Assignment Requirements

**In this Assignment, you will build a more refined 3D App using the skills, technologies and components acquired by completing lab tutorials from week 1 to week 8.**

To meet these assignment requirements a successful student is required to build an X3D based **Coca Cola Branded** **3D App**.

The suggested approach is to use the lab based skills you have acquired exploiting HTML5, CSS3, X3D (X3DOM), JavaScript (you may use JavaScript APIs such as jQuery, and a JavaScript frontend framework, e.g. Bootstrap), AJAX for asynchronous update of HTML contents, JSON for data transfer from the backend to the frontend, PHP to access data from a data source, e.g. SQLite database, and all configured in an MVC design pattern.

In detail, you are required to:

1. Redesign your existing lab tutorial based 3D App layout to provide a **new Coca Cola Brand**, you are specifically advised **NOT** to submit the brand designed during the lab work
   * You could take inspiration from existing Coca Cola brands (see <https://www.coca-cola.co.uk/brands>), while being mindful of copyright issues, and you may choose any three brands to represent your Coca Cola Product models, e.g. Coke, Fanta, Sprite, Dr Pepper …,
     + A simple way to do this would be to produce a Coke can, Fanta can, and Sprite can as your 3 Coca Cola Product 3D Models, this would give you access to the modelling marks defined in the rubric.
     + However, producing at least one of these Coca Cola Product 3D Models with more complexity, e.g. create a Sprite Bottle or a Coke Bottle, would automatically qualify for one of the 5 mark blocks reserved for ‘deeper understanding’.
2. You are required to integrate your 3D Models into a **Coca Cola Branded User Interface** — your 3D App.
   * + Your Lab 2 tutorial 3D models are acceptable as submission models
3. You should add appropriate levels of **interactivity** between your 3D models and your 3D App:
   * Your **3D model interactivity**, can be achieved using JavaScript to trigger interactive features.
     + For example, any animations applied to your 3D model could use an HTML onClick method or the jQuery.Click() method, etc. and appropriate JavaScript or jQuery code to trigger the animation — see Lab 4 and 6.
     + You **are strongly advised to use X3DOM (i.e. X3D inline in your HTML) for your Web 3D format**, however it is possible to use other methods such as three.js.
   * Your 3D App page:
     + **You must provide** a button to switch between wireframe mode to view your models — example code for this is provided in the Lab 9 Coca Cola VM on the Live Feedback Site.
   * Good use of JavaScript examples (not exhaustive):
     + Implement content swapping — remember how to change page elements with JavaScript, this can be applied not only to page elements. JQuery also abstracts this functionality in their show and hide functions.
     + Implement a texture panel in the HTML5 to select new textures on the model.
     + Implement a 3D model gallery to load new models in X3DOM.
     + Implement buttons to select different camera views.
     + Implement buttons to control lighting.
     + You are expected to exploit AJAX and JSON, with JQuery (or an alternative) to dynamically load web page contents from a JSON packet supplied by the backend — see Lab 6 and 7.

These features listed above are not exhaustive; it is up to you to make your own choices in relation to the proposed marking scheme.

1. Your **3D App** layout should have, as a minimum, the following responsive block elements:
   * A header — you should design a CSS logo, e.g. ‘My Coca Cola Brand’ with a strapline.
     + **You should not use images for the logo**, instead you should make effective use of CSS3.
   * A method for selecting new objects (i.e. 3D Object 1 to 3D Object n) via a **selection method in your user Interface** — remember the JavaScript gallery you created in Lab 6, you may be able to adapt this! Alternatively, you can completely redesign with other Bootstrap components.
   * A method for displaying the selected 3D object
     + You are advised, strongly, to use X3DOM (i.e. X3D inline in the HTML5)
   * A method for interacting with the 3D object and the 3D App web page, e.g. using JavaScript, in X3DOM
     + X3DOM provides a degree of [control built-in](http://x3dom.org/docs-old/navigation.html), but generally you will need to develop your own controls. Remember the Camera panel you created in Lab 4 to select Camera views in X3DOM — look at Lab 6 to see how to connect cameras to your models.
   * A method for displaying a description of the 3D objects displayed, which is synced with the object selection (e.g. JQuery.getJSON()) — Object Description.
   * You should integrate other media objects where appropriate, e.g. further information, video, audio, images, etc. You can organise this by ‘swapping’ out media, e.g. swap the X3DOM view for an Image view, etc.
2. To gain the marks associated with the MVC design pattern, clearly you mustuse MVC to implement your 3D App — you may adapt the codes from Lab 7 and 8 tutorials, or use other MVV methods, see the Live Feedback Site. Not: Lab 9 is not a complete version and should not be copied verbatim. In general, you are advised to:
   * Use AJAX methodology to asynchronously update content on your 3D App exploiting JSON to exchange data between the view and a ‘server side’ model (the data store being SQLite)
   * Use PHP scripting to access data from the data store (SQLite).
   * As a minimum, place your Object Descriptions in the data store; you might be able to think of other elements such as appropriate metadata, interaction properties, etc. that can be stored.
   * Note, while you are allowed to exploit a different frontend framework, you are limited to the backend PHP framework because of the University ITS Web Server. **More specifically, your backend technology should run on the ITS Web Server**.
3. **You must** provide a link to a GitHub archive of your complete codebase and 3D models, and code documentation.
   * You will need to set up a GitHub account. This allows us to very easily and in a consistent way check your codebase.
   * Make sure that your codebase also includes your complete **3D model archive** for your 3 models. **For example, check out in 3ds Max how to save an archive**
4. Your 3D App may also contain home page contents, and other page contents (content swapping) for items such as ‘About’, Statement of Originality’, Site Map, References, and GitHub links
5. You are **allowed to adapt open source code** (e.g. check out the X3DOM examples, etc.), **but make sure you cite the source in the object description and appropriate references**. However, clearly illustrating your own work in a system building approach will gain most marks.
6. You must publish and submit your work strictly according to the Canvas submission criteria:
   * **E-Submission Web Link submitted on Canvas:**
     + users.sussex.ac.uk/~username/3dapp/assignment/submission.html

This **Web Link** will access your submission.html file containing:

* + - **Web Link:** to your 3D App on the ITS Web Server URL:
      * + users.sussex.ac.uk/~username/3dapp/assignment/index.php
    - **Web Link:** to yourGitHub 3D App Codebase URL E-Submission
    - **Web Link:** to yourGitHub VRML/X3D Models URL E-Submission
    - **Text:** 3D App Deeper Understanding Statement E-Submission
      * + A list of deeper understanding statements, examples given
    - **Text:** ITS Web Server Date Time Last File Upload E-Submission
      * + [Separate instructions](https://canvas.sussex.ac.uk/courses/23366/pages/instructions-for-finding-last-file-uploaded?module_item_id=1209161) are given on how to use find your last file uploaded on the ITS Web Server using, for example, the command line and a shell script or other utilities

**NOTE: YOU ARE NOT ALLOWED TO SUBMIT YOUR 3D APP ON AN EXTERNAL ISP SITE, YOUR SUBMISSION MUST BE ON THE ITS WEB SERVER. IF YOU SUBMIT ANYWHERE ELSE YOUR WORK WILL NOT BE MARKED AND YOU WILL ACHIEVE A ‘NO SUBMISSION’!**