(Raceca)r (R)enderer
Milestone 1

CIS 5650, Team 6
Anthony Ge, Charles Wang, Saahil Gupta, Aaron Jiang
https://github.com/upgrade-central-tech/racecar

Premise

- Cars are cool. They’re dynamic, colorful, visually complex, and games like Gran Turismo 7, F1 25, and Forza do it really well.

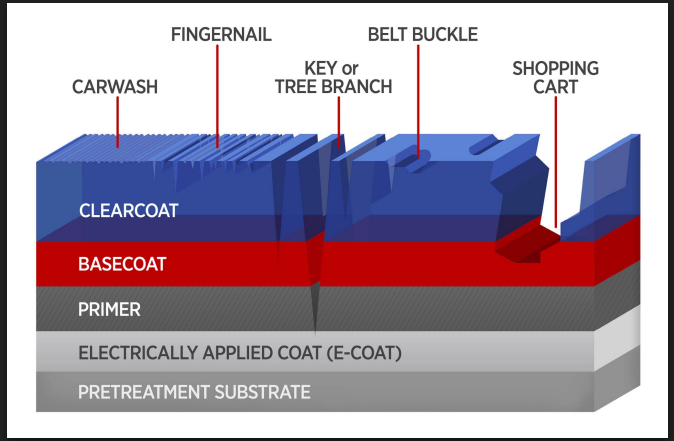
Photo of a McLaren Senna

F1 25 screenshot

Premise

Cars look good because of

- Reflections from the environment and strong lighting from atmospheres and skies
- Diverse and complex car paints and finishes



Focuses

Based on these reasons, we wanted to make a renderer that's interactable with:

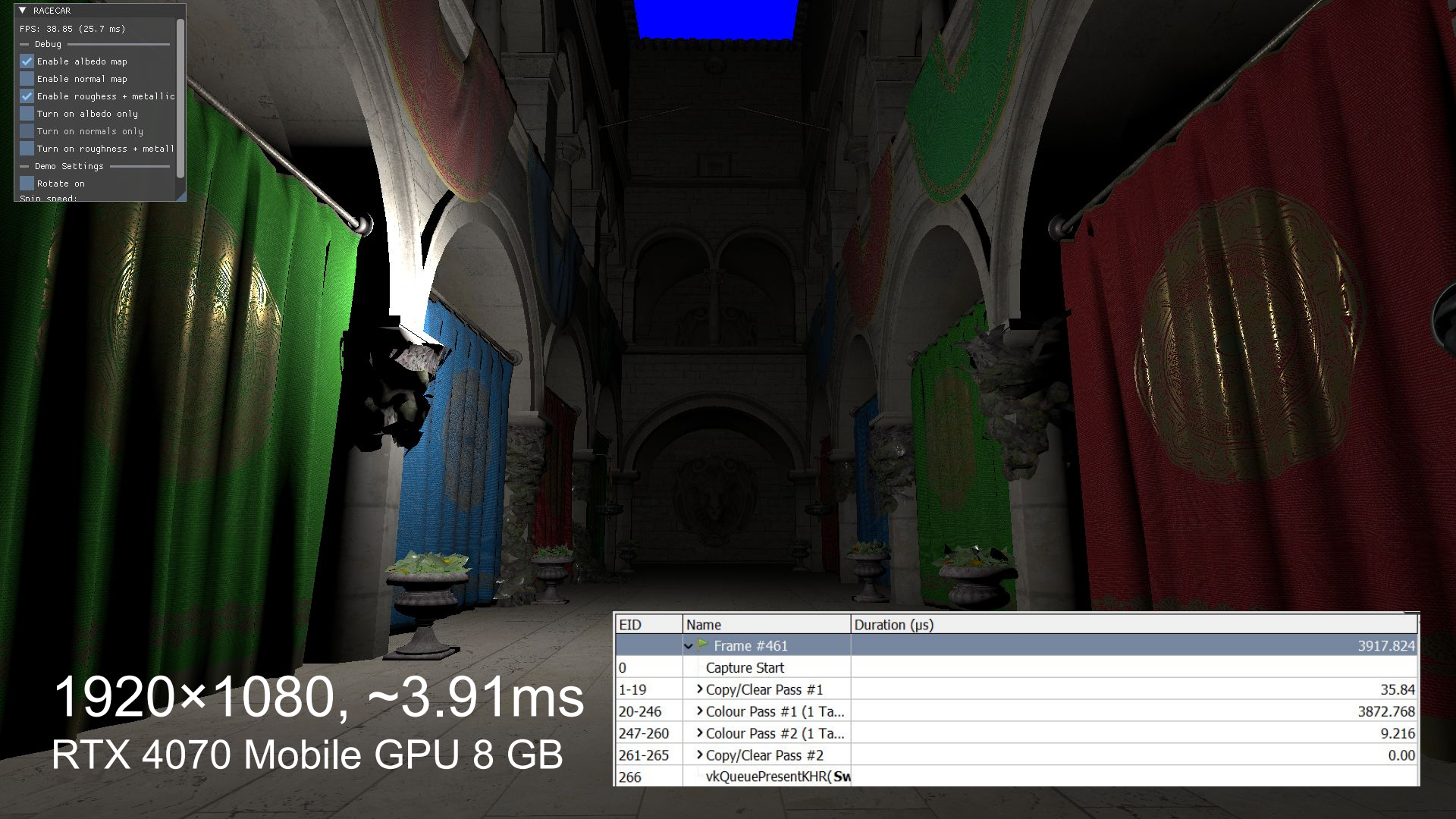
- **Car materials** (complex layered model)
- **Real-time reflections and lighting**
- **Dynamic skies/weather:** dusk skies, cloudy skies, rainy conditions
- **Environments:** high detail geo rendering (snow, mud, rocks, tracks, grass)

Milestone 1

To that end, we spent most of our milestone 1 developing the base engine.

- From scratch using C++20 and Vulkan
- glTF model, material, and texture loading (loads Sponza)
- Basic PBR material setup with GGX BRDF
- Other nice-to-haves: ImGui, orbit camera, task system, Slang compilation

Next we'll discuss some of the features and their implementations in more detail, and show images from our rasterizer.



EID	Name	Duration (µs)
	Frame #461	3917.824
0	Capture Start	
1-19	Copy/Clear Pass #1	35.84
20-246	Colour Pass #1 (1 Ta...	3872.768
247-260	Colour Pass #2 (1 Ta...	9.216
261-265	Copy/Clear Pass #2	0.00
266	vkQueuePresentKHR(**Sw**	

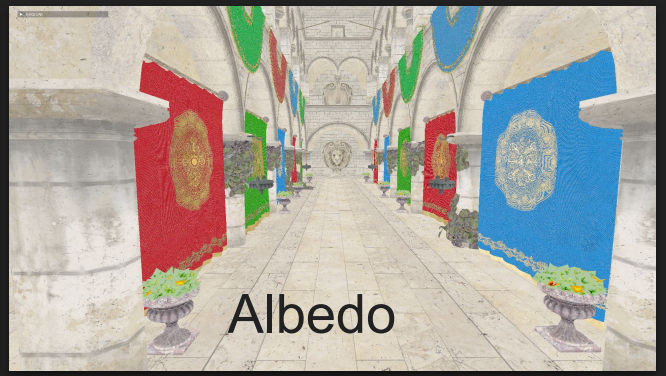
1920×1080, ~3.91ms

RTX 4070 Mobile GPU 8 GB

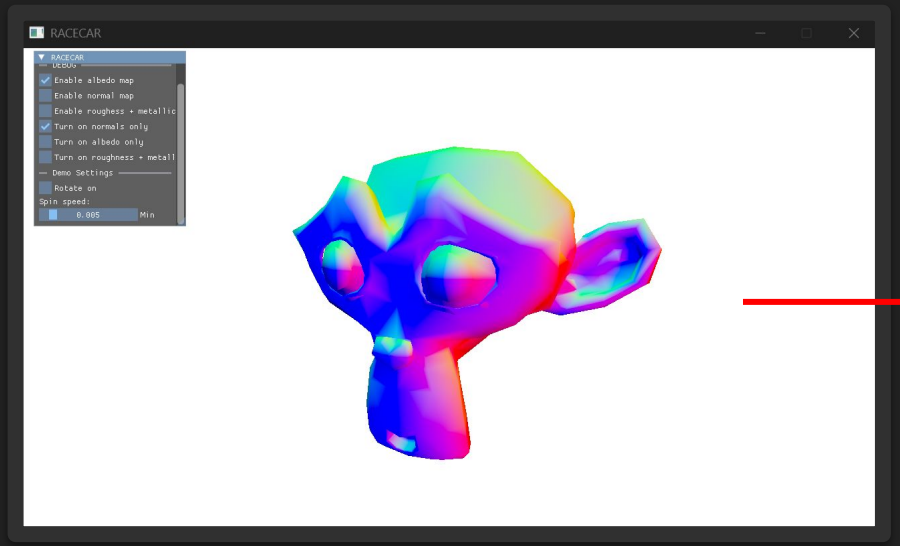

RACECAR
1920×1080, ~1430 FPS (0.7 ms)
RTX 5060 Ti 16 GB

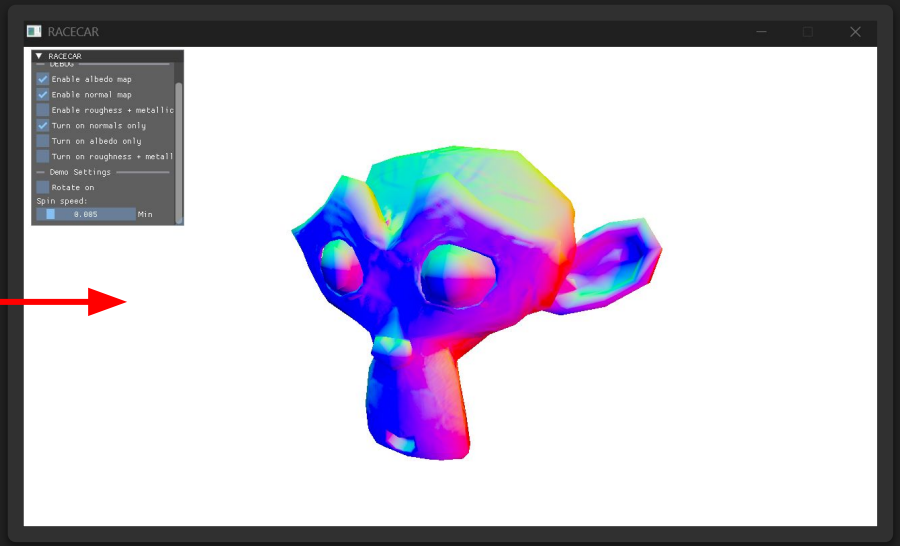
Engine Feature: PBR + Texture Loading

Supports glTF PBR extensions: metallicness, roughness, specular, IOR, clearcoat, sheen, transmission, emission



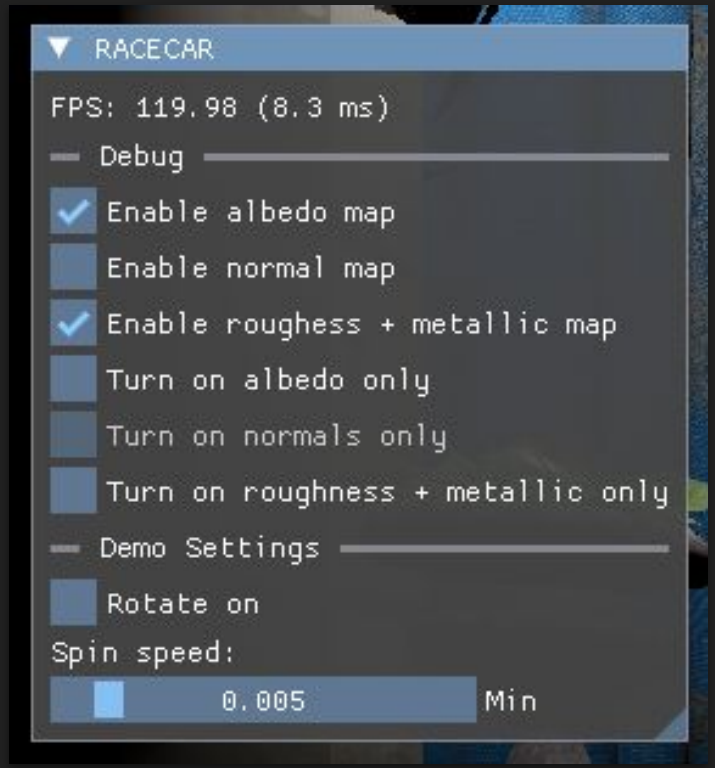
Engine Feature: PBR + Texture Loading





Normal maps

Engine Feature: ImGUI



- Toggleable features that interact with engine state and graphics pipeline

Challenges

Shift in direction: we initially pitched with glints and radiances cascades for global illumination but it didn’t fit cohesively → focused on engine this week while redefining goals.

Next steps

- Atmospheric Rendering
  - Precomputed skies + 2D LUT approach (GT7)
  - Volumetric clouds
- Reflections
  - Stochastic SSR fallback
  - IBL based on dynamic skies
  - VK ray tracing pipeline setup
- Materials
  - BSDF supporting Sheen, Clearcoat, Glints