Project 1

“Trouble”

By: Anthony Morales

CIS-5

Introduction:

You start with 4 pieces at home, in order to move a piece out of home you have to roll a 6. Once you roll a 6 you move your pawn the amount the die rolled(6). Once your piece(s) are out of home you can move them with any die number, once ALL pawns reach finish (29 steps) you win the game.

Summary

Amount of lines around 300

Variables around 15

Using what I learned in class I used loops for user validation. By using my primitive data types and strings and functions, I made counters and set certain values to pawns as long as the moves were valid.

Description

The main point of this project is to show how implemented most of what I've learned in this class to create a program that will roll die and show a pawn location board, if the die is rolled and the move is valid you can move your pawn. I also made sure the pawn doesn't exceed spaces

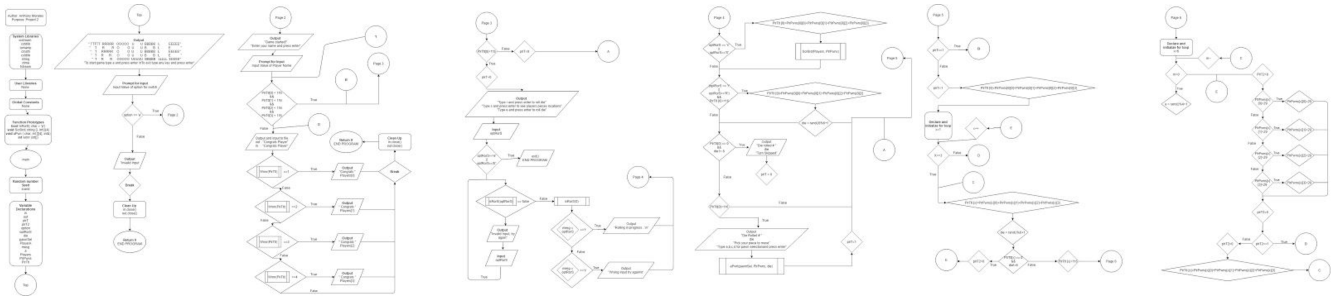
to make sure the user doesn't waste the die roll. Once all pawns are finished the program will output a message and end.

TO-DO	FINISHED	Lines
Libraries	✓	16-23
Integers	✓	46
Characters	✓	48
Strings	✓	51
Floats	✓	45
Bools	✓	200
Comments	✓	44-52
Type Casting	✓	42
Formatting Output	✓	74-78
Strings	✓	51
Math Library	✓	19
if	✓	164
if else	✓	90-128
Nesting	✓	133-134
if-else-if	✓	114-123
Logical Operators	✓	114
Validating User input	✓	107-110
Conditional operator	✓	103
Switch	✓	81-193
increment/decrement	✓	133 & 141
While	✓	98
Do-while	✓	134-174
For loop	✓	133
File input/output	✓	179 & 180

SAME

To-DO	FINISHED	Lines
Function Prototypes	✓	34-37
Pass by Value	✓	102
Return	✓	275
Returning Bool	✓	202
NO Global VARIABLES	✓	✓
Static Variables	✓	274
Defaulted Arguments	✓	34
Pass by Reference	✓	121
Overloading	✓	200 & 213
exit() function	✓	97
Single dimension	✓	66
Parallel arrays	✓	61 & 62
1-D Array as function Argument	✓	61
2-D Array	✓	62
Passing Arrays 2 & from functions	✓	273
Passing Vectors 2 & from functions	/	
Bubble Sort	/	
Selection Sort	/	
Linear or Binary Search	/	

Flowchart:



(Flowchart also in Final_V folder.)

Code:

```
/*

 * To change this license header, choose License Headers in Project Properties.

 * To change this template file, choose Tools | Templates

 * and open the template in the editor.

 */



/*

 * File:   main.cpp

 * Author: Anthony Morales

 *

 * Created on 6/6/2020 , 6:19 AM

 * Purpose: Project 2

 */
```

```
//System Level Libraries
#include <iostream>   //I/O Library
#include <cstdlib>
#include <iomanip>
#include <cmath>
#include <cstdlib>
#include <string>
#include <ctime>
#include <fstream>
using namespace std;  //Libraries compiled under std

//User Level Libraries



//Global Constants - Science/Math Related

//Conversions, Higher Dimensions

//Function Prototypes
bool isRorS (char = 'p');//input verification(r or s....defaulted to make false)
void ScrBrd(string [], int [][4]);//scoreboard function
void aPwn(char , int [][4] , int& );//Pawn selection
int wnnr( int []);//winner message
```

```
//Execution Begins Here!
int main(int argc, char** argv) {
//Random Number Seed Set Here
srand(static_cast<unsigned int>(time(0)));// random number seed
//Variable Declarations
fstream in , out; //for input and output to a file
float plrT; // counter for turn main player
int plrT2; //counter for turn bots
char option; // for option to roll, scoreboard, or end game
char optRorS; // input fo roption r or s
int die; //the die
char pawnSel;// select which pawn
string PlayerA; // Name of user
string mesg; //rolling in progress or invalid message
int a; // bot player selection
    //Variable Initialization
in.open("title.txt", ios::in);//inputs to a file
out.open("title.txt", ios:: out);//outputs to a file
die = 0;
pawnSel = 0;
plrT = 1;
plrT2 = 0;
string Players [4]= {" ", "Player 2","Player 3","Player 4"}; // player names
```

```
int PlrPwns[4][4] = {{0,0,0,0}, //pawn positions
            {0,0,0,0},
            {0,0,0,0},
            {0,0,0,0}};
int PlrTtl[4] = {0,0,0,0}; // players totals
a=0;
PlrTtl [0]=PlrPwns[0][0]+PlrPwns[0][1]+PlrPwns[0][2]+PlrPwns[0][3]; //setting player total equal to players pawns
PlrTtl [1] =PlrPwns[1][0]+PlrPwns[1][1]+PlrPwns[1][2]+PlrPwns[1][3]; //setting player total equal to players pawns
PlrTtl [2] =PlrPwns[2][0]+PlrPwns[2][1]+PlrPwns[2][2]+PlrPwns[2][3]; //setting player total equal to players pawns
PlrTtl [3] =PlrPwns[3][0]+PlrPwns[3][1]+PlrPwns[3][2]+PlrPwns[3][3]; //setting player total equal to players pawns
//Mapping Process Inputs to Outputs
//Display Outputs
cout<<"TTTTT  RRRRR   OOOOO  U   U  BBBBB   L      EEEEE"<<endl;   //Trouble Written with letters
cout<<"  T    R    R  O   O  U   U  B    B  L      E"<<endl;
cout<<"  T    RRRRR   O   O  U   U  BBBBB   L      EEEEE"<<endl;
cout<<"  T    R  R    O   O  U   U  B    B  L      E"<<endl;
cout<<"  T    R    R  OOOOO  UUUUU  BBBBB   LLLLL  EEEEE"<<endl<<endl;  // End Trouble with letters
cout<<"To start game type a and press enter \nTo exit type any key and press enter"<<endl;
cin>>option; // input to start game
switch(option){ //switch for start game or end program
```

```
    case 'a':    // If a is chosen game starts

        cout<< "Game Started!"<<endl;

        cout<<"Enter your name and press enter"<<endl;

        cin>>PlayerA; // users name

        Players[0] = PlayerA; //Players Name input

        while(PlrTtl [0]<116 || PlrTtl [1]<116 || PlrTtl [2]<116 || PlrTtl [3]<116){ // if any players total score equals 116 end loop

    do{

        PlrTtl [0]=PlrPwns[0][0]+PlrPwns[0][1]+PlrPwns[0][2]+PlrPwns[0][3];

    if(PlrTtl [0]<116){   // if player total doesnt equal 116

       plrT=0;

       cout<<"Type r and press enter to roll die"<<endl;

       cout<<"Type s and press enter to see players' pieces locations"<<endl;

       cout<<"Type e and press enter to end game"<<endl;

       cin>>optRorS;

    if (optRorS== 'e'||optRorS=='E') // if e is selected end game

       exit(0);

    while( isRorS(optRorS) == false){ //if r or s isnt chosen try again ( ask for user input)

        cout<<"Invalid Input, try again"<<endl;

        cin>>optRorS;

    }

        isRorS(0);

        mesg = optRorS=='r'?"Rolling in Progress...\n"://shows a message on your first move

        optRorS=='s'?"Pawn Board : \n":
```

```
        "Wrong input try again\n";

        cout<<mesg;

    if ((optRorS == 's')||(optRorS == 'S')) { //if r is chosen

        PlrTtl [0]=PlrPwns[0][0]+PlrPwns[0][1]+PlrPwns[0][2]+PlrPwns[0][3]; //sets player total to
pawns total

        ScrBrd(Players , PlrPwns ); //function to display score board

    }

    if ((optRorS == 'r')||(optRorS == 'R')&& PlrTtl [0]<116){  // if r is chosen roll die

        PlrTtl [0]=PlrPwns[0][0]+PlrPwns[0][1]+PlrPwns[0][2]+PlrPwns[0][3];

        die = rand()%6+1;// random number generator 1-6

    if (PlrTtl [0] == 0 && die != 6){//First move validation based on game rules

        cout<<"Die Rolled # "<<die<<endl<<"Turn Skipped"<<endl;

        plrT=0;

    }

    else if (PlrTtl [0]<116) { //if first roll isnt 6 do the following

        cout<<"Die Rolled # "<<die<<endl<<"Pick your piece to move"<<endl;

        cout<<"Type a,b,c,d for pawn selection and press enter"<<endl;

        aPwn(pawnSel, PlrPwns , die);    // Function for selecting pawn

        plrT=1;   // when equal to 1 turn ends due to loop ending

    }

    }

    }

    else{

        plrT=0; // 0 makes loop run again
```

```
}

}

while (plrT==1);

    plrT=1; // sets player turn back to one

    PlrTtl [0]=PlrPwns[0][0]+PlrPwns[0][1]+PlrPwns[0][2]+PlrPwns[0][3];

for(int x = 1; x<=3 ; x++){     // runs bot moves x=3, there are 3 bot turns

do{

    PlrTtl [x]=PlrPwns[x][0]+PlrPwns[x][1]+PlrPwns[x][2]+PlrPwns[x][3]; // set bots total score to total of pawns

    die = rand()%6+1;// random number generator 1-6

if (PlrTtl [x] == 0 && die != 6){//First move validation based on game rules

    plrT2=0;

}

else if (PlrTtl [x]<116) {  // if players total is not 116

for( int m=6;m>0;m--){ // this makes the bot randomly pick  pawn,bot may enter invalid move, this gives it 6 chances)

    a = rand()%4+1;

if(a==1 && PlrPwns[x][0]<29){ // if bot picks 1

    PlrPwns[x][0]+=die;

    m = 0;

}

else if(a==2 && PlrPwns[x][1]<29){ //if bot picks 2 and is less than 29

    PlrPwns[x][1]+=die;  // adds die to pawn

    m = 0;
}

else if(a==3 && PlrPwns[x][2]<29){ // if bot picks 3  and is less than 29

   PlrPwns[x][2]+=die; // adds die to pawn

   m =0;

}

else if(a== 4 && PlrPwns[x][3]<29){ //if bot picks 4  and is less than 29

   PlrPwns[x][3]+=die;     // adds die to pawn

   m =0 ;

}

else{

   plrT2=0; // ends turn

}

  }

   plrT2 = 0;

if(PlrPwns[x][0]>29) //sets to 29 if value is over

   PlrPwns[x][0]=29;//sets to 29 if value is over

if(PlrPwns[x][1]>29)//sets to 29 if value is over

   PlrPwns[x][1]=29;//sets to 29 if value is over

if(PlrPwns[x][2]>29)//sets to 29 if value is over

   PlrPwns[x][2]=29;//sets to 29 if value is over

if(PlrPwns[x][3]>29)//sets to 29 if value is over

   PlrPwns[x][3]=29;//sets to 29 if value is over

}

   plrT2 = 0;
```

```
    }while (plrT2==1); // if Bot player turns is one loop goes on
        plrT2=1; // sets bot turn to 1 again
        PlrTtl [x]=PlrPwns[x][0]+PlrPwns[x][1]+PlrPwns[x][2]+PlrPwns[x][3];
}
}
        out<<"Congrats Player "<<endl;
        in<<"Congrats Player "<<endl;
    if(wnnr(PlrTtl)==1) // if function returns 1
        cout<<"Congrats "<<Players[0]<<endl;
    if(wnnr(PlrTtl)==2)// if function returns 2
        cout<<"Congrats "<<Players[1]<<endl;
    if(wnnr(PlrTtl)==3)// if function returns 3
        cout<<"Congrats "<<Players[2]<<endl;
    if(wnnr(PlrTtl)==4)// if function returns 4
        cout<<"Congrats "<<Players[3]<<endl;
break;
default: //if a i'snt chosen at start program will print message and end
        cout<<"Invalid Input"<<endl;
break;
}
    //Clean Up
    in.close();//closes files open
    out.close();  //closes files open
    //Exit stage right!
    return 0;

}

bool isRorS (char optRorS){ //input validation. RorS is r or s (caps doesnt matter)

    if(optRorS == 'r') // if input is 'r' set function (isRorS) to true

        return true;

    if(optRorS == 'R')// if input is 'R' set function (isRorS) to true

        return true;

    if(optRorS == 's')// if input is 's' set function (isRorS) to true

        return true;

    if(optRorS == 'S')// if input is 'S' set function (isRorS) to true

        return true;

    else{

        return false;//If letter besides r or s is input,funtion returns false

    }

}

bool isRorS (int x){ // Input validation Overflowing if user inputs a number

        return false;

}

void ScrBrd(string x[], int a[4][4]){ //scoreboard function

   for(int b = 0; b<4;b++){ // loop to display players scores

        cout<<x[b]<<"'s Pawns"<<endl;

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

        cout<<"Pawn A:"<<a[b][0]<<endl;

        cout<<"Pawn B:"<<a[b][1]<<endl;
        cout<<"Pawn C:"<<a[b][2]<<endl;

        cout<<"Pawn D:"<<a[b][3]<<endl;

        cout<<endl;

    }

}

void aPwn(char a, int x[4][4], int& die){ // pawn selection function

    for(int cnt = 0;cnt<4;cnt++){ // gives user 4 tries to enter valid pawn before turn is skipped

        cin>>a;

    if (a=='a' && die!=6 && x[0][0]==0){ //validates selection

            cout<<"Invalid Input, try again"<<endl;

  }

    else if (a=='b' && die!=6 && x[0][1]==0){//validates selection

            cout<<"Invalid Input, try again"<<endl;

  }

    else if (a=='c' && die!=6 && x[0][2]==0){//validates selection

            cout<<"Invalid Input, try again"<<endl;

  }

    else if (a=='d' && die!=6 && x[0][3]==0){//validates selection

            cout<<"Invalid Input, try again"<<endl;

  }

    else{

        if(a=='a' && x[0][0]<29 ){//validates selection

            x[0][0]+=die;

            cnt=4;
        }
        else if(a=='b'&& x[0][1]<29){//validates selection
            x[0][1]+=die;
            cnt=4;
        }
        else if(a=='c'&& x[0][2]<29){//validates selection
            x[0][2]+=die;
            cnt=4;
        }
        else if(a=='d'&& x[0][3]<29){//validates selection
           x[0][3]+=die;
           cnt=4;
        }
        else{
        cout<<"invalid input try again"<<endl;
}
}
}
    if(x[0][0]>29) // if pawn is greater than 29 set to 29
        x[0][0]=29;
    if(x[0][1]>29)// if pawn is greater than 29 set to 29
        x[0][1]=29;
    if(x[0][2]>29)// if pawn is greater than 29 set to 29
        x[0][2]=29;
    if(x[0][3]>29)// if pawn is greater than 29 set to 29
        x[0][3]=29;
}
int wnnr( int PlrTotl[4]){ //winner function, returns number for winner
    static int dflt = 1;
    if(PlrTotl [0] == 116)
        return 1;
    else if(PlrTotl [1] == 116)
        return 2;
    else if(PlrTotl [2] == 116)
        return 3;
    else if(PlrTotl [3] == 116)
        return 4;
    else{
        return dflt;
    }
}
```