# Package: SpaDES.shiny (via r-universe)

June 12, 2026

**Type** Package

**Title** Shiny Apps for Visualizing 'SpaDES' Simulation Outputs

**Description** Tools to build 'shiny' applications for exploring outputs from 'SpaDES' (Spatial Discrete Event Simulation) projects. Includes 'shine()', an interactive viewer that recursively discovers raster ('.tif') and image ('.png') outputs, groups time-series of the same object, and presents them on web maps (with difference layers) and as figures.

**Version** 0.1.0

**Date** 2026-06-12

**Depends** R (>= 4.1)

**Imports** grDevices, htmltools, leafem, leaflet, shiny, terra, tools, utils

**Suggests** SpaDES.core, testthat (>= 3.0.0)

**License** GPL-3

**Encoding** UTF-8

**Language** en-CA

**URL** https://github.com/PredictiveEcology/SpaDES.shiny

**BugReports** https://github.com/PredictiveEcology/SpaDES.shiny/issues

**ByteCompile** yes

**Roxygen** list(markdown = TRUE)

**Config/testthat/edition** 3

**Config/roxygen2/version** 8.0.0

**Config/pak/sysreqs** libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev make libpng-dev libuv1-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev zlib1g-dev

**Repository** https://predictiveecology.r-universe.dev

**Date/Publication** 2026-06-12 21:01:57 UTC

**RemoteUrl** https://github.com/PredictiveEcology/SpaDES.shiny

**RemoteRef** development

**RemoteSha** b0a52c9539076768b8ff276eb90a903f4e58ac30

# Contents



---

`shine` *Interactive Shiny visualizer for SpaDES outputs*

---

## Description

Recursively scans a folder for raster (`.tif`) and image (`.png`) outputs, groups files that share a base name but differ by an embedded timestamp into time-series "objects", and launches a Shiny app to explore them. Maps (`.tif`) are drawn on a chooseable web basemap; figures (`.png`) are shown as images. Two further tabs show precomputed (`last - first`) and user-defined differences between map snapshots.

## Usage

```
shine(
  x,
  timePattern = "[0-9]+",
  maxCells = NULL,
  interval = 2,
  opacity = 1,
  launch = TRUE,
  ...
)
```

## Arguments

| | |
|---|---|
| `x` | What to visualize, resolved to a folder to scan recursively: a `simList` (its `SpaDES.core::outputPath()` is used), a single path string, or missing (uses `getOption("spades.outputPath")`). |
| `timePattern` | Regex matching the timestamp token in a file name. The **last** match in the (extension-stripped) base name is used as the numeric time. Default `"[0-9]+"`. |
| `maxCells` | Optional cap on pixels per side; if set, rasters larger than this are aggregated down before rendering. Default `NULL` (full resolution). |
| `interval` | Initial seconds between automatic time-slider steps; also the starting value of the in-app "seconds per step" slider. Default `2`. |
| `opacity` | Opacity of the raster layers (NoData stays transparent so the basemap shows through). Default `1`. Opaque layers animate seamlessly (double-buffered); values `< 1` use a single layer (a brief redraw per step, but no ghosting between semi-transparent frames). |
| `launch` | If `TRUE` (default), run the app with `shiny::runApp()`. If `FALSE`, return the `shiny::shinyApp()` object (useful for testing/embedding). |
| `...` | Passed to `shiny::runApp()` (e.g. `port`, `launch.browser`). |

## Details

Rasters are rendered with `leafem::addGeotiff()`, which draws them through the tiled `georaster-layer-for-leaflet` canvas layer: pan/zoom redraw per-tile on the client at native resolution (fast, no server round-trip per interaction). Each raster is reprojected once to a web-mercator Cloud-Optimized GeoTIFF (with overviews), cached on disk, and served same-origin via `shiny::addResourcePath()`.

Multi-band rasters contribute one object per band. Files whose name contains no digit run are treated as static (always-shown) layers with no position on the time slider.

## Value

Invisibly, the `shinyApp` object (also when `launch = TRUE`).

## Examples

```
## Not run:
shine("outputs")        # a path
shine(mySim)            # a simList -> uses outputPath(mySim)
shine()                 # uses getOption("spades.outputPath")

## End(Not run)
```

# Index