GetOSM

Release 0.1.2

Huidae Cho

May 21, 2022

TABLE OF CONTENTS

GetOSM is an OpenStreetMap downloader written in Python that is agnostic of GUI frameworks. It is used with tkinter by ProjPicker.

CHAPTER

ONE

REQUIREMENTS

GetOSM uses the following standard Python modules:

- sys
- math
- urllib.request

CHAPTER

TWO

INSTALLATION

GetOSM is available at https://pypi.org/project/getosm/.

```
pip3 install getosm
```

CHAPTER

THREE

DEMO GUIS

3.1 osmtk: tkinter demo GUI

osmtk.py

3.2 osmwx: wxPython demo GUI

osmwx.py

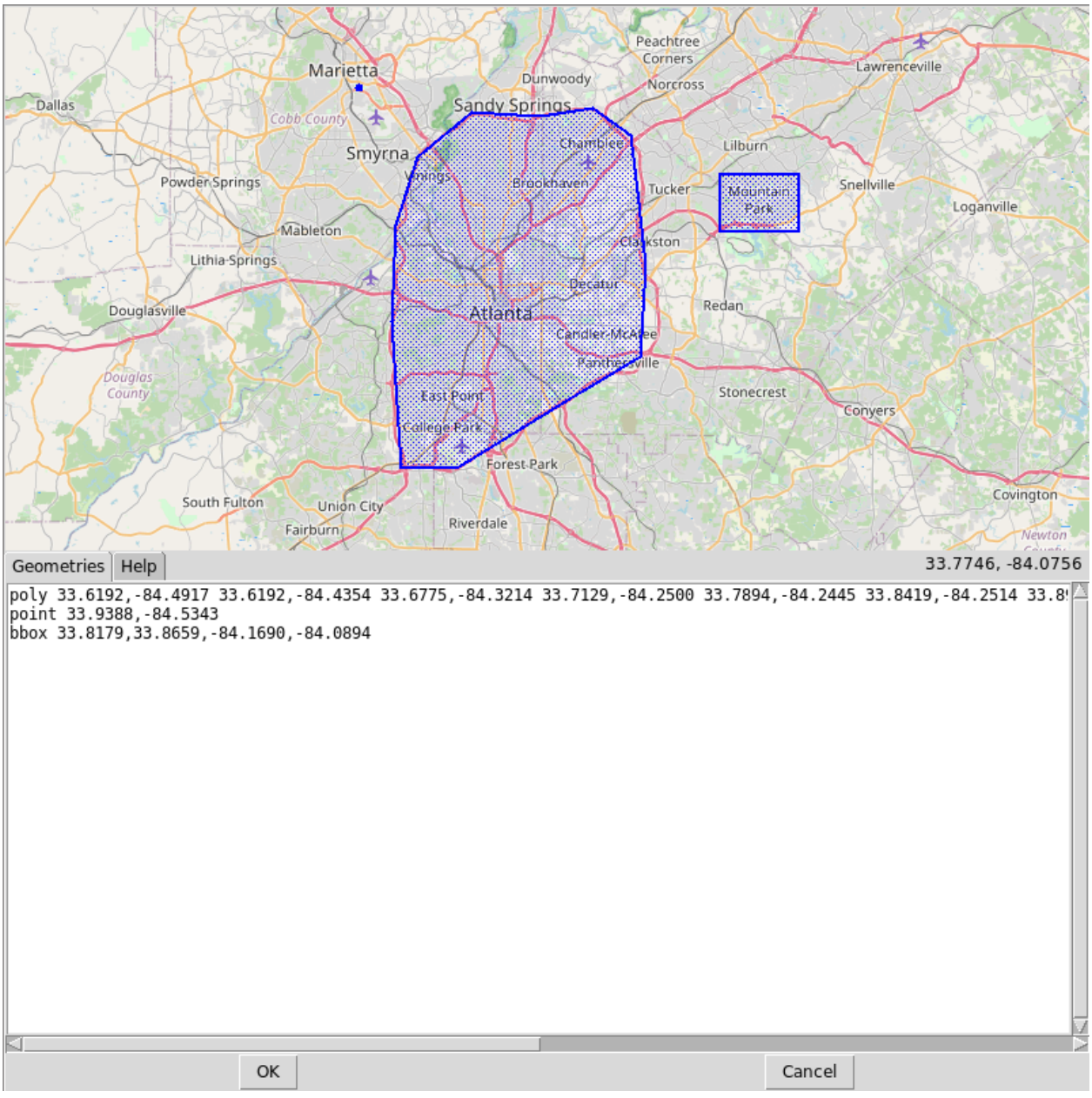

Fig. 1: osmtk: tkinter demo GUI

CHAPTER

FOUR

LICENSE

CHAPTER

FIVE

GETOSM MODULE

This module provides an OpenStreetMap tile downloader.

class getosm.**CachedTile**(*image*, *raw*)

Provide the data structure for cached tiles. Initially, when a tile is first downloaded in OpenStreetMap.download_tile(), its raw data is stored in the image attribute and the raw flag is set to True. Later, when the GUI framework needs to draw the tile on the canvas in OpenStreetMap.draw(), it needs to convert the raw data to its own native image object. This converted image object is stored back to the image attribute and the raw flag is now set to False. There is no way to go back to the original raw data.

class getosm.**OpenStreetMap**(*create_image*, *draw_image*, *create_tile*, *draw_tile*, *resample_tile*, *width=256*, *height=256*, *lat=0*, *lon=0*, *z=0*, *verbose=False*)

Provide the public-facing API for downloading, dragging, zooming, and coordinate conversions.

canvas_to_latlon(*x*, *y*)

Convert canvas x and y in pixels to latitude and longitude in decimal degrees in the current map centered at self.lat,self.lon at the zoom level self.z. Input x and y don't need to be ints (see latlon_to_canvas()). They start from the upper-left corner at 0,0 on the canvas.

Parameters

- **x** (*float*) – Canvas x starting from the left at x=0 growing towards the right by 1 for each pixel.
- **y** (*float*) – Canvas y starting from the top at y=0 growing towards the bottom by 1 for each pixel.

Returns Latitude and longitude in decimal degrees.

Return type float, float

download(*lat*, *lon*, *z*)

Download all tiles needed to cover the entire canvas centered at latitude and lonitude at the zoom level z. Downloaded tiles are saved in self.cached_tiles (see download_tile()). The function returns whether or not the download session was canceled by the main thread by setting self.cancel to True.

Parameters

- **lat** (*float*) – Latitude in decimal degrees.
- **lon** (*float*) – Longitude in decimal degrees.
- **z** (*int*) – Zoom level.

Returns Whether or not the download session was canceled externally by the main thread by setting the cancel attribute to True. It's the responsibility of the main thread to reset the cancel attribute to False. When the user scrolls the mouse wheel continuously to zoom faster,

the main thread needs to cancel any previous download sessions to save data traffic and CPU time.

Return type bool

`download_tile`(*x*, *y*, *z*)

Download the tile at tile x,y (see latlon_to_tile()) at the zoom level z and return its key in z/x/y. These x and y must be ints because they are used to construct a URL for tile downloading. The raw data of the tile is stored in self.cached_tiles with its key z/x/y (see the CachedTile class). The function returns the tile key.

Parameters

- **x** (*int*) – Tile x starting from lon=-180 growing towards the east by 1 for every tile.
- **y** (*int*) – Tile y starting from lat=85.0511 growing towards the south by 1 for each tile.
- **z** (*int*) – Zoom level.

Returns URL for the tile at tile x,y at the zoom level z.

Return type str

`drag`(*x*, *y*, *draw=True*)

Drag the map by x-self.drag_x and y-self.drag_y, and download necessary tiles using self.download(). The location at x,y follows the mouse cursor. By default, it draws the map using self.draw(). However, if draw is False, it only downloads tiles and doesn't draw the map. It returns x-self.drag_x and y-self.drag_y.

Parameters

- **x** (*float*) – Canvas x starting from the left at x=0 growing towards the right by 1 for each pixel.
- **y** (*float*) – Canvas y starting from the top at y=0 growing towards the bottom by 1 for each pixel.
- **draw** (*bool*) – Whether or not to draw the map. Defaults to True.

Returns Drag amounts in x and y in pixels.

Return type float, float

`draw`()

Draw cached tiles stored in tiles on the canvas by calling callback functions including self.create_image(), self.create_tile(), self.draw_tile(), and self.draw_image() (see the constructor). This function converts raw tile data to the GUI framework's native image format and sets its raw flag to False.

`draw_rescaled`()

Draw rescaled tiles stored in self.rescaled_tiles on the canvas by calling callback functions including self.create_image(), self.create_tile(), self.draw_tile(), self.resample_tile(), and self.draw_image() (see the constructor). This function may convert raw tile data to the GUI framework's native image format using self.create_tile() and sets its raw flag to False.

`get_bbox_xy`(*bbox*)

Converts a list of south, north, west, and east in decimal degrees to a list of canvas upper-left and lower-right corner points in pixels. The output is in [[left, top], [right, bottom]].

Parameters **bbox** (*list*) – List of south, north, west, and east in decimal degrees.

Returns List of canvas upper-left and lower-right corner points in pixels in [[left, top], [right, bottom]].

Return type list

`get_tile_url`(*x*, *y*, *z*)

Get the URL for the tile at tile x,y (see latlon_to_tile()) at the zoom level z. These x and y must be ints because they are used to construct a URL for tile downloading.

Parameters

- **x** (*int*) – Tile x starting from lon=-180 growing towards the east by 1 for every tile.
- **y** (*int*) – Tile y starting from lat=85.0511 growing towards the south by 1 for each tile.
- **z** (*int*) – Zoom level.

Returns URL for the tile at tile x,y at the zoom level z.

Return type str

`get_xy`(*latlon*)

Converts a list of lists of latitude and longitude in decimal degrees to a list of lists of canvas x and y in pixels. The input latlon must be in [[lat, lon], [lat, lon], ...].

Parameters **latlon** (*list*) – List of lists of latitude and longitude in decimal degrees.

Returns List of lists of canvas x and y floats (see latlon_to_canvas()) in pixels.

Return type list

`grab`(*x*, *y*)

Set self.grab_x and self.grab_y to x and y, respectively. This function is used to signal the start of dragging events. self.drag() uses these two attributes to calculate the x and y deltas of a dragging event. Both x and y are mostly ints because they are canvas coordinates in pixels, but they can also be floats.

Parameters

- **x** (*float*) – Canvas x starting from the left at x=0 growing towards the right by 1 for each pixel.
- **y** (*float*) – Canvas y starting from the top at y=0 growing towards the bottom by 1 for each pixel.

`latlon_to_canvas`(*lat*, *lon*)

Convert latitude and lonngitude to canvas x and y in pixels in the current map centered at self.lat,self.lon at the zoom level self.z. These x and y are pixel coordinates on the canvas starting from the upper-left corner at 0,0. For now, these are floats to keep their precision, but it can change in the future.

Parameters

- **lat** (*float*) – Latitude in decimal degrees.
- **lon** (*float*) – Longitude in decimal degrees.

Returns Canvas x and y in pixels starting from the upper-left corner at x,y=0,0 growing towards the right and bottom by 1 for each pixel.

Return type float, float

`latlon_to_tile`(*lat*, *lon*, *z*)

Convert latitude and lonngitude to tile x and y at the zoom level z. Tile x and y are not ints, but they are floats to be able to tell locations within a tile. To convert them to x and y for downloading the tile, type cast them to int. These int x and y would correspond to the upper-left cornder of the tile. Both x and y increase by 1 when we move from one tile to the next. In other words, x and y are not pixel coordinates, but they are rather the number or fractional number of tiles starting from latitude 85.0511 and longitude -180 at the given zoom level.

Parameters

- **lat** (*float*) – Latitude in decimal degrees.
- **lon** (*float*) – Longitude in decimal degrees.
- **z** (*int*) – Zoom level.

Returns Tile x, y starting from lon,lon=-180,85.0511 growing towards the east and south by 1 for each tile.

Return type float, float

message(**args*, *end=None*)

Print args to stderr immediately if self.verbose is True.

Parameters

- **args** (*str*) – Arguments to print. Passed to print().
- **end** (*str*) – Passed to print(). Defaults to None.

redownload()

Provide a shortcut for self.download(self.lat, self.lon, self.z). This function can be used to redownload already downloaded tiles as its name suggests or to download tiles for the first time when self.lat, self.lon, and self.z are already set.

repeat_xy(*xy*)

Repeat canvas xy points in pixels across the antimeridian. Unlike the latitude axis, the longitude axis crosses the antimeridian from west to east or from east to west. When the map repeats itself more than once horizontally, geometries also need to be repeated. This function is used to repeat xy points just enough times to cover the entire canvas. The xy points must be in [[x, y], [x, y], ...]. The return list is in [[[x, y], [x, y], ...], [[x, y], [x, y], ...], ...]. same list format.

Parameters **xy** (*list*) – List of lists of canvas x and y in pixels.

Returns List of lists of lists of canvas x and y in pixels.

Return type list

rescale(*x*, *y*, *dz*, *draw=True*)

Rescale the map by the delta zoom level dz relative to x,y and download necessary tiles using self.download() if the map is rescaled. The delta zoom level dz is a float that is accumulated in self.dz. Once self.dz reaches 1 or -1, rescaling starts. The location at x,y stays at the same location x,y. By default, it draws the map using self.draw(). However, if draw is False, it only computes necessary parameters and downloads tiles without drawing the map. In this case, self.draw_rescaled() needs to be called to actually rescale and draw the tiles. It returns whether or not the map was rescaled.

Parameters

- **x** (*float*) – Canvas x starting from the left at x=0 growing towards the right by 1 for each pixel.
- **y** (*float*) – Canvas y starting from the top at y=0 growing towards the bottom by 1 for each pixel.
- **dz** (*float*) – Delta zoom level.
- **draw** (*bool*) – Whether or not to draw the map. Defaults to True.

Returns Whether or now the map was rescaled.

Return type bool

reset_zoom()

Reset the delta zoom level self.dz to restart a zooming event.

resize(*width*, *height*)

Resize the canvas and redownload tiles at the same location and zoom level.

Parameters

- **width** (*int*) – New canvas width in pixels.
- **height** (*int*) – New canvas height in pixels.

tile_to_latlon(*x*, *y*, *z*)

Convert tile x,y to latitude and longitude at the zoom level z. Tile x and y don't have to be ints (see latlon_to_tile()) and are not pixel coordinates. The are the number or fractional number of tiles starting from latitude 85.0511 and longitude -180 at the given zoom level.

Parameters

- **x** (*float*) – Tile x starting from lon=-180 growing towards the east by 1 for every tile.
- **y** (*float*) – Tile y starting from lat=85.0511 growing towards the south by 1 for each tile.
- **z** (*int*) – Zoom level.

Returns Latitude and longitude in decimal degrees.

Return type float, float

zoom(*x*, *y*, *dz*, *draw=True*)

Zoom the map by the delta zoom level dz relative to x,y and download necessary tiles using self.download() if the map is zoomed. The delta zoom level dz is a float that is accumulated in self.dz. Once self.dz reaches 1 or -1, zooming starts. The location at x,y stays at the same location x,y. By default, it draws the map using self.draw(). However, if draw is False, it only downloads tiles and doesn't draw the map. It returns whether or not the map was zoomed.

Parameters

- **x** (*float*) – Canvas x starting from the left at x=0 growing towards the right by 1 for each pixel.
- **y** (*float*) – Canvas y starting from the top at y=0 growing towards the bottom by 1 for each pixel.
- **dz** (*float*) – Delta zoom level.
- **draw** (*bool*) – Whether or not to draw the map. Defaults to True.

Returns Whether or now the map was zoomed.

Return type bool

zoom_to_bbox(*bbox*, *draw=True*)

Zoom the map to the given bounding box bbox in south, north, west, and east in decimal degrees. South must be less than north, but west may not be less than east if the end point of the bbox is located to the west of the start point. In this reversed west and east case, the bbox becomes the NOT of the bbox that would be formed if west and east were switched. By default, it draws the map using self.draw(). However, if draw is False, it only downloads tiles and doesn't draw the map. In either case, it does not draw the bbox. It returns whether or not the map was zoomed. The function returns south, north, and west as is, and west plus the delta longitude as east so that east is always greater than west. In this case, east can be greater than 180.

Parameters

- **bbox** (*list*) – List of south, north, west, and east floats in decimal degrees.

- **draw** (*bool*) – Whether or not to draw the map. Defaults to True.

Returns South, north, west, and west plus the delta longitude in decimal degrees.

Return type float, float, float, float

class getosm.**Tile**(*key*, *x*, *y*, *z*)

Provide the referencing data structure for tiles. The key attribute is used to find cached raw tile data in OpenStreetMap.cached_tiles, a dictionary of keys to CachedTile objects. The x and y attributes store the location of the tile in pixels relative to the upper-left corner of the canvas. These and z attributes are known when a new tile is first downloaded and its location is computed by OpenStreetMap.download(). However, x and y can change when the tile's rescaling parameters are precomputed by OpenStreetMap.rescale(). One of the rescaling parameters is the delta zoom level dz which indicates how many zoom levels the tile should be rescaled from its original zoom level z. OpenStreetMap.draw_rescaled() actually rescales the tile and stores its image in the rescaled_image attribute. Only key and z never change once they are set. Initially, dz is set to 0 and rescaled_image to None, meaning that the original raw tile is not rescaled. Once the tile is rescaled x, y, dz, and rescaled_image are updated.

CHAPTER

SIX

INDICES AND TABLES

- genindex
- modindex
- search

PYTHON MODULE INDEX

INDEX