

My K-Means Clustering



The screenshot displays a Swift application running on an iPhone XR simulator. The app's main view shows a scatter plot of approximately 15 gray points. At the bottom of the simulator, there are three buttons: "3 Clusters", "2 Clusters", and "Clear".

The background shows the Xcode IDE with the following Swift code in `ViewController.swift`:

```

1 //
2 // ViewController.swift
3 // KMeansSwift
4 //
5 // Created by sdq on
  15/12/22.
6 // Copyright © 2015年
  sdq. All rights
  reserved.
7 //
8
9 import UIKit
10 import SafariServices
11
12 internal class
  ViewController :
  UIViewController {
13
14     @IBOutlet weak
      internal var
      spaceView:
      UIView!
15

```

System monitoring tools on the left show CPU at 0%, Memory at 59.9 MB, Disk at 8 KB/s, and Network at Zero KB/s. The desktop on the right contains folders for "swift_2019", "temp", "deepvis-master", and "iMovie".

K-means unsupervised machine learning algorithm (Swift)

 8 commits

 1 branch

 0 releases

 1 contributor

Branch: **master** ▾

[New pull request](#)

[Find File](#)








[Clone or download](#) ▾

This branch is 12 commits behind sdq:master.

[Pull request](#) [Compare](#)

 **sdq** and **sdq** swift 3

Latest commit efbe738 on 10 Nov 2016

 KMeans	swift 3	3 years ago
 KMeansSwift.xcodeproj	swift 3	3 years ago
 KMeansSwift	swift 3	3 years ago
 KMeansSwiftMac	swift 3	3 years ago
 .DS_Store	swift 3	3 years ago
 README.md	Update README.md	4 years ago
 demo.gif	finish algorithm and demo	4 years ago

Step. Add KMeans.swift

yusufkildan / KMeans-Swift
forked from sdq/deepvis

Watch 1

Star 0

Fork 8

Code

Pull requests 0

Projects 0

Security

Insights

Branch: master

KMeans-Swift / KMeans /

Create new file

Find file

History

This branch is 12 commits behind sdq:master.

Pull request Compare

sdq and sdq swift 3

Latest commit efbe738 on 10 Nov 2016

KMeans.swift

swift 3

3 years ago

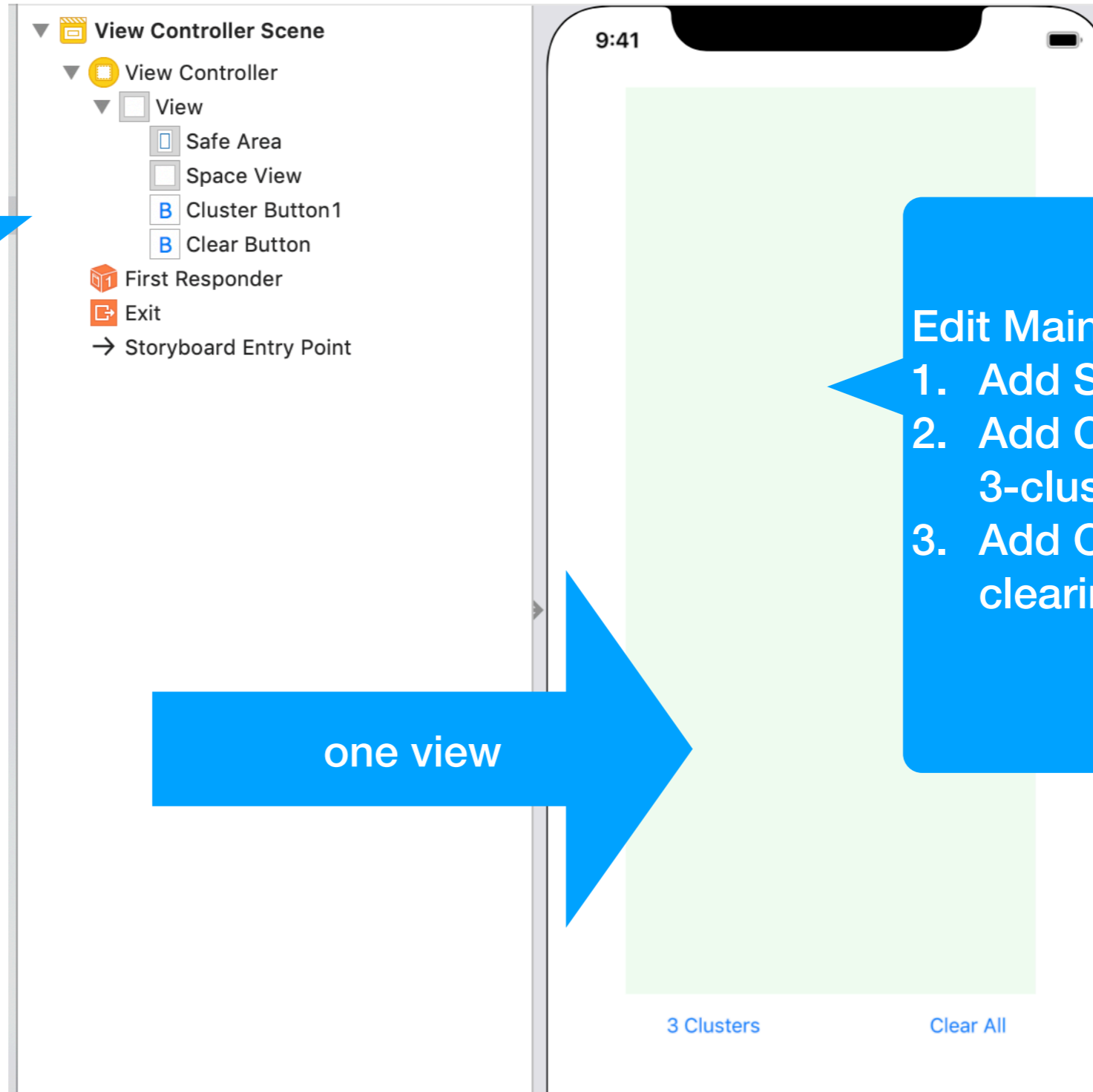
Add
KMeans.swift
To your
project

1. Goto <https://github.com/yusufkildan/KMeans-Swift>
2. download KMeans.swift
3. Add Means.swift to my_Kmeans_Clustering project

Step. design Main.storyboard



Two
buttons



one view

Edit Main.storyboard

1. Add Space view
2. Add Cluster button 1 for 3-cluster clustering
3. Add Clear Button for clearing all points

Step. Connections

1. Connect Space view to weak var spaceView
2. Connect Cluster button 1 to weak var clusterButton1
3. Connect Clear Button to weak var clearButton

1. Connect primary-action-trigger event of Cluster button1 to func strat3Clustering
2. Connect primary-action-trigger event of Clear Button to func clearAll

Declare constant KMeans

```
6 // Copyright © 2019 Apple. All rights reserved.
7 //
8
9 import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var clearButton:
14         UIButton!
15     @IBOutlet weak var clusterButton1:
16         UIButton!
17
18     @IBOutlet weak var spaceView: UIView!
19     let KMeans = KMeansSwift.sharedInstance
20
21     @IBAction func clearAll(_ sender: Any) {
22         KMeans.reset()
23         for point:UIView in
24             self.spaceView.subviews {
25             point.removeFromSuperview()
26         }
27     }
28
29     @IBAction func strat3Clustering(_
30         sender: Any) {
31
32     }
33 }
```

3 Clusters

Clear All

Edit func viewDidLoad()

```
override func viewDidLoad() {  
    super.viewDidLoad()  
    // Do any additional setup after loading the view.  
    let tapGesture = UITapGestureRecognizer(target: self, action:  
#selector(ViewController.tapSpaceView(_:)))  
    spaceView.addGestureRecognizer(tapGesture)  
    spaceView.isUserInteractionEnabled = true  
}
```

1. Declare a constant tapGesture
2. Notice the use of tapSpaceView
3. Call spaceView.addGestureRecognizer(tapGesture)
4. Enable spaceView.isUserInteractionEnabled

1. Get location
2. Create pointFrame
3. Create grayPoint using pointFrame
4. Set backgroundColor of grayPoint
5. Add subview grayPoint to spaceView
6. Add location to KMeans

```
@objc func tapSpaceView(_ recognizer:UITapGestureRecognizer) {  
    let location = recognizer.location(in: spaceView)  
    let pointFrame = CGRect(x: location.x - 5, y: location.y - 5, width: 10, height: 10)  
    let grayPoint = UIView(frame: pointFrame)  
    grayPoint.backgroundColor = UIColor(red: 1, green: 0, blue: 0, alpha: 1)  
    spaceView.addSubview(grayPoint)  
    KMeans.addVector([Double(location.x), Double(location.y)])  
}
```

How to add subview on spaceView?


```
@IBAction func clearAll(_ sender: Any) {  
    KMeans.reset()  
    for point:UIView in self.spaceView.subviews {  
        point.removeFromSuperview()  
    }  
}
```

1. Reset KMeans
2. Remove all subviews of spaceView

```

@IBAction func strat3Clustering(_ sender: Any) {
    if KMeans.vectors.count < 3 {
        return
    }
    // setButtonEnable(false)
    KMeans.clusteringNumber = 3
    KMeans.clustering(500) { [unowned self] (success, centroids, clusters) -> () in
        if success {
            for point:UIView in self.spaceView.subviews {
                point.removeFromSuperview()
            }
            for index in 0...2 {
                for vector in clusters[index] {
                    let pointFrame = CGRect(x: CGFloat(vector[0]) - 5, y: CGFloat(vector[1]) - 5, width: 10.0, height: 10.0)
                    let point = UIView(frame: pointFrame)
                    if index == 0 {
                        point.backgroundColor = UIColor(red: 1, green: 0, blue: 0, alpha: 1)
                    } else if index == 1 {
                        point.backgroundColor = UIColor(red: 0, green: 0, blue: 1, alpha: 1)
                    } else {
                        point.backgroundColor = UIColor(red: 0, green: 1, blue: 0, alpha: 1)
                    }
                    self.spaceView.addSubview(point)
                }
            }
        }
        //self.setButtonEnable(true)
    }
}

```