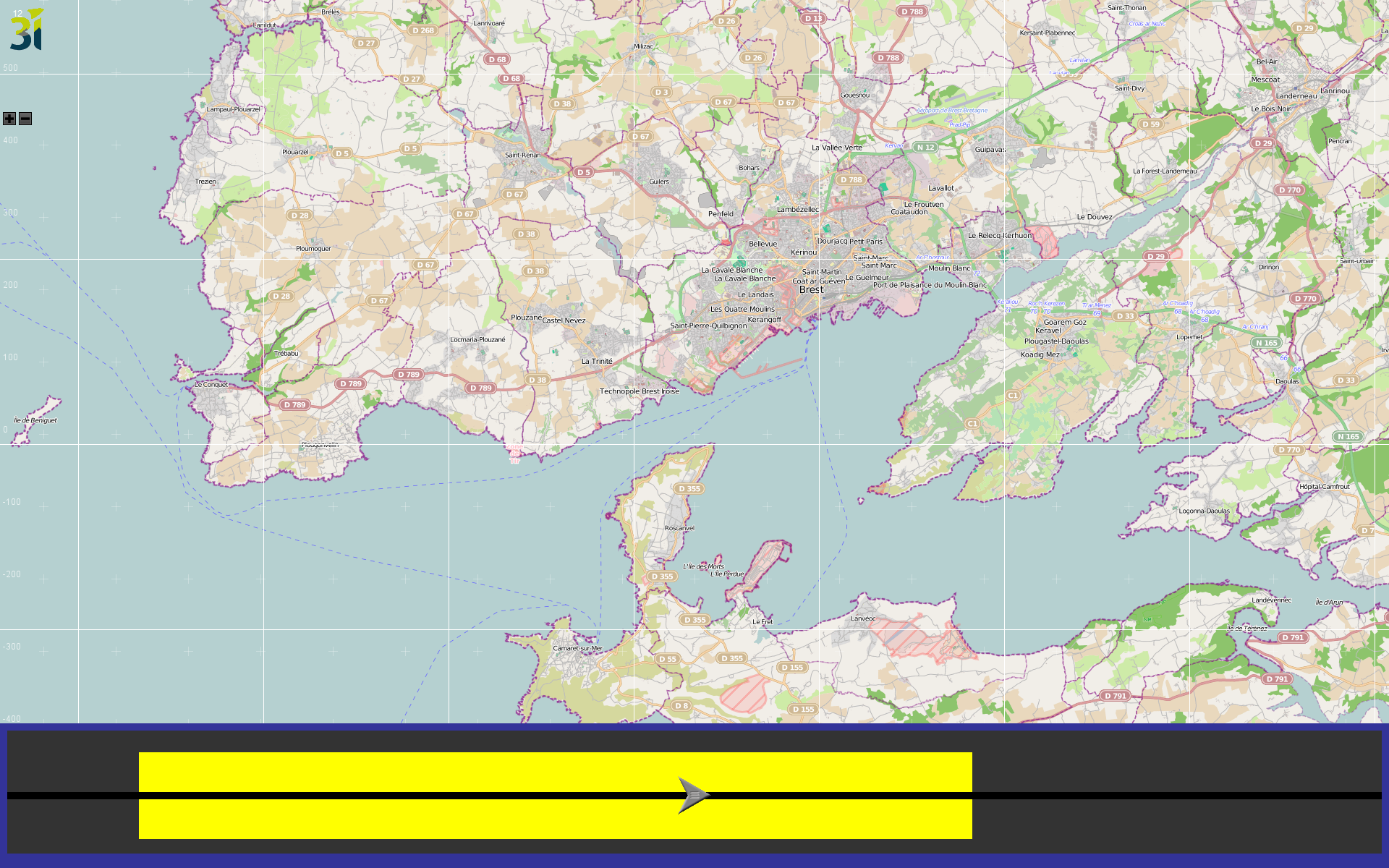
Intégration de la cartographie OpenStreetMap  
dans le projet *uav3i*

***Philippe TANGUY***

*Document créé le 24 septembre 2013*

*Dernière modification le 25 septembre 2013*



L’intégration des cartes *OpenStreetMap* se fait par l’intermédiaire du composant graphique Open Source JMapViewer. Le wiki du projet est hébergé à l’URL : <http://wiki.openstreetmap.org/wiki/JMapViewer>

## Utilisation d’*OpenStreetMap*

Méthodes utiles de la classe ***JMapViewer***

*Provides a simple panel that displays pre-rendered map tiles loaded from the OpenStreetMap project.*

|  |  |
| --- | --- |
| Void **setDisplayPositionByLatLon**(double lat, double lon, int zoom) | Changes the map pane so that it is centered on the specified coordinate at the given zoom level. |
| Point **getCenter**() | Return the center. |
| void **setCenter**(Point center) | The center to set. |
| Coordinate **getPosition**() | Calculates the latitude/longitude coordinate of the center of the currently displayed map area.  Return latitude / longitude. |
| Coordinate **getPosition**(Point mapPoint)  Coordinate **getPosition**(int mapPointX, int mapPointY) | Converts the relative pixel coordinate (regarding the top left corner of the displayed map) into a latitude / longitude coordinate.  Return latitude / longitude. |
| Point **getMapPosition**(double lat, double lon, boolean checkOutside)  Point **getMapPosition**(double lat, double lon) = getMapPosition(lat, lon, true)  Point **getMapPosition**(Coordinate coord) = getMapPosition(coord.getLat(), coord.getLon())  Point **getMapPosition**(ICoordinate coord, boolean checkOutside) | Calculates the position on the map of a given coordinate.  Return point on the map or null if the point is not visible and checkOutside set to true. |
| Integer **getLatOffset**(double lat, double offset, boolean checkOutside) | Calculates the position on the map of a given coordinate.  Return Integer the radius in pixels. |
| double **getMeterPerPixel**() | Gets the meter per pixel. |
| void **moveMap**(int x, int y) | Moves the visible map pane. |
| int **getZoom**() | Return the current zoom level. |
| void **zoomIn**()  void **zoomIn**(Point mapPoint)  void **zoomOut**()  void **zoomOut**(Point mapPoint)  void **setZoom**(int zoom, Point mapPoint)  void **setZoom**(int zoom) = setZoom(zoom, center) | Increases/decreases the current zoom level by one.  Param mapPoint 🡪 point to choose as center for new zoom level.  Set the zoom level and center point for display. |
| boolean **isTileGridVisible**()  void **setTileGridVisible**(boolean tileGridVisible) |  |
| boolean **getMapMarkersVisible**()  void **setMapMarkerVisible**(boolean mapMarkersVisible) { | Enables or disables painting of the {@link MapMarker} |
| void **setMapMarkerList**(List<MapMarker> mapMarkerList)  List<MapMarker> **getMapMarkerList**()  void **addMapMarker**(MapMarker marker)  void **removeMapMarker**(MapMarker marker)  void **removeAllMapMarkers**() { |  |
| void **setMapRectangleList**(List<MapRectangle> mapRectangleList)  List<MapRectangle> **getMapRectangleList**()  void **addMapRectangle**(MapRectangle rectangle)  void **removeMapRectangle**(MapRectangle rectangle)  void **removeAllMapRectangles**()  boolean **isMapRectanglesVisible**()  void **setMapRectanglesVisible**(boolean mapRectanglesVisible) { |  |
| void **setMapPolygonList**(List<MapPolygon> mapPolygonList)  List<MapPolygon> **getMapPolygonList**()  void **addMapPolygon**(MapPolygon polygon)  void **removeMapPolygon**(MapPolygon polygon)  void **removeAllMapPolygons**()  boolean **isMapPolygonsVisible**()  void **setMapPolygonsVisible**(boolean mapPolygonsVisible) |  |
| void **setZoomContolsVisible**(boolean visible)  boolean **getZoomContolsVisible**() |  |
| void **setTileSource**(TileSource tileSource) |  |
| void **addJMVListener**(JMapViewerEventListener listener)  void **removeJMVListener**(JMapViewerEventListener listener) { | Param listener listener to set/remove. |

Méthodes utiles de la classe ***OsmMercator***

*This class implements the Mercator Projection as it is used by Openstreetmap (and google). It provides methods to translate coordinates from 'map space' into latitude and longitude (on the WGS84 ellipsoid) and vice versa. Map space is measured in pixels. The origin of the map space is the top left corner. The map space origin (0,0) has latitude ~85 and longitude -180.*

|  |  |
| --- | --- |
| static int **getMaxPixels**(int aZoomlevel) | Returns the absolut number of pixels in y or x, defined as: 2^Zoomlevel \* TILE\_WIDTH where TILE\_WIDTH is the width of a tile in pixels. |
| static double **getDistance**(int x1, int y1, int x2, int y2, int zoomLevel) | Transform pixelspace to coordinates and get the distance. |
| static double **getDistance**(double la1, double lo1, double la2, double lo2) | Gets the distance using Spherical law of cosines. |
| static int **LonToX**(double aLongitude, int aZoomlevel)  static int **LatToY**(double aLat, int aZoomlevel) | Transform longitude/latitude to pixelspace. |
| static double **XToLon**(int aX, int aZoomlevel)  static double **YToLat**(int aY, int aZoomlevel) | Transforms pixel coordinate X/Y to longitude/latitude. |
|  |  |