МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ

УНИВЕРСИТЕТ им. Р.Е.АЛЕКСЕЕВА

Институт радиоэлектроники и информационных технологий

Кафедра информатики и систем управления

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Игра Уголки\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование темы проекта или работы)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовой работе

по дисциплине

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Алгоритмы и структуры данных\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(наименование дисциплины)

РУКОВОДИТЕЛЬ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (фамилия, и.,о.)

СТУДЕНТ:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (фамилия, и.,о.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(шифр группы)

Работа защищена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_

С оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород 2022

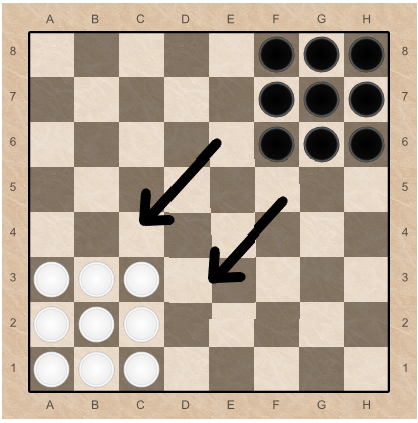
**Текст задачи:**

Цель игры переставить шашки из своего угла (дома) в угол соперника раньше, чем он построит свои в Вашем углу. Ходы делаются по очереди. Движение возможно по горизонталям и вертикалям только в направлениях к углу противника. За один ход передвигается не более одной шашки. Шашка может либо передвигаться на одно поле, либо перепрыгивать через одну или несколько шашек.

Игра заканчивается, когда один из игроков построил все свои шашки в доме противника. Все шашки необходимо вывести из своего дома не позднее чем за 3\*'количество шашек' ходов - иначе будет засчитан проигрыш.

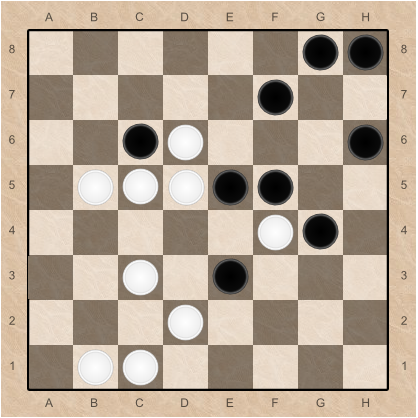
**Описание оценочной функции:**

Основная задача итогового алгоритма состоит в нахождении такого хода, который наиболее далеко продвинет фишку компьютера за один ход как можно ближе к финишу. Примем игру компьютера за игру за чёрных т.е. он будет ходить сверху вниз.

Для того, чтобы оценить состояние игры в текущий момент будем подсчитывать, насколько далеко фишка может продвинуться за n-ное количество ходов.

В данной игре важна такая позиция, при которой наша фишка может передвинуться не на одну клетку влево или вниз, а перепрыгивая через одну клетку, даже при перепрыгивании через одну фишку (неважно, чужую или собственную) т.к. это ход заменяет целых 3 обычных хода, а при нахождении такого хода, при перепрыгивании 2-х и фишек подряд, то вероятность победы гораздо возрастёт.

**Пример расчёта оценочной функции:**

Для нахождения оценки воспользуемся правилом: ход на одну клетку - 1 очко, ход через фишку – 3 очка.

Просчитаем для всех чёрных фишек оценки, которые покажут, насколько максимально далеко каждая фишка может продвинутся за 4 хода (4-величина, взятая рандомно, просматриваемое количество нужно будет подбирать в зависимости от уровня сложности).

Фишка на H8: 10 (перепрыгнуть чёрную, ещё раз чёрную, потом один обычный ход влево, перепрыгнуть через чёрную), аналогично, Фишка на G8: 9, Фишка на F7: 8, Фишка на C6: 8, Фишка на H6: 6, Фишка на E5: 8, Фишка на F5: 10, Фишка на G4: 10, Фишка на E3: 6.

Суммарная оценка соответственно будет равна 75(для белых 68) т.е. у чёрных фишек позиции лучше, а максимальная оценка будет равна 10. Именно фишки с этим числовым значением с определённой вероятностью могут продвинуться к финишу как можно ближе.

**Программный код:**

#include <iostream>  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
// Оценочная функция score //  
int score(int row, int col,int\*\* mass, int n, int ki, int kg){  
 int f1=n;  
 if(row>0 and mass[row-1][col]==0){ // Ход на одну влево  
 f1=f1+1; // Прибавление к   
 ki=ki+1; числу оценки +1  
 if(ki<4) {  
 f1=score(row-1, col,mass,f1,ki,kg); // Вызов

данной функции рекурсивно

для позиции слева  
 }  
 }  
 else{  
 if(row>1 and mass[row-2][col]==0){ // Ход через фишку . влево  
 f1=f1+3; // Прибавление к   
 ki=ki+1; числу оценки +3  
 if(ki<4) {  
 f1=score(row-2,col,mass,f1,ki,kg); // Вызов

данной функции рекурсивно

для позиции через фишку слева

}  
 }  
 }  
 int f2=n;  
 if(col>0 and mass[row][col-1]==0){ // Ход на одну вниз  
 f2=f2+1; // Прибавление к   
 kg=kg+1; числу оценки +1  
 if(kg<4) {  
 f2=score(row,col-1,mass,f2,ki,kg); // Вызов

данной функции рекурсивно для позиции вниз

}  
 }  
 else{  
 if(col>1 and mass[row][col-2]==0){ // Ход на через   
 f2=f2+3; фишку вниз // Прибавление к числу оценки +3  
 kg=kg+1;  
 if(kg<4) {  
 f2=score(row,col-2,mass,f2,ki,kg); // Вызов

данной функции для позиции через фишку вниз  
 }  
 }  
 }  
 if(f2>=f1){ // Блок выбора наибольшего числа оценки  
 return f2;  
 }  
 if(f2<f1){  
 return f1;  
 }  
}  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/