(Raceca)r (R)enderer
Milestone 2

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

Recap

Goals:

- High fidelity, real-time car rendering
- Nice lighting, materials, atmospheres, scenery…

Image from Gran Turismo 7

Image from Forza Horizon 5

Milestone 2 Achievements

- Vulkan engine finished!
  - Deferred
  - Compute-driven post-processing
  - Ray tracing setup
- Real-time Eric Bruneton skies
- Pre-baked volumetric cloud rendering
- Materials
  - Specular IBL
  - Anisotropic glints
  - Clearcoat + fancy diffuse

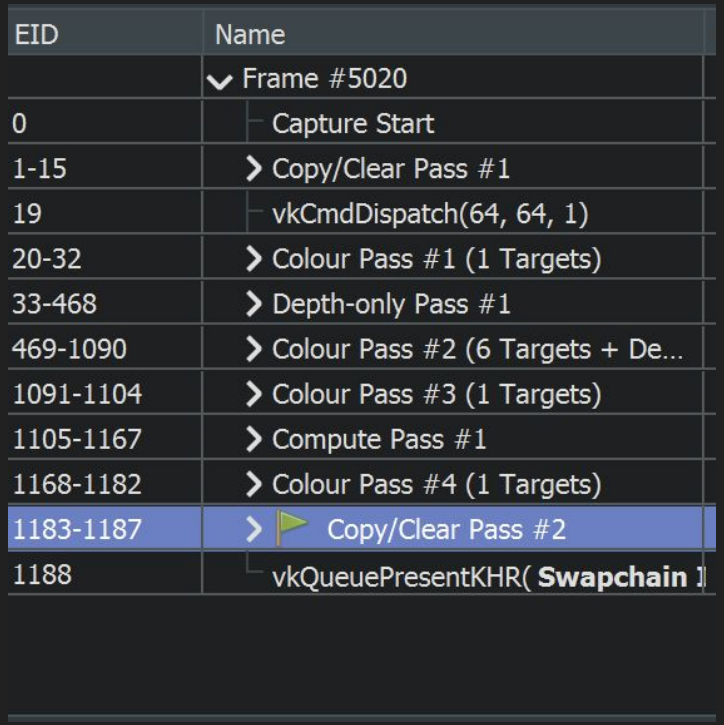
1280x720, ~395 FPS (~2.53ms)

RTX 4070 mobile GPU, 8GB

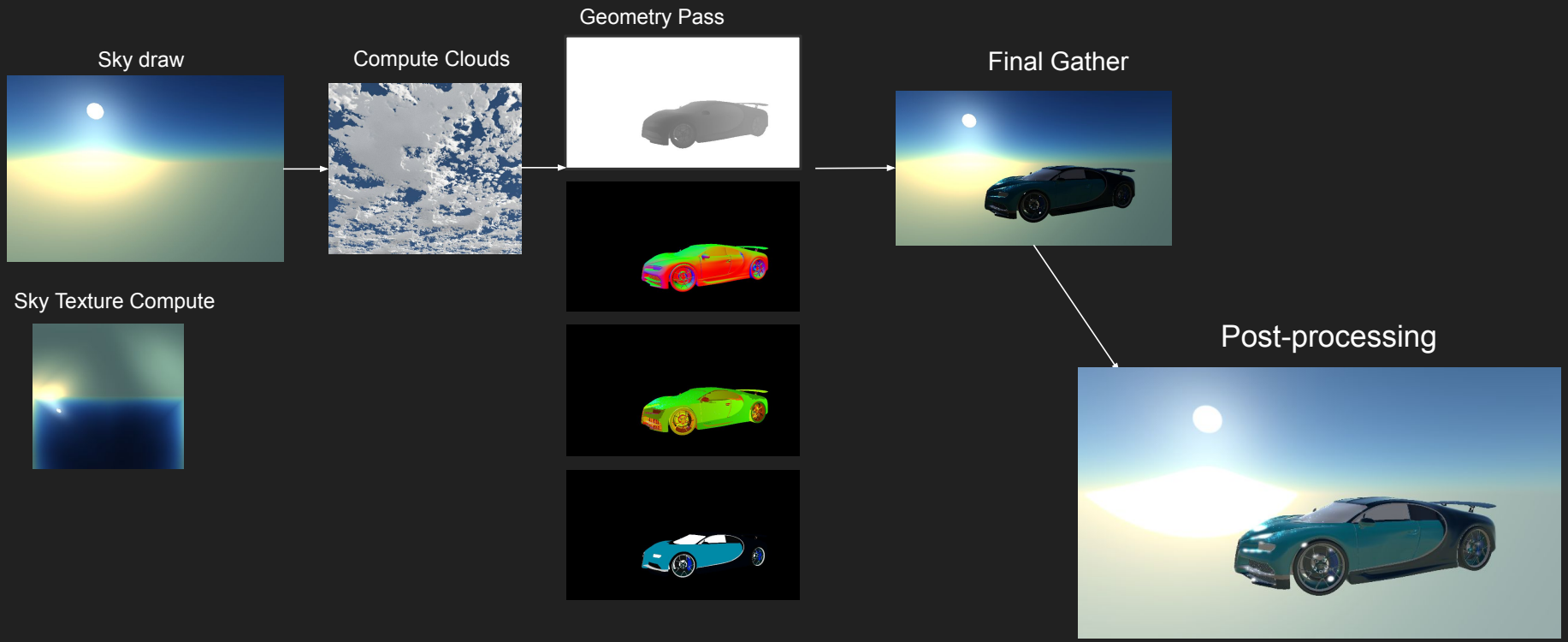

Configuration
FPS: 622.39 (1.6 ms)
Camera: [x=9.7, y=5.5, z=9]
General
Atmosphere
Post
Ping-pong zenith
3.554
Sun azimuth
1.358
Sun zenith
2.771
Radiance expos
1920×1080, ~620 FPS (1.6 ms)
RTX 5060 Ti 16 GB

Engine

- GfxTask, ComputeTask, BlitTask
- Deferred Rendering
- Sky Precompute Pipeline
- Post-processing Pipeline

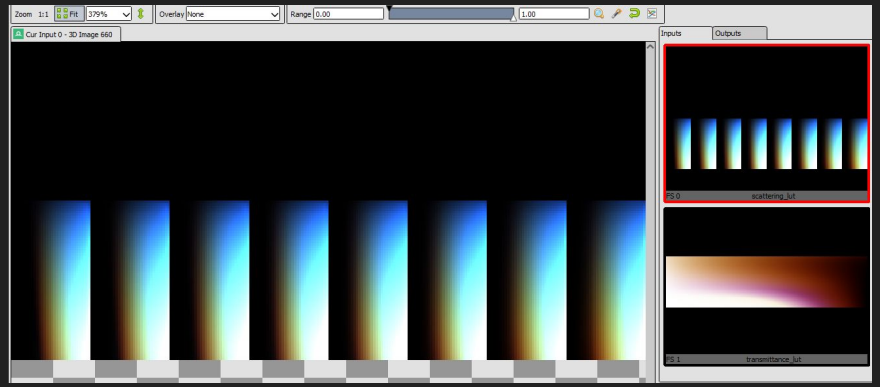


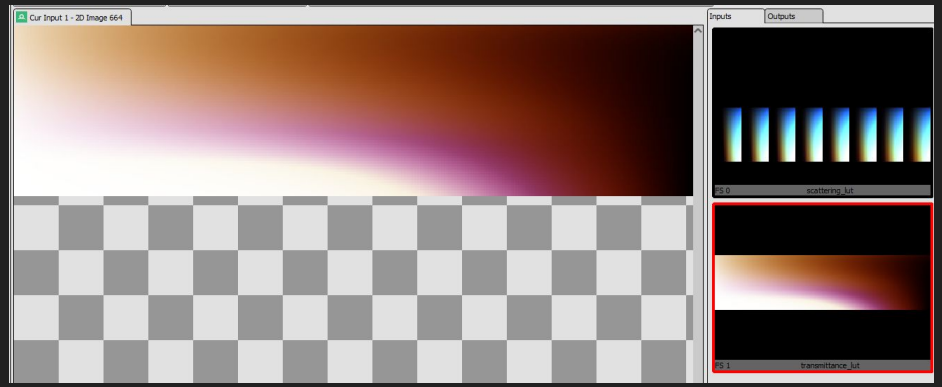
Engine: Frame breakdown



Sky and atmosphere

- Based on Eric Bruneton's (updated) 2017 paper
- Precomputed binary data for irradiance and transmittance
  - At runtime, we resolve the viewing angle/sun angle and perform texture look-ups
  - Solved/hardcoded for Earth atmosphere parameters since that's all we want
- Allows for real-time performance
- **Shout out to our shadow team~ (Avi + Jacky)**





Sky and atmosphere

Eric's codebase had many hardcoded assumptions: ray intersection with a "demo sphere", rendered Earth's atmosphere (another sphere intersection), hardcoded shadows, hardcoded light shafts

- Also only supported OpenGL + GLSL. Our pipeline is Vulkan + Slang…
- Rewrote the whole thing and stripped out unnecessary functionality
- Original code also had many assumptions that led to stupid bugs
  - Assumed +Z is up axis
  - Vulkan clip space points positive y-axis DOWNWARDS
  - Uses kilometers as base unit (so y = 9 was 9 KILOMETERS IN THE EARTH'S ATMOSPHERE)
- Takes approximately ~0.3 ms on Laptop 4070. Not bad!

Sky and atmosphere (YouTube link)

Sky and atmosphere

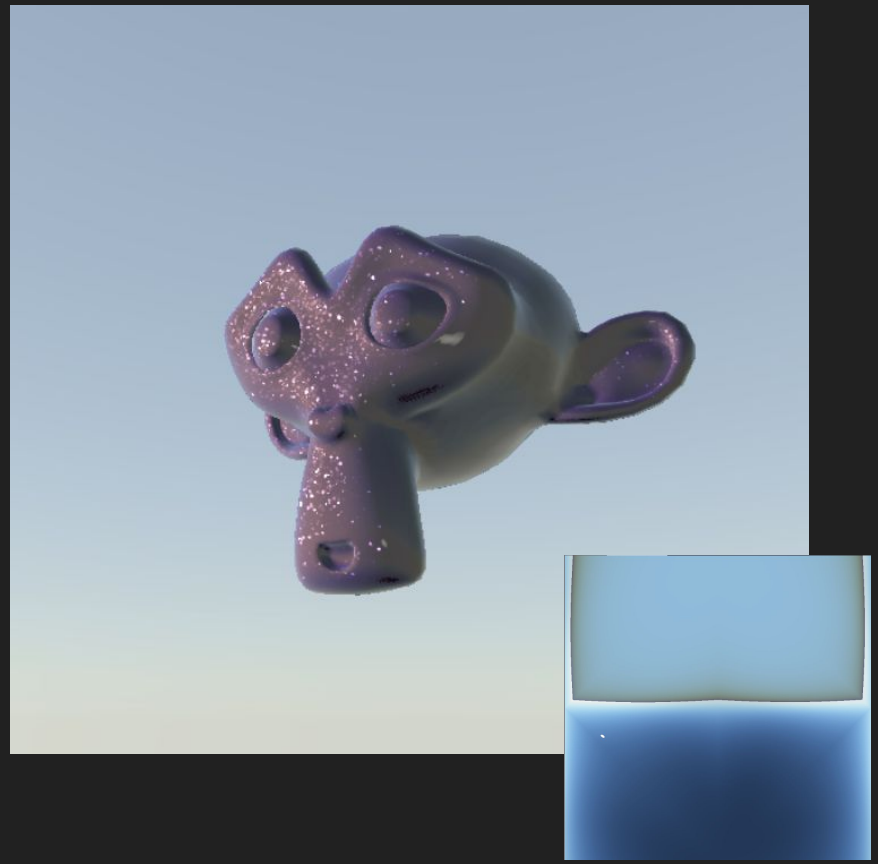
Clouds

- Based on Gran Turismo’s 2023 sky rendering talk at GDC
- Uses 3 layers of noises at different scales - cached into small textures
- Accounts Beer’s Law and Two Term Henyey-Greenstein for forward and back scattering
- Jitter to avoid banding artifacts

Materials + IBL

- More PBR features
  - Deliot[23] **Glints** (~0.07ms per deferred pre-pass)
  - Supports clearcoat now
- Dynamic sky-based IBL
  - Bake the sky into octahedral map, saves expensive cubemap space **(~0.31ms)**

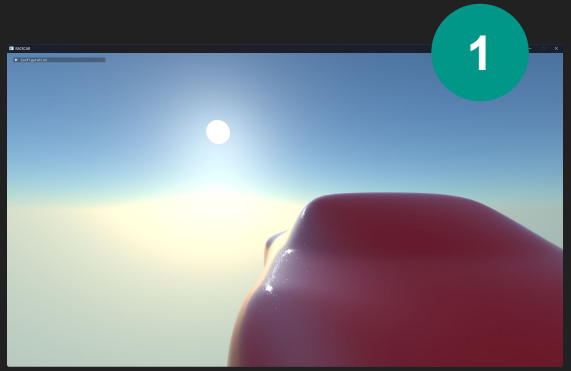
Perf logged with 4070 mobile GPU



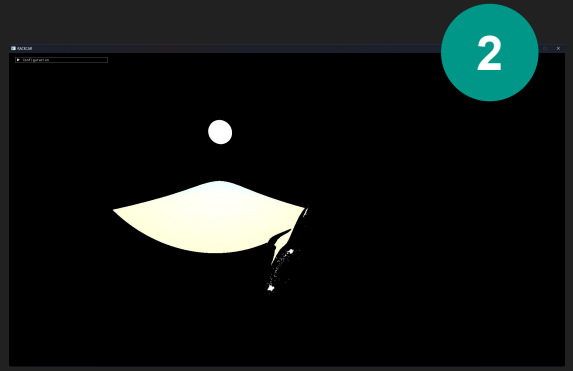
Materials + IBL (YouTube link)

Bloom

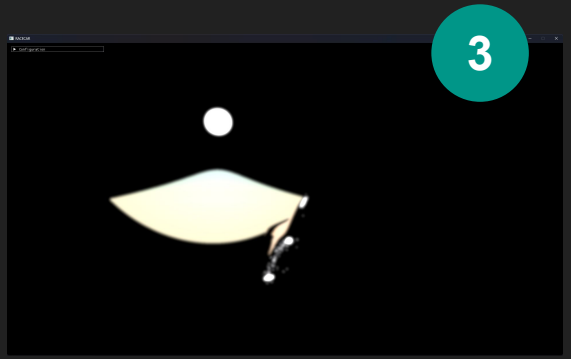
- First post-processing effect, utilizes our ComputeTask system
- Final gather is blit onto the screen
- Performance
  - 0.4ms (RTX 5060 Ti)
  - 1.3ms (4070 Laptop)
- Future improvements:
  - Linear filtering to reduce texture look-ups
  - Downsample then upsample passes
  - Combining Gaussian kernels for a smoother curve



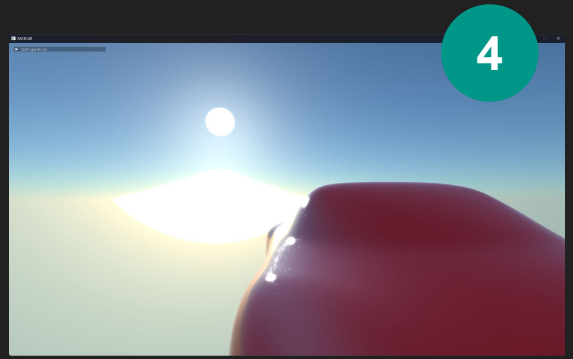
Render to VkImage



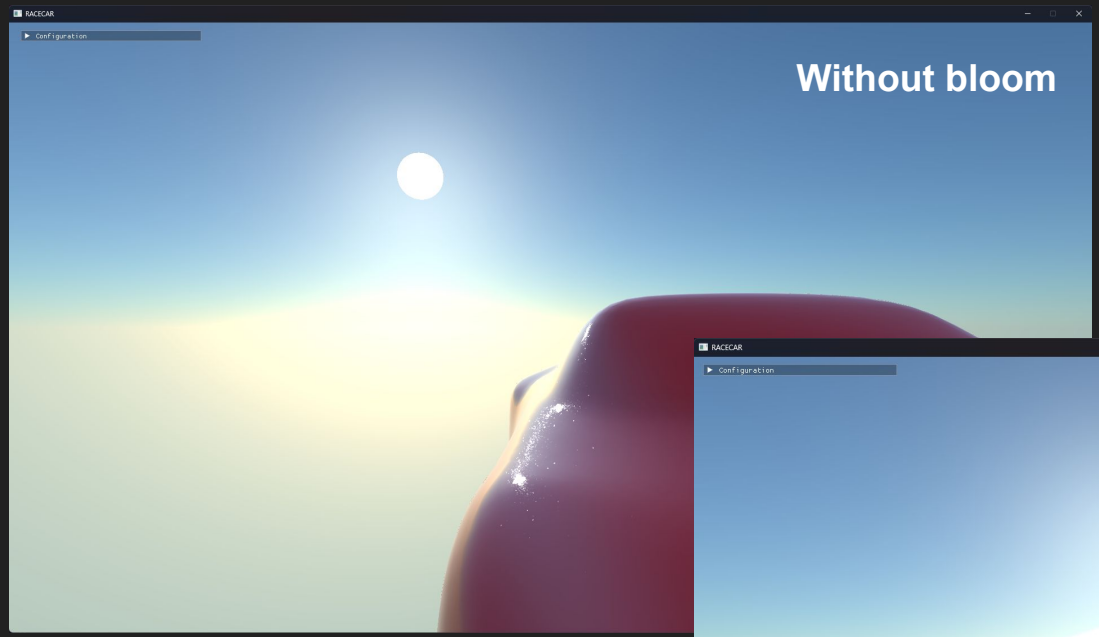
Compute pass: brightness threshold



5 compute passes: two-pass linear Gaussian blur



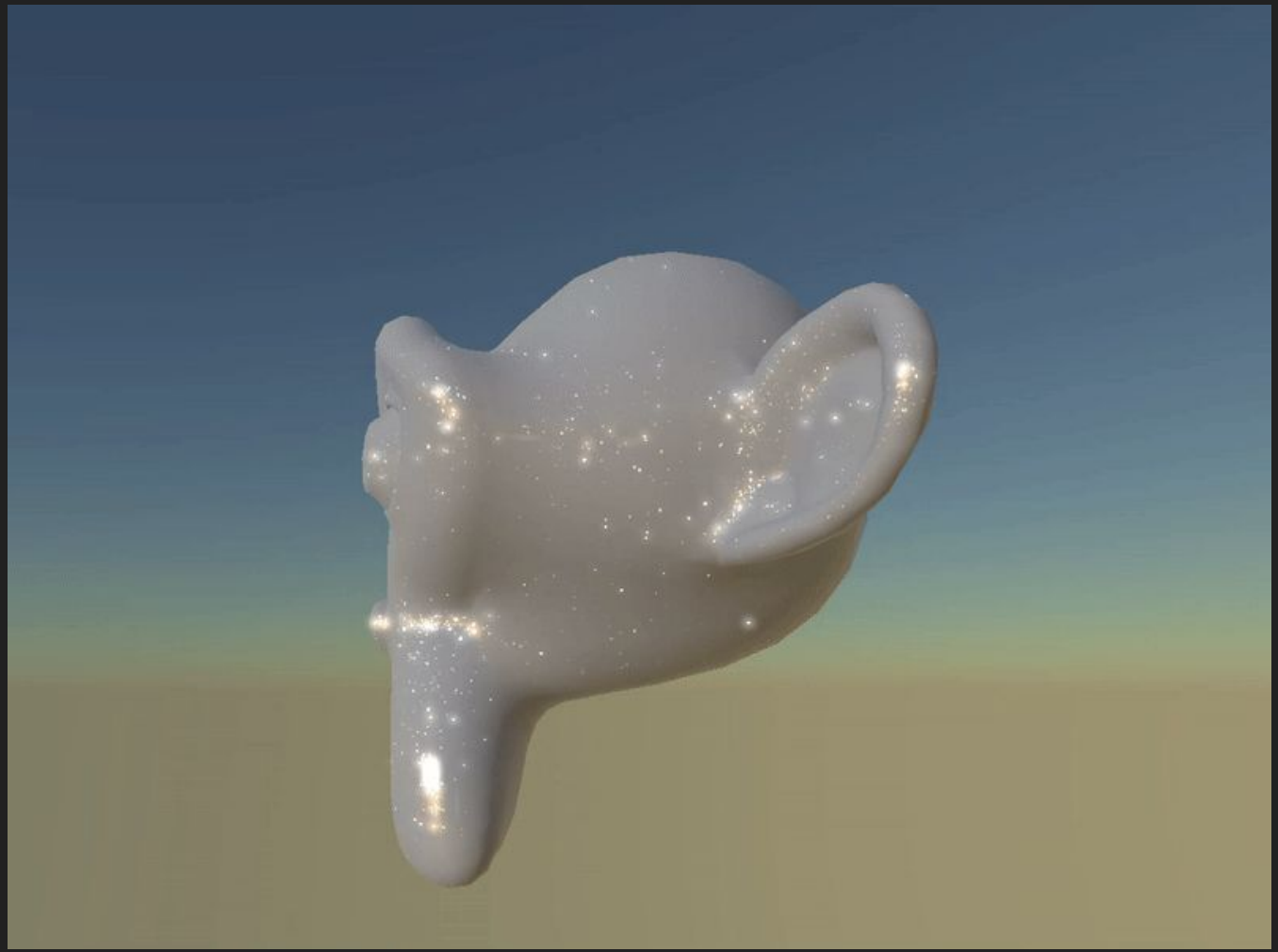
Compute pass: additive blending


RACECAR
Configuration
Without bloom


RACECAR
Configuration
With bloom

Bloom

Bloom interacting with glints!



Other General Improvements

- Ray Tracing
  - Acceleration structures finished
  - SBT finished
- MS 4x Depth Pass
  - Used for AA later

Next steps

- Materials
  - Better rough glossy, SH (Anthony)
- Atmosphere
  - Dynamic skies to LUT-based skies (Charles)
  - Optimized volumetric clouds (Aaron)
- Reflections
  - Simple ray tracing (Saahil)
- Terrain
  - Concurrent binary trees (Saahil + Anthony)
- GT7 “physically based” tone mapping (Charles)