Interacting with the map

The Maps API allows you to customize the way in which a user can interact with your map. You can decide which of the built in UI components will appear on the map, what gestures are allowed and how to respond to click events from your users.

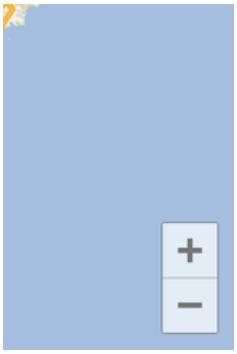
UI controls

The Maps API comes with some built-in UI controls that are similar to those found in the Google Maps application on your Android phone. You can toggle the visibility of these controls using the `UiSettings` class which can be obtained from a `GoogleMap` with the `GoogleMap.getUiSettings` method. Changes made on this class are immediately reflected on the map. To see an example of these features, look at the UI Settings demo activity in the sample application.

You can also configure most of these options when the map is created either via XML Attributes or using the `GoogleMapOptions` class. See Configuring initial state for more details.

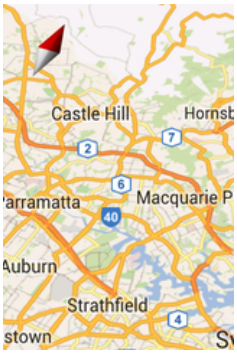
Zoom controls

The Maps API provides built-in zoom controls that appear in the bottom right hand corner of the map. These are enabled by default, but can be disabled by calling `UiSettings.setZoomControlsEnabled(boolean)`.



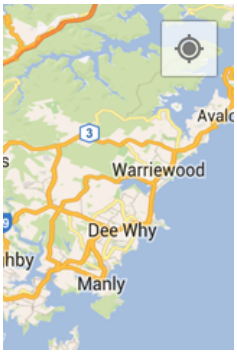
Compass

The Maps API provides a compass graphic which appears in the top left corner of the map under certain circumstances. The compass will only ever appear when the camera is oriented such that it has a non-zero bearing or non-zero tilt. When the user clicks on the compass, the camera animates back to a position with bearing and tilt of zero (the default orientation) and the compass fades away shortly afterwards. You can disable the compass appearing altogether by calling `UiSettings.setCompassEnabled(boolean)`. However, you cannot force the compass to always be shown.



My Location button

The My Location button appears in the top right corner of the screen *only* when the My Location layer is enabled. When a user clicks the button, the camera animates to focus on the user's current location if the user's location is currently known. You can disable the button from appearing altogether by calling `UiSettings.setMyLocationButtonEnabled(boolean)`.



Map gestures

A map created with the Google Maps Android API supports the same gestures as the Google Maps application. However, there might be situations where you want to disable certain gestures in order to preserve the state of the map. Zoom, pan, tilt and bearing can also be set programmatically - see Changing the View for more details. Note that disabling gestures does not affect whether you can change the camera position programmatically.

Like the UI controls, you can enable/disable gestures with the `UiSettings` class which can be obtained from a `GoogleMap` by calling `GoogleMap.getUiSettings`. Changes made on this class are immediately reflected on the map. To see an example of these features, look at the UI Settings demo activity in the sample application (see here for how to install it).

You can also configure these options when the map is created either via XML Attributes or using the `GoogleMapOptions` class. See Configuring the map for more details.

Zoom gestures

The map responds to a variety of gestures that can change the zoom level of the camera:

- Double tap to increase the zoom level by 1 (zoom in).
- Two finger tap to decrease the zoom level by 1 (zoom out).
- Two finger pinch/stretch
- One finger zooming by double tapping but not releasing on the second tap and then sliding the finger up or down the screen to zoom in and out respectively.

You can disable zoom gestures by calling `UiSettings.setZoomGesturesEnabled(boolean)`. This will not affect whether a user can use the zoom controls to zoom in and out.

Scroll (pan) gestures

A user can scroll (pan) around the map by dragging the map with their finger. You can disable scrolling by calling `UiSettings.setScrollGesturesEnabled(boolean)`.

Tilt gestures

A user can tilt the map by placing two fingers on the map and moving them down or up together to increase or decrease the tilt angle respectively. You can disable tilt gestures by calling `UiSettings.setTiltGesturesEnabled(boolean)`.

Rotate gestures

A user can rotate the map by placing two fingers on the map and applying a rotate motion. You can disable rotation by calling `UiSettings.setRotateGesturesEnabled(boolean)`.

Map events

The Maps API also allows you to listen to events on the map. To see a simple example of these features, look at the Events demo activity in the sample application (see here for how to install it).

Map click/long click events

If you want to respond to a user tapping on a point on the map, you can use an `OnMapClickListener` which you can set on the map by calling `GoogleMap.setOnMapClickListener(OnMapClickListener)`. When a user clicks (taps) somewhere on the map, you will receive an `onMapClick(LatLng)` event that indicates the location on the map that the user clicked. Note that if you need the corresponding location on the screen (in pixels), you can obtain a `Projection` from the map which allows you to convert between latitude/longitude coordinates and screen pixel coordinates.

You can also listen for long click events with an `OnMapLongClickListener` which you can set on the map by calling `GoogleMap.setOnMapLongClickListener(OnMapLongClickListener)`. This listener behaves similarly to the click listener and will be notified on long click events with an `onMapLongClick(LatLng)` callback.

Camera change events

If you want to keep track of the camera position, you can use an `OnCameraChangeListener` which is set on the map by calling `GoogleMap.setOnCameraChangeListener(OnCameraChangeListener)`. The listener will be notified when the camera changes with an `onCameraChange(CameraPosition)` callback. You can then obtain the target (latitude/longitude), zoom, bearing and tilt of the camera - see the developer guide for camera position for more details on these properties. This callback is guaranteed to be called at the end of every animation but may not be called for intermediate frames.

Last updated June 6, 2013.