**Project 1**

**Checkers (Human V.S. Computer)**

**CIS-42480**

**Name: Daniel Dunn**

**Date: 4/18/2018**

**Introduction:**

Title: Checkers

This is a recreation of the board game of checkers. During the start of the program, you learn about the rules of the game and how it works. The computer’s pieces are marked with the letter “q” and the player is marked by the letter “h”.

After the start, the computer goes first, starting off the game by making it’s first move. Afterwards, the player’s chance to play begins, and they must put in a designated location for the piece, and where the piece is going to move.

For example: One of the pieces the player uses is on the 1 by 3 plain, and he can move it to the 2 by 4 plain during his move. All you simply have to do during your turn is to enter in the row, and the column.

**Summary:**

Project size: 275 lines

Number of variables: about 25

The number of method: 8

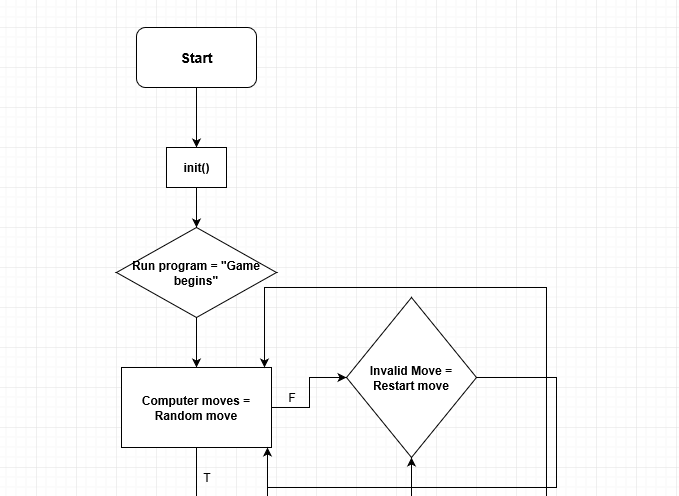
This project follows along with the chapters of the chapter book and can be extended/improved beyond what it can do now (graphics, better game control).

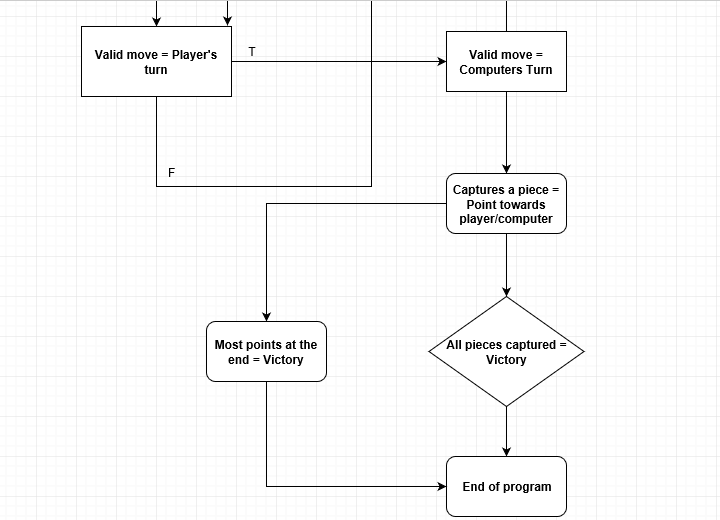
This project took me some time because it was a little more complicated than I initially thought it was going to be. There are a few problems with it (Like, after each move, when it’s the computers turn to move a piece, it constantly keeps saying “Invalid move” repeatedly until the computer makes a valid one.) I plan to fix and make this program more efficient during the second project and improve on the problems with it.

**Description:**

I wanted to recreate the board game of Checkers into a digital version people could play at home. It isn’t perfect, but it’s a work in progress none the less.

**Flow Chart:**





**Pseudo Code:**

***Initialize***

***Checkers board initialized Rules stated Computer moves***

***Invalid move Computer restarts***

***Valid move Piece Moves Turn Changes Players move***

***Invalid move Player restarts***

***Valid move Piece Moves Turn Changes Computers move***

***Piece Captured Point given to player/computer***

***All pieces taken/Most pieces taken Victory***

**Reference:**

1. Textbook
2. Slideshows

**Program:**

// Checkers game

#include <iostream>

#include <ctime>

#include <cmath>

#include <iomanip>

#include <cstdlib>

#include <fstream>

#include <string>

using namespace std;

class Board

{

public:

~Board(){};

Board();

void printb();

void human\_move();

void computer\_move();

void instructions();

void setpieces();

int checkwin();

int trymove (int r, int c);

void findp();

private:

char cboard[9][9];

};

Board::Board()

{

char w=177;

int row,column,side=0,top=0;

cout<<"Initializing the Checkers Board........"<<endl;

cout<<" 0 1 2 3 4 5 6 7 "<<endl;

cout<<"----------------------------"<<endl;

for(row=0;row<8;row++)

{

cout<<side<<" |";

++side;

for(column=0;column<8;column++)

if((row+column)%2==0)

cboard[row][column]=w;

else

cboard[row][column]=0;

cout<<"|"<<endl;

}

cout<<"----------------------------"<<endl;

}

void Board::setpieces()

{

int row=0, column=0;

for (row=1;row<=8;row++)

for (column=1;column<=8;column++)

{

if ((row+column)%2==0 && row<=3)

cboard[row][column]='h';

else if ((row+column)%2==0&&row>2&&row<=5)

cboard[row][column]=177;

else if ((row+column)%2==0&&row>5&&row<=8)

cboard[row][column]='q';

else

cboard[row][column]=0;

}

}

int Board::checkwin()

{

int i,j;

for (i=1;i<=8;++i)

for(j=1;j<=8;++j)

{

if (cboard[i][j]=='h'||cboard[i][j]=='q')

{

cout<<"Continue playing....."<<endl;

return (1);

}

else if (cboard[i][j]!='h')

{

cout<<"COMPUTER WINS!!!"<<endl;

return(0);

}

else if (cboard[i][j]!='q')

{

cout<<"HUMAN WINS!!!!"<<endl;

return(0);

}

}

}

void Board::findp()

{

int i,v=0,j;

do

{

for (i=0;i<=7;++i)

{

for (j=0;j<=7;++j)

{

if (cboard[i][j]=='q')

{

v=trymove (i,j);

if (v==1)

{

i=10;

j=10;

}

}

else

cout<<"INVALID MOVE"<<endl;

}

}

}

while (v!=1);

}

int Board::trymove(int r, int c) // R AND C REPRESENT ORIGINAL LOCATION OF THE PIECE

{

if (r>1&&c>1)

{

if (cboard[r][c]=='q')

{

cout<<"Hello1: "<<r<<" "<<c<<endl;

cboard[r][c]=177;

cboard[r-1][c-1]='q';

cout<<"The computer moves from row: "<<r<<endl;

cout<<"The computer moves from column: "<<c<<endl;

cout<<"To row: "<<r-1<<endl;

cout<<"To column: "<<c-1<<endl;

return(1);

}

else if (cboard[r][c]==177)

{

cout<<"Hello2"<<endl;

cboard[r][c]=177;

cboard[r-1][c+1]='q';

cout<<"The computer moves from row: "<<r<<endl;

cout<<"The computer moves from column: "<<c<<endl;

cout<<"To row: "<<r-1<<endl;

cout<<"To column: "<<c+1<<endl;

return(1);

}

else if (cboard [r-1][c-1]=='h'&&cboard[r-2][c-2]==177)

{

cboard[r][c]==177;

cboard[r-1][c-1]==177;

cboard[r-2][c-2]=='q';

return(1);

}

}

}

void Board::instructions()

{

int i=0;

char s=4;

for (i=0;i<=53;++i)

{

cout<<s;

}

cout<<endl<<s<<" THESE ARE THE INSTRUCTION FOR CHECKERS! "<<s<<endl;

cout<<s<<" 1) The computer is represented by a 'q' "<<s<<endl;

cout<<s<<" 2) The human is represented by a 'h' "<<s<<endl;

cout<<s<<" 3) When a piece is kinged it will be capitalized "<<s<<endl;

cout<<s<<" 4) When a piece is kinged it may move backwards "<<s<<endl;

cout<<s<<" 5) The computer will move first each time "<<s<<endl;

for (i=0;i<=53;++i)

{

cout<<s;

}

cout<<endl;

}

void Board::printb()

{

char w=177;

int row,column;

cout<<" 1 2 3 4 5 6 7 8"<<endl;

cout<<"----------------------------"<<endl;

for(row=1;row<=8;row++)

{

cout<<row<<" |";

for(column=1;column<=8;column++)

if((row+column)%2==0)

cout<<w<<cboard[row][column]<<w;

else

cout<<" "<<cboard[row][column]<<" ";

cout<<"|"<<endl;

}

cout<<"----------------------------"<<endl;

}

void Board::human\_move()

{

int oldrow=1, oldcolumn=1,row=1, column=1;

char w=177,h=104;

do

{

cout<<"Enter the row of the piece you would like to move from: "<<endl;

cin>>oldrow;

cout<<"Enter the column of the piece you would like to move from: "<<endl;

cin>>oldcolumn;

cout<<"Enter the new row you would like to move the piece to: "<<endl;

cin>>row;

cout<<"Enter the new column you would like to move the piece to: "<<endl;

cin>>column;

if (cboard[row][column]==0)

{

cout<<"INVALID MOVE"<<endl;

oldrow=0;

}

if (oldrow>=1&&oldcolumn>=1)

{

if (cboard[oldrow][oldcolumn]!='h')

{

cout<<"INVALID MOVE"<<endl;

oldrow=0;

}

else if (cboard[oldrow][oldcolumn]=='h')

{

cout<<"Hello1"<<endl;

cboard[oldrow][oldcolumn]=177;

cboard[row-1][column-1]=177;

cboard[row][column]='h';

}

else if (cboard[row-1][column-1]=='q')

{

cout<<"HELLO AGAIN"<<endl;

cboard[oldrow][oldcolumn]=177;

cboard[row-1][column-1]=177;

cboard[row][column]='h';

}

else

{

cout<<"INVALID MOVE 2"<<endl;

}

}

cout<<"The user moves from row: "<<oldrow<<endl;

cout<<"The user moves from column: "<<oldcolumn<<endl;

cout<<"To row: "<<row<<endl;

cout<<"To column: "<<column<<endl;

}

while (oldrow<1||oldcolumn<1||row<1||column<1||oldrow>8||oldcolumn>8||row>8||column>8);

}

int main()

{

srand(time(0));

int i=0,m=0;

Board play\_game;

play\_game.instructions();

play\_game.setpieces();

play\_game.printb();

cout<<"Computer Move...."<<endl;

do

{

play\_game.findp();

m=play\_game.checkwin();

if (m==0)

{

i=100;

}

play\_game.printb();

++i;

play\_game.human\_move();

m=play\_game.checkwin();

if (m==0)

{

i=100;

}

play\_game.printb();

++i;

}

while (i<100);

}