I know this topic was discussed some (long) time ago, but do we really have those two different structures? It makes things more complicated especially if one uses rainbow renderer that supports transparency but due to draw interface he cannot easily use it.

For example I would like to paint transparent red rectangle. Normally I would do:

```
void Paint(Draw& w) {
    w.DrawRect(10, 10, 100, 100, Color(Red, 128));
}
```

Instead I need to cast draw to SystemDraw. In framebuffer it lookes like this:

```
void Paint(Draw& w) {

    SystemDraw& sw = (SystemDraw&) w;
    col.r = Red().GetR();
    col.g = Red().GetG();
    col.b = Red().GetB();
    col.a = 128;
    sw.RectPath(10, 10, 100, 100).Fill(col);
}
```

In WinGL

```
void Paint(Draw& w) {

    SystemDraw& sw = (SystemDraw&) w;
    sw.alpha = 128;
    sw.DrawRect(10, 10, 100, 100, Red);
}
```

Could not Color simply support transparency as RGBA? (in rainbow backend without transparency this part of information from Color would be simply ignored (not used))

PS: I also wonder if it wouldn't be better if Draw interface used doubles instead of integers as coordinates)