isySUR

it's easy!

Manual

Published January 4, 2015

Contents

1. Introduction to isySUR

Welcome to isySUR (pronounce isy as ['i:zi]), a tool helping you to find the sphere of influence of a space usage rule. For starting the command line version please see section 1.1.1. For using the window version please read section 1.1.2. You can also run the window version on your android device. This is described in section 2.3.3. For more details about the program see section 1.2.

1.1 Quick start

First decide if you like to use the command line version of isySUR or if you prefer a graphical user interface. Please proceed with section 1.1.1 or section 1.1.2 depending on your choice.

1.1.1 isySUR in command line version

Explain how to quick start

1.1.2 isySUR in window version

Explain how to quick start

1.2 What is isySUR?

isySURis a program that calculates the sphere of influence of a space usage rule by its coordinates. It was developed within the informatiCup, a programming competition for students organized by the German Informatics Society (Gesellschaft für Informatik e.V.). Our team is called isy and consists of Adriana-Victoria Dreyer, Jacqueline Hemminghaus, Jan Pöppel and Thorsten Schodde, four students of the master study course 'intelligent systems' at Bielefeld University.

using OSM

Features: why to use this software

1.3 Content of isySUR

In the archive you can find the following data (bold printed files will be installed with the install command):

noch up-to-date?

add GUI when finished

- **isySUR** (directory) - the source files for the isySUR Python package
  - tests (directory) - isySUR was developed using test driven development. In this directory you will find the test files.
- testData (directory) - in here one possible input file named Data.txt is found
  - dataOnlyForTests (directory) - Some of the test files need data that is stored in this directory
- kmlComperator.py - comparison tool for comparing our computed .kml-files with the given .truth.kml-files (the areas of the placemarks), used for development
- manual.pdf - this manual
- README.txt
- **run_isySUR.py** - the Python script to run the command line version of isySUR
- run_tests.py - script that runs all tests in isySUR/tests/, used for development
- setup.py - the setup script to install isySUR
- surConfig.cfg - a configuration file that can optionally be used for computation (see section 2.3)

onieren die pat- eigentlich noch?

2. Using isySUR

In this chapter the usage of isySUR is described in detail. First you can find installation instructions in section 2.1, detailed usage instruction for both program versions are following in section 2.3.

2.1 Requirements and installation

isySUR is written in Python so you get a Python script, which is the main program, and a package with the tools. For running the command line version you need not much more than basic Python. If you prefer to use a graphical user interface you will have to install some more packages.

2.1.1 Requirements

1. **requirements for the command line version:**
   - Python 2.7
   - requests (HTTP library)
   - internet connection

   To run a Python script you need Python. Python comes with most Linux distributions. For the use on Windows machines you will have to install Python manually[1]. isySUR uses data from OpenStreetMap[2] to calculate the sphere of influence. Therefore an internet connection is required. The data transfer in Python is realised with the requests[3] library. You can easily install it using pip[4], a Python installation tool

   ```
   $ pip install requests
   ```

   or see requests web page[3].
2. **additional requirements for the GUI version:**
   - requirements for GUI

[1] get Python here: https://www.python.org
[2] http://www.openstreetmap.org
[3] http://docs.python-requests.org
[4] https://pip.pypa.io

2.1.2 Installation

Unterschiede bei GUI?

You got an archive of type .tar.gz. Please unpack the archive into a directory you want. Because isySUR is a Python script you do not need to install it. It is also possible to install only the dependencies and run isySUR through typing:

```
$ python run_isySUR.py [parameters]
```

For installing the script and tool package browse the directory you chose and install isySUR with help of Python:

```
$ python setup.py install
```

Now you are able to start isySUR anytime through

```
$ run_isySUR.py [parameters]
```

2.2 Input and output

The input of the space usage rules was specified in the programming task as well as the output in .kml format. For more information about the decision to add a configuration file please see section 3.1.

2.2.1 Space usage rules

The SUR file is a plain text file. The first line holds the number of space usage rules in the file. Line-serially the SURs follow. Each of this lines starts with the id of the SUR. Then latitude and longitude follow and the last entry is the name or description of the SUR. The entries are comma separated. Decimal separator in latitude and longitude is a point.

Example file:

```
1
0072, 50.9313, 5.39570, smoking="no"
```

2.2.2 Configuration file

The configuration file is a plain text file. In this file one or more blocks starting with '[Indoor]', '[Outdoor]' or '[Both]' are found. After these headlines the names of the SURs follow that belongs to these categories of area of influence.

Example file:

```
[Indoor]
access:age="21+"
camera="no"
[Outdoor]
fishing="no"
littering="no"
[Both]
access:dog="no"
open_fire="no"
```

2.2.3 Output: Keyhole Markup Language

Output files are in .kml format where kml stands for Keyhole Markup Language. For syntax definition and documentation please visit `https://developers.google.com/kml`.

2.3 Command line and GUI version

isySUR comes with two version: One for command line use and another one for users who prefer to use a graphical user interface.

2.3.1 Command line version

Usage of the command line version:

```
$ run_isySUR.py [-h] [-c CONFIG] input output
```

Parameters

up to date?

The parameters of run_isySUR.py (bold arguments are required):

- **input** - Path to the input file containing the SURs. See 2.2 for the required format.
- **output** - Path where to put the output file(s). If this path points to a file, a single output .kml-file containing all computed areas is created. If it points to a directory, one .kml-file for each SUR is created in this directory as well as one .kml-file containing all computed areas.
- -h - If this parameter is given, a short help message containing all parameters is shown.
- -c CONFIG (–config CONFIG) - With this parameter an optional config file with SUR classifications (indoor, outdoor, both) is used for computation. For the format of the file see 2.2.

2.3.2 GUI version

fill me!

2.3.3 Using android

fill me!

3. Technical and implementation details

Berechnen Sie nun die Geltungsbereiche der Space Usage Rules aus Ihrer Umgebung mit Ihrer Implementierung aus der ersten Runde. Diskutieren Sie die Korrektheit und die Präzision der von Ihnen berechneten Lösungen für die Space Usage Rules des Testdatensatzes und denen aus Ihrer Umgebung. Vergleichen Sie dazu die berechneten mit den intuitiv beabsichtigten Geltungsbereichen.

create sections

3.1 The configuration file

why do we have it?

4. Added SURs

For the competition ten new space usage rules should be found.

Add our SURs (maybe with screenshot of intuitive area?)

Todo list