# ICE Hello RIT-2

## Animation, Lambda functions and an Optional value

### Overview

Let us go ahead and add some animations to our app. This involves some more advanced features on Xamarin, but at the risk of getting ahead of ourselves, it’s worth it because we get to see some nice effects quickly.

1. After the code that added all of the controls we will add the following code:

var tapGestureRecognizer = new TapGestureRecognizer();

tapGestureRecognizer.Tapped += async (s, e) => {

await topLabel.TranslateTo(100, 500, 1000, Easing.BounceOut);

};

mainStack.GestureRecognizers.Add(tapGestureRecognizer);

What does this do? It adds a tap gesture to the mainStack object. On a tap, it will call the async method and move the topLabel object to the new location in 1 sec.

1. The interesting thing about this the async Lambda function that get called when the stack is tapped. You could also use a function and set the Tapped callback like this:  
     
   tapGestureRecognizer.Tapped += someFunction;

Some people do not like to use lambda function like this, but they are very common.

1. Experiment with the animation, change the position, the time, or the Easing value to see what they all do.

Valid animation function include:

TranslateTo – moves the control  
RotateTo – rotates the control

ScaleTo – scales the control

FadeTo – animates the opacity

1. ould now see your project in a Visual Studio window.
   * The **Solution Explorer** will have an hierarchy of the 3-4 projects in the solution.
   * To get to the project settings of the iOS or Android program, right click on that project and then go down to **Properties**.

*Optional Challenge: Animate all of the controls using different types of animations*

**Submission: ZIP and Post to the dropbox before the due date**