Project 1

<MasterMind>

**CSC5-45276**

**Name: Javier Borja**

**Date: 7/18/2016**

**Introduction**

Title: MasterMind

For my first project in CSC5, I have decided to replicate one of my favorite games. MasterMind is a puzzle game, where a player has to “crack” a code. The first player is a codemaker and the second player is the codebreaker. The codemaker chooses a 4 color combination, and the codebreaker has to guess the code. The codemaker gives feedback to the codebreaker by setting white and colored pegs. Colored pegs correspond to correct color and position, and white pegs correspond to correct color but incorrect position. The game is over if the codebreaker guesses the correct code or the number of turns runs out.

My program has some slight changes to the classic game. Instead of colored codes, this game is based upon numbers. You may choose a 4 digit code consisting of the digits 1-8. No duplicate numbers are allowed for this version, and the max number of turns is limited to 10. This game will give you instant feedback each turn.

X’s = Correct number and position

O’s = Correct number but incorrect position

Guess the correct combination in 10 turns and win.

**Summary**

|  |  |
| --- | --- |
| Project size | 218 Lines |
| Lines of code | 179 Lines |
| Comment lines | 19 Lines |
| Blank lines | 20 Lines |

Version 1

Most of the program was written in version one. The main game and turn functions were developed in this stage. Reading the rules from another file and writing to another file was not working properly and many debugging output statements are included.

Version 2

Two new functions were added. A menu function which displays the menu and the results function that displays the results of the game. Formatting was improved in the game function and the rules function was working properly.

Version 3

Output formatting is finalized in this version. Win/loss counter is working correctly, writing results to file working properly, and all debugging statements are removed.

**Pseudocode**

//Function prototypes

Game

Turn

Results

Read directions

Menu

Initialize

Input name

Output menu

Input choice

Switch (choice)

Case 1: game function

Case 2: read rules

Default end program

Output wins/losses

Write results to file

Return 0

//Game function

Get random 4-digit combination

Turn function ten times or until player wins

Results function

//Turn function

Input player 4-digit combination

Output “X” for correct number and position

Output “O” for correct number and incorrect position

If combinations match

Win=true

Else

Next turn

//Results Function

If (Win=false)

Output “You Lose”

Else switch (turn)

Case 1-10: “You win”

//Different messages display depending on which turn you win

//Read function

Open rules file

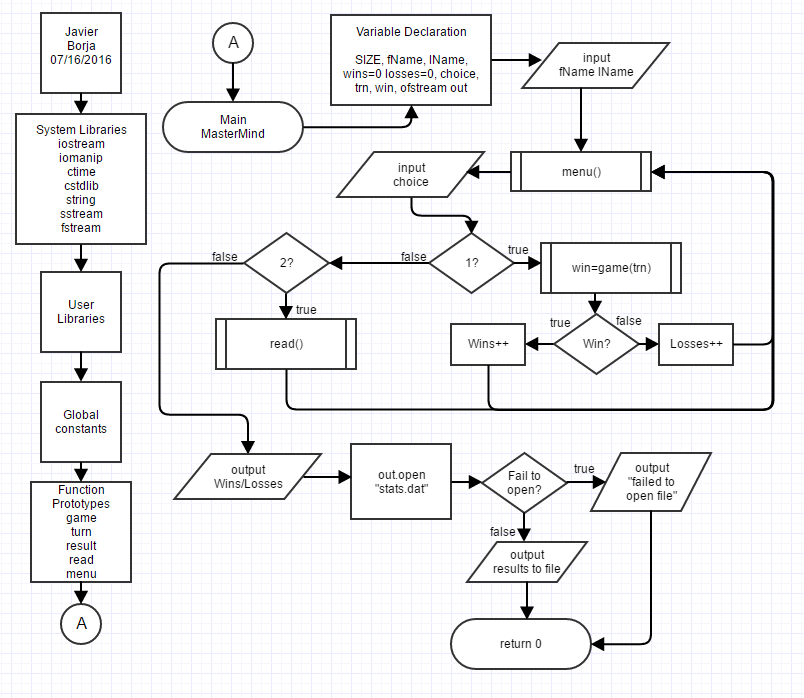
Output rules

Close rules file

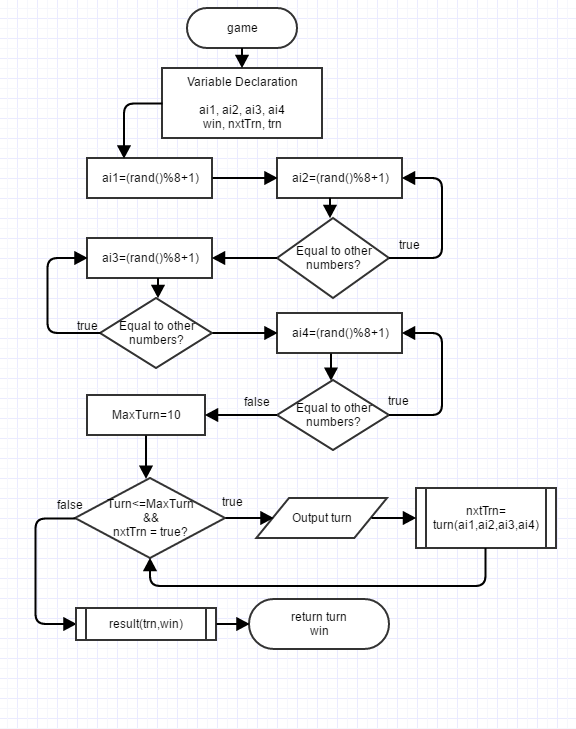
//Menu function

Output menu options

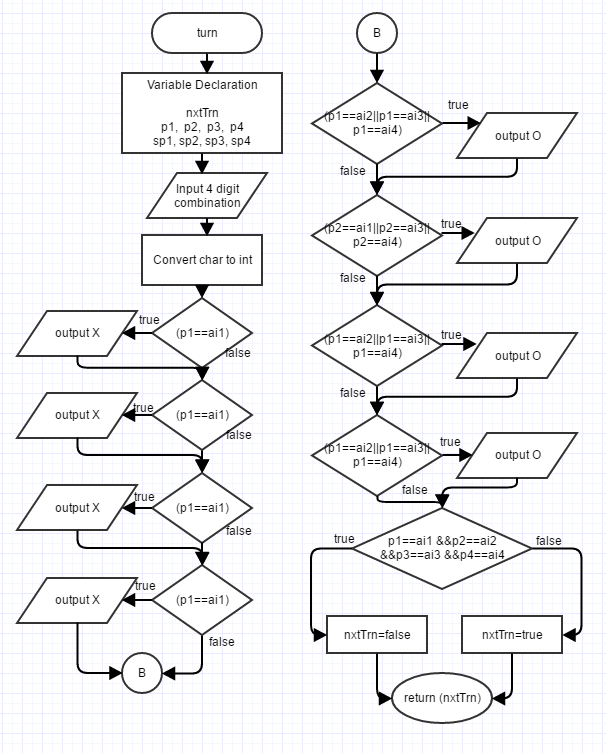
**Flowcharts:**

Main Function:

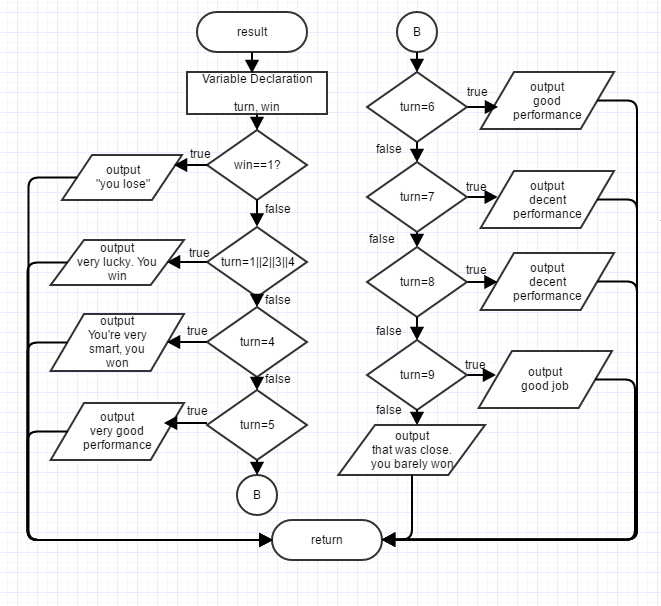
Game Function:



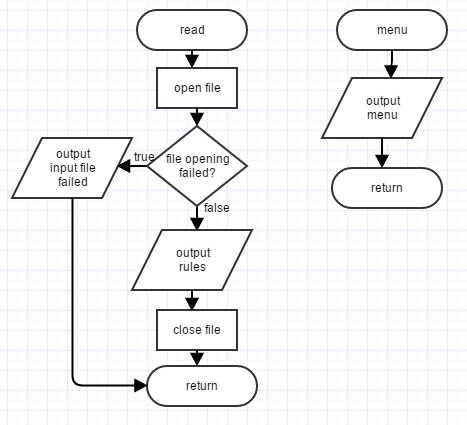
Turn Function:

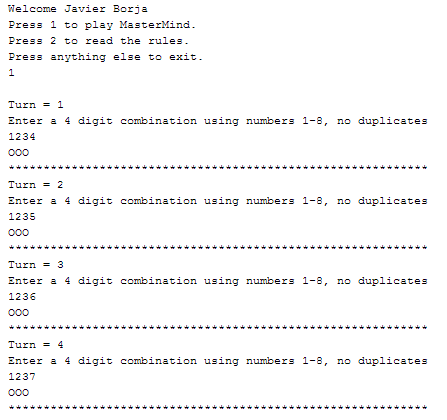
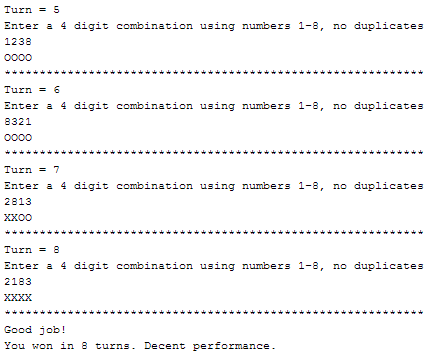


Result Function:



Read and Menu Functions:(next page)



Sample Gameplay:

**Major Variables:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Type** | **Variable Name** | **Description** | **Location** |
| bool | win | Game result. True results in win | Main, Result |
|  | nxtTrn | Determines whether to go to another turn | Main, Game, Result |
| int | SIZE | Size of char string | Main |
|  | trn | Which turn player is on | Game, Turn, Result |
|  | ai1 | Computer number choice | Game, Turn |
|  | ai2 | Computer number choice | Game, Turn |
|  | ai3 | Computer number choice | Game, Turn |
|  | ai4 | Computer number choice | Game, Turn |
|  | p1 | Player number choice | Turn |
|  | p2 | Player number choice | Turn |
|  | p3 | Player number choice | Turn |
|  | p4 | Player number choice | Turn |
|  | maxTrn | Max number of turns | Game |
| unsigned int | wins | Number of wins | Main |
|  | loss | Number of losses | Main |
| char | fName[SIZE] | Player first name | Main |
|  | sp1 | Player number choice | Turn |
|  | sp2 | Player number choice | Turn |
|  | sp3 | Player number choice | Turn |
|  | sp4 | Player number choice | Turn |
| string | lName | Player Last name | Main |
|  | string | String from rules.txt | Read |
| ofstream | out | Output to file | Main |
| ifstream | line | Input from rules.txt | Read |
| stringstream | ssp1 | Player number choice conversion | Turn |
|  | ssp2 | Player number choice conversion | Turn |
|  | ssp3 | Player number choice conversion | Turn |
|  | ssp4 | Player number choice conversion | Turn |

**Concepts utilized:**

From Savitch 9th Edition, concepts may be used in more than one location.

|  |  |  |
| --- | --- | --- |
| **Chapter** | **Construct** | **Location** |
| 2.1 | Variables and Identifiers | char choice; |
|  | Assigned statements | ai1=(rand()%8+1); |
| 2.2 | Input and Output | cin>>choice;  cout<<endl; |
|  | Escape sequences | cout<<"Press 1 to play MasterMind.\n" |
| 2.3 | Number types | int trn; |
|  | Type char | char choice; |
|  | Type bool | bool win; |
|  | Class string | string lName; |
| 2.4 | If-else statement | if(win==true) wins++;  else loss+=1; |
|  | Boolean expressions | if(p1==ai1) |
|  | Compound statements | if(win==0){  cout<<"You lose!... <<endl;  } |
|  | While loop | while (choice=='1'||choice=='2'); |
|  | Increment and decrement operator | wins++; |
| 2.5 | Naming constants | const int SIZE=20; |
| 3.2 | Nested statements | else  switch(turn){ |
|  | Switch statements and menus | switch(choice){ |
| 3.3 | For statement | for(int maxTrn=10;(trn<=maxTrn &&nxtTrn==true);trn++) |
| 3.4 | Sentinel values | "Press anything else to exit."<<endl;  cin>>choice; |
| 4.1 | Functions | bool game(int &turn); |
| 4.2 | Random Number Generation | ai1=(rand()%8+1); |
|  | Type Casting | srand(static\_cast<unsigned int>(time(0))); |
| 4.3 | Programmer defined functions | bool turn(int,int,int,int); |
|  | Return statements | return(nxtTrn); |
| 4.5 | Local variables | bool game(int &trn){  int ai1, ai2, ai3, ai4; |
| 5.1 | Void Function | void result(int turn,bool win); |
| 5.2 | Call-by-reference | bool game(int &turn); |
| 5.3 | Function calling functions | bool game(int &trn){  nxtTrn=turn(ai1,ai2,ai3,ai4);  } |
| 6.1 | Streams and File I/O | ofstream out;  ifstream line; |
|  | Writing to file | out<<"Last session stats for "<<fName<<" "<<lName; |
| 6.3 | Get from file | while(getline(line,string)){ |