

Exercise

When solving a complex problem, you should always break it down into smaller (and smaller) problems and solve them individually. This technique is called *Divide and Conquer* in computer science.

To build this app using this approach:

1- First, focus on the placement of elements: ensure that all elements are rendered at the right position.

For the buttons, don't worry about using an icon. Use "Previous" and "Next" as their text. Both buttons should be 5% from the top and 5% from the left/right side of the page.

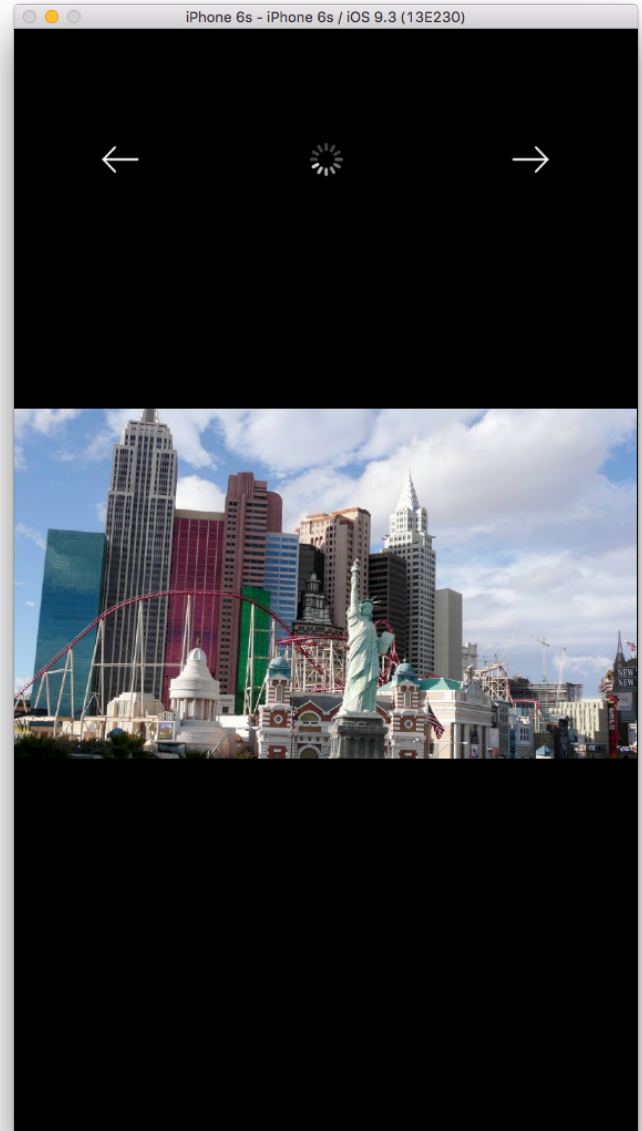
For the activity indicator, set `IsRunning` to true so it's displayed at all times.

For the the image, allocate the entire page using `AbsoluteLayout.LayoutBounds`. Set `Aspect` to `AspectFit` and use a hardcoded source:

<http://lorempixel.com/1920/1080/city/1>

2- Replace the text with the icon: once you get the layout working, replace the text for the buttons with the icons I've supplied. These icons are different across different platforms.

3- Render activity indicator dynamically: bind the `IsRunning` property of `ActivityIndicator` to `IsLoading` property of the `Image`. Now, when you run the application, activity indicator appears for a few seconds and then it disappears.



4- **Handle the click event of the buttons:** for this exercise, use the following URL to load the images:

<http://lorempixel.com/1920/1080/city/{id}>

where {id} is a number between 1 to 10. You need to keep track of the current ID in the code-behind. As the user taps the previous and next buttons, ID is updated and the image is refreshed. If the user taps the next button while viewing image #10, {id} should be reset to 1.

Make sure to disable caching for the image so you can see the activity indicator appearing every time the image changes.

Note that the order of elements in the `AbsoluteLayout` matters. If you put the image *after* the buttons, your buttons won't fire the clicked event because the image will be on top of them, even though it appears to be in the middle of the screen. This is because earlier we allocated the entire page to the image. To have a better understanding of what is happening here, set the `BackgroundColor` of the image to yellow. You'll see that it takes the entire page, even though the actual image is rendered in the middle. This is because we set `Aspect` to `AspectFit`, so the image is scaled down to fit in the page.

So, image should be the first child of the absolute layout. Other elements (activity indicator and buttons) come next.

With this approach, we focus on one and only one aspect of this app in each step. This is how you should think as a "software engineer".