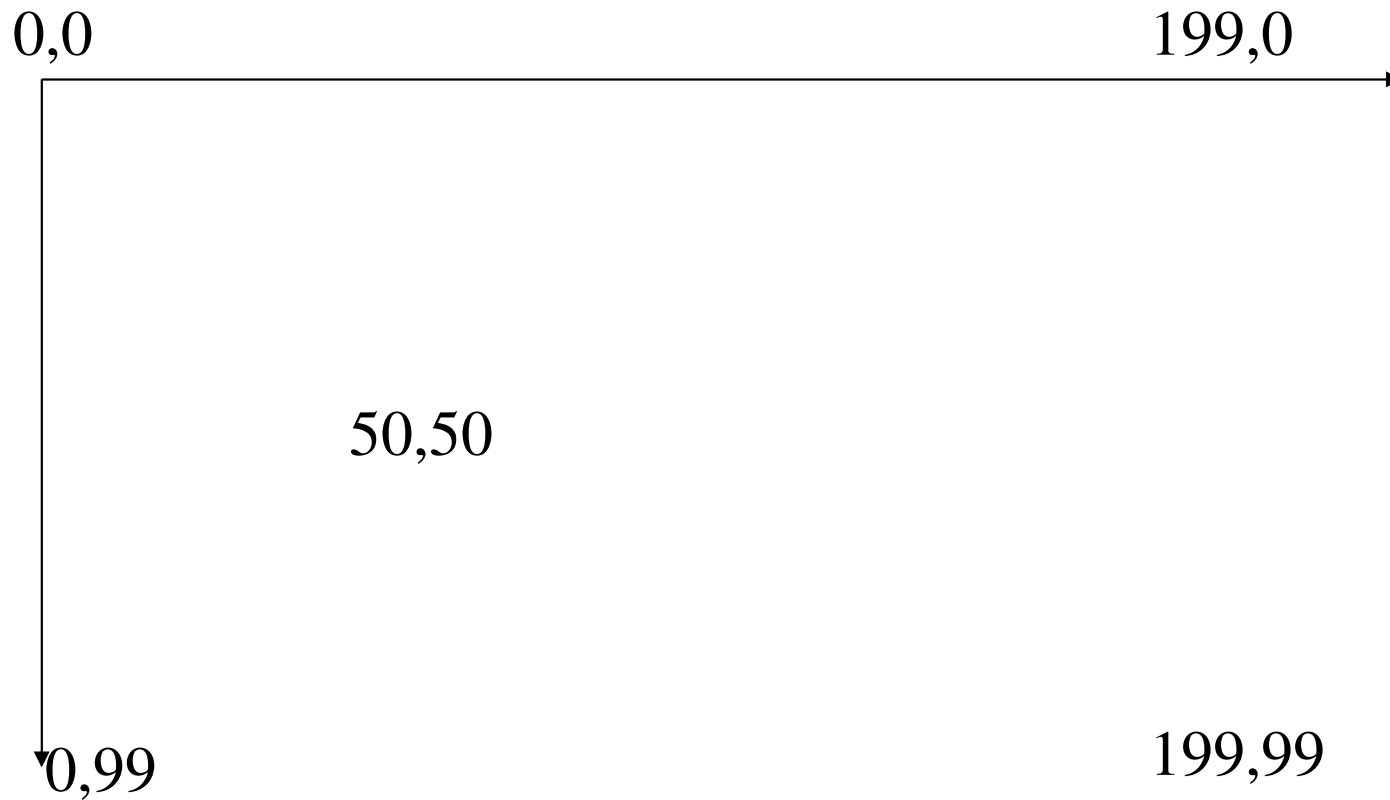


Модел графичког приказа

Зашто графички приказ?

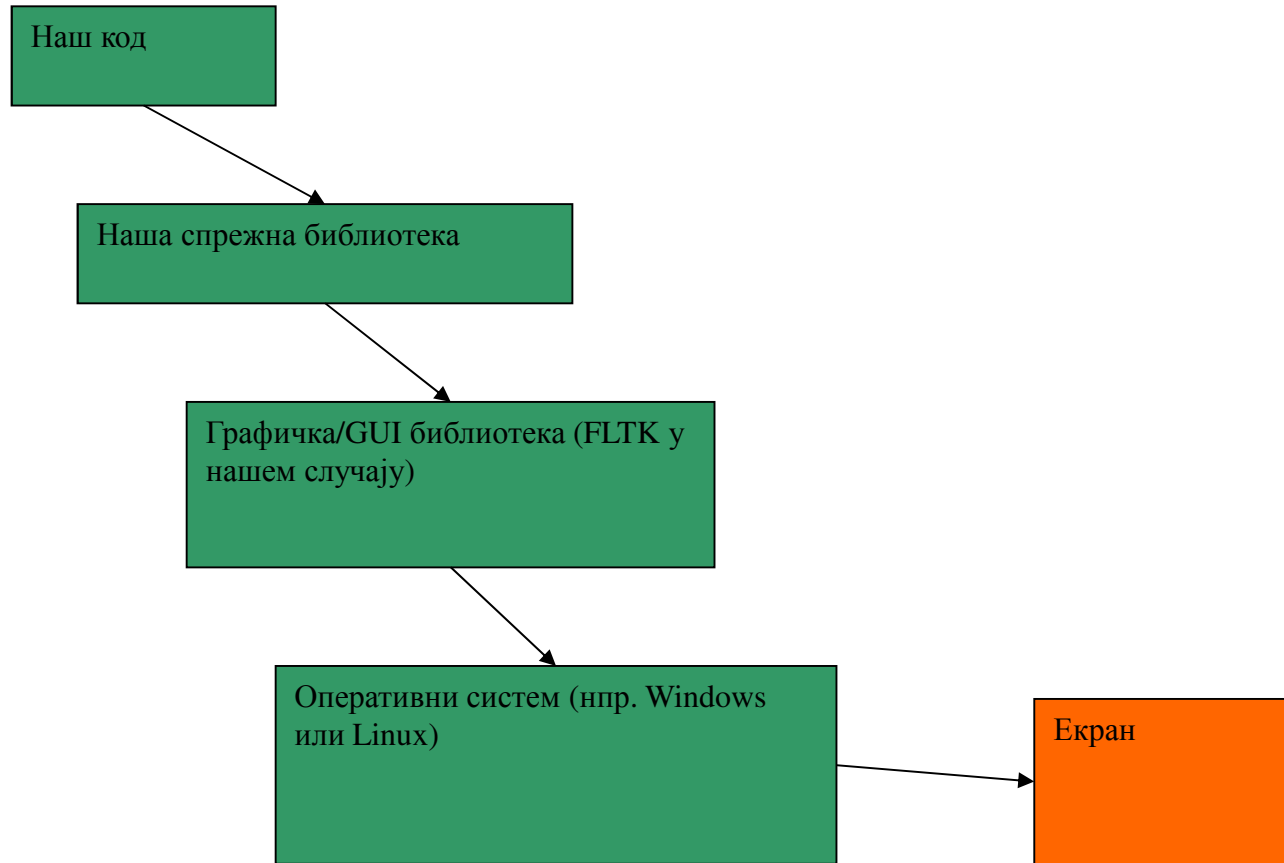
- Данас је то уобичајена спрега са корисницима
- Погодно је да илуструје неке корисне концепте и технике
- Доста zgodних, малих примера
- Није тривијално – исплати се знати
- Забавно је

Координатни систем



- Пиксел (Picture Element -> Pixel)

Структура графичке спреге



- Слојевита архитектура

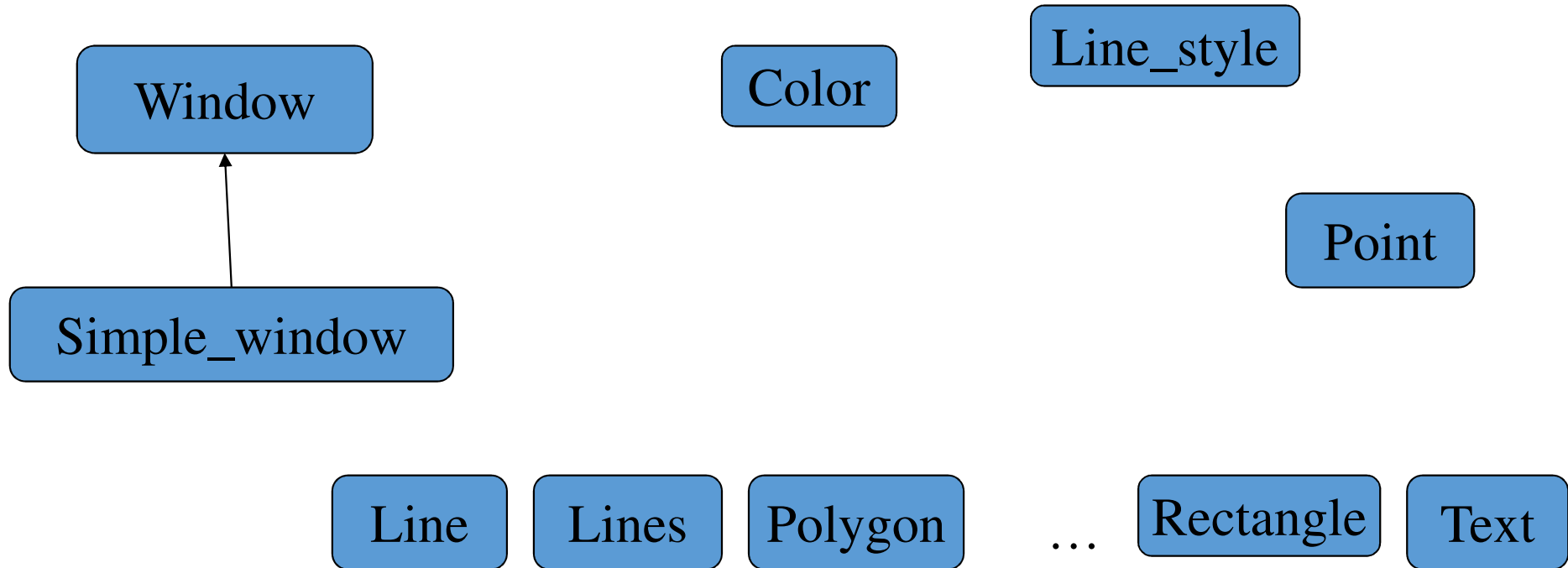
Графичке/GUI библиотеке

- Наша спрежна библиотека се наслања на графичку библиотеку.
 - На овај начин можемо брзо и једноставно да почнемо рад са графиком, а уједно нам омогућава увид у проблематику која се јавља код графичких спрежних библиотека.
 - Наша спрега има мање од 20 класа и мање од 500 линија кода
 - Спрега се може лако проширивати
- У нашем конкретном случају се наслања на FLTK (Fast Light Tool Kit), али би се могла наслањати на било коју
 - Наша спрега се може, по потреби, лако прилагодити некој другој основној библиотеци

Графичке/GUI библиотеке

- Коришћење проверених библиотека (као што је FLTK) доноси неколико предности
- Код је преносив
 - Windows, Unix, Mac, ...
- Библиотека обично већ има подршку за већину уобичајених случајева коришћења

Спрежне класе

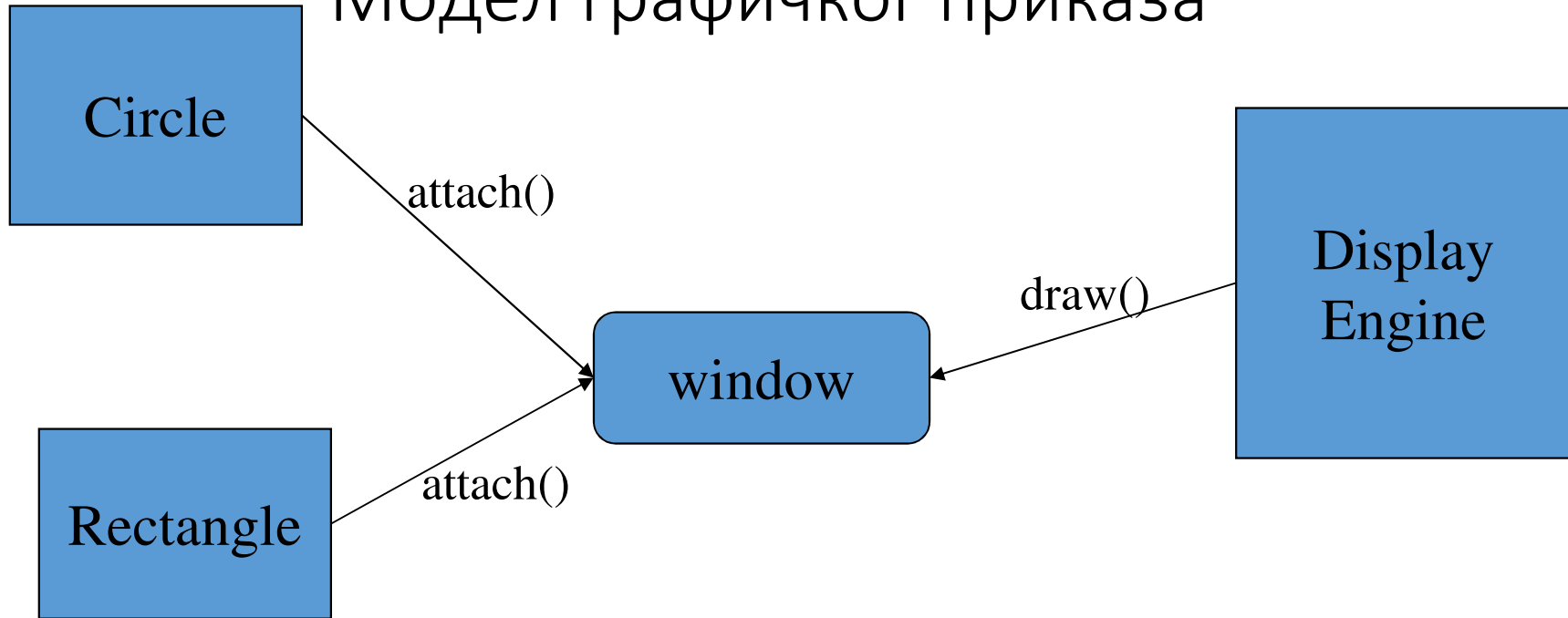


- Color, Line_style и Point су **помоћне класе**
- Window је главна спрега према графичкој/GUI библиотеци (FLTK)

Спрежне класе

- Тренутне:
 - Color, Line_style, Font, Point (помоћне класе)
 - Window, Simple_window
 - Text, Polygon, Line, Lines, Rectangle, Circle, ...
 - Axis
- Могу се лако додати:
 - Grid, Block_chart, Pie_chart, ...
- Елементи графичке корисничке спреге:
 - Button, In_box, Out_box, ...

Модел графичког приказа



- Облици се „каче“ на прозор.
- Графички подсистем позива команде приказа (типа: „нацртај линију од x до y “) за објекте који су накачени на прозор

Први пример – празан прозор

```
// Укључивање спреге ка библиотекама
#include "Simple_window.h" // за сада користимо једноставан прозор
#include "Graph.h" // графички облици

using namespace Graph_lib; // чинимо имена доступним
                            // у VC2013 негде се мора експлицитно
                            // користити Graph_lib::

// у main():

Simple_window win(Point(100, 100), 600, 400, "Canvas");
    // Координате (100,100) горње леве тачке прозора
    // Величина прозора (600 пиксела ширина и 400 висина)
    // Назив: Canvas (Платно)
win.wait_for_button(); // Прикажи
//gui_main();
```

Други пример - правоугаоник

```
Rectangle r(Point(200, 200), 100, 50);  
    // горња лева тачка, ширина и висина  
  
win.attach(r);  
win.wait_for_button();
```

Трећи пример – координатна оса

```
Axis xa(Axis::x, Point(20, 300), 280, 10, "x axis");  
  // Axis::x значи да је хоризонтална  
  // почиње од (20,300)  
  // дугачка 280 пиксела  
  // 10 подеока  
  // назив "x axis"  
  
win.attach(xa);  
win.wait_for_button();
```

Четврти пример – координатна оса

```
Axis ya(Axis::y, Point(20, 300), 280, 10, "y axis");  
ya.set_color(Color::cyan); // бирамо боју за осу  
ya.label.set_color(Color::dark_red); // посебна боја за назив  
  
win.attach(ya);  
win.wait_for_button();
```

Пети пример – синус

```
win.set_label("Canvas #4");
```

```
Function sine(sin, 0, 100, Point(20, 150), 1000, 50, 50);
```

```
    // исцртај  $\sin()$  у опсегу  $[0:100)$ 
```

```
    // са координатним почетком у  $(20,150)$ 
```

```
    // са 1000 тачака
```

```
    // скалирај по  $x$  и по  $y$  са 50
```

```
win.attach(sine);
```

```
win.wait_for_button();
```

Шести пример – полигон (троугао)

```
Polygon poly;  
poly.add(Point(300, 200));  
poly.add(Point(350, 100));  
poly.add(Point(400, 200));  
  
poly.set_color(Color::red);  
poly.set_style(Line_style::dash);  
  
win.attach(poly);  
win.wait_for_button();
```

Седми пример – затворени низ линија (правоугаоник)

- Облик који изгледа као правоугаоник

```
Closed_polyline poly_rect;  
poly_rect.add(Point(100, 50));  
poly_rect.add(Point(200, 50));  
poly_rect.add(Point(200, 100));  
poly_rect.add(Point(100, 100));
```


Осми пример – затворени низ линија

- ...али није квадрат

```
poly_rect.add(Point(50, 75)); // сада poly_rect има 5 тачака
```

Бојење унутрашњости облика (пуњење)

```
r.set_fill_color(Color::yellow);
```

```
poly.set_style(Line_style(Line_style::dash, 4));
```

```
poly_rect.set_fill_color(Color::green);
```

```
poly_rect.set_style(Line_style(Line_style::dash, 2));
```

Осми пример – текст

```
Text t(Point(100, 100), "Hello, graphical world!");  
    // позиција се односи на доњу леву тачку текста
```

Девети пример – текст (изглед и величина слова)

```
t.set_font(Font::times_bold);  
t.set_font_size(20); // висина у пикселима
```

Девети пример – слика

```
Image ii(Point(100, 50), "image.jpg");  
win.attach(ii);
```

Десети пример – померање слике

```
ii.move(100, 200);    // помери за 100 пиксела у десно (-100 је у лево)  
                       // помери за 200 пиксела доле (-200 је горе)  
  
win.wait_for_button();
```

Једанаести пример – круг, елипса, тачка

```
Circle c(Point(100, 200), 50);
```

```
Ellipse e(Point(100, 200), 75, 25);
```

```
e.set_color(Color::dark_red);
```

```
Mark m(Point(100, 200), 'x');
```

```
ostringstream oss;
```

```
oss << "screen size: " << x_max() << "*" << y_max()
```

```
    << "; window size: " << win.x_max() << "*" << win.y_max();
```

```
Text sizes(Point(100, 20), oss.str());
```

ОМОТ

```
#include "Graph.h"
#include "Simple_window.h"

int main ()
try
{
    // главни део кода
}
catch(exception& e) {
    cerr << "exception: " << e.what() << '\n';
    return 1;
}
catch (...) {
    cerr << "Some exception\n";
    return 2;
}
```