

# Skybox Painter: Realtime Texture Editing

Marc Weber & Jashan Dhillon

# The Problem

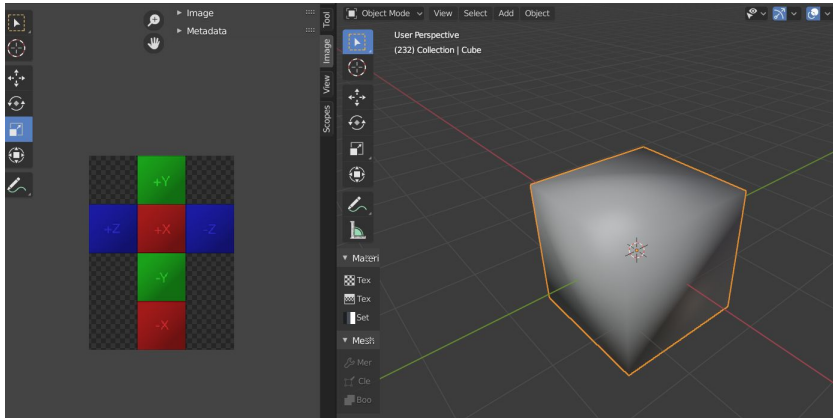
- Skyboxes are an integral part of game design
- Creating skyboxes can be time consuming
  - Drawing a custom skybox usually uses a cubemap, but it is difficult to guess what it will look like because of seams and distortion
- For custom skybox shaders, there is no way to in real-time see the changes to the sky as the texture is edited; you have to make an edit, import, and then see if it's as desired
- The Goal: Create a tool to allow easy customization of skyboxes and simple import into pre-existing game engines (Unity)

# Approach

- Create an application using OpenGL that allows painting on the inside of a cube
- Export the texture on the painted cube as a CubeMap
- Import the CubeMap into Unity to render as a skybox
- Also allow the user to import existing CubeMaps to edit

# Methodology + Workflow

- Configure the 3 main “threads”: checking user input, rendering to the display, and rendering to the texture
- GLFW and IMGUI for user input
- Construct a mesh cube with inverted normals, use that with custom shaders for rendering to the display (blender + model-converter)
- But how to update the texture based on only a mouse click position?



# Texture Drawing Techniques

- Raycast vs Rasterize
  - Issues caused by painting on seams
- Our solution: GPU texture redraw, simple cube UV map
- Other solutions that we did not take
  - Sphere map
  - Cylinder map

# GPU Texture Redrawing

- Treat the texture as a framebuffer to draw to, implement all drawing as a shader
- Fragments become pixels on the texture
- No manual projecting needed; texture coordinates interpolated for us by the pipeline

# Results

- Live Demo (switch screens)

# Limitations

- No blending feature
- Currently no way to preview the skybox in a specific scene, with objects
- Can't zoom in and out on certain parts of it to finetune drawing



# Future Directions

- Short Term (A few weeks)
  - Zoom in and out to add finer details
  - Add blending modes (additive, transparent, etc.)
  - Better brush projection onto the skybox + brush rotation
  - Custom brush textures
- Medium Term (A few months)
  - Add Unreal Engine support
  - Add an option to preview the skybox in a scene
  - Undo button, brush history
  - Edit multiple textures for the same shader
  - Move shader params around as you edit
- Long Term (A year or more)
  - Get it published as an addon for Unity/Unreal Engine

# References

- Ray Tracing from the Ground Up by Kevin Suffern, chapter 8 + 11
- <https://math.rice.edu/~polking/cartography/cart.pdf> - sphere mapping, John Polking
- <http://www.opengl-tutorial.org/intermediate-tutorials/tutorial-14-render-to-texture/> - MRT for image editing, open source webpage
- <https://assetstore.unity.com/packages/vfx/shaders/star-nest-skybox-63726> - a shader altered and used in the demo