u265d " || mass[Neon\_z, l] == "\u265e " || mass[Neon\_z, l] == "\u265c " || mass[Neon\_z, l] == "\u265F ")

{

return false;

}

}

}

return true;

}

private bool Slon (int Old\_x, int Old\_y, int Old\_Z, int New\_x, int New\_y, int New\_Z)

{

if (New\_x - Old\_x > 0 && New\_y - Old\_y <0)//Верх право

{

int Neon\_z = Old\_Z;

for (int l = Old\_x + 2; l < New\_x; l+=2)

{

Old\_y--;

if (mass[l - Neon\_z, Old\_y] == "\u2654 " || mass[l - Neon\_z, Old\_y] == "\u2655 " || mass[l - Neon\_z, Old\_y] == "\u2656 " || mass[l - Neon\_z, Old\_y] == "\u2657 " || mass[l - Neon\_z, Old\_y] == "\u2658 " || mass[l - Neon\_z, Old\_y] == "\u2659 " || mass[l - Neon\_z, Old\_y] == "\u2654 " || mass[l - Neon\_z, Old\_y] == "\u2655 " || mass[l - Neon\_z, Old\_y] == "\u2656 " || mass[l - Neon\_z, Old\_y] == "\u2657 " || mass[l - Neon\_z, Old\_y] == "\u2658 " || mass[l - Neon\_z, Old\_y] == "\u2659 " || mass[l - Neon\_z, Old\_y] == "\u265b " || mass[l - Neon\_z, Old\_y] == "\u265a " || mass[l - Neon\_z, Old\_y] == "\u265d " || mass[l - Neon\_z, Old\_y] == "\u265e " || mass[l - Neon\_z, Old\_y] == "\u265c " || mass[l - Neon\_z, Old\_y] == "\u265F " || mass[l - Neon\_z, Old\_y] == "\u265b " || mass[l - Neon\_z, Old\_y] == "\u265a " || mass[l - Neon\_z, Old\_y] == "\u265d " || mass[l - Neon\_z, Old\_y] == "\u265e " || mass[l - Neon\_z, Old\_y] == "\u265c " || mass[l - Neon\_z, Old\_y] == "\u265F ")

{

return false;

}

}

Neon\_z++;

}

else if (New\_x - Old\_x < 0 && New\_y - Old\_y > 0)//Низ лево

{

int Neon\_z = Old\_Z;

for (int l = Old\_x - 2; l < New\_x; l -= 2)

{

Old\_y++;

if (mass[l - Neon\_z, Old\_y] == "\u2654 " || mass[l - Neon\_z, Old\_y] == "\u2655 " || mass[l - Neon\_z, Old\_y] == "\u2656 " || mass[l - Neon\_z, Old\_y] == "\u2657 " || mass[l - Neon\_z, Old\_y] == "\u2658 " || mass[l - Neon\_z, Old\_y] == "\u2659 " || mass[l - Neon\_z, Old\_y] == "\u2654 " || mass[l - Neon\_z, Old\_y] == "\u2655 " || mass[l - Neon\_z, Old\_y] == "\u2656 " || mass[l - Neon\_z, Old\_y] == "\u2657 " || mass[l - Neon\_z, Old\_y] == "\u2658 " || mass[l - Neon\_z, Old\_y] == "\u2659 " || mass[l - Neon\_z, Old\_y] == "\u265b " || mass[l - Neon\_z, Old\_y] == "\u265a " || mass[l - Neon\_z, Old\_y] == "\u265d " || mass[l - Neon\_z, Old\_y] == "\u265e " || mass[l - Neon\_z, Old\_y] == "\u265c " || mass[l - Neon\_z, Old\_y] == "\u265F " || mass[l - Neon\_z, Old\_y] == "\u265b " || mass[l - Neon\_z, Old\_y] == "\u265a " || mass[l - Neon\_z, Old\_y] == "\u265d " || mass[l - Neon\_z, Old\_y] == "\u265e " || mass[l - Neon\_z, Old\_y] == "\u265c " || mass[l - Neon\_z, Old\_y] == "\u265F ")

{

return false;

}

Neon\_z--;

}

}

else if (New\_x - Old\_x > 0 && New\_y - Old\_y > 0)//Низ право

{

int Neon\_z = Old\_Z;

for (int l = Old\_x + 2; l < New\_x; l += 2)

{

Old\_y++;

if (mass[l - Neon\_z, Old\_y] == "\u2654 " || mass[l - Neon\_z, Old\_y] == "\u2655 " || mass[l - Neon\_z, Old\_y] == "\u2656 " || mass[l - Neon\_z, Old\_y] == "\u2657 " || mass[l - Neon\_z, Old\_y] == "\u2658 " || mass[l - Neon\_z, Old\_y] == "\u2659 " || mass[l - Neon\_z, Old\_y] == "\u2654 " || mass[l - Neon\_z, Old\_y] == "\u2655 " || mass[l - Neon\_z, Old\_y] == "\u2656 " || mass[l - Neon\_z, Old\_y] == "\u2657 " || mass[l - Neon\_z, Old\_y] == "\u2658 " || mass[l - Neon\_z, Old\_y] == "\u2659 " || mass[l - Neon\_z, Old\_y] == "\u265b " || mass[l - Neon\_z, Old\_y] == "\u265a " || mass[l - Neon\_z, Old\_y] == "\u265d " || mass[l - Neon\_z, Old\_y] == "\u265e " || mass[l - Neon\_z, Old\_y] == "\u265c " || mass[l - Neon\_z, Old\_y] == "\u265F " || mass[l - Neon\_z, Old\_y] == "\u265b " || mass[l - Neon\_z, Old\_y] == "\u265a " || mass[l - Neon\_z, Old\_y] == "\u265d " || mass[l - Neon\_z, Old\_y] == "\u265e " || mass[l - Neon\_z, Old\_y] == "\u265c " || mass[l - Neon\_z, Old\_y] == "\u265F ")

{

return false;

}

}

Neon\_z++;

}

else if (New\_x - Old\_x < 0 && New\_y - Old\_y < 0)//Верх лево

{

int Neon\_z = Old\_Z;

for (int l = Old\_x - 2; l < New\_x; l -= 2)

{

Old\_y--;

if (mass[l - Neon\_z, Old\_y] == "\u2654 " || mass[l - Neon\_z, Old\_y] == "\u2655 " || mass[l - Neon\_z, Old\_y] == "\u2656 " || mass[l - Neon\_z, Old\_y] == "\u2657 " || mass[l - Neon\_z, Old\_y] == "\u2658 " || mass[l - Neon\_z, Old\_y] == "\u2659 " || mass[l - Neon\_z, Old\_y] == "\u2654 " || mass[l - Neon\_z, Old\_y] == "\u2655 " || mass[l - Neon\_z, Old\_y] == "\u2656 " || mass[l - Neon\_z, Old\_y] == "\u2657 " || mass[l - Neon\_z, Old\_y] == "\u2658 " || mass[l - Neon\_z, Old\_y] == "\u2659 " || mass[l - Neon\_z, Old\_y] == "\u265b " || mass[l - Neon\_z, Old\_y] == "\u265a " || mass[l - Neon\_z, Old\_y] == "\u265d " || mass[l - Neon\_z, Old\_y] == "\u265e " || mass[l - Neon\_z, Old\_y] == "\u265c " || mass[l - Neon\_z, Old\_y] == "\u265F " || mass[l - Neon\_z, Old\_y] == "\u265b " || mass[l - Neon\_z, Old\_y] == "\u265a " || mass[l - Neon\_z, Old\_y] == "\u265d " || mass[l - Neon\_z, Old\_y] == "\u265e " || mass[l - Neon\_z, Old\_y] == "\u265c " || mass[l - Neon\_z, Old\_y] == "\u265F ")

{

return false;

}

}

Neon\_z--;

}

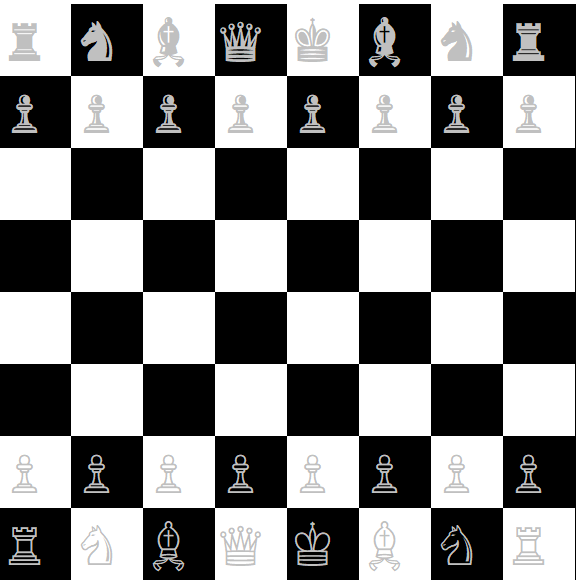
return true;

}

}

}

Скриншот работы программы:



Нужно доделать:

1.Провести тесты игры, выявив ошибки и недоработки

2.Решение выявленных ошибок и недоработок