

# CS324 Coursework Assignment

## WebGL/GLSL 2021

Shan E Ahmed Raza, Rossella Suma

November 2021

### 1 Introduction

Your coursework assignment task is to write a graphics program using WebGL and GLSL. You should use what you have learnt in the practical laboratories. Note that the program **must** run on the department’s Linux machines on Google Chrome browser and you should test this out before you submit your code for assessment.

Your task is to create a video game taking this [example](#) as inspiration. The demo code can be downloaded [here](#).

You can modify the game play however you see fit, so use your creativity! Table 1 lists the essential (labelled as “E”) and the desirable (i.e., not mandatory, labelled as “D”) requirements.

You can use any of the materials from the labs or adopt external libraries. Please note that any code that is borrowed from other sources ***has to be acknowledged*** in the list of references and in the code itself.

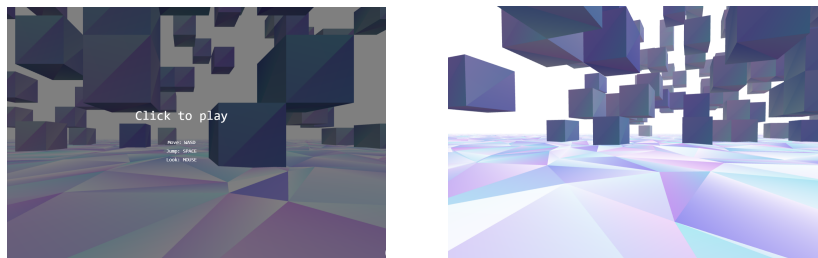


Figure 1: Screenshots of the inspiration demo.

We expect that to complete the task well, you need to spend about a week on it (20-30 hours). Please, try to complete the required functionality in the first instance and work on the cosmetics and desirable features later on.

If you have any questions, please get in touch with the module faculty. We have an [FAQ page](#), so please check this page before submitting any new questions.

Good luck and have fun!

Table 1: List of Essential (E) and Desirable (D) Features

Requirement	Description	E/D
Interactivity	The player must be able to interact with the game using mouse <b>and</b> keyboard.	E
Camera	The game should offer at least <b>two</b> different camera views (e.g., First Person Perspective, Top-Down View etc.) and let the user be able to switch between the views.	E
Menu	At least <b>one</b> initial menu containing a short description of the game and how to play it.	E
Levels	The game should have at least TWO levels.	E
Models	You should create <b>yourself</b> at least <b>one</b> model using Blender and import it into your game. (More on this in the supporting document and the assessment section).	E
Lighting	You should create a scene and add at least <b>two</b> different light sources.	E
Textures	The scene should have at least <b>one</b> texture introduced by you.	E
Heads-Up Display (HUD)	You might want to visualise on screen useful info for the player using transparency e.g., health bar, points, remaining lives etc.	D
Sounds	You might want to add sounds to your game e.g., play a sound every time the player scores a point etc.	D
Collision Detection	You might want to detect collisions between different objects e.g., <a href="#">Raycaster</a> .	D
Physics	You might want to add physics to your game e.g., gravity.	D
Environment Map	You might want to include environment maps to add more realism to your scene.	D
Animation	You might want to introduce simple animations to your scene.	D

**Supporting Document** You need to submit a supporting document (**no more than 1000 words** in length). Please indicate the word count at the beginning of the document. Add at least 4 screenshots of the **intermediate stages** of the creation of your Blender 3D model. **Note:** References are not included in the word count.

## 2 Assessment Criteria

The coursework marks (out of 20) will be allocated **equally** to the following four areas:

<b>Functionality</b>	How much of the <b>required functionality</b> is implemented by the solution and how well it has been done. Note: Features labelled as desirable (D) may allow you to achieve a maximum of 10% bonus marks, however, this is conditional to ALL the essential features been implemented properly.
<b>Code</b>	Appropriate features of WebGL/GLSL have been used in the solution. Things to consider are <b>readability</b> (i.e., meaningful names to auxiliary variables and functions, proper use of indentation), <b>handling of exceptions</b> , <b>comments</b> - especially for complex functions explain what you are trying to achieve. <b>Organisation</b> of the code.
<b>Look &amp; Feel</b>	The general <b>quality</b> of the graphics produced, and the ease of use of the solution. In particular, for the Blender model, model <b>opens and renders</b> (with Eevee or Cycles) without errors; model is made of <b>Quads and triangles</b> only; there are <b>no multiple-stacked polygons</b> ; there are <b>no open edges</b> ; <b>unique and meaningful names</b> for objects, materials, textures; model <b>centred at origin</b> ; <b>clean UVs</b> ; and <b>min texture resolutions</b> is $512 \times 512$ .
<b>Communication skills:</b>	The <b>readability</b> and <b>quality</b> of the supporting documentation. The document should be <b>clear and concise</b> and contain: <b>brief description</b> of the game; illustrate the <b>main design aspects</b> , with <b>design decisions</b> clearly justified; <b>highlight which features</b> have been implemented in your solution; and how <b>specific WebGL/GLSL and three.js features</b> have been employed. The contents should describe well the features of the solution. The document is of an <b>appropriate length (max 1000 word)</b> . <b>Instructions</b> (if needed) are given for running the game e.g., running http server.

## 3 Collaboration and Plagiarism

As for all pieces of assessment you undertake, the submission must be your **own** work. You may have been working in pairs in the practical labs, but this piece of

work must be done individually. Discussion of ideas with your colleagues is OK, but the final piece of work must be your own.

Also, you must not copy without attribution from work you may have found online. Using any code that is given in the laboratory tutorial examples is permitted.

We will use anti-plagiarism software to look for copying between submission, with external sources and past submissions. Any breaches of the rules will incur a severe penalty.

## 4 Deadlines and Submission Formats

The deadline for this coursework is **12pm Monday, 17th January 2022**.

Please submit a **single** ZIP file with the following structure:

- Solution: Folder containing your original source files (`.html`, `.js`, `.css`)
- Extra Libraries: Anything useful to your project that you have not developed yourself
- Assets: Anything used in your game that is not HTML, CSS, or JavaScript goes in here: textures, 3D models, fonts, sounds, and so on.
- PDF of a supporting document which describes your code.

## 5 Some useful resources

- [Games Database](#), it might give you some ideas on what game you want to implement;
- Fullerton Tracy, *Game Design Workshop*, 2014, (available through Warwick Library): Book on how to create a game;
- [Three.js homepage](#).

*Shan E Ahmed Raza, Rossella Suma, November 2021*