Task:

Create a sample MVC(S) Robotlegs project using Robotlegs version 1.5.2. The project must implement the following concepts. Please make sure the concepts are implemented in a practical and meaningful way.

-Context

-Model

-View

-Mediator

-Controller (Command)

-Service

-Factory

-Interface

-Signals

-Unit tests

-Tweens (using Greensock)

You are free to choose the functionality of the application if you think another application would showcase the above concepts better, but an example of what we expect would be an application that loads images and presents them in a collage-like fashion (calculate the optimal layout based on each image’s aspect ratio to obtain the minimal unused screen space). After an image is clicked, it fades-out and is removed from the collage and another image is loaded, fades-in, and the layout is updated. The removed images must get garbage-collected.

We will be looking at your ability to understand the concepts and utilize them in a practical way. Code readability (variable naming, code formatting), simplicity and modularity are values we will be looking for. Also animation skills and tying animations into the otherwise synchronous state machine.

DONE.

I did example of the application which performs search images and navigation by pages.

I have used Google AS3 API.

Please host this project on GitHub to showcase your knowledge of Git.

DONE.

Extra task. Create a Maven configuration pom.xml for the project. It should include a test runner and working build config.

MISSED.

Sorry, don't have experience with Maven.

Need more time to research and understand main concept of Maven.

Extra task. Write method that takes two sorted arrays of integers (can be of any range, numbers can repeat, sizes of arrays can be different) and returns array of n + m length which is sorted in the same order. Do not use any type of built-in sorting.

DONE.

Can find 'concat' method in the 'src.tuvisum.util.ArrayUtil.as'

Use it like:

**var** concatedArray : **Array** = ArrayUtil.*concat*([1, 1, 3, 6, 89, 2, 106], [152, 4, 4, 3, 560]);

trace(**'concatedArray: '** + (concatedArray));

Extra question. Explain what does this code do, how it works and what’ s wrong with it:

var xmlData:XML =

                <root>

                                <node myVal="1">data</node>

                                <node myVal="2"> data </node>

                                <node myVal="3"> data </node>

                                <node myVal="4"> data </node>

                                <node myVal="5"> data </node>

                </root>;

xmlData.children().(@myVal % 2 && trace(@myVal));

XML.children() returns an XMLList object of the XML object's children. Expression XMLList.(condition) returns XMLList with items satisfied some condition which set. Expression @myVal % 2 returns 1 (boolean true) in case of attribute myVal is odd in other case one returns 0 (boolean false).

trace():void does not return any value and can be considered as undefined. In this way condition (@myVal % 2 && trace(@myVal)) equals to false in case of members values (false && undefined), in this case trace() function is not called. Condition (@myVal % 2 && trace(@myVal)) equals to undefined in case of members values (true && undefined).

As result xmlData.children().(@myVal % 2 && trace(@myVal)) returns XMLList which is empty. In the same time trace() prints odd values: 1, 3, 5 in case of @myVal % 2 equals true.