**Guide**

1. Add InfinityScrollView.swift and kInfinityScrollView.xib file in your project.
2. Implement the protocol InfinityScrollingViewDelegate and two of its method in the viewController (1- func updating(\_ view: InfinityScrollView, for imageType: CarouselImageType), 2- func didClickOnMiddleView(\_ ofIndex: Int))

**Initiating the View:**

1. In your view controller, make a normal container UiView infinityScrollingViewContainer = UiView() , where you will perform infinity scrolling and create its outlet(if done by storyboard).
2. In viewDidAppear (or any) initiate a view from the method using viewForInfinityScrolling= Bundle.main.loadNibNamed("kInfinityScrollView", owner: self, options: nil).first as! InfinityScrollView
3. Give the count of the array to be looped infinitely, viewForInfinityScrolling.countOfTheDataSource = array.count
4. Make its delegate to the self
5. You can autoRotate it by providing (canAutoRotate property) (optional)
6. Give the frame size of the viewForInfinityScrolling.frame.size = infinityScrollingViewContainer.frame.size
7. You can give timing for autoRotateTimeInterval = 5.0 (optional)
8. You can give separation between the views by the property separationBetweenTheViews = 50 (optional)
9. You can give the size of the one displaying view by fraction of the main container view, sizeOfTheView = (1600/1920) (optional)
10. Can give animation duration of the swipe by durationOfAnimation = 10 (optional)
11. Can give alpha of the views, alphaOfTheViews = 1.0 (optional)
12. viewForInfinityScrolling.loadViews() – For Loading the views with property
13. Add viewForInfinityScrolling view to the infinityScrollingViewContainer

**Updating the views with delegate methods:**

1. Updating the delegate call for updating the views on each swipe,

Functionality same as UiButton

func updating(\_ view: InfinityScrollView, for imageType: CarouselImageType) , Here update the view.clickableMiddleView. setBackgroundImage(carouselArray[forImageType.middle!], for: .normal)

view.clickableLeftView. setBackgroundImage(carouselArray[forImageType.middle!], for: .normal)

view.clickableRightView. setBackgroundImage(carouselArray[forImageType.middle!], for: .normal)

view.clickableExtraLeftView. setBackgroundImage(carouselArray[forImageType.middle!], for: .normal)

view.clickableExtraRightView. setBackgroundImage(carouselArray[forImageType.middle!], for: .normal)

1. Delegate method for clicking on the middle View, arrays index number will be provided in that method func didClickOnMiddleView(\_ ofIndex: Int))