

Building an Interactive Storymap using {mapgl}

A Visual Journey Through My Life.

Pukar Bhandari

pukar.bhandari@outlook.com

2025-09-03

Table of contents

0.1 Introduction	1
0.2 Why {mapgl} for Storymaps?	1
0.3 Interactive Storymap Demo	2
0.4 Source Code	5
0.5 Understanding the App Structure	8
Setting Up the User Interface	8
Configuring the Map	9
Implementing Section Behaviors	10
0.6 Conclusion	10
0.7 Additional Resources	11

0.1 Introduction

Interactive storymaps have become an increasingly popular tool for data visualization and story-telling, particularly in fields like urban planning, transportation analysis, and geographic data science. They allow us to combine spatial data with narrative elements, creating compelling visual stories that engage audiences while conveying complex information effectively.

In this post, I'll walk you through creating an interactive storymap using the {mapgl} R package—a powerful tool that leverages MapLibre GL JS to create stunning, performant web maps. This particular storymap traces my personal journey across different locations where I've lived, studied, and worked, from Nepal to the United States.

0.2 Why {mapgl} for Storymaps?

The {mapgl} package offers several advantages for transportation planners and data analysts:

- **Performance:** Built on MapLibre GL JS, it handles large datasets and complex visualizations smoothly
- **Interactivity:** Native support for user interactions, animations, and dynamic content
- **Customization:** Extensive styling options and the ability to use custom map tiles
- **Integration:** Seamless integration with Shiny for reactive web applications

- **Open Source:** No API keys required when using open map styles

0.3 Interactive Storymap Demo

Below is the complete interactive storymap. You can scroll through the different sections to see how the map smoothly transitions between locations, each representing a significant milestone in my journey.

i Note

The `add_globe_control()` function isn't working because it's only available in `mapgl` version 0.2.2 and higher. The version on CRAN is 0.4.0, while the one on `repo.r-wasm.org` is still 0.2.1.

Once `mapgl` is updated on `repo.r-wasm.org`, this storymap will work as expected. Thank you for your understanding!

```
#| '!! shinylive warning !!': |
#| shinylive does not work in self-contained HTML documents.
#| Please set `embed-resources: false` in your metadata.
#| standalone: true
#| viewerHeight: 600

# Install mapgl from Kyle Walker's r-universe
# webr::install("mapgl", repos = "https://walkerke.r-universe.dev/")

## file: app.R
library(shiny)
library(mapgl)

ui <- shiny::fluidPage(
  tags$link(
    href = "https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&
display=swap",
    rel="stylesheet"
  ),
  mapgl::story_map(
    map_id = "map",
    font_family = "Poppins",
    sections = list(
      "intro" = mapgl::story_section(
        title = "Introduction",
        content = "Hello I am Pukar Bhandari; and this is my life journey."
      ),
      "childhood" = mapgl::story_section(
        title = "Birtamode, Jhapa",
        content = "This is where I was born and spent the early part of my
life until 2013."
      ),
    )
  )
)
```

```

    "bachelors" = mapgl::story_section(
      title = "Pulchowk, Lalitpur",
      content = "I then moved to Pulchowk in the pursuit of higher
education. I obtained my Bachelor's Degree in Architecture from Institute of
Engineering Pulchowk Campus in 2018."
    ),
    "masters" = mapgl::story_section(
      title = "Salt Lake City, UT",
      content = "In pursuit of even higher education, I moved to Salt Lake
City, UT where I obtained my Master of City & Metropolitan Planning degree
from the University of Utah in May 2023."
    ),
    "metro_analytics" = mapgl::story_section(
      title = "Atlanta, GA",
      content = "I then moved to Atlanta in June 2023 to work as a
transportation planner & analyst at Metro Analytics until September 2025."
    ),
    "wfrc_ut" = mapgl::story_section(
      title = "Back to Salt Lake City, UT",
      content = "After gaining entry-level experience in transportation
planning, travel demand modeling, and data analytics, I moved back to Salt
Lake City - a place I am personally and professionally attached to - in order
to join the awesome analytics team at the Wasatch Front Regional Council."
    )
  ),
  map_type = "maplibre"
)
)

server <- function(input, output, session) {
  output$map <- mapgl::renderMaplibre({
    mapgl::maplibre(
      style = "https://tiles.openfreemap.org/styles/liberty",
      center = c(0, 0),
      zoom = 2.75,
      scrollZoom = FALSE
    ) |>
    mapgl::set_projection(projection = "globe") |>
    mapgl::add_globe_control() |>
    mapgl::add_navigation_control(visualize_pitch = TRUE) |>
    mapgl::add_globe_minimap(position = "bottom-left")
  })

  mapgl::on_section("map", "intro", {
    mapgl::maplibre_proxy("map") |>
    mapgl::clear_markers() |>
    mapgl::fly_to(
      center = c(0, 0),

```

```

        zoom = 2.75,
        pitch = 0,
        bearing = 0
      )
    })

    mapgl::on_section("map", "childhood", {
      mapgl::maplibre_proxy("map") |>
        mapgl::fly_to(center = c(87.99371750247397, 26.646533355308083),
          zoom = 16,
          pitch = 49,
          bearing = 12.8)
    })

    mapgl::on_section("map", "bachelors", {
      mapgl::maplibre_proxy("map") |>
        mapgl::fly_to(center = c(85.31840215760691, 27.681136558618093),
          zoom = 16,
          pitch = 49,
          bearing = 12.8)
    })

    mapgl::on_section("map", "masters", {
      mapgl::maplibre_proxy("map") |>
        mapgl::fly_to(center = c(-111.84448004883544, 40.76106105996334),
          zoom = 16,
          pitch = 49,
          bearing = 12.8)
    })

    mapgl::on_section("map", "metro_analytics", {
      mapgl::maplibre_proxy("map") |>
        mapgl::fly_to(center = c(-84.3938999520061, 33.76728402587881),
          zoom = 16,
          pitch = 49,
          bearing = 12.8)
    })

    mapgl::on_section("map", "wfrc_ut", {
      mapgl::maplibre_proxy("map") |>
        mapgl::fly_to(center = c(-111.90422445885304, 40.76973849954712),
          zoom = 16,
          pitch = 49,
          bearing = 12.8)
    })
  }

  shiny::shinyApp(ui, server)

```

0.4 Source Code

The storymap application follows a typical Shiny structure with some specialized {mapgl} components. Following is the source code for this dashboard (Please click on the numbers on the far right for additional explanation of syntax and mechanics):

```
library(shiny)
library(mapgl)

ui <- shiny::fluidPage(
  tags$link(
    href = "https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&
display=swap",
    rel="stylesheet"
  ),
  mapgl::story_map(
    map_id = "map",
    font_family = "Poppins",
    sections = list(
      "intro" = mapgl::story_section(
        title = "Introduction",
        content = "Hello I am Pukar Bhandari; and this is my life journey."
      ),
      "childhood" = mapgl::story_section(
        title = "Birtamode, Jhapa",
        content = "This is where I was born and spent the early part of my
life until 2013."
      ),
      "bachelors" = mapgl::story_section(
        title = "Pulchowk, Lalitpur",
        content = "I then moved to Pulchowk in the pursuit of higher
education. I obtained my Bachelor's Degree in Architecture from Institute of
Engineering Pulchowk Campus in 2018."
      ),
      "masters" = mapgl::story_section(
        title = "Salt Lake City, UT",
        content = "In pursuit of even higher education, I moved to Salt Lake
City, UT where I obtained my Master of City & Metropolitan Planning degree
from the University of Utah in May 2023."
      ),
      "metro_analytics" = mapgl::story_section(
        title = "Atlanta, GA",
        content = "I then moved to Atlanta in June 2023 to work as a
transportation planner & analyst at Metro Analytics until September 2025."
      ),
      "wfrc_ut" = mapgl::story_section(
        title = "Back to Salt Lake City, UT",
        content = "After gaining entry-level experience in transportation
planning, travel demand modeling, and data analytics, I moved back to Salt
```

Lake City - a place I am personally and professionally attached to - in order to join the awesome analytics team at the Wasatch Front Regional Council."

```
)
),
map_type = "maplibre"
)
)

server <- function(input, output, session) {
  output$map <- mapgl::renderMaplibre({
    mapgl::maplibre(
      style = "https://tiles.openfreemap.org/styles/liberty",
      center = c(0, 0),
      zoom = 2.75,
      scrollZoom = FALSE
    ) |>
    mapgl::set_projection(projection = "globe") |>
    mapgl::add_globe_control() |>
    mapgl::add_navigation_control(visualize_pitch = TRUE) |>
    mapgl::add_globe_minimap(position = "bottom-left")
  })

  mapgl::on_section("map", "intro", {
    mapgl::maplibre_proxy("map") |>
    mapgl::clear_markers() |>
    mapgl::fly_to(
      center = c(0, 0),
      zoom = 2.75,
      pitch = 0,
      bearing = 0
    )
  })

  mapgl::on_section("map", "childhood", {
    mapgl::maplibre_proxy("map") |>
    mapgl::fly_to(center = c(87.99371750247397, 26.646533355308083),
      zoom = 16,
      pitch = 49,
      bearing = 12.8)
  })

  mapgl::on_section("map", "bachelors", {
    mapgl::maplibre_proxy("map") |>
    mapgl::fly_to(center = c(85.31840215760691, 27.681136558618093),
      zoom = 16,
      pitch = 49,
      bearing = 12.8)
  })
}
```

```

mapgl::on_section("map", "masters", {
  mapgl::maplibre_proxy("map") |>
    mapgl::fly_to(center = c(-111.84448004883544, 40.76106105996334),
                  zoom = 16,
                  pitch = 49,
                  bearing = 12.8)
})

mapgl::on_section("map", "metro_analytics", {
  mapgl::maplibre_proxy("map") |>
    mapgl::fly_to(center = c(-84.3938999520061, 33.76728402587881),
                  zoom = 16,
                  pitch = 49,
                  bearing = 12.8)
})

mapgl::on_section("map", "wfrs_ut", {
  mapgl::maplibre_proxy("map") |>
    mapgl::fly_to(center = c(-111.90422445885304, 40.76973849954712),
                  zoom = 16,
                  pitch = 49,
                  bearing = 12.8)
})
}

shiny::shinyApp(ui, server)

```

Line 1

Load {shiny} package for building reactive web applications

Line 2

Load {mapgl} package for interactive mapping and storymap functionality

Line 4

Define the user interface - what users see and interact with

Line 9

Create the main storymap interface with sidebar and map coordination

Line 12

Define each story section with title and descriptive content

Line 42

Define the server function - handles user interactions and updates

Line 43

Render the interactive MapLibre map with styling and controls

Line 45

Use free OpenFreeMap tiles (no API key required)

Line 50

Set up 3D globe projection for better geographic context

Line 56

Handle section transitions - when user reaches each story step

Line 108

Launch the Shiny application combining UI and server components

0.5 Understanding the App Structure

Setting Up the User Interface

```
library(shiny)
library(mapgl)

ui <- shiny::fluidPage(
  tags$link(
    href = "https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;600&
display=swap",
    rel="stylesheet"
  ),
  mapgl::story_map(
    map_id = "map",
    font_family = "Poppins",
    sections = list(
      "intro" = mapgl::story_section(
        title = "Introduction",
        content = "Hello I am Pukar Bhandari; and this is my life journey."
      ),
      "childhood" = mapgl::story_section(
        title = "Birtamode, Jhapa",
        content = "This is where I was born and spent the early part of my
life until 2013."
      ),
      # Additional sections...
    ),
    map_type = "maplibre"
  )
)
```

Line 1

Shiny Framework: Essential for creating reactive web applications

Line 2

MapGL Package: Provides the storymap functionality and MapLibre integration

Line 5

Custom Fonts: Loading Google Fonts for better typography (important for professional presentations)

Line 9

Story Map Container: The main wrapper that coordinates the sidebar content with map interactions

Line 11

Typography: Consistent font family improves readability and professional appearance

Line 12

Section Definitions: Each section corresponds to a step in the story with associated map behaviors

Line 23

Map Engine: Using MapLibre GL JS for performant vector tile rendering

Configuring the Map

```
server <- function(input, output, session) {  
  output$map <- mapgl::renderMaplibre({  
    mapgl::maplibre(  
      style = "https://tiles.openfreemap.org/styles/liberty",  
      center = c(0, 0),  
      zoom = 2.75,  
      scrollZoom = FALSE  
    ) |>  
    mapgl::set_projection(projection = "globe") |>  
    mapgl::add_globe_control() |>  
    mapgl::add_navigation_control(visualize_pitch = TRUE) |>  
    mapgl::add_globe_minimap(position = "bottom-left")  
  })  
}
```

Line 2

Render Function: Creates the reactive map output that responds to user interactions

Line 4

Open Map Style: Using OpenFreeMap's Liberty style (free, no API key required)

Line 5

Initial Center: Starting at global view (longitude 0, latitude 0)

Line 6

Zoom Level: Appropriate for showing global context initially

Line 7

Scroll Control: Disabled to prevent interference with story navigation

Line 9

Globe Projection: Creates a 3D globe effect for better geographical context

Line 10

Globe Controls: Adds rotation controls for the 3D globe

Line 11

Navigation Tools: Zoom and rotation controls with pitch visualization

Line 12

Minimap: Provides geographic context when zoomed in on specific locations

Implementing Section Behaviors

```
# Example section handler
mapgl::on_section("map", "childhood", {
  mapgl::maplibre_proxy("map") |>
    mapgl::fly_to(
      center = c(87.99371750247397, 26.646533355308083),
      zoom = 16,
      pitch = 49,
      bearing = 12.8
    )
})
```

Line 2

Section Trigger: Responds when user navigates to the “birth_place” section

Line 3

Map Proxy: Allows server-side control of the existing map instance

Line 4

Smooth Transition: `fly_to()` creates animated transitions between locations

Line 5

Coordinates: Precise longitude/latitude coordinates for the specific location

Line 6

Detail Zoom: High zoom level to show local context and landmarks

Line 7

Viewing Angle: Tilted perspective provides better spatial understanding

Line 8

Map Rotation: Oriented to show the location from the most informative angle

0.6 Conclusion

The {mapgl} package provides a powerful platform for creating engaging, interactive storymaps that can effectively communicate spatial narratives. Whether you’re documenting personal jour-

neys, presenting planning scenarios, or analyzing transportation patterns, this approach offers the flexibility and performance needed for professional applications.

The combination of R's data processing capabilities with modern web mapping technology creates opportunities for innovative visualization approaches that can enhance public engagement and decision-making processes in transportation planning and urban analytics.

0.7 Additional Resources

- `{mapgl}` package documentation
- MapLibre GL JS documentation
- OpenFreeMap styles
- Quarto documentation

This post demonstrates practical applications of open-source geospatial tools for transportation planning and data visualization. All code is available for adaptation to your own projects and use cases.