

## BÀI 2. SCROLLVIEW VÀ PAGE CONTROL



### ✓ Mục tiêu:

- *Nắm được ý nghĩa, tầm quan trọng và các ứng dụng của ScrollView.*
- *Biết cách sử dụng ScrollView.*
- *Biết cách kết hợp ScrollView với Page Control để hiển thị chế độ phân trang.*

### Bài tập 2.1. Tạo một ứng dụng zoom ảnh

**Đề bài:** Tạo mới một ứng dụng. Kéo thả **ScrollView** vào màn hình giao diện và làm ứng dụng zoom ảnh như hình bên dưới:



Hình 1.1



Hình 1.2

Yêu cầu thực hiện:

- Thêm 1 hình vào project.
- Trong phương thức **viewDidLoad** của tập tin **ViewController.m**:
  - Khởi tạo một đối tượng imageView từ lớp UIImageView có chiều rộng và chiều cao bằng với kích thước của ScrollView.
  - Load hình lên đối tượng imageView mới tạo.
  - Thêm imageView vào ScrollView.



- Xét content size cho ScrollView bằng kích thước của imageView.
- Xét tỷ lệ thu nhỏ là 0.3.
- Xét tỷ lệ phóng to là 3.0.
- Gán delegate cho ScrollView.
- Áp dụng protocol **UIScrollViewDelegate** cho ViewController.
- Triển khai phương thức “**viewForZoomingInScrollView**”:
  - Trả về view muốn zoom trong ScrollView.

#### Mục tiêu:

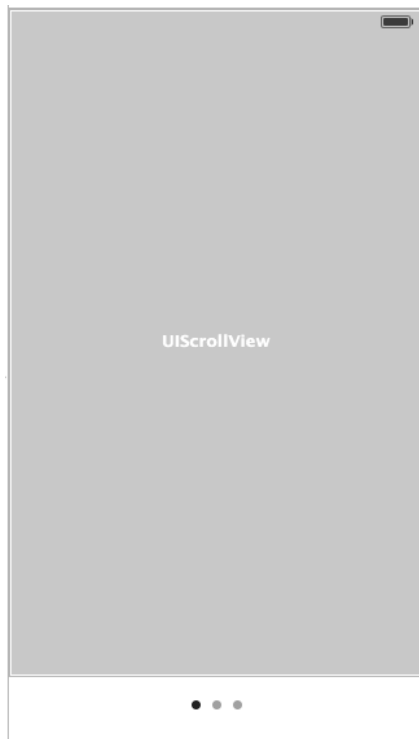
- Biết cách sử dụng sự kiện zoom của ScrollView.
- Biết cách tạo một control bằng code.

#### Gợi ý thực hiện:

- Khởi tạo một đối tượng view với chiều rộng và chiều cao truyền vào ta sử dụng phương thức “**initWithFrame: (CGRect)**”.
- Để tạo một **CGRect** ta sử dụng hàm **CGRectMake (float x, float y, float width, float height)**.
- Thêm một view vào ScrollView ta sử dụng phương thức “**addSubview: (UIView \*)**”.
- Để xét content size cho ScrollView ta sử dụng phương thức “**setContentSize: (CGSize)**”.
- Xét tỷ lệ thu nhỏ cho ScrollView bằng phương thức “**setMinimumZoomScale: (float)**”.
- Xét tỷ lệ phóng to cho ScrollView bằng phương thức “**setMaximumZoomScale: (float)**”.
- Để lấy ra các subview trong ScrollView ta sử dụng thuộc tính **subviews**.

## Bài tập 2.2. Scrollview kết hợp Page Control

**Đề bài:** Tạo một ứng dụng mới. Kéo thả ScrollView và Page Control vào màn hình giao diện như hình 2.1 bên dưới:



Hình 2.1



Hình 2.2

Yêu cầu thực hiện:

- Thêm 8 hình vào project.
- Dùng vòng lặp để khởi tạo 8 Image View và thêm Image View vào subview của Scroll View. Trong đó: **ImageView** có chiều rộng và chiều cao bằng với chiều rộng và chiều cao của **ScrollView**. Các hình được sắp xếp liên tiếp nhau theo hàng ngang. Do đó **tọa độ x** của mỗi hình khác nhau, cụ thể như sau:
  - Hình 1:  $x = 0$
  - Hình 2:  $x = x$  của hình 1 + chiều rộng của hình 1
  - Hình 3:  $x = x$  của hình 2 + chiều rộng của hình 2.
  - ...
- ScrollView cho phép phân trang, khi scroll sang trang mới thì currentPage của Page Control cũng được chuyển theo.

**Mục tiêu:**

- Biết cách sử dụng ScrollView kết hợp với Page Control

**Gợi ý thực hiện:**

- Bật chế độ xem trang bằng phương thức “**setPagingEnabled**”.



- Để bắt sự kiện scroll sang trang mới ta sử dụng delegate của Scroll View, áp dụng protocol UIScrollViewDelegate và triển khai phương thức scrollViewDidEndDecelerating tại lớp ViewController
- Trang trong Page Control được bắt đầu từ 0, để tính trang hiện tại ta thấy:
  - Trang 0 chứa Hình 1 có tọa độ  $x = 0$
  - Trang 1 chứa Hình 2 có tọa độ  $x = \text{chiều rộng của hình 1}$
  - Trang 2 chứa Hình 3 có tọa độ  $x = \text{chiều rộng của hình 1} + \text{chiều rộng của hình 2}$ .
  - Mà **chiều rộng hình 1 = chiều rộng hình 2 = chiều rộng của ScrollView**. Do đó ta có thể viết lại như sau:
    - Trang 0 có tọa độ  $x = 0 \times \text{chiều rộng ScrollView}$ .
    - Trang 1 có tọa độ  $x = 1 \times \text{chiều rộng ScrollView}$ .
    - Trang 2 có tọa độ  $x = 2 \times \text{chiều rộng ScrollView}$ .
  - Như vậy ta chỉ cần lấy **tọa độ x chia cho chiều rộng của ScrollView** là sẽ được trang hiện tại.
- Để lấy tọa độ x của content đang hiển thị trên ScrollView bằng thuộc tính **contentOffset**.
- Để xét giá trị trang hiện tại cho Page Control ta sử dụng thuộc tính **currentPage**.