

Package: hexglobe (via r-universe)

October 23, 2026

Title Render 'hexify' Globes on the Graphics Card Without a Browser

Version 0.1.0

Description Draws the globes of the 'hexify' package offscreen on the graphics card through 'wgpu', the 'Rust' implementation of 'WebGPU', with the same shader and pipelines as the 'hexify' globe widget, and returns the image as pixels or a PNG file. 'hexify' calls it from hex_globe_png(); no browser is needed.

License MIT + file LICENSE

Encoding UTF-8

SystemRequirements Cargo (Rust's package manager), rustc

Depends R (>= 4.2)

Suggests testthat (>= 3.0.0), hexify

Config/testthat/edition 3

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0

Config/pak/sysreqs libclang-dev

Repository <https://gcol33.r-universe.dev>

Date/Publication 2026-10-22 23:14:50 UTC

RemoteUrl <https://github.com/gcol33/hexglobe>

RemoteRef HEAD

RemoteSha 75068ace6416e47f45d4bd254bc15b9763287ef4

Contents

gpu_adapter	2
read_png	2
render_scene	3
write_png	4

Index	5
--------------	----------

gpu_adapter	<i>The graphics adapter scenes are drawn on</i>
-------------	---

Description

The adapter is picked once per session, preferring a discrete graphics card. The environment variables WGPU_POWER_PREF ("low" or "high") and WGPU_BACKEND (such as "dx12", "vulkan" or "metal"), set before the first scene is drawn, choose another.

Usage

```
gpu_adapter()
```

Value

The adapter's name and graphics backend, one string.

read_png	<i>Read a PNG file</i>
----------	------------------------

Description

Read a PNG file

Usage

```
read_png(file)
```

Arguments

file	Path of an 8-bit PNG file.
------	----------------------------

Value

An image, as from [render_scene](#).

render_scene	<i>Draw a globe scene</i>
--------------	---------------------------

Description

Draws a scene on the graphics card, offscreen, with the shader and the pipelines of the 'hexify' globe widget. Scenes are built by 'hexify' from a globe widget; `hexify::hex_globe_png()` is the usual way in.

Usage

```
render_scene(scene)
```

Arguments

scene	A scene, as above.
-------	--------------------

Details

A scene is a list of

shader the WGSL source, one string;

camera the camera uniform, 112 32-bit floats as a raw vector;

width, height the image size in pixels;

layers the layers in drawing order, each a list with kind (0 an opaque mesh, 1 a blended mesh coloured by value, 2 the cells of an ISEA grid, 3 polylines), count (indices of a mesh, segments of a polyline), and its buffers as raw vectors: uniform, pos, item, tri, index, values, ramp, grid, points, segment; and textures, the textures bound beside them, each a list of binding, format ("r32uint" or "rg32uint"), dimension ("2d" or "2d-array"), size (width, height and layers) and data, its texels row by row and layer by layer as a raw vector.

Value

An image: a list of rgba, the pixels as a raw vector of red, green, blue and alpha bytes, row by row from the top, alpha not premultiplied, and width and height.

write_png	<i>Write an image as a PNG file</i>
-----------	-------------------------------------

Description

Write an image as a PNG file

Usage

```
write_png(image, file)
```

Arguments

image	An image from render_scene or read_png .
file	Path of the PNG file to write.

Value

file, invisibly.

Index

gpu_adapter, [2](#)

read_png, [2](#), [4](#)

render_scene, [2](#), [3](#), [4](#)

write_png, [4](#)