/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Exercise 4.1. The program pick.cpp demonstrated how to pick a color and change the background color of a widget.

This exercise is to take elements of pick.cpp and merge them into the current program to change color of the polygon.

Hand in your program listing. Mail to me a screen shot of an aquamarine hexagon. It need not be a regular hexagon.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**File Polygon.h**

#ifndef POLYGON\_

#define POLYGON\_

#include <FL/Fl\_Box.H>

class Polygon : public Fl\_Widget

{

float xv[100];

float yv[100];

void draw();

int handle(int event);

void marks();

void fill();

public:

// change for homework 4

Fl\_Color polygon\_color;

int n;

bool state;

// constructor

Polygon(int x, int y, int w, int h);

// default destructor

~Polygon();

};

#endif

**File Constants.h**

#ifndef CONSTANTS

#define CONSTANTS

const int bh = 30; // button height

const int bw = 100; // button width

const int sp = 5; // a little space

const int ww = 600; // window width

const int wh = 600; // window height

#endif

**File Polygon.cpp**

#include <cstdlib>

#include <cstdio>

#include <cstring>

#include <FL/fl\_draw.H>

#include <FL/Fl.H>

// custom header files

#include "Constants.h"

#include "Polygon.h"

Polygon::Polygon(int x, int y, int w, int h) : Fl\_Widget(x, y ,w ,h)

{

box(FL\_ENGRAVED\_BOX);

n = 0;

// change for homework 4

// default color - wildcat blue

polygon\_color = 0x0033a000;

}

Polygon::~Polygon()

{

}

void Polygon::marks()

{

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

fl\_rectf(x()+xv[i]-2, y()+yv[i]-2, 5, 5);

redraw();

}

void Polygon::fill()

{

// change for homework 4 - change the color of the polygon

fl\_color(polygon\_color);

fl\_begin\_complex\_polygon();

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

fl\_vertex(x() + xv[i], y() + yv[i]);

fl\_end\_complex\_polygon();

redraw();

}

// this draw rountine will be called in redraw()

void Polygon::draw()

{

// draw background

fl\_color(FL\_WHITE);

fl\_rectf(x(), y(), w(), h());

fl\_color(FL\_BLACK);

fl\_rect(x(), y(), w(), h());

if(state and n > 2)

fill();

else

marks();

}

int Polygon::handle(int event)

{

int ret = Fl\_Widget::handle(event);

switch (event)

{

case FL\_PUSH:

ret = 1;

break;

case FL\_DRAG:

ret = 1;

break;

case FL\_RELEASE:

xv[n] = Fl::event\_x() - x();

yv[n] = Fl::event\_y() - y();

n++;

ret = 1;

redraw();

break;

default:

ret = Fl\_Widget::handle(event);

}

return ret;

}

**File poly.cpp**

#include <FL/Fl.H>

#include <FL/Fl\_Double\_Window.H>

#include <FL/Fl\_Box.H>

#include <FL/Fl\_Check\_Button.H>

#include <FL/Fl\_Group.H>

#include <FL/fl\_show\_colormap.H>

// custom header files

#include "Constants.h"

#include "Polygon.h"

// global variable initializations

Polygon \*poly = 0;

Fl\_Box \*spacer = 0;

Fl\_Check\_Button \*fill = 0;

Fl\_Group \*group = 0;

Fl\_Button \*clear = 0;

// change for homework 4

Fl\_Button \*choose\_color = 0;

// callback to fill polygon

void fill\_cb(Fl\_Widget\* w, void\*)

{

poly->state = ((Fl\_Check\_Button\*) w)->value();

poly->redraw();

}

// callback to clear the polygon

void clear\_cb(Fl\_Widget\* w, void\*)

{

poly->n = 0;

poly->redraw();

}

// change for homework 4

// a callback to change the color of polygon when a color is chosen

void choose\_colr\_cb(Fl\_Widget\*, void\*)

{

Fl\_Color selected\_color = fl\_show\_colormap(poly->polygon\_color);

poly->polygon\_color = selected\_color;

poly->redraw();

}

// main function

int main()

{

Fl\_Double\_Window\* win = new Fl\_Double\_Window(ww, wh, "Draw a Polygon");

win->begin();

poly = new Polygon(sp, sp, ww-2\*sp, wh-bh-3\*sp);

int x = sp;

int z = wh - bh -sp;

group = new Fl\_Group(x, z, ww-2\*sp, bh);

group->begin();

fill = new Fl\_Check\_Button(x, z, bw, bh, "Fill");

x += bw + sp;

clear = new Fl\_Button(x, z, bw, bh, "Clear");

x += bw + sp;

// change for homework 4

choose\_color = new Fl\_Button(x, z, bw, bh, "Choose color");

x += bw + sp;

spacer = new Fl\_Box(FL\_NO\_BOX, x, z, 1, bh, "");

group->end();

group->resizable(spacer);

win->resizable(poly);

win->end();

// assign callbacks

fill->callback(fill\_cb);

clear->callback(clear\_cb);

// change for homework 4

choose\_color->callback(choose\_colr\_cb);

win->show();

return(Fl::run());

}

**Makefile**

# automatic variables

# $^ the names of all the prerequsites

# $@ the name of the target

# $< the name of the first prerequsite

INCLUDES = `fltk-config --cxxflags`

LIBRARY = `fltk-config --ldstaticflags`

.PHONY : clean

poly : poly.o Polygon.o

    g++ -g -Wall $^ ${LIBRARY} -o $@

Polygon.o : Polygon.cpp Polygon.h

    g++ -g -Wall ${INCLUDES} -c $< -o $@

poly.o : poly.cpp Polygon.h

    g++ -g -Wall ${INCLUDES} -c $< -o $@

clean:

    rm \*.o poly

**Output:**

