

# A bit about the Pencil

Fabrice.Kordon@lip6.fr



# As an introduction



**Pencil 1 introduced with the first iPadPro**

September 2015



**Handled together with «classic touches»**

UITouch class enriched

# New properties & methods

## Property type

 direct

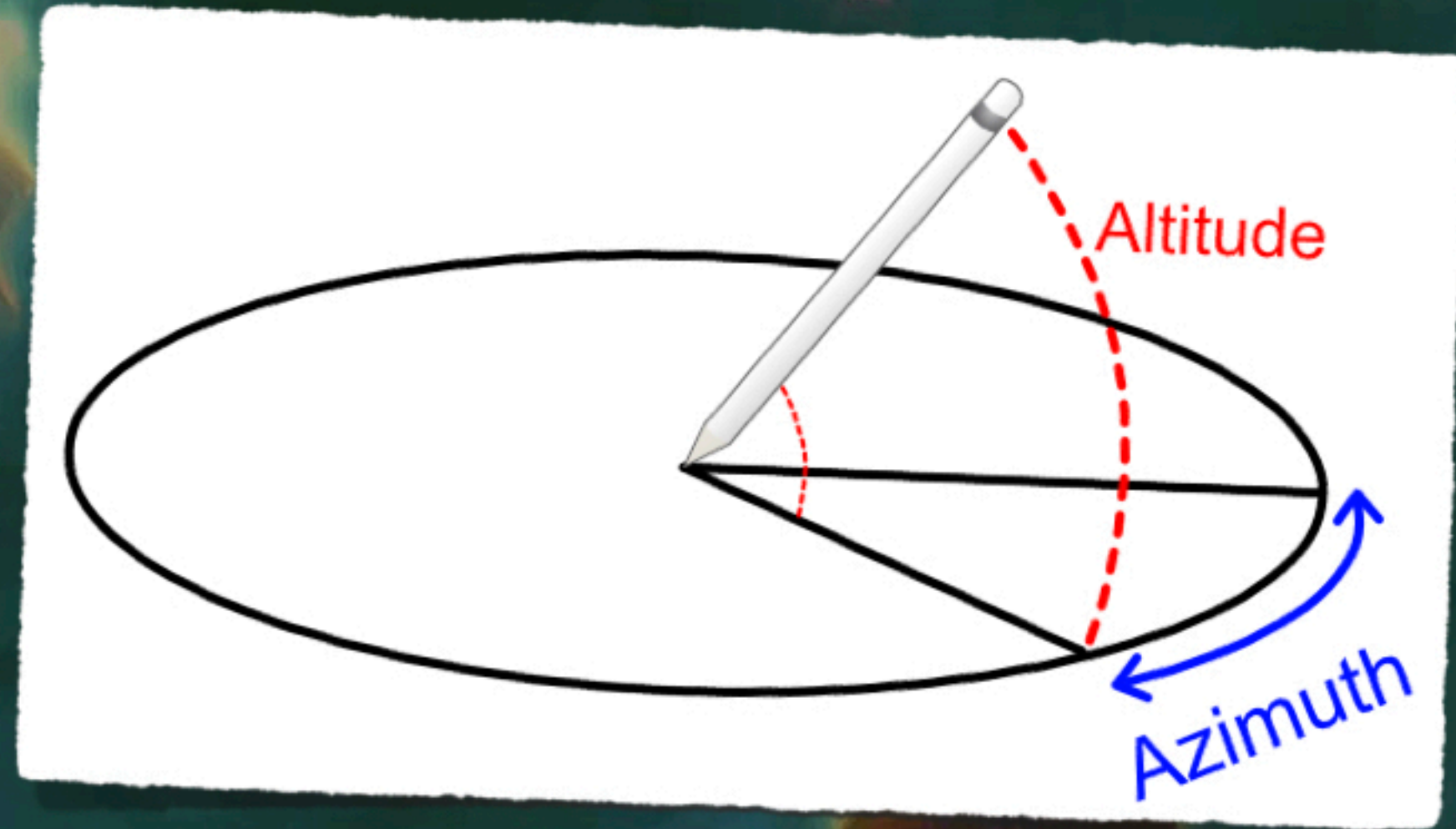
▶ «Classical» touch

 indirect

▶ Not coming from the screen  
(e.g. Apple TV's trackpad)

 pencil

▶ From the pencil



## Property altitudeAngle

 CGFloat between 0 and  $\pi/2$

## Method azimuthAngle(in:)

 azimuthAngle(in view: UIView?) -> CGFloat

 Returns a value in radian

# New properties & methods

## Property type

direct

▶ «Classical» touch

indirect

▶ Not coming from the screen  
(e.g. Apple Pencil)

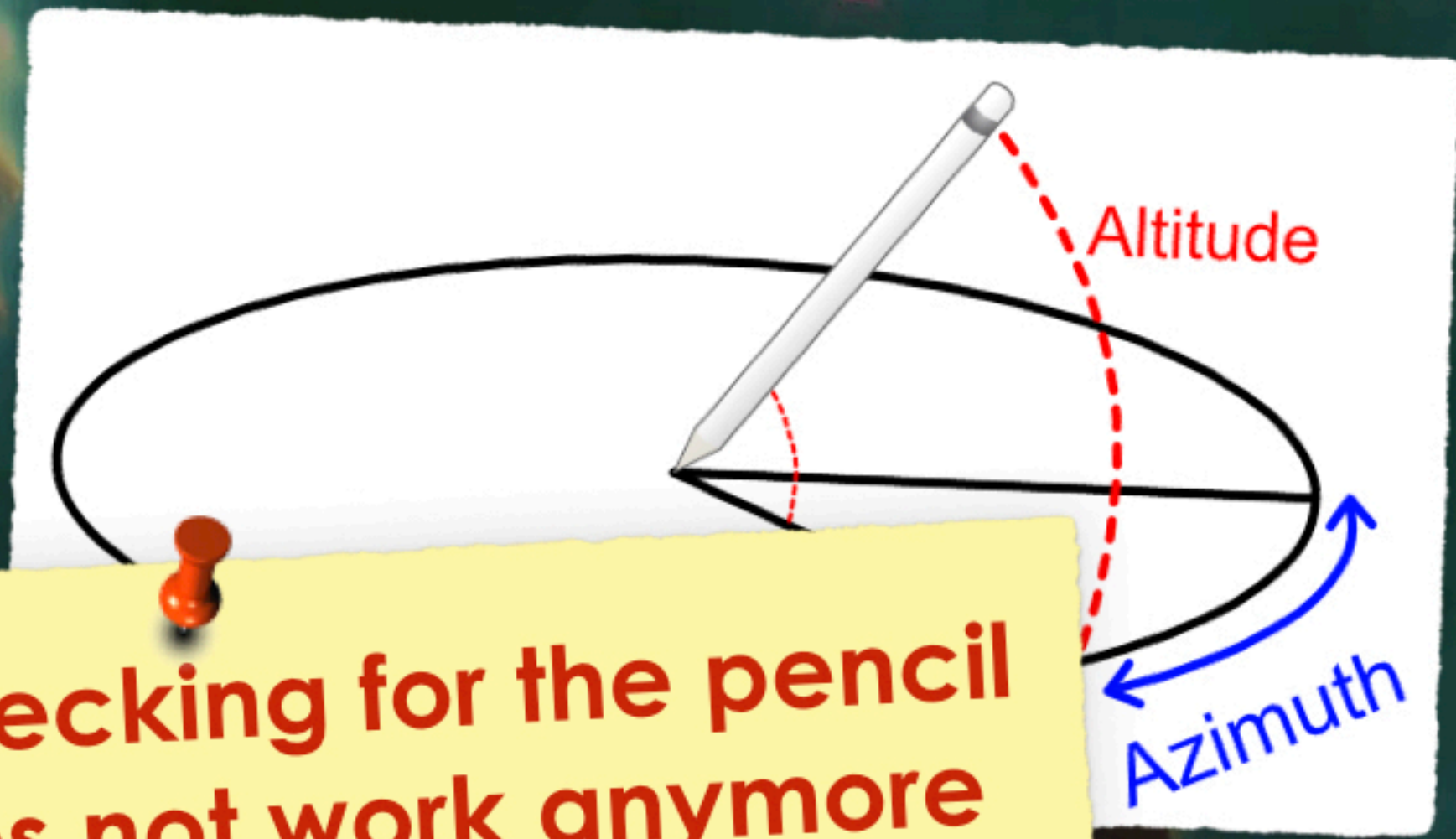
pencil

▶ From the pencil



**In iOS 12, checking for the pencil  
like this does not work anymore**

`traitCollection.forceTouchCapability`



## Property of

CGFloat between 0 and  $\pi/2$

## Method azimuthAngle(in:)

azimuthAngle(in view: UIView?) -> CGFloat

Returns a value in radian

# Testing pencil's presence?

4

## 📱 Quite difficult

- 👤 Requires to play with CoreBluetooth

## 📱 Theoretically

- 👤 You create a `CBCentralManager`
  - ▶ Requires the support of the `CBCentralManagerDelegate` protocol
- 👤 You fill the mandatory method

```
func centralManagerDidUpdateState(_ central: CBCentralManager)
```

- ▶ Check if state is `poweredOn`
- ▶ Create a `CBUUID` dictionary
- ▶ Retrieve the list of connected peripheral

```
let pencilID = [CBUUID(string: "180A")]
```

```
func retrieveConnectedPeripherals(withServices serviceUUIDs: [CBUUID])  
    -> [CBPeripheral]
```

- ▶ Browse this array to check if the pencil is there

# Testing pencil's presence?



## Enable bluetooth

in the info.plist

```
<key>NSBluetoothPeripheralUsageDescription</key>  
<string>To look for the pencil</string>
```

## Quite difficult

Requires to p

## Theoretically

You create a CBCentralManager

Requires the of the CBCentralManagerDelegate protocol

You fill the v method

```
func centr
```

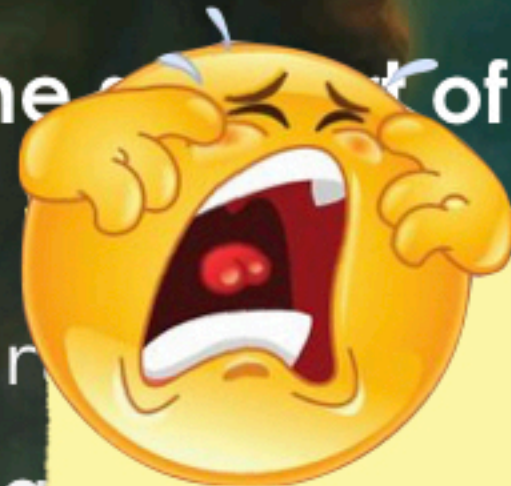
Check if sta

Create a C

```
let pencil
```

Retrieve the

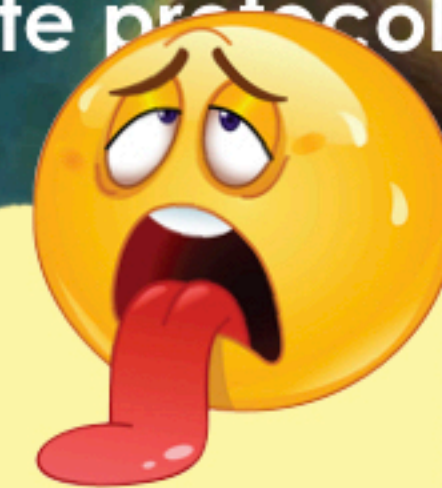
```
func retri
```



## Theoretically...

I could not get it working  
Obscure message

«[CoreBluetooth] XPC connection invalid»  
Unresolved yet in forums






Browse this array to check if the pencil is there

```
[CBUID])
```

# As a conclusion...

 No so difficult...

 Key issues

-  Use appropriately the pencil's information
-  Check for the presence of this `<beeeeeep>` pencil

