Building Conway's Game of Life

The Game of Life consists of of a grid of square cells which are either alive or dead. Each cell will live or die in the next generation based on its neighbouring cells in the current generation according to the following rules:

1. Any live cell with fewer than two live neighbours dies
2. Any live cell with two or three live neighbours lives
3. Any live cell with more than three live neighbours dies
4. Any dead cell with exactly three live neighbours becomes a live cell

You can also see an explanation of the rules here <https://www.youtube.com/watch?v=0XI6s-TGzSs>

**Coding challenge**

Starter code: <http://plnkr.co/edit/eF3M1wMuI9avMi3tLDcv?p=preview>

The following steps correspond to marked TODOs in the code.

1. Add the angular.js script to index.html. You can get it from here <https://angularjs.org/>.
2. There is an Angular module named "life" defined in script.js. Use ng-app to initialize angular with the "life" module on the <html> tag. Try adding a binding inside the <body> to verify that angular is working. e.g. {{21 \* 2}}  
   **Sample solution:** <http://plnkr.co/edit/Hw2dmeOxreOc66vLwQYF?p=preview>
3. In script.js register the LifeCtrl constructor function as an angular controller named “LifeCtrl”.
4. In index.html use ng-controller tell angular that you want to use the LifeCtrl controller inside the html tag. Alias the controller as 'ctrl'. Try binding to a variable in your controller to test that it is working. e.g. {{ctrl.cells}}  
   **Sample solution:** <http://plnkr.co/edit/9ZOqNKWZuQ0JZ6oCqzme?p=preview>
5. Remove the hardcoded rows and cells in the table and dynamically generate a row (<tr>) for each array in ctrl.cells and generate a cell (<td>) for each cell in a row.  
   **Hint:** you’ll need to use the ng-repeat directive.
6. Dynamically add the alive class to a cell if it is alive.  
   **Hint:** you’ll need to use the ng-class directive.
7. Toggle whether a cell is alive when you click on it.  
   **Hint:** you need to use the ng-click directive.  
   **Sample solution:** <http://plnkr.co/edit/3EIFsMdVTf1t2cR9N4mf?p=preview>
8. In script.js implement the getNextGeneration() function. It should iterate over all the cells for the current generation and return an array of all the cells for the next generation. every 100 milliseconds call getNextGeneration() and replace the current generation of cells with the next generation of cells.  
   **Hint:** There are some helper methods in the controller to help you implement getNextGeneration()  
   **Hint:** you'll need to use the $interval service to accomplish this.  
   **Pro Tip:** you can dramatically improve the performance of your ng-repeats by adding "track by $index" to them. Try setting the interval down to 0 ms and notice it runs faster when you add "track by $index". This will be significantly more noticeable on a bigger grid. <http://www.codelord.net/2014/04/15/improving-ng-repeat-performance-with-track-by/>   
   **Sample solution:** <http://plnkr.co/edit/1AWhLaLwjVvTWr1NGQVT?p=preview>
9. Create a variable to store the desired width/height of the grid (e.g. ctrl.size = 30;). Then dynamically create an N by N grid where each cell starts out with a 40% chance of being alive. Assign this dynamically created grid to ctrl.cells instead hard coding it as SAMPLE\_CELLS\_FIGURE\_EIGHT  
   **Sample solution:** <http://plnkr.co/edit/Rmpwsfcl5RGEwq9JhlnW?p=preview>
10. Reset the alive property of each cell so that it has a 40% chance of being alive when the randomize button is clicked
11. Set the alive property of all cells to false when the clear button is clicked  
    **Sample solution:** <http://plnkr.co/edit/WMeejkCAbPTnm7vhXtij?p=preview>
12. Toggle whether the game is running when you click Play button. While the game is running change the text of the play button to “Pause”. While the game is paused add the class “paused” to the <table>.  
    **Hint:** you’ll need to use the ng-show directive to dynamically change the text of the play button  
    **Sample solution:** <http://plnkr.co/edit/8lHm13ylUAl6En0edMbc?p=preview>

Sample apps for reference

Two way binding - <http://plnkr.co/edit/gmjVs7dwmjwRNaCq0LBd?p=preview>

Carousel - <http://plnkr.co/edit/0xnbTKo2zBhAIV7OBD9L?p=preview>

Todo: <http://plnkr.co/edit/4qCEX99eSXSro5yiDaOE?p=preview>