STUDENT AND ACADEMIC SERVICES

PROGRAMME SPECIFICATION

Part 1: Basic Data	
Awarding Institution	UWE
Teaching Institution	UWE
Delivery Location	UWE
Faculty responsible for programme	Environment and Technology
Department responsible for programme	Computer Science and Creative Technologies
Modular Scheme Title	FET
Professional Statutory or Regulatory Body Links	TIGA
Highest Award Title	BSc(Hons) Games Technology
Default Award Title	
Interim Award Titles	BSc Games Technology Dip HE Games Technology Cert HE Games Technology
UWE Progression Route	n/a
Mode(s) of Delivery	Full time and Sandwich
ISIS Codes	**G611** G611 (SW) G61113 (FT)
Relevant QAA Subject Benchmark Statements	Computing
Approval Date	Jan 2020
Valid From	September 2020
Version	4

Part 2: Educational Aims of the Programme

The BSc(Hons) Games Technology has the following general aims:

1. To enable students to embark upon professional careers by developing problem-solving and other transferrable skills.
2. To enable students to work effectively and productively as a member of a team.
3. To develop study skills that will enable students to become independent, lifelong learners.

Part 2: Educational Aims of the Programme

4. To prepare students for progressing to study for higher degrees in computing and digital media.
5. To encourage the discerning use of reference material from a variety of sources.

The BSc (Hons) Games Technology has the following specific aims:

1. To provide skills in the design and implementation of computer games, including an understanding of the mathematical and technological principles required, as well as an exploration of the creative potential presented within the development of electronic games, and the cultural and technological contexts out of which they arise.
2. To provide practical skills in computer games development, including high and low level programming for a variety of deployment environments, such as dedicated consoles, desktop computers and mobile devices.
3. To develop the students' ability to make efficient, innovative and robust contributions to companies engaged in the development of computer games entertainment and related digital media.
4. To develop the students' understanding of the importance and mechanisms of project management, and associated tools, within computing, with particular reference to the development of computer games.

Part 3: Learning Outcomes of the Programme

The award route provides opportunities for students to develop and demonstrate knowledge and understanding, qualities, skills and other attributes in the following areas:

Learning Outcomes	**Teaching, Learning and Assessment Strategies**
A Knowledge and Understanding	
A Knowledge and understanding of: 1. Historical and cultural perspectives of computer game development, including principles and applications of games design, interactivity and user involvement, and related supporting technologies. 2. The mathematical foundations of computer games in 2D and 3D, and techniques used to simulate physical events. 3. Software design concepts, programming languages, methods, notations and algorithms, as applicable in modern Computer Games development.	Teaching/learning methods and strategies: Throughout the programme there is a strong emphasis on coupling theory with practice, and all modules have a strong emphasis on studio and/or other practical classroom work. Knowledge and understanding is typically developed initially by giving students the opportunity to implement examples of concepts listed on the left, progressing to more free-form synthesis and implementation of own ideas at levels two and three. The development of knowledge in many of the areas listed pervades the entire programme; nonetheless, specific examples of places where these outcomes can be identified are listed below, itemised and numbered accordingly:

Part 3: Learning Outcomes of the Programme	
4. The asset creation and level design process, its technical implications and the development of tools to support the computer games production pipeline. 5. Hardware architecture and supporting software technologies required for the production and deployment of contemporary computer games. 6. A range of advanced topics in Computer Games development, including physics and simulations, artificial intelligence (AI), networking, multi-core processing, low level programming and profiling/optimisation. 7. Professional, ethical and sustainability issues affecting the development and deployment of computer games within an international market place.	1. Acquisition is through a series of modules. Games Development Evolution and Principles of 3D Environments at level one, Play & Games at level two, with further consolidation through Commercial Games Development, Audio Visual Studio and the Creative Technology Projects at level three. Students with a particular interest in this area are offered a chance to develop specific technical skills in Games Technology: 101 or as a specialism through the optional Gameplay Programming module at level two. 2. The formal delivery of mathematical foundations is mainly through the Games in C++ and Game Engine Architecture modules at levels one and two respectively. This is expanded to introduce physics simulations through the Game Level Design module, also at level two. Option modules Game Engine Programming and Gameplay Programming at level two further explore the practical implementations of these concepts as appropriate within their respective contexts. Level three provides plenty of opportunities for students to further develop this understanding through Advanced Technologies, the Creative Technology Projects and Commercial Games Development. An emphasis is put on mathematical concepts being delivered in a Computer Games related context where possible, with plenty of topical, practical exercises to illustrate and cement taught concepts. 3. Delivery spans a majority of modules on the award, including Games in C++ and Games Technology: 101 in level 1, More Games in C++, Game Engine Architecture and option modules Game

Part 3: Learning Outcomes of the Programme	
	Engine Programming and Gameplay Programming at level two, as well as Commercial Games Development, the Creative Technology Projects and Advanced Technologies at level three. 4. Acquisition is through level one modules Games Development Evolution and Principles of 3D Environments, core modules Game Engine Architecture, Game Level Design and option modules Game Engine Programming and Gameplay Programming at level two. Further to this, students are expected to demonstrate an understanding of these topics throughout their final year work. 5. Fundamental concepts are introduced through Games Development Evolution and developed through Game Engine Architecture. Further depth is explored through option modules Game Engine Programming and Gameplay Programming at level two. The third year places an emphasis on the practical application of software technologies through modules Advanced Technologies, Commercial Games Development and the Creative Technology Projects. 6. Acquisition is through Games Technology: 101, More Games in C++, Commercial Games Development, Advanced Technologies and the Creative Technology Projects along with the option module AI for Creative Technologies in level 3. At level two, students particularly interested in the in depth implementation aspects of these topics are offered chance to specialise within the optional Game Engine Programming module. 7. Delivery of these topics forms a strand through all three levels starting with Games Development Evolution at level one, continuing through Play & Games at level two, and culminating in a

Part 3: Learning Outcomes of the Programme	
	arrange of option modules in level 3 from Mobile Applications or Audio Visual Studio and in the core module Commercial Games Development in the final year. Throughout the programme, the learner is encouraged to undertake independent reading (suggested via module indicative reading lists and staff recommendation) and development to supplement and consolidate what is being taught and learnt to both broaden and deepen their understanding of the overall discipline. NB. It is important to note that sound software development principles and practices are taught throughout, with a firm grounding in modern software engineering, providing transferrable skills in the wider disciplines of Computing/Computer Science/Software Engineering.
	Assessment: All outcomes are assessed in core modules, through a variety of methods including: • Projects • Exams • Portfolio assignments • Coursework assignments • Presentations • Essays • Vivas
B Intellectual Skills	
B Intellectual Skills Students on this programme will develop their intellectual skills in areas of: 1. Critical thinking 2. Analysis 3. Synthesis of different types of information 4. Evaluation 5. Problem solving 6. Appreciate problem contexts 7. Balance conflicting objectives 8. Creative and interpretive thinking	Teaching/learning methods and strategies: Programmes in the general areas of Computer Science and Creative Technologies readily lend themselves to the development of the cognitive skills listed on the left. The central focus of this programme is the development of design and implementation skills relating to the development of Computer Games, requiring students to consider games scenarios and devise solutions that meet associated requirements and constraints throughout. In year 1, the contexts presented are typically

Part 3: Learning Outcomes of the Programme	
	well-bounded and defined by the tutor in charge of the module in which it occurs. Advancing through the programme, students are required to be increasingly self-directed, moving towards confidently dealing with problems with conflicting requirements, the resolution of which they need to evaluate and justify. At all levels students are required to synthesise (3) the knowledge and skills required in a range of modules to determine new ways of working. The extent of this progressively increases from level 1, where significant guidance is offered in the process; to year 3, where students are expected to more independently draw on all of their current and previous learning to undertake extensive individual and group projects. At level 1 Analysis (2), Evaluation (4) and Problem Solving (5) are developed through solving small-scale problems across a number of modules. The focus is on conceptual understanding of a problem, its practical solution largely free from the complications and constraints of large-scale projects within commercial environments. This allows students to initially address these issues without the need for in depth examinations of alternative strategies or having to balance conflicting goals. At level 2 there is a move away from small-scale problems to the design of larger systems. With this comes the need to evaluate (4) alternative methods and designs and to balance potentially conflicting objectives (7). Level 3 sees the move to specific application examples, many demonstrating the types of problems and pressures that may be faced in a commercial/professional environment. With these, the ability to appreciate problems contexts (6) and practice creative and interpretive thinking (8) is developed, as well as the skills to strike an appropriate balance when facing conflicting requirements and objectives (7).
	Assessment:

Part 3: Learning Outcomes of the Programme	
	Games development requires demonstration of all of the intellectual skills. At level 1 the focus in coursework assessment, undertaken in a number of modules, is on the skills of Analysis (2), Evaluation (4) and Problem Solving (5). At levels 2 and 3 the emphasis grows to include all the remaining skills. Many of the coursework assessments and exam papers include elements of programming work. Independent reading is used to enable students to focus on their own areas of interest and in the process assess skills 1-4 in the submitted reports, essays and exam answers. Design-work, even when not implemented in a programming language, requires demonstration of skills 1,2,5,6,7 and a number of coursework assessments, project reports and exam questions are devoted to such work.
C Subject, Professional and Practical Skills	
C Subject, Professional and Practical Skills Students will be able to: 1. Create high and low-level game designs corresponding to stated requirements. 2. Interpret game designs to form technical requirements and design code/software that meets them. 3. Write high and low-level games code that fulfill a given design. 4. Utilise professional standard tools and practices throughout the development process, to design, compile, debug, test, profile/optimise, package and quality assure their products. 5. Have a working knowledge of the fundamental mathematics underpinning the development of computer games.	Teaching/learning methods and strategies: By and large, conceptual frameworks and principles are initially delivered through lectures, and further explored in practical, studio-based sessions, with a focus around points 1 - 5 in the list. Progressing through the award, however, there is an increased emphasis on individual and group project work, gathering pace at level 2 and culminating with a range of project modules in the final year, introducing and enhancing, respectively, the skills in points 6 - 8. The delivery of 9 runs throughout the award, with an initial focus on small-scale principles developing into large-scale practices in the final year. Assessment: The possession of these skills is demonstrated by the development of practical pieces of coursework (game and software design, and implementation) and by examination.

Part 3: Learning Outcomes of the Programme	
6. Apply a range of techniques from key areas to games development, including: • artificial intelligence • physics and simulations • graphics • memory management • multiprocessor and network programming • etc. 7. (Re)use existing components and frameworks to build new applications. 8. Critically and comparatively evaluate games and their designs. 9. Employ a range of tools and notations to support the activities listed above, including: • Software design packages • Programming languages (such as C++, C#, etc), • Integrated Development Environments (IDEs), compilers, debuggers, profiling/optimisation tools • RAD, level design and asset creation software and associated scripting languages • Audio-visual production tools • Project management and source control software etc.	Skills 1, 2, 3, and 9 form an integral part of many assessments throughout the award, and are assessed, in some form, across most modules on the award. Particular emphasis on skills 1, 2 and 8 is provided in optional module Gameplay Programming at level two, for students looking to develop this as a specialism. Skill 5 is assessed mainly through modules Games in C++ and Advanced Technologies, though additional exposure is expected through the Creative Technology Project and Commercial Games Development. Level two optional module Game Engine Programming offers particularly in depth coverage of this skill for students looking to develop this as a specialism. While students are exposed to concepts encompassed by points 4, 6, 7 and 8 throughout the programme, these outcomes are mainly assessed through modules at levels 2 and 3, including Game Engine Architecture, More Games in C++, Game Level Design, Commercial Games Development, Advanced Technologies and the Creative Technology Project and the option modules AI for Creative Technologies, Mobile Applications and Audio Visual Studio. Optional module Game Engine Programming at level two offers in depth coverage of skill 6 for students looking to develop these areas as specialisms.
D Transferable Skills and other attributes	
D Transferable Skills and other attributes Students will be able to: 1. Communication skills: communicate orally or in writing. 2. Self-management skills: manage one's own time; meet deadlines and work with others. 3. IT skills in context: use software tools in the context of application development.	Teaching/learning methods and strategies: 1. Communication skills are developed through a variety of methods and strategies including the following: • Students maintain laboratory logbooks and individual and group development diaries. • Students participate in tutor facilitated discussions. • Students participate in group projects, requiring verbal and written communication with team members, the latter

Part 3: Learning Outcomes of the Programme	
4. Logical reasoning skills: undertake analysis and interpretation of information in the context of Creative Technology and Computer Science. 5. Problem formulation: express problems in appropriate notations. 6. Progression to independent learning: gain experience of, and to develop skills in, learning independently of structured class work. For example, developing the ability to use on-line facilities to further self-study. 7. Comprehension of professional literature: read and to use literature sources appropriate to the discipline to support learning activities.	typically through a forum or otherwise electronically provisioned group working tools. • Students produce written reports on coursework and projects. • Students participate in group and individual presentations. 2. Self-management skills are developed through a variety of methods and strategies including the following: • Students conduct self-managed practical work. • Students participate in practically-oriented tutorial/laboratory/studio sessions. • Students practice games and software design, and programming. • Students self-manage individual and group projects under academic supervision. 3. IT skills in context are developed through a variety of methods and strategies including the following: • Students make use of online teaching materials and discussion forums. • Students use a range of software development tools, methods and packages, alongside other discipline specific software. • Students are encouraged to undertake their own games development projects, outside of taught sessions. • Students partake in electronically facilitated group projects, using industry-grade project management and source control tools. 4. Logical reasoning skills are developed through a variety of methods and strategies including the following:

Part 3: Learning Outcomes of the Programme	
	• Students develop problem-solving algorithms and programs. • Case studies are used to explore design and implementation issues with students. • Students practice design and programming. • Students design and develop components of large systems. • Students make use of extensive existing software libraries to produce technical demonstrations, tools and games for a range of target platforms. • Students analyse apparently conflicting requirements and devise suitable solutions. 5. Problem formulation skills are developed through a variety of methods and strategies including the following: • Students develop problem-solving algorithms and programs. • Students decompose game development scenarios into appropriate games and technical design components. • Students practice games and software design and programming. 6. Progression to independent learning is developed through a variety of methods and strategies including the following: • Students are encouraged to research relevant topics. • Students are encouraged to use the library, journals and trade literature, the internet and other online facilities to discover information and broaden their knowledge. • Students are encouraged to articulate and reflect upon their own ideas and experiences. • Students negotiate the content and structures of their individual and group projects and

Part 3: Learning Outcomes of the Programme	
	portfolios with academic staff and individual tutors. • Students are encouraged to develop their own technical demos, tools, and games outside of taught sessions. 7. Comprehension of professional literature is developed through a variety of methods and strategies including the following: • Students are introduced to key texts, and encouraged to utilise other relevant discipline specific literature available online and through the library. • Material is recommended to the students in module syllabi and by tutors. • Students are required to research and refer to appropriate literature in assignments as well as individual and group projects.
	Assessment: Communication skills are assessed by a mix of examination, coursework, essays, presentations and group and individual project reports. Other skills are assessed through a number of similar instruments including the following: Individual and group projects Practical assignments Portfolios of exercises In addition, self-management skills are assessed by both peers and tutors through a range of additional activity such as personal tutor sessions throughout the course.

Part 4: Programme Structure

This structure diagram demonstrates the student journey from Entry through to Graduation for a **full time student**, including:level and credit requirements;interim award requirements;module diet, including compulsory and optional modules

ENTRY ↓	Year 1	Compulsory Modules Games Development Evolution (UFCFF5-30-1) Games in C++ (UFCFWA-30-1) Principles of 3D Environments (UFCFY4-30-1) Games Technology: 101 (UFCFJL-30-1)	Optional Modules None	Interim Awards **Certificate of Higher Education in Games Technology** 120 credits, of which not less than 100 are at Level 1 or above
	Year 2	Compulsory Modules More Games in C++ (UFCFXG-30-2) Play & Games (UFCFC6-30-2) Game Engine Architecture (UFCFAM-15-2) Game Level Design (UFCF8M-15-2)	Optional Modules Game Engine Programming (UFCF9M-30-2) Or Gameplay Programming (UFCF7M-30-2)	Interim Awards **Diploma of Higher Education in Games Technology** 240 credits, of which not less than 100 are at Level 2 or above and a further 120 are at Level 1 or above.
	Year Out: Students on the Sandwich route complete a placement year. For students completing the industrial placement, must take 15 credits from: UFCFE6-15-3 Professional Experience (studied during placement year), UFCFWJ-15-3 International Experience (studied during placement year) UFCFVJ-15-3 Professional Development. If placement students take Creative Technology Project (UFCFS4-30-3) they have a choice of either UFCF7H-15- 3 Mobile Applications or the new 15 credit AVP module or UFCFSN-15-3 AI for CT. Placement Students taking UFCFHQ-45-3 do not have this option.			

	Year 3	Compulsory Modules Commercial Games Development (UFCFM4-30-3) Advanced Technologies (UFCFW3-30-3)	Optional Modules All students must take a module from either: UFCFS4-30-3 Creative Technologies Project OR UFCFHQ-45-3 Comprehensive Creative Technologies Project Placement students *must take 15 credits from:* UFCFE6-15-3 Professional Experience (studied during placement year) UFCFWJ-15 3 International Experience (studied during placement year) UFCFVJ-15-3 Professional Development If placement students take Creative Technology Project (UFCFS4-30-3) they have a choice of either: • UFCF7H-15-3 Mobile Applications • UFCFMR-15-3 Audio Visual Studio • UFCFSN-15-3 Artificial Intelligence for Creative Technologies Placement Students taking UFCFHQ-45-3 do not have this option. NON-PLACEMENT STUDENTS: If non-placement students take Creative Technology Project (UFCFS4-30-3) they should take 30 credits from: • UFCF7H-15-3 Mobile Applications • UFCFMR-15-3 Audio Visual Studio • UFCFSN-15-3 Artificial Intelligence for Creative Technologies If non-placement students take Comprehensive Creative Technologies Project (UFCFHQ-45-3) they have a choice of either: • UFCF7H-15-3 Mobile Applications	Interim Awards **BSc Games Technology** 300 credits with at least 60 credits at level 3, plus a further 100 credits at level 2 or above and a further 120 credits at level 1 or above Highest Award **BSc(Hons) Games Technology** 360 credits, of which at least 100 must be at Level 3 or above, at least a further 100 at Level 2 or above and a further 140 at Level 1 or above.

			• UFCFMR-15-3 Audio Visual Studio • UFCFSN-15-3 Artificial Intelligence for Creative Technologies When going through option choices, we would advise that students take the 45 credit project and then they have a choice of option module.	

GRADUATION

Part 5: Entry Requirements
The University's Standard Entry Requirements apply

Part 6: Assessment
Approved to University Regulations and Procedures

Part 7: Student Learning
Teaching, learning and assessment strategies to enable learning outcomes to be achieved and demonstrated At UWE, Bristol there is a policy for a minimum average requirement of 12 hours/week contact time over the course of the full undergraduate programme. This contact time encompasses a range of face:face activities as described below. In addition a range of other learning activities will be embedded within the programme which, together with the contact time, will enable learning outcomes to be achieved and demonstrated. On the BSc(Hons) Games Technology programme teaching is a mix of scheduled, independent and placement learning. **Scheduled learning** includes lectures, seminars, tutorials, project supervision, demonstration, practical classes; external visits. Scheduled sessions may vary slightly depending on the module choices made. **Independent learning** includes hours engaged with essential reading, case study preparation, assignment preparation and completion etc. These sessions constitute an average time per level as indicated in the table below. Scheduled sessions may vary slightly depending on the module choices made. **Placement learning**: may include a practice placement. The time devoted to this is indicated below.

Part 7: Student Learning

Description of Distinctive Features and Support

Class-based Activities Classes use a range of activities. The particular mode of delivery of a module is determined by its Module Leader, and typically involves a combination of one or more lectures, practical sessions, group activities and individual or group project work. Modules on the programme that require laboratory classes are commonly delivered by means of a combination of lecture and practicals/tutorials.

Academic Support Academic advice and support is the responsibility of the staff delivering the module in question. Staff can be contacted outside of normal timetabled hours, either by appointment or during published "surgery" hours, in order to offer advice and guidance on matters relating to the material being taught and on its assessment.

On-line Academic Support Extensive on-line support for this programme is provided through the University portal (myUWE). This provides access to the University's e-library, which allows students to read academic journals and study-skills material. Of particular interest to students of this programme is access to the ACM, IEEE and British Standards Online databases. The portal also gives entry to UWE's Virtual Learning Environment (Blackboard) which is used by academics to make available general information about the module delivery, handbooks, lecture notes and other materials. In addition, the portal publishes individual student timetables, marks and other aspects of the operation of the programme and University life.

Pastoral Support Pastoral care is provided through the University-wide Student Advisers, a team of staff who provide comprehensive, full-time student support service on a drop-in basis or by appointment. Advisers are trained to provide advice on matters commonly of concern, including regulatory and other matters; the Adviser will, when necessary, advise the student to seek advice to from other professional services including the University's Student Services Department or from members of academic staff.

Independent Study
Many modules require students to carry out independent study, such as research for projects and coursework assignments, and a full range of facilities are available to help students with these. The philosophy is accordingly to offer students both guided support and opportunities for independent study. Guided support, mainly in the form of timetabled sessions, takes the form of lectures, tutorials, seminars and practical laboratory sessions. Students are expected to attend all sessions on their timetable, and this is especially important because of the high content of practical work in the programme.

The development of independent study will also be assisted by the nature of the support offered in individual modules. Typically, module leaders will provide a plan for the module indicating the activities to be carried out and the forms of learning to be undertaken during the delivery of the module, with a view to encouraging students to plan ahead and to take responsibility for managing their time and resources.

Computing Facilities The Faculty offers a specialised computing facility along side the general University provisions. There are multiple computing laboratories of 20 plus seats all running Linux based systems required for this program. The specialist laboratories are augmented with software resources and hardware equipment necessary for the delivery of the modules. One of the most popular areas within the Faculty is the Open Access laboratory. This area is never timetabled and gives students the opportunity to access machines at all times during opening hours. This is a mixed environment consisting of PCs and Unix workstations.

Part 8: Reference Points and Benchmarks

Description of ***how*** the following reference points and benchmarks have been used in the design of the programme:

QAA subject benchmark statements
University strategies and policies
Staff research projects
Employer interaction and feedback

In designing this programme, the faculty has drawn upon the following external reference points:

1. The QAA Framework for Higher Education Qualifications in England, Wales and Northern Ireland
2. The QAA Benchmark Statement for Computing
3. The SkillSet Undergraduate Course Accreditation Guidelines for Computer Games - Technical Path
4. UWE's Learning & Teaching Strategy

The QAA Framework for Higher Education Qualifications in England, Wales and Northern Ireland describes the attributes and skills expected of Honours graduates. The learning outcomes of this programme are fully consistent with the qualification descriptor in the Framework, and hence graduates will be able to demonstrate that they meet the expectations of the Framework.

The QAA Subject Benchmark Statement for Computing (2000, amended 2007)
The QAA Subject Benchmark Statement for Computing is applicable to this proposal. The proposal falls clearly within the scope of the Computing benchmark, in that it is precisely concerned with "the understanding, design and exploitation of computation and computer technology" (Benchmark Statement, p.1 section 1). The Games Technology curriculum falls within the cognate area identified in the document and draws from the topics listed at Annex A of the document. In terms of the Statement's high-level characterisation of Computing, the programme has at its heart ***practice*** and ***software*** with its **application-oriented** approach focused on the development of Games. Nevertheless, ***theory*** and ***hardware*** are important and significant strands.

Great attention has been paid in the design of this programme to create a teaching and learning programme which will foster a good and effective mix of the cognitive, practical and generic (transferable) skills discussed in 3.2 of the Benchmark Statement. The programme matches well with the course design principles listed in 4.1 of the Statement.

UWE's Learning & Teaching Strategy has informed the faculty's policy for the delivery of its programmes, whose main features are described in section 7.

FOR OFFICE USE ONLY

First Approval Date	January 2013			
Revision Approval Date		Version	1	
			1.1	
			1.2	
			2	
	6 November 2017		3	Link to RIA (ID 4506)
	Jan 2020		4	