

Project 2

<Mastermind with AI>

CIS-7-43527

Name: Koksai, Attila

Date: 6/1/2022

Introduction

This project revolves around the game, Mastermind but with an AI included. I am doing this game since it's a game that is really enjoying and fun to play.

Summary

The program is 262 lines of code. This project meets the criteria of the first project because of how it implements a simple version of the game Mastermind in C++ code. This project utilizes pointers, arrays, and many more as well as 23 variables are used in this project. 45 constructs were utilized in this project. It was challenging for me, especially to be able to provide the computer or AI the necessary information to play the mastermind game by itself like a normal human would play it. It took approximately 20 hours, including research, review, debugging, and coding, in total to complete this project.

Description

With the goal of creating a playable mastermind with AI game, there are various sections and challenges to be completed. These steps include using a random number generator to generate a random correct answer for the user to try to guess at for a designated number of guesses. Then, enable the user to input possible combinations to try to get the correct answer through the usage of hints, which are derived from the user's previous inputs in order to assist the user in getting the correct answer and winning the game. In addition, having an inbuilt AI play the mastermind game with knowing prior background knowledge is a challenge in and of itself.

Pseudocode

Set a random number seed generator

Initialize code breaker to 0 and current guess holder

Initialize number of guesses to 0 and correct and wrong placeholders

Do

Add 1 to number of guesses

Take the current guess holder and run through compAI function

*While calculating the comparison between guess and solution and while number of
 guesses < 100*

Program

```
// Mastermind with AI
```

```
#include <iostream>
```

```
#include <cstdlib>
```

```
#include <ctime>
```

```

#include <string>
#include <iomanip>
using namespace std;

//Function Prototypes
string compAI(char,char);
bool calculate(string,string,char &,char &);
string solution();

int main(int argc, char** argv) {
    //Set random number seed generator
    srand(static_cast<unsigned int>(time(0)));

    //Declare variables
    string code,guess; //code breaker and holds the current guess
    char correct,wrong; //right digit in correct place and right digit in incorrect place
    int numGuess;      //number of guesses

    //Initialize Values
    numGuess=0;
    code=solution();
    correct=0;
    wrong=0;

    //Loop until code is solved and keep track of the number of guesses used to find the final
    solution
    do{
        numGuess++;
        guess=compAI(correct,wrong);
    }

```

```

    }while(calculate(code,guess,correct,wrong)&&numGuess<100);

    //Exit the program
    return 0;
}

string compAI(char correct,char wrong){
    static const int arrSize = 10;
    static const int gSize = 4;

    static bool firstTime = true;
    static bool allFour = false;
    static bool firstAI = true;
    static bool firstFind = false;
    static bool secondFind = false;
    static bool thirdFind = false;
    static bool allFind = false;

    static char rightArray[] = {0,0,0,0,0,0,0,0,0,0};
    static char rightGuess[] = {0,0,0,0}; //Holds the correct digit out of order

    static int index = 0;
    static char index1 = 0; //rightGuess Index
    static char index2 = 0; //solution Index
    static char index3 = 0; //userGuess Index

    int sum = 0;

```

```
string userGuess = "xxxx";  
static string solution = "xxxx";
```

```
// Determine which values are correct
```

```
if(index <= arrSize && !allFour){
```

```
    if(firstTime){
```

```
        userGuess = "0000";
```

```
        index++;
```

```
        firstTime = false;
```

```
    }
```

```
    else if(!firstTime){
```

```
        // Check previous value to see if its one of the numbers
```

```
        // if correct is not equal to 0, then the previous value is correct
```

```
        if(correct!=0){
```

```
            rightArray[index-1]=correct;
```

```
        }
```

```
        for(int i=0;i<arrSize;i++){
```

```
            sum += (int)rightArray[i];
```

```
        }
```

```
        if(sum!=4){
```

```
            for(int i=0;i<gSize;i++){
```

```
                userGuess[i] = index + '0';
```

```
            }
```

```

    }
    else if(sum==4){
        allFour = true;
    }
    index++;
}

if(allFour){
    for(int i=0;i<arrSize;i++){
        if(rightArray[i]!=0){
            for(int j=0;j<(int)rightArray[i];j++){
                rightGuess[index1++]=i;
            }
        }
    }
}

}

else{
    if(firstAI){
        index1=0;
        index2=0;
        index3=0;
        //Find the first number
        //Set the first index of rightGuess to the first index of userGuess
        userGuess[index3] = rightGuess[index1] + '0';
        firstAI=false;
    }
}

```

```

else if(firstAI==false){

    if(firstFind==false){
        if(correct==1){
            //Keep the first digit in the first position of solution
            solution[index2]=rightGuess[index1++] + '0';

            firstFind = true;
            if(solution[0]=='x'){
                index2=index3=0;
            }
            else if(solution[0]!='x'){
                index2=index3=1;
            }
            userGuess[index3] = rightGuess[index1] + '0';
        }
        else if(correct==0){
            index2++;
            userGuess[++index3]=rightGuess[index1] + '0';
        }
    }

    else if(firstFind==true && secondFind==false){
        //After finding the first digit, find the second digit
        if(correct==1){
            solution[index2++]=rightGuess[index1++] + '0';

            secondFind = true;

```

```

        if(solution[0]=='x'){
            index2=index3=0;
        }
        else if(solution[0]!='x' && solution[1]=='x'){
            index2=index3=1;
        }
        else if(solution[0]!='x' && solution[1]!='x' && solution[2]=='x'){
            index2=index3=2;
        }
        userGuess[index3] = rightGuess[index1] + '0';
    }

    else if (correct==0){
        if(solution[index2+1]=='x'){
            userGuess[++index3]= rightGuess[index1] + '0';

            index2++;
        }
        else if(solution[index2+1]!='x'){
            index3+=2;
            index2+=2;
            userGuess[index3]=rightGuess[index1] + '0';
        }
    }
}

else if(firstFind==true && secondFind==true && thirdFind==false){
    if(correct==1){

```



```

solution[index2++]=rightGuess[index1++] + '0';

thirdFind=true;
if(solution[0]=='x'){
    index2=index3=0;
}
else if(solution[0]!='x' && solution[1]=='x'){
    index2=index3=1;
}
else if(solution[0]!='x' && solution[1]!='x' && solution[2]=='x'){
    index2=index3=2;
}
else if(solution[0]!='x' && solution[1]!='x' && solution[2]!='x'){
    index2=index3=3;
}
userGuess[index3] = rightGuess[index1] + '0';
}

else if(correct==0){
    if(solution[index2+1]=='x'){
        index2++;
        index3++;
    }
    else if(solution[index2+1]!='x' && solution[index2+2]=='x'){
        index2+=2;
        index3+=2;
    }
    else if(solution[index2+1]!='x' && solution[index2+2]!='x' &&
solution[index2+3]=='x'){

```

```

        index2+=3;
        index3+=3;

    }

    userGuess[index3]=rightGuess[index1] + '0';
}

if(thirdFind==true){
    //Find the last digit by finding the position that's not found within solution
    for(int i=0;i<gSize;i++){
        if(solution[i]=='x'){
            solution[i] = rightGuess[index1] + '0';
            userGuess = solution;
            break;
        }
    }
}

return userGuess;
}

```

```

bool calculate(string code,string guess,char &correct,char &wrong){
    string check="  ";
    correct=0,wrong=0;
    //Check how many digits are in the correct place
    for(int i=0;i<code.length();i++){
        if(code[i]==guess[i]){

```

```

        correct++;

        check[i]='x';

        guess[i]='x';
    }
}

//Check how many digits are in the incorrect place
for(int j=0;j<code.length();j++){
    for(int i=0;i<code.length();i++){
        if((i!=j)&&(code[i]==guess[j])&&(check[i]=='')){
            wrong++;

            check[i]='x';

            break;
        }
    }
}

//Is the code broken or not
if(correct==4)return false;
return true;
}

string solution(){
    string code="0000";

    for(int i=0;i<code.length();i++){
        code[i]=rand()%10+'0';
    }

    return code;
}

```