/\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*|

Program: TT06\_A1\_MOHAMED\_HAMIDI\_BIN\_MOHAMED\_SAUFI.cpp

Course: TCP1101 Programming Fundamentals

Year: 2017/18 Trimester 1

Name: Mohamed Hamidi bin Mohamed Saufi

ID: 1161303937

Email: hamidisaufi@gmail.com

Phone: 011-11392329

\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*|\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <string>

#include <cstdlib>

using namespace std;

string newFileName(string textEnter); //For creating a new file

string emptySpace(string textEnter); //For adding empty spaces to tiles with respect to the length of string

string tableLine(string textEnter); //For adding the table lines with respect to the length of string

string textInsert(string textEnter); //Function for updating text in a tile

void insertWrongEmptyTiles(int tileNumber); //Function to be called when the tiles selected has empty previous tile

void emptyTextInTiles(); //Function to be called in the attempt to overwrite text in an empty tile

int main()

{

char menu;

string filename;

int tile;

string text[8] = { " "," "," "," "," "," "," "," " };

string emptySpaces[8] = { " "," "," "," "," "," "," "," " };

string tableLines[8] = { "-","-","-","-","-","-","-","-" }; //Default table arrays

int newFile = 0;

int insertText = 0;

int deleteText = 0;

int overwriteText = 0;

cout << endl;

cout << "Welcome to the One Line Stone-Age Tiled Editor" << endl;

cout << "What do you want to do today?" << endl;

cout << endl;

cout << "File name: NULL" << endl;

cout << endl;

do

{

cout << "\*============================================================" << endl;

cout << endl;

cout << "[N]ew, " << "[I]nsert, " << "[D]elete, " << "[O]verwrite, " << "[Q]uit" << endl;

cout << "==> ";

cin >> menu;

cout << endl;

if (menu == 'q' || menu == 'Q') {

cout << "Thank you for using One Line Stone-Age Tiled Editor" << endl;

cout << endl;

cout << "Here are your final stats:-" << endl;

cout << "Number of new file attempt: " << newFile << endl;

cout << "Number of insert attempt: " << insertText << endl;

cout << "Number of delete attempt: " << deleteText << endl;

cout << "Number of overwrite attempt: " << overwriteText << endl;

cout << endl;

}

else {

if (filename == "") { //When there is no new file created

if (menu == 'n' || menu == 'N') {

newFile += 1;

filename = newFileName(filename);

system("cls");

cout << endl;

cout << "|----+" << tableLines[0] << "--+" << tableLines[1] << "--+" << tableLines[2] << "--+"

<< tableLines[3] << "--+" << tableLines[4] << "--+" << tableLines[5] << "--+" << tableLines[6]

<< "--+" << tableLines[7] << "--|" << endl;

cout << "| |" << emptySpaces[0] << "1 |" << emptySpaces[1] << "2 |" << emptySpaces[2]

<< "3 |" << emptySpaces[3] << "4 |" << emptySpaces[4] << "5 |" << emptySpaces[5]

<< "6 |" << emptySpaces[6] << "7 |" << emptySpaces[7] << "8 |" << endl;

cout << "|----+" << tableLines[0] << "--+" << tableLines[1] << "--+" << tableLines[2] << "--+"

<< tableLines[3] << "--+" << tableLines[4] << "--+" << tableLines[5] << "--+" << tableLines[6]

<< "--+" << tableLines[7] << "--|" << endl;

cout << "| 00 | " << text[0] << " | " << text[1] << " | " << text[2] << " | " << text[3] << " | "

<< text[4] << " | " << text[5] << " | " << text[6] << " | " << text[7] << " |" << endl;

cout << "|----+" << tableLines[0] << "--+" << tableLines[1] << "--+" << tableLines[2] << "--+"

<< tableLines[3] << "--+" << tableLines[4] << "--+" << tableLines[5] << "--+" << tableLines[6]

<< "--+" << tableLines[7] << "--|" << endl;

cout << endl;

cout << "File name: " << filename << ".txt" << endl;

cout << endl;

}

else if (menu == 'I' || menu == 'i' || menu == 'D' || menu == 'd' || menu == 'O' || menu == 'o') {

system("cls");

cout << endl;

cout << "No new file detected. Please create a new one" << endl;

}

else {

system("cls");

cout << endl;

cout << "Invalid command. Please try again" << endl;

}

}

else { //When a new file is created

if (menu == 'n' || menu == 'N') {

newFile += 1;

for (int i = 0; i < 8; i++)

{

text[i] = " ";

emptySpaces[i] = " ";

tableLines[i] = "-";

}

filename = newFileName(filename);

system("cls");

}

else if (menu == 'i' || menu == 'I') {

insertText += 1;

if (text[0] == " ") {

cout << "Tile number ==> 1" << endl;

text[0] = textInsert(text[0]);

system ("cls");

emptySpaces[0] = emptySpace(text[0]);

tableLines[0] = tableLine(text[0]);

}

else {

do

{

cout << "Tile number ==> ";

cin >> tile;

if (tile >= 1 || tile <=8)

{

if (text[tile - 1] == " ")

{

if (text[tile - 2] == " ")

{

for (int i = 0; i < (tile - 1); i++)

{

if (text[i] == " ")

{

cout << "Please enter the text in tile " << (i + 1) << " first" << endl;

text[i] = textInsert(text[i]);

system ("cls");

emptySpaces[i] = emptySpace(text[i]);

tableLines[i] = tableLine(text[i]);

break;

}

}

}

else

{

text[tile - 1] = textInsert(text[tile - 1]);

system ("cls");

emptySpaces[tile - 1] = emptySpace(text[tile - 1]);

tableLines[tile - 1] = tableLine(text[tile - 1]);

}

}

else

{

if (text[7] == " ")

{

for (int i = 7; i > (tile - 1); i--)

{

text[i] = text[i - 1];

emptySpaces[i] = emptySpaces[i - 1];

tableLines[i] = tableLines[i - 1];

}

text[tile - 1] = textInsert(text[tile - 1]);

system ("cls");

emptySpaces[tile - 1] = emptySpace(text[tile - 1]);

tableLines[tile - 1] = tableLine(text[tile - 1]);

}

else

{

system("cls");

cout << "The tiles are already full. Please delete or overwrite any tile" << endl;

}

}

}

else

{

cout << endl;

cout << "Invalid tile number. Please try again" << endl;

}

} while (tile != 1 && tile != 2 && tile != 3 && tile != 4 && tile != 5 && tile != 6 && tile != 7 &&

tile != 8);

}

}

else if (menu == 'd' || menu == 'D') {

deleteText += 1;

do

{

cout << "Tile number ==> ";

cin >> tile;

if (tile >= 1 || tile <= 8)

{

if (text[tile - 1] == " ") {

system("cls");

emptyTextInTiles();

}

else {

for (int i = (tile - 1); i < 7; i++)

{

text[i] = text[i + 1];

emptySpaces[i] = emptySpaces[i + 1];

tableLines[i] = tableLines[i + 1];

}

text[7] = " ";

emptySpaces[7] = " ";

tableLines[7] = "-";

system("cls");

}

}

else

{

cout << endl;

cout << "Invalid tile number. Please try again" << endl;

}

}while (tile != 1 && tile != 2 && tile != 3 && tile != 4 && tile != 5 && tile != 6 && tile != 7 &&

tile != 8);

}

else if (menu == 'o' || menu == 'O') {

overwriteText += 1;

do

{

cout << "Tile number ==> ";

cin >> tile;

if (tile >= 1 || tile <= 8)

{

if (text[tile - 1] == " ") {

system("cls");

emptyTextInTiles();

}

else {

text[tile - 1] = " ";

text[tile - 1] = textInsert(text[tile - 1]);

emptySpaces[tile - 1] = emptySpace(text[tile - 1]);

tableLines[tile - 1] = tableLine(text[tile - 1]);

}

}

else

{

cout << endl;

cout << "Invalid tile number. Please try again" << endl;

}

} while (tile != 1 && tile != 2 && tile != 3 && tile != 4 && tile != 5 && tile != 6 && tile != 7 &&

tile != 8);

}

else {

system("cls");

cout << endl;

cout << "Invalid command. Please try again" << endl;

}

cout << endl;

cout << "|----+" << tableLines[0] << "--+" << tableLines[1] << "--+" << tableLines[2] << "--+"

<< tableLines[3] << "--+" << tableLines[4] << "--+" << tableLines[5] << "--+" << tableLines[6]

<< "--+" << tableLines[7] << "--|" << endl;

cout << "| |" << emptySpaces[0] << "1 |" << emptySpaces[1] << "2 |" << emptySpaces[2]

<< "3 |" << emptySpaces[3] << "4 |" << emptySpaces[4] << "5 |" << emptySpaces[5]

<< "6 |" << emptySpaces[6] << "7 |" << emptySpaces[7] << "8 |" << endl;

cout << "|----+" << tableLines[0] << "--+" << tableLines[1] << "--+" << tableLines[2] << "--+"

<< tableLines[3] << "--+" << tableLines[4] << "--+" << tableLines[5] << "--+" << tableLines[6]

<< "--+" << tableLines[7] << "--|" << endl;

cout << "| 00 | " << text[0] << " | " << text[1] << " | " << text[2] << " | " << text[3] << " | "

<< text[4] << " | " << text[5] << " | " << text[6] << " | " << text[7] << " |" << endl;

cout << "|----+" << tableLines[0] << "--+" << tableLines[1] << "--+" << tableLines[2] << "--+"

<< tableLines[3] << "--+" << tableLines[4] << "--+" << tableLines[5] << "--+" << tableLines[6]

<< "--+" << tableLines[7] << "--|" << endl;

cout << endl;

cout << "File name: " << filename << ".txt" << endl;

cout << endl;

}

}

} while (menu != 'q' && menu != 'Q');

return 0;

}

void emptyTextInTiles()

{

cout << endl;

cout << "No text detected in selected tile" << endl;

return;

}

string newFileName(string textEnter)

{

cout << "Please enter your file name" << endl;

cout << "New file name ==> ";

cin >> textEnter;

cout << endl;

return textEnter;

}

string textInsert(string textEnter)

{

cout << "Text ==> ";

cin >> textEnter;

cout << endl;

system("cls");

return textEnter;

}

void insertWrongEmptyTiles(int tileNumber)

{

cout << endl;

cout << "Please fill out tile number " << tileNumber << " first" << endl;

cout << "Tile number ==> " << tileNumber << endl;

return;

}

string emptySpace(string textEnter)

{

string emptySp = "";

for (int i = 0; i < textEnter.length(); i++)

{

emptySp += " ";

}

return emptySp;

}

string tableLine(string textEnter)

{

string tableLi = "";

for (int i = 0; i < textEnter.length(); i++)

{

tableLi += "-";

}

return tableLi;

}