3 - Reactive Web Programming and Streams

1. Learning Outcomes

On completion of this lab you will have:

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

2. Organisation

Please complete the exercises individually.

3. Grading

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

You may work on this worksheet during lab 8, 9 and 10 with instructor assistance.

4. Submission

The deadline for submission is Sunday Nov 27th 2021 @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
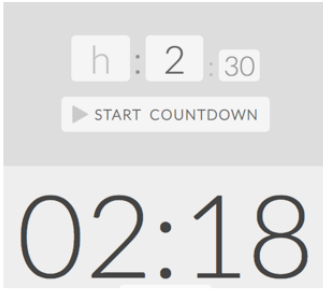- Upload to Brightspace Lab 3 / Lab 3 - Upload

3. 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 lessons from Egghead - Introduction to Reactive Programming
- http://reactivex.io/rxjs/manual/overview.html#introduction

4. Problem Sets

Provide RxJS 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 mouse 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) - User is presented with (A) on page load, enters 2m 30s - When he inserts a count down time, the timer starts counting downwards (B) - Finishes at 00 (C)	40
3	Task 3 to be added next week	30

5. Exercises

Consider your answers to the following questions:

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?
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?
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?