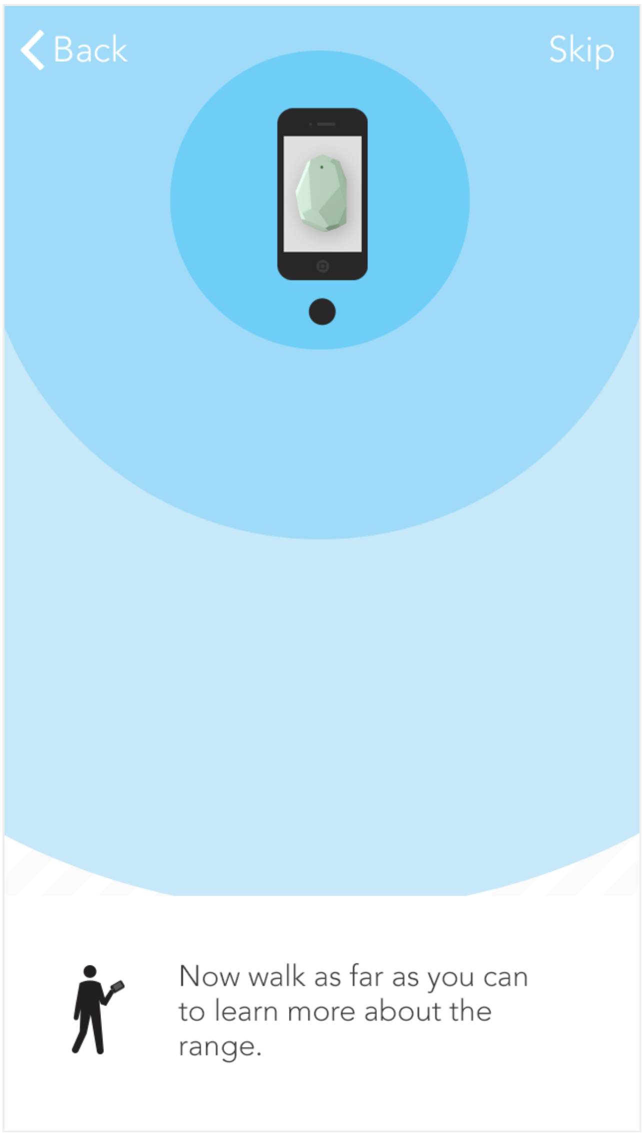
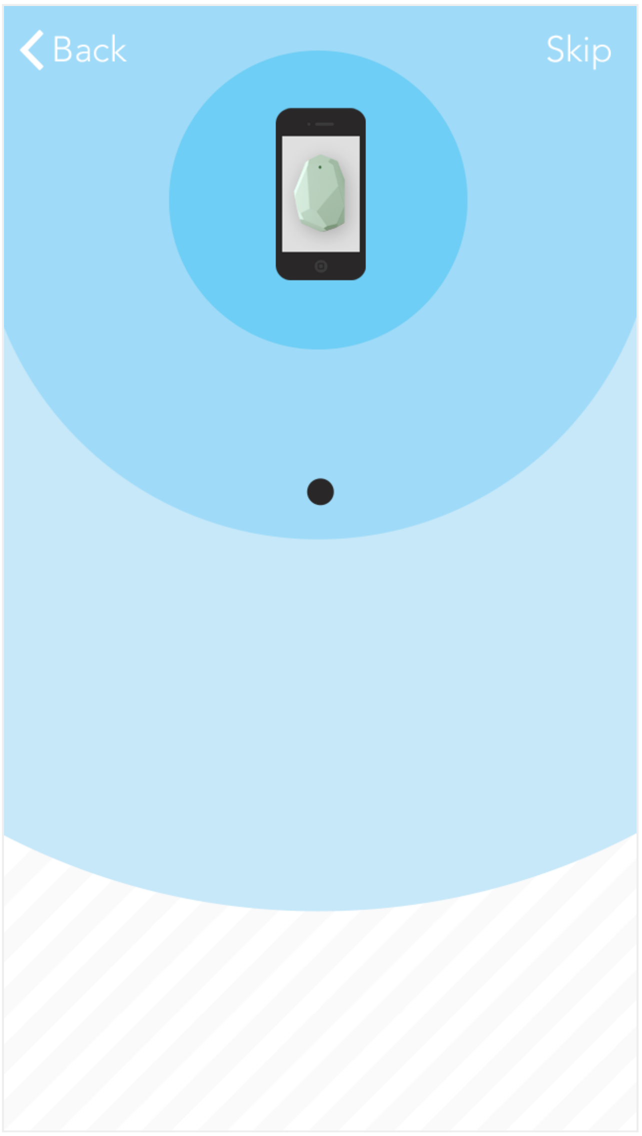
API Documentation

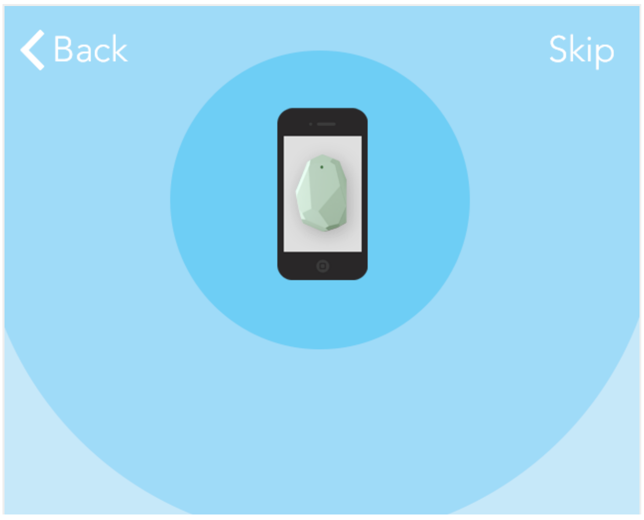
Introduction

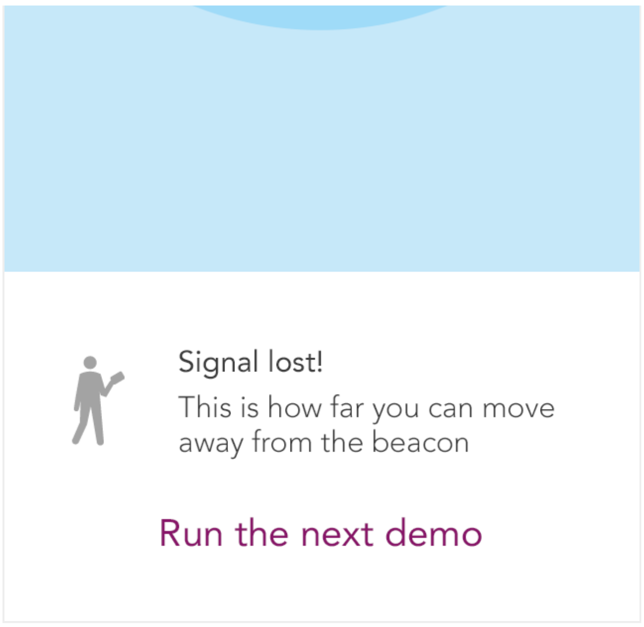
The distance app is one of the simplest Estimote demo apps, yet also one of the most useful to understand as its functionality lies at the very core of all the other things you can do with Estimote beacons.



Now walk as far as you can to learn more about the range.







How it works

This app measures and visualizes the distance between user's position and the beacon. This information can be then used e.g. to tell if the person is coming closer to the beacon, moving away or if they have stopped, and to create events relevant to the context the person is in.

The way it works is really simple - a beacon broadcasts its ID and the strength of the signal, or RSSI (Received Signal Strength Indication), is being continuously monitored by the cell phone. Based on the signal strength and calibration data it is possible to infer how far away the cell phone is from the beacon.

Eliminating interferences

Because of possible interferences, it is important to sample the signal strength in a way that will filter out sudden spikes and dips, which would influence the read-out. This is done by using a rolling average, which takes into account multiple read-outs, not just one. This stabilizes the distance value, at the expense of slight degradation of responsivity.

Displaying results in the user interface

In the user interface, the area around the beacon is divided into three “zones”: “immediate”, “near” and “far”. These zones are not used in this demo, but are used in other demo applications. However, in this demo the striped “out of range” zone is used, as the user is asked to go away as far as possible until they completely lose signal. For practical reasons though, the “signal lost” message is shown quite a bit earlier, lest the user could not walk far enough to really lose the signal.

How to do it

Recreating the distance demo is quite simple. Let's walk through some key elements of this demo app.

First, make sure you added the CoreBluetooth.framework and CoreLocation.framework to your project as well as libEstimoteSDK.a , which is included in the Examples project.

```
- (void)viewDidLoad
{
    [super viewDidLoad];

    /////////////////////////////////////////////////////////////
    // set up Estimote beacon manager

    // create a beacon manager instance
    self.beaconManager = [[ESTBeaconManager alloc] init];
    self.beaconManager.delegate = self;
    self.beaconManager.avoidUnknownStateBeacons = YES;

    // create a sample region object (you can also pass major or major+minor 
    ESTBeaconRegion* region = [[ESTBeaconRegion alloc] initWithProximityUUID:

    // start looking for Estimote beacons in the region
    // when beacons are found in range, beaconManager:didRangeBeacons:inRegio
    [self.beaconManager startRangingBeaconsInRegion:region];

    /////////////////////////////////////////////////////////////
    // setup view

    [self setupView];
}
```

First step is to create an Estimote Beacon Manager object using the ESTBeaconManager class and set the delegate object. In this example we are using avoidUnknownStateBeacons flag. Core Location framework has some inertia in object removal from the list of discovered beacons and setting this property to YES allows us to force the removal of unknown state objects from the list. We also invoke startRangingBeaconsInRegion: method to start looking for Estimote beacons that are described with the basic ESTBeaconRegion object.

```
-(void)setupView
{
    ////////////////////////////////////////////////////////////////
    // setup background image

    CGRect          screenRect          = [[UIScreen mainScreen] bounds];
    CGFloat         screenHeight        = screenRect.size.height;
    UIImageView*    backgroundImage;

    if (screenHeight > 480)
        backgroundImage = [[UIImageView alloc] initWithImage:[UIImage imageNar
    else
        backgroundImage = [[UIImageView alloc] initWithImage:[UIImage imageNar

    [self.view addSubview:backgroundImage];

    ////////////////////////////////////////////////////////////////
    // setup dot image

    self.positionDot = [[UIImageView alloc] initWithImage:[UIImage imageNamed
    [self.positionDot setCenter:self.view.center];
    [self.positionDot setAlpha:1.];

    [self.view addSubview:self.positionDot];

    self.dotMinPos = 150;
    self.dotRange = self.view.bounds.size.height  - 220;
}
```

In the setupView method we define the background image as well as the dot image that indicates the distance between the beacon and the phone.

```
-(void)beaconManager:(ESTBeaconManager *)manager
     didRangeBeacons:(NSArray *)beacons
            inRegion:(ESTBeaconRegion *)region
{
    ESTBeacon* closestBeacon;

    if([beacons count] > 0)
    {
        // beacon array is sorted based on distance
        // closest beacon is the first one
        closestBeacon = [beacons objectAtIndex:0];

        // calculate and set new y position
        float newYPos = self.dotMinPos + ((float)closestBeacon.rssi / -100.)
        self.positionDot.center = CGPointMake(self.view.bounds.size.width / 2
    }
}
```

When our UI and ESTBeaconManager are in place, the only remaining thing to do is to handle discovered beacons. We use beaconManager:didRangeBeacons:inRegion: delegate method to do that. The list of discovered beacons is sorted based on the distance (the nearest beacon is first on the list). We get the nearest ESTBeacon object using index 0. We are using it to get the current RSSI value, and based on it we update dot position. RSSI value ranges from -100 to 0, and the closer the value is to 0, the closer the beacon is to the device.

You can download the complete Examples project (which includes the Distance demo) from our GitHub.

Download the source code

Proximity demo app

Display a message upon crossing zones

Notification demo app

Push notification sent upon leaving a zone

Home
How does it work?
Enterprise solutions
Contact

Press & Media
Support
Privacy & Policy
Jobs at Estimote

Twitter
Facebook
Blog
Community

Estimote, Inc.
35 E 19th St. 3rd floor
New York, NY 10003
United States

Estimote Polska Sp z o.o.
ul. Krakusa 11
30-535 Kraków, Poland
European Union