“Memory Game”

Name: Johandy Jonarto

ID: 2001586363

Description:

There are 2 types of memory games, the first is matching game where you match a pair of cards with the same number until you managed to finish all of them, and the second is connecting game where you must open the card with ascending order from 1 to 25. The game will keep repeating until you finished.

Design/Plan:

My idea for the project is to create a memory game. A game which use your brain to memorize the numbers/pictures within a limited amount of time, the game will be divided by levels and the games will become more harder with every increase in the lv. The program will count the number of moves you made during the games, so you can compete your score to others,

Function/Class:

1. Main Function

The entire program are used here, works as the core of the program

1. Class Access

Storing getAccess which will be put in the main function and the condition to start the game

1. getAccess

This function is used for the user to select the type of they want to play

1. void matchingGame

this will be the function that store the workings of the matching game, the game will not stop unless you managed to open all the random numbers folded

1. void shuffle

this function store the numbers we choose and randomizing a pair of number from 1 - 8

What I learned during the process:

* understanding bool much better than before
* how to randomize a custom number
* using do while to make things more efficient
* how to pair cards in the array while storing them if it’s a pair

Problem that I have overcome:

* randomize a custom number
* opening the \* which contains random number
* matching the random number with the number that I input
* how to keep repeating a certain pattern of move
* how to stop the game

Coding :

* “Main.cpp”

#include <iostream>

#include "Access.hpp"

#include "matchingGame.cpp"

using namespace std;

int main ()

{

int homeAccess;

Access access(homeAccess);

cout << "============================" << endl;

cout << "\t Memory Game" << endl;

cout << "============================" << endl << endl << endl;

cout << "Matching Game" << endl << endl << endl;

cout << "Press 1 to play the game. " << endl;

cin >> homeAccess;

if (homeAccess == 1)

{

cout << string( 150, '\n' );

matchingGame();

}

try

{

access.ifAccess(access.getAccess(homeAccess));

}

catch(string error)

{

cout << error;

cout << endl;

exit (0);

}

return 0;

}

* “Access.hpp”

#ifndef FinalProject\_hpp

#define FinalProject\_hpp

#include <iostream>

using namespace std;

class Access {

private:

int homeAccess;

public:

Access(int homeAccess)

{

this -> homeAccess = homeAccess;

}

int getAccess(int homeAccess)

{

return homeAccess;

}

void setAccess(int homeAccess)

{

this -> homeAccess = homeAccess;

}

int ifAccess(int)

{

if (homeAccess > 1)

{

string error = "error";

throw error;

cout << error;

}

return homeAccess;

}

};

#endif

* “shuffle.cpp”

#include <iostream>

#include <ctime>

#include <cstdlib>

#include <unistd.h>

using namespace std;

void shuffle(int random[][4])

{

int start = 0;

int number[16] = {1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8}; // the random cards

srand((unsigned)time(NULL));

for (int j = 0; j < 16; j++) // random 16 times

{

int start = rand()%(15-1+1)+1;

int point = number[j];

number[j] = number[start];

number[start] = point;

}

for (int i = 0; i < 4; i++)

{

for (int j = 0; j < 4; j++)

{

random[i][j] = number[start];

cout << random[i][j];

start = start + 1;

}

cout << endl;

}

}

* “matchingGame.cpp”

#include <iostream>

#include <string>

#include <ctime>

#include <cstdlib>

#include <unistd.h>

#include "shuffle.cpp"

using namespace std;

void shuffle(int [][4]);

void matchingGame()

{

char comma;

char ans = 'y';

int r1;

int c1;

int r2;

int c2;

int random[4][4];

bool randomStatus[4][4];

bool gameOver = false;

int moves;

int lv;

int time;

do

{

cout << "Choose the lv of difficulty you wanto to play(1-10): ";

cin >> lv;

while (lv > 10) {

cout << "Error !!!";

cout << string( 150, '\n' );

cout << "Choose the lv of difficulty you wanto to play(1-10): ";

cin >> lv;

}

cout << string(150, '\n');

time = 20 - lv;

moves = 0;

// call shufle

shuffle (random);

sleep(time);

cout << string(150, '\n');

// display \*

cout << endl;

for (int r = 0; r < 4; r++)

{

for (int c = 0; c < 4; c++)

{

cout << "\* ";

randomStatus[r][c] = false;

}

cout << endl;

}

do //game start

{

do

{

//selection

cout << "Please insert the first card row and column seperated by comma: ";

cin >> r1 >> comma >> c1;

if(randomStatus[r1-1][c1-1] == true) //-1 because array usually start from 0

{

cout << "This card is already flipped! Select another card." << endl;

}

}

while (randomStatus[r1-1][c1-1] != false);

do

{

cout <<"Please insert the second card row and column separated by comma: ";

cin >> r2 >> comma >> c2;

if(randomStatus[r2-1][c2-1] == true)

{

cout << "This card is already flipped! Select another card." << endl;

}

}

while(randomStatus[r2-1][c2-1] != false);

// fix

r1--;

c1--;

r2--;

c2--;

// to show \* cards

for (int r = 0; r < 4; r++)

{

for (int c = 0; c < 4; c++)

{

if ((r == r1) && (c == c1))

{

cout << random[r][c] << " ";

}

else if((r == r2) && (c == c2))

{

cout << random[r][c] << " ";

}

else

{

cout << "\* ";

}

}

cout << endl;

}

// matching the number

if (random[r1][c1] == random[r2][c2]) //if cards match, keep them flipped

{

randomStatus[r1][c1] = true;

randomStatus[r2][c2] = true;

}

cin.get();

cin.get();

cout << string(150, '\n');

for (int r = 0; r < 4; r++) //to show new board

{

for (int c = 0; c < 4; c++)

{

if (randomStatus[r][c] == true)

{

cout << random[r][c] << " ";

}

else

{

cout << "\* ";

}

}

cout << endl;

}

gameOver = true;

for (int r = 0; r < 4; r++) // if all card is flipped the game is over, else continue

{

for (int c = 0; c < 4; c++)

{

if(randomStatus[r][c]==false)

{

gameOver = false;

break;

}

}

if(gameOver == false)

{

break;

}

}

moves++;

// repeat the game

}

while(gameOver != true); // repeat until the game is finished

cout << "Total: " << moves << " moves"<<endl<<endl;

cout << "Game Over" << endl << endl << endl;

cout << "Would you like to play again? (y=Yes/n=No) : ";

ans = cin.get();

}

while(ans == 'y' || ans == 'Y'); // to play again

cin.get();

}