Module Descriptor

Title	JavaScript Games: Programming Fundamentals		
Session	2026/27	**Status**	Published
Code	COMP09090	**SCQF Level**	9
Credit Points	20	**ECTS (European Credit Transfer Scheme)**	10
School	**Computing, Engineering and Physical Sciences**		
Module Co-ordinator	Derek Turner		

Summary of Module

This module is intended to develop student's abilities in the use of core web technologies: HTML5, CSS and JavaScript generated through TypeScript. This is done within the context of game development.

Students are introduced to HTML5 elements to add structure to web documents.

Cascading style sheets are added to control the appearance of structured elements.

The common features of CSS will be presented, and students will also be introduced to the range of CSS frameworks available which includes Bootstrap, Bulma, Material Design Lite, Materialize, Milligam, Pure, SemanticUI, Skeleton, Tacit, Tailwind, UIkit and encouraged to explore one of these to enhance the style of their web page design.

JavaScript is the most popular programming language because it can be applied to both client and server-side programming of web applications. Typescript is a Microsoft extension of JavaScript which adds structures and supervision to reduce code errors. Programmes written in Typescript are transcoded to JavaScript in the background.

The basic syntax and common constructs of TypeScript will be introduced.

The object oriented and functional approaches to programming will be contrasted.

The primary 3D gaming engine used will be BabylonJS. Students will gain a working knowledge of the use of BabylonJS code for aspects of scene production and interaction.

By the end of the module students will have a solid foundation in the fundamental elements required for JavaScript-based games and which could be used to develop an expanded game within a follow-on module.

• This module embeds the key "I am UWS" graduate attributes and in particular: Universal (critical and analytical thinking, Emotionally intelligent, Collaborative, Research-minded), Work Ready (digitally literate, problem solver, effective communicator, Motivated, Potential leader, Ambitious) and Successful (Autonomous, Innovative, Driven, Transformational)

Module Delivery Method	**On-Campus**[1] ☒	**Hybrid**[2] ☐	**Online**[3] ☐	**Work -Based Learning**[4] ☐
Campuses for Module Delivery	☐ Ayr ☐ Dumfries	☐ Lanarkshire ☐ London ☒ Paisley	☐ Online / Distance Learning ☐ Other (specify)	

Terms for Module Delivery	Term 1	☒	Term 2	☐	Term 3	☐
Long-thin Delivery over more than one Term	Term 1 – Term 2	☐	Term 2 – Term 3	☐	Term 3 – Term 1	☐

Learning Outcomes	
L1	Demonstrate and work with an understanding of the scope and defining features of HTML5, CSS and JavaScript, and Typescript, with an awareness of forefront developments.
L2	Apply JavaScript/TypeScript programming knowledge skills and understanding to a range of the principal elements required for the development of game scenes.
L3	Practise programming and design in ways that show awareness of own and others' roles and responsibilities.
L4	
L5	

Employability Skills and Personal Development Planning (PDP) Skills	
SCQF Headings	**During completion of this module, there will be an opportunity to achieve core skills in:**

[1] Where contact hours are synchronous/ live and take place fully on campus. Campus-based learning is focused on providing an interactive learning experience supported by a range of digitally-enabled asynchronous learning opportunities including learning materials, resources, and opportunities provided via the virtual learning environment. On-campus contact hours will be clearly articulated to students.

[2] The module includes a combination of synchronous/ live on-campus and online learning events. These will be supported by a range of digitally-enabled asynchronous learning opportunities including learning materials, resources, and opportunities provided via the virtual learning environment. On-campus and online contact hours will be clearly articulated to students.

[3] Where all learning is solely delivered by web-based or internet-based technologies and the participants can engage in all learning activities through these means. All required contact hours will be clearly articulated to students.

[4] Learning activities where the main location for the learning experience is in the workplace. All required contact hours, whether online or on campus, will be clearly articulated to students

Knowledge and Understanding (K and U)	**SCQF 9** A sound understanding of website production using HTML5, CSS and JavaScript. A working knowledge of a JavaScript framework supporting the development of a game and graphic applications.
Practice: Applied Knowledge and Understanding	**SCQF 9** Ability to work with standard approaches to create graphic elements for games development implemented in JavaScript, HTML5 and appropriate frameworks.
Generic Cognitive skills	**SCQF 9** Identify and analyse code-based approaches to routine professional aspects of 3D scene implementation.
Communication, ICT and Numeracy Skills	**SCQF 9** Offer professional level insights, interpretations, and solutions to communicate plan and implement 3D game scene ideas.
Autonomy, Accountability and Working with Others	**SCQF 9** Exercise autonomy and initiative in designing and developing web 3D interactive scenes.

Prerequisites	**Module Code**	**Module Title**
	Other	
Co-requisites	**Module Code**	**Module Title**

Learning and Teaching	
In line with current learning and teaching principles, a 20-credit module includes 200 learning hours, normally including a minimum of 36 contact hours and maximum of 48 contact hours. Learning materials are provided as live in-class presentations and code demonstrations, supported by recorded video materials. Staff will provide feedback within face-to-face labs to guide and assist students working towards the creation of 3D scene elements and interactions.	
Learning Activities During completion of this module, the learning activities undertaken to achieve the module learning outcomes are stated below:	**Student Learning Hours** (Note: Learning hours include both contact hours and hours spent on other learning activities)
Tutorial / Synchronous Support Activity	12
Laboratory / Practical Demonstration / Workshop	36
Independent Study	152
Please select	
Please select	
Please select	
TOTAL	200

Indicative Resources

The following materials form essential underpinning for the module content and ultimately for the learning outcomes:

Development environment comprising:

Windows subsystem for Linux.

Docker desktop (for containerised development environment featuring Vite development server, Nodejs and typescript)

GitHub desktop as a convenient interface between local PC and cloud repository.

Visual Studio Code as a code editor (IDE)

Visual Studio Code extensions:

- Prettier as a code formatter
- Remote development which incorporates:

o WSL

o Dev Containers: allowing local code to be run and edited in a container

o Remote SSH

o Remote Tunnels

- Live server for website development

GIT to facilitate VSC connection to code repository.

Online Resources:

BaylonJs current documentation at http://doc.babylonjs.com

Course notes available via UWS VLE and external links.

Recommended reading:

Mcgrath, M. (2020). HTML, CSS & JavaScript. Leamington Spa, Warwickshire, United Kingdom: In Easy Steps.

Shenoy, A. (2018). CSS framework alternatives: explore five lightweight alternatives to Bootstrap and Foundation with project examples. California: Apress.

Chenard, J. (2017). Learning Babylon. Js.

Ferguson, R. (2019). Beginning javascript: The ultimate guide to modern javascript development. Berkerley, Ca.

Haverbeke, H. (2024). Eloquent JavaScript. [online] Eloquentjavascript.net. Available at: https://eloquentjavascript.net/.

Kereki, F. (2023). Mastering JavaScript functional programming- write clean, robust, and maintainable web and server code using functional JavaScript. Birmingham: Packt Publishing.

(N.B. Although reading lists should include current publications, students are advised (particularly for material marked with an asterisk*) to wait until the start of session for confirmation of the most up-to-date material)

Attendance and Engagement Requirements
In line with the Student Attendance and Engagement Procedure, Students are academically engaged if they are regularly attending and participating in timetabled on-campus and online teaching sessions, asynchronous online learning activities, course-related learning resources, and complete assessments and submit these on time. **For the purposes of this module, academic engagement equates to the following:** Submission of a 3D graphic scene exercises and active participation in 80% of scheduled sessions. The School of Computing, Engineering and Physical Sciences considers attendance and engagement to mean a commitment to attending, and engaging in, timetabled sessions. You will scan your attendance via the scanners each time you are on-campus and you will login to the VLE several times per week. Where you are unable to attend a timetabled learning session due to illness or other circumstance, you should notify the Programme Leader that you cannot attend. Across the School an 80% attendance threshold is set. If you fall below this, you will be referred to the Student Success Team to see how we can best support your studies.

Equality and Diversity
The University's Equality, Diversity and Human Rights Procedure can be accessed at the following link: UWS Equality, Diversity and Human Rights Code. Aligned with the University's commitment to equality and diversity, this module supports equality of opportunity for students from all backgrounds and learning needs. Using the VLE, material will be presented electronically in formats that allow flexible access and manipulation of content. This module complies with University regulations and guidance on inclusive learning and teaching practice. This module has lab-based teaching and as such you are advised to speak to the Module Co-ordinator to ensure that specialist assistive equipment, support provision and adjustment to assessment practice can be put in place, in accordance with the University's policies and regulations.
(N.B. Every effort will be made by the University to accommodate any equality and diversity issues brought to the attention of the School)

Supplemental Information

Divisional Programme Board	**Computing**
Overall Assessment Results	☐ **Pass / Fail** ☒ **Graded**
Module Eligible for Compensation	☒ **Yes** ☐ **No** **If this module is eligible for compensation, there may be cases where compensation is not permitted due to programme accreditation requirements. Please check the associated programme specification for details.**
School Assessment Board	Creative Computing
Moderator	Marco Gilardi
External Examiner	Professor Sylvester Arnab
Accreditation Details	TIGA

Module Appears in CPD catalogue	☐ **Yes** ☒ **No**
Changes / Version Number	

Assessment (also refer to Assessment Outcomes Grids below)
Assessment 1
Single assessed Component comprising: Practical Development Project – A portfolio website featuring 3D scenes and documentation according to scenarios specified in the module handbook 100%
Assessment 2
Assessment 3
(N.B. (i) Assessment Outcomes Grids for the module (one for each component) can be found below which clearly demonstrate how the learning outcomes of the module will be assessed. (ii) An indicative schedule listing approximate times within the academic calendar when assessment is likely to feature will be provided within the Student Module Handbook.)

Component 1							
Assessment Type	**LO1**	**LO2**	**LO3**	**LO4**	**LO5**	**Weighting of Assessment Element (%)**	**Timetabled Contact Hours**
Creative output/ Documented 3D scenes	☒	☒	☒	☐	☐	100	4

Component 2							
Assessment Type	**LO1**	**LO2**	**LO3**	**LO4**	**LO5**	**Weighting of Assessment Element (%)**	**Timetabled Contact Hours**
	☐	☐	☐	☐	☐		

Component 3							
Assessment Type	**LO1**	**LO2**	**LO3**	**LO4**	**LO5**	**Weighting of Assessment Element (%)**	**Timetabled Contact Hours**
	☐	☐	☐	☐	☐		
Combined total for all components						100%	4 hours

Change Control

What	**When**	**Who**
