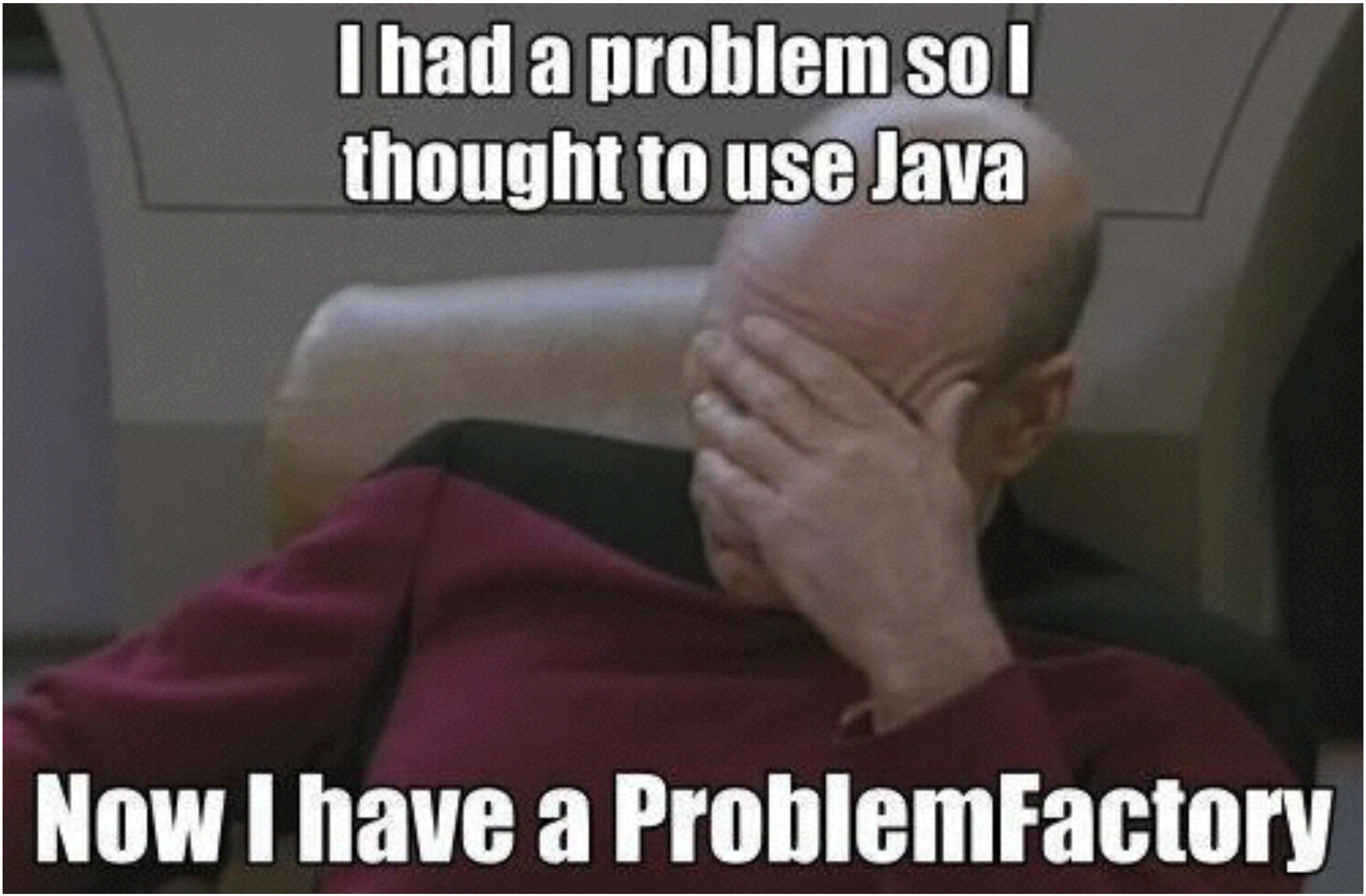
ONIONs

Organic Networks of Interconnected Object Nodes

ONIONNs

- A variation on the “Factory/Pattern” system

- But in the spirit of True Tcl, we implement a factory of factories


I had a problem so I thought to use Java
Now I have a ProblemFactory

ONIONs

Organic Networks of Interconnected Object Nodes

Mega Widgets in TaoTk

In this Presentation…

- We will implement a scenario visualizer for the 2014 Hackathon
- We'll do it in taotk
- And if you missed my paper from last year it's on:

http://fossil.etoyoc.com/fossil/taolib/doc/trunk/doc/taotk.pdf

Or…

- mkdir -p ~/repos
- cp [path to fossil repostitories on the USB Stick]/ taolib.fos ~/repos/taolib.fos
- mkdir -p ~/sandbox/taolib
- cd ~/sandbox/taolib
- fossil open ~/repos/taolib.fos
- ~/sandbox/taolib/docs/taotk.pdf

This program will…

- Create a master controller object for the application
- Create a canvas megawidget to display the data from the output log
- Open the output log and index the data for random access
- Allow the user to step through the run

In the Beginning…

```
# GameViewer.tcl
# [insert initialization code]

mysupercanvas create appmain
appmain signal startup
```

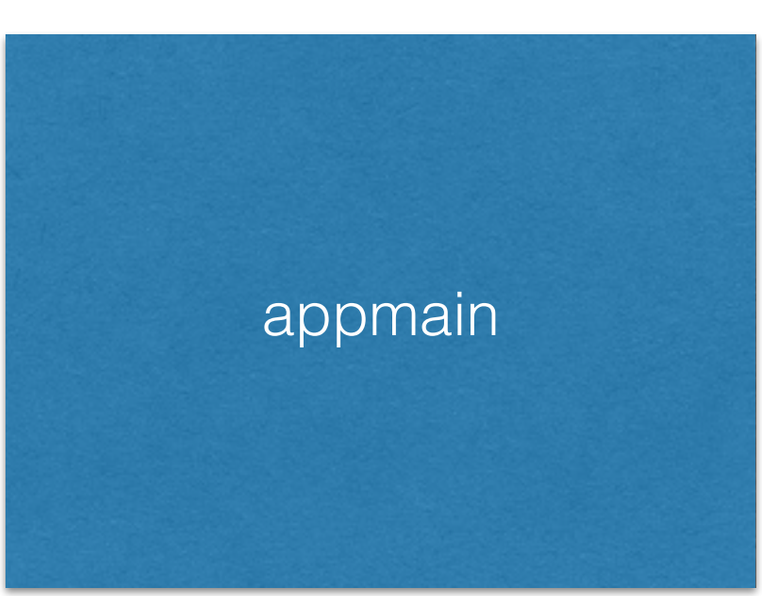
```
# main.tcl
package require mysupercanvas

mysupercanvas create appmain
appmain signal startup
```

after idle [namespace code {my Signal_Pipeline}]

```
tao::class ubercanvas {
   signal startup {
      triggers {layers components}
   }
}
```

```
tao::class ubercanvas {
   signal layers {
      follows {components}
      triggers {content}
      action {my activate_layers}
   }
}
```



```
tao::class ubercanvas {
   signal components {
      triggers {content}
      action {my widget components}
   }
}
```

appmain

```
tao::class ubercanvas {
   signal content {
    follows {layers components}
    action {
my widget content
}
   }
}
```

Todo:

1. my widget components
2. my activate_layers
3. my widget content
4. …?
5. Profit!

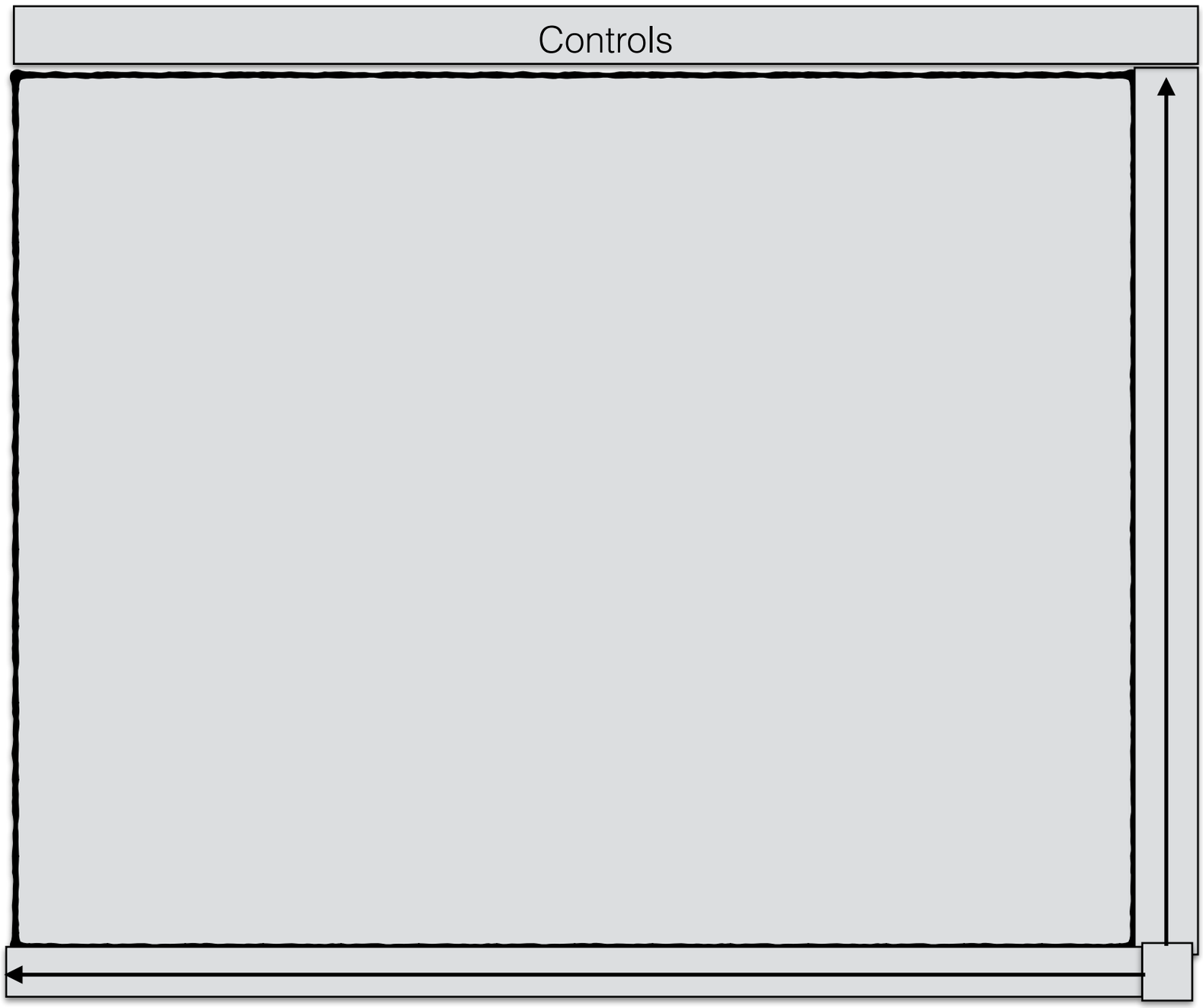
Todo:

1. my widget components
2. my activate_layers
3. my widget content

Todo:

1. my widget components
2. my activate_layers
3. my widget content

appmain



appmain

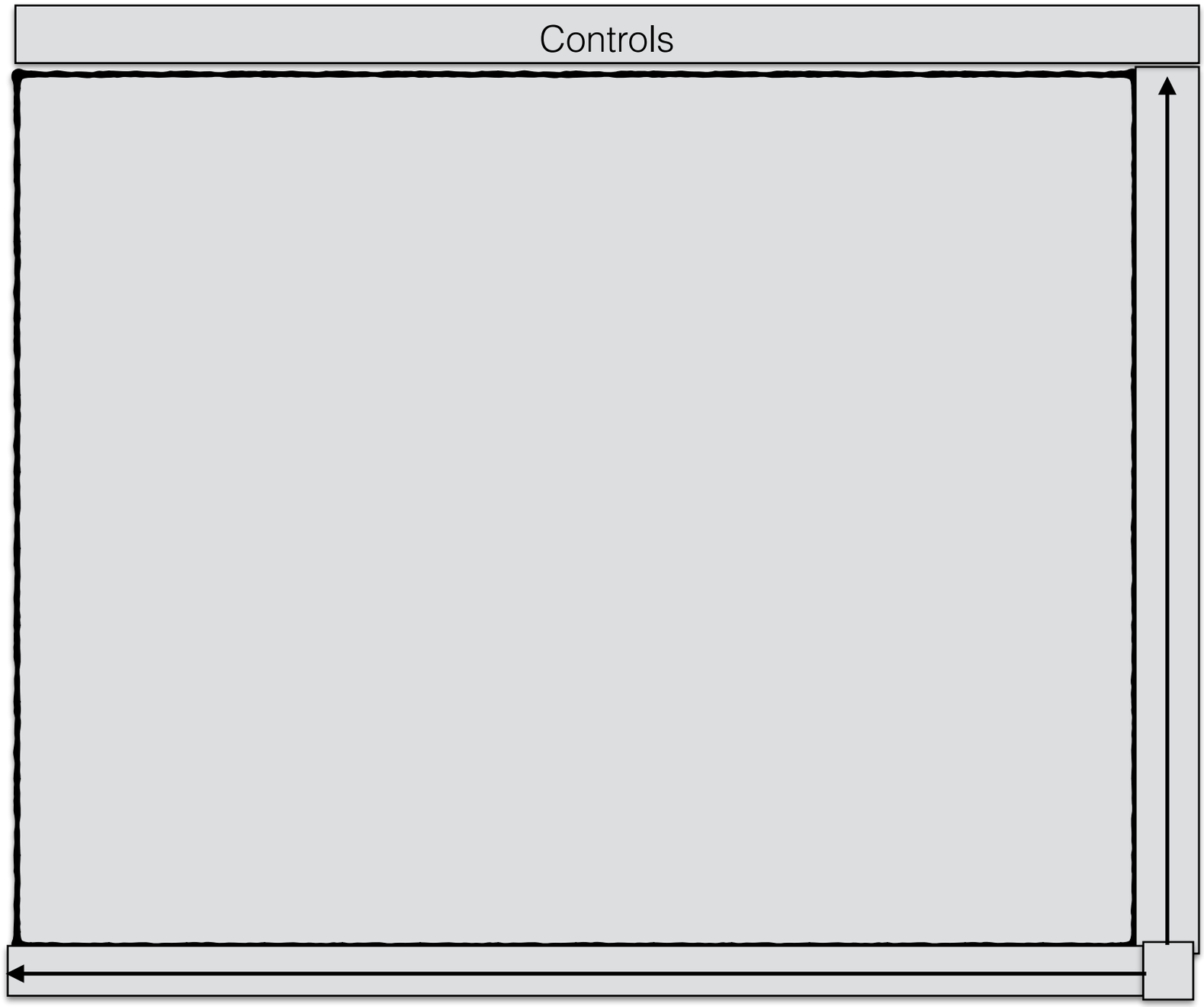
layer.map

layer.player#1

layer.player#2

Todo:

1. my widget components
2. my activate_layers
3. my widget content



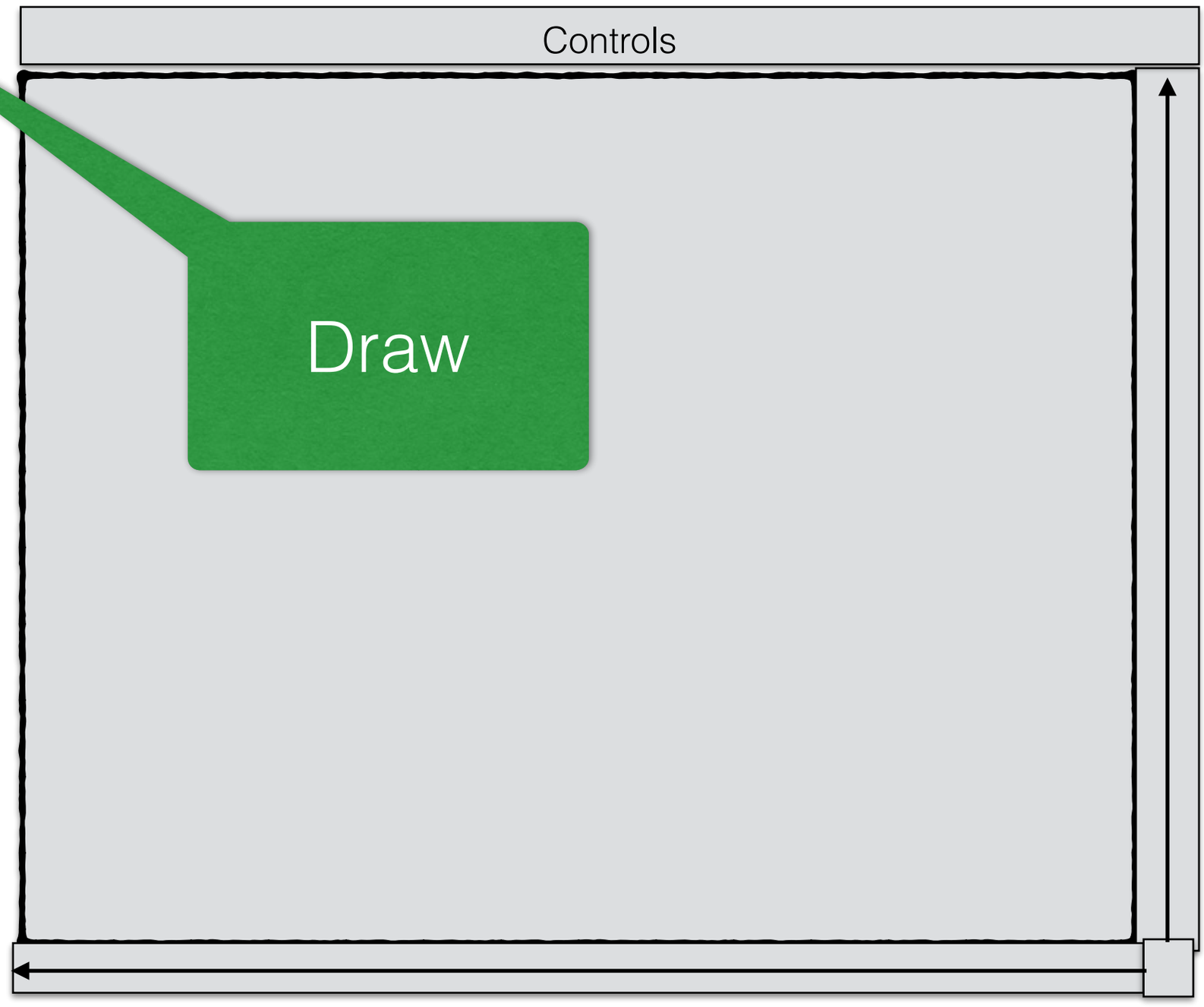
appmain

layer.map

layer.players

Todo:

1. my widget components
2. my activate_layers
3. my widget content



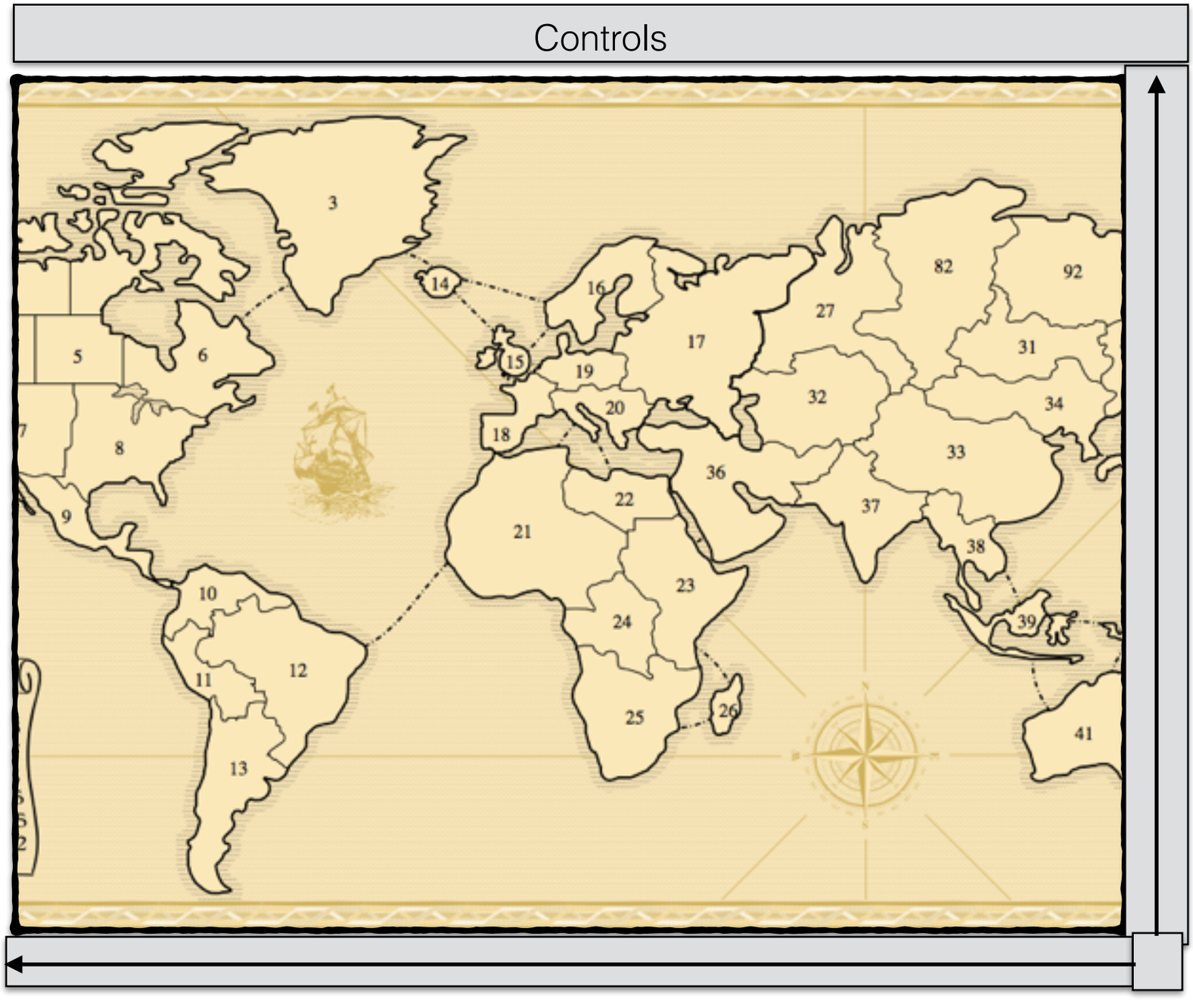
appmain

layer.map

layer.players

Todo:

1. my widget components
2. my activate_layers
3. my widget content



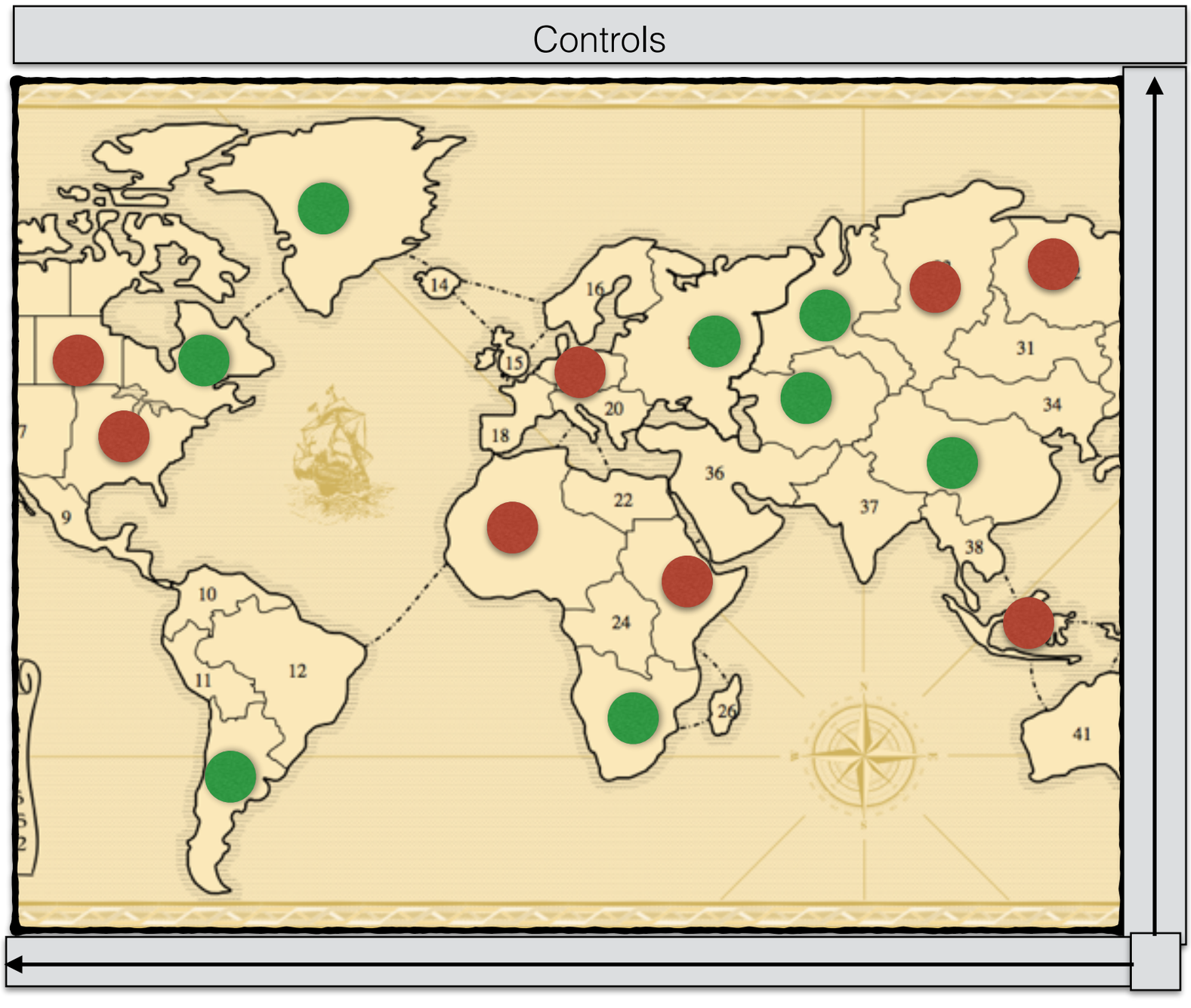
appmain

layer.map

layer.players

Todo:

1. my widget components
2. my activate_layers
3. my widget content

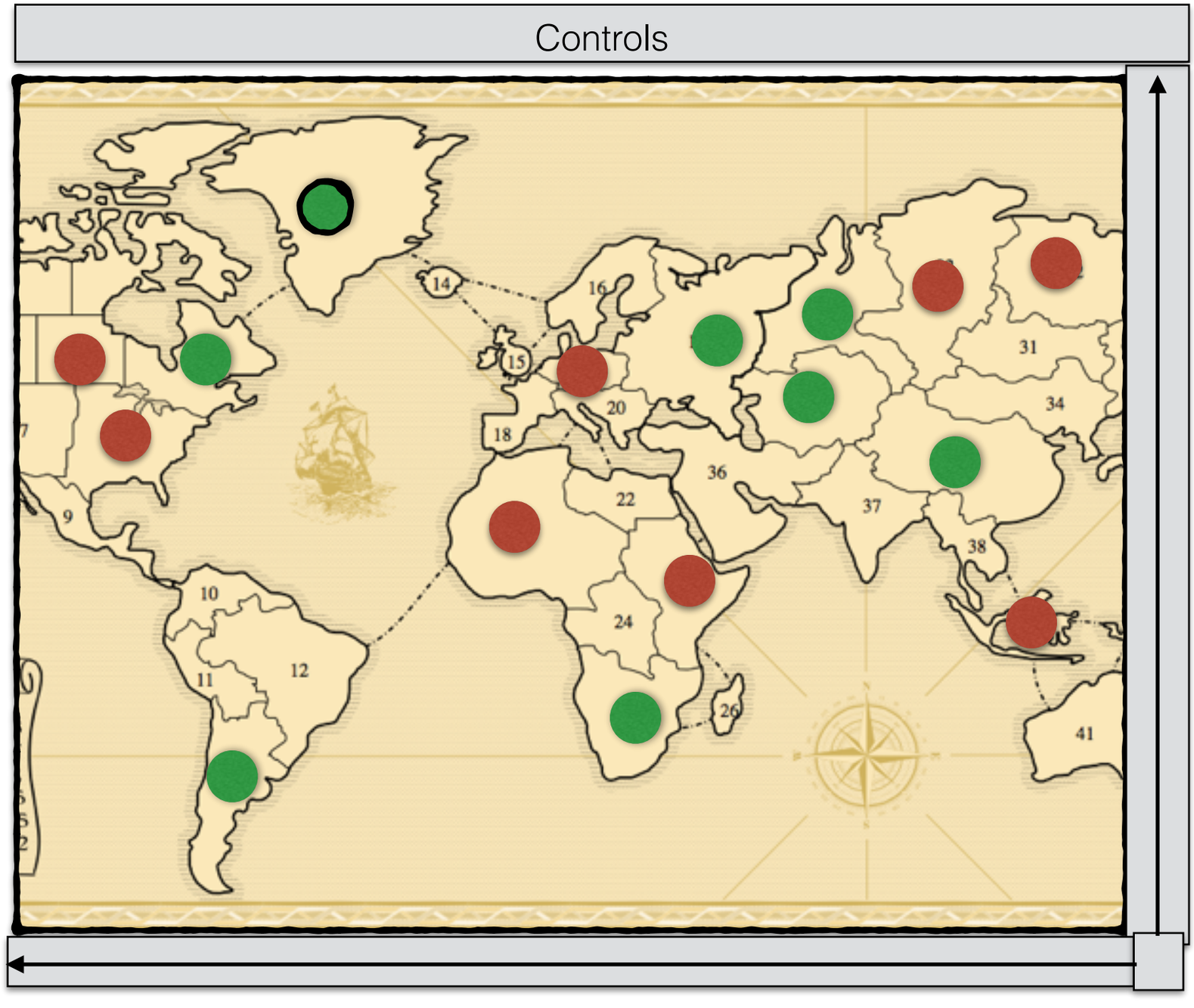


appmain

layer.map

layer.players

Event: User clicks on Dot



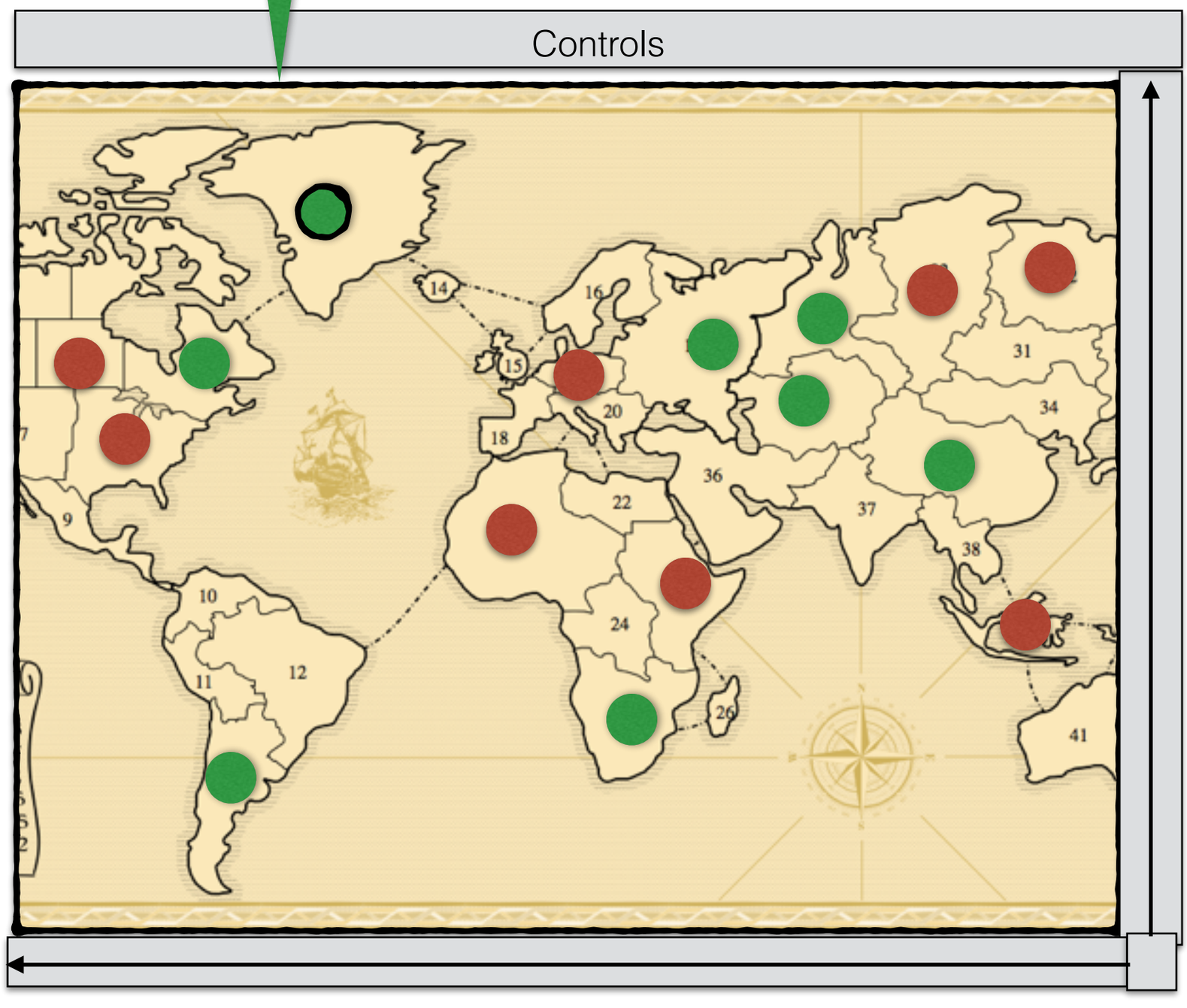
appmain

layer.map

layer.players

Event: User clicks on Dot

player#1.army#1 clicked

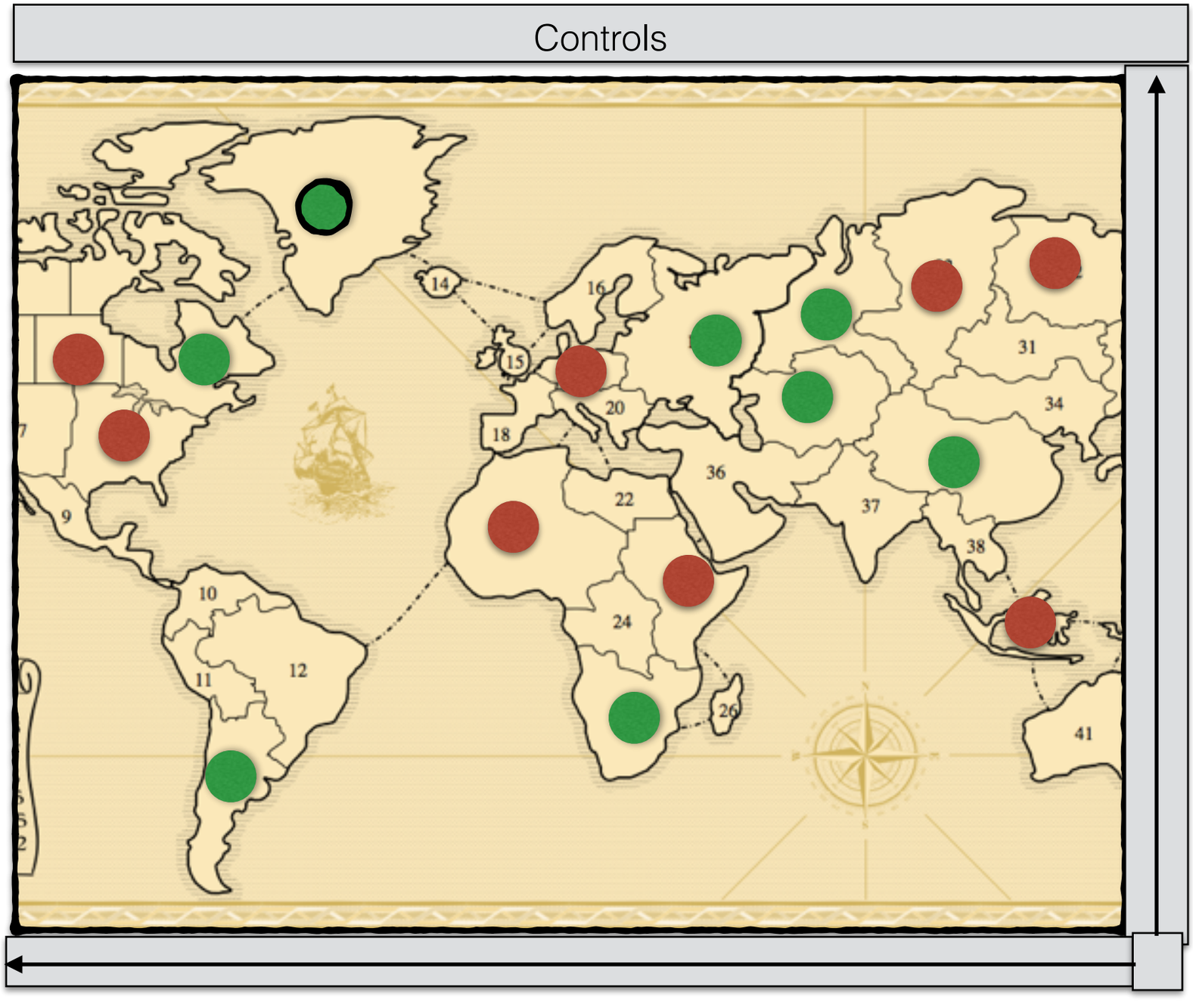


appmain

layer.map

layer.players

notify clicked player#1.army#1



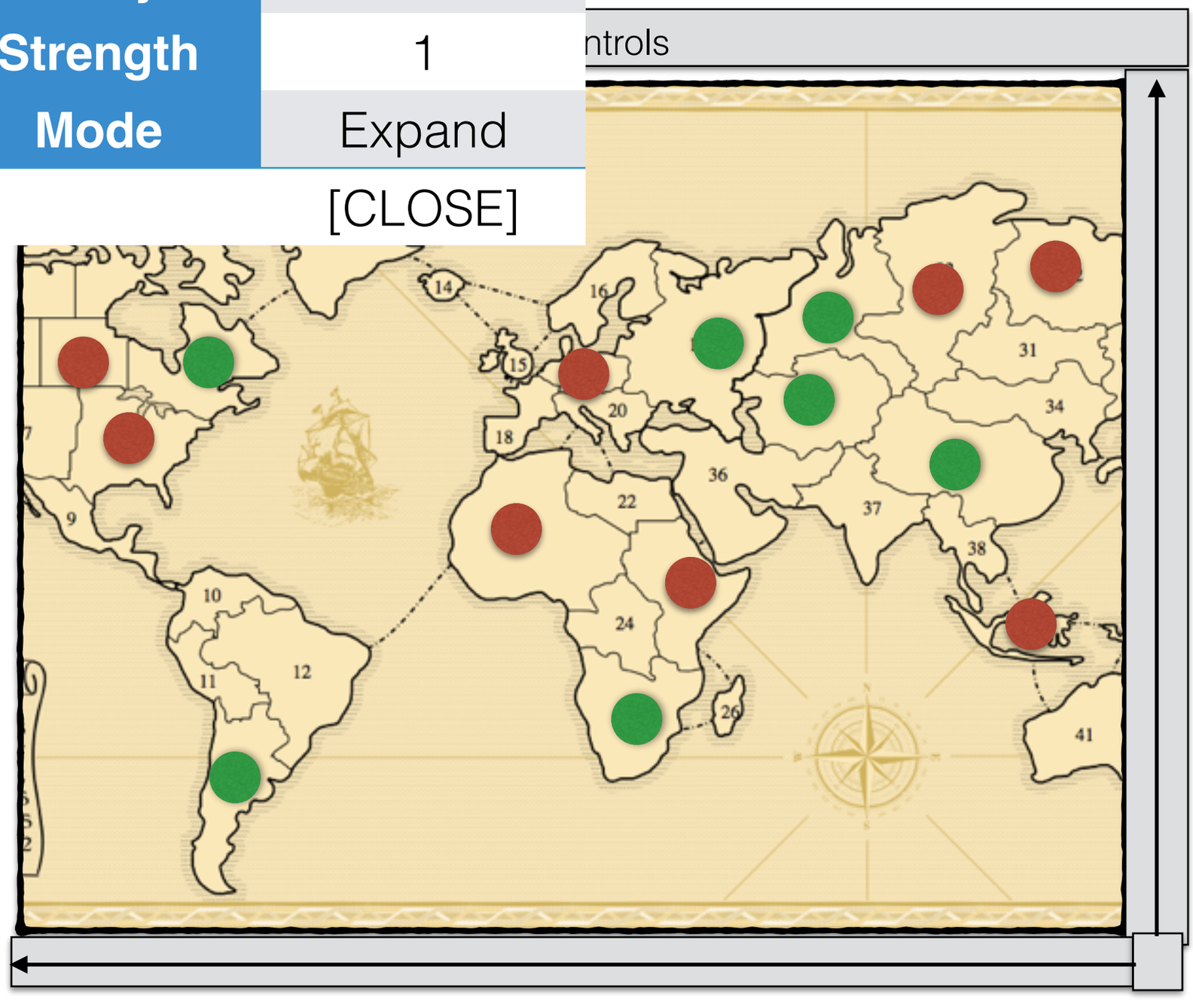
appmain

layer.map

layer.players

notify clicked player#1.army#1

Field	Value
Player	1
Army	1
Strength	1
Mode	Expand
	[CLOSE]

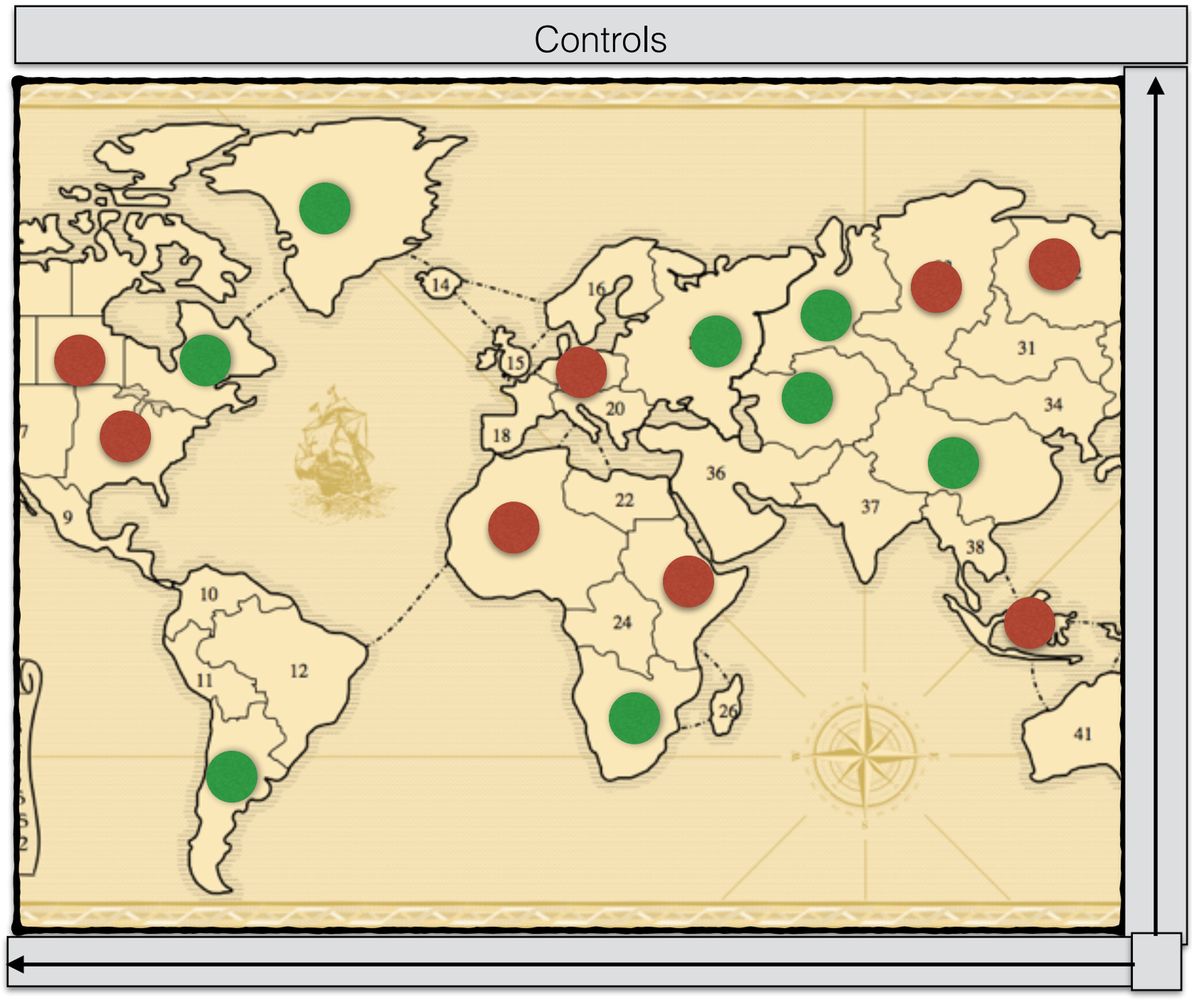


appmain

layer.map

layer.players

Event: Simulation Step Complete



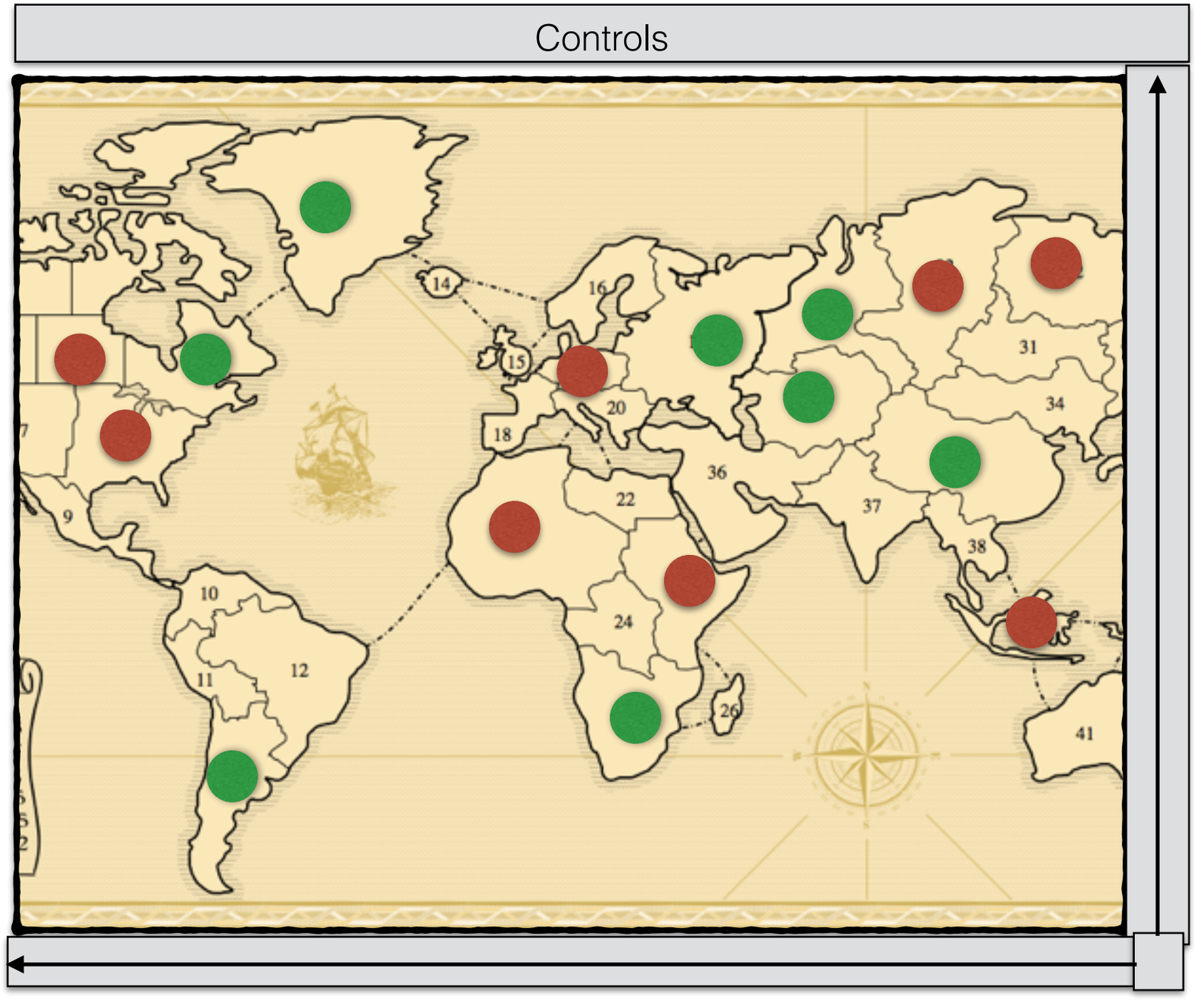
Event: Simulation Step Complete

appmain

read_one_log_step

layer.map !

layer.players !



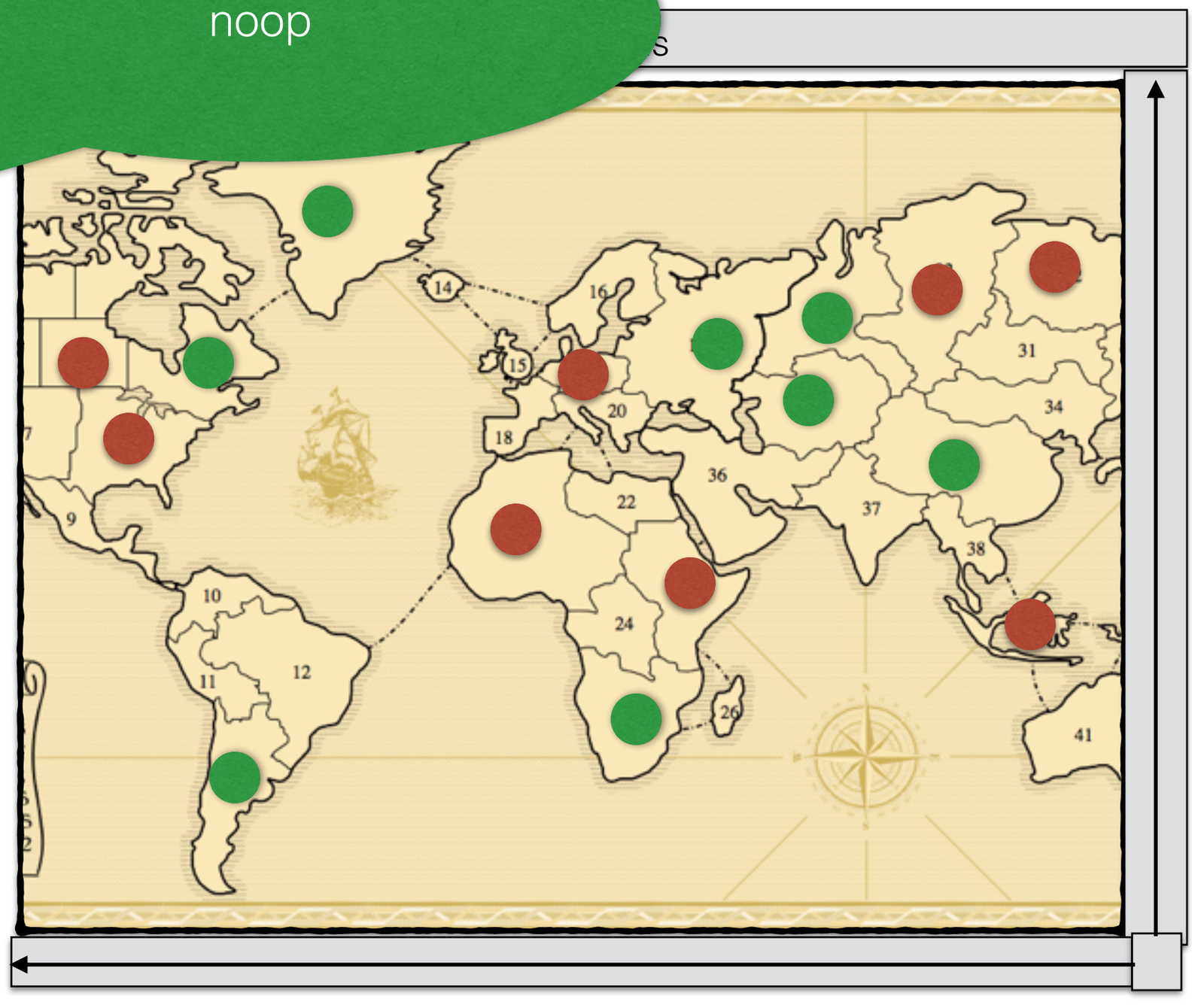
Event: Simulation Step Complete

appmain

layer.map !

layer.players !

noop



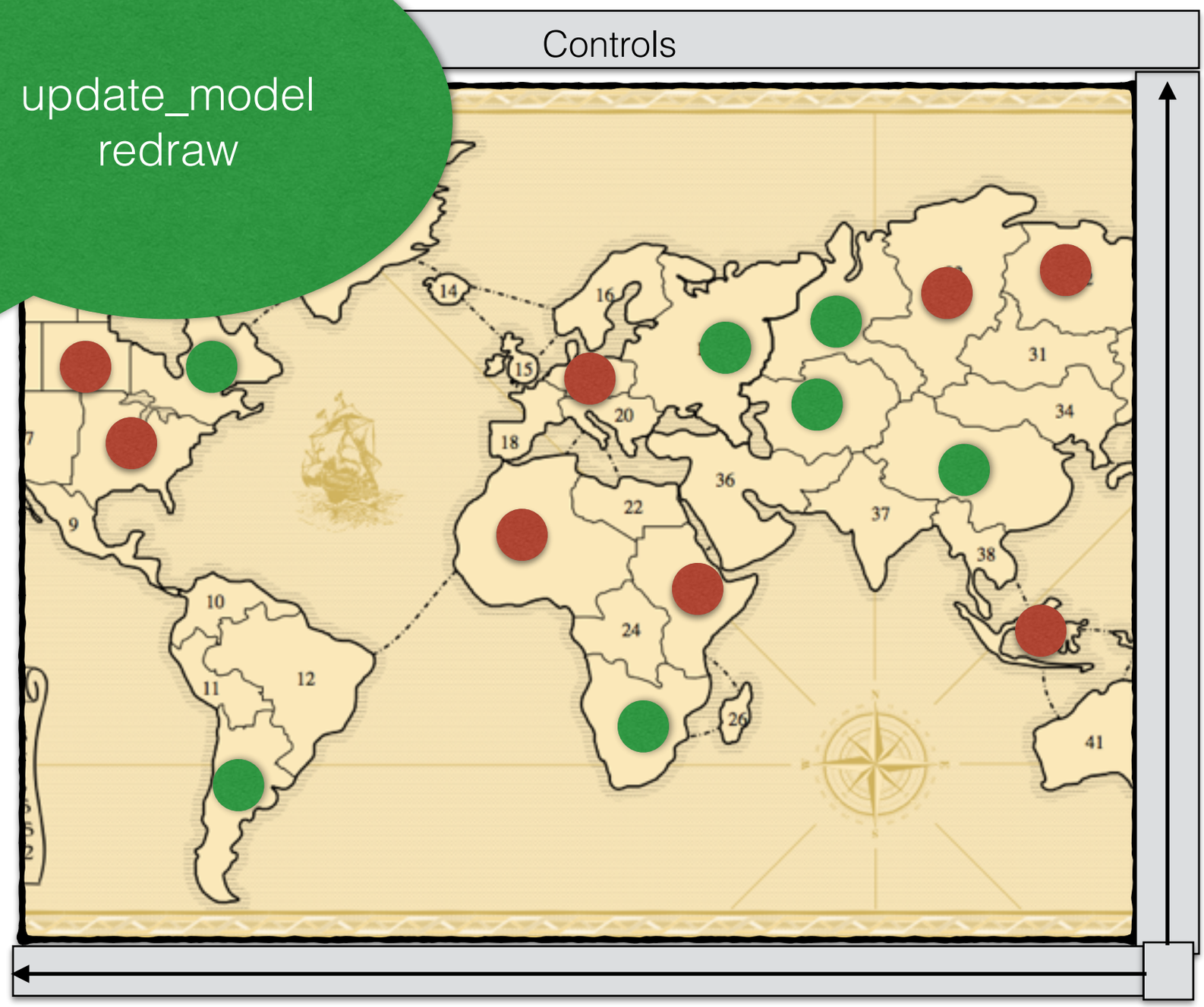
Event: Simulation Step Complete

appmain

layer.map

layer.players !

update_model
redraw



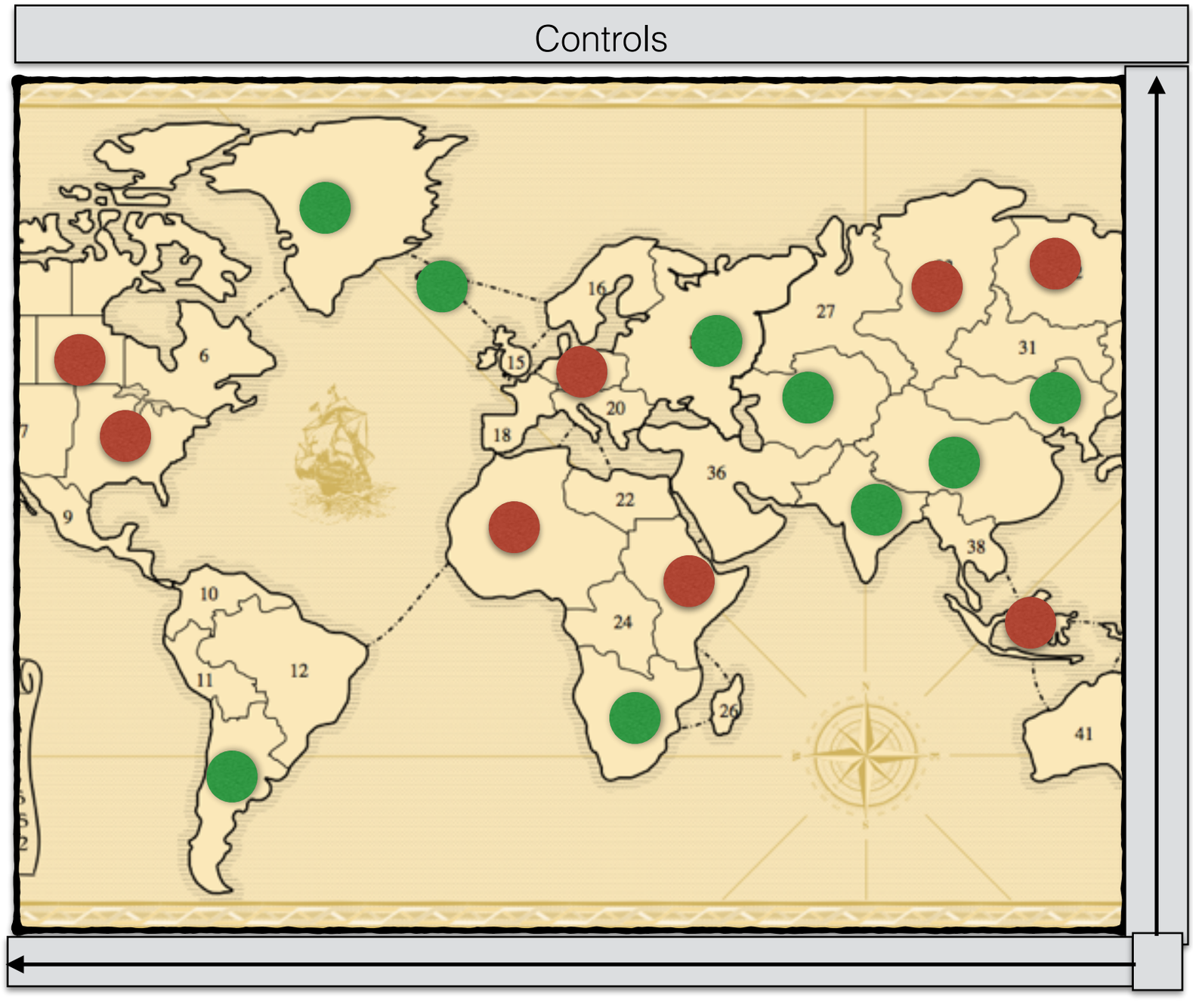
Event: Simulation Step Complete

appmain

read_one_log_step

layer.map

layer.players



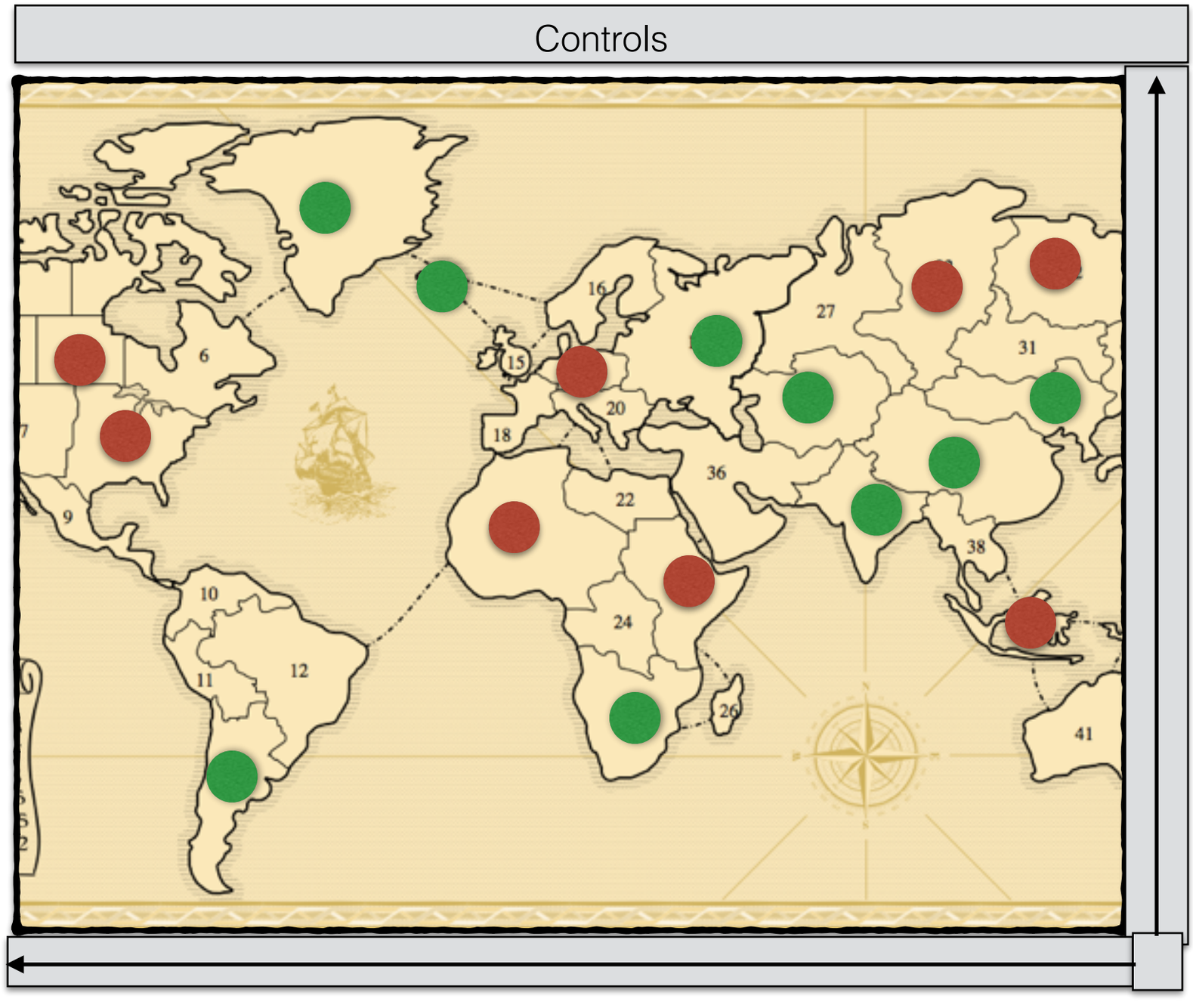
Event: Simulation Step Complete

appmain

read_one_log_step

layer.map

layer.players



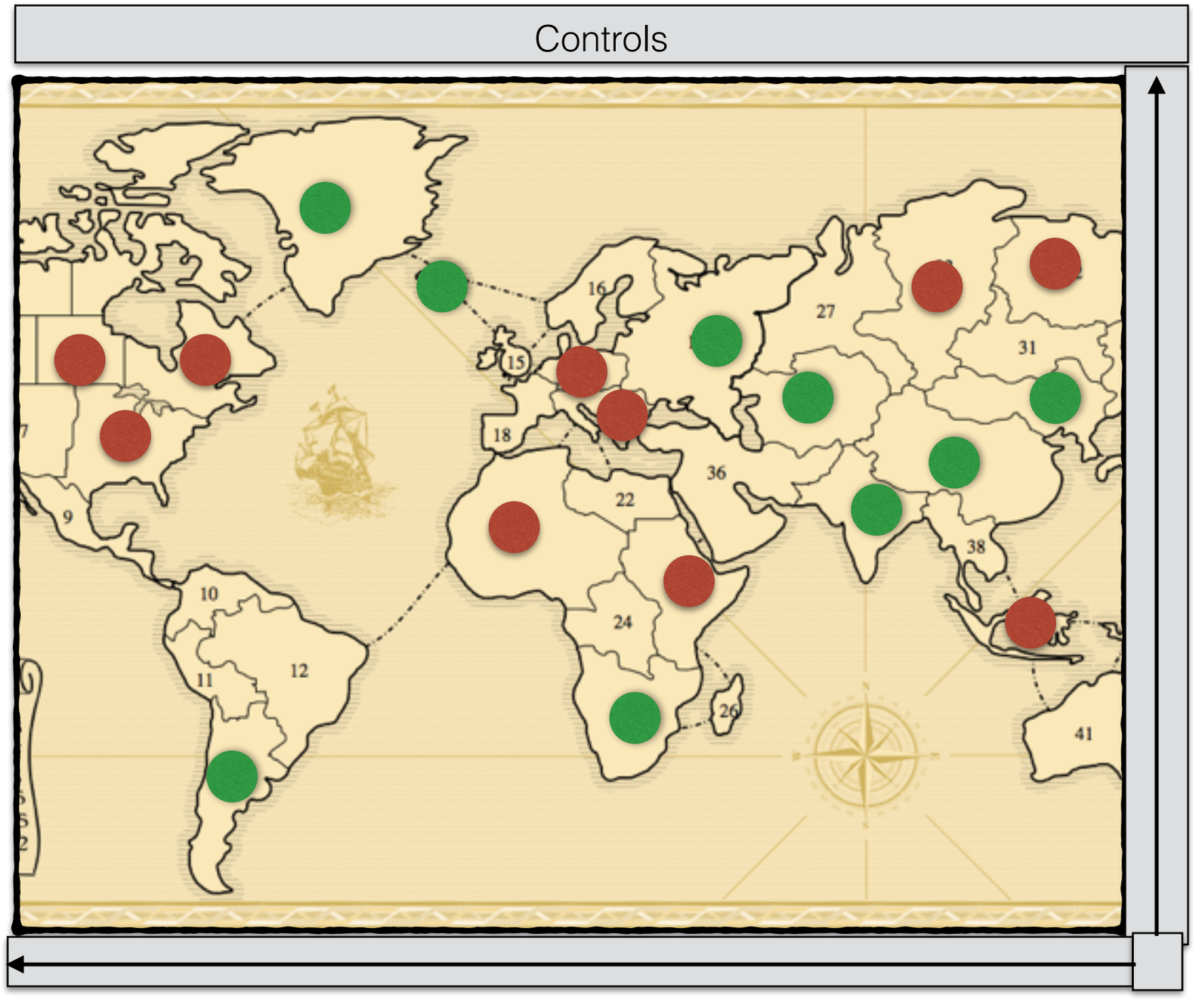
Event: Simulation Step Complete

appmain

read_one_log_step

layer.map

layer.players



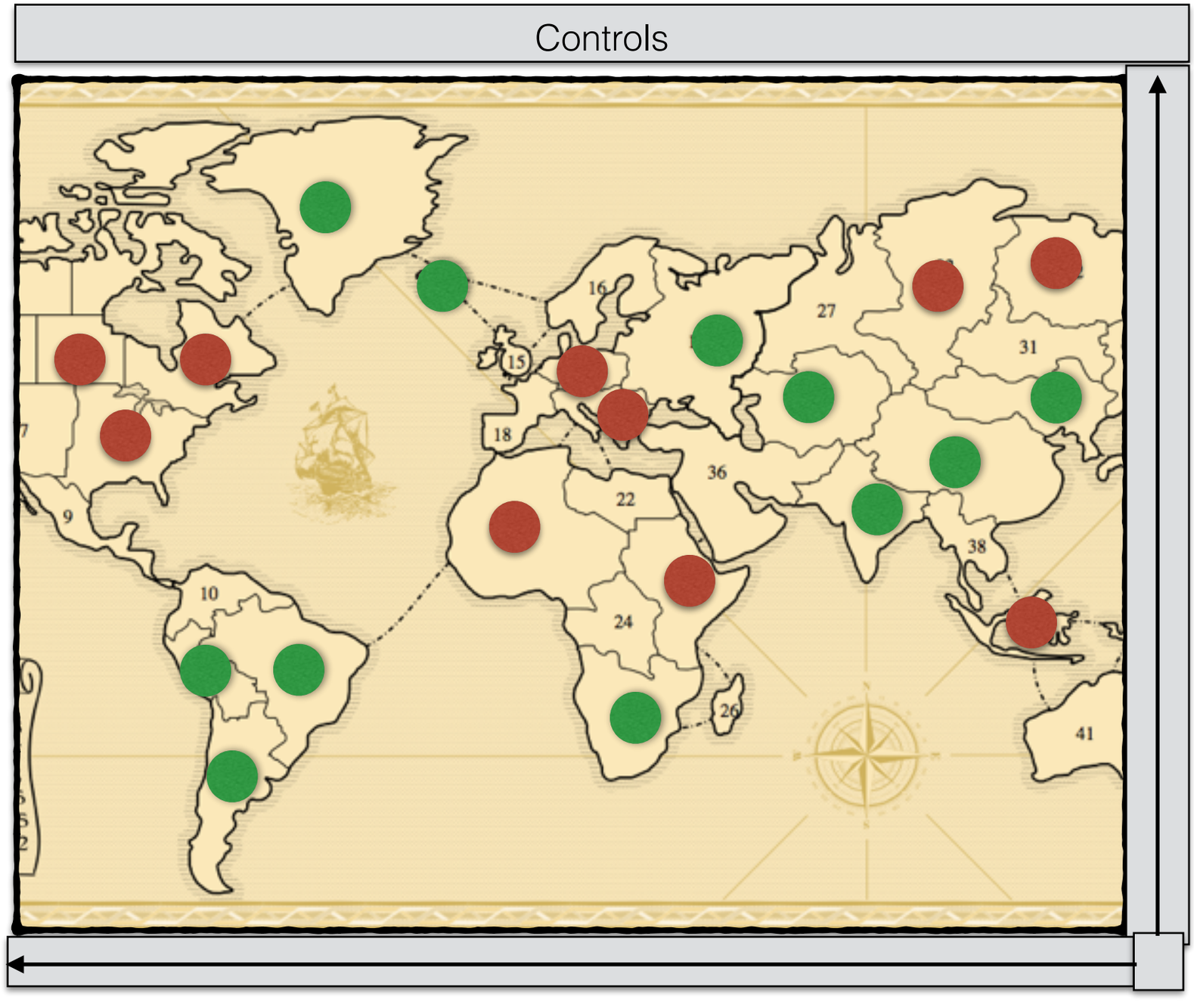
Event: Simulation Step Complete

appmain

read_one_log_step

layer.map

layer.players



Event: Simulation Step Complete

appmain

read_one_log_step

layer.map

layer.players

Done