Chapter 3

* The MVC Paradigm – Logical way of dividing the code that makes up a GUI-based application; Ensures maximum reusability.
  + Model – Classes that hold the application’s data
  + View – Windows, controls, and other elements that the user can interact with
  + Controller – Code that binds model and view
* Launch screen storyboard file – An asset catalog for storing images our app uses
* Launch screen storyboard file and Info.plist implement classes:
  + Application delegate
    - AppDelegate.swift – Is implemented at certain well-defined times during an application’s execution
  + View controller class – responsible for displaying the view in the storyboard
    - Viewcontroller.swift – The implementation file for the view controller that owns the view you’re looking at
* Outlet – A property that allows the controller class to refer to objects in a storyboard or nib file
* Outlet properties have to be declared as optional with either ! or ?
  + ! must be set later so it’s not nil
* Action – Methods triggered in the controller class to interface objects in our storyboard or nib file.
* Action fields
  + Event – Lets you specify when the method is called
  + Arguments – Lets you choose between 3 different method signatures that can be used for action methods
* Automatic Reference Counting (ARC) – Memory management
* Auto Layout – using constraints to specify how you want controls to be placed
* 3 Issues with Auto Layout:
  + Not enough constraints
  + Ambiguous constraints
  + Constraint position or size will be off
* Align button – Aligns the selected view relative to another view
* Pin button – Lets you set the position of a view relative to another view
* Resolve Auto Layout Issues button – Corrects layout problems