CS354P

DR SARAH ABRAHAM

OVERVIEW: GRAPHICS

WHAT IS GRAPHICS?

- Broad area that includes anything involved in the process of getting pictures onto a screen
  - Rendering pipeline
  - Physical simulation
  - Procedural generation
  - Animation
  - Geometry and modelings
  - etc...

WE'LL FOCUS ON THE RENDERING FEATURES

- This will be as high-level as possible, since we won't have time to cover the actual math/hardware in any detail
  - We'll come back to some of these features when we talk more about the GPU pipeline

GRAPHICS PIPELINE OVERVIEW

- CPU (Central Processing Unit) passes functionality and data to the GPU (Graphics Processing Unit)
- GPU architecture designed for *throughput*
  - High bandwidth, high latency
  - Goal is to process many similar operations in a parallel manner (i.e. efficiently apply mathematical operations to scene data)
- Considerations:
  - What data does the GPU need?
  - How do we get it to the GPU?
  - How do we specify what the GPU should do?

GRAPHICS LIBRARIES

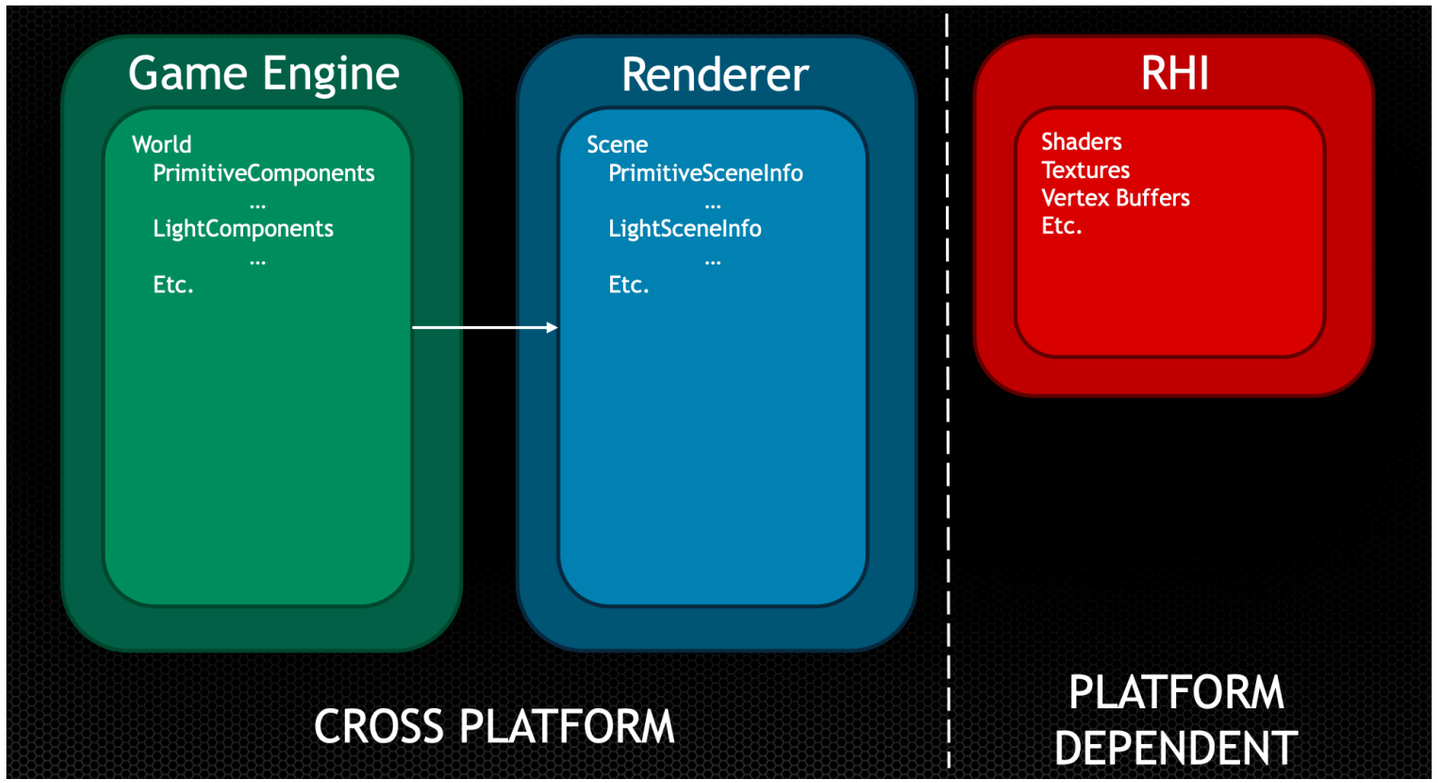
- Provide APIs for communicating data between the CPU and GPU
- OpenGL is a higher-level library created by the Khronos Group
  - Performs more of the setup and makes assumptions about memory to simplify developer interactions
    - OpenGLES is graphics library for embedded systems such as mobile devices and web applications
- Vulkan is a lower-level library created by the Khronos Group
  - Allows greater flexibility and developer control by having developers perform setup and determine things like memory management/thread management
- DirectX is the family of libraries created by Microsoft
  - DirectX12 is equivalent to Vulkan in most functionality
- Metal is graphics library created by Apple and Sony has their own library as well…

HOW DOES THESE RELATE TO THE GRAPHICS HARDWARE?

- Graphics hardware has *API* **specifications** that these graphics libraries adhere to
  - Graphics libraries supported in hardware via drivers
- The choices that graphics libraries make effect their support by drivers:
  - OpenGL has tremendous backwards compatibility and support, and this complexity effects its performance
  - DirectX11 has similar issues but also more hand-optimized due to marketshare
  - DirectX12 and Vulkan are in the process of replacing OpenGL/DirectX11 in high-end games

UNREAL: SUPPORTING MULTIPLE HARDWARES

- Rendering Hardware Interface (RHI) is a C++ interface to allow communication from UE5's rendering code to platform-dependent implementations of graphics APIs
- Also use of an internal **shader** cross compiler (HLSLCC)



WHAT ARE SHADERS?

- Small programs that run on GPU hardware
- GPUs have **programmable** pipelines which allow these compiled programs to be linked to pipeline stages and dictate how data passed from the CPU is processed
  - Apply transforms to vertex data
  - Use texture information
  - Apply post-processing effects
  - etc...
- Final output is an image buffer with each pixel “shaded” accordingly

AT LAST...THE PRETTY STUFF...

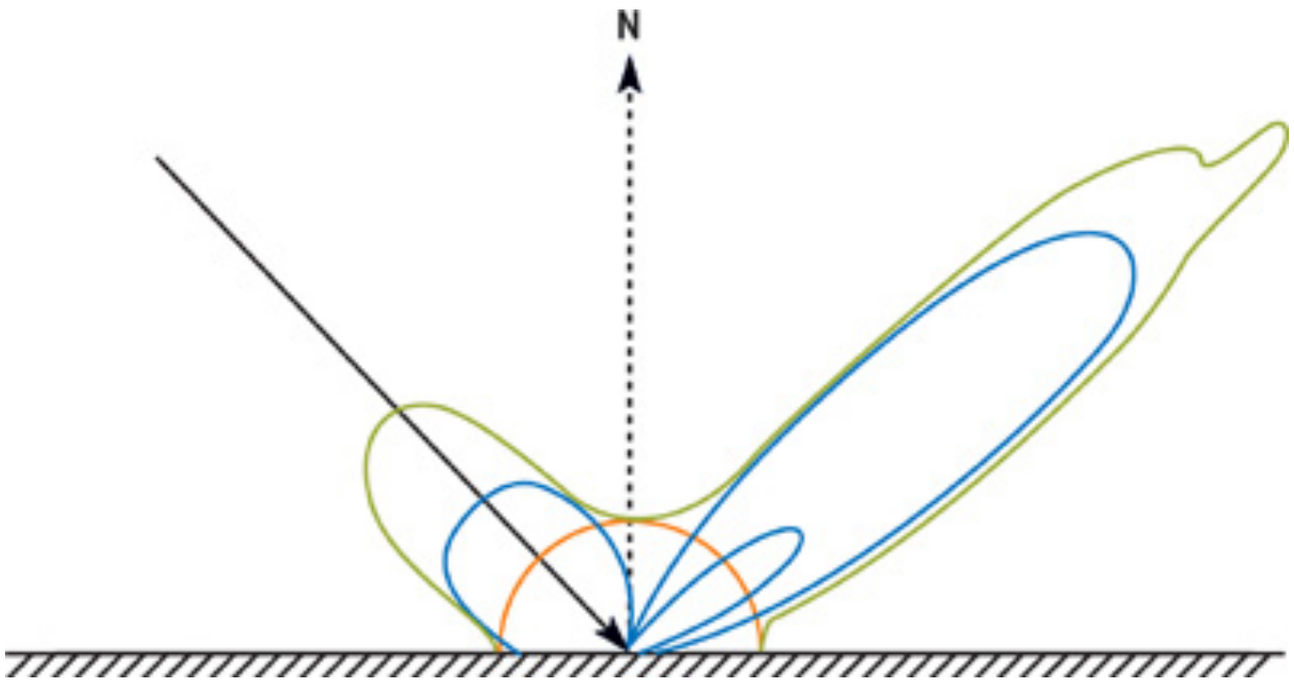
- Shaders are where we specify things like lighting models, texture mapping, material interactions and more
  - i.e. they make things pretty

Guilty Gear Strive

Backbone

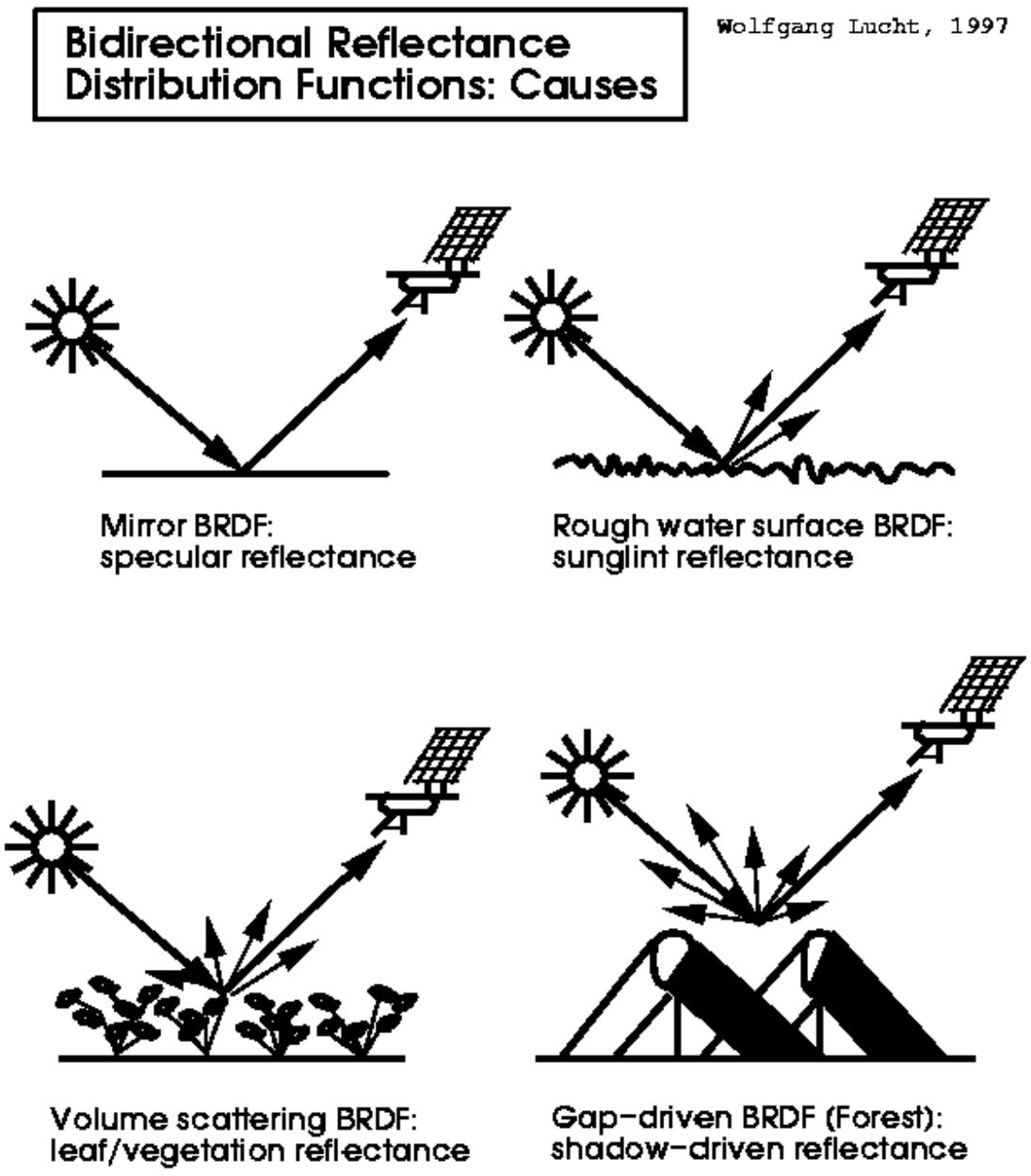
MATERIALS AND PHYSICALLY-BASED RENDERING (PBR)

- Concept of the visual qualities a mesh object has
  - Textures are part of this but called **materials** because they represent the actual material properties in relation to the **lighting equation**
- Take incoming light data and apply it to the physically-based lighting function of the material to determine the final pixel color output



BRDFS

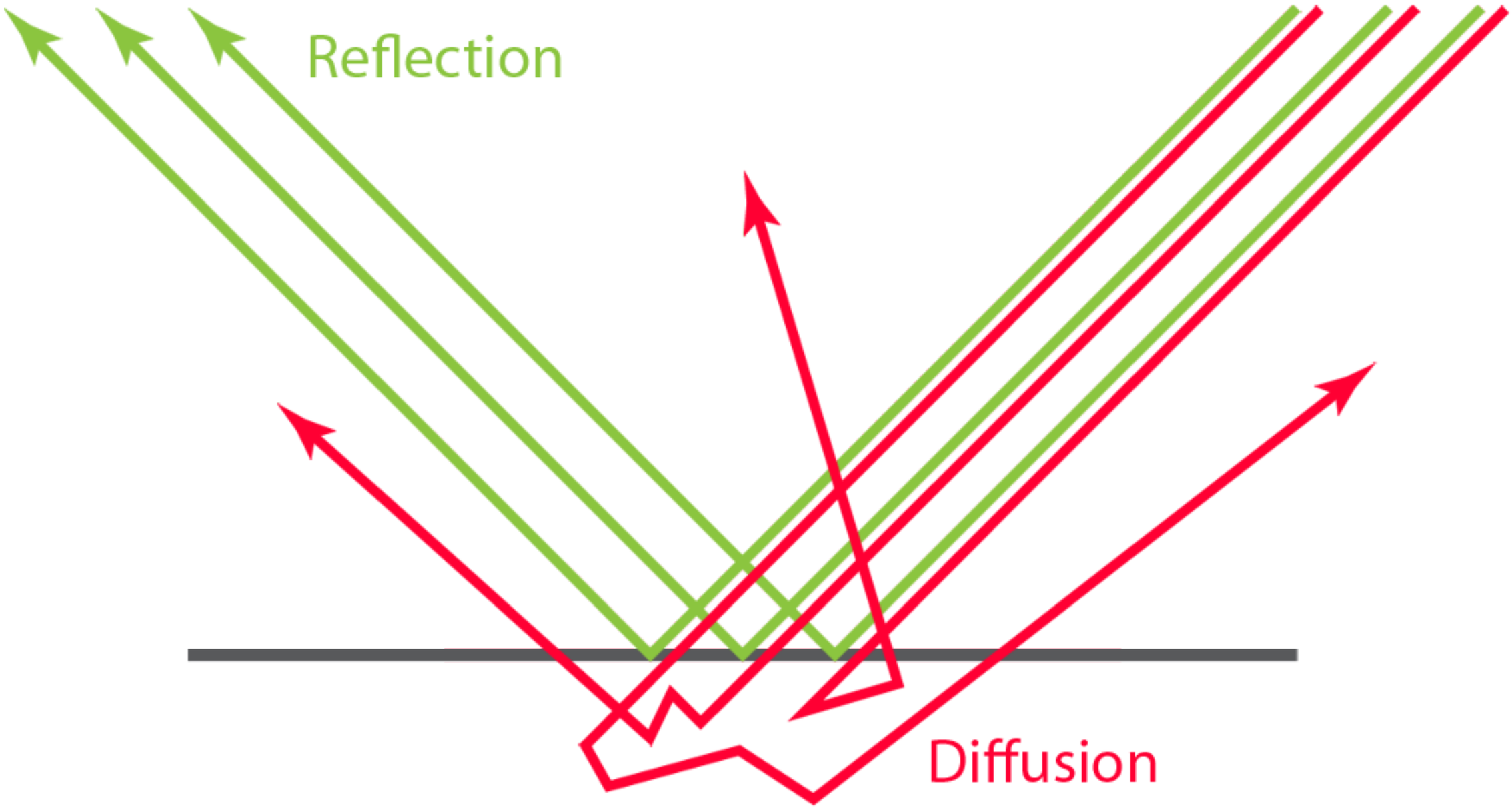
- Bidirectional reflectance distribution function
- Defines how a material reflects light based on the angle of observation
- Determines ratio of reflected radiance
  - Physically-based
  - Empirically studied by material sample



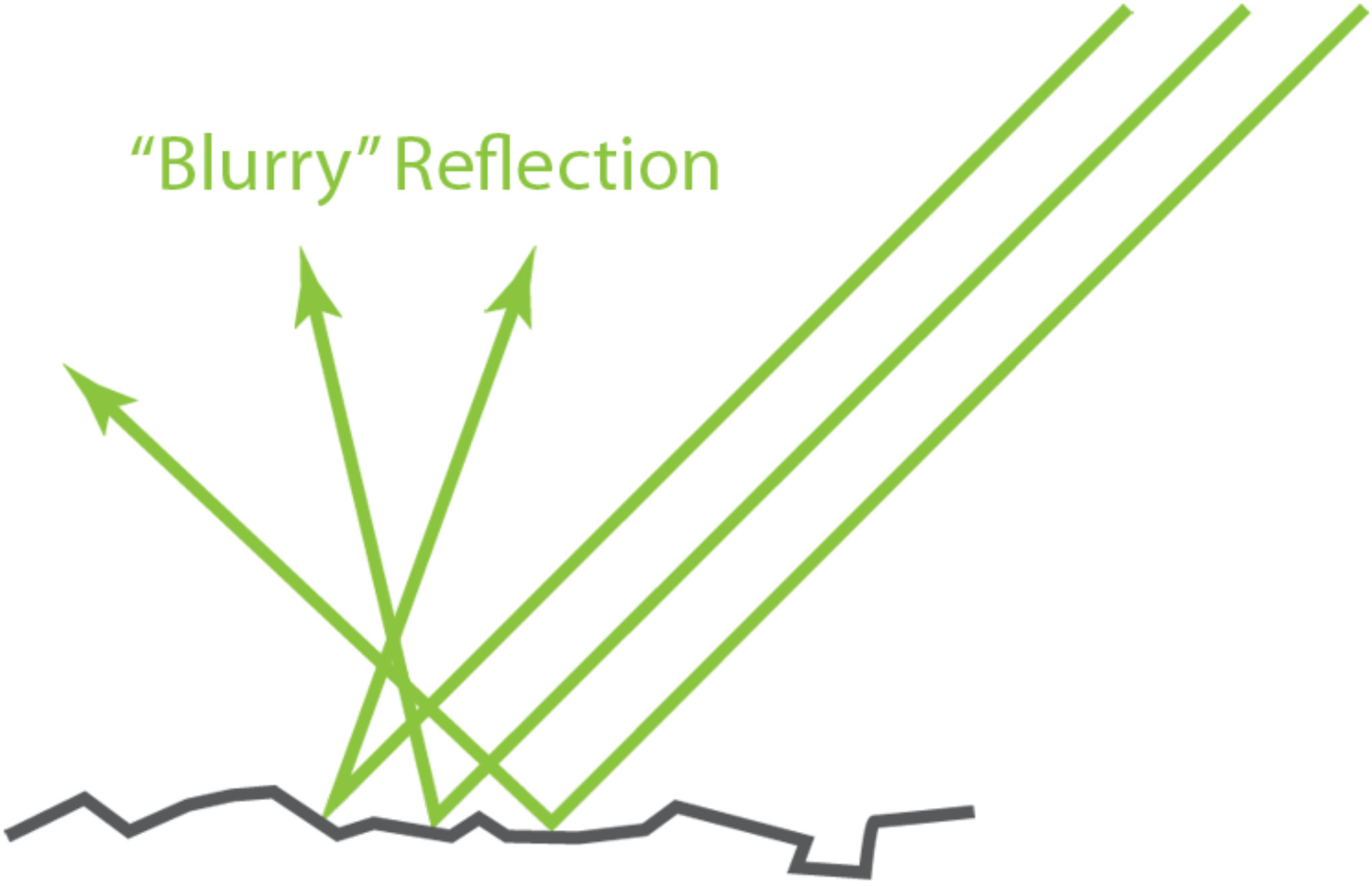
MATERIAL PARAMETERIZATION

- Base Color (Albedo)
  - Diffuse color based on scattering/absorption of light wavelengths
- Roughness
  - Amount of microsurfaces and imperfections on material's surface leading to light scatter
- Metallic
  - Degree of "metalness" including colored reflections and any diffusion from corrosion/dirt on surface
- Reflectance
  - Amount of reflected light on non-metallic surfaces

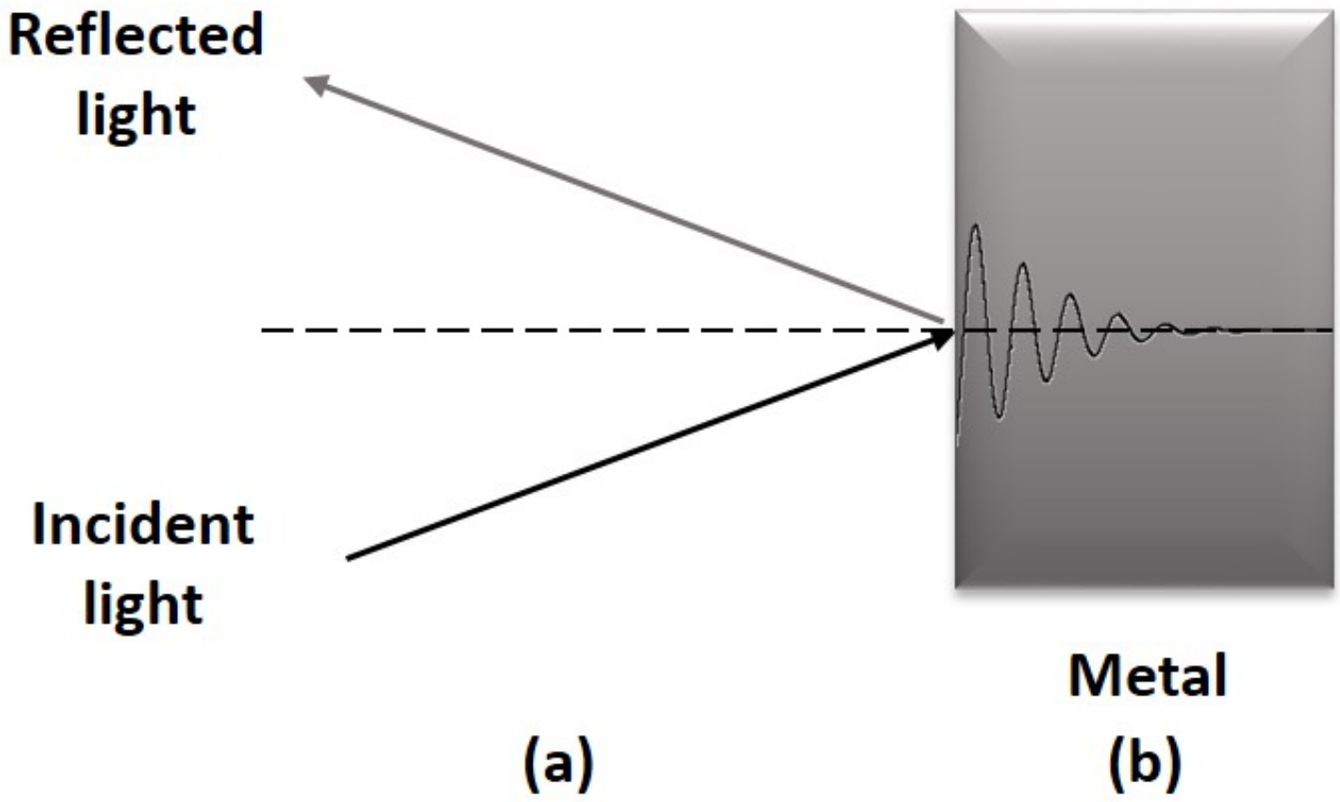
ALBEDO

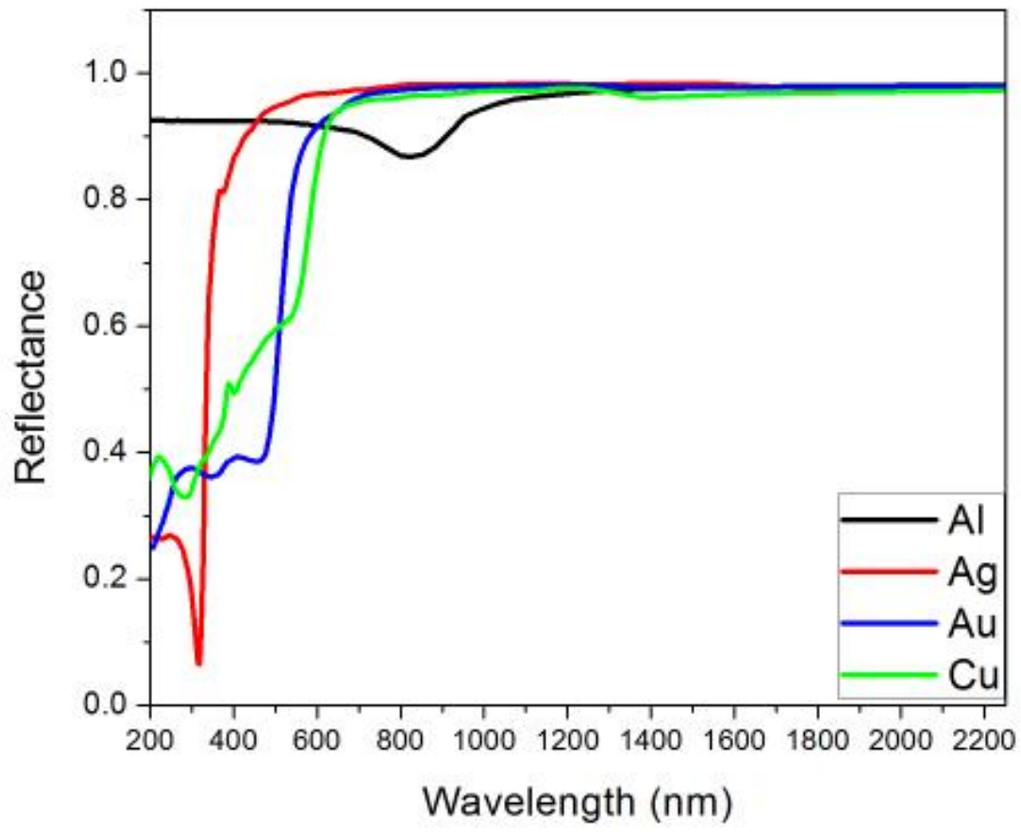


ROUGHNESS

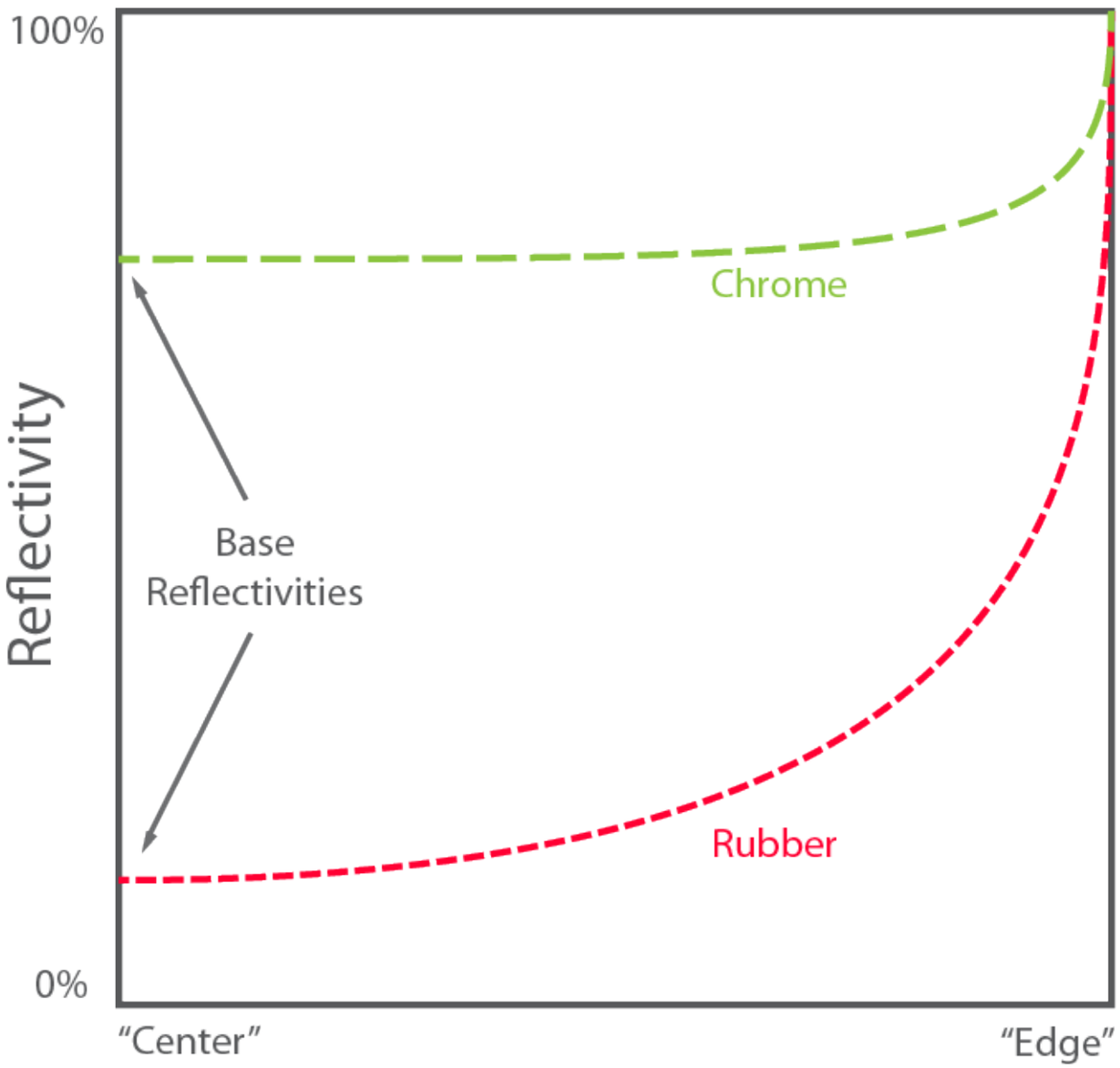


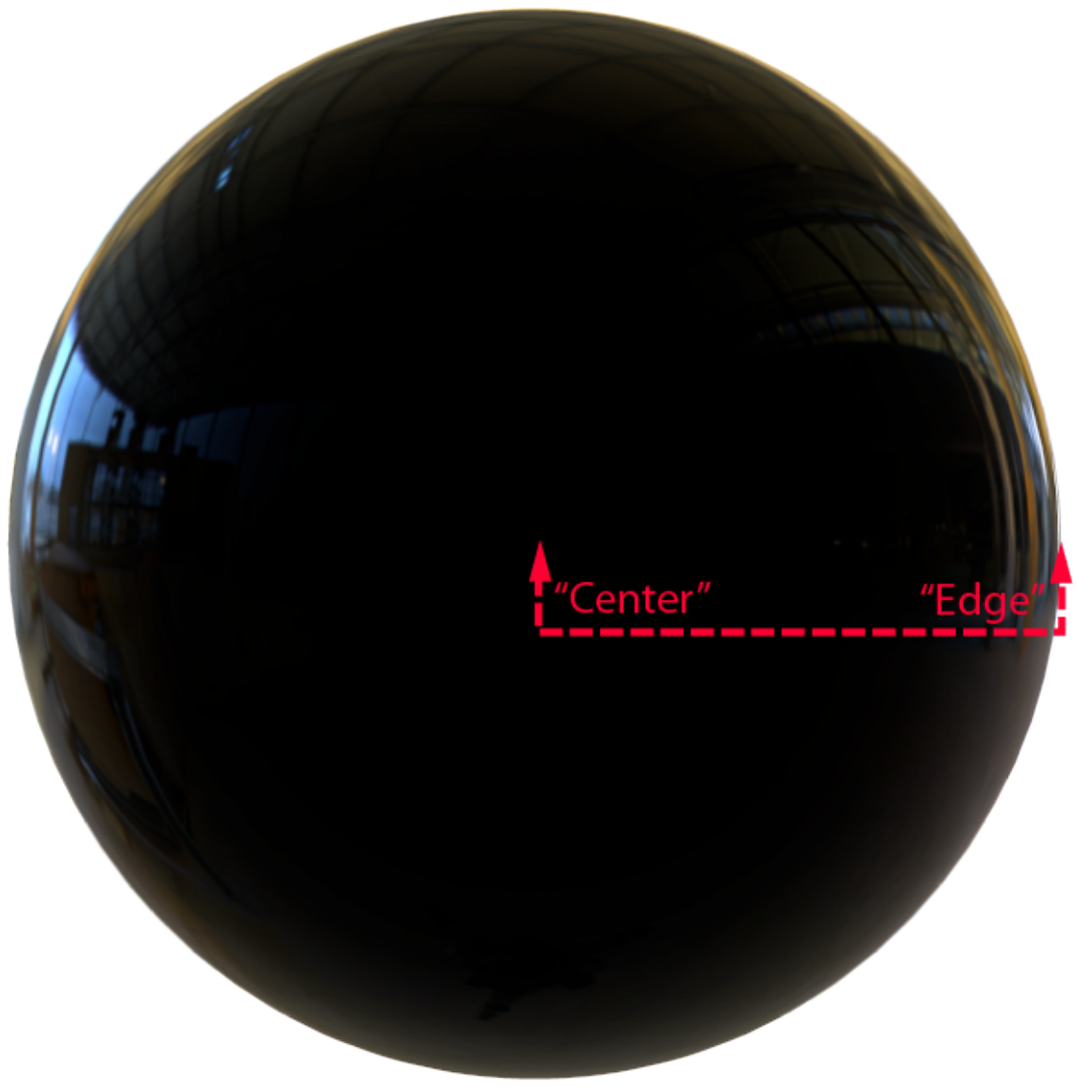
METALLIC


Reflected light
Incident light
Metal
(a)
(b)


Reflectance
Wavelength (nm)
Al
Ag
Au
Cu
0.0
0.2
0.4
0.6
0.8
1.0
200
400
600
800
1000
1200
1400
1600
1800
2000
2200

REFLECTANCE



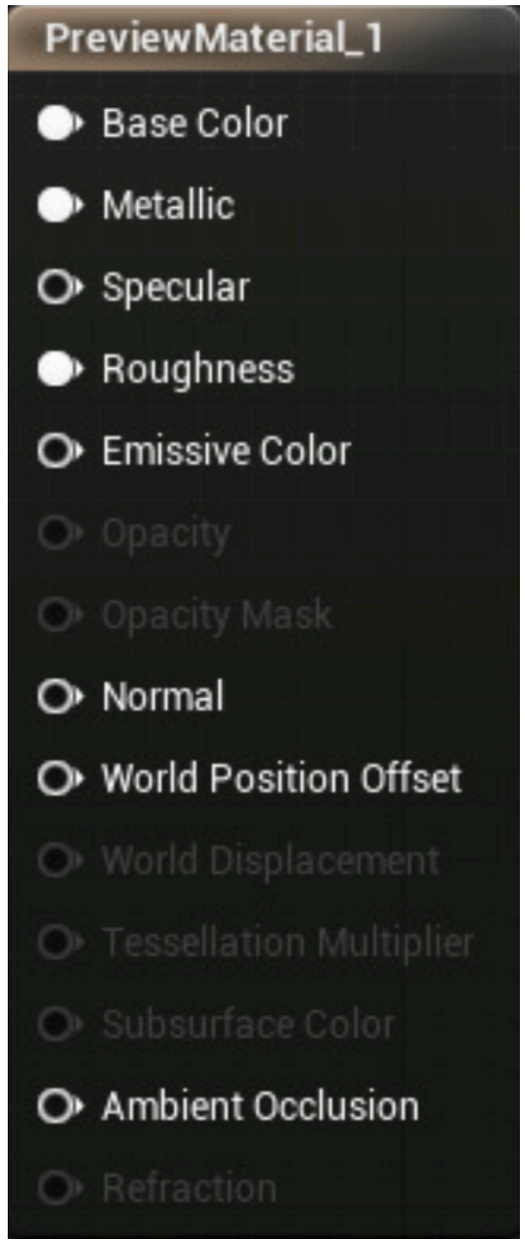


MATERIALS IN UNREAL

- Assets that can be applied to meshes to control the mesh's lighting properties
- Uses a node-based scripting language that connects to the underlying shader programming language (in this case, HLSL)
  - Allows artists to create visual effects without any shader programming knowledge
  - Possible to access HLSL directly but not required in many cases

MATERIAL PROPERTIES AND INPUTS

- Material properties specify things like blend mode, shading model, level of detail, translucency, and shader pipeline optimizations among others
- Material inputs specify the material parameterization discussed earlier
  - Can connect to art programs like Substance, which specialize in generating procedural, PBR-based textures and materials

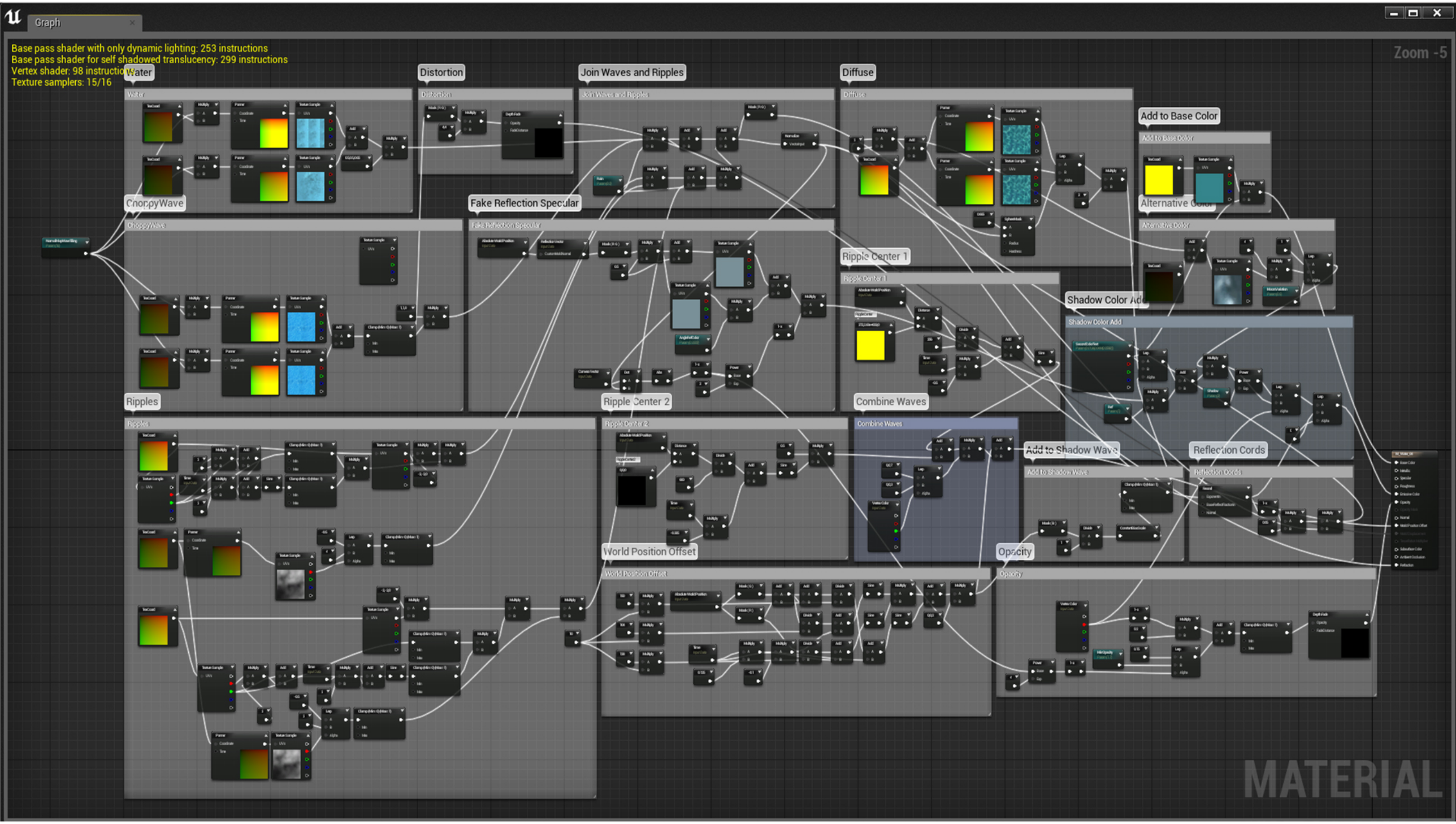


Material Inputs

PUTTING IT ALL TOGETHER...

- Can create very simple to very complex effects...

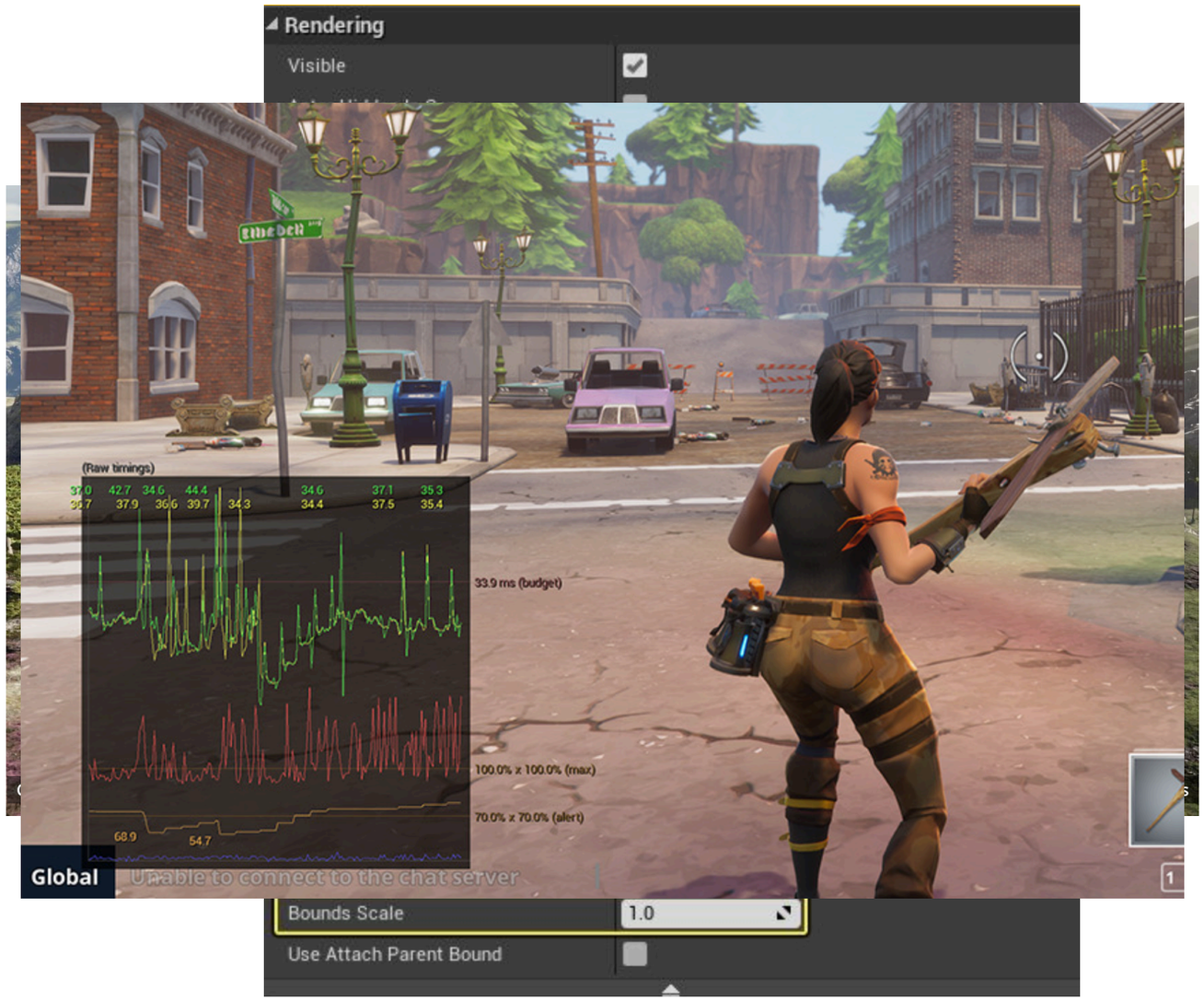
https://forums.unrealengine.com/community/work-in-progress/7372-water-material



https://forums.unrealengine.com/community/work-in-progress/7372-water-material

BUILT-IN EFFECTS

- Unreal has a ton of beautiful effects/features you can use “out of the box”
- Sky Atmospheres create physically-based sky and atmospheric rendering with time of day
- Multiple types of visibility culling plus per-instance settings
- Many, many pre-baked and dynamic lighting setups
- Dynamic resolution support for adjusting resolution per frame



POST-PROCESSING EFFECTS

- Effects done at the end of the shading pipeline to apply visual changes globally to the scene
  - Unreal uses Post-Process Volumes that apply effect within that volume
- Effects include:
  - Anti-aliasing
  - Bloom
  - Depth of Field
  - Lens Flare
  - Chromatic Aberration
  - Vignette

POST PROCESS MATERIALS

- Can also apply Post Process Materials, which are shaders that work in the scene's texture space*

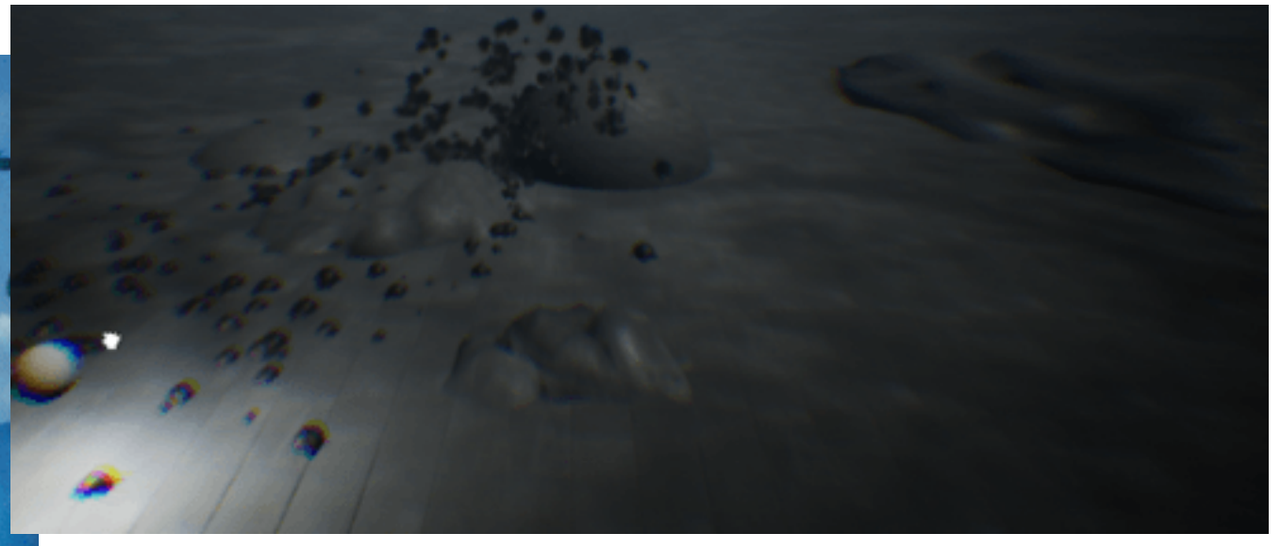
Some post-process material examples

* Note to students who have taken graphics: I'm differentiating texture and screen space because Unreal assumes a **deferred shading pipeline** (which we'll touch on later) but you can think of this as a fragment shader

PARTICLE SYSTEMS

- Rules and memory management for a large body of point masses to create visual effects
  - Creation of fluid effects
  - Creation of crowd behaviors/flocking
  - etc..
- UE5 has two particle systems:
  - Cascade is older, better documented system with less flexibility
  - Niagara is newer, less documented system with greater flexibility
- Cascade and Niagara both designed for designer/artist use
  - Niagara is more “next-gen” allowing designers/artists to create more lower-level functionality with programmer assistance

PARTICLE EFFECTS IN ACTION



Created by Ashif Ali in Niagara (https://cghow.com/members/asif786ali/)

RAY TRACING

- Technique that emulates the physical equations of **light transport** to get an accurate representation of light-material interaction
- Increasingly common in modern systems with growing hardware support
- Unreal supports two kinds of ray tracing
  - Path tracing (offline, expensive form of raytracing to correctly emulate light transport)
  - Hybrid ray tracing (real-time form of raytracing that is used in tandem with “raster” style effects)

HYBRID RAYTRACING EXAMPLE: ARCHITECTURE STUDIOS

https://www.youtube.com/watch?v=YSZnX6P7-MM

FURTHER READING

- NVIDIA Bringing Unreal Engine 4 to OpenGL [https://de45xmedrsdbp.cloudfront.net/Resources/files/UE4_OpenGL4_GDC2014-514746542.pdf]
- OpenGL vs DirectX -- what really happened? [https://www.back2gaming.com/reviews/b2g-games/pc/opengl-vs-directx-what-really-happened/]
- UE4 Rendering and Graphics [https://docs.unrealengine.com/en-US/Engine/Rendering/index.html]