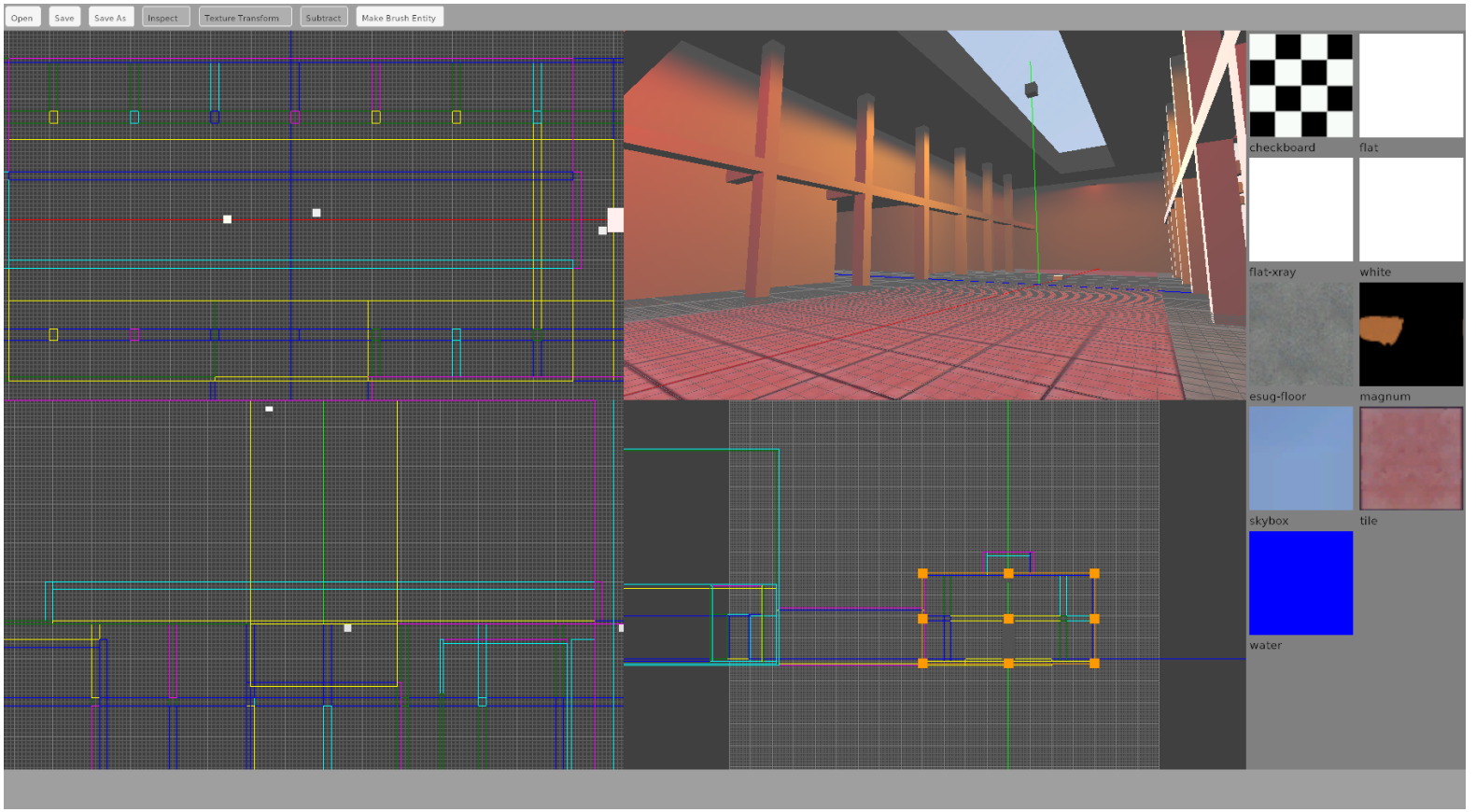
Woden 2: Developing a modern 3D graphics engine

Ronie Salgado
Universidad de Chile



Talk Outline: Foundations

- Why rewriting Woden?
  - OpenGL problems
  - Classic graphics API (OpenGL, Direct 3D except 12)
  - “Modern” PC architecture
  - Modern graphics API (Vulkan, Direct 3D 12 and Metal)
- Core abstraction layers
  - AbstractGPU
  - Dastrel (Data Stream Language)
  - Slovim (Dastrel back-end)

Talk Outline: Woden 2 Layers

- WodenMath library
- Core
  - Pipeline state cache
  - Resource cache
  - GPU resource management
- Basic layers
  - Scene graph
  - Woden athens backend
- Application layers
  - Woden Roassal ported to Woden 2
  - Woden 2 Game System
  - Level Editor

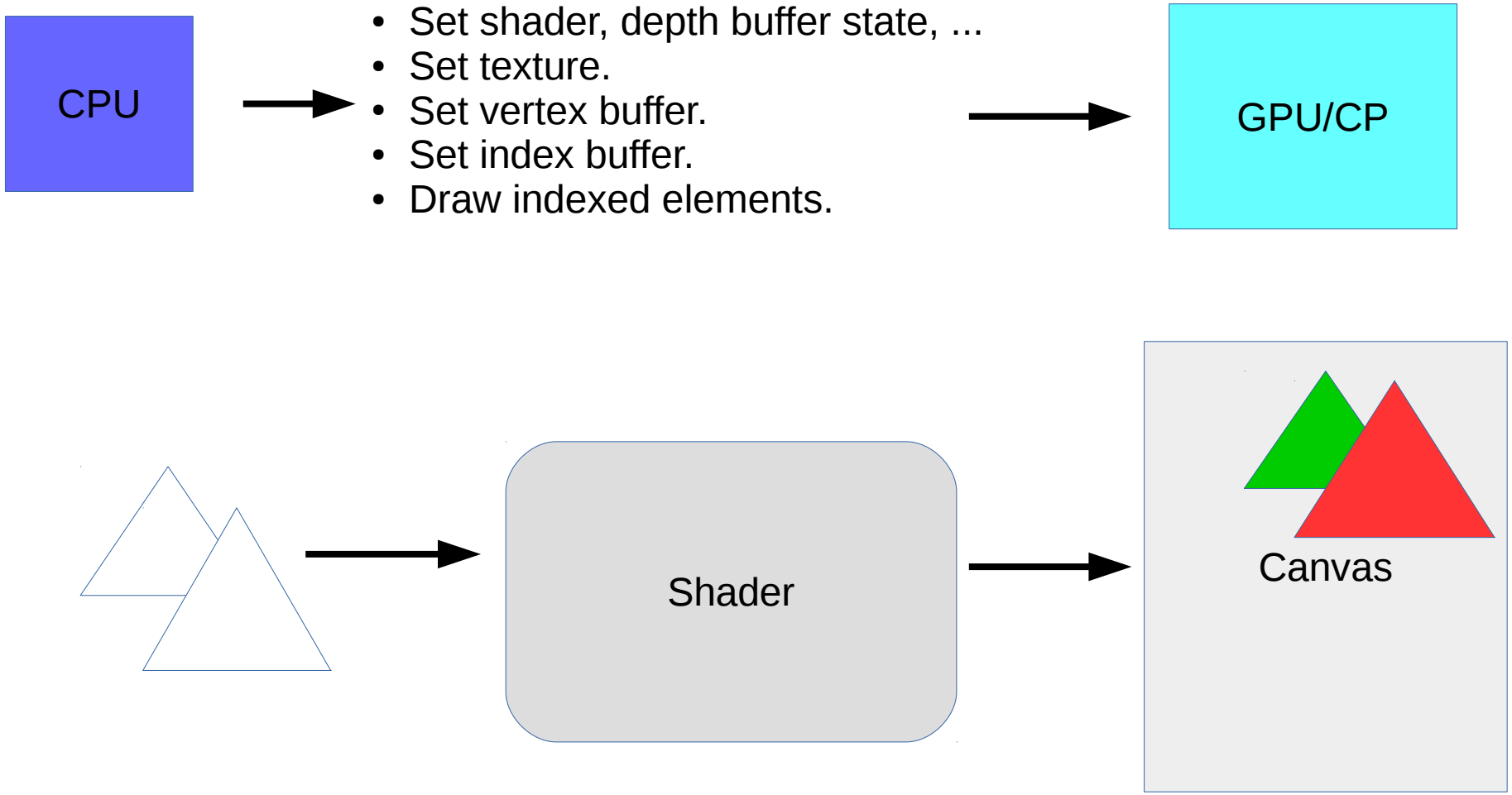
Why rewriting Woden 2?

- OpenGL is **very** complex.
- OpenGL drivers are **very** buggy!
  - NVIDIA does not follow strictly the spec.
  - AMD is very strict.
  - Intel, OS specific behavior.
- OpenGL uses a thread local context.
- OpenGL is old.
- GLSL compiler is in the driver.

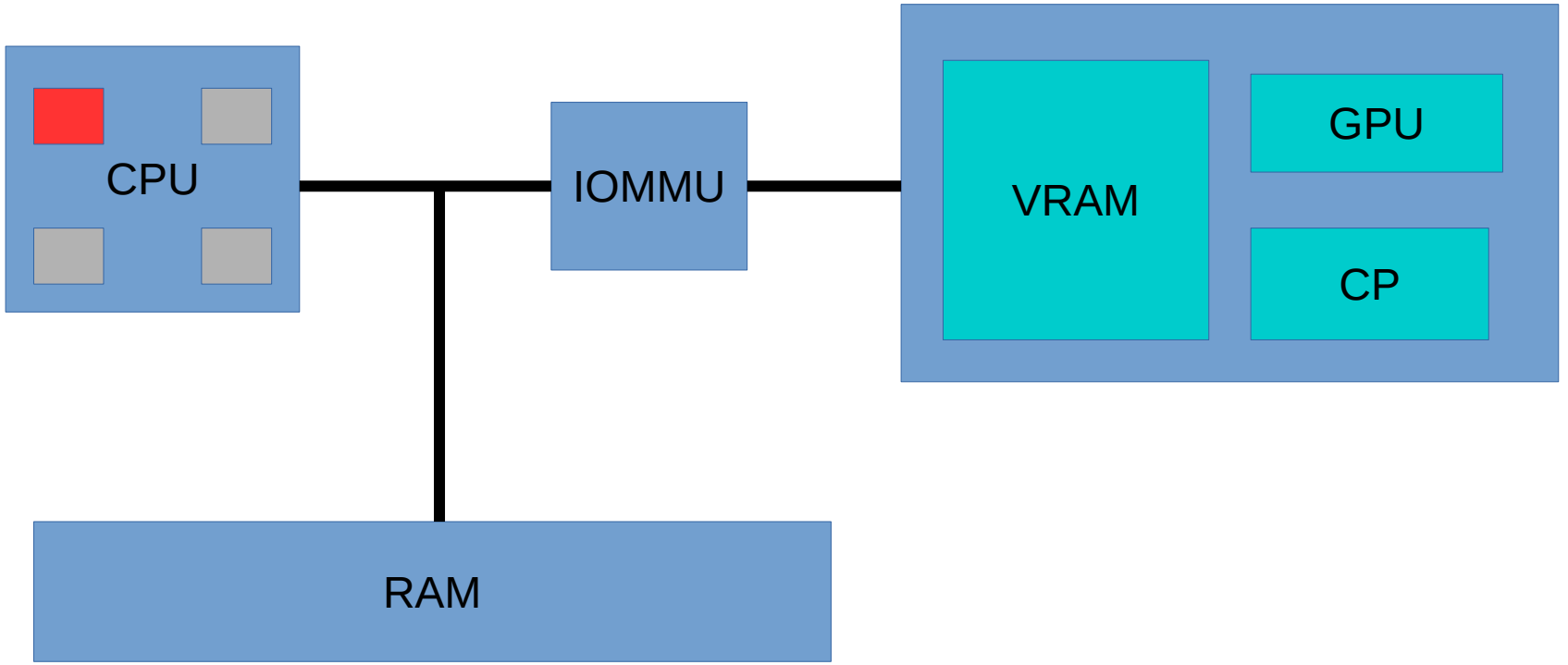
Choose a desktop OpenGL version

OpenGL	GLSL
1.2	Not available
2.0	1.10
2.1	1.20
3.0	1.30
3.1	1.40
3.2	1.50
3.3	3.30
4.0	4.00
4.1	4.10
4.2	4.30
4.4	4.40
4.5	4.50

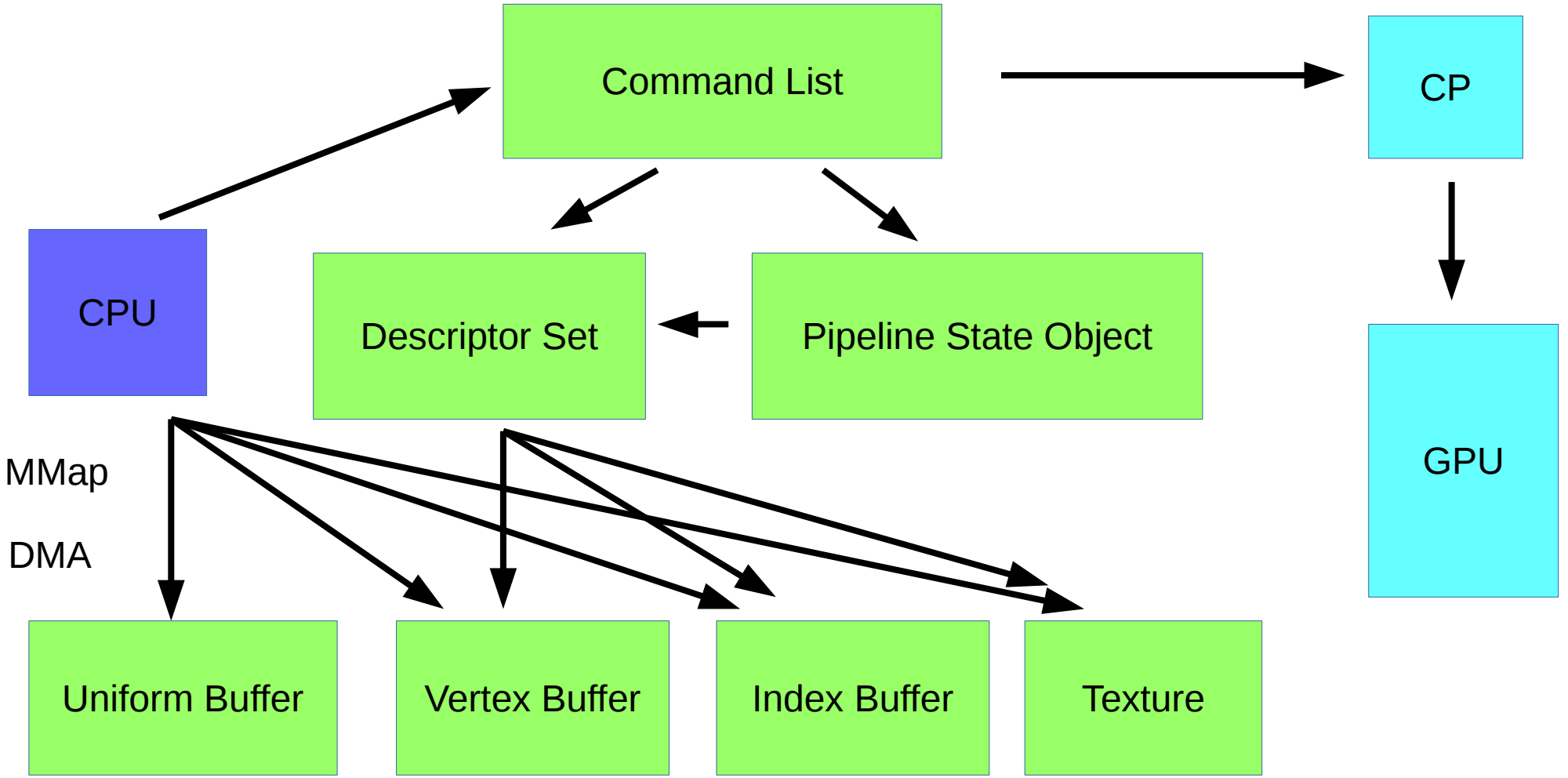
Classic graphics API



“Modern” PC Architecture



Modern Low-Level Graphics API



Core Abstraction Layers

- AbstractGPU
  - https://github.com/ronsaldo/abstract-gpu
- Dastrel (Data Stream Language)
  - http://smalltalkhub.com/#!/~ronsaldo/Dastrel
- Slovim (LLVM inspired typed SSA form)
  - http://smalltalkhub.com/#!/~ronsaldo/Slovim

AbstractGPU

- Abstraction above Vulkan, Metal and D3D 12.
- API in C inspired by OpenCL
- Implementation in C++
- XML spec for headers and bindings
- Mostly tested with Vulkan in Linux

Dastrel (Data Stream Language)

- Intended to be a semi-visual language
- Statically typed
- Syntax inspired by C, Rust, Swift and Smalltalk
- Dumb type inference
- Compiler front-end written in Pharo
- Emits Slovim SSA form

Dastrel Sample

```
import environment;
import fragment;
import material;
import lighting;

code_block(fragment) main
{
    let N = normalize(FragmentInput.normal);
    let V = normalize(-FragmentInput.position);

    color: FragmentOutput.color <== forwardLightingModel
        albedo: FragmentInput.color*MaterialState.albedo
        fresnel: MaterialState.fresnel smoothness: MaterialState.smoothness
        normal: N viewVector: V position: FragmentInput.position;
}
```

Dastrel Sample

```
struct ObjectStateData
{
    matrix: float4x4;
    inverseMatrix: float4x4;
    color: float4;
    visible: int;
}

uniform(set=0, binding=0) ObjectState
{
    objectState: ObjectStateData;
}

buffer(set=0, binding=1) InstanceObjectState
{
    instanceStates: ObjectStateData[];
}

uniform(set=1, binding=0) CameraState
{
    inverseViewMatrix: float4x4;
    viewMatrix: float4x4;

    projectionMatrix: float4x4;
    currentTime: float;
}
```

Dastrel

```
function transformPositionToWorld(position: float3) -> float4
{
    using ObjectState;
    using InstanceObjectState;
    using VertexStage;

    return objectState.matrix *
(instanceStates[instanceID].matrix * float4(position, 1.0f));
}
```

Slovim

- LLVM style SSA form in Pharo.
- Strongly typed.
- Some very basics optimizations.
- Backends for:
  - Spir-V (Done)
  - GLSL (Done for Vulkan, missing OpenGL)
  - C++ (For testing/experimenting)
  - HLSL (To be done)
  - Metal Shading Language (To be done)

WodenMath

- 3D graphics linear algebra
  - Vectors (2D, 3D, 4D)
  - Complex
  - Quaternions
  - Matrices (3x3 and 4x4)
- Same memory layout as C/GPU structures.
- Optimized using Lowcode extended bytecodes.

WodenMath

- 3D graphics linear algebra
  - Vectors (2D, 3D, 4D)
  - Complex
  - Quaternions
  - Matrices (3x3 and 4x4)
- Same memory layout as C/GPU structures.
- Optimized using Lowcode extended bytecodes.

WodenMath

```
NativeStructure subclass: #WMVector3F
    layout: StructureLayout
    slots: { #x &=> #float.
             #y &=> #float.
             #z &=> #float.
             #padw &=> #float }
    classVariables: {  }
    category: 'WodenMath-Core-LinearAlgebra'
```

```
+ other
    <argument: #other type: #(SelfType object)>
    <returnType: #(SelfType object)>
    ^ self class x: x + other x y: y + other y z: z + other z
```

Woden 2 Core

- WTEngine is the main entry point
- WTPipelineStateCache loads PSO from JSON
- WTResourceCache for loading files
- WTApplication provides support for Morphic and OSWindow
- They are wrappers above the AbstractGPU object for supporting image session saving and restoring

Woden 2 Core Utilities

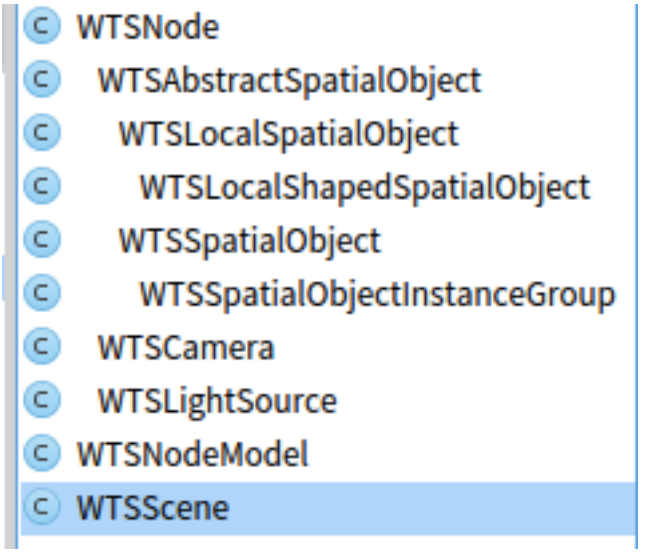
- Custom bitmap font format
- Mesh building
- Some GPU structures matching shader definitions
  - WTObjectState
  - WTCameraState
  - WTLightSourceData
- 3D Model with skeletal animation loading

How do I get Woden 2

- https://github.com/ronsaldo/woden2
- newImage.sh builds dependencies and creates a Woden 2 image
- Warning: this currently requires a modified VM at https://github.com/ronsaldo/pharo-vm-lowcode

Woden 2 Scene Graph

- Simpler than the scene graph in Woden 1.
- Only rendering a 3D scene (except for WTSNodeModel)



WTFPSSampleApplicationCube

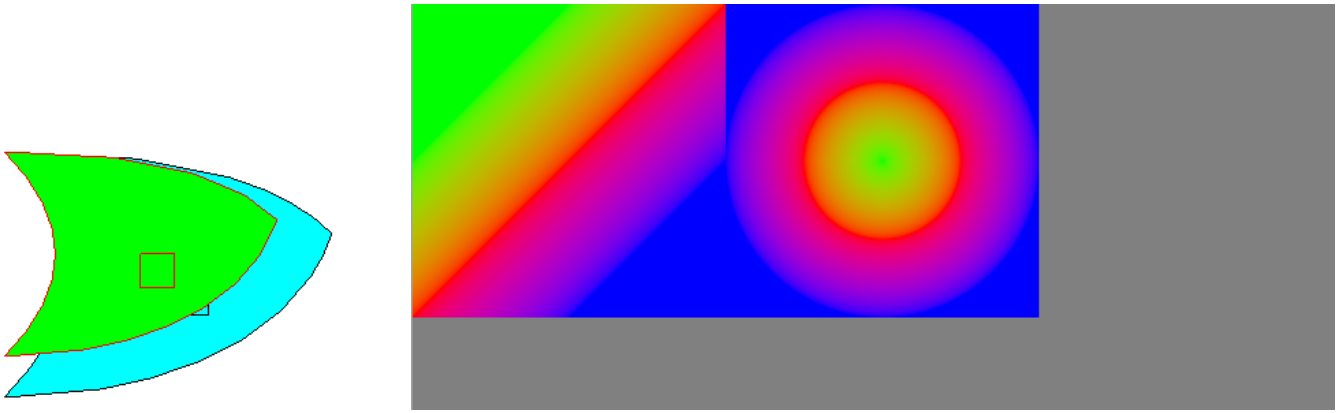
```
initializeSceneContent
   | meshBuilder mesh cube light |
   camera transform translateByZ: 3.0.

   meshBuilder := WTGenericMeshBuilder for: engine.
   meshBuilder addCubeWithWidth: 1.0 height: 1.0 depth: 1.0.
   mesh := meshBuilder mesh.

   cube := WTSSpatialObject new.
   cube renderable: mesh.
   scene add: cube.

   light := WTSLightSource new.
   light point.
   light state
      intensity: Color yellow asWMVector4F.
   light transform
      translateByX: -1.5 y: 1.5 z: 1.6.
   scene add: light.
```

3D Scene Rendering

- Standard forward rendering.
- WTSForwardSceneRenderer
- Command lists are reused if the scene topology does ot change
- Shadows and a deferred shading renderer are not yet implemented

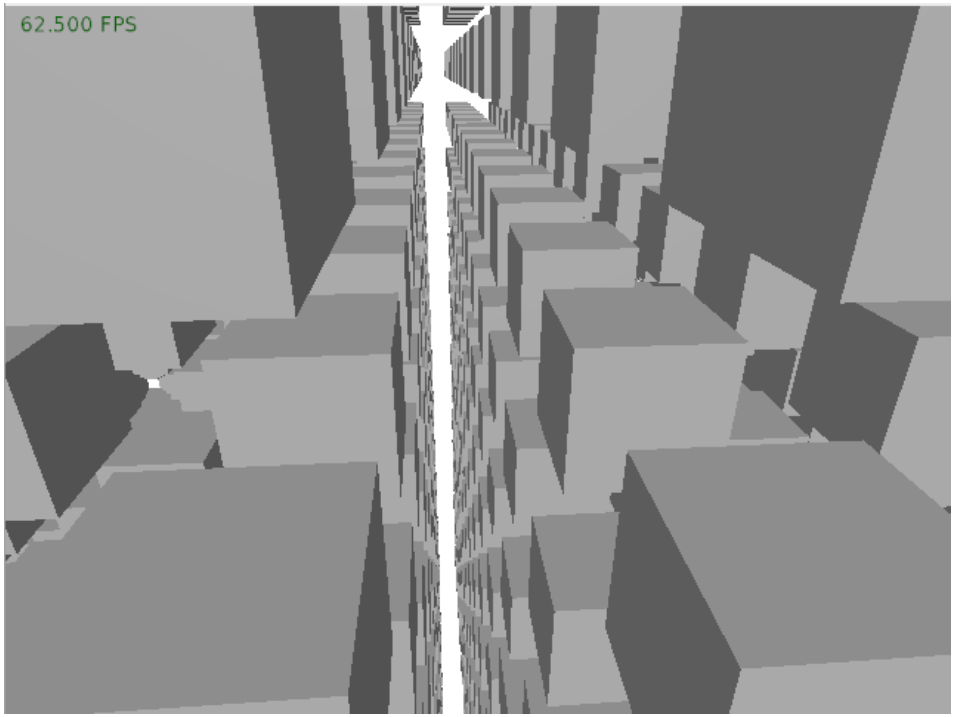
Woden 2 Athens Backend

Hello Athens! 62.500 FPS



Woden Roassal

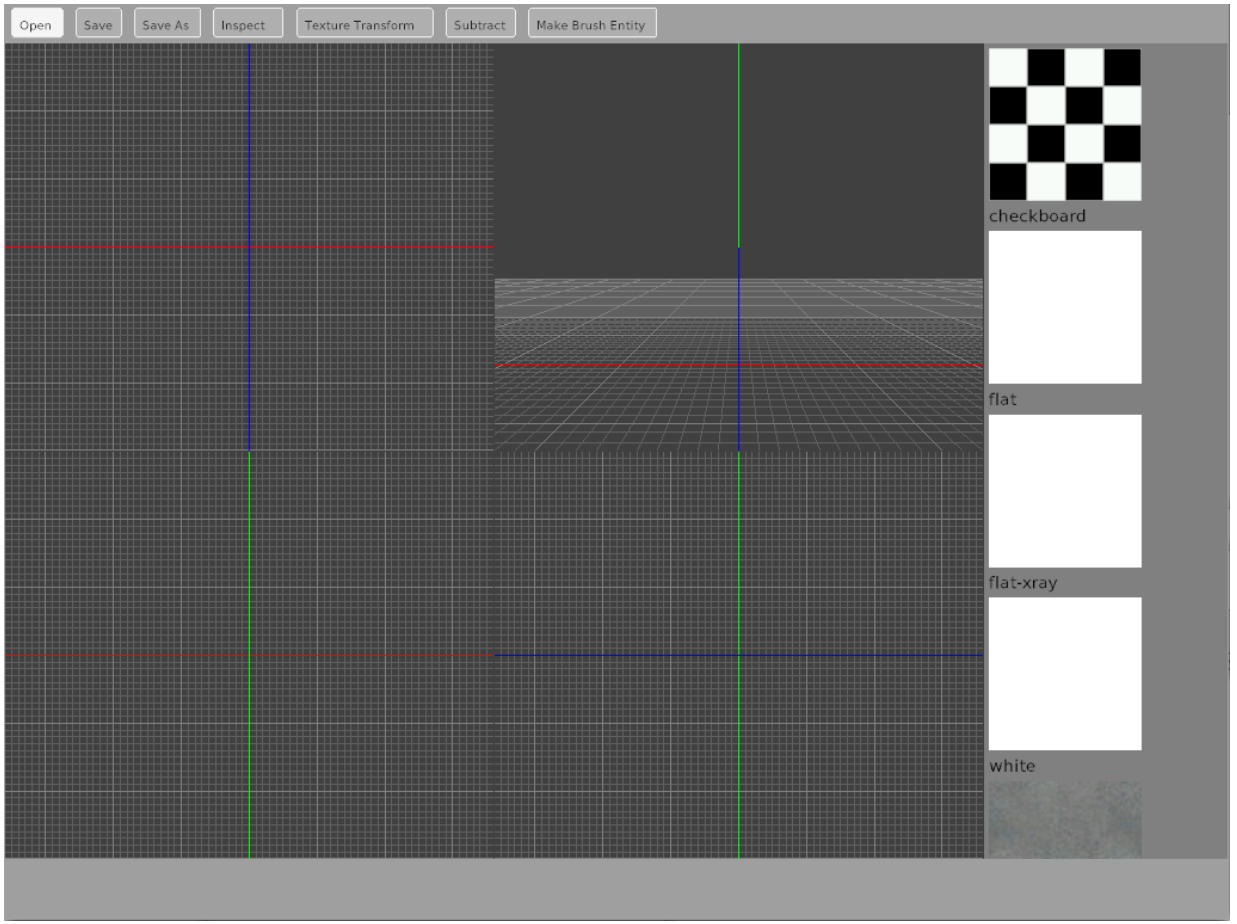
```
v := RWView new.
1 to: 20000 do: [ :i |
   v add: RWCube element.
].

v camera translateByZ: 3.0.
RWCubeLayout on: v elements.

v addInteraction: RWMouseKeyControl.
v open
```



Game System

Level Editor

- Bloc over Woden 2



Where is Woden 2?

- https://github.com/ronsaldo/woden2
- But, it requires a modified Pharo VM
- https://github.com/ronsaldo/pharo-vm-lowcode