Notes – Creating the entire Circuits App process (Copying Match App steps)

Lesson 20:

* Create the storyboard
* Add reuse identifier for Collection View Cell (the way to reference the prototype cell in the code). Ours will be “CircuitCell”

Lesson 21

* Create the Card Model Class (Ours: Circuit Model)
  + One function – getCircuits() that returns an array of circuits
* Create Card Class (Ours: Circuit)
  + Include whatever properties (Ours: Name, exercise array, color)
* Call the getCards (ours: getCircuits) method in the viewDidLoad function
  + Assign the returned array to a variable. Above the function, one variable creates an instance of the CircuitModel, and one creates an array of type circuit to hold the created array when getCircuits is called inside the function
* Build subclass of the collection view cell
  + Create a CocoaTouchClass as a subclass of UICollectionViewCell named CardCollectionViewCell (Ours: CircuitCollectionViewCell)
  + Connect/expose the two image views (Ours: the Label) so that we can manipulate them through code
    - Change the Custom Class of the CircuitCell (in the storyboard) to the custom class (Ours: CircuitCollectionViewCell
    - Create IBOutlets to the label
      * Open assistant editor (storyboard on one side, class CircuitCollectionViewCell on other) and ctl + drag from document outline
      * Name frontImageView (Ours: circuitLabel)

Lesson 23

* Expose our collection view to the view controller so they can communicate
  + Open assistant editor (storyboard on left, class ViewController on right) and create outlet named collectionView
* Set the view controller as the object supplying data to collection view and handling user interaction events from the collection view
  + State that the view controller conforms to the correct protocols by adding UICollectionViewDataSource, UICollectionViewDelegate as superclasses of ViewController. This should cause an error because the required methods are missing. Click fix to create the required stubs (These are both part of the Data Source, the delegate has no required ones).
    - In numberOfItemsInSection, return the number of items -cardsArray.count (Ours: circuitsArray.count)
    - In cellForItemAt, configure each cell and return it
      * To be memory-efficient, use the collectionView.dequeueReusableCell method to reuse invisible cell’s space in memory (if none, creates new object)
        + The type of cell returned is based on the Reuse Identifier (Set for the cell in the storyboard) – Use the one set earlier! (Ours: CircuitCell)
      * Attach the dequeued cell to a constant
      * Configure (Can do later) & return the cell/constant
* \*\*Set the view controller object as the delegate & data source of the collection view\*\*!!
  + In the viewDidLoad – collectionView.dataSource(or .delegate) = self

At this point, you should be able to run the project and see the correct number of cells

Lesson 26

* In cellForItemAt method, cast the dequeueReuseableCell as CardCollectionViewCell (Ours: CircuitCollectionViewCell)
* Create a configureCell method in the CircuitCollectionViewCell and call it