

The image shows a title screen for a game called "NEON RACE". The title is rendered in a large, stylized, blocky font with a neon effect. The letters "NEON" are primarily yellow and green, while "RACE" is primarily blue and cyan. The background is a dark, stylized landscape with a road leading towards a horizon under a full moon. There are some buildings and trees in the distance, and a car is visible on the right side of the road.

NEON RACE

press ANY KEY to START

by Ata Berktürk, Mehmet Dereli
and Lars Hinnerk Grevsmühl

Projection

- Geometry Shader
- Extrusion of sides
- Fragment function to create “ray effect”



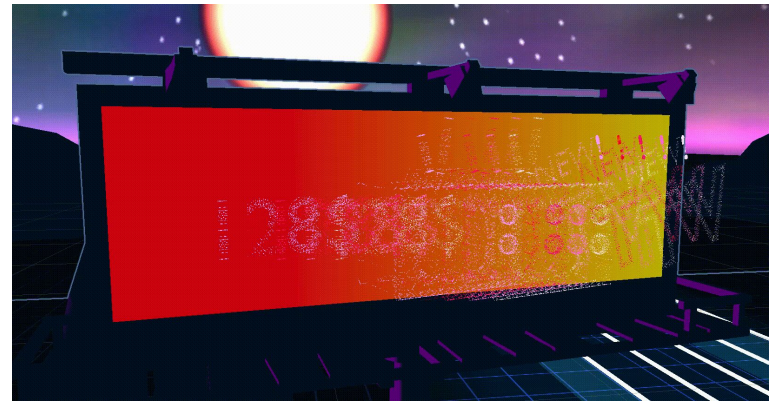
Dissolve

- Surface Shader
- Clip the MainTex according
to the Dissolve Texture



Monroe

- Geometry Shader
- Tessellation
- Stationary copies passed to the fragment shader are colored by their id.



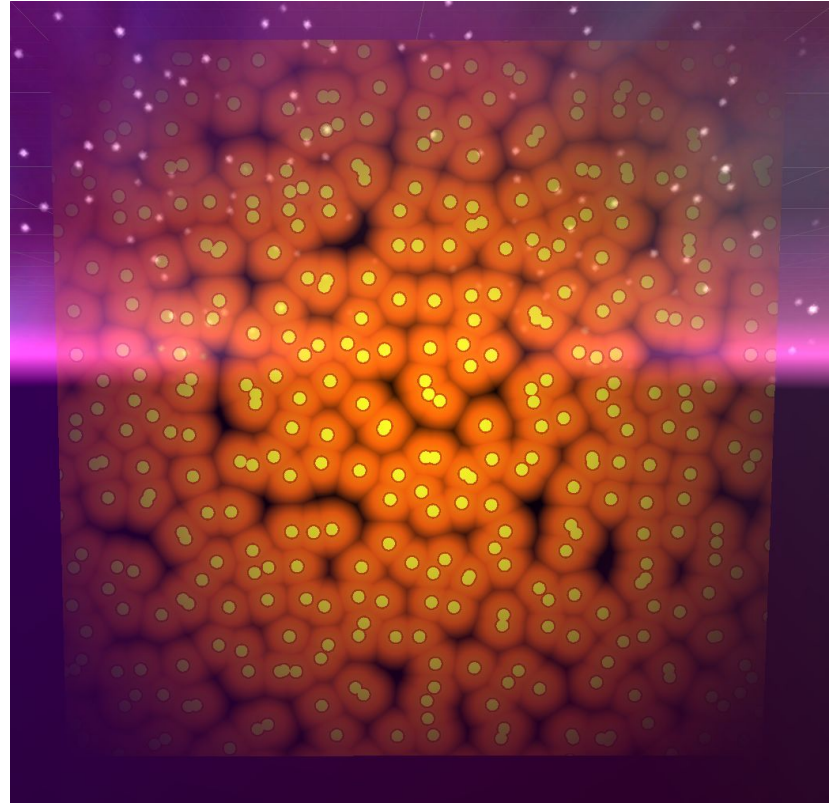
Monroe-Circle

- Geometry Shader
- Tessellation
- Stationary copies passed to the fragment shader are colored by their id.



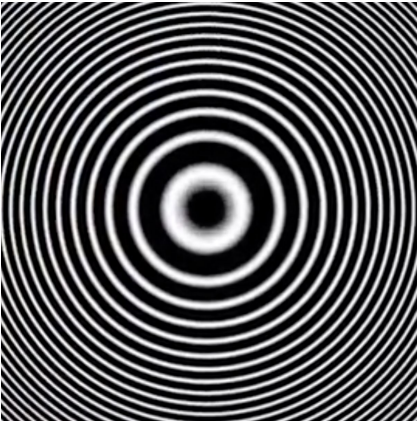
Laser Barrier

- Voronoi Displacement
- UV-Grid for efficiency



Sun

- Modified sin/cos wave
loop



Old-School Hologram

- 2 B/W Textures for alpha
- Fresnel Effect
 - Inner -> Outer effect that depends on Normal and viewDir vectors.
- View angle independent horizontal transparency slices



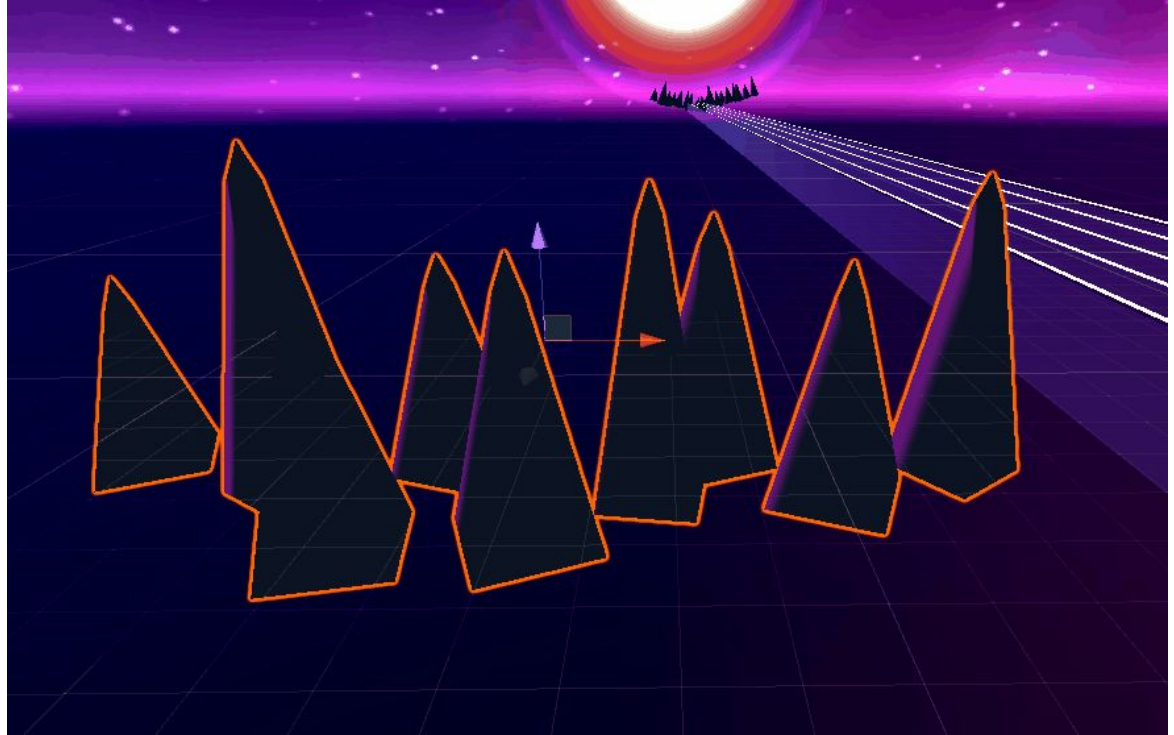
Neon Bar

- simple use of uv.x for setting bar to value
- uvs are mirrored so its always moving towards the center



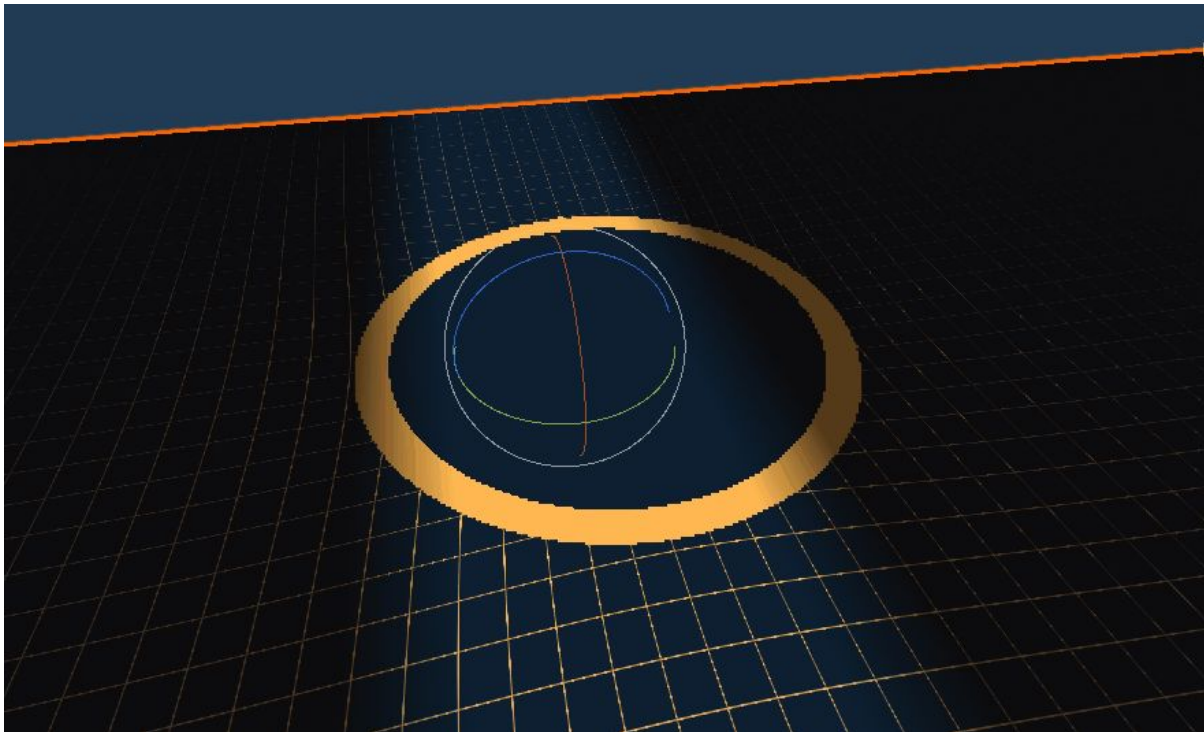
World Normal Color

- Lerps between 2 colors based on worldspace normal
- has bias to shift color in one direction



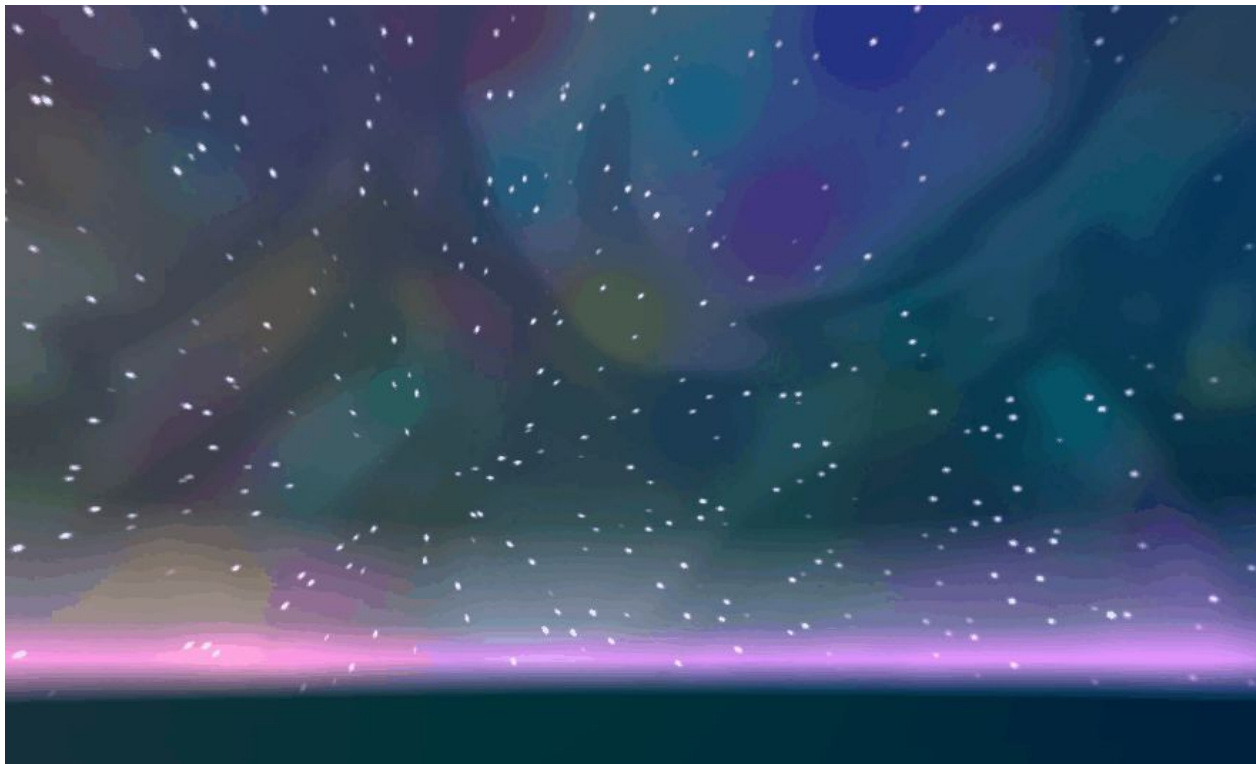
Terrain

- grid using texture (better due to mipmaps)
- blend color based on distance
- move uvs based on distance



Skybox

- base color
blended with
heights
- add stars
- add clouds
- add uvs



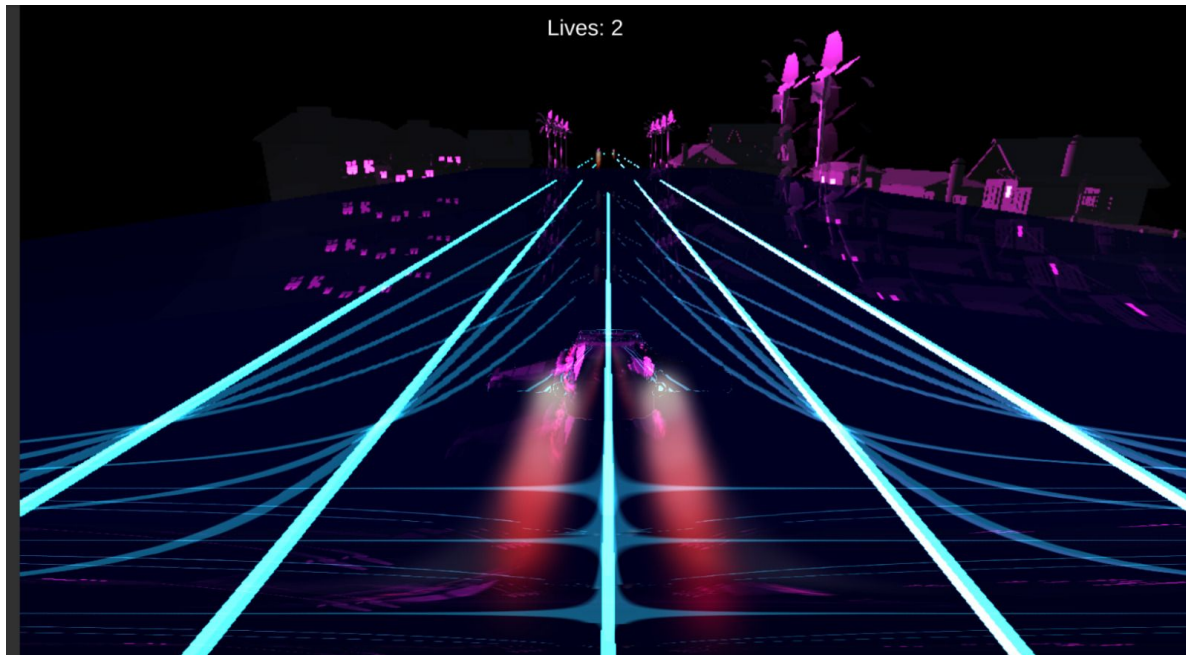
Bloom

- 8 px bloom kernel radius
 - fake gaussian blur (different smoothing function)
1. select pixels at half size
 2. blur X
 3. blur Y
 4. repeat
 5. add (reused for damage)



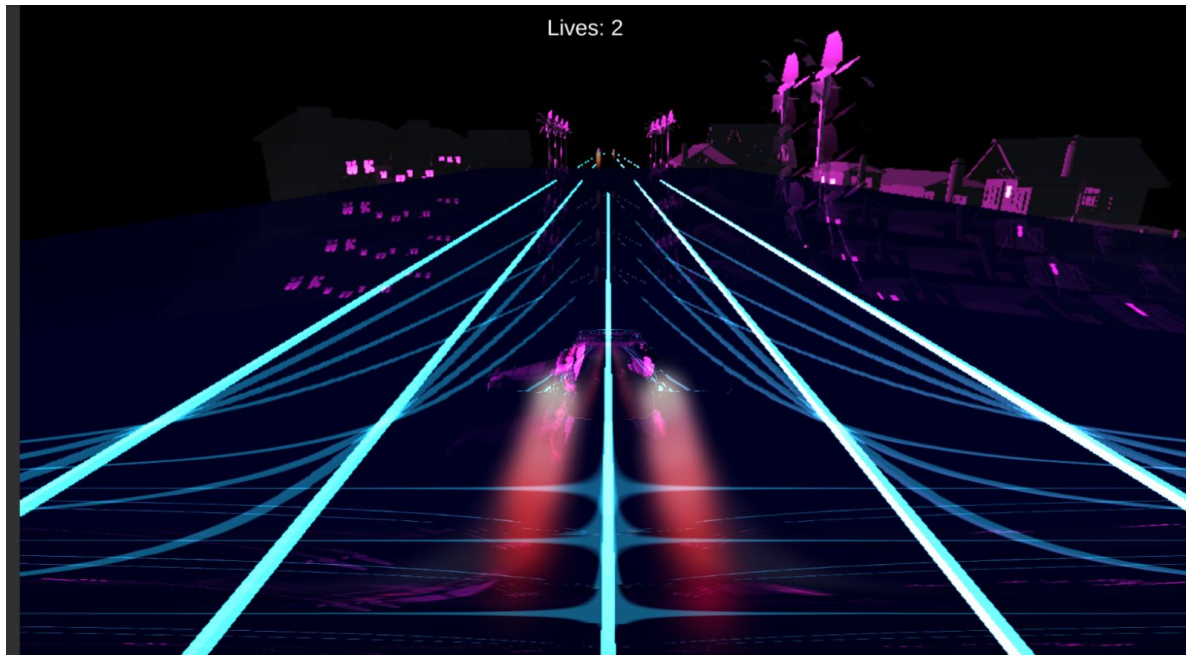
Motion Blur

- using camera motion vectors



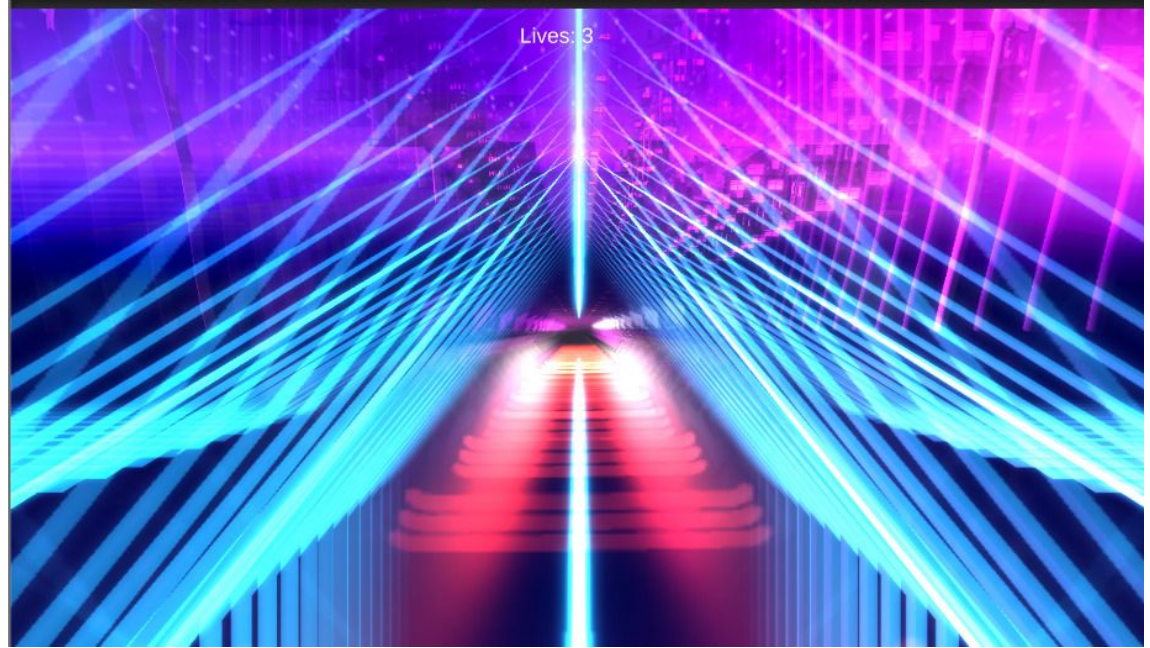
Motion Blur

- using camera motion vectors



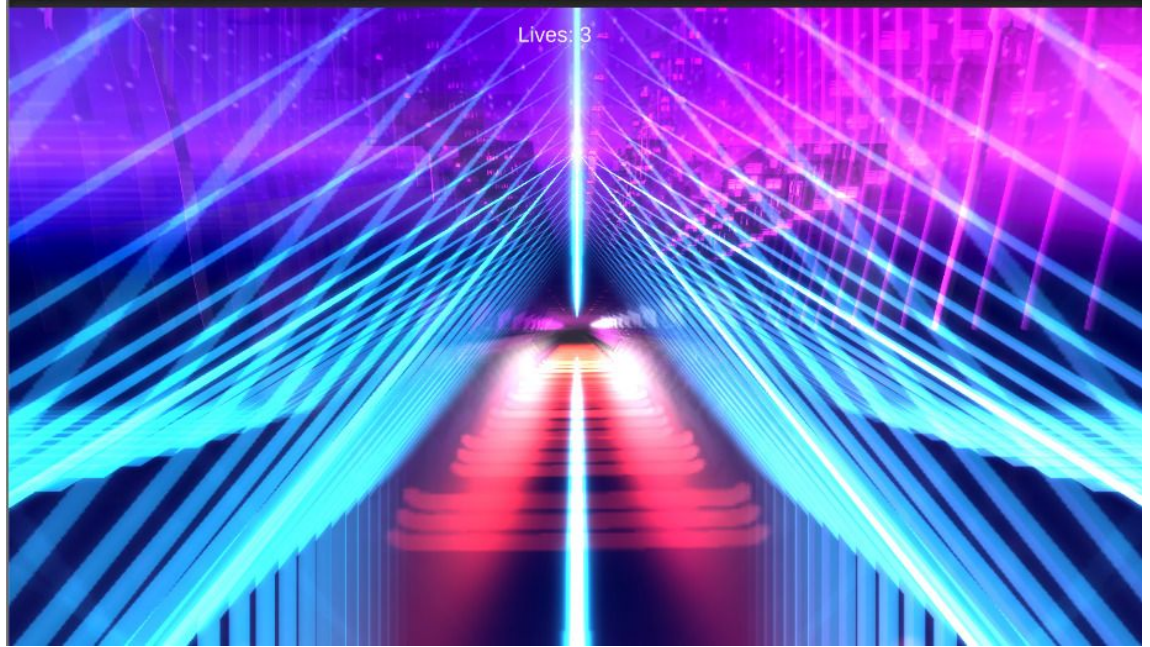
Motion Blur

- reading into custom stuff



Motion Blur

- reading into custom
shader



Motion Blur

- fake Motionblur!
- Just blurs depending on distance to vanishing point
- special sampling needed to remove artifacts

