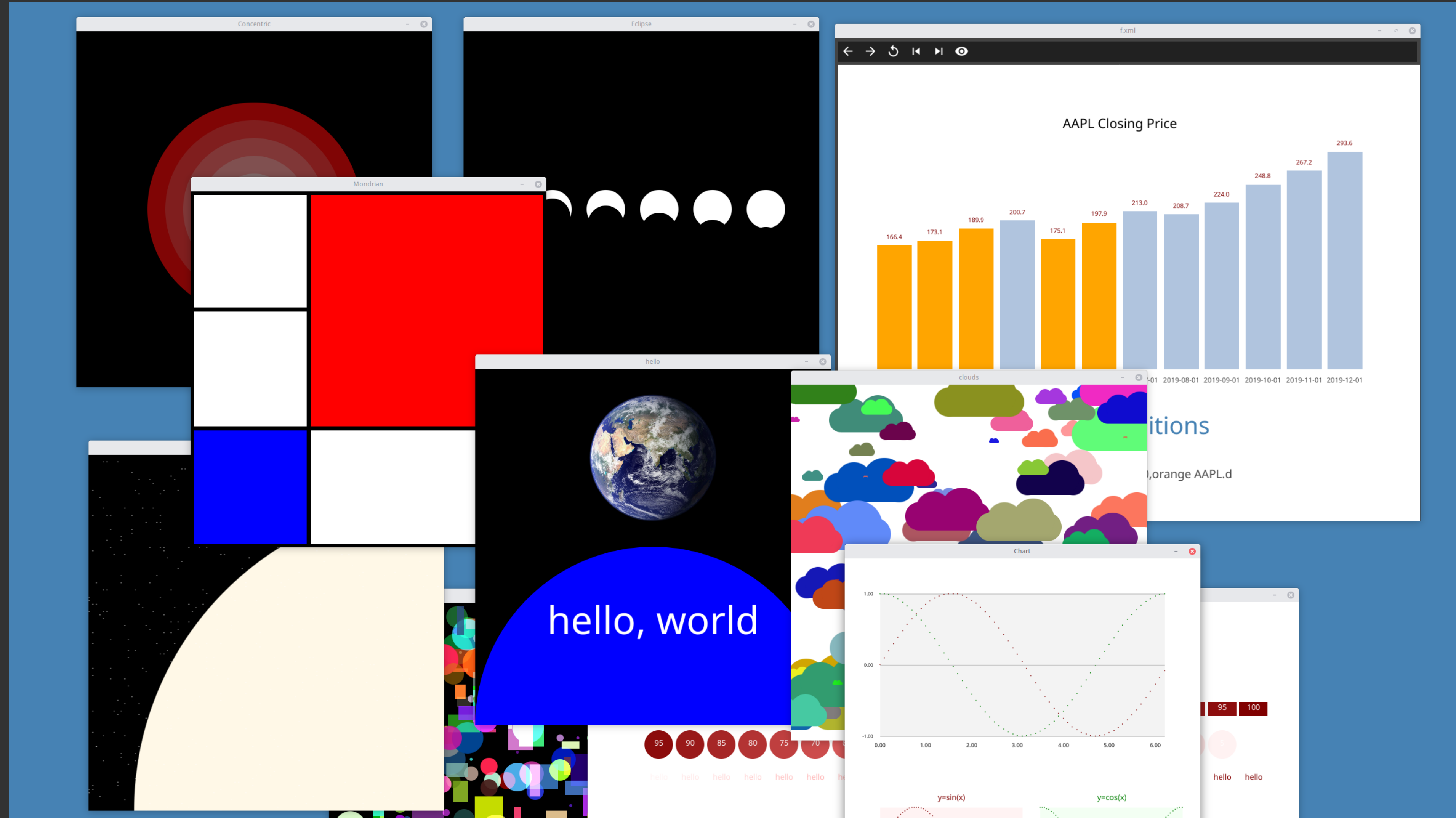


fc: a high-level canvas API for the fyne toolkit

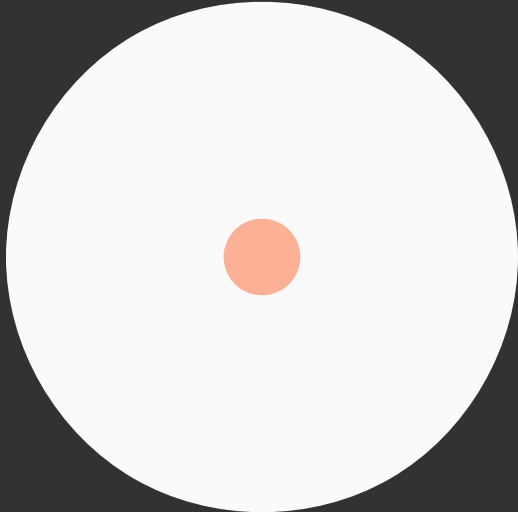


Elements

Text

CText

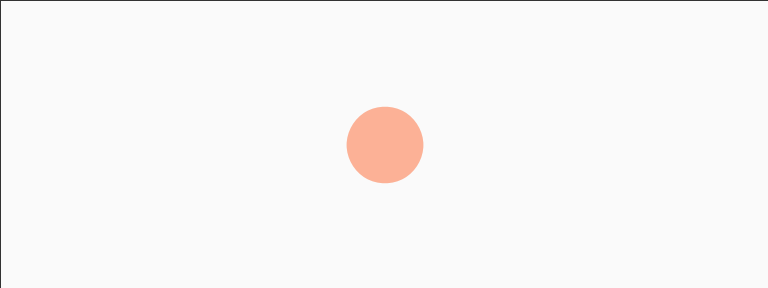
EText



circle



line



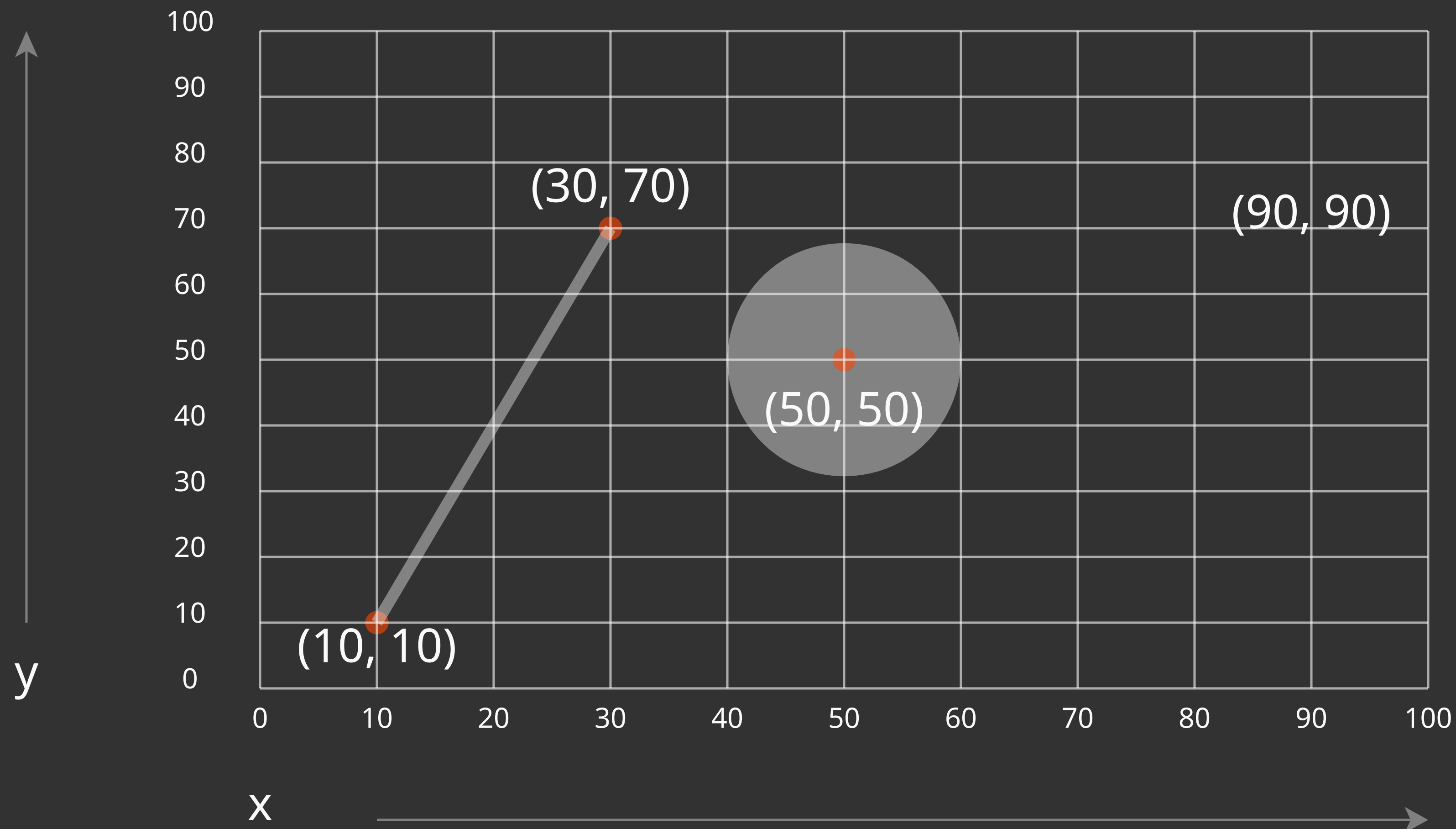
rectangle



image

Percentage-based Grid

Using the Percentage-based Grid



Rect(90, 80, ...)

Circle(50, 50, ...)

Line(10, 10, 30, 70, ...)

Percentage-based Methods

Make a new canvas	<code>NewCanvas(name string, w, h int) Canvas</code>
Place text, left-aligned	<code>Text(x, y, size float64, s string, fill color.RGBA)</code>
Place centered text	<code>CText(x, y, size float64, s string, fill color.RGBA)</code>
Place end-aligned text	<code>EText(x, y, size float64, s string, fill color.RGBA)</code>
Obtain the text width	<code>TextWidth(s string, size float64) float64</code>
Make a circle centered (x,y)	<code>Circle(x, y, r float64, fill color.RGBA)</code>
Rectangle with corner at (x,y)	<code>CornerRect(x, y, w, h float64, fill color.RGBA)</code>
Rectangle centered at (x,y)	<code>Rect(x, y, w, h float64, fill color.RGBA)</code>
Draw line from (x1,y) to (x2,y2)	<code>Line(x1, y1, x2, y2, size float64, stroke color.RGBA)</code>
Place an image centered at (x,y)	<code>Image(x, y float64, w, h int, name string)</code>
Display and run	<code>EndRun()</code>

Convenience methods

Lookup colors by name

```
ColorLookup(s string) color.RGBA
```

Map one range into another

```
MapRange(value, low1, high1, low2, high2 float64) float64
```

Polar to Cartesian coordinates

```
Polar(x, y, r, angle float64) (float64, float64)
```

Convert degrees to radians

```
Radians(deg float64) float64
```

Absolute Coordinate Methods

Place text, left-aligned	<code>AbsText(c *fyne.Container,x,y int,s string,size int,fill color.RGBA)</code>
Place centered text	<code>AbsTextMid(c *fyne.Container,x,y int,s string,size int,fill color.RGBA)</code>
Place end-aligned text	<code>AbsTextEnd(c *fyne.Container,x,y int,s string,size int,fill color.RGBA)</code>
Make a circle centered (x,y)	<code>AbsCircle(c *fyne.Container,x,y,r int,fill color.RGBA)</code>
Rectangle with corner at (x,y)	<code>AbsCornerRect(c *fyne.Container,x,y,w,h int,fill color.RGBA)</code>
Rectangle centered at (x,y)	<code>AbsRect(c *fyne.Container,x,y,w,h int,fill color.RGBA)</code>
Draw line from (x1,y) to (x2,y2)	<code>AbsLine(c *fyne.Container,x1,y1,x2,y2 int,size float32,stroke color.RGBA)</code>
Image upper-left corner at (x,y)	<code>AbsCornerImage(c *fyne.Container,x,y,w,h int,name string)</code>
Image centered at (x,y)	<code>AbsImage(c *fyne.Container,x,y,w,h int,name string)</code>
New app context	<code>AbsStart(name string,w,h int) (fyne.Window,*fyne.Container)</code>
Display and run	<code>AbsEndRun(window fyne.Window,c *fyne.Container,w,h int)</code>

hello, (fc) world

```
package main

import (
    "image/color"

    "github.com/ajstarks/fc"
)

func main() {
    width, height := 500, 500
    blue := color.RGBA{0, 0, 255, 255}
    white := color.RGBA{255, 255, 255, 255}

    canvas := fc.NewCanvas("hello", width, height)

    canvas.Circle(50, 0, 100, blue)
    canvas.CText(50, 25, 10, "hello, world", white)
    canvas.Image(50, 75, 200, 200, "earth.jpg")

    canvas.EndRun()
}
```

