

Makie

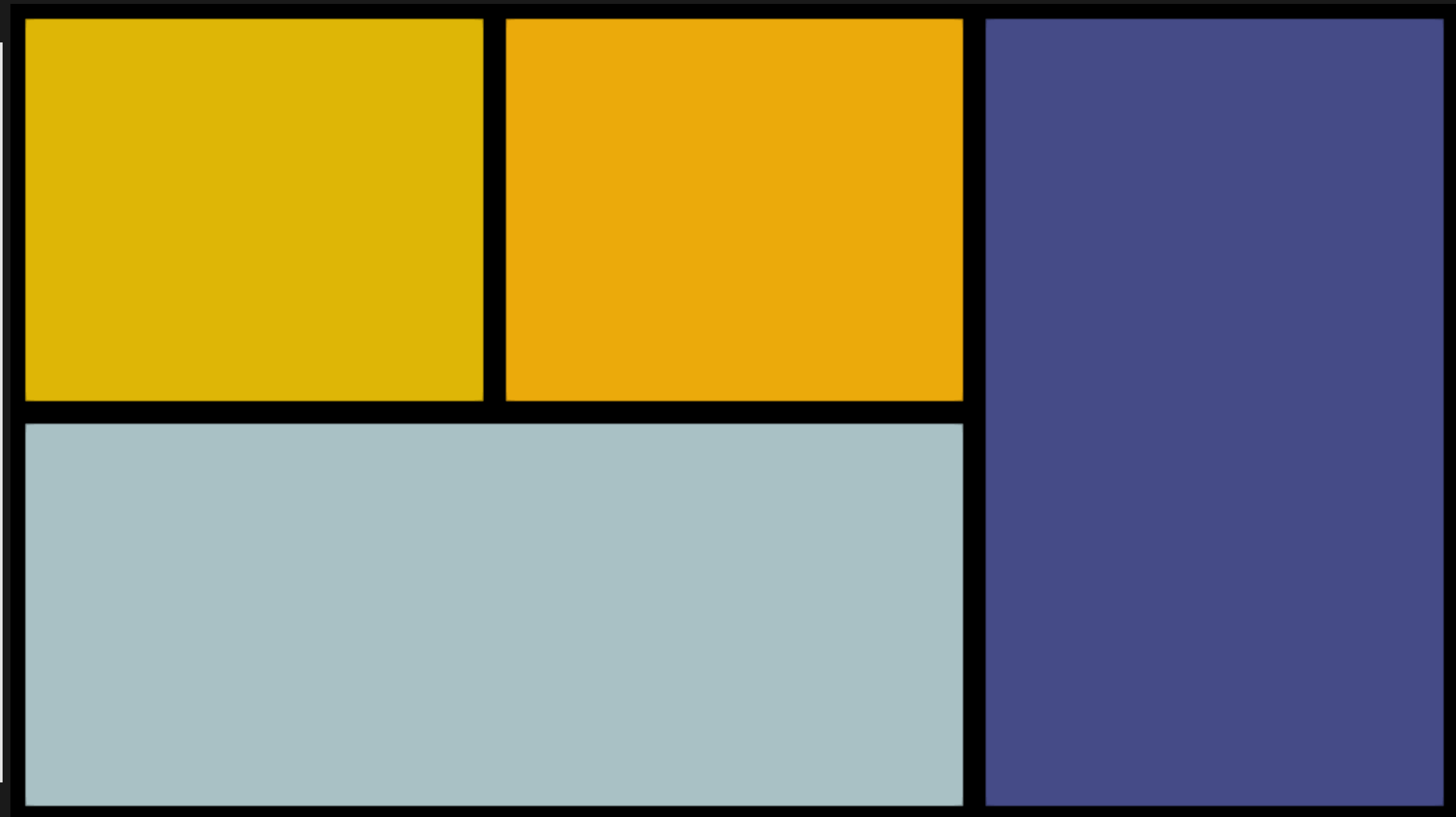
GridLayouts

Danisch & Krumbiegel, (2021). Makie.jl: Flexible high-performance data visualization for Julia.
[Journal of Open Source Software](#), 6(65), 3349

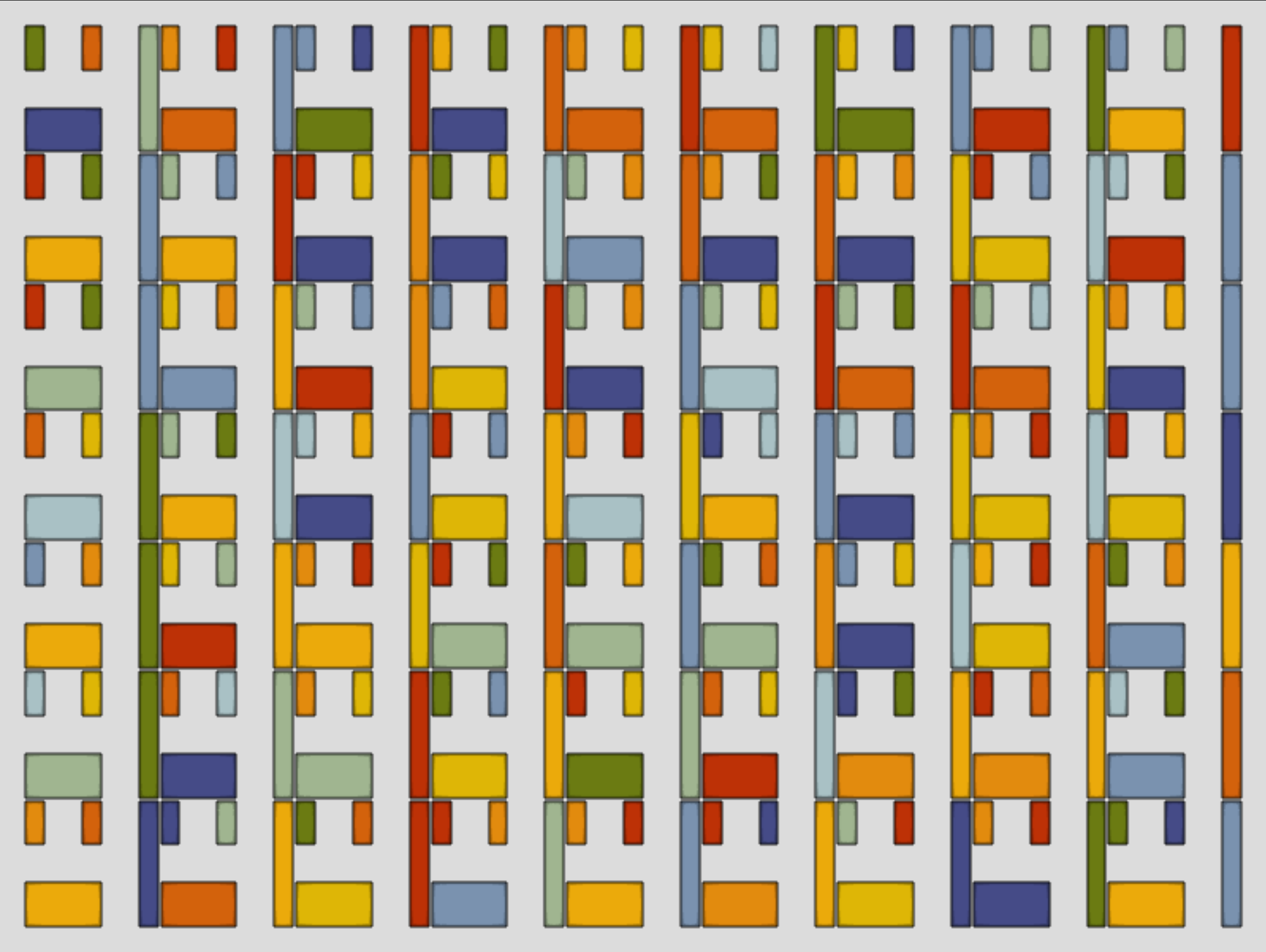
```

using GLMakie, Colors, Random
using ColorSchemes
GLMakie.activate!()
GLMakie.set_window_config!(float=true)
Random.seed!(12133)
## Nested boxes
function nested_boxes!(fig; strokecolor = :black)
    colors = shuffle(resample_cmap(:VanGogh2, 10))
    b1 = Box(fig[1, 1]; color=colors[1], strokecolor)
    b2 = Box(fig[1, 2]; color=colors[2], strokecolor)
    b3 = Box(fig[2, 1:2]; color=colors[3], strokecolor)
    b4 = Box(fig[1:2, 3]; color=colors[4], strokecolor)
    #return nothing #(b1, b2, b3, b4)
end
fig = Figure(backgroundcolor= :black, resolution = (1600,900))
nested_boxes!(fig)
fig

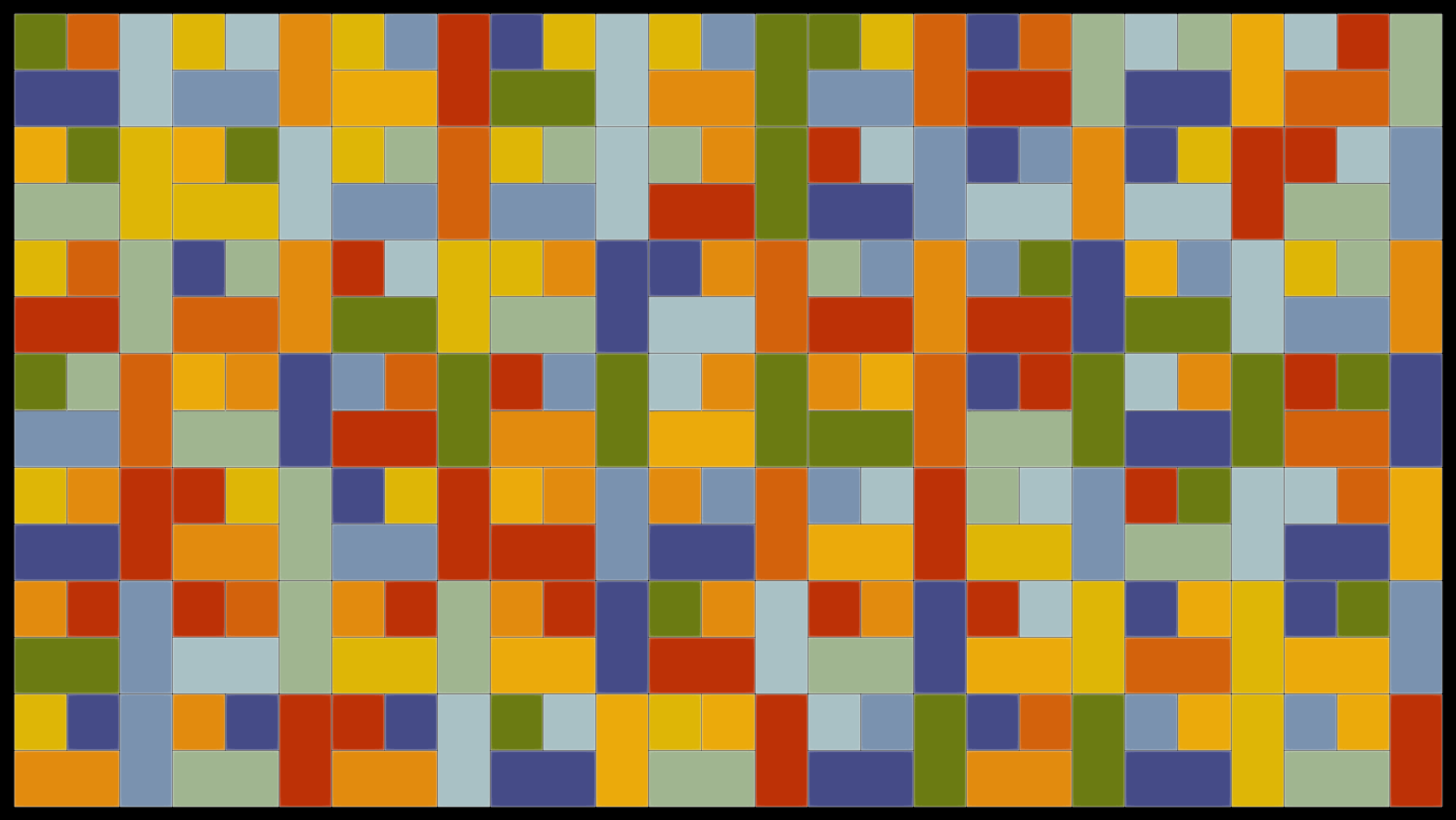
```



```
fig = Figure(backgroundColor = :gainsboro)
[nested_boxes!(fig[i,j]) for i in 1:7, j in 1:9]
colgap!(fig.layout, 2)
rowgap!(fig.layout, 2)
fig
```



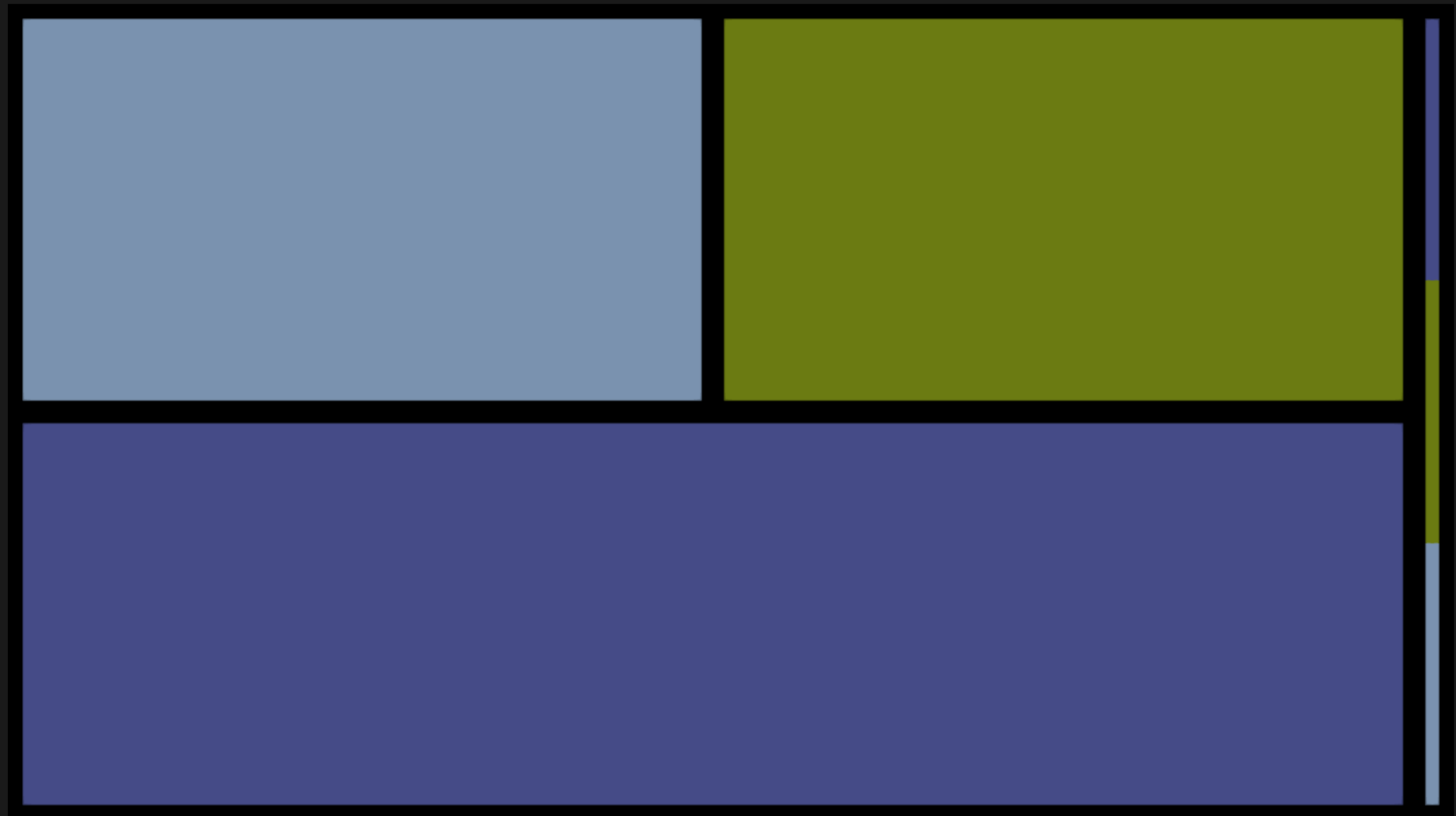
```
fig = Figure(backgroundColor= :black, resolution = (1600,900))
gls = [fig[i, j] = GridLayout() for i in 1:7, j in 1:9]
[nested_boxes!(g; strokecolor = (:white, 0.5)) for g in gls]
colgap!.(gls, 1)
rowgap!.(gls, 1)
colgap!(fig.layout, 1)
rowgap!(fig.layout, 1)
fig
```



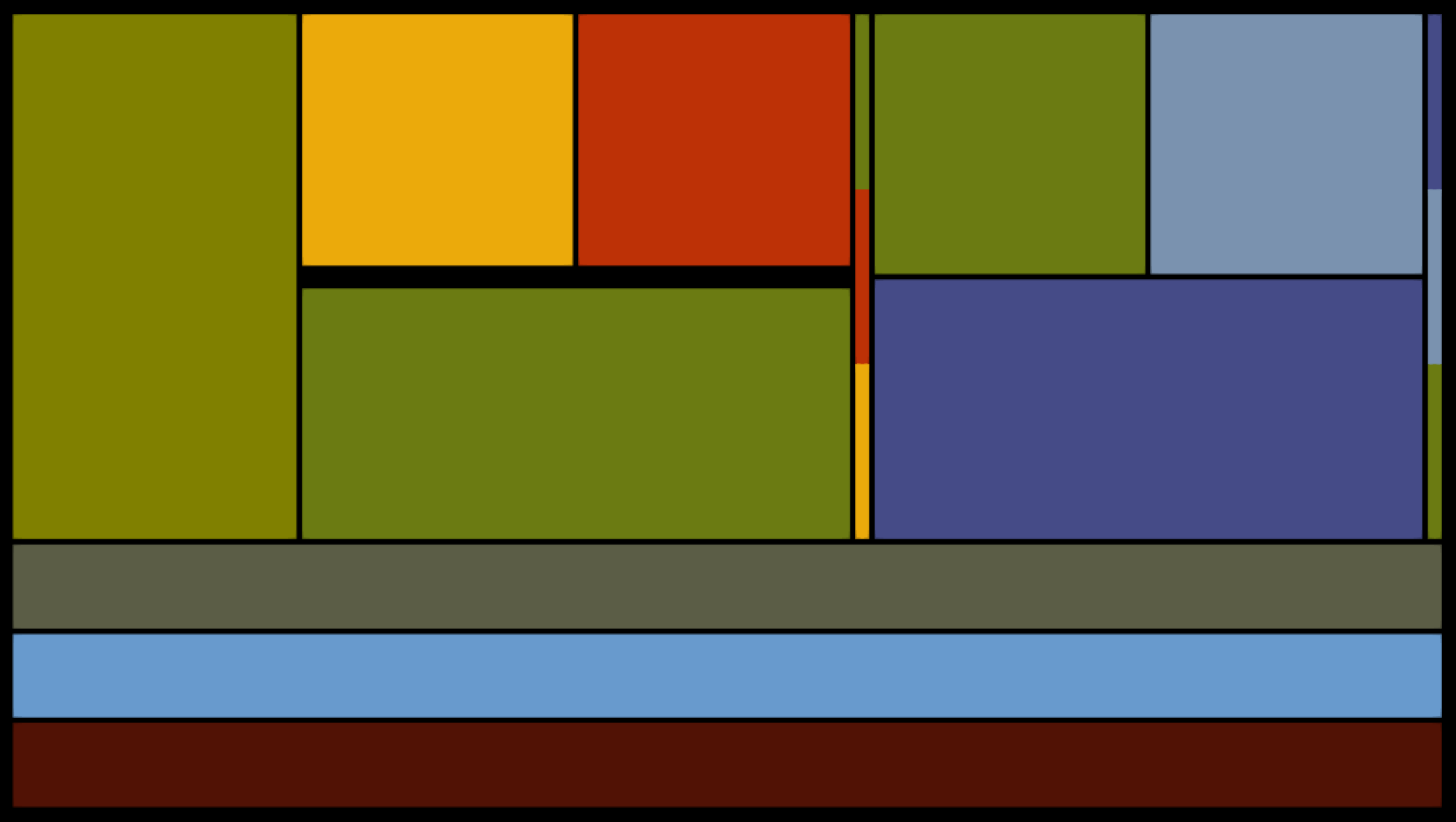
```

using GLMakie, Colors, Random
using ColorSchemes
GLMakie.activate!()
GLMakie.set_window_config!(float=true)
Random.seed!(21343)
function nested_boxes_cbar!(fig)
    colors = shuffle(resample_cmap(:VanGogh2, 10))
    b1 = Box(fig[1, 1], color=colors[1])
    b2 = Box(fig[1, 2], color=colors[2])
    b3 = Box(fig[2, 1:2], color=colors[3])
    b4 = Colorbar(fig[1:2, 3],
        colormap=cgrad(ColorScheme(colors[1:3])),
        categorical = true,
        ticksvisible=false,
        ticklabelsvisible=false)
    #return nothing #(b1, b2, b3, b4)
end
fig = Figure(backgroundcolor= :black, resolution = (1600,900))
nested_boxes_cbar!(fig)

```



Task 1:
Your code goes here!



```

fig = Figure(backgroundcolor= :black, resolution = (1600,900))
ga = fig[1, 1] = GridLayout()
gb = fig[1, 2] = GridLayout()
gc = fig[1, 3] = GridLayout()
gd = fig[2, 1:3] = GridLayout()
gA = Box(ga[1, 1], color = :olive)
nested_boxes_cbar!(gb)
nested_boxes_cbar!(gc)
Box(gd[1,1], color = rand(RGBAf))
Box(gd[2,1], color = rand(RGBAf))
Box(gd[3,1], color = rand(RGBAf))
colgap!(gb, 5)
colgap!(gc, 5)
rowgap!(gc, 5)
rowgap!(gd, 5)
rowgap!(fig.layout, 5)
colgap!(fig.layout, 5)
rowsize!(fig.layout, 2, Auto(0.5))
colsize!(fig.layout, 1, Auto(0.5))
fig

```

