Link: <http://code.google.com/android/add-ons/google-apis/reference/index.html>

MapPro book: trang 267

MapController

MapController được sử dụng để điều khiển bản đồ, cho phép thiết lập vị trí giữa của bản đồ và các mức phóng to/thu nhỏ. Có một số phương thức được địn nghĩa cho việc phóng to, thu nhỏ thông quan lớp này. Android định nghĩa 21 mức phóng to/thu nhỏ cho bản đồ. Ở mức 1, xích đạo của trái đất dài 256 pixels. Mỗi lần phóng to thì con số này tang gấp 2 lần.

Có thể tham chiếu đến controller của MapView bằng cách sử dụng phương thức getController.

MapController mapController = myMapView.getController();

Các vị trí các điểm bản đồ của các lớp mapping trên Android biểu diễn bằng các đối tượng GeoPoint, chứa kinh độ và vĩ độ được đo ở đơn vị microdegree. Để chuyển độ thành microdegree ta nhân số này với 1E6 (1.000.000).

Trước khi có thể sử dụng tọa độ lưu trong đối tượng Location, cần chuyển tọa độ này thành microdegree và lưu lại thành đối tượng GeoPoint.

Double lat = 37.422006\*1E6;

Double lng = -122.084095\*1E6;

GeoPoint point = new GeoPoint(lat.intValue(), lng.intValue());

Để định vị vị trí giữa và điều chỉnh kích thướt bản đồ chúng ta sử dụng phương thức setCenter và setZoom có trong đối tượng MapController của MapView.

mapController.setCenter(point);

mapController.setZoom(1);

Phương thức setCenter sẽ nhảy đến một vị trí mới. Để chuyển động này một cách mượt mà chung ta sử dụng thêm phương thức animateTo().

mapController.animateTo(point);

Dưới đây là đoạn code mô tả cách sử dụng lớp này MapController

1. Đầu tiên chúng ta thiết kế giao diện cho ứng dụng như sau:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"

android:orientation="vertical"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent">

<TextView

android:id="@+id/myLocationText"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="@string/hello"

/>

<com.google.android.maps.MapView

android:id="@+id/myMapView"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:enabled="true"

android:clickable="true"

android:apiKey="myMapKey"

/>

</LinearLayout>

1. Cấu hình MapView và lưu một tham chiếu đến MapController của nó.

**MapController mapController;**

@Override

public void onCreate(Bundle savedInstance

State) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

// Get a reference to the MapView

MapView myMapView = (MapView)findViewById(R.id.myMapView);

**// Get the Map View’s controller**

**mapController = myMapView.getController();**

// Configure the map display options

myMapView.setSatellite(true);

myMapView.setStreetView(true);

myMapView.displayZoomControls(false);

// Zoom in

**mapController.setZoom(17);**

}

adsa

|  |  |
| --- | --- |
| Phương thức | Mô tả |
| public void **stopPanning**() | Đặt lại tình trạng của pan để bản đồ đứng yên một chỗ. Điều này cần thiết nếu chúng ta nhận một sự kiện keydown nhưng không nhận sự kiện key-up tương ứng |
| public boolean **onKey**(android.view.View v,  int keyCode,  android.view.KeyEvent event) | Xử lý các sự kiện nhấn phím và chuyển các sự kiện này thành “pan” phù hợp với map. Được định nghĩa trong lớ[ View.OnKeyListener. |
| public void **animateTo**([GeoPoint](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/GeoPoint.html) point) | Chuyển bản đồ về phía điểm được chọn |
| public void **animateTo**([GeoPoint](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/GeoPoint.html) point,android.os.Message message) | Chuyển bản đồ về điểm được chọn. Nếu và khi sự chuyển động đến điểm cuối cùng, một thông điệp sẽ được gửi đi (nếu không phải null). |
| public void **animateTo**([GeoPoint](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/GeoPoint.html) point,  java.lang.Runnable runnable) |  |
| public void **scrollBy**(int x,  int y) | Cuộn bản đồ theo lượng pixel được cho sẵn.  Parameters  x- cuộng theo chiều ngang  y – cuộn theo chiều dọc |
| public void **setCenter**([GeoPoint](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/GeoPoint.html) point) | Đặt góc nhìn bản đồ đến điểm được cho. Sẽ không có bất kỳ chuyển động nào được thực hiện |
| public void **stopAnimation**(boolean jumpToFinish) | Dừng bất kỳ một chuyển động nào đang được thực hiện. |
| public int **setZoom**(int zoomLevel) | Sets the zoomlevel of the map. The value will be clamped to be between 1 and 21 inclusive, though not all areas have tiles at higher zoom levels. This just sets the level of the zoom directly; for a step-by-step zoom with fancy interstitial animations, use [zoomIn()](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapController.html#zoomIn()) or [zoomOut()](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapController.html#zoomOut()).  Thiết lập mức độ phóng to thu nhỏ của bản đồ. Tất cả giá trị sẽ được nằm giữa 1 và 21, mặc dầu không phải tất cả khu vực đều có các khối vật ở mức cao hơn các mức zoom. Hàm này chỉ thiết lập mức zoom một cách trực tiếp.  Sử dụng zoomIn() hay ZoomOut() để thay đổi mức zoom của bản đồ. |
| public void **zoomToSpan**(int latSpanE6,  int lonSpanE6) | Điều chỉnh độ phóng to thu nhỏ của bản đồ vì vậy khoảng cách được trao của kinh độ và vĩ độ sẽ được hiển thị. Bời vì zoom chỉ có thể lấy về các mức riên lẻ, và bởi vì tỉ lệ kích cỡ màn hình của bản đồ có thể không khớp với tỉ lệ được cho, nên chất lượng khớp sẽ có thể biến đổi. |
| public boolean **zoomIn**() | Phóng to một đơn vị. |
| public boolean **zoomOut**() | Thu nhó đơn vị |
| public boolean **zoomInFixing**(int xPixel,  int yPixel) | Zoom in by one zoom level. This begins an animated zoom step. If you have added the ZoomControls View (from [MapView.getZoomControls()](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapView.html#getZoomControls())), then this will also call through to [MapView.displayZoomControls(boolean)](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapView.html#displayZoomControls(boolean)) causing the ZoomControls to appear briefly, but they will not take focus.  This zoom will also pan the map to keep a single point fixed on the screen. Specify the point you want fixed by giving its pixel coordinates.  **Parameters:**  xPixel - offset, in pixels from the left of the map, where the fixed point of our zoom will be.  yPixel - offset, in pixels from the top of the map, where the fixed point of our zoom will be.  **Returns:**  true if we managed to zoom in, false if we hit the limit. |
| public boolean **zoomOutFixing**(int xPixel,  int yPixel) | Zoom out by one zoom level. This begins an animated zoom step. If you have added the ZoomControls View (from [MapView.getZoomControls()](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapView.html#getZoomControls())), then this will also call through to [MapView.displayZoomControls(boolean)](http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapView.html#displayZoomControls(boolean)) causing the ZoomControls to appear briefly, but they will not take focus.  This zoom will also pan the map to keep a single point fixed on the screen. Specify the point you want fixed by giving its pixel coordinates.  **Parameters:**  xPixel - offset, in pixels from the left of the map, where the fixed point of our zoom will be.  yPixel - offset, in pixels from thetop of the map, where the fixed point of our zoom will be.  **Returns:**  true if we managed to zoom out, false if we hit the limit. |