Tell Your Graphics Stack That the Display Is Circular

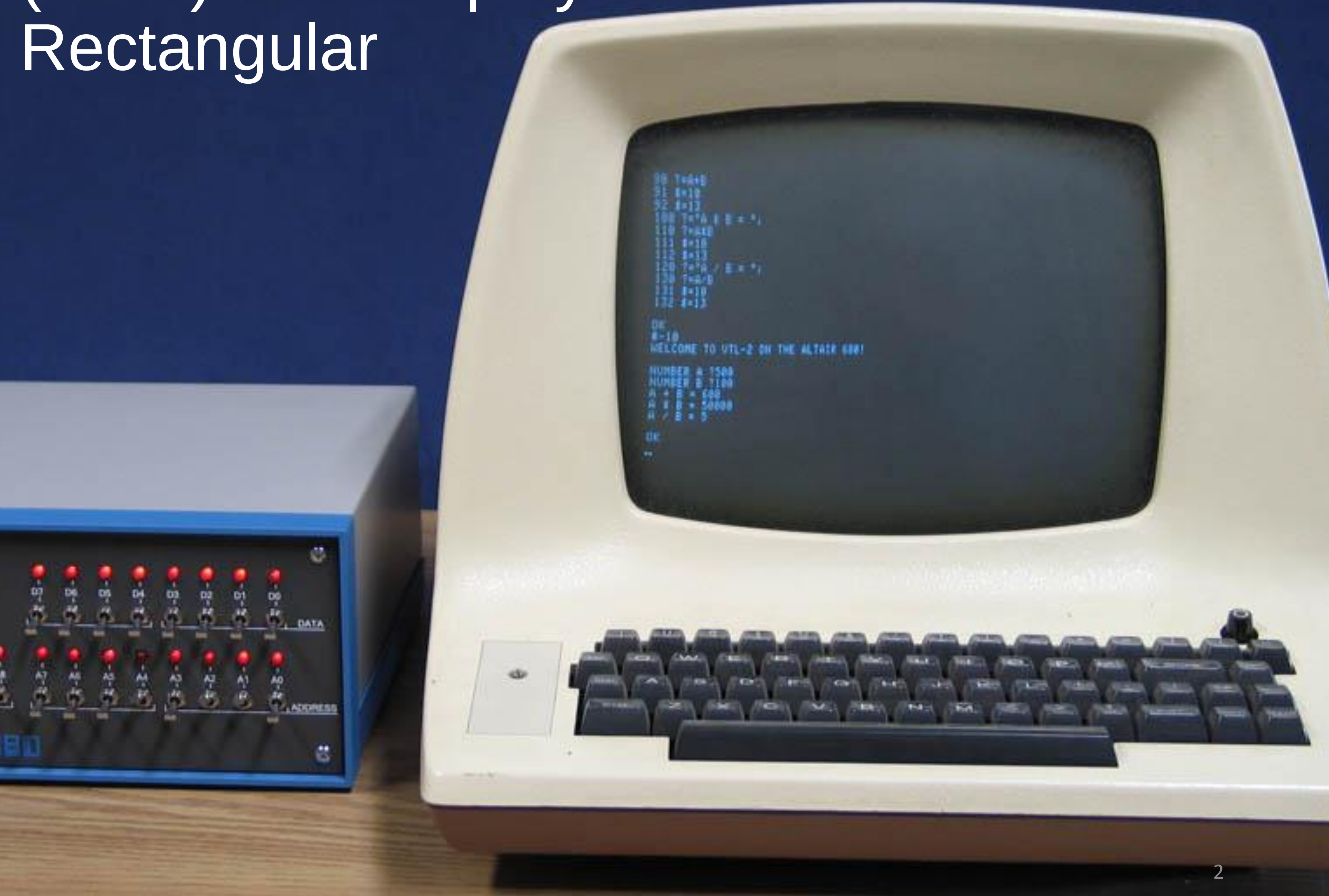
Hongyu Miao
miaoh@purdue.edu

Felix Xiaozhu Lin
xzl@purdue.edu

PURDUE UNIVERSITY

http://xsel.rocks

(1975) Text Display Rectangular

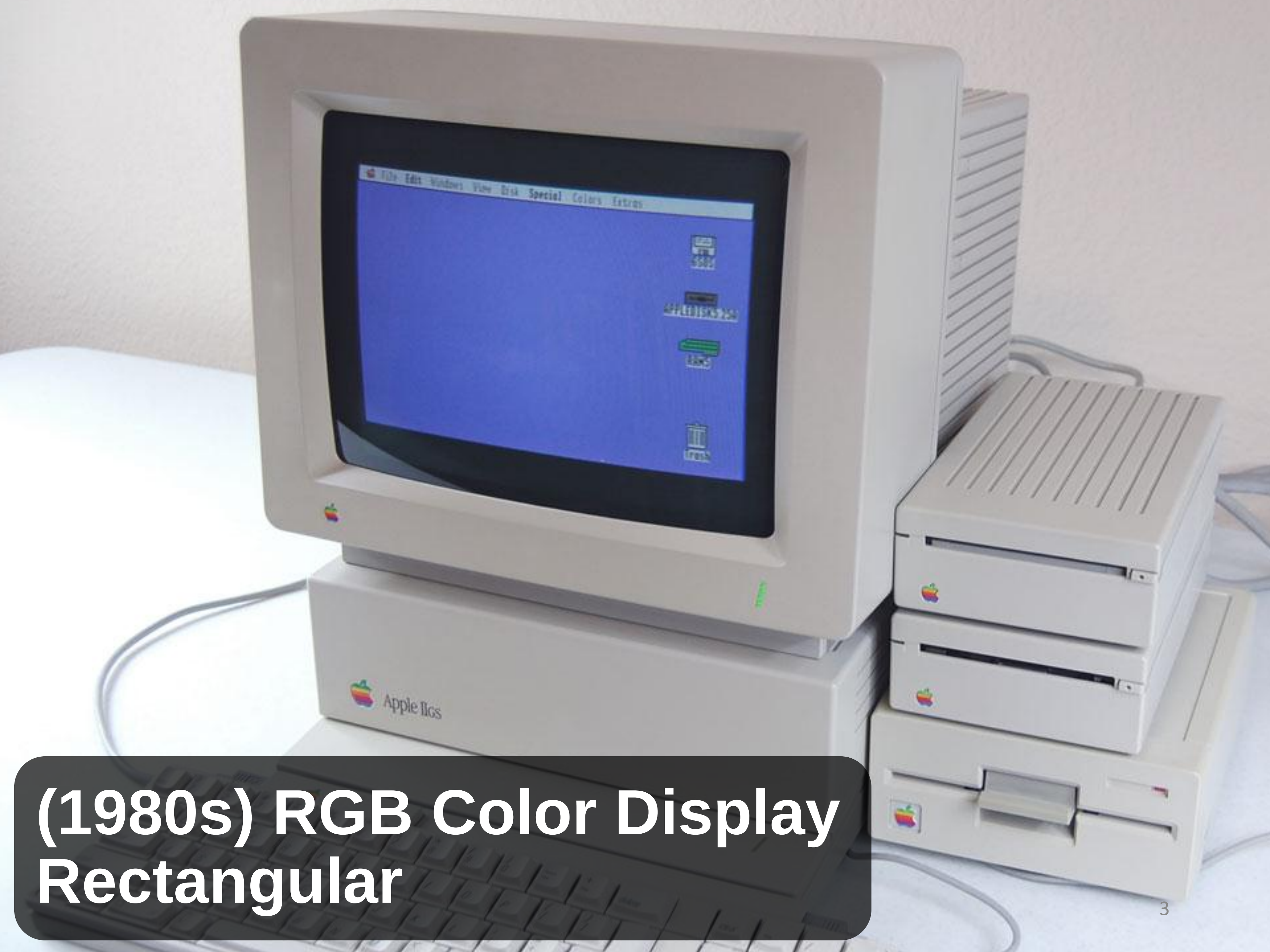


```
90 ?=A+B
91 $=10
92 $=13
100 ?="A * B = ";
110 ?=A*B
111 $=10
112 $=13
120 ?="A / B = ";
130 ?=A/B
131 $=10
132 $=13

OK
#=10
WELCOME TO VTL-2 ON THE ALTAIR 680!

NUMBER A ?500
NUMBER B ?100
A + B = 600
A * B = 50000
A / B = 5

OK
*
```



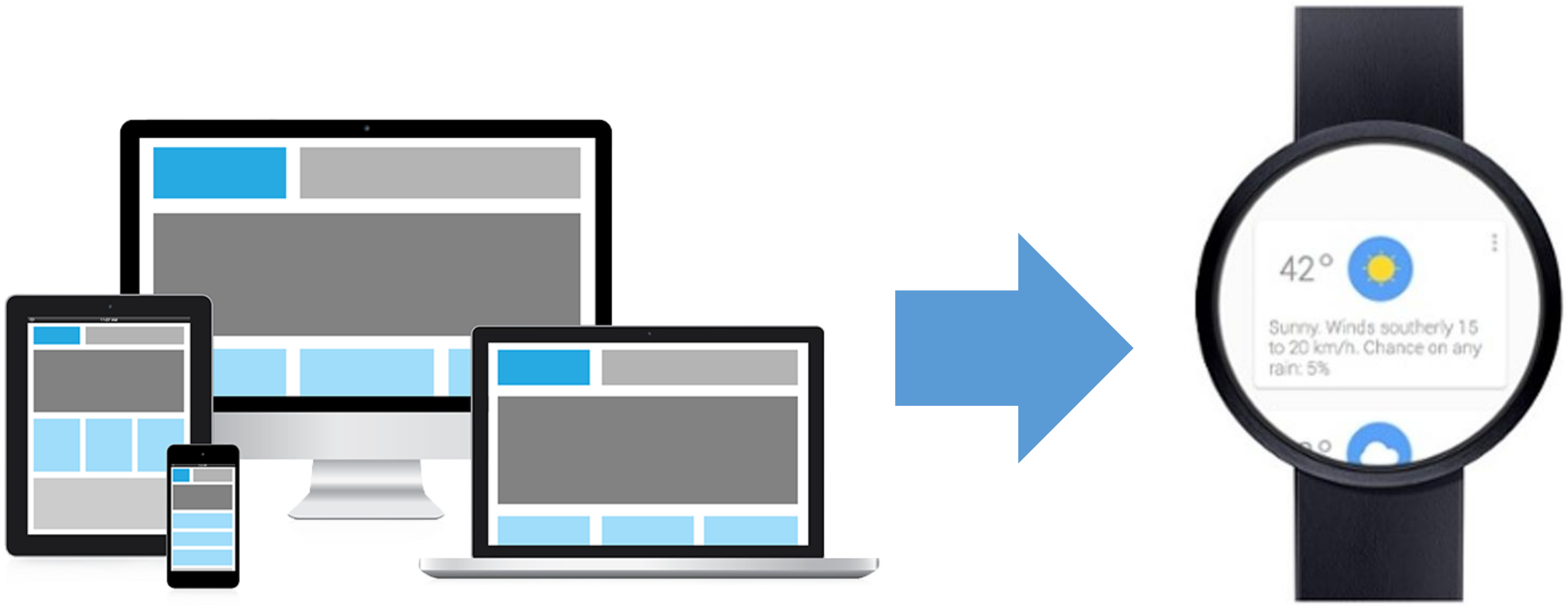
(1980s) RGB Color Display Rectangular

Modern Desktop: Rectangular

Smartphone Display: Rectangular

(2016) Wearable Displays
Circular!

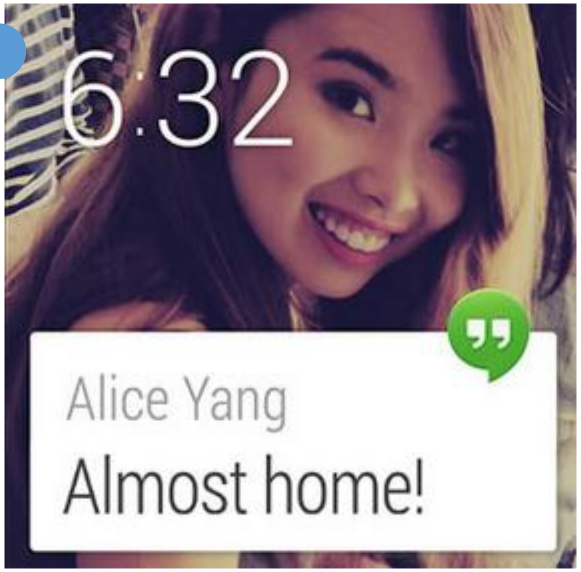
Display hardware: Not only rectangular any longer!


42°
Sunny. Winds southerly 15 to 20 km/h. Chance on any rain: 5%

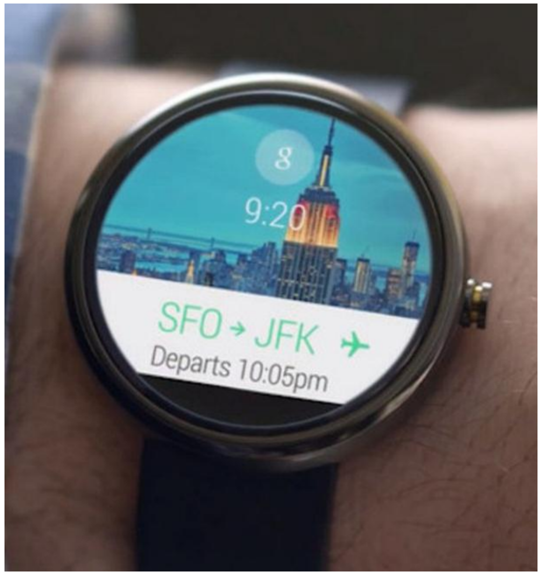
Software Challenges ?

What's the problem?

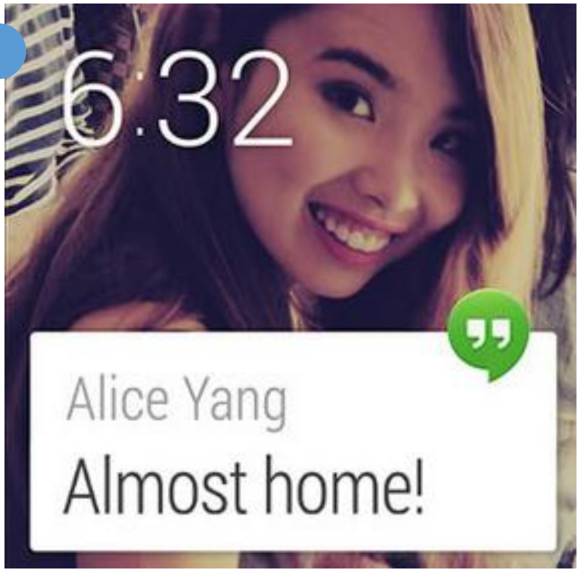
Graphics Stack:
Rectangular area



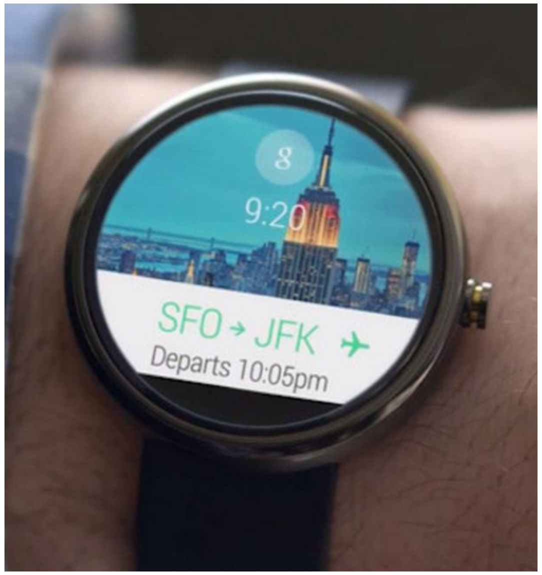
What’s the problem?



Graphics Stack:
Rectangular area

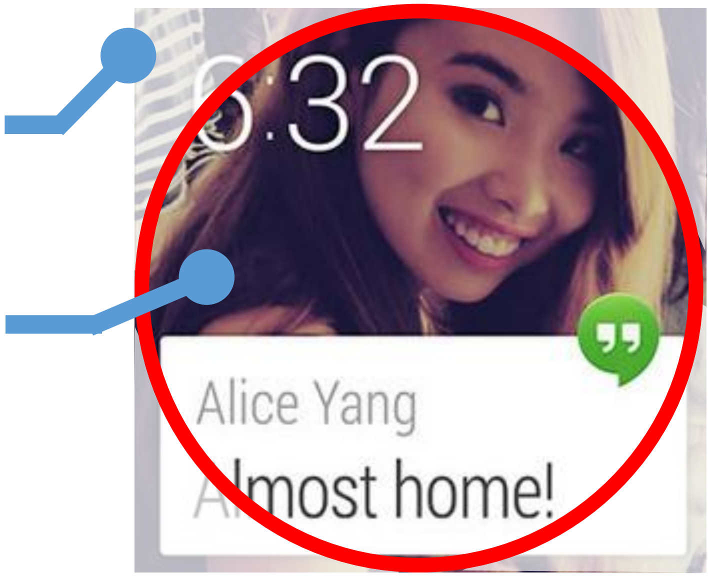


What's the problem?

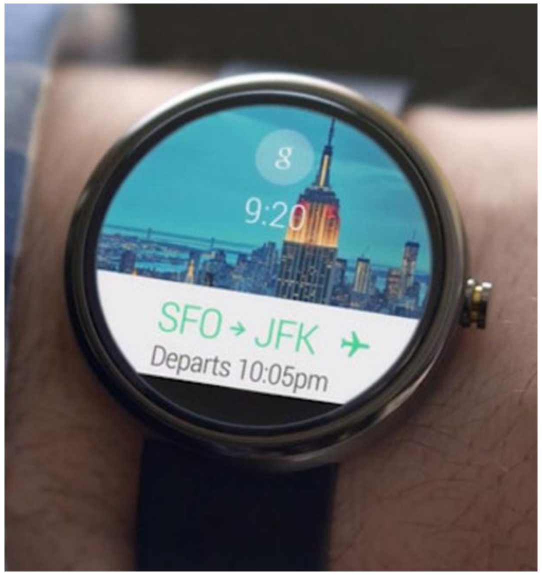


Graphics Stack:
Rectangular area

Displayed:
Circular area



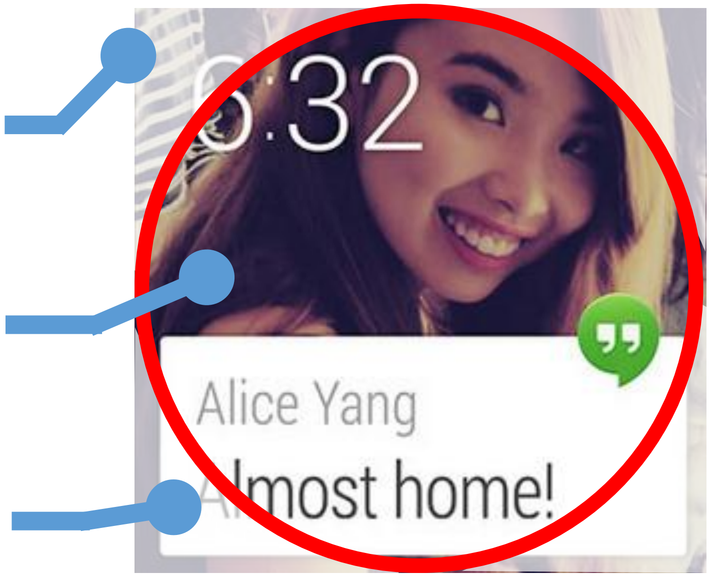
What’s the problem?



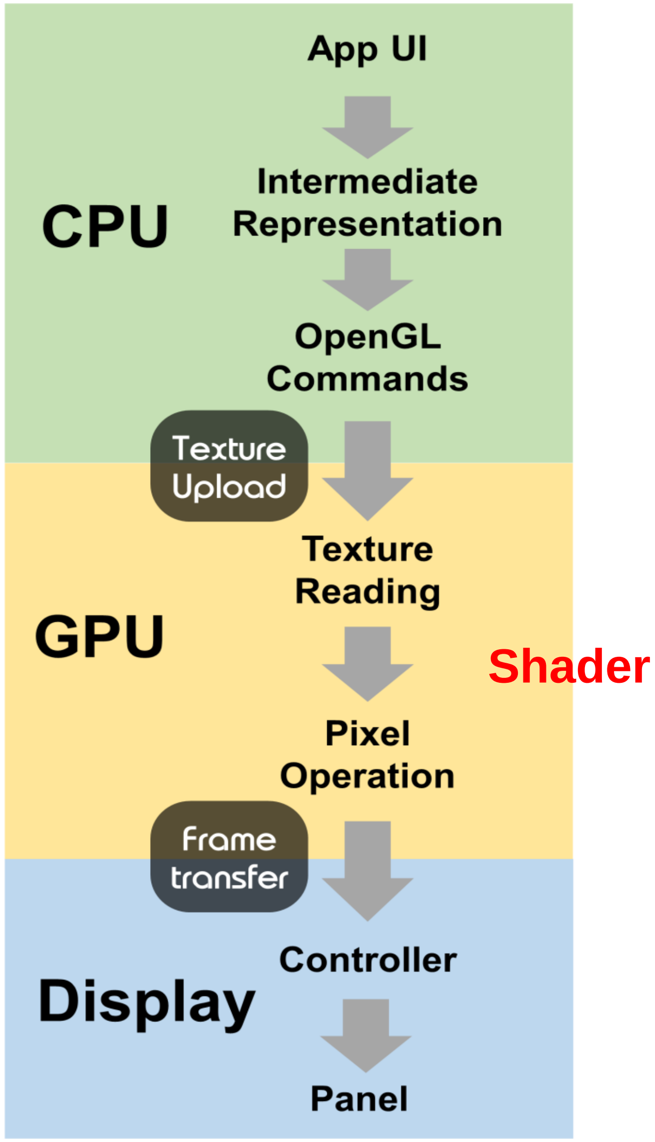
Graphics Stack:
Rectangular area

Displayed:
Circular area

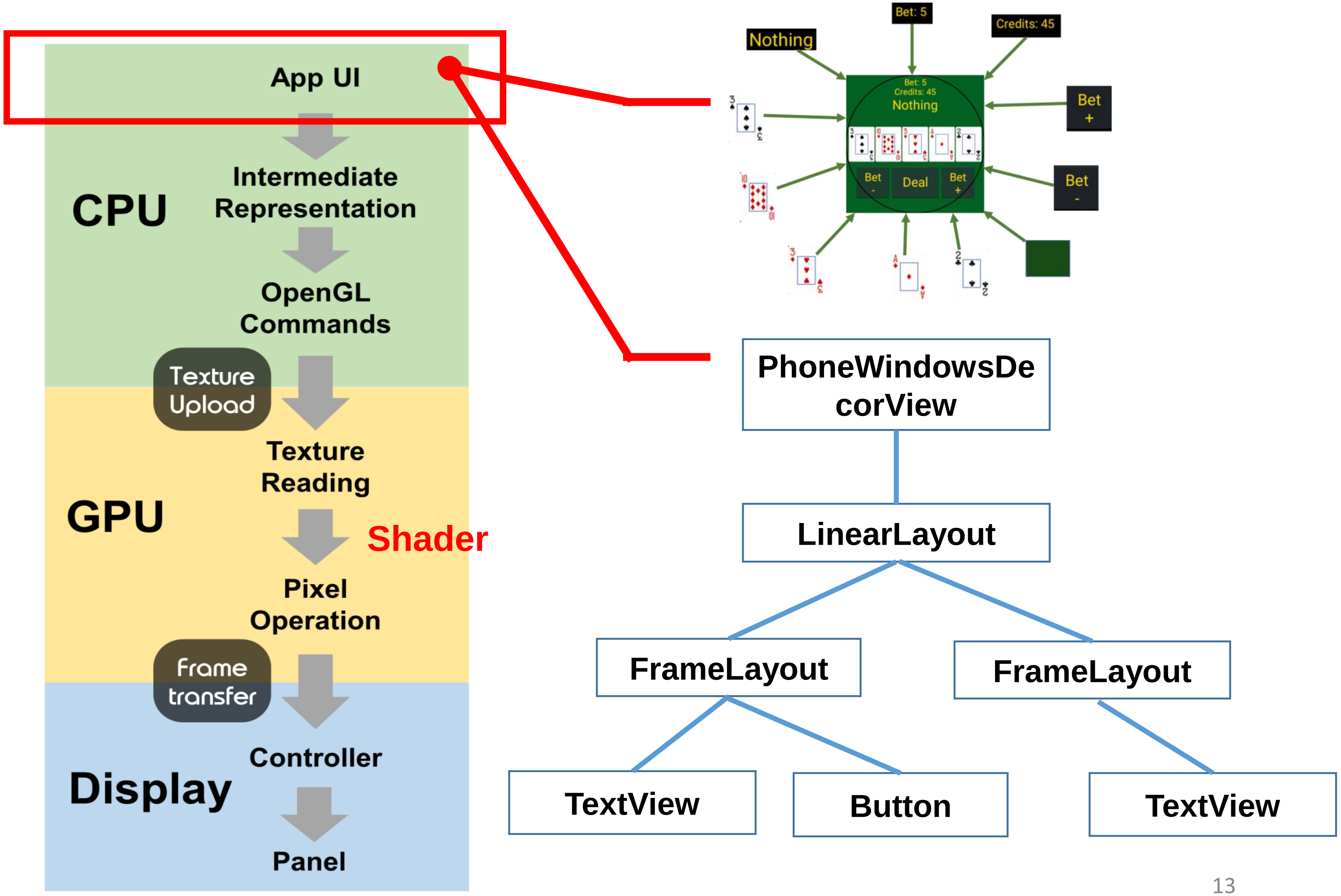
Invisible area:
WASTED!



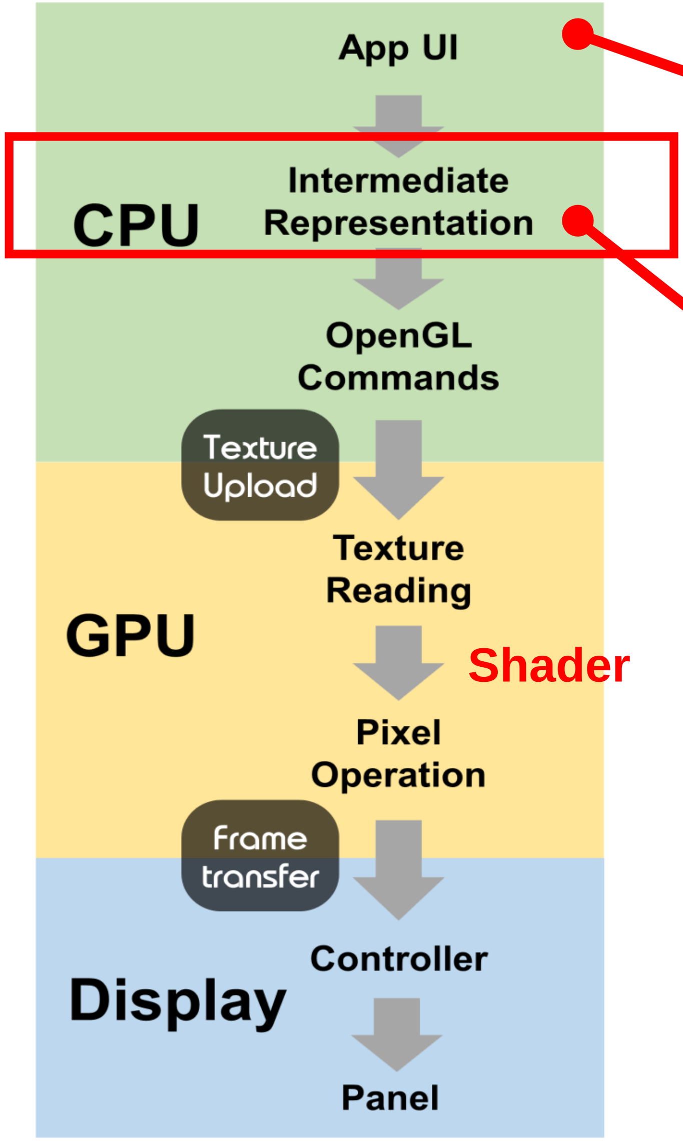
A 20,000 foot view of graphics stack

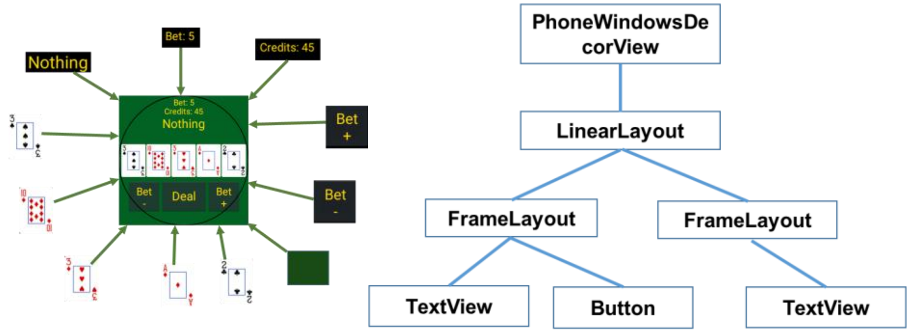


A 20,000 foot view of graphics stack



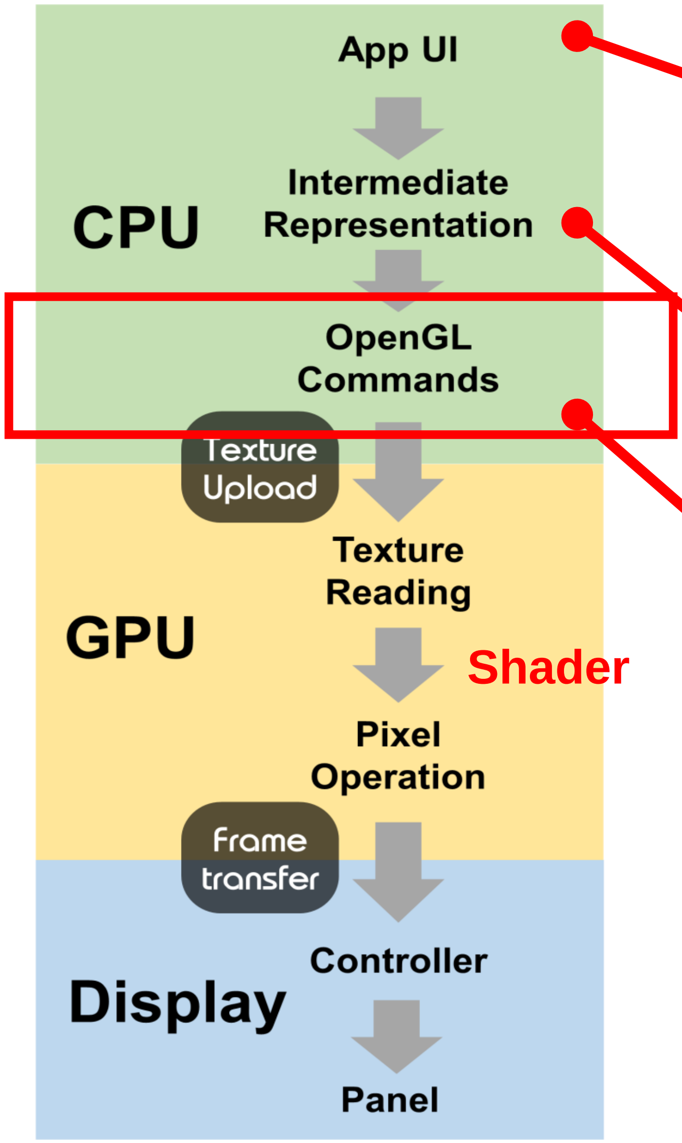
A 20,000 foot view of graphics stack

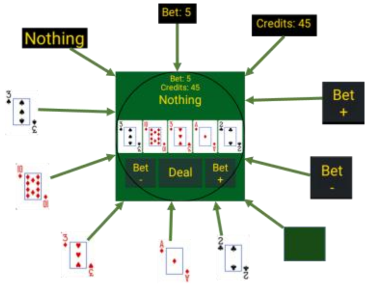


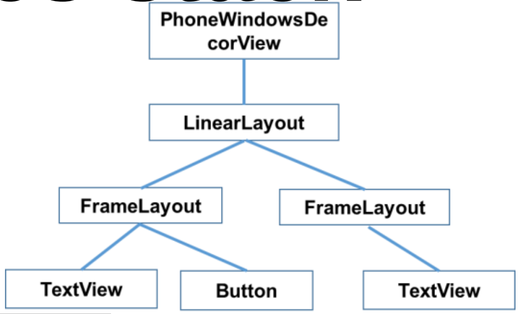


```
Display List
Save
DrawRenderNode
        DrawRenderNode
             DrawRect
```

A 20,000 foot view of graphics stack



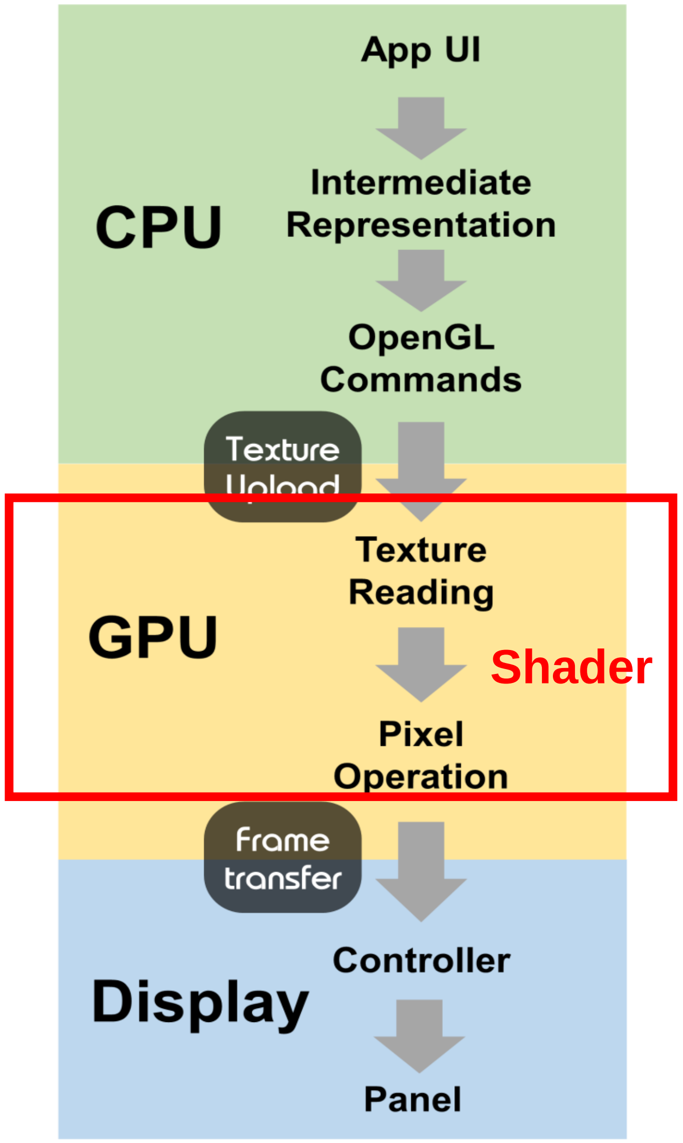




```
Display List
Save
DrawRenderNode
        DrawRenderNode
             DrawRect
```

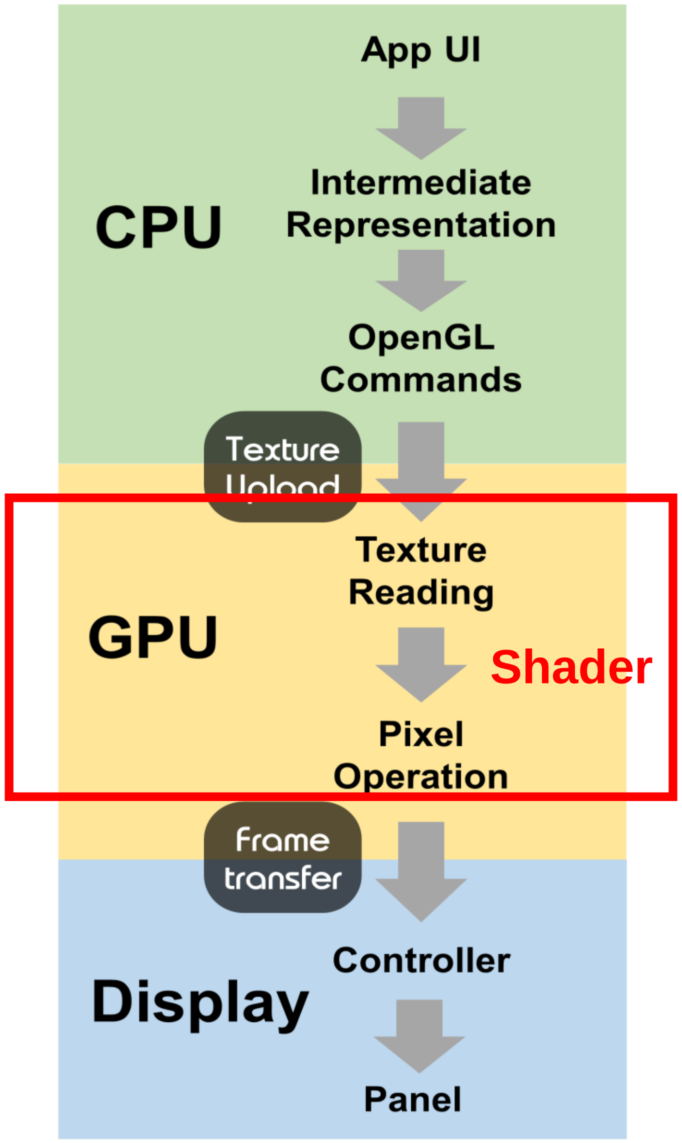
```
OpenGL Commands
glDisable(cap=GL_SCISSOR_TEST)
glActiveTexture(texture=GL_TEX
TURE0)
glGenBuffer(n=1,buffer=[2])
glBindBuffer(target=GL_ARRAY_B
UFFER,buffer=2)
```

A 20,000 foot view of graphics stack



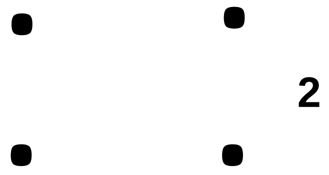
1. Load texture

A 20,000 foot view of graphics stack

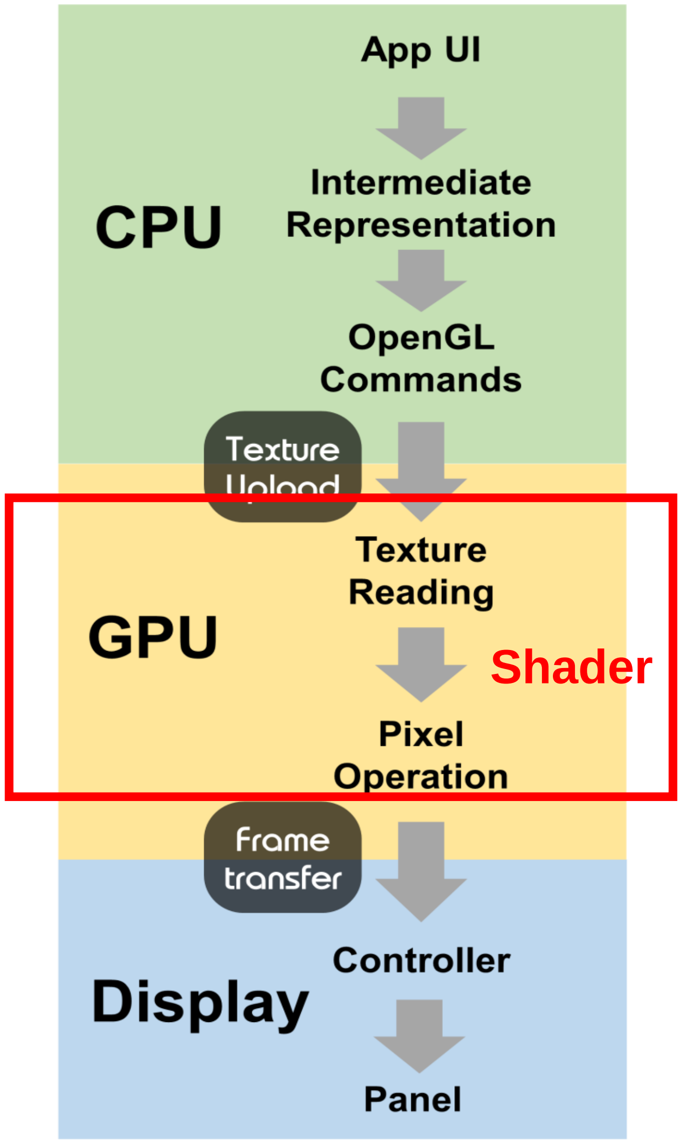


1. Load texture

2. Read vertex data



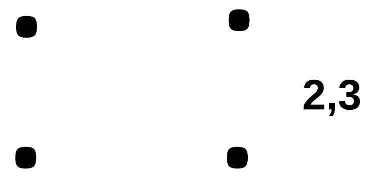
A 20,000 foot view of graphics stack



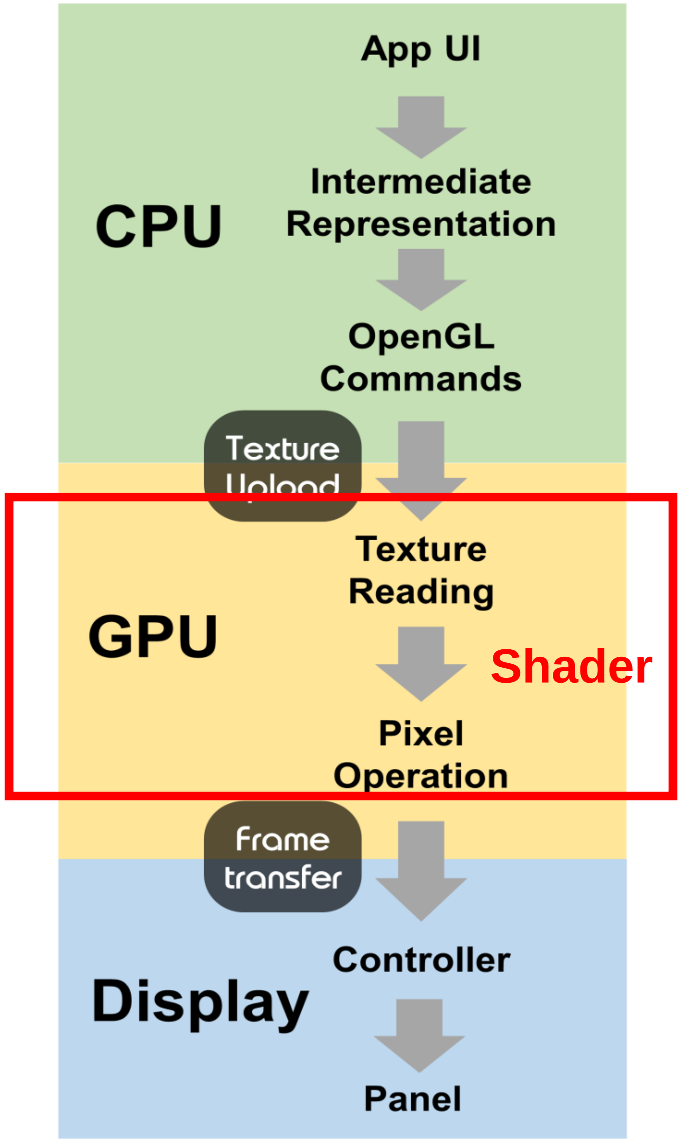
1. Load texture

2. Read vertex data

3. Execute vertex shader



A 20,000 foot view of graphics stack

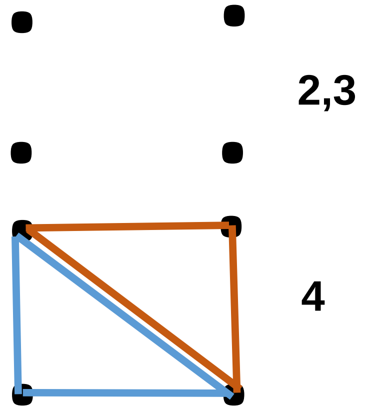


1. Load texture

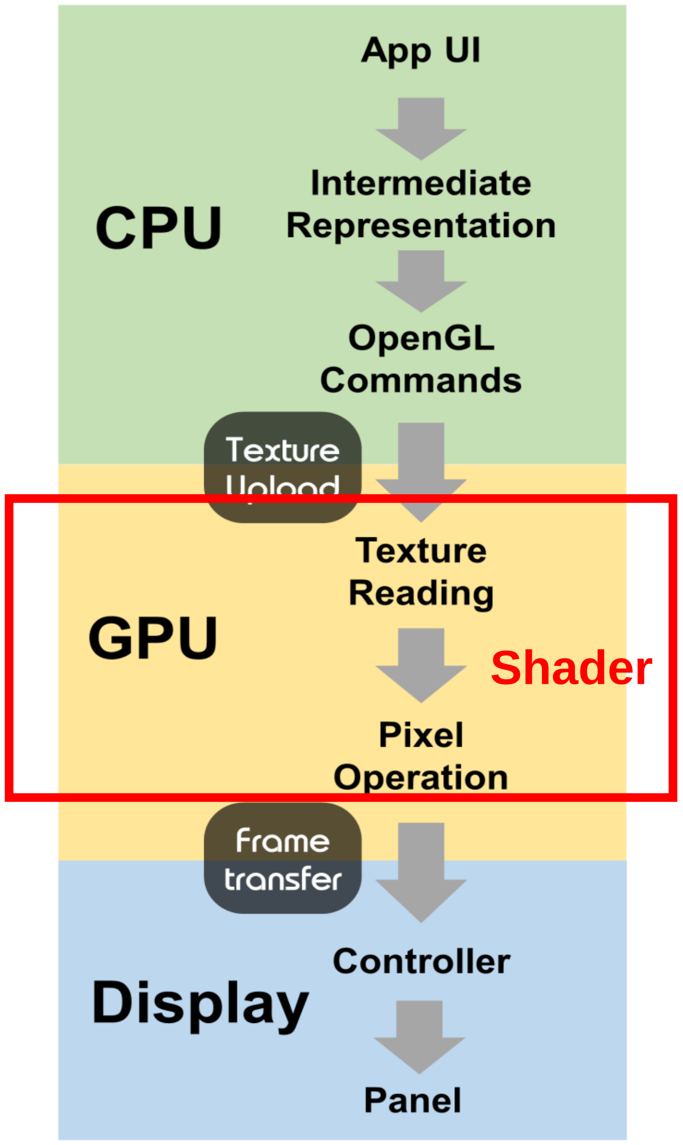
2. Read vertex data

3. Execute vertex shader

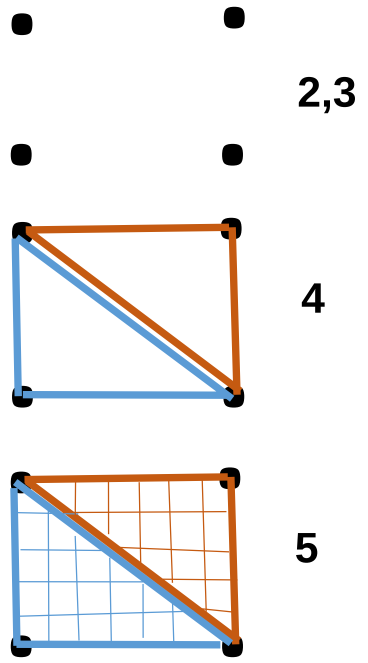
4. Assemble vertices



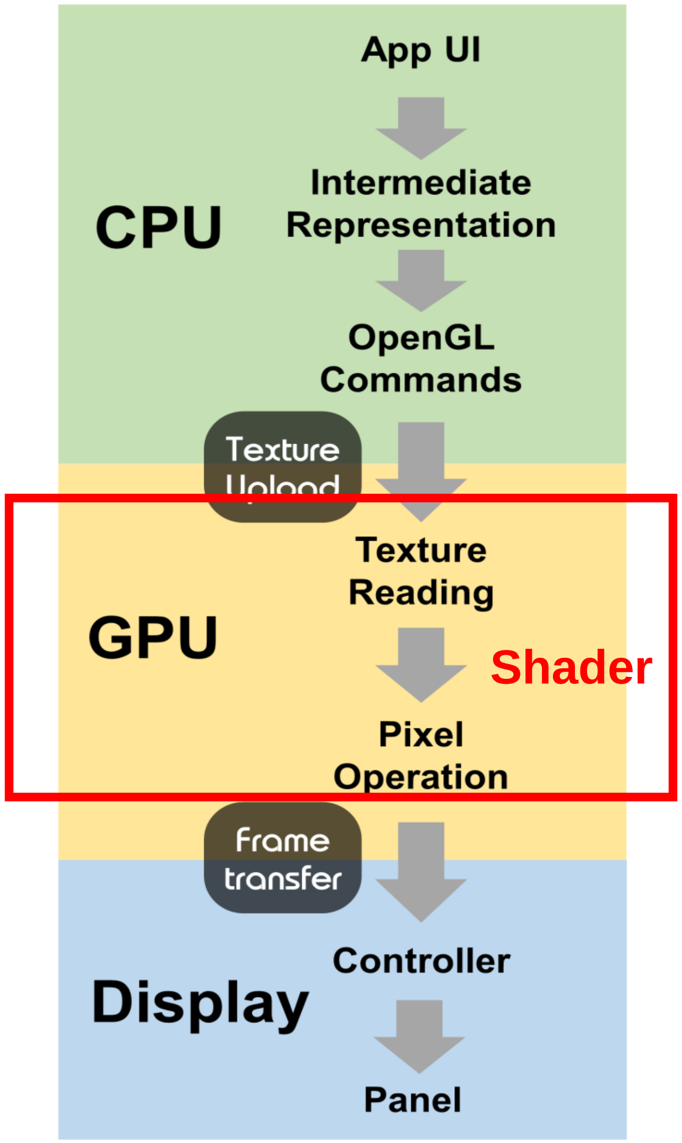
A 20,000 foot view of graphics stack



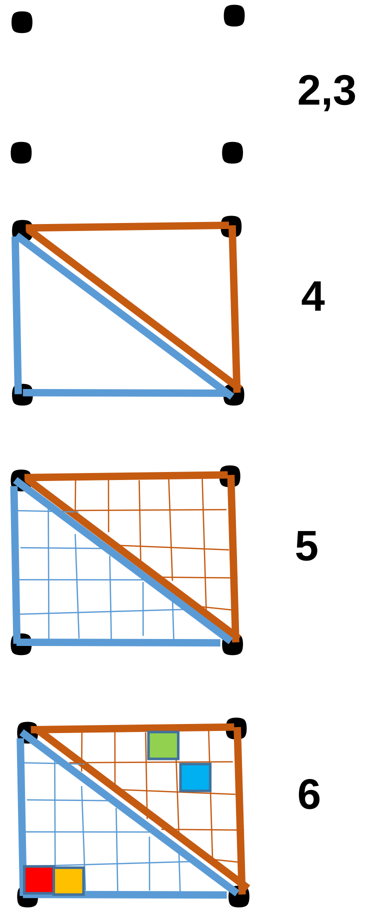
1. Load texture
2. Read vertex data
3. Execute vertex shader
4. Assemble vertices
5. Rasterize into fragments



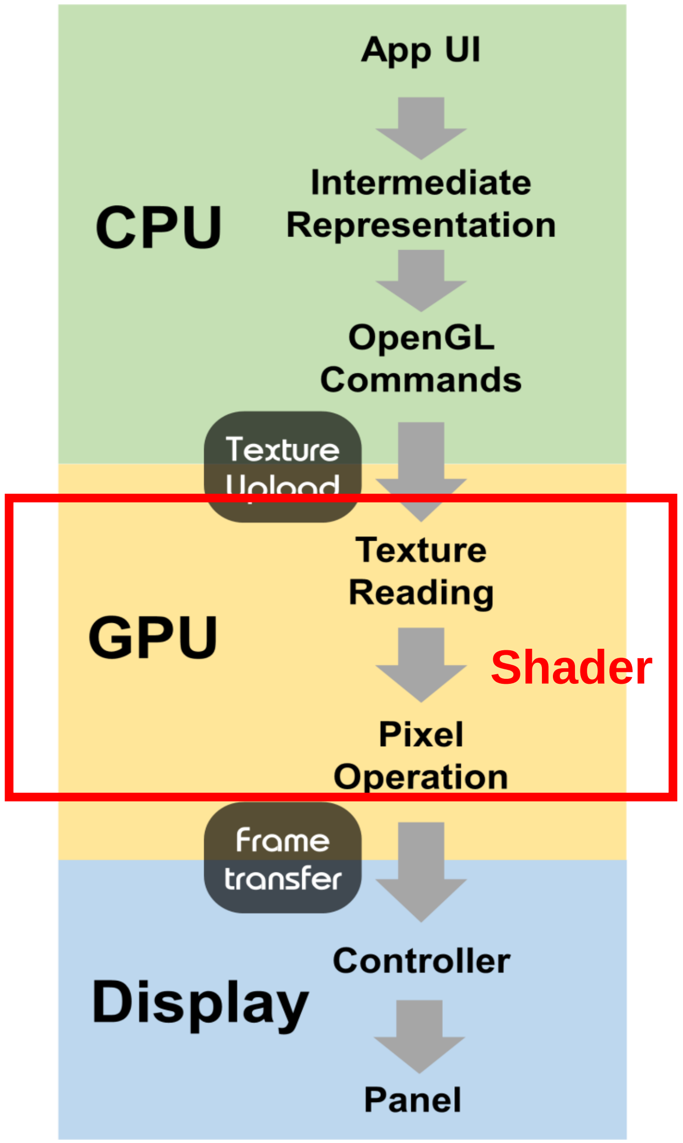
A 20,000 foot view of graphics stack



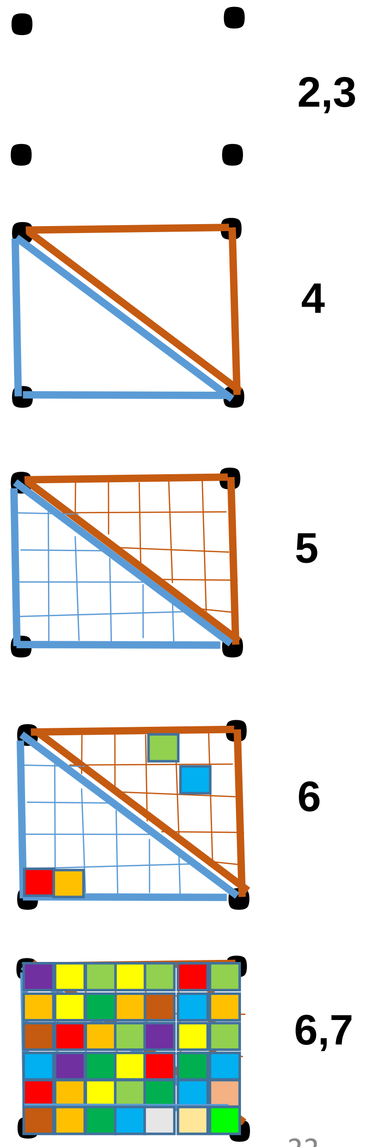
1. Load texture
2. Read vertex data
3. Execute vertex shader
4. Assemble vertices
5. Rasterize into fragments
6. Execute fragment shader



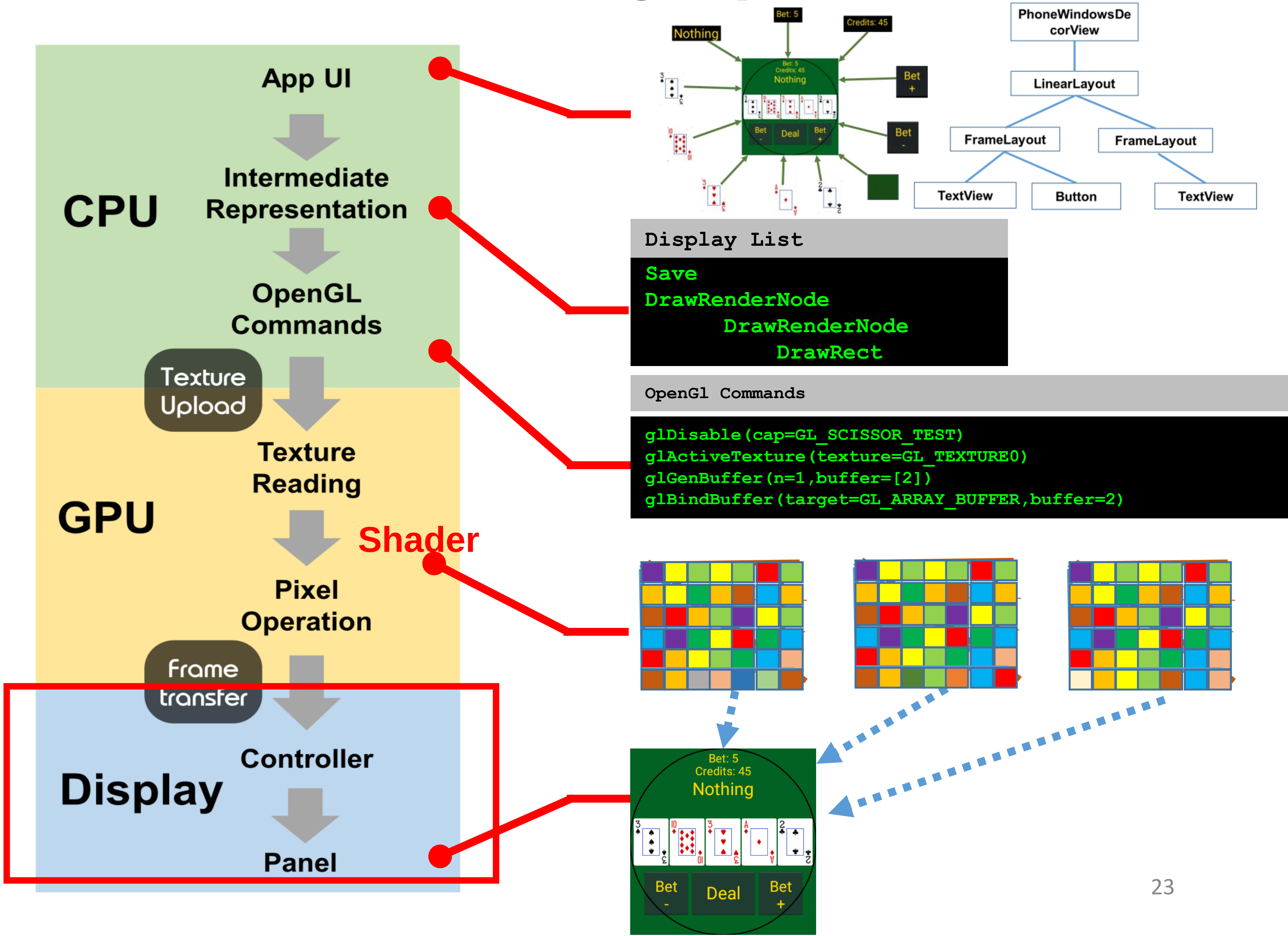
A 20,000 foot view of graphics stack



1. Load texture
2. Read vertex data
3. Execute vertex shader
4. Assemble vertices
5. Rasterize into fragments
6. Execute fragment shader
7. Write to frame buffer



A 20,000 foot view of graphics stack



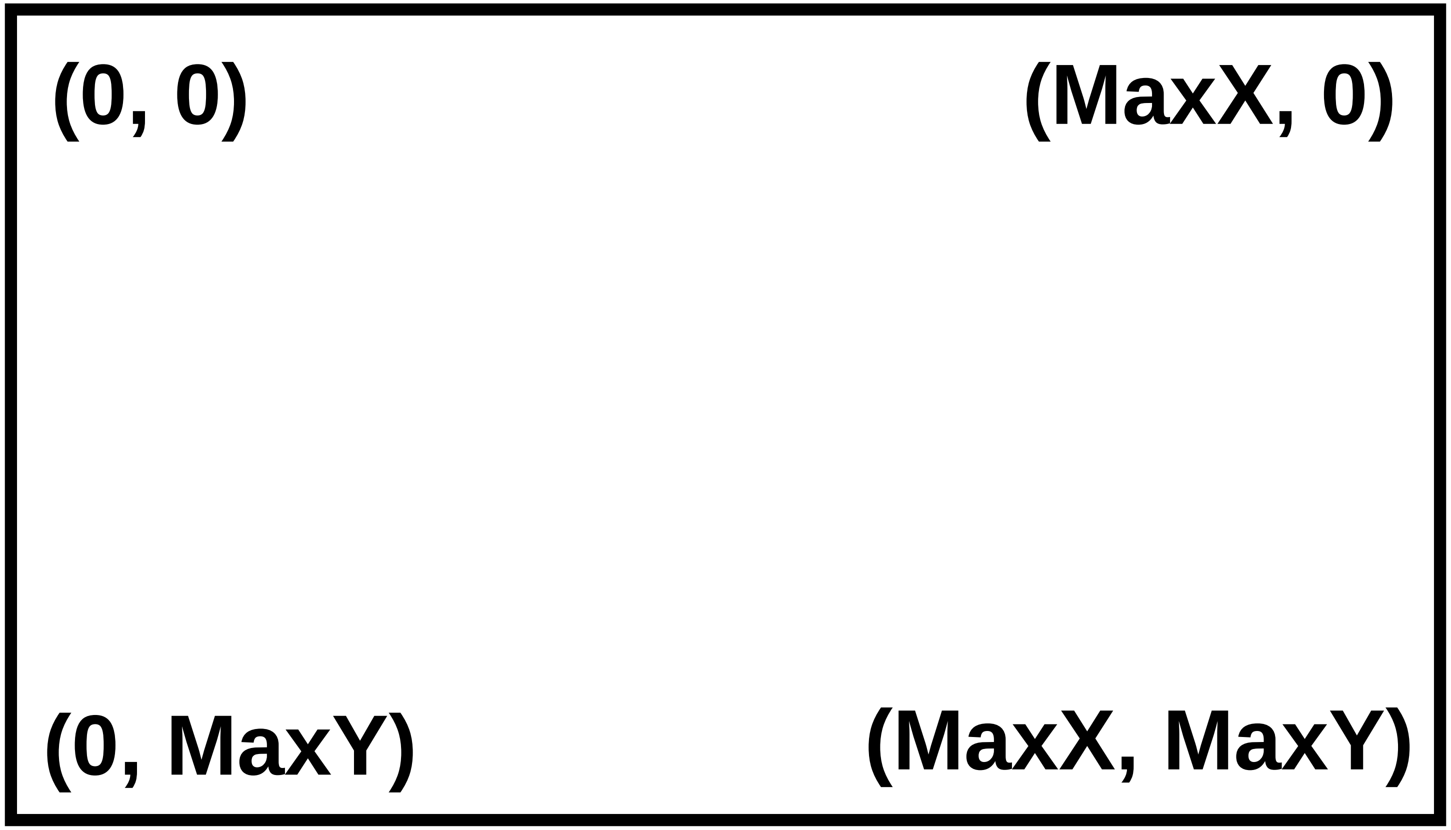
Display List

```
Save
DrawRenderNode
      DrawRenderNode
          DrawRect
```

OpenGl Commands

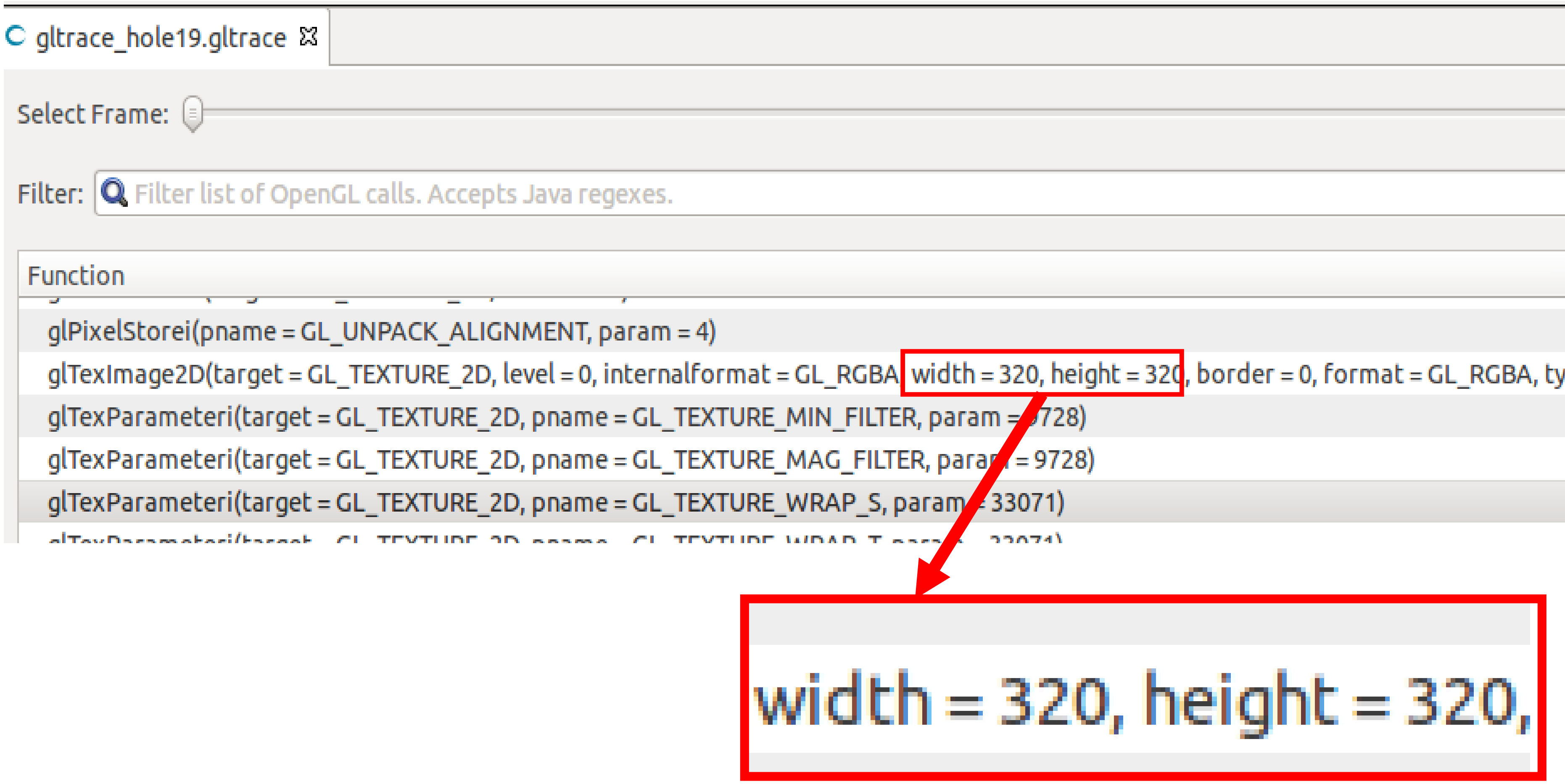
```
glDisable(cap=GL_SCISSOR_TEST)
glActiveTexture(texture=GL_TEXTURE0)
glGenBuffer(n=1,buffer=[2])
glBindBuffer(target=GL_ARRAY_BUFFER,buffer=2)
```

Graphics stack is oblivious to display shape
app evidence



Graphics stack is oblivious to display shape

OpenGL evidence



```
glPixelStorei(pname = GL_UNPACK_ALIGNMENT, param = 4)
glTexImage2D(target = GL_TEXTURE_2D, level = 0, internalformat = GL_RGBA, width = 320, height = 320, border = 0, format = GL_RGBA, ty
glTexParameteri(target = GL_TEXTURE_2D, pname = GL_TEXTURE_MIN_FILTER, param = 9728)
glTexParameteri(target = GL_TEXTURE_2D, pname = GL_TEXTURE_MAG_FILTER, param = 9728)
glTexParameteri(target = GL_TEXTURE_2D, pname = GL_TEXTURE_WRAP_S, param = 33071)
```

width = 320, height = 320,

Texture is specified as a rectangular

Graphics stack is oblivious to display shape

Device driver evidence

Device tree code from Linux kernel (for LG Watch R)

```
//apq8026-lenok-panel.dtsi:
qcom,mdss-dsi-panel-name = "LG4237 320P OLED
    command mode dsi panel";
qcom,mdss-dsi-panel-width = <320>;
qcom,mdss-dsi-panel-height = <320>;
```

Top questions

How many resources are wasted?

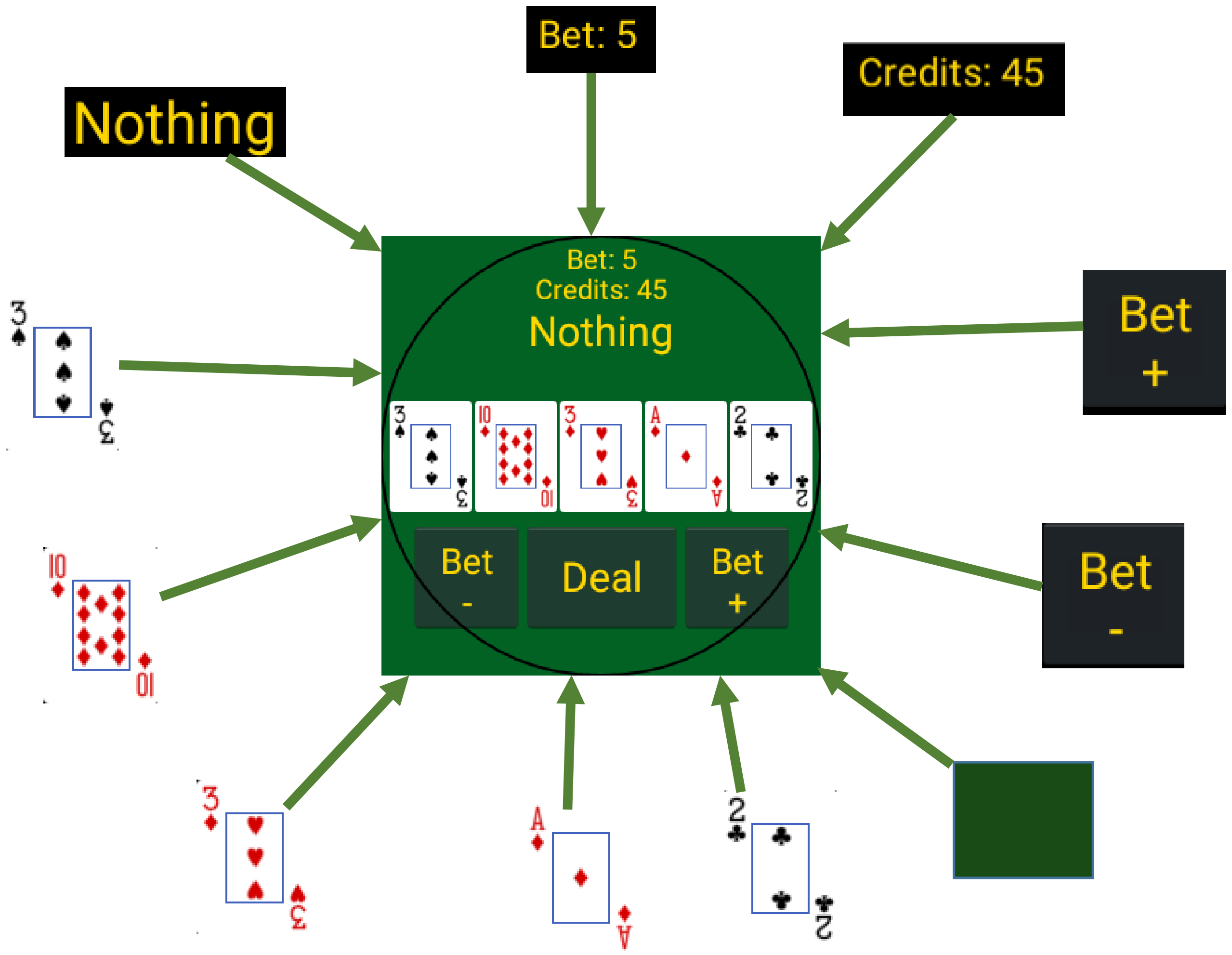
How should existing graphics stack adapt?

Top questions

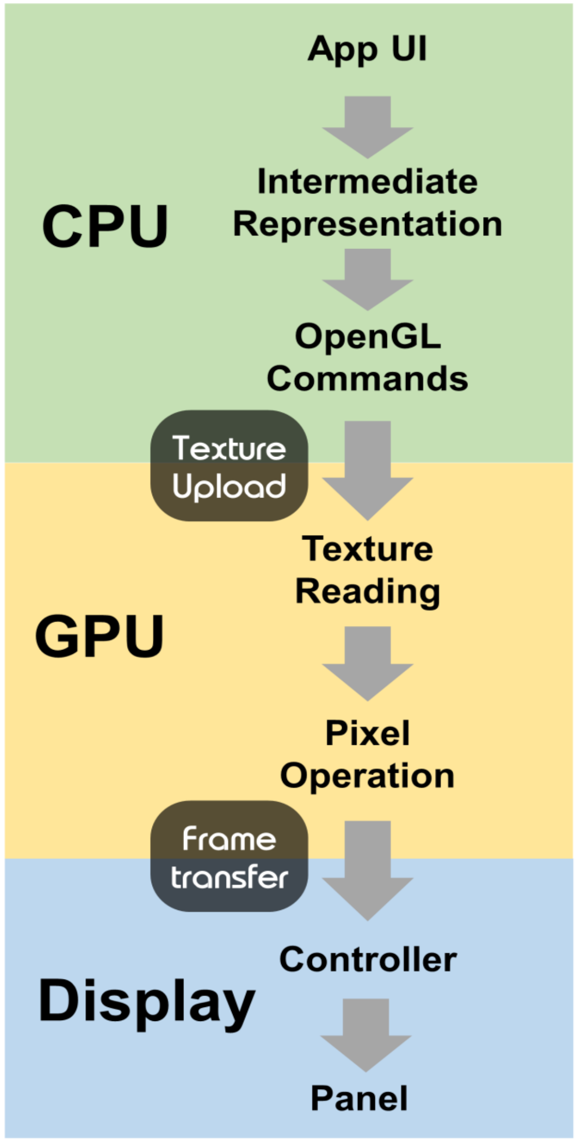
How many resources are wasted?

How should existing graphics stack adapt?

UI elements hidden & clipped by display boundary



Wasted CPU & GPU computation



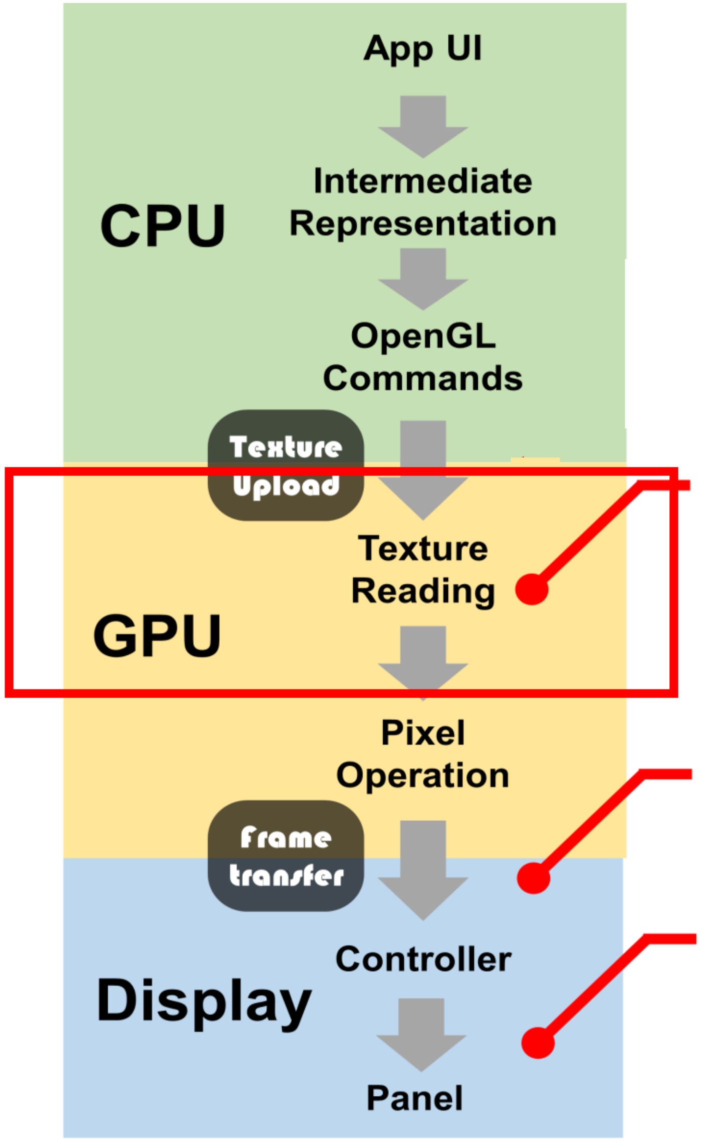
Drawing

Shader compile:8.2ms
Shader link: 1.2ms
Other: 2.4ms

Upload texture: 25ms

Rendering time: 4.5ms

Wasted memory traffic



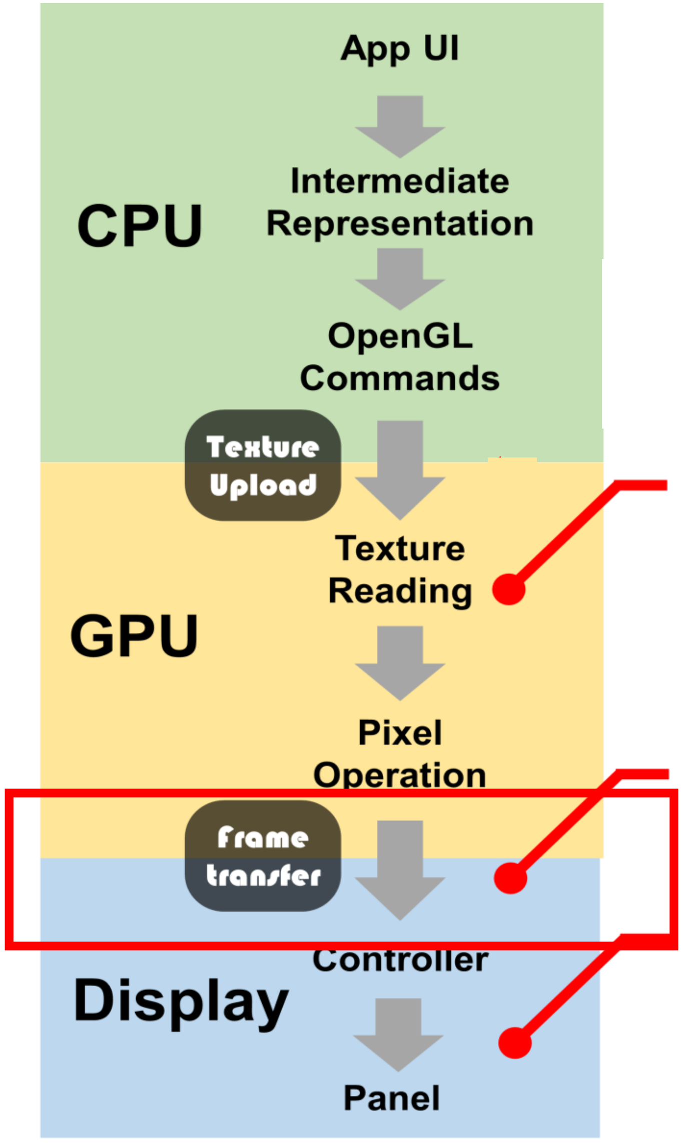
Wasted resources

Wasted Memory Traffic: 10 MBps

Wasted Memory Traffic: 10 MBps

Display Interface Traffic: 5 MBps

Wasted memory traffic



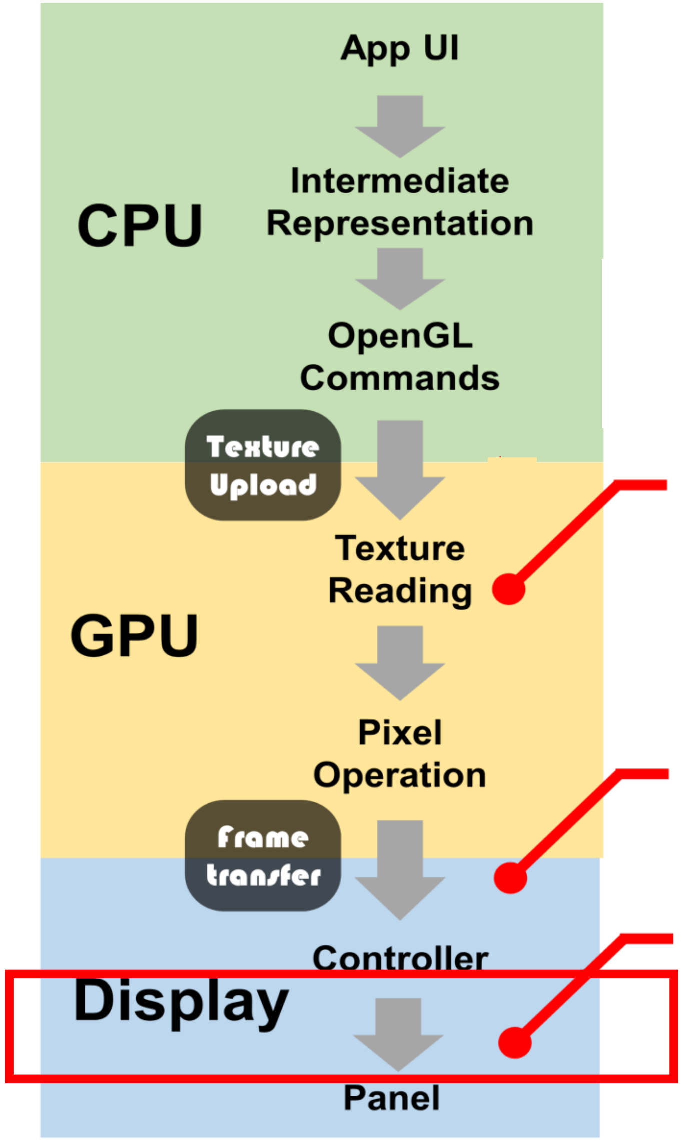
Wasted resources

Wasted Memory Traffic: 10 MBps

Wasted Memory Traffic: 10 MBps

Display Interface Traffic: 5 MBps

Wasted memory traffic



Wasted resources

Wasted Memory Traffic: 10 MBps

Wasted Memory Traffic: 10 MBps

Display Interface Traffic: 5 MBps

Top questions

How many resources are wasted?

- **Few views are completely hidden**
- **Not too much GPU/CPU computation is wasted**
- **Much memory traffic is wasted**

Study of tens of wearable Apps

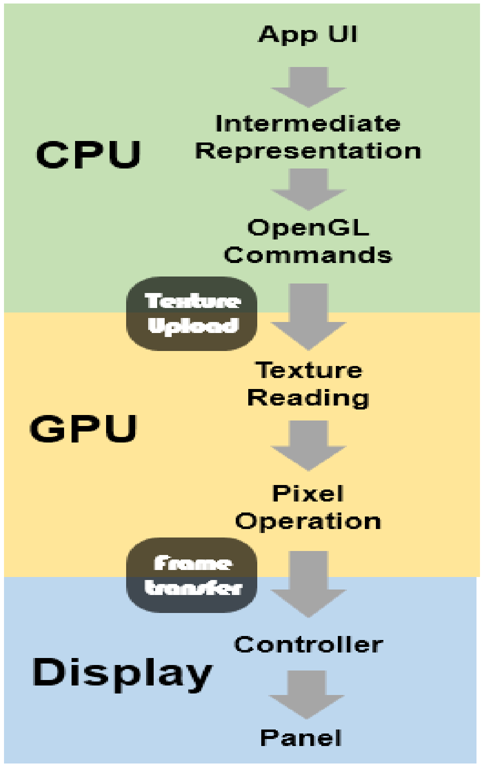
Apps	# of UI Views			Drawing Time				Rdr. Time
	Hidden	Clipped	Total	Shader compile	Shader link†	Texture upload†	Other cmds	
Google keep	×	×	×	8.6	1.3	4.4	2.9	4.3
Attopedia	0	9	10	8.2	1.2	25.0	2.4	4.5
Hole19	0	5	8	30.4	1.1	4.9	4.1	2.6
WearbottleSpinner	0	4	5	18.0	3.2	116.2	2.1	3.0
GridViewPager	0	6	9	23.9	4.4	2.0	2.0	2.8
Runtastic*	0	14	17	-	-	-	-	3.9
ReminderByTime*	0	13	14	-	-	-	-	3.8
Fit*	0	13	16	-	-	-	-	3.3
Weatherlive*	0	14	17	-	-	-	-	4.6
Instaweather*	0	13	16	-	-	-	-	3.8
Hangout*	0	13	16	-	-	-	-	3.7

Top questions

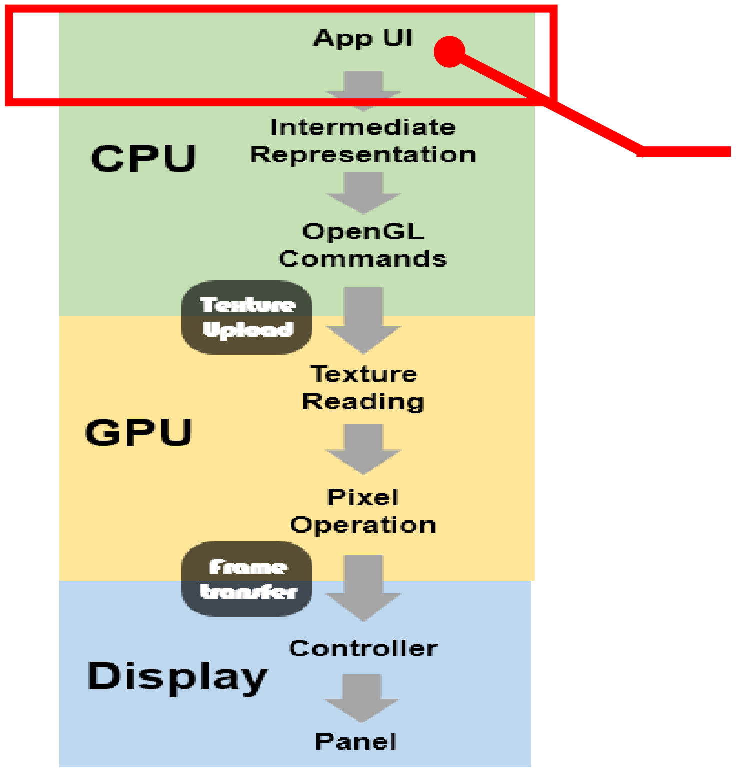
✓ How many resources are wasted?

How should existing graphics stack adapt?

Key: which layer should be aware of display shape



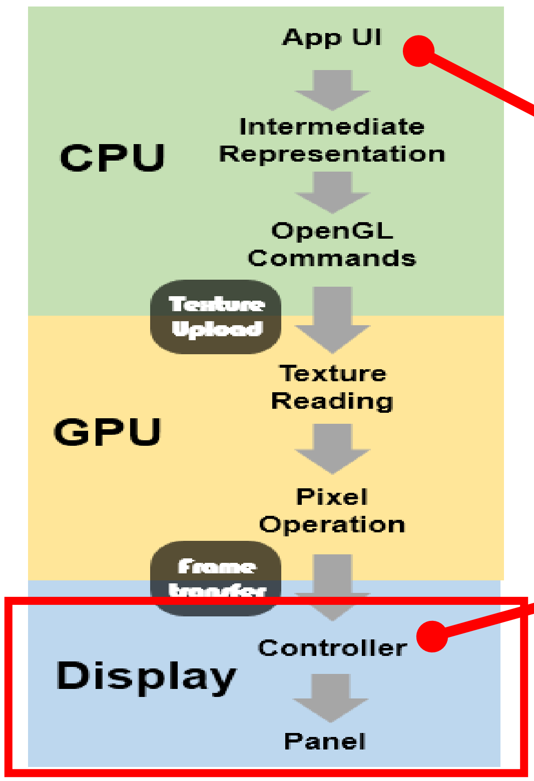
Key: which layer should be aware of display shape



Developer-controlled layout
Tedious & not portable

Complicated UI library
Tens of thousands of SLoC to be changed

Key: which layer should be aware of display shape

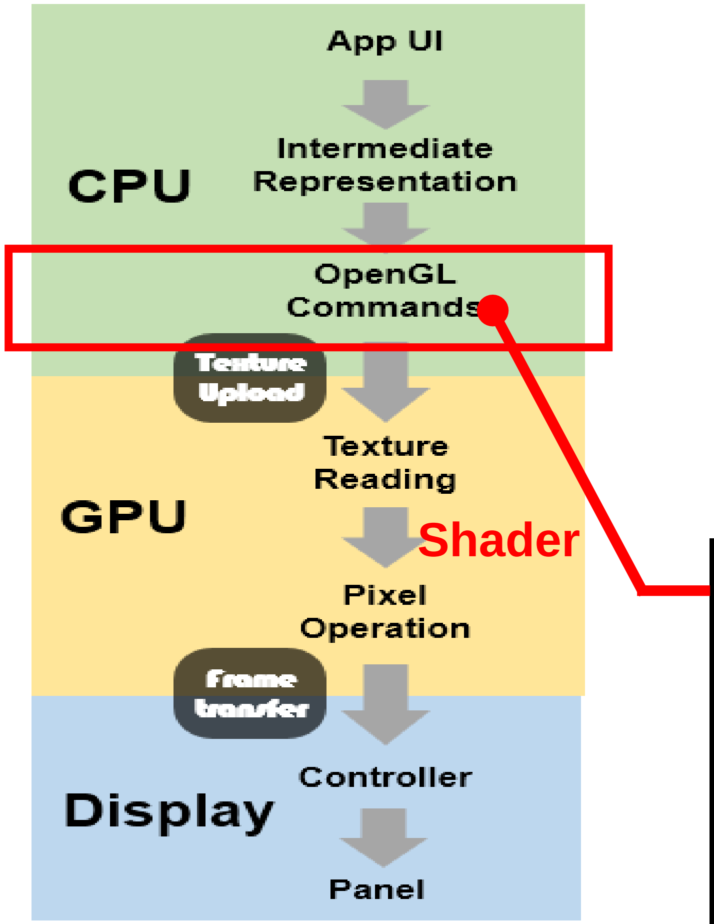


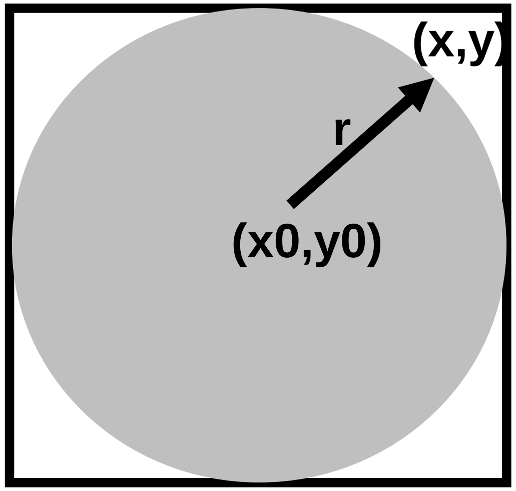
Developer-controlled layout
Tedious & not portable
Complicated UI library
Tens of thousands of SLoC to be changed

Cannot reduces wastes above this layer
LPD [ATC15]
Incomplete

Pilot solution: OpenGL interposition

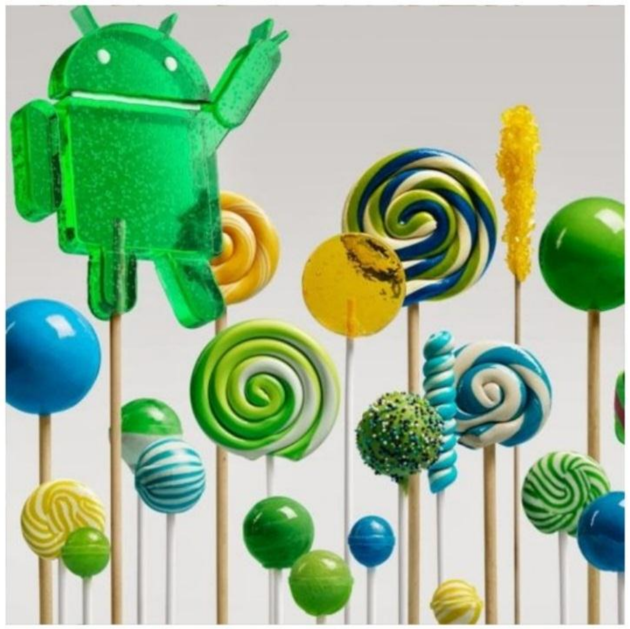
- Key point: rewrite shader program one-the-fly



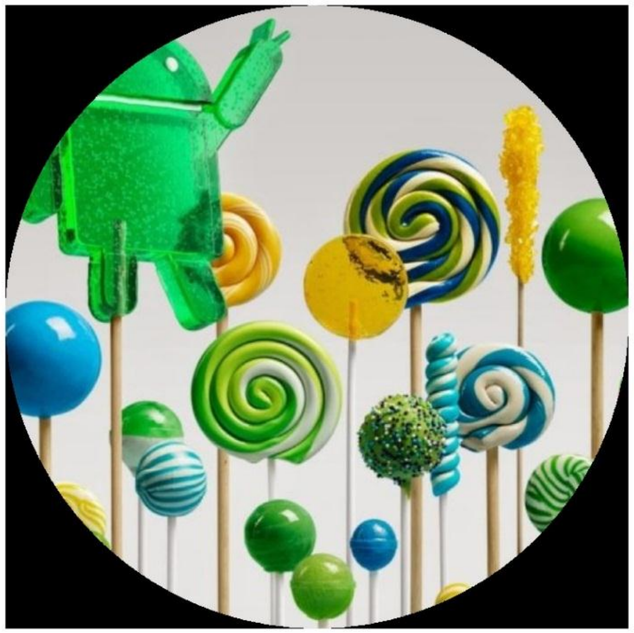


```
void main()
{
  if((x-x0)*(x-x0)+(y-y0)*(y-y0)<r*r){
    //Render the pixel if in circular area
    gl_FragColor = \
      texture2D(textureUnit, textureCoordinate);
    ... ...
  }
}
```

Pilot solution: OpenGL interposition

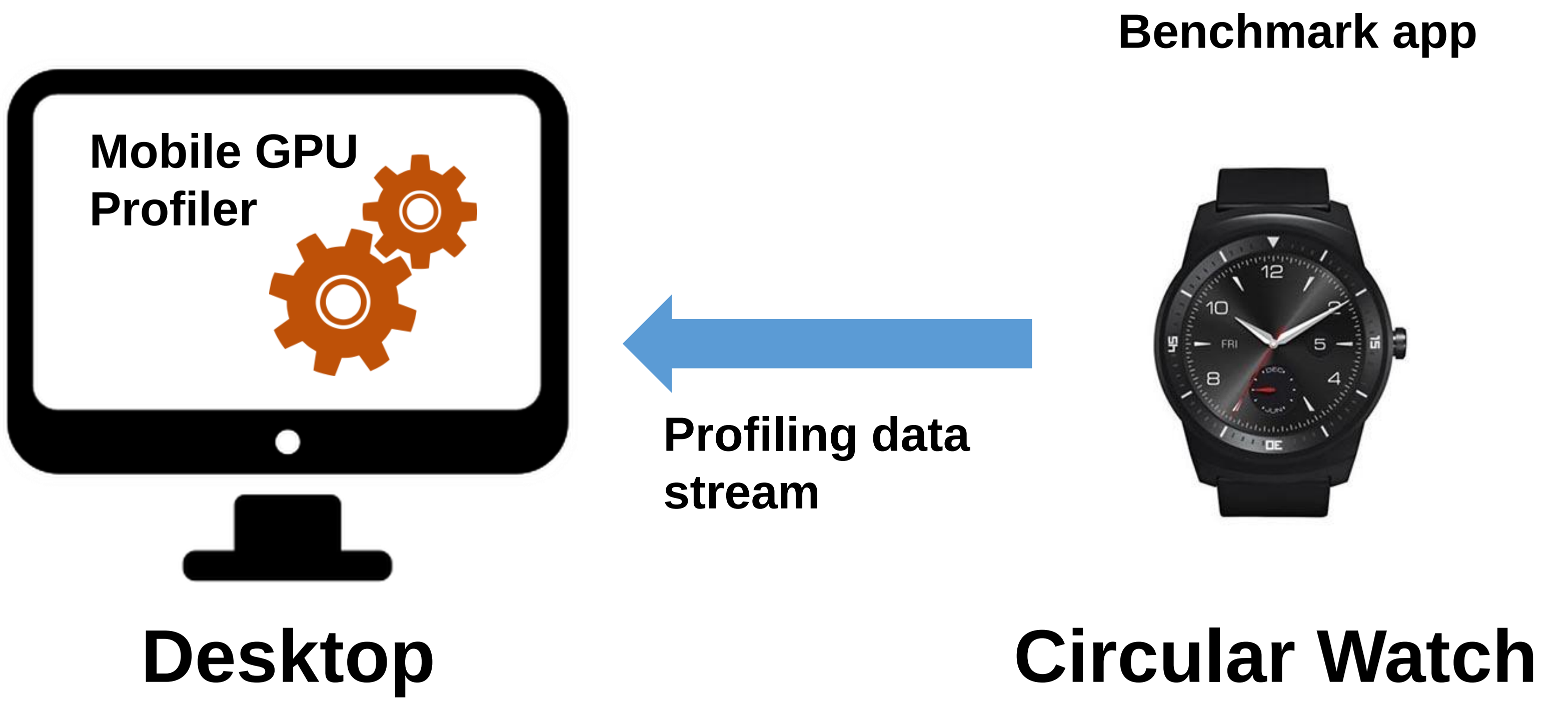


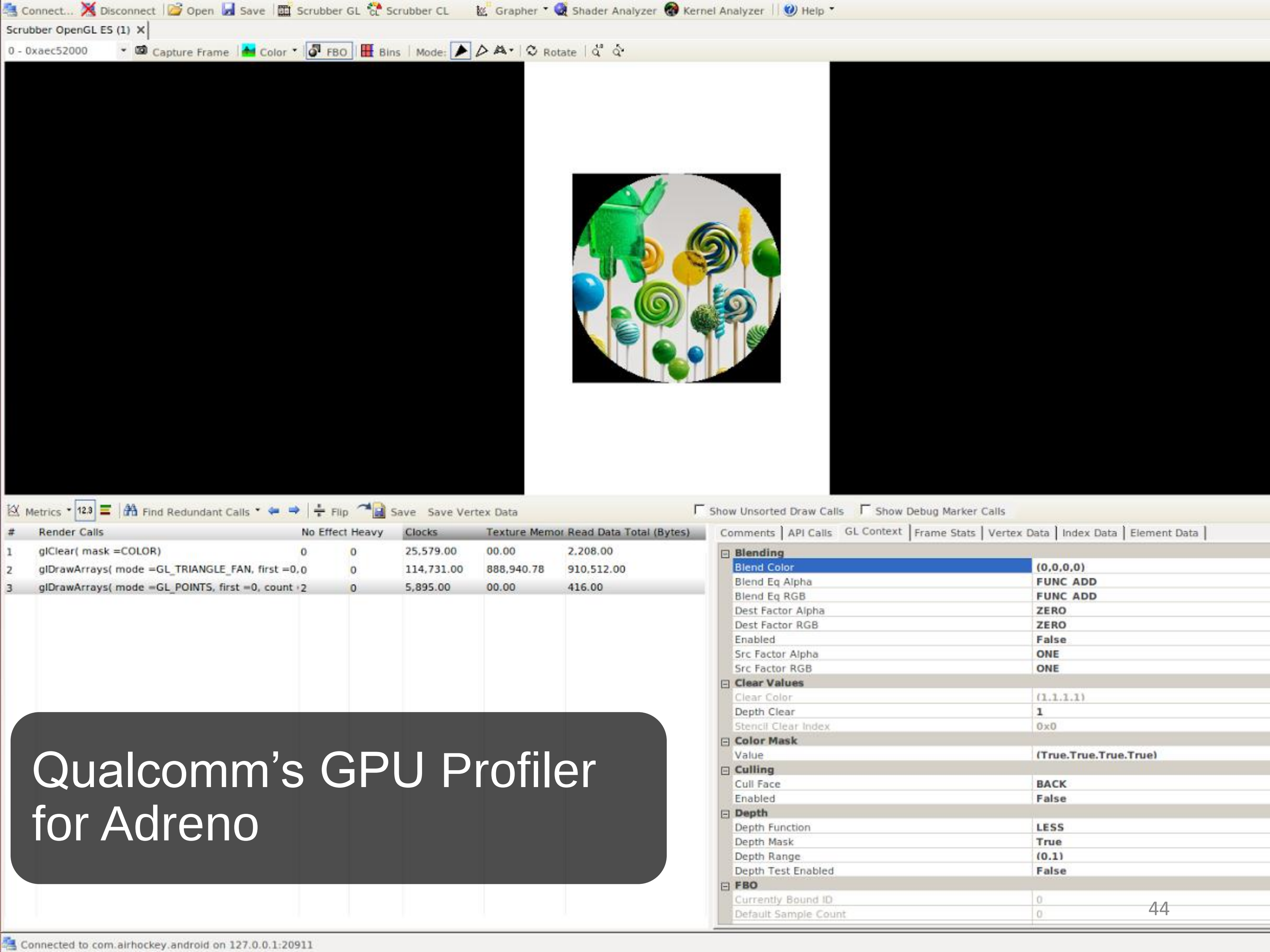
Before Rewriting Shader



After Rewriting Shader

Evaluation: setup (ideal)




Connect... Disconnect Open Save Scrubber GL Scrubber CL Grapher Shader Analyzer Kernel Analyzer Help
Scrubber OpenGL ES (1)
0 - 0xaec52000 Capture Frame Color FBO Bins Mode: Rotate
Metrics 12.3 Find Redundant Calls Flip Save Save Vertex Data
Show Unsorted Draw Calls Show Debug Marker Calls
Comments API Calls GL Context Frame Stats Vertex Data Index Data Element Data

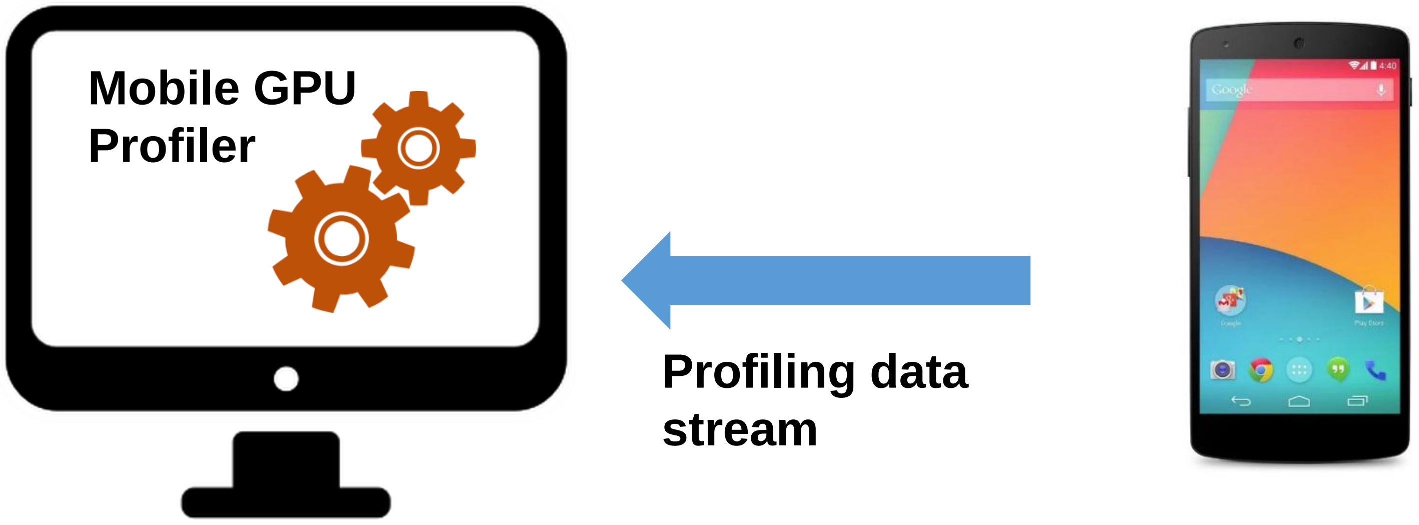
#	Render Calls	No Effect	Heavy	Clocks	Texture Memor	Read Data Total (Bytes)
1	glClear(mask =COLOR)	0	0	25,579.00	00.00	2,208.00
2	glDrawArrays(mode =GL_TRIANGLE_FAN, first =0,	0	0	114,731.00	888,940.78	910,512.00
3	glDrawArrays(mode =GL_POINTS, first =0, count	2	0	5,895.00	00.00	416.00

Blending	
Blend Color	(0,0,0,0)
Blend Eq Alpha	FUNC ADD
Blend Eq RGB	FUNC ADD
Dest Factor Alpha	ZERO
Dest Factor RGB	ZERO
Enabled	False
Src Factor Alpha	ONE
Src Factor RGB	ONE
Clear Values	
Clear Color	(1,1,1,1)
Depth Clear	1
Stencil Clear Index	0x0
Color Mask	
Value	(True,True,True,True)
Culling	
Cull Face	BACK
Enabled	False
Depth	
Depth Function	LESS
Depth Mask	True
Depth Range	(0,1)
Depth Test Enabled	False
FBO	
Currently Bound ID	0
Default Sample Count	0

Connected to com.airhockey.android on 127.0.0.1:20911

Qualcomm's GPU Profiler for Adreno

Evaluation: setup (actual)

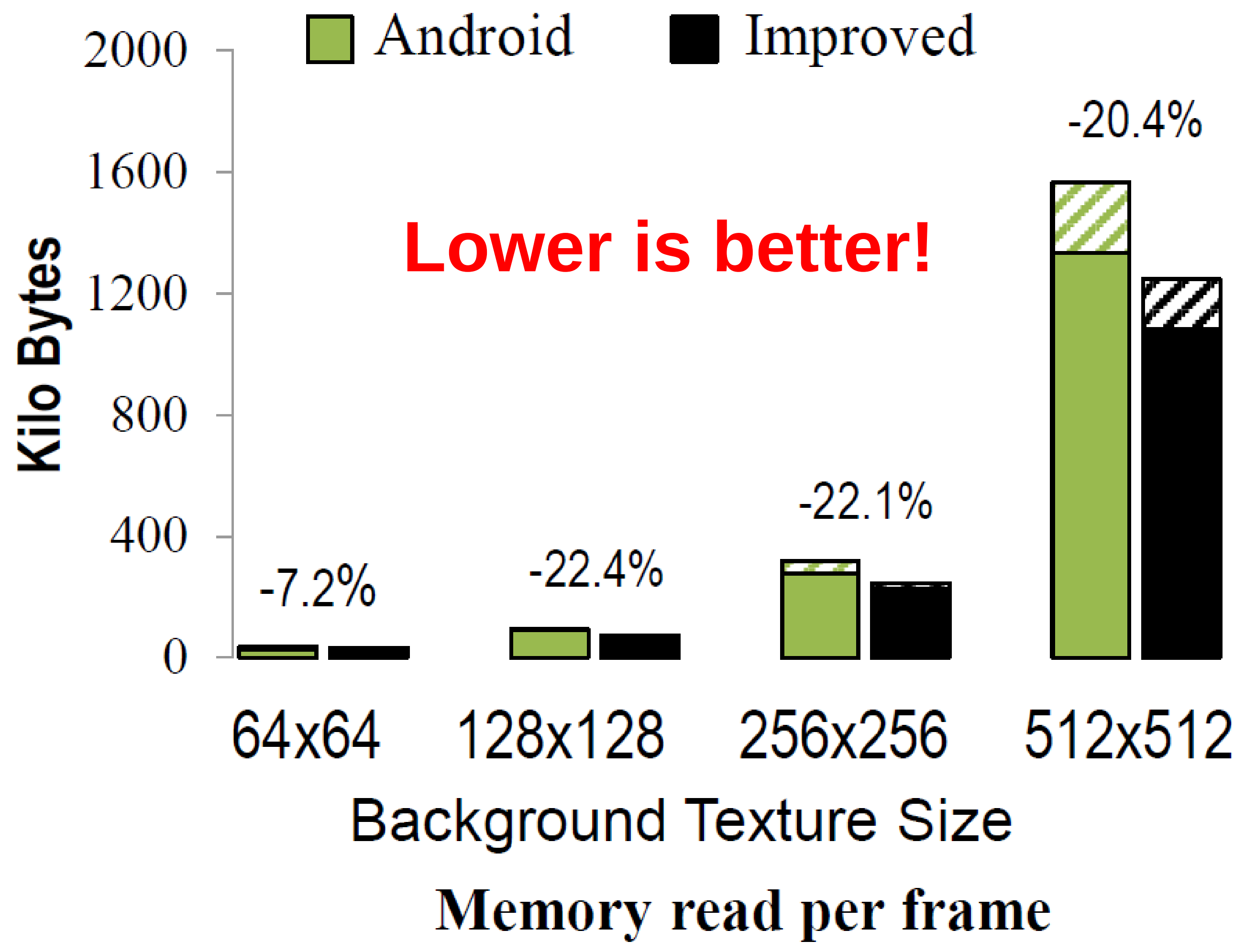


Desktop

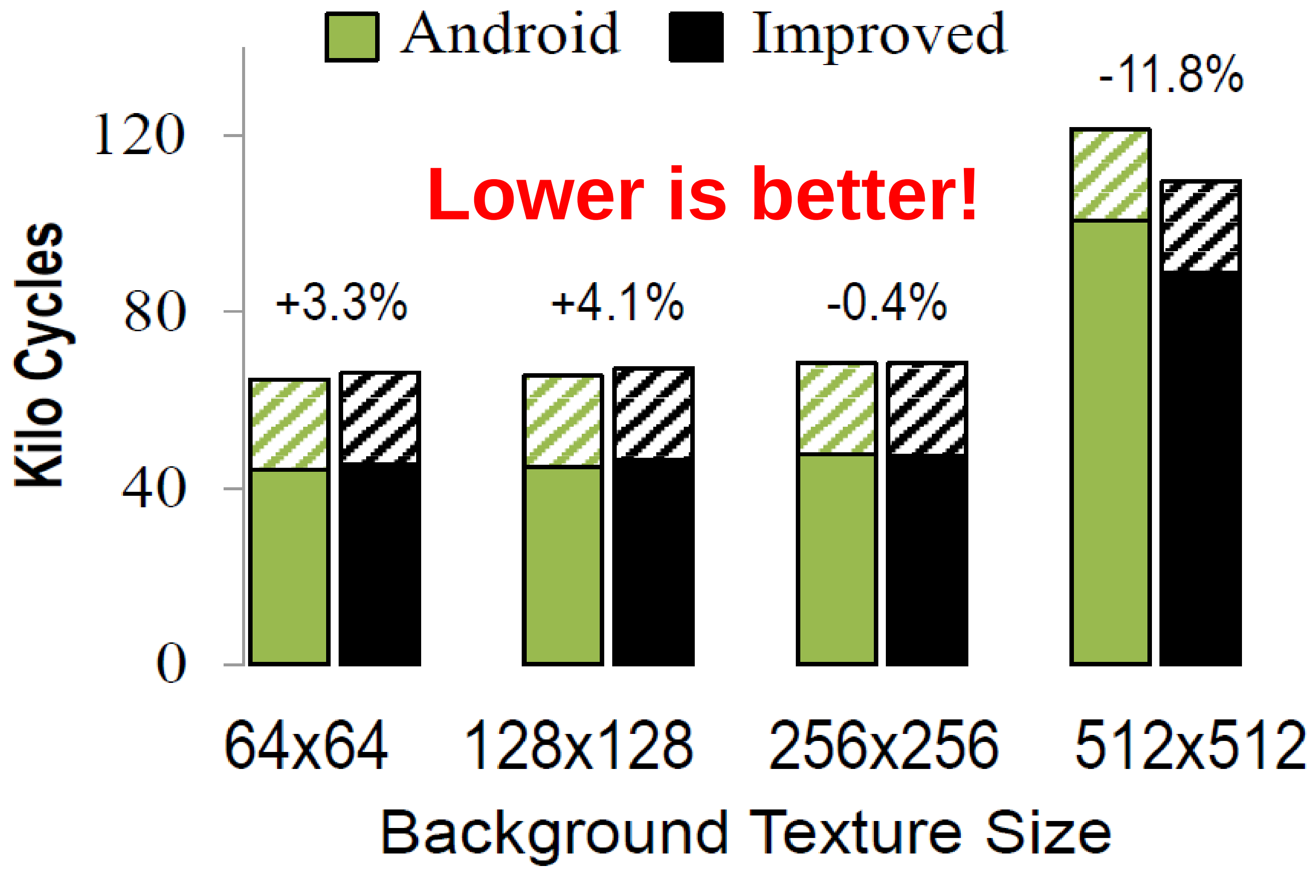
Nexus 5

Similar QCOM SoC
Same-generation GPU

Result: Reduced GPU memory read

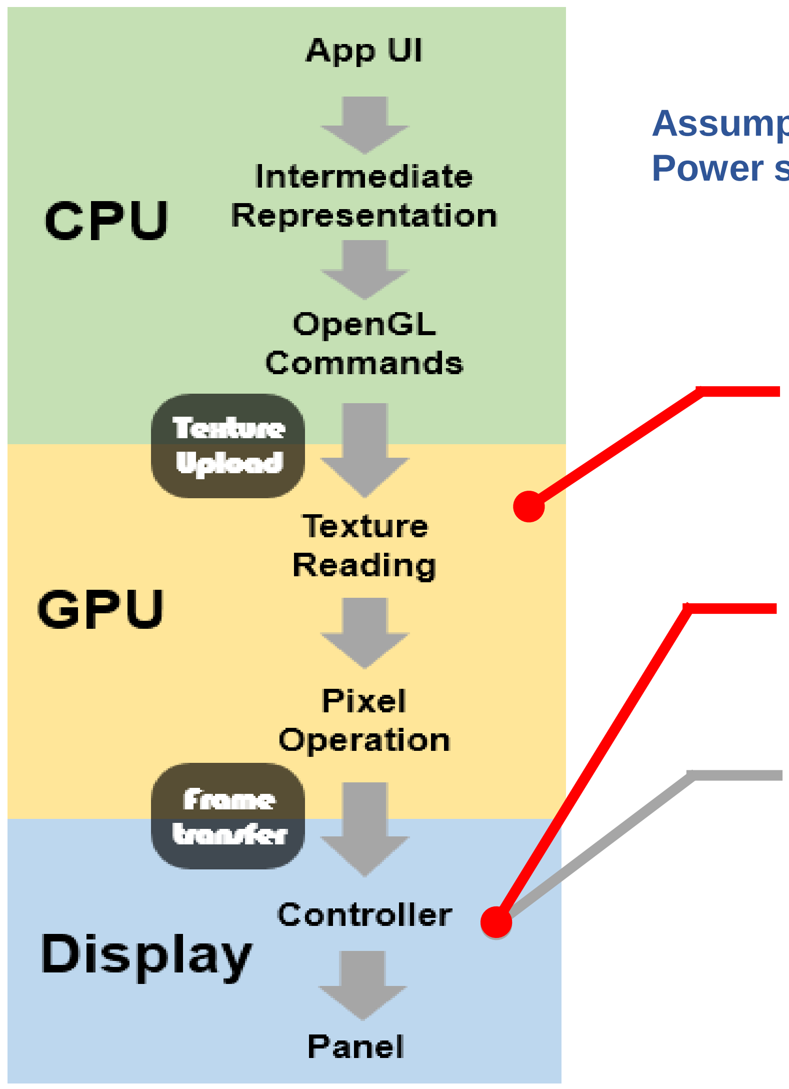


Result: Reduced GPU cycles



Total number of cycles per frame

Estimated Power Saving



Assumption:
Power saving roughly proportional to reduced traffic

Our method
Reduced memory traffic 10 MBps
Will save power 3.9 mW

Our method (if novel display controller)
Will reduce memory traffic 15 MBps
Will save power 5.8 mW

LPD [ATC15]
Reduced DRAM-to-display traffic 7 MBps
Saved power 2.7 mW

Summary

✓ How many resources are wasted?

- Graphics stack is wasting resources due to screen shape
- Quantified the resource wasted on the LG watch R

✓ How should existing graphics stack adapt?

- Pilot solution: interposing OpenGL + shader program
- Reduced 22.4% memory traffic + 11.8% GPU cycles

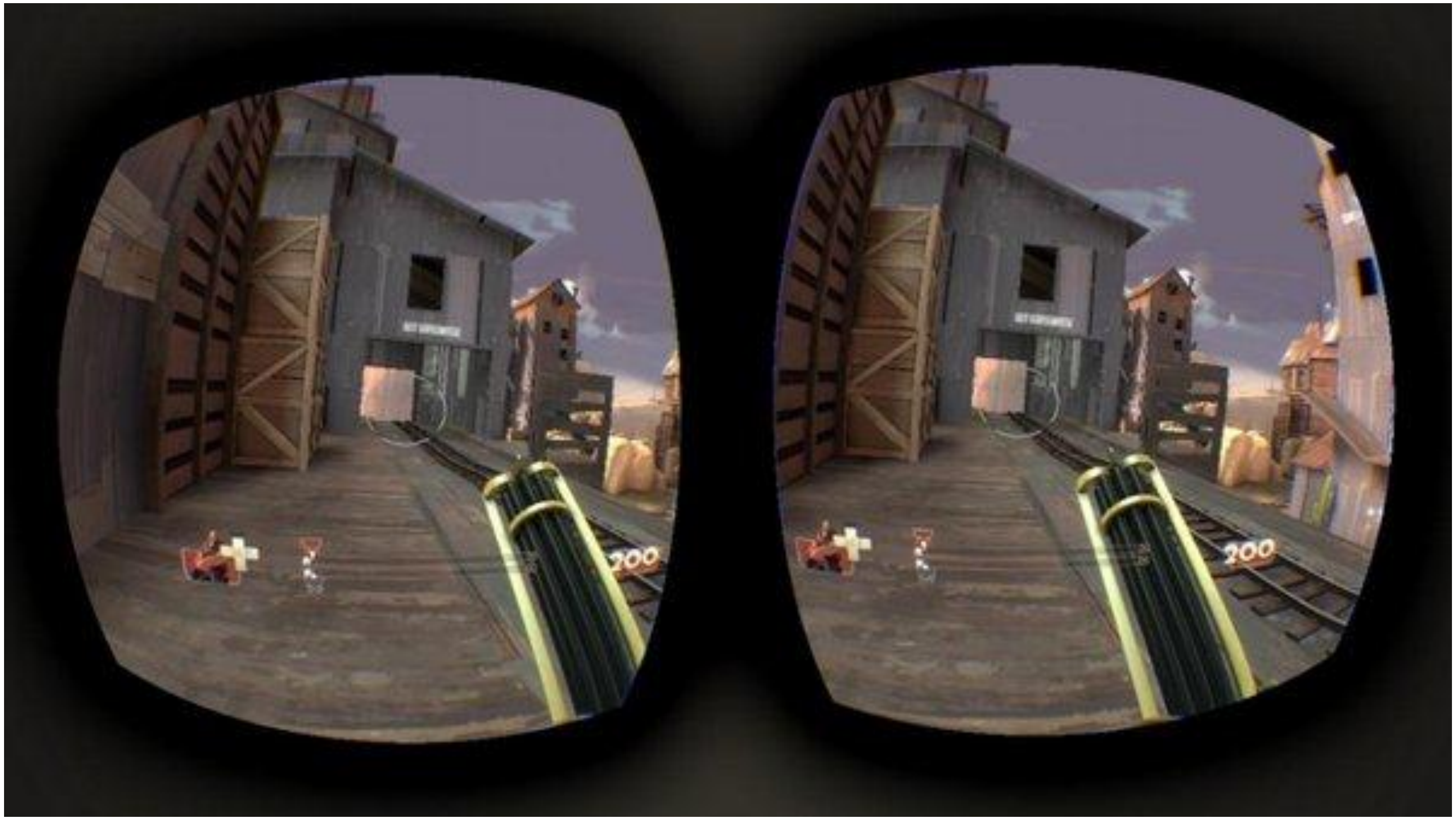
Outlook: future irregular displays

Outlook: future irregular displays

Dashboard

Outlook: future irregular displays

Virtual Reality Helmet



Designing for future irregular displays

- Higher waste → compelling to adapt graphics stack
- Redesigning graphics stack may be justified

A key lesson

- New form factors drive system software design

Summary

✓ **How many resources are wasted?**

- Graphics stack is wasting resources due to screen shape
- Quantified the resource wasted on the LG watch R

✓ **How should existing graphics stack adapt?**

- Pilot solution: interposing OpenGL + shader program
- Reduced 22.4% memory traffic + 11.8% GPU cycles

Future thoughts

- New form factors drive system software design

Q/A

- How often does the waste occur?
- Why can't developers just design for irregular displays?
- Why should we care about 10mW?
- Why can't you measure the power physically?
- Can't we just overhaul the UI library?
  - (From one Microsoft guy)
- Can you solve this problem completely?
- How do you rewrite the GPU shaders?