Lab 8

#include "library.h"

const double space\_x= 400;

const double space\_y= 500;

//button sizes

const double button\_w= 60;

const double button\_h= 60;

//grid sizes

const double startx= 50;

const double divider= 75;

const double vdivider= 70;

char calculator\_row1[]= {'7','8','9','+','p'}; //p is plus/minus sign

char calculator\_row2[]= {'4','5','6','-','q'}; //q is squareroot

char calculator\_row3[]= {'1','2','3','x','s'}; //s is square

char calculator\_row4[]= {'C','0','=','/','!'};

int factorial(int n){

if(n==0 ||n==1){

return 1;

}else{

return n\*factorial(n-1);

}

}

void whited\_visual(){

set\_pen\_color(color::white); //prevents number visuals from overflowing screen

fill\_rectangle(startx/2+space\_x-startx,space\_y\*0.1,startx,button\_h\*2);

}

void draw\_button(double x, double y, double width, double height, char digit){

set\_font("Arial");

set\_font\_size(32);

//write\_string("string",direction::center);

double center\_x= x-width/2;

double center\_y= y-height/2;

double border\_x= width\*0.05;

double border\_y= height\*0.05;

fill\_rectangle(center\_x,center\_y,width,height,color::black);

fill\_rectangle(center\_x+border\_x-1,center\_y+border\_y-1,width\*0.90+1,height\*0.90+1,color::grey);

//write\_char(digit);

move\_to(x,y);

if(digit=='/'){ //specifically for division sign

set\_font\_size(50);

move\_to(x-12,y+15);

write\_char(L'÷');

}else if(digit=='p'){

move\_to(x-8,y+8);

write\_char(L'±');

}else if(digit=='q'){

move\_to(x-11,y+10);

write\_char(L'√');

}else if(digit=='s'){

set\_font("Times New Roman");

move\_to(x-10,y+10);

write\_char('x');

move\_to(x,y+7);

write\_char(L'²');

}else{

write\_string(digit,direction::center);

}

}

void draw\_calculator(){

// display

set\_pen\_color(color::dark\_grey);

fill\_rectangle(startx/2,space\_y\*0.1 ,space\_x-startx,button\_h\*2);

set\_pen\_color(color::black);

int counter= 0;

while(counter<5){ //row 1

draw\_button(startx+divider\*counter,250,button\_w,button\_h,calculator\_row1[counter]);

counter++;

}

counter=0;

while(counter<5){ //row 2

draw\_button(startx+divider\*counter,250+vdivider,button\_w,button\_h,calculator\_row2[counter]);

counter++;

}

counter=0;

while(counter<5){ //row 3

draw\_button(startx+divider\*counter,250+vdivider\*2,button\_w,button\_h,calculator\_row3[counter]);

counter++;

}

counter=0;

while(counter<5){ //row 4

draw\_button(startx+divider\*counter,250+vdivider\*3,button\_w,button\_h,calculator\_row4[counter]);

counter++;

}

}

double mouse\_test(){

wait\_for\_mouse\_click();

const int x = get\_click\_x(), y = get\_click\_y();

//cout << "Mouse clicked at position (" << x << ", " << y << ")\n";

int clicked\_row=0;

int clicked\_col=0;

// Row values

if(y>=250-button\_h/2 && y<=250+button\_h/2){ //250 is the center of first row buttons plus/minus half of height

clicked\_row= 1;

}else if(y>=(250+vdivider)-button\_h/2 && y<=(250+vdivider)+button\_h/2){

clicked\_row= 2;

}else if(y>=(250+vdivider\*2)-button\_h/2 && y<=(250+vdivider\*2)+button\_h/2){

clicked\_row= 3;

}else if(y>=(250+vdivider\*3)-button\_h/2 && y<=(250+vdivider\*3)+button\_h/2){

clicked\_row= 4;

}

//Column values

if(x>=startx-button\_w/2 && x<=startx+button\_h/2){

clicked\_col= 1;

} else if(x>=(startx+divider\*1)-button\_w/2 && x<=(startx+divider\*1)+button\_h/2){

clicked\_col= 2;

}else if(x>=(startx+divider\*2)-button\_w/2 && x<=(startx+divider\*2)+button\_h/2){

clicked\_col= 3;

}else if(x>=(startx+divider\*3)-button\_w/2 && x<=(startx+divider\*3)+button\_h/2){

clicked\_col= 4;

}else if(x>=(startx+divider\*4)-button\_w/2 && x<=(startx+divider\*4)+button\_h/2){

clicked\_col= 5;

}

//cout<<"You clicked row "<<clicked\_row<<" and column "<<clicked\_col<<endl;

//returning the values

switch (clicked\_col){

case 1: //COLUMN 1

switch(clicked\_row){

case 1:

return 7; //(1,1)

case 2:

return 4;//(1,2)

case 3:

return 1; //(1,3)

case 4:

return 99; //(1,4) CLEAR BUTTON

default:

break;

}

break;

case 2: //COLUMN 2

switch(clicked\_row){

case 1:

return 8; //(2,1)

case 2:

return 5;//(2,2)

case 3:

return 2; //(2,3)

case 4:

return 0; //(2,4)

default:

break;

}

break;

case 3: //COLUMN 3

switch(clicked\_row){

case 1:

return 9; //(3,1)

case 2:

return 6;//(3,2)

case 3:

return 3; //(3,3)

case 4:

return 42; //(3,4) EQUALS BUTTON

default:

break;

}

break;

case 4: //COLUMN 4

switch(clicked\_row){

case 1:

return 10; //(4,1) PLUS BUTTON

case 2:

return 11;//(4,2) MINUS BUTTON

case 3:

return 12; //(4,3) TIMES BUTTON

case 4:

return 13; //(4,4) DIVISION BUTTON

default:

break;

}

break;

case 5: //COLUMN 5

switch(clicked\_row){

case 1:

return 55; //(5,1) DECIMAL

case 2:

return -1;//(5,2) SQRT

case 3:

return 22; //(5,3) SQ

case 4:

return 9001; //(5,4) FACTORIAL

default:

break;

}

default:

break;

}

//cout<<"end of loop"<<endl;

return 69; //value for null

}

void refresh(){

set\_pen\_color(color::dark\_grey);

fill\_rectangle(startx/2,space\_y\*0.1 ,space\_x-startx,button\_h\*2); //refresh

}

double number(double calculation){

//double calculation= 0;

double saved\_num=0;

int mod\_version=0;

while(true){

int x= mouse\_test();

//specific conditions

switch (x){

case 10: //PLUS BUTTON

mod\_version=10;

saved\_num=saved\_num+calculation;

cout<<"saved num: "<<saved\_num<<endl;

refresh();

calculation= 0;

//number();

break;

case 11: //MINUS BUTTON

mod\_version=11;

saved\_num=saved\_num+calculation;

cout<<"saved num: "<<saved\_num<<endl;

refresh();

calculation= 0;

//number();

break;

case 12: //MULTIPLYING

mod\_version=12;

saved\_num=saved\_num+calculation;

cout<<"saved num: "<<saved\_num<<endl;

refresh();

calculation= 0;

//number();

break;

case 13: //DIVISION

mod\_version=13;

saved\_num=saved\_num+calculation;

cout<<"saved num: "<<saved\_num<<endl;

refresh();

calculation= 0;

//number();

break;

case 42: //EQUALS BUTTON

if(mod\_version==10){ //ADDING

//debug

cout<<"adding "<<saved\_num<<" and "<<calculation<<endl;

//cout<<calculation<<endl;

//

refresh();

set\_pen\_color(color::yellow);

move\_to(startx\*0.7,space\_y\*0.1+button\_h);

write\_string(saved\_num+calculation,direction::east);

cout<<"exiting number function.."<<endl;

return saved\_num+calculation;

}else if(mod\_version==11){ //SUBSTRACTING

//debug

cout<<"subtracting "<<saved\_num<<" and "<<calculation<<endl;

//cout<<calculation<<endl;

//

refresh();

set\_pen\_color(color::yellow);

move\_to(startx\*0.7,space\_y\*0.1+button\_h);

write\_string(saved\_num-calculation,direction::east);

cout<<"exiting number function.."<<endl;

return saved\_num-calculation;

}else if(mod\_version==12){ //SUBSTRACTING

//debug

cout<<"Multiplying "<<saved\_num<<" and "<<calculation<<endl;

//cout<<calculation<<endl;

//

refresh();

set\_pen\_color(color::yellow);

move\_to(startx\*0.7,space\_y\*0.1+button\_h);

write\_string(saved\_num\*calculation,direction::east);

cout<<"exiting number function.."<<endl;

return saved\_num\*calculation;

}else if(mod\_version==13){ //DIVISION

//debug

cout<<"Dividing "<<saved\_num<<" and "<<calculation<<endl;

//cout<<calculation<<endl;

//

refresh();

set\_pen\_color(color::yellow);

move\_to(startx\*0.7,space\_y\*0.1+button\_h);

write\_string(saved\_num/calculation,direction::east);

cout<<"exiting number function.."<<endl;

return saved\_num/calculation;

}

cout<<"fail"<<endl;

break;

case 99: //CLEAR BUTTON

refresh();

//return 0;

calculation=0;

case 69: //ANYWHERE THAT IS NOT BUTTON

calculation= calculation;

break;

case 55: //plus/minus button

calculation= -calculation;

break;

case -1: //square root button

refresh();

set\_pen\_color(color::yellow);

move\_to(startx\*0.7,space\_y\*0.1+button\_h);

write\_string(sqrt(calculation),direction::east);

cout<<"exiting number function.."<<endl;

return sqrt(calculation);

case 22: //square button

refresh();

set\_pen\_color(color::yellow);

move\_to(startx\*0.7,space\_y\*0.1+button\_h);

write\_string(calculation\*calculation,direction::east);

cout<<"exiting number function.."<<endl;

return calculation\*calculation;

case 9001: //factorial button

refresh();

set\_pen\_color(color::yellow);

move\_to(startx\*0.7,space\_y\*0.1+button\_h);

write\_string(factorial(calculation),direction::east);

cout<<"exiting number function.."<<endl;

return factorial(calculation);

default: //WHAT NORMALLY RUNS

calculation= x+calculation\*10;

}

cout<<"out of swtich "<<endl;

//printing values on the screen

set\_font\_size(80);

set\_pen\_color(color::dark\_grey);

refresh();

set\_pen\_color(color::yellow);

move\_to(startx\*0.7,space\_y\*0.1+button\_h);

write\_string(calculation,direction::east);

whited\_visual();

} //end of while loop

return calculation;

}

void main(){

make\_window(space\_x,space\_y);

draw\_calculator();

double calculation = 0;

while(true){

calculation=number(calculation);

cout<<"main function"<<endl;

cout<<calculation<<endl;

whited\_visual();

}

}

