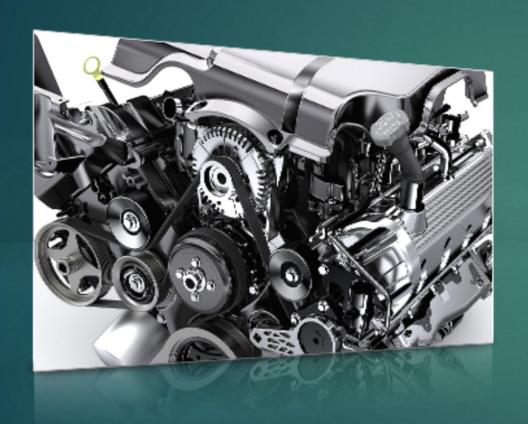
# GFX-rs

ARCHITECTURE

# Mhys

- ► Safety in types and hardware states
- ▶ API abstraction
- ► Scalable performance
- ▶ Convenience

### 1. Safety



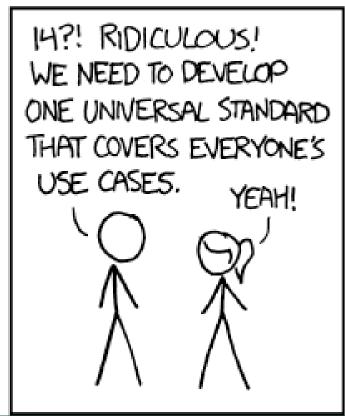
"Most bugs are a result of the execution state not being exactly what you think it is."

@John Carmack, 2007

#### 2. Abstraction

HOW STANDARDS PROLIFERATE: (SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)

SITUATION: THERE ARE 14 COMPETING STANDARDS.

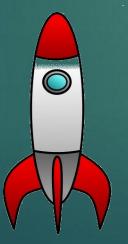


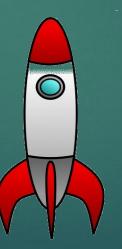
500N:

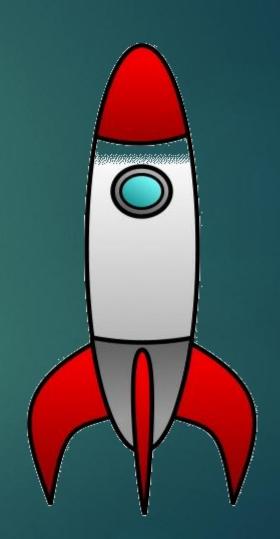
SITUATION: THERE ARE 15 COMPETING STANDARDS.

#### 3. Performance

- ► Automatic state caching
- ► Multi-threading





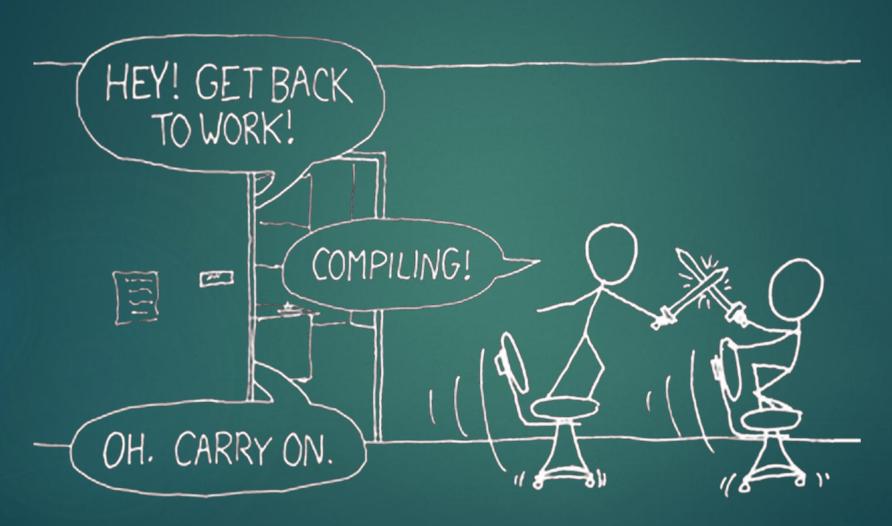


#### 4. Convenience

```
#[vertex_format]
struct Vertex {
  #[as_float]
  #[name = "a_Pos"]
  pos: [i8, ..3],
  #[as_float]
  #[name = "a_TexCoord"]
  tex_coord: [<u>u8</u>, ..2],
```

```
#[shader_param(CubeBatch)]
struct Params {
  #[name = "u_Transform"]
  transform: [[f32, ..4], ..4],
  #[name = "t_Color"]
  color: gfx::shade::TextureParam,
```

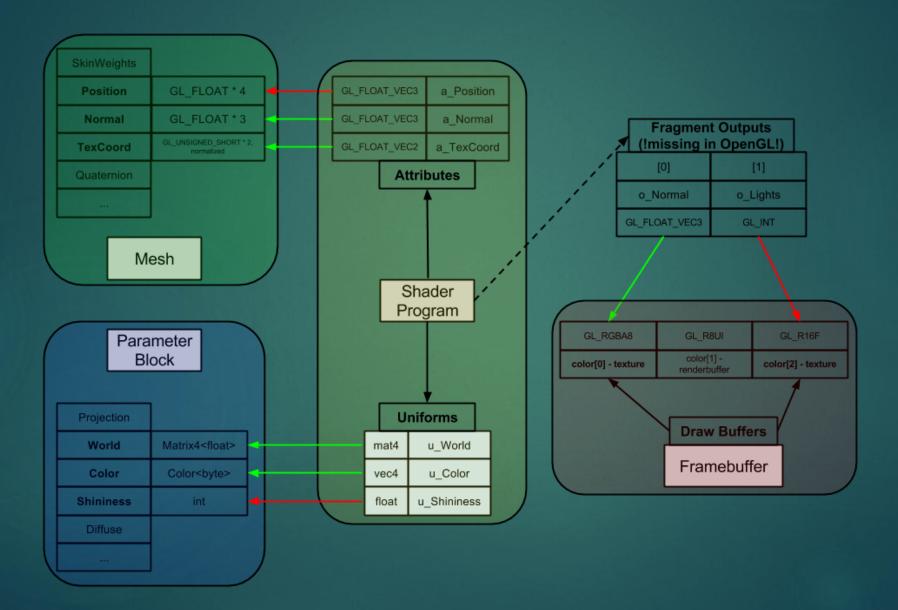
## // sleep(1) //



### Hows

- ▶ Bind-less and state-less\* interface:
  - "rederer.draw(&batch, &frame);"
- Matching input requirements against the provided data:
  - Vertex Attributes
  - Shader Uniforms
  - ► Textures and Buffers
- Draw buffers construction != execution
  - Multi-threaded rendering
  - ▶ Buffer serialization and reuse
- State caching

### Draw-call verification



### Draw buffers

Draw buffer 1 on Thread 1:

clear

batch

update

batch

batch

..

Draw buffer 2 on Thread 2:

•••

Draw buffer 3 on Thread 2:

• • •

Command buffer 1:

Graphics states:

▶ frame buffer, mesh

▶ shader, rasterizer

Draw commands:

▶ draw(), clear(), ...

▶ Update commands:

▶ buffers, textures

Data buffer 1:

buffer data

texture data

buffer 2 data

Device:

glBindFramebuffer()

glClear()

glVertexAttribPointer()

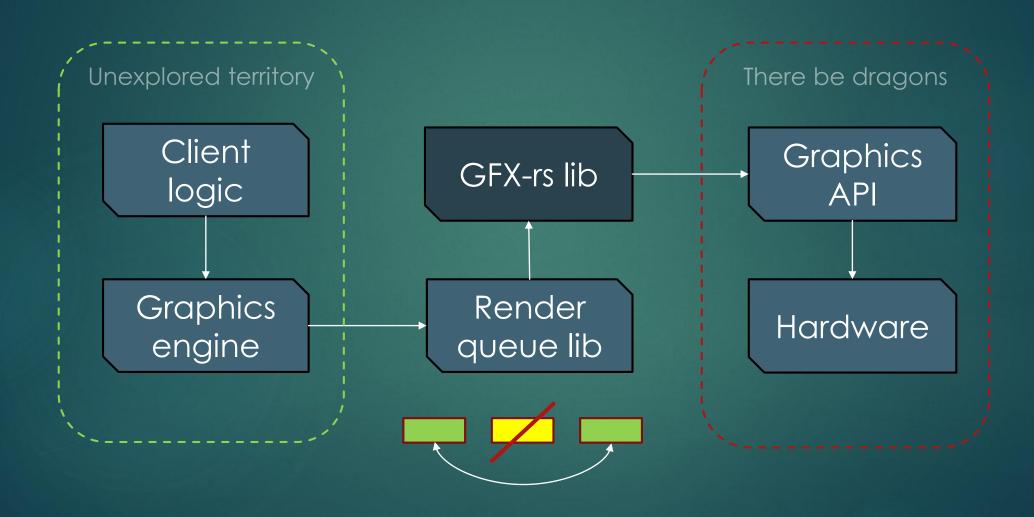
glBufferSubData()

glUseProgram()

glDrawElements()

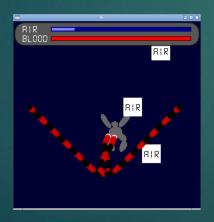
• • •

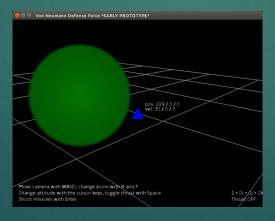
### Application structure in layers



### Existing projects

- https://github.com/PistonDevelopers/gfx\_graphics
  - https://github.com/bvssvni/rust-snake
- Von Neumann Defense Force
  - ► Sci-fi MMO by Hanno Braun, <a href="http://www.vndf.de">http://www.vndf.de</a>
- ► My experiments in <a href="https://github.com/kvark/scene-rs">https://github.com/kvark/scene-rs</a>
  - ➤ Yet another asteroids game!



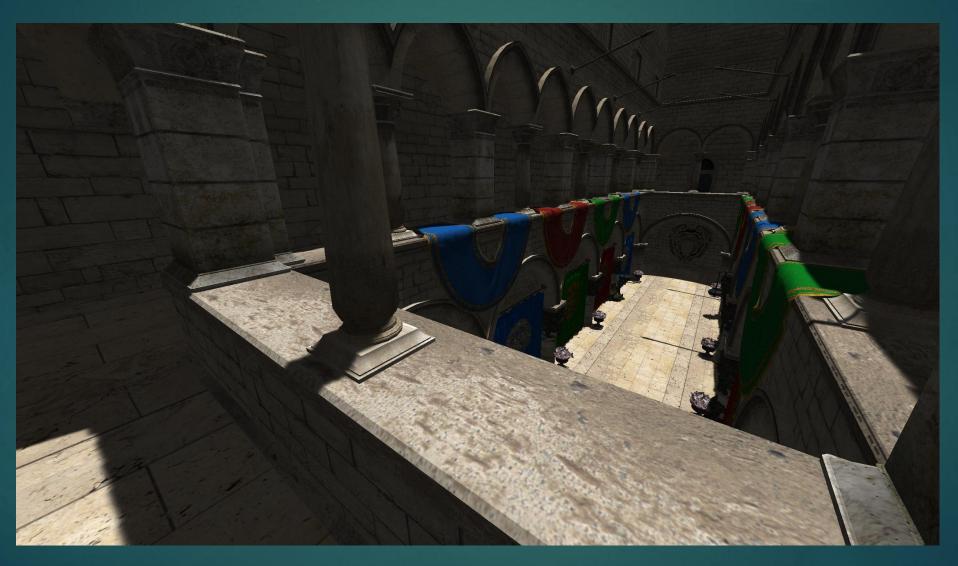




# Hematite



## Snowmew



### Join GFX community

- ▶ Use it, give feedback
- ► Say hello at <a href="https://gitter.im/gfx-rs/gfx-rs">https://gitter.im/gfx-rs/gfx-rs</a>
- ▶ Lots of work to do:
  - ► Latest and greatest features
  - More API back-ends
  - Shader abstraction
  - Bug fixes and refactoring