Project 1

Mastermind

Csc-7 (42645)

Jarone Jabonillo

4/19/2017

Introduction:

Title: Mastermind

Mastermind is a logical guessing game where the player must guess a code generated by the Ai within a certain number of tries. The game will output a result based on what the player guesses, returning a ‘x’ for an incorrect number, a ‘o’ for a correct number but wrong spot, or a ‘\*’ for a correct number and correct spot. Afterwards, the player must use these clues in order to figure out the correct code. If the player can do this within the set amount of tries the player wins. If the player fails to guess the code correctly within the amount of tries, then the player loses.

Summary:

Code size: 540 Lines

Variables: 15

Methods: 15

When I started creating the project, I coded it all up in main to get a sense of how the logic worked when comparing the code with the guesses. After creating and fixing the functions for checking the code, I transferred it into two separate classes for the normal game and the Ai game.

For the normal game, I used character pointer arrays to store the code and the guesses and looped and compared them through the use of for loops.

Pseudo-code:

*Output main menu*

*Prompt user input*

*If they select 1*

*Start normal game*

*If they select 2*

*Start ai game*

*If they select 3*

*Quit game*

*Normal game*

*Generate ai code*

*Set tries = 10*

*While the tries is greater than 0 and the game is continuing*

*Player guesses code*

*Check player code*

*If the code and guess match along with the spot*

*increment correct spot*

*If the code and guess match but not the spot*

*increment correct but wrong*

*Else*

*increment incorrect*

*for each correct, correct but wrong, and incorrect*

*add each to the result*

*Output result of guess*

*If it doesn’t match*

*Decrement tries*

*Else*

*Flag the game end*

*Ai game*

*Receive player code*

*Ai guesses code*

*Check Ai code*

*If the code and guess match along with the spot*

*increment correct spot*

*If the code and guess match but not the spot*

*increment correct but wrong*

*Else*

*increment incorrect*

*for each correct, correct but wrong, and incorrect*

*add each to the result*

*Output result of guess*

*If it doesn’t match*

*Decrement tries*

*Else*

*Flag the game end*

Program:

//Main================================================

#include <cstdlib>

#include <iostream>

#include "Game.h"

#include "GameAi.h"

using namespace std;

/\*

\*

\*/

void cls(); //Clear screen

int main(int argc, char\*\* argv) {

bool quit = false; //Flag for game end

int size = 4; //Board size

Game gameplay(size); //Game object

GameAi gameplayAi(size); //Ai game object

do{

//User prompt

string ans;

cls();

cout << "Mastermind Menu\n"

<<"=====================\n"

<<"1] Start new game\n"

<<"2] Play vs AI\n"

<<"3] Quit\n"

<<"=====================\n"

<<"Enter Selection: ";

cin >> ans;

//Process user input

if(ans[0] == '1'){

//Start normal game

gameplay.start();

}

else if(ans[0] == '2'){

//Start ai game

gameplayAi.start();

}

else if(ans[0] == '3'){

//Quit game

quit = true;

}

cls();

}while(!quit);

cls();

cout <<"Thank you for playing!\n";

//Program end.

return 0;

}

void cls(){

for(int i = 0 ; i < 10; i++){

cout <<endl;

}

}

//Normal game============================================

class Game {

private:

int size; //Array size

int tries; //Tries

char\* ai; //Ai array

char\* player; //Player array

char\* result; //Array for results

bool compare(char[], char[], int); //Compares arrays

char\* check(char[], char[], int); //Checks and returns result

void output(char[], int); //Output array

void cls(); //Clear screen

public:

Game(int); //Constructor

~Game(); //Deconstructor

void start(); //Start game

};

#include "Game.h"

#include <cstdlib>

#include <iostream>

Game::Game(int size){

this->size = size; //Size for arrays

tries = 10; //Attempts to guess

ai = new char[this->size]; //Ai code

player = new char[this->size]; //Player array

result = new char[this->size]; //Result array

}

Game::~Game(){

//Remove and delete objects

ai = NULL;

delete ai;

player = NULL;

delete player;

result = NULL;

delete result;

}

void Game::start(){

std::string ans;

bool quit = false; //Flag for game end

bool game = false; //Flag for game success

srand(static\_cast<unsigned int>(time(0)));

do{

cls();

std::cout << "Game start\n";

game = true;

tries = 10;

//Generate random code

for(int i = 0; i < size; i++){

ai[i] = ((rand()%6)+1)+'0';

}

//Output Key table

std::cout <<"========================================"

<<std::endl <<"Key: "

<<std::endl <<"[x]Incorrect color"

<<std::endl <<"[o]Correct color "

<<std::endl <<"[\*]Correct color and spot"

<<std::endl <<"========================================\n";

do{

do{

//Prompt user guess

std::string num;

std::cout <<"[" <<tries <<" attempts] "

<<"Guess the 4 digit code(1-6):";

std::cin >> num;

//Assign input

for(int i = 0; i < size; i++){

player[i] = num[i];

}

std::cout <<"Confirm (y/n)?";

std::cin >> ans;

}while(ans[0] != 'y' && ans[0] != 'Y');

//If guess is wrong

if(!compare(ai, player, size)){

//Remove try & check result

tries--;

result = check(ai, player, size);

//Output result

std::cout <<"Result:";

output(result, size);

}

else{

game = false;

}

//When the game ends or player runs out of tries

}while(tries > 0 && game == true);

//Output answer

std::cout <<"Answer: ";

output(ai, size);

//Output result

if(game == true){

std::cout <<"You lost!\n";

}

else{

std::cout <<"You win!\n";

}

//Prompt restart

std::cout <<"Restart (y/n)? ";

std::cin >> ans;

if(ans[0] == 'n' || ans[0] == 'N'){

quit = true;

}

else{

quit = false;

}

}while(!quit);

}

bool Game::compare(char code[], char guess[], int size){

//Compares arrays and returns resulting bool

bool success = true;

for(int i = 0; i < size; i++){

if(guess[i] != code[i]){

success = false;

}

}

return success;

}

char\* Game::check(char code[], char guess[], int size){

//Compares and returns the resulting array

char\* ans = new char[size]; //Array for resulting answer

int index = 0; //Index for array

bool correct = false; //Flag for correct

bool almost = false; //Flag for correct but wrong spot

int star = 0; //Counter for correct

int o = 0; //Counter for correct but wrong spot

//Check through each line of the code

for(int i = 0; i < size; i++){

//Reset flags

correct = false;

almost = false;

//Check through guess

for(int j = 0; j < size; j++){

//If the guess matches the code and spot

if(j == i && code[i] == guess[j]){

//Raise correct flag

correct = true;

almost = false;

}

//If the guess matches the code but not the spot

else if(!correct && code[i] == guess[j]){

//Raise almost flag

almost = true;

}

}

//Increment counter based on result

if(correct){

star++;

}

else if(almost){

o++;

}

}

//Insert results into the answer array

for(int i = 0; i < size; i++){

if(star > 0){

ans[i] = '\*';

star--;

}

else if(star < 1 && o > 0){

ans[i] = 'o';

o--;

}

else{

ans[i] = 'x';

}

}

//Return answer

return ans;

}

void Game::output(char array[], int size){

//Output array

for(int i = 0; i < size; i++){

std::cout <<array[i];

}

std::cout <<std::endl;

}

void Game::cls(){

//Clear screen

for(int i = 0 ; i < 10; i++){

std::cout <<std::endl;

}

}