3 - Reactive Web Programming and Streams

## Learning Outcomes

On completion of this lab you will have:

* Learn how to handle asynchronous events using ReactiveX/RxJS streams

## Organisation

Please complete the exercises individually.

## Grading

This worksheet is worth up to 10% of your overall module grade.

You may work on this worksheet during lab 5 and lab 6 with instructor assistance. You may also be requested to demonstrate your submission to the lab instructor in order to receive credit.

## Submission

The deadline for submission is Sunday Nov 17th 2019 @23:59 through Brightspace

* The work and submission workflow is as follows:
* Create a sub-folder for each problem below. problem-1, problem-2, lecture-review.
* Put your solution for each problem in their respective folders.
* If you could create a mark down file for the lecture review questions (ie answers.md)
* When you are finished developing your worksheet solution, compress and zip all problems into one zip file. Name this ***<student-id>***-lab-3.zip
* ***<student-id>*** is something like C12345678
* Upload to Brightspace Lab 3 / Lab 3 - Upload

## Demonstration

10-12 names will be asked to demo this project.

## Resources

You are free to research whatever you need to solve the problems in this lab. Some recommended resources include:

* <https://gist.github.com/staltz/868e7e9bc2a7b8c1f754>
* <https://github.com/ReactiveX/rxjs>
* <https://github.com/ReactiveX/rxjs/blob/master/docs_app/content/guide/v6/migration.md>
* <https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API>
* <https://www.learnrxjs.io/>
* Take the free online first lesses from Egghead - [Introduction to Reactive Programming](https://gist.github.com/staltz/868e7e9bc2a7b8c1f754)
* <http://reactivex.io/rxjs/manual/overview.html#introduction>

## Problem Sets

Provide Javascript ES6 code for the following problems using in your own development environment

|  |  |  |
| --- | --- | --- |
| 1 | Convert your notes application from **worksheet 1** to use RxJS streams to handle the keyboard events **instead of** the method you originally used (most likely event handlers) | 30 |
| 2 | Implement a count down timer similar using RxJs. The UI would look something like the following (taken from timer-tab.com – you can ignore the colours etc from image – this is just to get an idea of UI) | 40 |

## Lecture Review Questions

(1-2 paragraphs/bullets, etc) answers for the following. It is important that you have reviewed the module material in advance

|  |  |  |
| --- | --- | --- |
| 1 | Explain what is meant by the stream abstraction. What is the relationship between streams and the observer pattern? What are streams useful for modeling and when might you use them in Rich Web development? | 10 Marks |
| 2 | Assume that you are building an interface to an API in your Rich Web App. Describe in detail how you could use the RxJS library to handle asynchronous network responses to API requests. In your opinion, what are the benefits to using a streams library for networking over, say, promises? And what do you think are the downsides? | 10 Marks |
| 3 | Consider three asynchronous tasks, A,B & C. What are the consequences of these functions sharing global state? What is a good practice to alleviate any problems associated with this? | 10  Marks |

## 