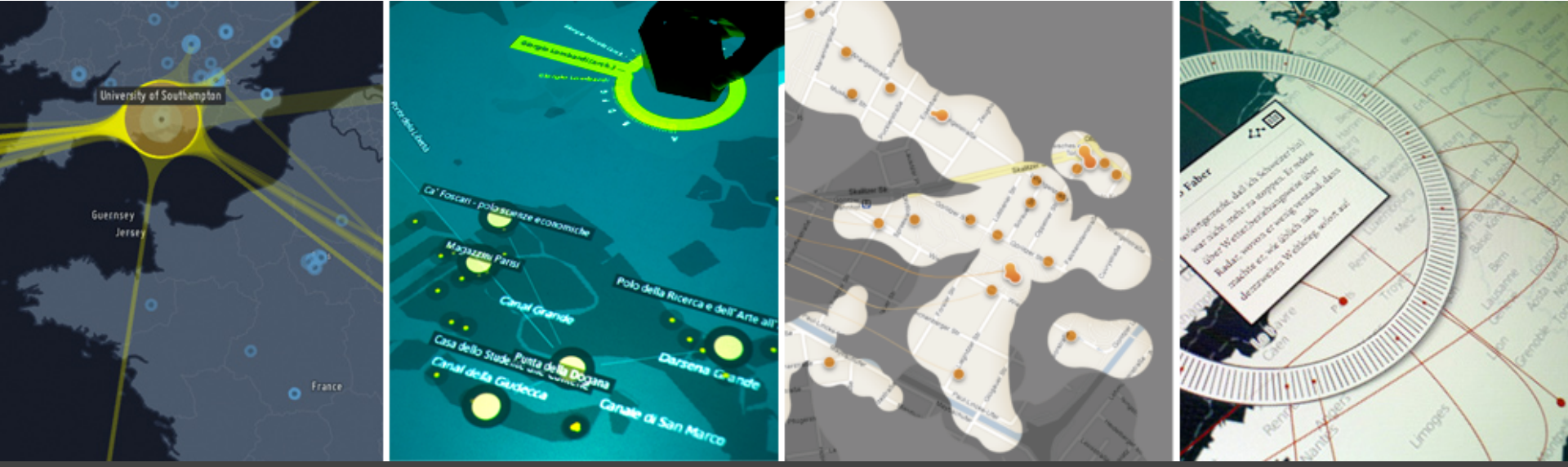


Reflektionen zur Nutzung einer GeoVis–Bibliothek in der Lehre

Dr. Till Nagel

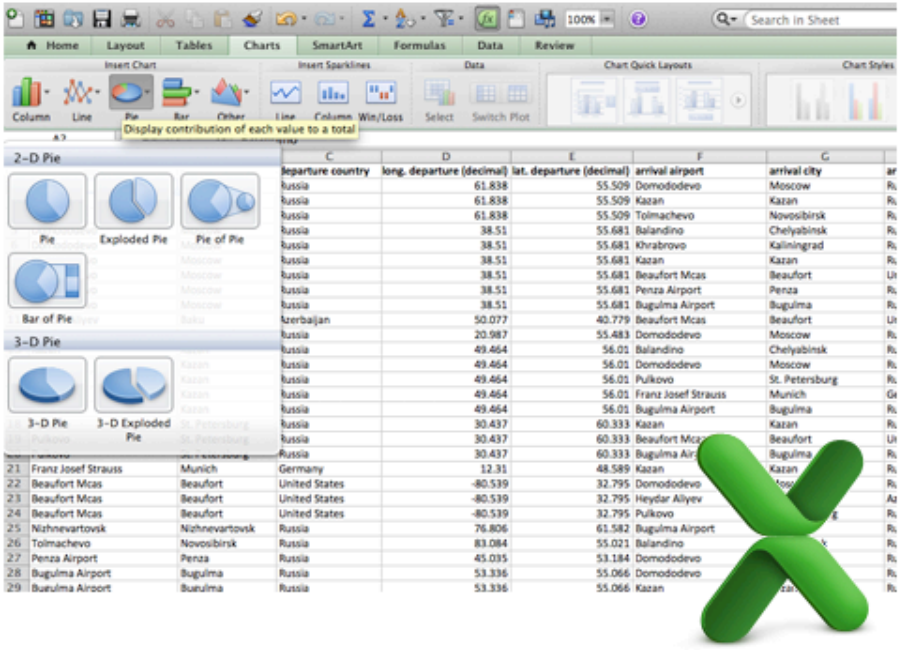
DGfK Nachwuchswissenschaftlerworkshop 2015

FHP:-) KATHOLIEKE UNIVERSITEIT LEUVEN

RQ: How to support different stakeholders to create tempo-spatial data visualizations?

Support constructing geovis

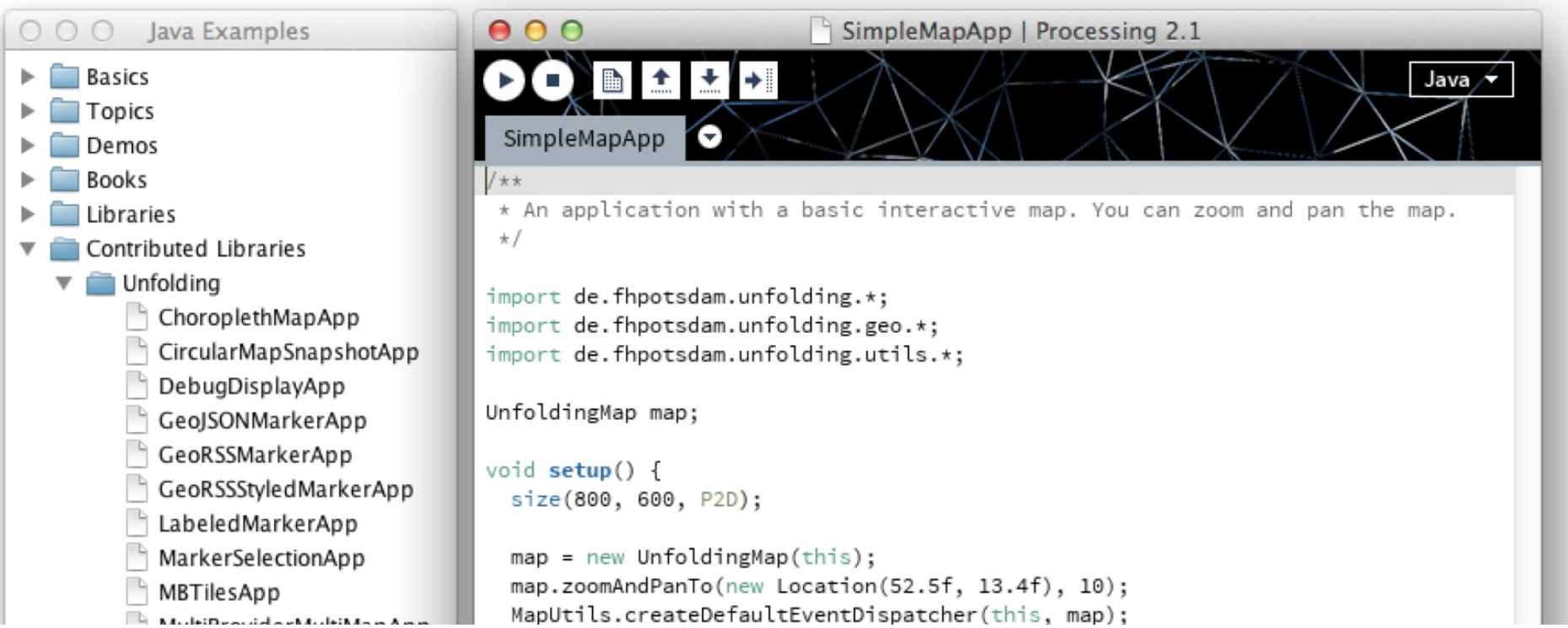
- Increasing need to design visualizations as the demand for “rapid visual data exploration” as well as for “engaging communication through custom visualizations” grows [Grammel et al. 2013]
- Ease the development of interactive maps and geovisualizations.
- Support beginners, designers, and researchers


Home
Layout
Tables
Charts
SmartArt
Formulas
Data
Review
Insert Chart
Insert Sparklines
Chart Quick Layouts
Chart Styles
Column
Line
Pie
Bar
Other
Win/Loss
Select
Switch Plot
Display contribution of each value to a total
2-D Pie
Pie
Exploded Pie
Pie of Pie
Bar of Pie
3-D Pie
3-D Pie
3-D Exploded Pie
departure country
long. departure (decimal)
lat. departure (decimal)
arrival airport
arrival city
Russia 61.838 55.509 Domodedevo Moscow
Russia 61.838 55.509 Kazan Kazan
Russia 61.838 55.509 Tolmachevo Novosibirsk
Russia 38.51 55.681 Balandino Chelyabinsk
Russia 38.51 55.681 Khrabrovo Kaliningrad
Russia 38.51 55.681 Kazan Kazan
Russia 38.51 55.681 Beaufort Mcas Beaufort
Russia 38.51 55.681 Penza Airport Penza
Russia 38.51 55.681 Bugulma Airport Bugulma
Azerbaijan 50.077 40.779 Beaufort Mcas Beaufort
Russia 20.987 55.483 Domodedevo Moscow
Russia 49.464 56.01 Balandino Chelyabinsk
Russia 49.464 56.01 Domodedevo Moscow
Russia 49.464 56.01 Pulkovo St. Petersburg
Russia 49.464 56.01 Franz Josef Strauss Munich
Russia 49.464 56.01 Bugulma Airport Bugulma
Russia 30.437 60.333 Kazan Kazan
Russia 30.437 60.333 Beaufort Mcas Beaufort
Russia 30.437 60.333 Bugulma Airport Bugulma
Germany 12.31 48.589 Kazan Kazan
Franz Josef Strauss Munich Germany 12.31
Beaufort Mcas Beaufort United States -80.539 32.795 Domodedevo
Beaufort Mcas Beaufort United States -80.539 32.795 Heydar Aliyev
Beaufort Mcas Beaufort United States -80.539 32.795 Pulkovo
Nizhnevartovsk Nizhnevartovsk Russia 76.806 61.582 Bugulma Airport
Tolmachevo Novosibirsk Russia 83.084 55.021 Balandino
Penza Airport Penza Russia 45.035 53.184 Domodedevo
Bugulma Airport Bugulma Russia 53.336 55.066 Domodedevo
Bueulma Airport Bueulma Russia 53.336 55.066 Kazan

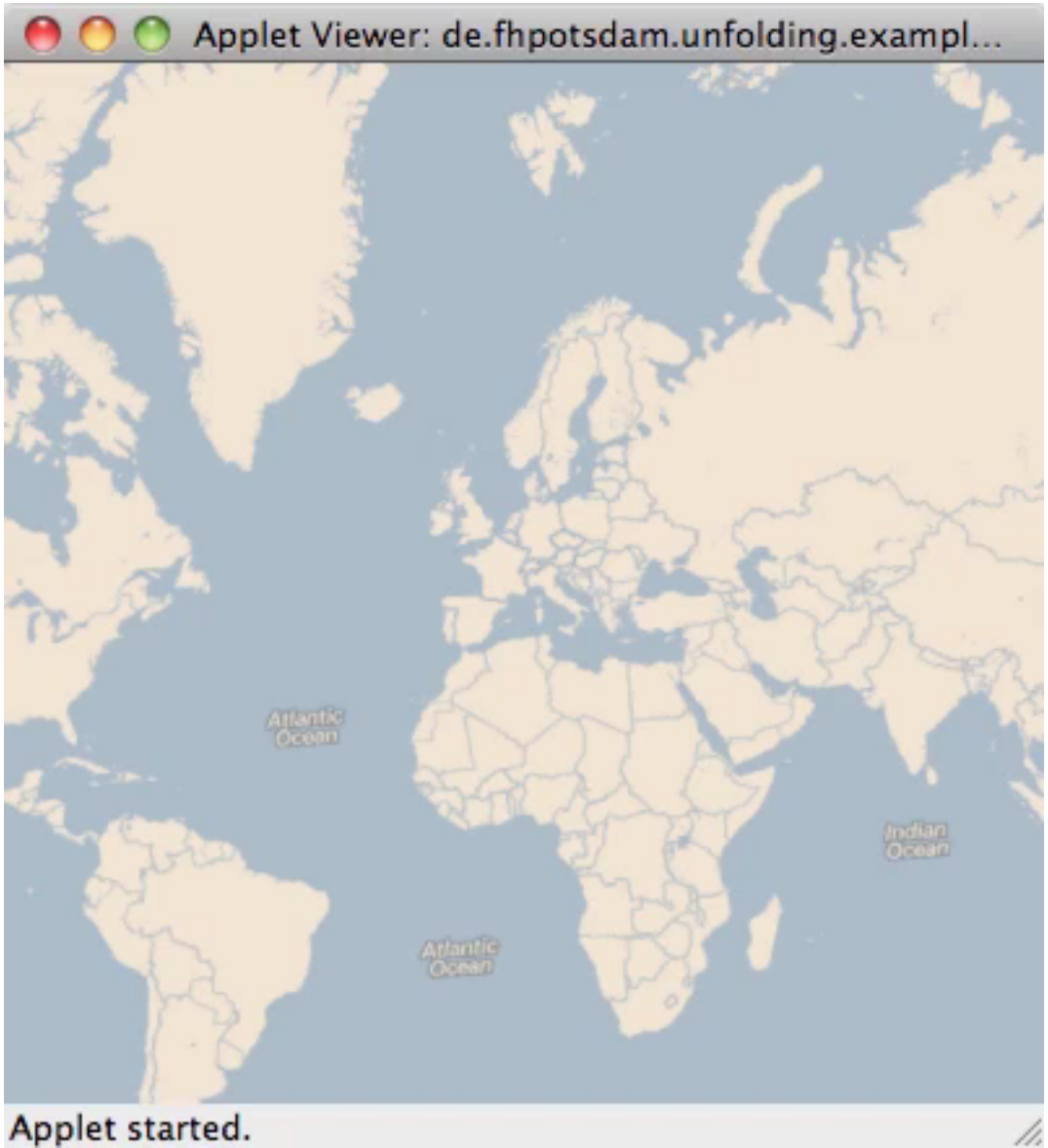

SimpleMapApp | Processing 2.1
Java
SimpleMapApp
/**
* An application with a basic interactive map. You can zoom and pan the map.
*/
import de.fhpotsdam.unfolding.*;
import de.fhpotsdam.unfolding.geo.*;
import de.fhpotsdam.unfolding.utils.*;
UnfoldingMap map;
void setup() {
size(800, 600, P2D);
map = new UnfoldingMap(this);
map.zoomAndPanTo(new Location(52.5f, 13.4f), 10);
MapUtils.createDefaultEventDispatcher(this, map);
}
void draw() {
map.draw();
}
1


Ai
Color
Color Guide
Layers
Ai

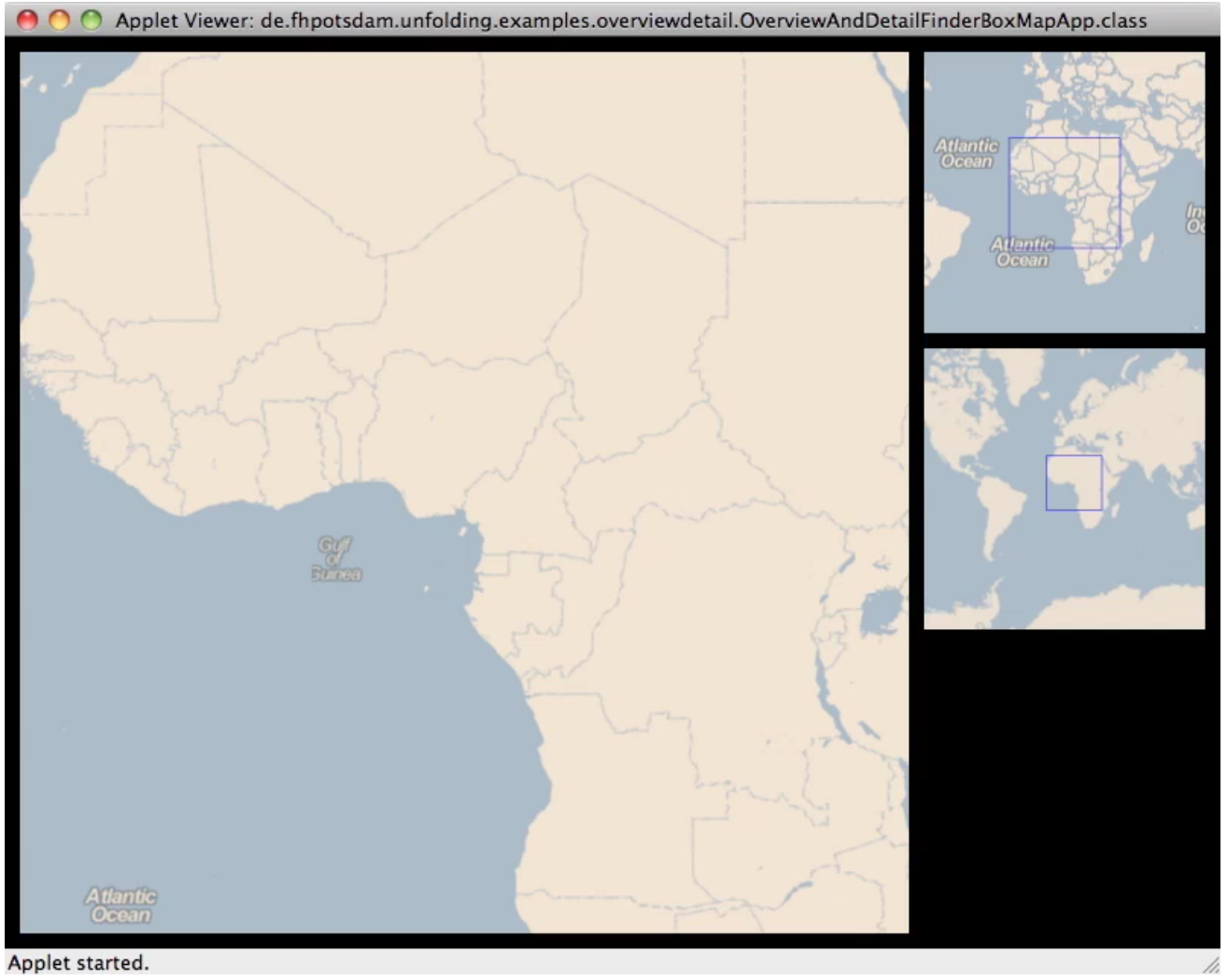
Software library



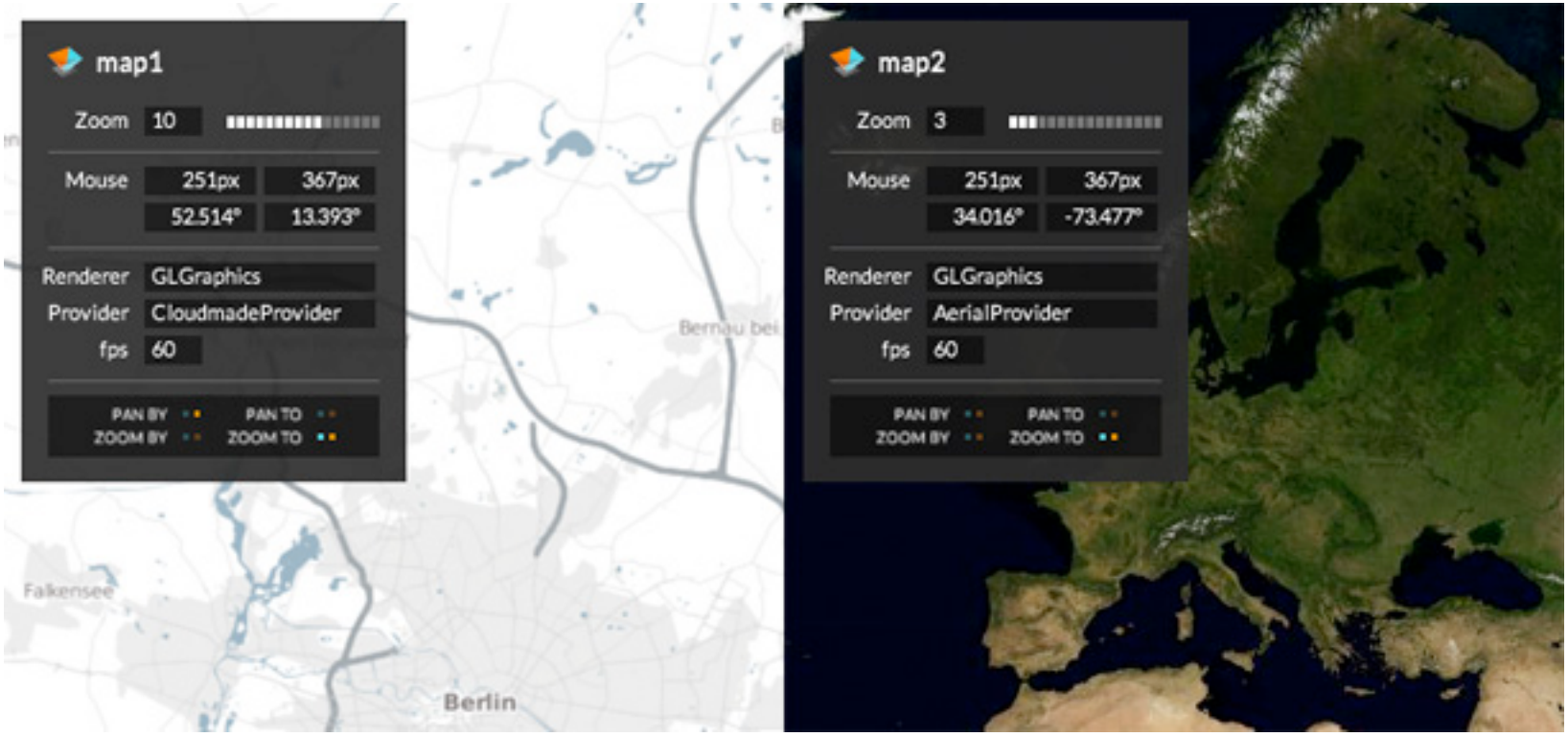
Multitouch interactions



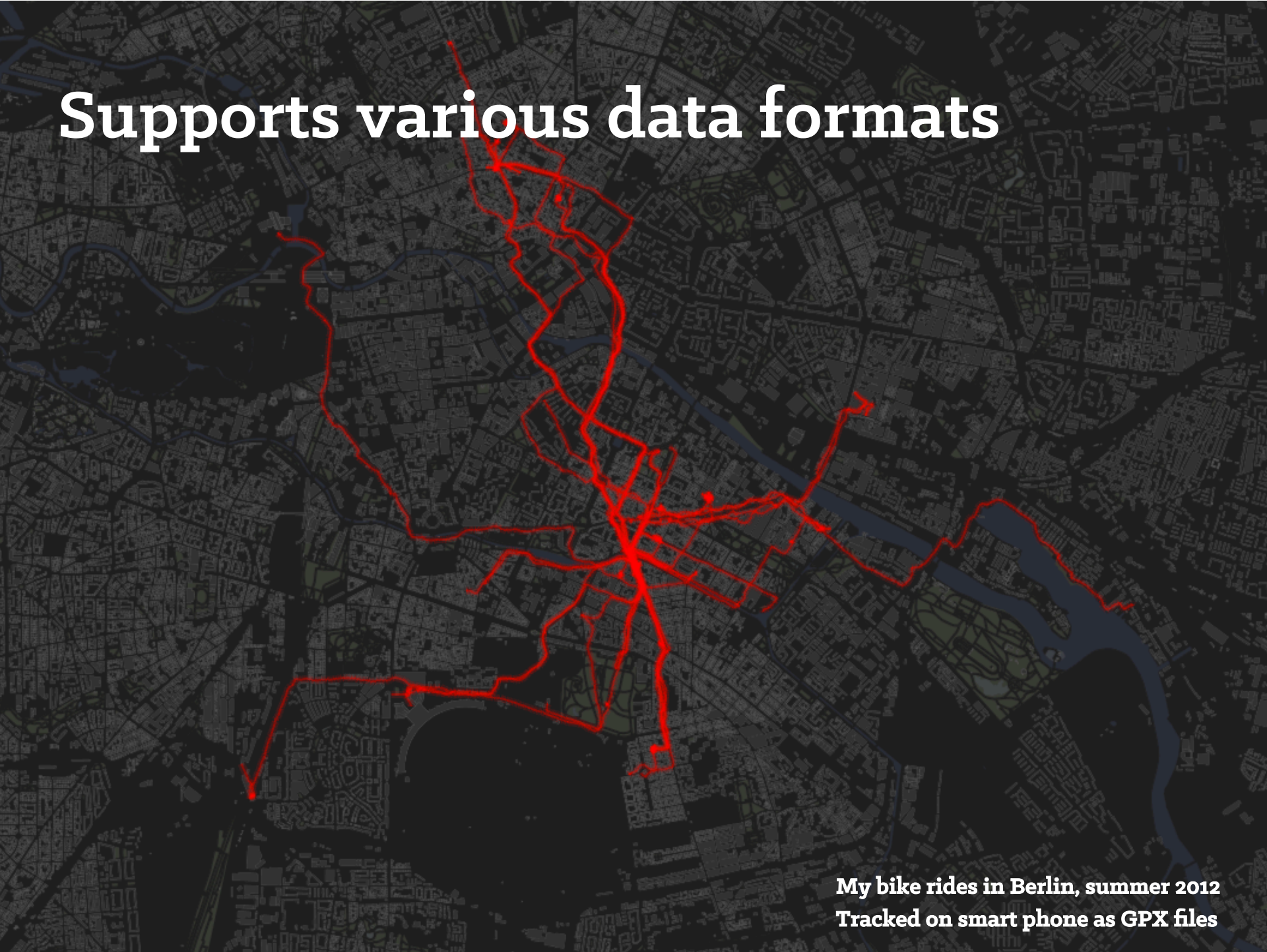
Coordinated Multiple Views



Event system



Supports various data formats



My bike rides in Berlin, summer 2012
Tracked on smart phone as GPX files

Supports various data formats

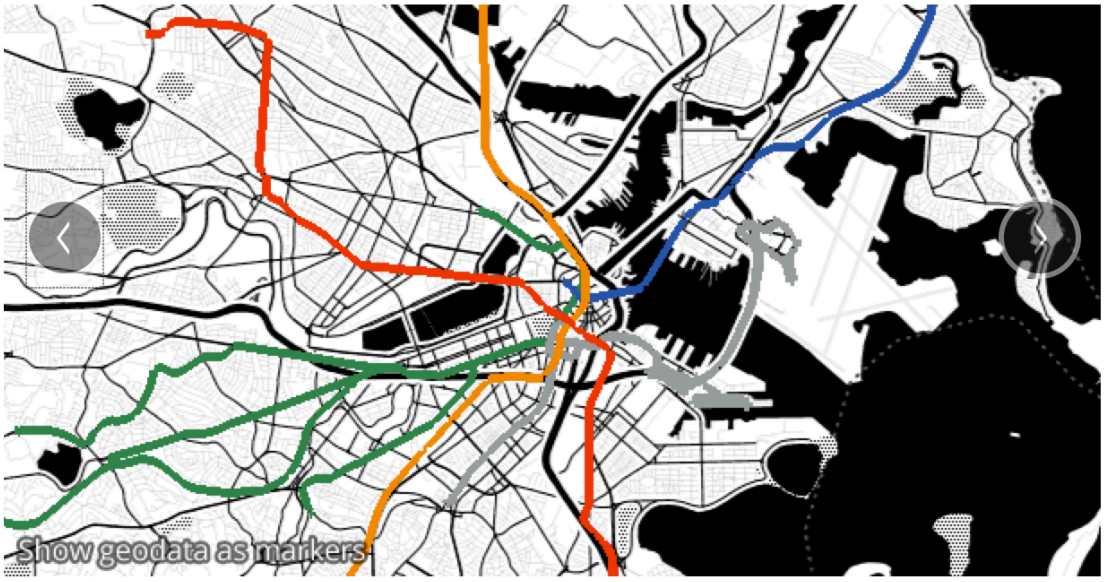
- GeoRSS
- GeoJSON
- GPX
- GTFS
- …

See tutorials for other formats such as KML, Shapefile, CSV

unfolding

Exhibition | Tutorials | Examples | API

Unfolding is a library to create interactive maps and geovisualizations in Processing and Java



Download

FOR PROCESSING 2 | FOR ECLIPSE

Or download more packages with examples, the jar only, versions for Processing 1.5, etc.

Getting Started

Its really easy to make your own interactive maps using Unfolding. Check out these tutorials to kick-start your application.

START IN PROCESSING | START IN ECLIPSE

Features

Interaction Events

Unfolding enables you to quickly create interactive maps. Basic interactions such as

Data Visualization

Simply create geo-positioned markers to display data on a map. The visual style can be

Tutorials & Examples

unfolding

How to use Unfolding

Prerequisite: You already have installed Unfolding for your programming environment. (Otherwise, do it now: Unfolding for Processing or Unfolding for Eclipse.)

Let's begin with our basic Unfolding sketch.

```
1  UnfoldingMap map;
2
3  void setup() {
4      size(800, 600, GLConstants.GLGRAPHICS);
5      map = new UnfoldingMap(this);
6      MapUtils.createDefaultEventDispatcher(this, map);
7  }
8
9  void draw() {
10     map.draw();
11 }
```

Geolocations and screen positions

You can easily convert a screen position to a location, and vice versa. As an example, let's display the geo-position of the mouse pointer.

Here, we get the Location of the map at the current mouse position, and show its latitude and longitude as black text.

```
1  void draw() {
2      map.draw();
3      Location location = map.getLocation(mouseX, mouseY);
4      fill(0);
5      text(location.getLat() + ", " + location.getLon(), mouseX, mouseY);
6  }
```

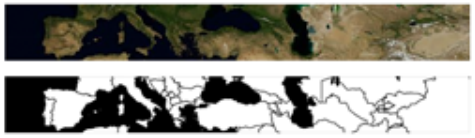
Map styles

Unfolding displays maps in a default style, with cartographic data from OpenStreetMaps and tiles from Cloudmade. To use another map style, simply specify it as second parameter when constructing an UnfoldingMap.

```
1  map = new UnfoldingMap(this, new Microsoft.AerialProvider());
```

(Don't forget importing `de.fhpotsdam.unfolding.providers.*`)

This way, you can easily switch to one of the pre-configured map tile providers. To see the different map styles, go to the MapProvider & Tiles tutorial. There you'll also find how to create your own map provider, and even how to create a completely new map style.



Keep in mind you need to check the terms and conditions of the map providers on how you are allowed to use their map tiles. We are providing the example providers for educational purposes, only.

Zooming and panning the map

By creating the default event dispatcher (as shown above), users already can interact with your map. They can pan the map by dragging it with the mouse, or by using the arrow keys on the keyboard. Using the mouse wheel zooms in or out, which also works by pressing + or - keys. Double-clicking on the map centers it around that location, and zooms in one level.

Now, let's say you want to focus your visualization on a city. Manually set the location and zoom level in the setup() method.

```
1  map.zoomAndPanTo(new Location(52.5f, 13.4f), 10);
```

Here, we pan to Berlin and zoom to a level users can see the whole city area.

You might want to restrict the map interactions, for instance because you only have data for a specific area. For that, we create a Location for the city, and use it to center the map (as before), but we are using that Location also as center for the panning restriction.

```
1  Location berlinLocation = new Location(52.5f, 13.4f);
```

unfolding

Markers

Displaying markers on a map is very straight-forward. Just create a marker with a location and add it to the map once. Here, for instance, we are creating two point markers, one for Berlin and one for Dublin.

```
1  Location berlinLocation = new Location(52.5, 13.4);
2  Location dublinLocation = new Location(53.35, -6.26);
3
4  // Create point markers for locations
5  SimplePointMarker berlinMarker = new SimplePointMarker(berlinLocation);
6  SimplePointMarker dublinMarker = new SimplePointMarker(dublinLocation);
7
8  // Add markers to the map
9  map.addMarkers(berlinMarker, dublinMarker);
```

They will be drawn automatically on top of the map, always at the correct position.

Unfolding provides a default marker style, and has point, line, and polygon markers out of the box.

Style your markers

All default markers provide some very basic styling methods, e.g. to set stroke and fill colors.

```
1  // Adapt style
2  berlinMarker.setColor(color(255, 0, 0, 100));
3  berlinMarker.setStrokeColor(color(255, 0, 0));
4  berlinMarker.setStrokeWeight(4);
```

For more sophisticated marker customization or for creating data glyphs, there are two options:

- Drawing it yourself
- Creating own marker class (advanced)

The easiest method to create a custom style is to draw the marker yourself instead of adding it to the map.

For this you need to make the marker global, i.e. define the variable first in your sketch (here: line 2), then assign a new marker in setup (line 11). In draw() get the current position of the marker (line 19), and draw some visual representation with Processing's drawing functions (lines 20-23).

```
1  UnfoldingMap map;
2  SimplePointMarker berlinMarker;
3
4  void setup() {
5    size(800, 600, GLConstants.GLGRAPHICS);
6
7    map = new UnfoldingMap(this);
8    MapUtils.createDefaultEventDispatcher(this, map);
9
10   Location berlinLocation = new Location(52.5, 13.4);
11   berlinMarker = new SimplePointMarker(berlinLocation);
12
13   // Do not add marker to the map
14 }
15
16 void draw() {
17   map.draw();
18
19   ScreenPosition berlinPos = berlinMarker.getScreenPosition(map);
20   strokeWeight(16);
21   stroke(67, 213, 227, 100);
22   noFill();
23   ellipse(berlinPos.x, berlinPos.y, 36, 36);
24 }
```

`marker.getScreenPosition(map)` returns the current x,y-position of the marker on the map. The method converts the geo-location of that marker to the current `ScreenPosition` on the canvas of the sketch.

Now, you can style your markers in any way you want. In the following example we draw the marker as two arcs with a text label.

unfolding

Markers & Data 2 - Other data sources

Loading and displaying geospatial data from *CSV*, databases, etc.

Combine geo-spatial data with other data sources

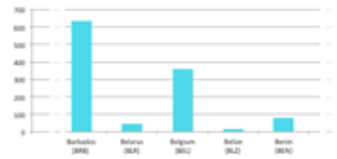
When you have data from multiple sources, you need to load and combine them via some ID or other unique value in both data sets.

As an example, let's combine geometry data with population density data. The country shapes are loaded from a GeoJSON file via a data reader, while the population density values come from an external CSV file.

The GeoJSON includes:


```
1  {"type":"Feature","id":"BEL","properties":{"name":"Belgium"},"geometry":{"type":"Polygon"
```

The CSV file includes:



```
1  Belgium;BEL;359.8414795
```

(Note the *BEL* for Belgium as ID in GeoJSON and as second value in the CSV. This is the three-letter country code.)

First, we load all country polygons as markers (line 9), and all data from the CSV into a hashmap with country codes as keys (line 14).

```
1  List<Marker> countryMarkers;
2  HashMap<String, DataEntry> dataEntriesMap;
3
4  void setup() {
5      size(800, 600, GLConstants.GLGRAPHICS);
6      map = new UnfoldingMap(this);
7
8      // Load country polygons and adds them as markers
9      List<Feature> countries = GeoJSONReader.loadData(this, "countries.geo.json");
10     countryMarkers = MapUtils.createSimpleMarkers(countries);
11     map.addMarkers(countryMarkers);
12
13     // Load population data
14     dataEntriesMap = loadPopulationDensityFromCSV("population-density.csv");
15 }
```

In order to find matching population values for each country, we use the country code existing in both files to lookup a CSV value for the current marker (lines 3-4 below)

```
1  for (Marker marker : countryMarkers) {
2      // Find data for country of the current marker
3      String countryId = marker.getId();
4      DataEntry dataEntry = dataEntriesMap.get(countryId);
5
6      // Encode value to transparency (value range: 0-700)
7      float alpha = map(dataEntry.value, 0, 700, 10, 255);
8      marker.setColor(color(255, 0, 0, alpha));
9  }
```

This example visualizes population density of the world as a choropleth map. Countries are shaded in proportion to the population density. The data value is encoded to transparency via a simplistic linear mapping (line 7 above).

Task Areas

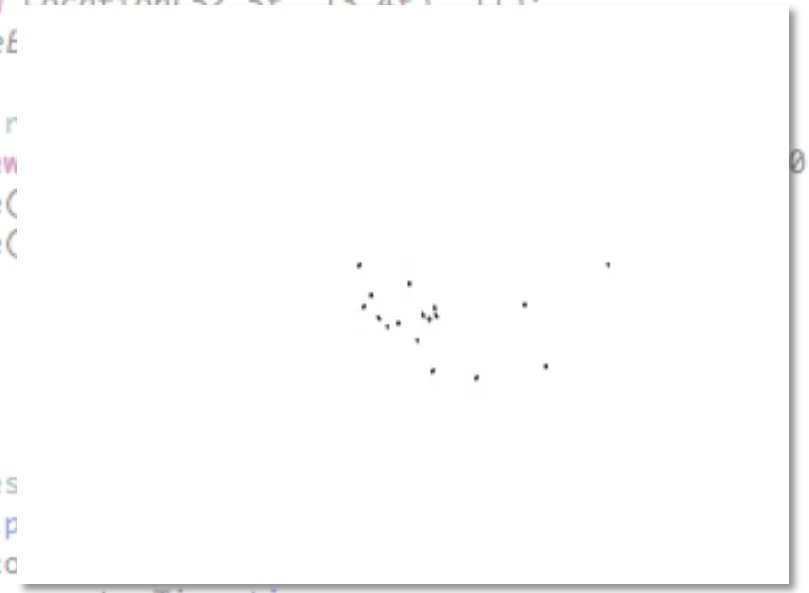
- **Learning**
  Learn to display geo-spatial data.
- **Prototyping**
  Try out new ideas in the design process.
- **Creating**
  Design projects for audience use or evaluation.

Hello World, literally

```
UnfoldingMap map = new UnfoldingMap(this);

map.draw();
```

Explore data sets

```
public void setup() {
    size(1200, 700, GLConstants.GLGRAPHICS);

    // Load trips from CSV files
    DataUtils dataUtils = new DataUtils(this);
    trips = dataUtils.loadAllTrips();

    // Map: Creates interactive map, and centers around Berlin
    map = new UnfoldingMap(this);
    map.zoomAndPanTo(new Location(52.5f, 13.4f), 11);
    MapUtils.createMouseE

    // UI: Creates time r
    timeRangeSlider = new                                    0, 16,
            new DateTime(
            new DateTime(
}

public void draw() {
    map.draw();

    // Shows all vehicles
    for (Trip trip : trip
        for (StopTime sto
            DateTime time = stopTime.time;
            if (time.isAfter(currentStartDateTime) && time.isBefore(currentEndDateTime)) {
                ScreenPosition pos = map.getScreenPosition(stopTime.stop.location);
                ellipse(pos.x, pos.y, 6, 6);
            }
        }
    }

    timeRangeSlider.draw();
}
```



```
// Data: Loads time table from GTFS files
DataUtils dataUtils = new DataUtils(this);
trips = dataUtils.loadAllTrips();
```

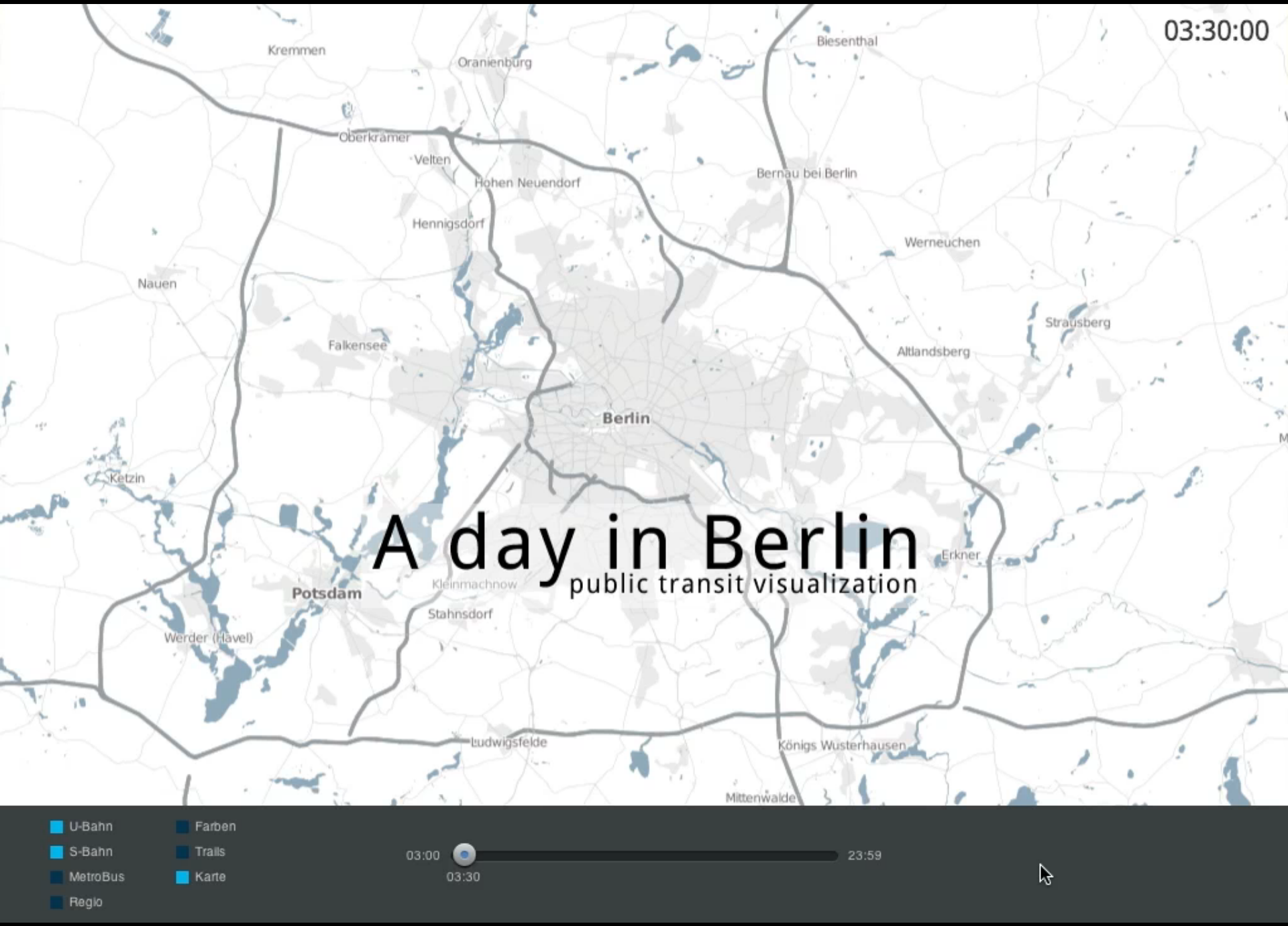
```
List<Trip> trips;

UnfoldingMap map;

TimeRangeSlider timeRangeSlider;
DateTime startTime;
DateTime endTime;

public void setup() {
    size(1200, 700, GLConstants.GLGRAPHICS);

    // Data: Loads time table from GTFS files
    DataUtils dataUtils = new DataUtils(this);
    trips = dataUtils.loadAllTrips();

    // Map: Creates interactive map, and centers around Berlin
    map = new UnfoldingMap(this);
    map.zoomAndPanTo(new Location(52.5f, 13.4f), ZOOM_LEVEL_CITY);
    MapUtils.createMouseEventDispatcher(this, map);

    // UI: Creates time range slider (4am until midnight)
    timeRangeSlider = new TimeRangeSlider(this, 200, 740, 300, 16,
            new DateTime(2011, 12, 10, 4, 0, 0),
            new DateTime(2011, 12, 10, 23, 59, 0), 60);
}

public void draw() {
    map.draw();

    // Shows all vehicles of all trips at current time
    for (Trip trip : trips) {
        for (StopTime stopTime : trip.stopTimes) {
            DateTime time = stopTime.time;
            if (time.isAfter(startTime) && time.isBefore(endTime)) {
                Location location = stopTime.stop.location;
                ScreenPosition pos = map.getScreenPosition(location);
                ellipse(pos.x, pos.y, 6, 6);
            }
        }
    }

    timeRangeSlider.draw();
}
```

```
// Map: Creates interactive map, and centers around Berlin
map = new UnfoldingMap(this);
map.zoomAndPanTo(new Location(52.5f, 13.4f), ZOOM_LEVEL_CITY);
MapUtils.createMouseEventDispatcher(this, map);
```

```
List<Trip> trips;

UnfoldingMap map;

TimeRangeSlider timeRangeSlider;
DateTime startTime;
DateTime endTime;

public void setup() {
    size(1200, 700, GLConstants.GLGRAPHICS);

    // Data: Loads time table from GTFS files
    DataUtils dataUtils = new DataUtils(this);
    trips = dataUtils.loadAllTrips();

    // Map: Creates interactive map, and centers around Berlin
    map = new UnfoldingMap(this);
    map.zoomAndPanTo(new Location(52.5f, 13.4f), ZOOM_LEVEL_CITY);
    MapUtils.createMouseEventDispatcher(this, map);

    // UI: Creates time range slider (4am until midnight)
    timeRangeSlider = new TimeRangeSlider(this, 200, 740, 300, 16,
            new DateTime(2011, 12, 10, 4, 0, 0),
            new DateTime(2011, 12, 10, 23, 59, 0), 60);
}

public void draw() {
    map.draw();

    // Shows all vehicles of all trips at current time
    for (Trip trip : trips) {
        for (StopTime stopTime : trip.stopTimes) {
            DateTime time = stopTime.time;
            if (time.isAfter(startTime) && time.isBefore(endTime)) {
                Location location = stopTime.stop.location;
                ScreenPosition pos = map.getScreenPosition(location);
                ellipse(pos.x, pos.y, 6, 6);
            }
        }
    }

    timeRangeSlider.draw();
}
```

```
if (time.isAfter(startTime) && time.isBefore(endTime)) {
    Location location = stopTime.stop.location;
    ScreenPosition pos = map.getScreenPosition(location);
    ellipse(pos.x, pos.y, 6, 6);
}
```

```
List<Trip> trips;

UnfoldingMap map;

TimeRangeSlider timeRangeSlider;
DateTime startTime;
DateTime endTime;

public void setup() {
    size(1200, 700, GLConstants.GLGRAPHICS);

    // Data: Loads time table from GTFS files
    DataUtils dataUtils = new DataUtils(this);
    trips = dataUtils.loadAllTrips();

    // Map: Creates interactive map, and centers around Berlin
    map = new UnfoldingMap(this);
    map.zoomAndPanTo(new Location(52.5f, 13.4f), ZOOM_LEVEL_CITY);
    MapUtils.createMouseEventDispatcher(this, map);

    // UI: Creates time range slider (4am until midnight)
    timeRangeSlider = new TimeRangeSlider(this, 200, 740, 300, 16,
            new DateTime(2011, 12, 10, 4, 0, 0),
            new DateTime(2011, 12, 10, 23, 59, 0), 60);
}

public void draw() {
    map.draw();

    // Shows all vehicles of all trips at current time
    for (Trip trip : trips) {
        for (StopTime stopTime : trip.stopTimes) {
            DateTime time = stopTime.time;
            if (time.isAfter(startTime) && time.isBefore(endTime)) {
                Location location = stopTime.stop.location;
                ScreenPosition pos = map.getScreenPosition(location);
                ellipse(pos.x, pos.y, 6, 6);
            }
        }
    }

    timeRangeSlider.draw();
}
```

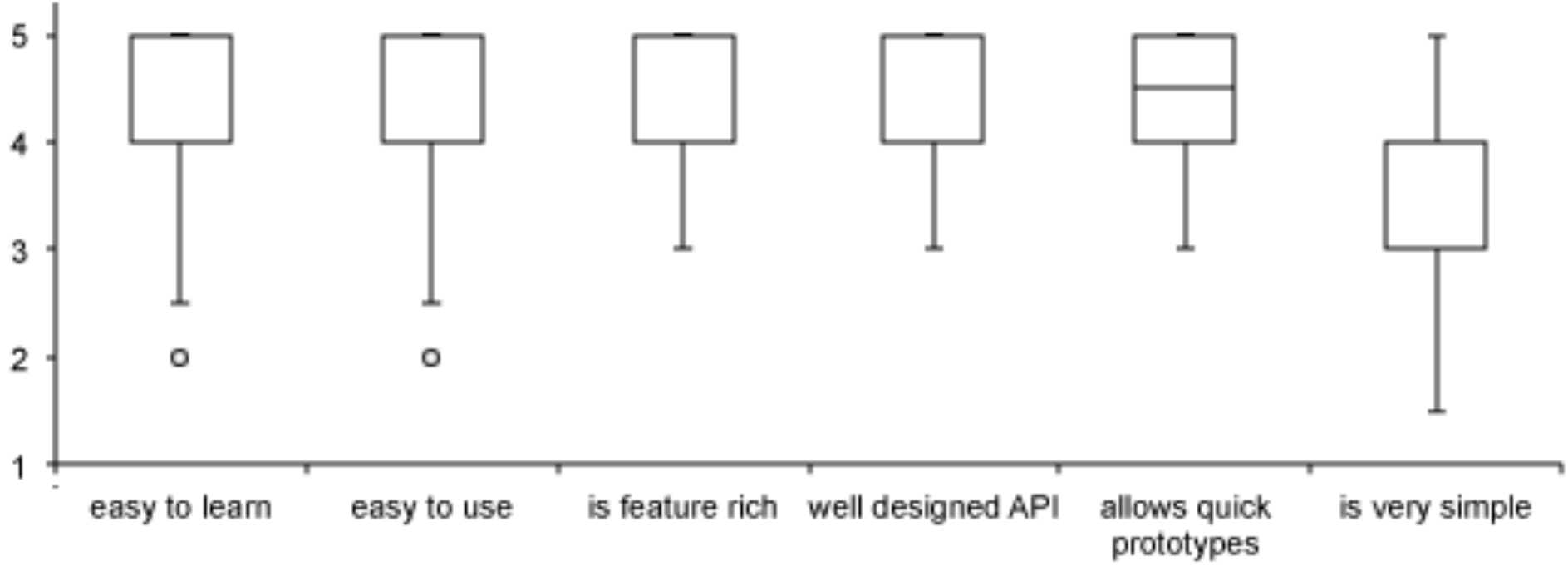

03:30:00
Kremmen
Oranienburg
Biesenthal
Oberkrämer
Velten
Hohen Neuendorf
Bernau bei Berlin
Hennigsdorf
Werneuchen
Nauen
Strausberg
Falkensee
Altlandsberg
Berlin
Ketzin
A day in Berlin
public transit visualization
Erkner
Potsdam
Kleinmachnow
Stahnsdorf
Werder (Havel)
Ludwigsfelde
Königs Wusterhausen
Mittenwalde
U-Bahn
S-Bahn
MetroBus
Regio
Farben
Trails
Karte
03:00
23:59
03:30

User Survey

- Online questionnaire
- 37 participants
- Based on ISO to evaluate software & SUS

- Asked about
  - Expertise
  - Projects & Features
  - Satisfaction

Satisfaction with Unfolding



See details in chapter

User Survey Results

- Satisfied with basic features, e.g. displaying maps (97%), enabling zoom + pan (91%), etc.
- Fewer satisfied with more advanced features, e.g. loading geo-spatial data (54%)
- Satisfied with examples (62%) and tutorials (53%)

See details in paper

User Survey Results

- Satisfied with Unfolding (91%)
- Achieved what they planned in project (81%)
- Plan to use Unfolding in the future (88%)

Impact

Unfolding Software Library

Downloads		*Version*	*Months*	
3,000	×	0.8	08/11–08/12	250
6,000	×	0.9	09/12–07/13	545
6,100	×	0.9.3	08/13–06/14	554
2,200	×	0.9.5	07/14–10/14	550
17,300				

Source: Github, Google Code, Amazon S3, Google Analytics. Rounded to the next hundred.

Citations: Unfolding

Unfolding - A Library for Interactive Maps

5 citations total

Source: Google Scholar, Oct 2015
Google Analytics, Slideshare Stats

Citations: Unfolding

Unfolding - A Library for Interactive Maps

5 citations total

http://unfoldingmaps.org

17 citations total

Source: Google Scholar, Oct 2015
Google Analytics, Slideshare Stats

Citations: Unfolding

Unfolding - A Library for Interactive Maps

5 citations total

http://unfoldingmaps.org

17 citations total

Website: 53,700 visitors

Presentations: 42,000 views

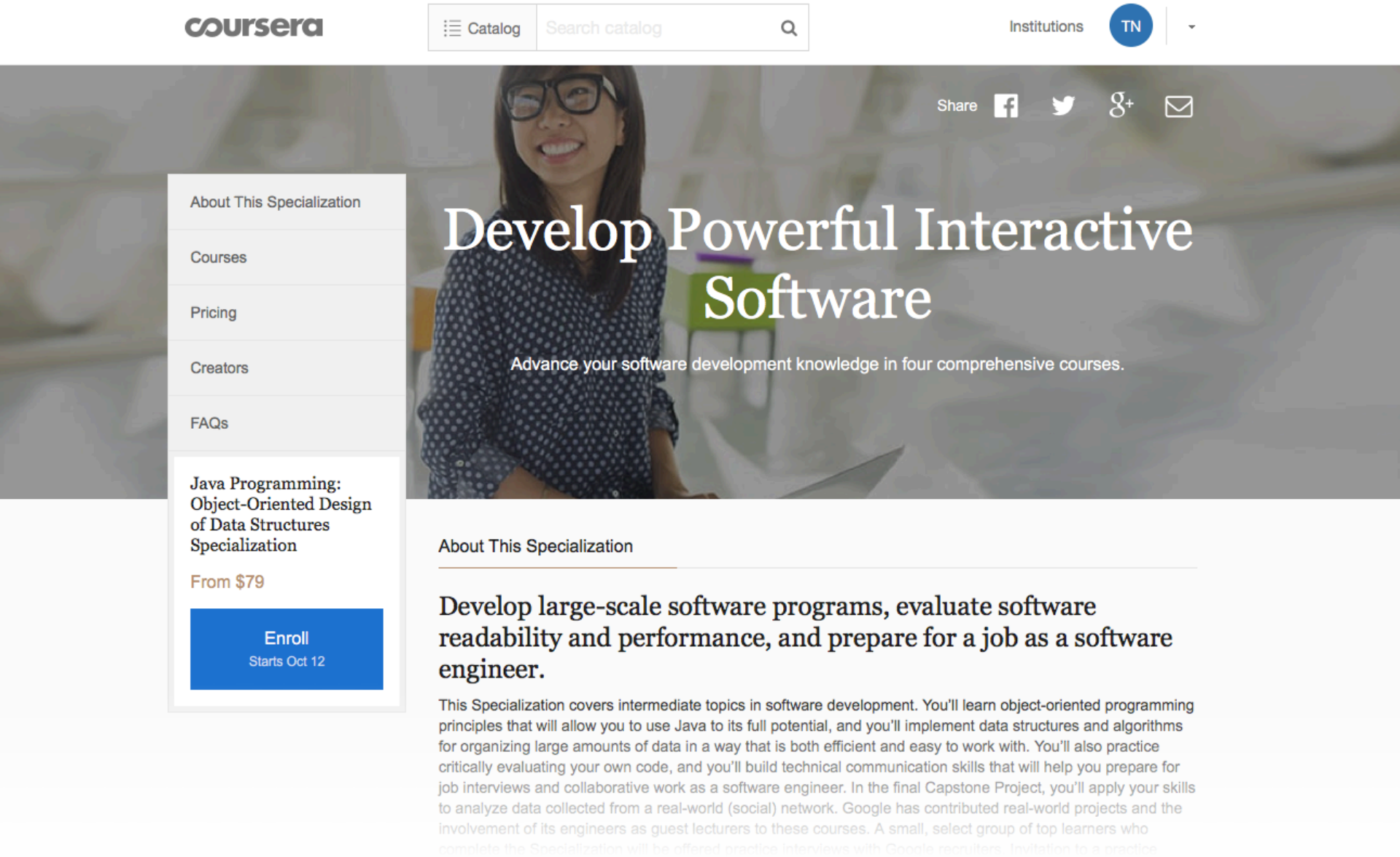
Used in university courses at

FHP, KUL, IUAV, RCA, HMS, …

MIT, UCLA, CMU, NYU, ITP, HfK, Cambridge, …

Source: Google Scholar, Oct 2015
Google Analytics, Slideshare Stats

Major tool in MOOC



Unfolding Software Library

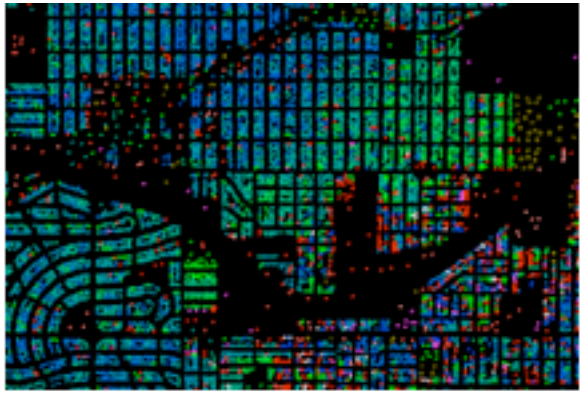
"the true measure of the toolkit's value lies in the creation [...] of successful visualizations by others"

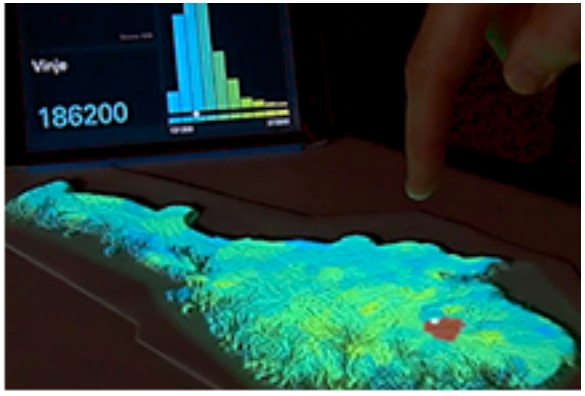
Bostock, M., Heer, J. Protovis: A graphical toolkit for visualization. IEEE TVCG 15 (6), 2009, pp. 1121–1128.

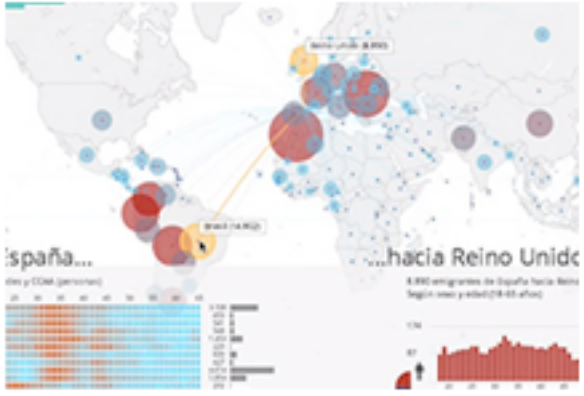
Unfolding Software Library

"the true measure of the toolkit's value lies in the creation [...] of successful visualizations by others"

Bostock, M., Heer, J. Protovis: A graphical toolkit for visualization. IEEE TVCG 15 (6), 2009, pp. 1121–1128.



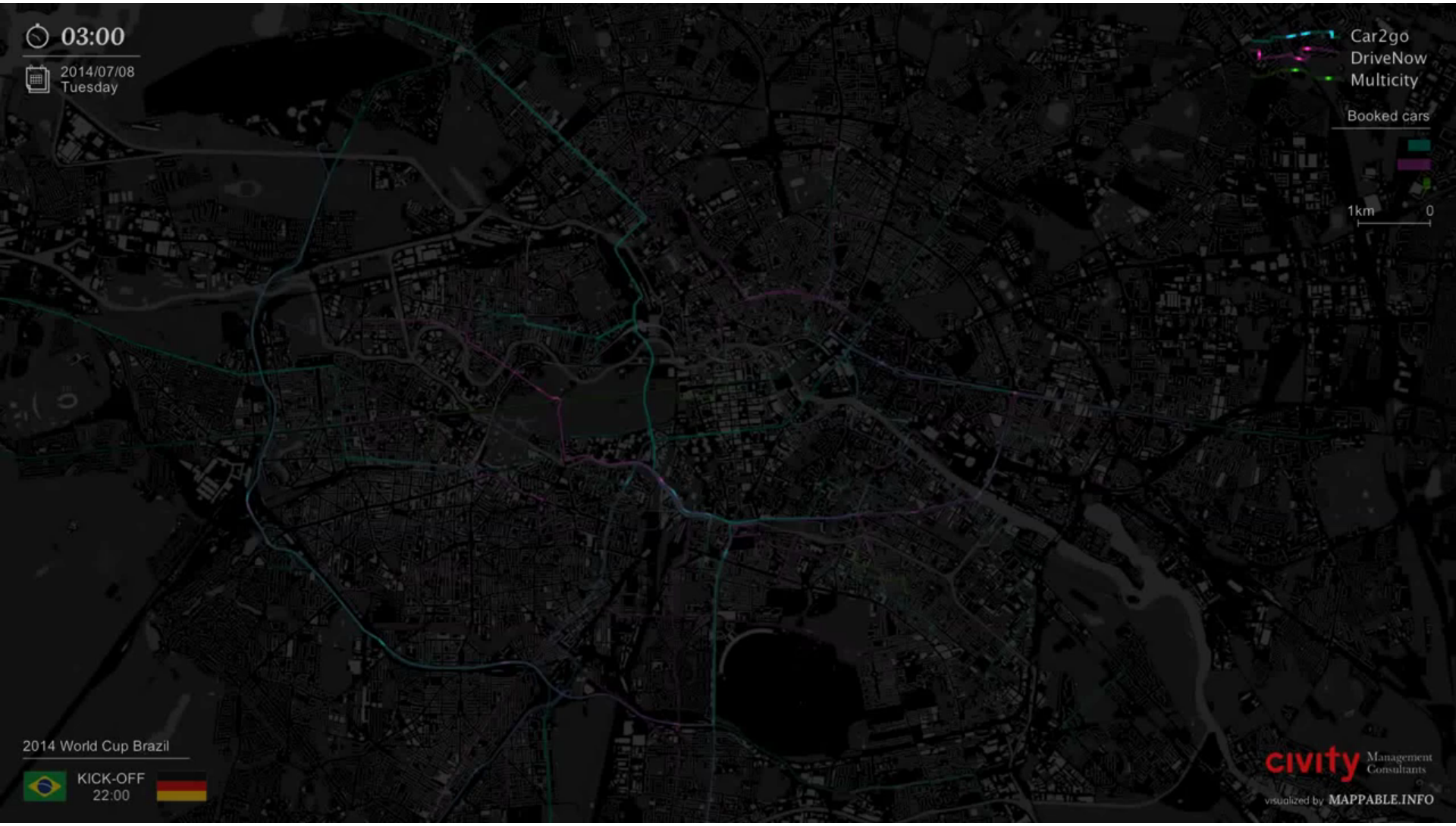




visualizing.org **Visualizing**
@VisualizingOrg

Following

Visualizing 2014: A look at the year's major events and trends through #dataviz invent.ge/13AOTTS


03:00
2014/07/08
Tuesday
Car2go
DriveNow
Multicity
Booked cars
1km
0
2014 World Cup Brazil
KICK-OFF
22:00
civity Management Consultants
visualized by MAPPABLE.INFO

Contributions

- Supports a diverse set of users, and eases developing visualizations of geo-referenced data.
- An effective means to create state-of-the-art geovisualizations.

Fragen

Fragen

- Welche Werkzeuge setzt ihr in der Lehre ein?
- Was sind eure Erfahrungen mit der Veröffentlichung von Bibliotheken?
- Wie lange unterstützt man ein OpenSource-Projekt?
- Sinnvoll, verschiedene Nutzergruppen zu unterstützen?
- Dokumentation vs Beispiele?

Vielen Dank.

Dr. Till Nagel

DGfK Nachwuchswissenschaftlerworkshop 2015