**FINAL PROJECT**

First Semester

**QUICK COUNT GAME**

Kevin Kurnia Santosa

2001585253

**Quick Count Game**

**Section 1**

Description

Quick count program is a game that test your ability in counting simple arithmethic questions like addition, substraction, and multiplication. Players can choose either to play or practice, in this both sections players will be given a random numbers to count as possible as they can and at the last, the program will display the score with the standard valuation.

Play section

In the play section, there are three levels with simple arithmethic questions using random numbers and random operations. In the first and second level, there are only two operations which are addition and subtraction but each time player pass it, it will be harder and will be more complicated with the presence of multiplication in the third level. The most important thing in this section is that the player should count quickly because there is a time limit given to the player. If the player can not done within the time limit, then game over and display the score.

Practice section

Unlike the play section, in this section player can train and practice their arithmethic skills. The random number will be bigger than in the play section but there is no time limit and there are no random operations. Player will be given three choices, addition, subtraction, and multiplication. In each operation there are ten numbers and at the last the score will be displayed.

**Section 2A**

Design / Plan **Hierarchy Chart**

Main Menu

int main()

Play

play()

Practice

int practice()

Practice Options

endOptionsPractice()

Display score

displayPoints()

Practice Question

questionPractice()

Multiplication

mulPractice()

Subtraction

subPractice()

Addition

addPractice()

timer()

Play Options

endOptionsPlay()

Play level 3

level3()

Play level 2

level2()

Play level 1

level1()

Countdown

count()

Play section will test and challenge the player’s to count as quickly as possible with the countdown time. This section will make the player to become more quickly in critical thinking and solving a problem.

Practice section is for train player to be smarter in count a simple arithmetic. Besides train, it also entertain the player to get the highest score in the game.

Class diagrams

Play.cpp

|  |
| --- |
| Play |
| * formula : int * signs : string * sign[2] : string * answer : int |
| + level1(num1 : int, num2 : int, operations : int, scores : int) : void  + level2(num1 : int, num2 : int, num3 : int, operations1: int, operations2 : int, scores : int) void  + level3(num1 : int, num2 : int, num3 : int, operations1: int, operations2 : int, scores : int) void  + timer(scores : int) : void  + endOptionsPlay()  + count() |

Practice.cpp

|  |
| --- |
| Practice |
| * formula : int * symbol : string * point : int = 0 |
| + addPractice() : void  + subPractice() : void  + mulPractice() : void  + questionPractice(number : int, answer : int, n1 : int, n2 : int, symbol : string, points : int) : int  + displayPoints(practiceScore : int) : void  + endOptionsPractice() : int |

**Section 2B**

**Classes**

* Class Play This class is used for the play section. In the int play() function will call the function in the class Play. This class is declared in the play.cpp.
* Class Practice This This class is used for the practice section. In the int practice() function will call the function in the class Practice. This class is declared in the practice.cpp.

**Functions**

* int play ()

This function is used to declared the score, random numbers, and the random operations and call the function level1(), level2(), and level3().

* void level1(int num1, int num2, int operations, int scores)

There are 1 random operations and 2 random numbers. Give the question to the player and if the answer not equal to the formula or the time runs out then game over

* void level2(int num1, int num2, int num3, int operations1, int operations2, int scores) There are 2 random operations and 2 random numbers. Give the question to the player and if the answer not equal to the formula or the time runs out then game over
* void level3(int num1, int num2, int num3, int operations1, int operations2, int scores) There are 2 random operations and 2 random numbers. Give the question to the player and if the answer not equal to the formula or the time runs out then game over
* void timer(int scores)

To start the timer and if the timer greater than 3 then display game over to the player

* void endOptionsPlay()

After the game over display, the player can choose either to back to the menu or play the game again.

* int practice()

This function is use to

* void addPractice ()

Declare two random numbers and the symbol (addition) , after that call the questionPractice function, display the score, and display the options.

* void subPractice ()

Declare two random numbers and the symbol (subtraction), after that call the questionPractice function, display the score, and display the options.

* void mulPractice ()

Declare two random numbers and the symbol (multiplication) , after that call the questionPractice function, display the score, and display the options.

* int questionPractice (int number, int answer, int n1, int n2, string symbol, int points)

Display the questions that being called by addPractice(), subPractice(), and mulPractice() functions.

* void displayPoints (int practiceScore)

Display the score and called by the addPractice(), subPractice(), and mulPractice() functions.

* int endOptionsPractice ()

Display the options either the player want to play the practice section again or back to the main menu.

**Section 3**

1. Things that I have learnt during the process of making this program have made me more diligent and better in solving a problems.
2. Making multiple files and declaring class of each file needed.
3. Make a random operations using seed random.
4. Learn how to start / make a timer.
5. Learn to make multiple functions (call functions and be called)
6. Problems that I have overcome with.
7. At first there are so many lines in the practice and play section, because there are so many operations and different formula. To make the play section I need to declare all the possibilities of the operations which is make the line so long. For hours at last I managed to solved it by making the sign as a variable, so the I need to declare the questions only once.
8. Declare score in the function level1(), level2(), and level3() is not working because that three functions is inside the loop from the play() function and it will restart the score as 0 if I declared it in the level1(), level2(), and level3() functions, so I decided to declare the score variable in the play() function and cout / print it after the gameover displayed.

**Section 4**

**Code**

main.cpp

#include <iostream>

#include <ctime> // Given time for user to input

#include <cstdlib> // For seed rand (Generate random number)

#include <iomanip> // Setw

#include <string> // String

#include <conio.h> // kbhit

#include <windows.h> // Sleep function

// Include the separated file ( Play and Practice )

#include "Practice.cpp"

#include "Play.cpp"

using namespace std;

int main () // Start main function

{

int inputOptions;

string userName;

system("COLOR 0F");

cout << "\n\t\tQUICK COUNT GAME"; // Title of the Program

cout << "\n\n\n\tWelcome !\n\tEnter your name : ";

getline (cin, userName); // Store username

system ("CLS");

cout << "\n\t\tQUICK COUNT GAME" << "\n\n\t";

for (int line = 1; line <= 33; line++) {

cout << "=";

}

cout << endl << endl << " Hello " << userName << "!!";

cout << endl << endl << "\t1. Play" ;

cout << endl << "\t2. Practice";

cout << endl << "\t3. Exit";

cout << endl << endl << "\tPlease enter your choices (1, 2, 3) : ";

cin >> inputOptions;

switch (inputOptions)

{

case 1 :

play ();

break;

case 2 :

practice ();

break;

case 3 :

exit(0);

default :

cout << endl << "\tYou enter the wrong option.\n\n" << "\t";

system("PAUSE");

main();

break;

}

getchar ();

}

Play.cpp

#include <iostream>

#include <ctime> // Given time for user to input

#include <cstdlib> // For seed rand (Generate random number)

#include <string> // String

#include <conio.h> // kbhit

#include <windows.h> // Sleep

using namespace std;

// Declare function prototypes

int main();

int play();

// Class for play section

class Play {

private :

int formula;

string signs;

string sign[2];

int answer;

public :

void level1(int num1, int num2, int operations, int scores)

{

if (operations == 1) {

formula = num1 + num2;

signs = " + ";

}

else if (operations == 2) {

formula = num1 - num2;

signs = " - ";

}

cout << endl << "\tTimer : 3 sec" << endl;

cout << "\t" << num1 << signs << num2 << " = ";

timer(scores);

cin >> answer;

system ("CLS");

if (answer != formula) {

cout << endl << "\tTimer : 3 sec" << endl;

cout << "\tTry Again!" << endl;

cout << "\t" << num1 << signs << num2 << " = ";

timer(scores);

cin >> answer;

system ("CLS");

if (answer != formula) {

cout << endl << "\tTimer : 3 sec" << endl;

cout << "\tTry Again!" << endl;

cout << "\t" << num1 << signs << num2 << " = ";

timer(scores);

cin >> answer;

system("CLS");

if (answer != formula) {

cout << endl << "\tGAME OVER!!\n You have wrong 3 times!\n\n\t";

cout << "Score : " << scores << endl << endl;

Sleep(1500);

system("PAUSE");

endOptionsPlay();

exit(0);

}}}

}

void level2(int num1, int num2, int num3, int operations1, int operations2, int scores)

{

if (operations1 == 1 && operations2 == 1) {

formula = num1 + num2 + num3;

sign[0] = " + ";

sign[1] = " + ";

}

else if (operations1 == 1 && operations2 == 2) {

formula = num1 + num2 - num3;

sign[0] = " + ";

sign[1] = " - ";

}

else if (operations1 == 2 && operations2 == 1) {

formula = num1 - num2 + num3;

sign[0] = " - ";

sign[1] = " + ";

}

else if (operations1 == 2 && operations2 == 2) {

formula = num1 - num2 - num3;

sign[0] = " - ";

sign[1] = " - ";

}

cout << endl << "\tTimer : 3 sec" << endl;

cout << "\t" << num1 << sign[0] << num2 << sign[1] << num3 << " = ";

timer(scores);

cin >> answer;

system ("CLS");

if (answer != formula) {

cout << endl << "\tTimer : 3 sec" << endl;

cout << "\tTry Again!" << endl;

cout << "\t" << num1 << sign[0] << num2 << sign[1] << num3 << " = ";

timer(scores);

cin >> answer;

system ("CLS");

if (answer != formula) {

cout << endl << "\tTimer : 3 sec" << endl;

cout << "\tTry Again!" << endl;

cout << "\t" << num1 << sign[0] << num2 << sign[1] << num3 << " = ";

timer(scores);

cin >> answer;

system("CLS");

if (answer != formula) {

cout << endl << "\tGAME OVER!!\n You have wrong 3 times!\n\n\t";

cout << "Score : " << scores << endl << endl;

Sleep(1500);

system("PAUSE");

endOptionsPlay();

exit(0);

}}}

}

void level3(int num1, int num2, int num3, int operations1, int operations2, int scores)

{

if (operations1 == 1 && operations2 == 1) {

formula = num1 + num2 \* num3;

sign[0] = " + ";

sign[1] = " x ";

}

else if (operations1 == 1 && operations2 == 2) {

formula = num1 - num2 \* num3;

sign[0] = " - ";

sign[1] = " x ";

}

else if (operations1 == 2 && operations2 == 1) {

formula = num1 \* num2 + num3;

sign[0] = " x ";

sign[1] = " + ";

}

else if (operations1 == 2 && operations2 == 2) {

formula = num1 \* num2 - num3;

sign[0] = " x ";

sign[1] = " - ";

}

cout << endl << "\tTimer : 3 sec" << endl;

cout << "\t" << num1 << sign[0] << num2 << sign[1] << num3 << " = ";

timer(scores);

cin >> answer;

system ("CLS");

if (answer != formula) {

cout << endl << "\tTimer : 3 sec" << endl;

cout << "\tTry Again!" << endl;

cout << "\t" << num1 << sign[0] << num2 << sign[1] << num3 << " = ";

timer(scores);

cin >> answer;

system ("CLS");

if (answer != formula) {

cout << endl << "\tTimer : 3 sec" << endl;

cout << "\tTry Again!" << endl;

cout << "\t" << num1 << sign[0] << num2 << sign[1] << num3 << " = ";

timer(scores);

cin >> answer;

system("CLS");

if (answer != formula) {

cout << endl << "\tGAME OVER!!\n You have wrong 3 times!\n\n\t";

cout << "Score : " << scores << endl << endl;

Sleep(1500);

system("PAUSE");

endOptionsPlay();

exit(0);

}}}

}

void timer(int scores)

{

clock\_t start = clock();

while (!kbhit()) { // Check if user hit the keyboard

if (((clock () - start) / CLOCKS\_PER\_SEC ) >= 3) { // Check if the timer runs out or not

cout << "\n\n\tGAME OVER!!\n THE TIME HAS RUN OUT!" << endl << endl << "\t"; // If the time runs out, then game over

cout << "Score : " << scores << endl << endl;

system("PAUSE");

Sleep(1500);

system("CLS");

endOptionsPlay(); // Function

exit(0);

}

}

}

void endOptionsPlay()

{

int menu;

system("CLS");

cout << endl << " 1. Play again.";

cout << endl << " 2. Back to main menu.";

cout << endl << " Choose 1 or 2 : ";

cin >> menu;

if (menu == 1)

play();

else if (menu == 2)

main();

else {

cout << "\n Wrong option." << endl << endl << endl;

system("PAUSE");

endOptionsPlay();

}

}

void count()

{

for (int countdown = 3; countdown >= 0; countdown--) {

cout << endl << "\t " << countdown;

Sleep(1000);

system("CLS");

}

}

};

int play(){

unsigned s = time(0);

srand(s);

int score = 0;

Play play;

system ("CLS");

play.count();

for (int no = 1; no <= 10; no++) {

// First level : there are 1 operations with 2 possible signs (addition and subtraction) and 2 random numbers

int operation = (rand()%(2-1+1))+1; // Generate random number between 1 and 2

int n1 = (rand()%(10-1+1))+1; // Generate random number between 1 and 10

int n2 = (rand()%(10-1+1))+1; // Generate random number between 1 and 10

play.level1(n1, n2, operation, score); // Go to function questionPlay1 (First level)

score = score + 10;

}

cout << endl << "\tCongratulations reaching the first level!" << endl << endl << "\t";

system ("PAUSE");

system ("CLS");

for (int no = 1; no <= 10; no++) {

// Second level : there are 2 operations with 2 possible signs (addition and subtraction) and 3 random numbers

int operation1 = (rand()%(2-1+1))+1; // Generate random number between 1 and 2

int operation2 = (rand()%(2-1+1))+1; // Generate random number between 1 and 2

int n1 = (rand()%(10-1+1))+1; // Generate random number between 1 and 10

int n2 = (rand()%(10-1+1))+1; // Generate random number between 1 and 10

int n3 = (rand()%(10-1+1))+1; // Generate random number between 1 and 10

play.level2(n1, n2, n3, operation1, operation2, score); // Go to function questionPlay2 (Second level)

score = score + 10;

}

cout << endl << "\tCongratulations reaching the second level!" << endl << endl << "\t";

system ("PAUSE");

system ("CLS");

for (int no = 1; no <= 500; no++) {

// Third Level : there are 2 operations with 3 possible signs(addition, subtraction, and multiplication) and 3 random numbers

int operation1 = (rand()%(2-1+1))+1; // Generate random number between 1 and 2

int operation2 = (rand()%(2-1+1))+1; // Generate random number between 1 and 2

int n1 = (rand()%(10-1+1))+1; // Generate random number between 1 and 10

int n2 = (rand()%(10-1+1))+1; // Generate random number between 1 and 10

int n3 = (rand()%(10-1+1))+1; // Generate random number between 1 and 10

play.level3(n1, n2, n3, operation1, operation2, score); // Go to function questionPlay3 (Third level)

score = score + 10;

}

}

Practice.cpp

#include <iostream>

#include <ctime> // Given time for user to input

#include <cstdlib> // For seed rand (Generate random number)

#include <iomanip> // Setw

#include <string> // String

using namespace std;

// Declare function prototypes

int main();

int practice();

class Practice {

private :

int formula;

string symbol;

int point = 0;

public :

void addPractice ()

{

for (int number = 1; number <= 10; number++) {

int n1 = (rand()%(50-10+1))+10;

int n2 = (rand()%(50-10+1))+10;

formula = n1 + n2;

symbol = " \_\_\_\_+";

point = questionPractice (number, formula, n1, n2, symbol, point);

}

displayPoints(point);

endOptionsPractice ();

}

void subPractice ()

{

for (int number = 1; number <= 10; number++) {

int n1 = (rand()%(100-60+1))+60;

int n2 = (rand()%(60-10+1))+10;

formula = n1 - n2;

symbol = " \_\_\_\_-";

point = questionPractice (number, formula, n1, n2, symbol, point);

}

displayPoints(point);

endOptionsPractice ();

}

void mulPractice ()

{

for (int number = 1; number <= 10; number++) {

int n1 = (rand()%(12-1+1))+1;

int n2 = (rand()%(10-1+1))+1;

formula = n1 \* n2;

symbol = " \_\_\_\_x";

point = questionPractice (number, formula, n1, n2, symbol, point);

}

displayPoints(point);

endOptionsPractice ();

}

int questionPractice (int number, int answer, int n1, int n2, string symbol, int points)

{

int input;

cout << endl << setw(2) << number << ".";

cout << " " << n1 << endl;

cout << " " << n2 << endl;

cout << symbol << endl << " ";

cin >> input;

if (input != answer) {

points = points + 0;

cout << endl << " WRONG!! Right answer : " << answer << endl;

}

else if (input = answer) {

points = points + 10;

cout << endl << " RIGHT!!" << endl;

}

return points;

}

void displayPoints (int practiceScore)

{

cout << endl << " Score : " << practiceScore;

if (practiceScore == 100)

cout << endl << " Excellent!! You got perfect score!!";

else if (practiceScore >= 60 && practiceScore <= 100)

cout << endl << " Good work!!";

else

cout << endl << " You need more practice.";

}

int endOptionsPractice ()

{

string yesno;

int backMenu;

cout << endl;

cout << endl << " Try again ? \n Yes/No ? ";

cin >> yesno;

if (yesno == "Yes" || yesno == "yes" || yesno == "y") {

system ("CLS");

practice ();

}

else if (yesno == "No" || yesno == "no" || yesno == "n"){

system("CLS");

cout << endl << " 1. Back to menu.";

cout << endl << " 2. Exit the program.";

cout << endl << " Choose 1 or 2 : ";

cin >> backMenu;

if (backMenu == 1){

main();

}

else if (backMenu == 2){

exit(0);

}

else {

cout << "\n Wrong option.";

exit(0);

}

}

else {

cout << endl << " You choose the wrong option." << endl << endl;

system("PAUSE");

system("CLS");

endOptionsPractice();

}

}

};

int practice ()

{

char optionPractice;

unsigned s = time(0);

srand(s);

Practice exercise;

system ("CLS");

cout << endl << endl << " a. Addition";

cout << endl << " b. Subtraction";

cout << endl << " c. Multiplication";

cout << endl << " d. Back to main menu";

cout << endl << endl << " Which operations do you want to choose (a, b, c, d) ? ";

cin >> optionPractice;

switch (optionPractice)

{

case 'a' :

system("CLS");

exercise.addPractice();

break;

case 'b' :

system("CLS");

exercise.subPractice();

break;

case 'c' :

system("CLS");

exercise.mulPractice();

break;

case 'd' :

main();

default :

cout << endl << " You choose the wrong option.\n\n" << " ";

system("PAUSE");

system("CLS");

practice();

break;

}

}