



# MCAST

Malta College of Arts, Science & Technology

MQF Level 6

IT6-A3-19

**MCAST Bachelor of Science (Honours) in Multimedia  
Software Development**

**Course Specification**

## **Course Description**

This course is intended for those students who are keen on developing interactive and media rich software applications such as games and mobile applications. In this course you will learn how to develop software applications, create the required multimedia content and integrate this content with the created application. This course covers 2D and 3D Graphic Design, Computer Animation, Sound Design, Visual Effects, Object Oriented Programming, Mobile Application Development and other similar modules. This course is intended for individuals who have an artistic inclination and are also good in programming.

Within this degree students will be required to choose one minor stream of 5 units (amounting to 30 ECTS) from the following 4 options: Advanced Software Applications; Organizational Decision Making; Engineering Manufacture; Creative Design.

## **Programme Learning Outcomes**

At the end of the programme the learner will be able to:

1. Design, create and document the multimedia content needed in software applications.
2. Design, implement and document the game and mobile applications for a given requirement.
3. Integrate and synchronize the multimedia content with application events.
4. Test and secure the software application and its content to conform to industry standards.

## **Entry Requirements**

MCAST Advanced Diploma in IT  
(recommended stream: “Multimedia Software Development”)

or

MCAST Advanced Diploma in Electronics (Computer Engineering)

or

2 A-Level passes and 2 I-Level passes

Compulsory A-Level: Computing

Compulsory A-Level or I-Level: one subject from Pure Mathematics, Applied Mathematics and Physics

## Current Approved Programme Structure

<b>Unit Title</b>	<b>ECTS</b>
Object Oriented Programming	6
Software Engineering	6
UX Design	6
Client Side Scripting	6
Mobile Applications Development	6
Data Structures & Algorithms	6
Sound Design	6
Applied Maths	6
Interactive Mobile Development	6
2D Graphics	6
Soft Computing for Games	6
Securing Applications	6
Database Programming I	6
Test Driven Development	6
Visual Effects	6
3D Graphics	6
Game Level Design	6
Project	6
Image Processing and Computer Vision	6
Mobile Game Development	6
Connected Gaming	6
Sound Engineering	6
Programming for the Cloud	6
IT Project	6
Content Management Systems	6
English	6
Entrepreneurship	6
Critical Thinking	6
Dissertation	12
<b>Total ECTS</b>	<b>180</b>

## Object Oriented Programming

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

This unit assumes students are already familiar with OOP principles and concepts, and is as such designed for learners who have completed the level 4 OOP module or who have equivalent experience.

This unit covers advanced object-oriented concepts, allowing the students to design object-oriented solutions from start to finish. Students will learn how to create applications using a markup-driven GUI framework and how to link created applications to a database using object-relational mapping (ORM). It is up to the lecturer to choose the specific technologies to be used in this unit. For example, one can teach markup-driven GUI with WPF/XAML, ANT, GTK or QT, as well as any other markup-driven GUI language. ORM can be delivered using LINQ, Hibernate, TopLink or any other object-relational language.

The unit starts with a revision of object-oriented concepts to ensure that all learners are on track with their knowledge. This revision includes access modifiers, properties, constructors, inheritance, overriding virtual methods, abstract classes, interfaces and polymorphism.

Following this, advanced object-oriented concepts will be covered including aggregation vs inheritance, dependency injection, open-closed principle, single responsibility principle, Liskov substitution principle, interface segregation principle and dependency inversion principle.

Learners will then develop applications using a markup-driven GUI framework. The markup language (such as XML/XAML or similar markup) will be introduced, as well as layout and display control. Learners will learn how to create and style controls as well as how to use WPF/ANT/GTK/QT (or other) user controls. The GUIs created will then be made functional with a focus on event handling and data binding.

Finally, the unit covers persistence. In this part of the unit, learners will integrate with and store data in a database using an object-relational language such as LINQ, Hibernate, or any other ORM language. Students will appreciate the differences between a data-centric approach and an object-centric approach when designing the model layer of an application.

## Learning Outcomes

On completion of this unit the student will be able to

1. *Design and build object-oriented solutions using both fundamental and advanced object-oriented concepts to be able to address business requirements.*
2. *Implement persistence in created applications to allow created applications to store and read data from multi-user database management systems.*
3. *Design applications that leverage object-oriented design principles to ensure best practices are adhered to.*

MCAST

## Software Engineering

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

This unit has been designed to introduce learners to the main concepts behind the science of software engineering. Throughout the course of their studies, students will acquire the skills to understand and support the complete life cycle of a software system - from inception, requirements elicitation and design, through the various stages until release and maintenance. Students will gain an understanding of different software development techniques and will learn how to critically select which technique is best suited to the development of different systems.

The unit places focus on some of the more recent software development processes, making particular emphasis on the Agile philosophy of software development. Students will understand the agile process and its constituent components, its applicability to modern software development and the various actors involved in the process together with their roles and responsibilities. Another core component of this module will be that of introducing students to the Unified Modelling Language, UML, as a tool to facilitate and speed up the software development process. The various constructs of this modelling language will be covered, together with explanations of how they can be utilised to specify and document the software and business processes.

This unit will also present students with a range of advanced software engineering concepts and approaches which will give them the skills required to be able to support new and evolving developments. Students will be introduced to a number of different software architectures and design approaches and they will be encouraged to analyse which setups are most adequate as solutions for diverse scenarios.

## Learning Outcomes

On completion of this unit the student will be able to

1. *Plan and tackle a small software design project as part of a team using an Agile approach.*
2. *Perform a requirements acquisition exercise in order to identify the main functional and non-functional requirements of a proposed software system.*
3. *Identify and construct the most applicable UML modelling diagrams to use in particular phases in a software system's development process to achieve a specified goal.*
4. *Design a solution to a problem by proposing the most suitable architecture and utilising known design patterns.*

MCAST

## UX Design

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

This is a skills based and practical unit, which will allow learners to demonstrate they have the necessary skills to be able to create a satisfying user experience. Learners will develop a deeper understanding of UX design which incorporate utility, ease of use, and efficiency by determining what people want to experience from websites, apps and games. Learners will learn to analyse popular markets, how users search for information and how to structure their content to take advantage of this.

The Unit is relevant to learners wishing to use graphics to help rather than distract visitors, balance advertising and content in a way to benefit the product owner, and integrate any necessary media to explain all information without overloading the user. On completion of the Unit learners will understand how to understand the progressive navigation and format the information exchange provided on a page to guide users into the ideal user experience intended. This Unit will provide the Learner with the ability to understand user's problems through research and insight, be able to communicate their designs through deliverables by using personas, sitemaps, user flows and wireframes. Learners will also be thought how to monetize and generate money from an ideal user experience by concentrating on the user flow and timing of their experience provided.

Learners will carry out formatting and presentation features to prepare their documentation for publication or sharing with other users which will prepare them to adapt to the industry. This will therefore require learners to be confident in carrying out more advanced interactions breaching out of their comfort zone and deeper knowledge of a user's psychology, while applying different types of fundamental features in their UX designs.

Finally, learners should have the underpinning knowledge and understanding to how to present, defend and critique design decisions.

## Learning Outcomes

On completion of this unit the student will be able to

1. *Recognise the different types of games and recognize a Game Designer's role in the gaming industry.*
2. *Analyse the importance of understanding the target audience and the different designs applied to each.*
3. *Identify and plan game play using core mechanics present in games to adapt to players by understanding how to achieve the fun element in a game.*
4. *Communicate and illustrate a game idea in full depth to be able to generate a game design document.*

MCAST

## Unit: ITSFT-506-1611 Client Side Scripting

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

This unit covers advanced client side scripting. In particular, an emphasis will be placed on using a design pattern that allows the developer to separate the business logic of a web application from the UI design. This unit also covers some aspects of HTML5.

The unit starts with a brief revision of client side scripting, whereby the learner is shown how HTML, CSS and JavaScript can be used together to build the presentation layer of a web application. This includes basic tasks covered in pure JavaScript.

Following this, learners will be introduced to JavaScript libraries, which simplify client-side scripting by offering a functional layer above pure JavaScript. Learners will learn how to bind to DOM elements via the JavaScript library and hence how to manipulate DOM objects. They will also learn how to listen for changes in the DOM and react accordingly with JavaScript functions. Learners will also be introduced to asynchronous JavaScript, and how this can improve the performance of a web application.

Using these libraries and JavaScript, the learners are then shown how to create and consume data, and also how to implement object-oriented concepts in JavaScript.

At this point, the learners are introduced to an MV\* architecture. Using this methodology, learners will learn to separate business logic from presentation. The learners will present data using this architecture, including how data can be presented in this model. Learners will also be exposed to HTML5 (and to a lesser degree, CSS3). They will learn about the impetus for a new version of HTML and the new tags and functionality that it brings.

Learners will have practical sessions where they will apply HTML5 elements to web applications and manipulate them via JavaScript libraries.

For this unit, the lecturer is free to use any JavaScript library of their choice.

However, importance should be given to the most popular, ‘defacto’ JavaScript libraries. The lecturer is also free to choose a development environment.

## Learning Outcomes

On completion of this unit the student will be able to

1. *Build the presentation and controller layers of a web application using HTML, CSS and JavaScript to meet system requirements.*
2. *Use JavaScript libraries in web applications to simplify client side scripting and build interactive elements for the system being developed.*
3. *Describe a design pattern and implement it to separate the concerns of business logic and presentation in web applications.*
4. *Explain the motivation for the development of a new version of HTML5 and use the new elements introduced in web applications.*

MCAST

## Mobile Applications Development

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

Mobile phones have evolved significantly in technology and application within the short period of their existence. From the original purpose of providing voice calls for travelling office executives these devices have now evolved into handheld, multi-media, smart devices boasting an array of sensors, multi-core processing power and a market penetration across young and old, personal and professional use. A significant factor behind the success of smart phone devices is the ability for the user to personalise with well designed, productive mobile applications. This unit is designed to introduce the learner to key aspects in mobile application development from Operating System and development environment through to application state and data storage.

This is a skills based unit and will allow learners to demonstrate they have the necessary skills to be able to design, program and test a mobile application. The unit will guide learners through the basics of Operating Systems (OS), development environments, device sensors and data storage. Learners will use software programming concepts and as a consequence should be able to operate effectively at more than a basic level of competence before commencing this unit.

Outcome 1 introduces the mobile Operating System of choice as well as the development tools available. The learner will familiarise themselves with the development environment and test device/simulator.

Outcome 2 concentrates on implementing the application (Graphical) User Interface (UI). Learners will explore the fundamentals behind building a GUI for the particular OS. They will investigate the various libraries and classes as well as designing and using OS menus.

Outcome 3 focuses on working with differing user states. Learners will understand how to handle application start up, background and resuming as well as considering state changes associated with screen rotation or notifications.

Outcome 4 looks at various options for storing/loading context data internally or externally (through the use of memory cards).

## Learning Outcomes

**On completion of this unit the student will be able to**

1. *Examine the fundamental components of mobile application development (OS architecture, development environment).*
2. *Develop a practical mobile application user interface.*
3. *Manipulate a mobile application depending upon state change.*
4. *Demonstrate the ability to store and load data in a mobile application.*

MCAST

## Data Structures & Algorithms

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

The basis of solving a problem requires an understanding of how to break it down into a series of much more manageable small parts. In order to do this, students need to be able to assess the complexity of a problem. Once the algorithm has been broken down into smaller sections, a student should start to write logical instructions using pseudocode. Each instruction in turn will manipulate data, which may be for instance array structures, which are similar to vectors in mathematics or abstract data types such as pointers, these are similar to how machine code uses memory addresses to access data.

In this unit students will learn about writing algorithms for common problems such as Queues and thereby choosing the most appropriate data structure.

Students need to implement a series of algorithms which are well known in Computer Science. For a given algorithm a student will need to analyse the complexity and make a decision on how this may affect the efficiency of the algorithm in terms of run time. Although computers now have very powerful processors, students still need to estimate the time it would take for their algorithm to process a given amount of data. In particular, as the amount of data becomes larger the amount of time it takes to process the data can grow exponentially.

Students will learn and appreciate that algorithms can be translated into programming code. This in turn will give them an insight into solving problems on paper before typing their code into a text editor for a given programming language. This experience will allow them to see how their programs run as originally indented in the specification. Also it has been written in such a way that it runs efficiently, avoiding complexity in their solution as well as making best use of the processing power of their computer system.

Data structures such as pointers, which allow a programmer to use memory addresses to access data, give the student a much more flexible method to manipulate data. For each algorithm a student needs to select the most appropriate data structure, in

order to produce a solution which will carry out the required tasks as set out in the specification for a computer program.

## Learning Outcomes

On completion of this unit the student will be able to

1. *Construct programs using Abstract Data Types and Structured Data types.*
2. *Design efficient algorithms for commonly encountered problems using existing examples.*
3. *Make use of algorithm analysis to determine the efficiency of an algorithm.*
4. *Compare algorithms in terms of their correctness, proof and intractability.*

MCAST

## Sound Design

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

This unit is designed to provide learners with the knowledge, understanding and skills involved in sound design for the games industry. It should be learner centred and lecturers should use discussion, both individual and group, for analysis of existing sound designs from a variety of software applications platforms. The learner will also acquire the underpinning knowledge of the properties and processes of sound, and the effects on sound with different settings applied. They will plan a sound production, using a sound editor to create their sound and present it as a final product for their chosen game.

Discussion and analysis should be integrated into structured tuition in techniques specific to software/hardware being used and also specific to sound design practice in a software application/multimedia setting. Learners will develop technical skills in the creation of audio files and will apply these skills in the creation of a finalized sound design for a game as well as learning how to use the hardware and software related to the production of a sound design.

Learners will be able to experiment and gain hands on experience of audio equipment and audio applications in the creation of a sound design which will also enhance their skills in critical thinking, planning, organisation and production. They can work as part of a team or individually in all aspects of the practical development and production of a sound design for a sound design brief. If working as a team each learner could be given a specific role within the team or alternatively, roles could be rotated. Each learner's role should be meaningful to the development and production of the sound design and the individual contribution of each learner must be apparent.

This Unit is suited to learners who have an interest, and may be considering a career, in the multimedia and software applications industry as well as the information technology industry. It would also suit learners who wish to enter the audio technology industry.

## Learning Outcomes

**On completion of this unit the student will be able to**

1. *Understand the influence of sound and recognize the properties of digital audio.*
2. *Plan the acquisition of audio in accordance with a sound design brief for software applications.*
3. *Prepare a sound design project using a sound editor.*
4. *Produce a finalized audio track in accordance with a sound design brief.*

MCAST

## Applied Maths

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

This unit is designed to introduce learners to the mathematical concepts they need to solve problems that can be met while developing multimedia content and any other content involving computer graphics. The unit also gives learners a head start to some mathematical concepts related to the rapidly developing industry of online betting.

Although nowadays we have a lot of graphical tools which aid in the development of such content, it is still of fundamental importance to understand the mathematical concepts that lie beneath. This would lead to a better understanding, appreciation and usage of these tools, better adaptability to other tools, and more ability to solve unseen problems that may arise.

All the content of this unit is not only delivered, but also applied, to actual multimedia/graphics scenarios. This helps to better gauge the learners' understanding and interest, as well as realise its applicability in different scenarios.

### Learning Outcomes

**On completion of this unit the student will be able to**

1. *Understand basic mathematical concepts and apply them to a graphical context.*
2. *Understand the mathematical foundations of lighting and apply them in graphical contexts.*
3. *Understand the mathematical foundations of mechanics (such as acceleration, and collisions) and apply them in graphical contexts.*
4. *Understand the mathematical foundations of combinatronics and probability theory and their application in online betting/gambling scenarios.*

## Interactive Mobile Development

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

Mobile phones have evolved significantly in technology and application within the short period of their existence. From the original purpose of providing voice calls on the move these devices have now evolved into handheld, multi-media, smart devices boasting multi-core processing power, a wide range of sensors, and a market penetration across young and old, personal and professional use. A significant factor behind the success of smart phone devices is the ability for developers to innovate upon the device hardware and Operating System in order to provide beneficial and lucrative mobile applications (Apps). This unit is designed to afford the learner with vital skills and experience in mobile application development from the use of OS multi-tasking features and network capabilities through to sensor utilisation.

This is a skills based unit and will allow learners to demonstrate they have the necessary skills to be able to design, program and test a mobile application. The unit will guide learners through the basics of multi-tasking, Internet services, device sensors and telephony features. This unit is a continuation from Mobile Programming (Level 4) and it is expected that Learners will be familiar with this unit's content.

Outcome 1 introduces the concept of multi-tasking/threading typically found on multi-core processor enabled mobile devices. The learner will familiarise themselves with the software development requirements when working with multi-tasking Operating Systems.

Outcome 2 concentrates on the common libraries provided for delivering World Wide Web content in a mobile device use case. Learners will investigate various development libraries and classes as well as implementing a web service application.

Outcome 3 focuses on working with differing device sensors. Learners will understand how to handle sensor initialisation as well as considering state changes associated and notifications.

Outcome 4 looks at information provided by the mobile devices phone Application Programming Interface. The learner will experience the types of information provided including connection status, access to the address book and use of text messaging.

## Learning Outcomes

**On completion of this unit the student will be able to**

1. *Utilise multi-threading within a mobile Operating System.*
2. *Develop a mobile application to utilise Web API's.*
3. *Manipulate a mobile application depending upon on-board sensor readings.*
4. *Demonstrate the ability to work with a mobile telephony Application Programming Interface.*

MCAST

## 2D Graphics

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

This is a skills-based unit that will enable learners to gain a practical understanding of basic principles and techniques employed in interactive 2D graphics, and apply these principles to the development of graphics programs. Learners will be introduced to key concepts that underpin 2D graphics, relevant programming techniques will be reviewed and an appropriate development environment will be made available.

An overarching theme of this unit is to provide hands-on practical experience. In this context, learners will undertake a project-based activity in which an animated and interactive 2D game will be designed for a specified audience and implemented. This will provide valuable experience in understanding and applying basic principles of 2D digital design and graphics.

Project work will be carried out within a structured framework and so learners will have the opportunity to gain valuable knowledge and experience 2D game design, project planning, implementation and testing. This framework will facilitate key aspects of the project work by, for example, ensuring that realistic goals, techniques and project milestones are clearly defined from the outset. This approach will allow the project to proceed in an orderly way that supports completion within the allotted time period. Subsequently, learners will demonstrate project achievements and will describe and critically appraise their work by means of a formal report. Learners will work under the guidance of a project tutor/supervisor.

### Learning Outcomes

**On completion of this unit the student will be able to**

- 1. Demonstrate the use of 2D graphics design, animation, and interaction techniques.*
- 2. Formulate a project specification and plan a practical solution for an interactive graphics application, which is to be developed for a specified audience.*
- 3. Design and implement a project solution in accordance with applicable techniques, specifications and time constraints.*
- 4. Demonstrate and describe project outcomes.*

## Soft Computing for Games

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

This unit will introduce students to various soft computing and artificial intelligence techniques aimed at giving them advanced tools for developing games or multimedia. The students will be introduced to some game artificial intelligence middleware where applicable and will use the different artificial intelligence techniques learnt to develop and augment games or multimedia. The game or multimedia development will be visually communicated through storyboards.

The unit will also delve into navmesh agents, which are components in unity which do the pathfinding themselves as independent agents. Furthermore, students will have the opportunity to implement and utilise readymade artificial intelligence solutions from other projects.

This is a skills based unit and will allow students to demonstrate that they have the necessary skills to be able to use a variety of tools and techniques. The unit will guide the student through a range of soft computing and artificial intelligence options, including readymade solutions which they will be able to utilise to enhance their ability to storyboard and develop games and multimedia.

Outcome 1 concentrates on building the students working knowledge of the variety of soft computing and artificial intelligence techniques and tools. The learner will familiarise themselves with these in a classroom setting as well as during laboratory time and will be able to describe and explain the benefits they have within gaming and multimedia.

Outcome 2 focuses on the learner being able to apply the knowledge they have gained through Outcome 1 and begin to map out their ideas using storyboarding techniques. They will be able to develop their ideas using informed decisions.

Outcome 3 emphasises application of the skills and knowledge that have been developed throughout Outcome 1 and 2 in the creation of an augmented game or multimedia. The learner will be able to justify the decisions they have made and say how it has impacted on the game or multimedia.

Outcome 4 will see the learner successfully implementing available readymade artificial intelligence solutions and utilising them to support and inform their own code development skills.

## Learning Outcomes

On completion of this unit the student will be able to

1. *Demonstrate knowledge of a variety of soft computing and artificial intelligence techniques for games or multimedia.*
2. *Apply understanding of various soft computing and artificial intelligence techniques in the storyboarding of games and multimedia.*
3. *Produce an augmented game and/or create multimedia.*
4. *Implement and utilise available readymade artificial intelligence solutions.*

## Securing Applications

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

Securing applications is becoming an important concern for most development companies, mainly, due to the fact that in the last decade, many companies are utilizing applications over the network for their business, thus they are more vulnerable to a wider variety of external threats. Examples are Web applications such as an online banking system, an auction site and so on. Whilst the new capabilities of these web applications offers a variety of benefits to the business, if not developed with security in mind, they can cause vulnerabilities in the system that can expose the business to various risks, such as for example a simple configuration error might leave a door open for the hackers to access the business database.

Security should be a design requirement for a project even though there are time and cost constraints to the project. In order to develop secure applications, important security considerations should be adapted throughout the software development life cycle.

The majority of the application attacks are based on common vulnerabilities. OWASP which stands for Open Web Application Security Project publishes a list of most ten most popular programming mistakes whilst developing an application. These vulnerabilities are common by the hackers and are exploited frequently. Through this unit, students will be exposed to possible categories of vulnerabilities. Each category contains a set of threats, such as SQL Injection, Cross Site Scripting and so on.

The majority of these threats can be mitigated by applying some well-known security standards, and throughout this unit, the student will be provided with practical examples on how these vulnerabilities can be avoided.

This unit will provide students with a level 5 theoretical and practical knowledge of application security aspects.

## Learning Outcomes

On completion of this unit the student will be able to

1. *Discuss the importance of securing an application.*
2. *Apply security throughout the development life cycle.*
3. *Identify vulnerabilities and Implement practical countermeasures for the threats.*
4. *Apply a variety of cryptographic algorithms to encrypt data.*

MCAST

# Database Programming I

**Unit level (MQF):** 5

**Credits:** 6

---

## Unit Description

This unit provides the basis of advanced database theory and design principles that shall be used in future modules. Following this module, a student should be confident in concepts about relational theory and database design. The theory presented is independent of any specific database management system, however the database design conforms to agreed-upon notation easily adopted to other relational database management systems.

The unit starts with the data manipulation via data manipulation language. In this part of the unit, the learner will learn how to build databases in a relational database management system by creating tables and choosing indices. Attention is given to the enforcement of data integrity rules and their associated delete and update repercussions via foreign keys.

Following this, data query via data query language is discussed. Basic select statements are covered including selecting from multiple tables and using conditions (such as WHERE). This is then extended to include predicates and combining predicates as well as filtering character, date and time data. Finally, data is ordered and paged using appropriate techniques.

Advanced concepts then follow, including performing different types of joins based on the database content as well as using set operations. As part of this topic, different types of joins will also be discussed. At this point, data insertion, updates and deletion will be discussed using the appropriate SQL keywords.

Grouping (via GROUP BY) and windowing are also discussed in this topic, allowing for both single and multiple grouping as well as pivoting. For windowing, aggregation, ranking and offsets will be discussed.

The unit concludes with a discussion and practical use of views, as well as how inline functions can be incorporated into SQL to extract calculated fields.

## Learning Outcomes

On completion of this unit the student will be able to

1. *Create the structure of a relational database.*
2. *Retrieve data from a database.*
3. *Manipulate the data in a database.*
4. *Prepare advanced reports from data within a database.*

MCAST

## Test Driven Development

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

Computer programs are essentially a product which is supplied to a customer for a given fee. Initially a client would agree on a specification provided by analysts and once approved the client will enter into a legally binding contract to receive a software program which will carry out specific tasks, for example it would lead to an improvement in the day-to-day operations of the client's organisation. Similar to other products which we purchase in the shops, it would be essential that it works correctly and carries out the specific tasks which were promised by the manufacturer. There is same expectation for computer software to meet the similar standards. For this precise reason, it is important to thoroughly test programs using well known techniques in Computer Science. Often the reputation of the persons developing the computer software would be at risk if testing is overlooked or inadequate.

Test Driven Development is a development approach, where a test written before writing the code. This approach will make sure that the source code is tested well, this has the effect of having code which is in modules, it is also flexible so it can be changed more easily and it is extensible so it can be adapted without much further coding. The stages involved in Test Driven Development Cycle include: adding a test, running all tests and checking if the new code fails, writing code, running automated tests, re-factoring code and finally repeating the cycle as in previous steps with a new test.

The initial testing process would start with receiving messages from Compilers or Interpreters telling the programmer that there is a syntax error in a given line of code. Secondly, test data would need to be used check whether the program has any logic errors and to see that the program carries out the tasks which were stated in the specification. If the wrong output is being shown, then the programmer needs to find and correct the errors, often by using special tool known as debugger software, this is piece of software is built into the compiler.

Taking the testing process further, the computer program needs to ensure the integrity of the data is maintained by means of a verification process of checking data. Secondly, data validation is the processor of getting the computer software to check data is valid in terms of the context in which it is being used. These tests will ensure the correctness of the data being processed and will involve writing complex code. Nevertheless, if computer programs are to gain the respect of business and industry, then this important aspect should not be overlooked.

This unit is geared towards enabling students to learn about the current well known methods of testing software. These will include White Box and Black Box testing. Also students will learn about the different levels of testing which include Unit Testing (using white box testing), Integration Testing (using black box testing) and System Testing.

Businesses that produce computer software often have a dedicated team or department who take on the job of testing programs written by developers, once it is fully functioning. Alpha testing is the initial testing phase carried out by a dedicated team of testers, in order to find bugs that were not found originally through previous tests. Alpha testing is also known as acceptance testing, once the software is approved at this stage we move to Beta testing. For example, Beta testing involves the distribution of pre-release Mobile application (app) software to a select group of people so that they can test it in their own homes. The beta version of the software is as close to perfect as the company can make it.

As per a product in the phases of production, in for instance a manufacturing environment, each product has a product life cycle where it is initially designed, developed, tested and sold to customers in shops. In the same way software programs have systems life cycle. There are various models for the system life cycle available to students which can be applied in developing computer software. We need to select the model which best suits the organisation which requires the computer system.

## **Learning Outcomes**

**On completion of this unit the student will be able to**

- 1. Compose tests using Test Driven Development methodology.*
- 2. Explain the process involved in the different levels of testing.*
- 3. Use the Systems Life Cycle in relation to developing a computer system.*
- 4. Develop code which will verify and validate data utilised in the program.*

## Visual Effects

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

Creation of VFX in 3D environment for gaming material is an essential part of any audio-visual gaming production. Without the process of removing, adding and manipulating source material in all gaming and other interactive products adding appropriate effects as required.

The process of editing and effects involves making creative decisions about source material. It provides an opportunity for the creativity of the gaming stage to be continued through to the finished product. Through study of the development and principles of editing VFX effects, learners will develop an insight into the 'language' of create VFX effects and the technical conventions used by editors to communicate with audiences.

Skills will be practiced through 3D VFX material. Learners will also develop an understanding of how their work can affect the final outcome of a production.

Learners will be able to experiment with appropriate applications and use their skills in the production of their VFX products. VFX should be learnt as well as the importance that learners develop knowledge of VFX, not just the technical process but the aesthetics too. Learners will then be able to have a clear knowledge and understanding shown through practical skills and being able to apply articulation.

During this module the learner will be creating and preparing VFX through 3D software and implement it during game development.

### Learning Outcomes

**On completion of this unit the student will be able to**

1. *Examine and discuss the evolution of digital compositing and digital visual effects.*

2. *Analyse critically and discuss the implications of emerging technologies in the visual effects design process.*
3. *Design and develop a range of creative visual effects solutions to design problems.*
4. *Interpret the influence of visual effects in the creation of innovative digital media.*

MCAST

## 3D Graphics

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

3D animation is part of the art form which creates moving images by using 3D computer graphics formats. Often the platform for the animation is the computer itself, or it is another media such as photo or film. The animations are often referred to as 'computer-generated images' (CGI). Animators of 3D media are given responsibility for the correct portraying of behaviour and movement. This is often applied to bring creatures and characters to life, but on occasions animations are applied to other entities such as scenery, vegetation, objects and environmental matter. A number of Specialist software packages can and are used to create the animated features and the animators try to give portrayal of behaviours and moving in such a way that makes best use of the available technology.

The Unit will allow students to have the opportunity to use types of 3D animation software applications to produce a 3D animated piece of work. Students will develop an awareness of the basics of 3D geometry definition how to translate, scale, rotate and properly combine all transformations together through the basic principles of 3D model structure such skeletons and meshes for game characters.

Students will develop skills in preparing assets for game engines and be able to implement their 3D creation during their game development and drafting by using pre-visualisation work and storyboards. Preparation and workflow management skills will take place, develop and become habitual when preparing works as well as creating 3D resources for the use of game development and implementing the appropriate programming language for interaction.

### Learning Outcomes

**On completion of this unit the student will be able to**

1. *Identify the basic elements in the process of creating a 3D scene.*
2. *Construct 3D models using well proven techniques.*
3. *Apply appropriate techniques and methods to refine and render 3D models.*
4. *Apply 3D transformation on the 3D models.*

## Game Level Design

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

This is a skills based unit and will allow learners to demonstrate they have the necessary skills to be able to create a satisfying gaming experience by developing a deeper understanding of game design and the use of social and ethical context in which video games are developed, marketed and played. Learners will learn to analyse popular games from commonly used genres, examining from both form and functionality. The learners will apply game design concepts and principles to solve problems.

The unit is relevant to learners wishing to further develop their knowledge of games as a tool to help provide them with informative knowledge on how to apply the skills learnt in their games. On completion of the Unit learners will understand how to produce games documentation that may be more technically complex in content and analysis, as well as developing the understanding, knowledge and skills required to produce them. This unit will provide the learner with the ability to use any type of game design and analyse it to be able to use the data extracted and apply it to any type of game genre. The learner will also be able to learn the impact of the games and how they affect their players, which will tie into the learners understanding of different types of players.

Learners will carry out formatting and presentation features to prepare their documentation for publication or sharing with other users which will prepare them to adapt to the game industry. This will therefore require learners to be confident in carrying out more advanced game types and deeper knowledge of player psychology, while applying different types of fundamental features in their game designs.

Finally, learners should have the underpinning knowledge and understanding to check completed games and understand how to evaluate, understand the game mechanics behind it, solve the final outcome and also explain the core loop of the game.

## Learning Outcomes

On completion of this unit the student will be able to

1. *Recognize the different types of games and recognize a Game Designer's role in the gaming industry.*
2. *Analyse the importance of understanding the target audience and the different designs applied to each.*
3. *Manipulate and plan game play using core mechanics present in games to adapt to targeted players by understanding how to achieve the "fun" element in a game.*
4. *Communicate and illustrate a game idea in full depth to be able to generate a game design document.*

MCAST

## Project

**Unit level (MQF):** 5

**Credits:** 6

---

### Unit Description

The rise and evolution of network technology and the internet has made significant changes and improvements to the world around us. Individuals, businesses and countries have benefited greatly through global dissemination of information, growth of electronic commerce, and business innovation through collaboration.

Recent technology trends and the rapid and significant developments in mobility and cloud computing have all played a major role in promoting and enhancing growth in global economies. Whilst this growth and the benefits derived from it have been welcomed they have compounded to a growing shortage of highly skilled employees with the necessary skillsets to work within the industry.

The primary aim of the MCAST Higher Diploma in Network Design and Implementation is to provide participants with a comprehensive range of knowledge and practical skills in designing, building and maintaining computer network infrastructures.

The structure of the award and the inclusion of units particularly in the areas of Virtualisation, cloud computing, network security and intrusion prevention will ensure the award is well placed to address the growing demand for the aforementioned skillsets.

The purpose of this unit is to allow students to demonstrate the practical and technical skills they have acquired whilst studying on the diploma.

The unit should help students to consolidate their learning of network design, implementation and maintenance and should help students considering a career in IT with a particular emphasis on networks gain successful employment or progression to studies at a higher level. Students should be encouraged to choose a project in an area of their own specific interest derived from the course. The chosen project should however allow the students to demonstrate their competence in the field of network design and should reflect the other units they have learnt on the course namely, installation and configuration of computer platforms, management of databases and Content Management Systems, Virtualisation/Cloud computing and Wider Area Networks, and network security.

## Learning Outcomes

**On completion of this unit the student will be able to**

1. *Analyse a networking project assignment brief and develop a solution to meet a given specification.*
2. *Plan and organize a technical solution that meets the given specification.*
3. *Implement the chosen solution in phases through to project completion.*
4. *Reflect and evaluate the success of the implemented solution against the given specification.*

MCAST

# Image Processing and Computer Vision

**Unit level (MQF):** 6

**Credits:** 6

---

## Unit Description

This unit is designed to give the students a basic yet solid understanding of Image Processing techniques, as well as the application of such techniques in the rapidly-developing area of Computer Vision. The latter has been known to be a difficult problem, due to the complexities involved in the human vision system. Nonetheless, it is finding its way through many sectors such as safety and security, health, and entertainment. Moreover, methods of acquiring such data (digital images and videos) have become even more available and affordable.

The unit first introduces the theory behind Image Processing and Computer Vision, and moves on to the application of fundamental operations in Image Processing, such as quantisation and removal of noise. These fundamental operations are then used in more complex problems, such as finding edges, corners and other interest points or shapes in the image.

Finally, some typical problems of Computer Vision will be examined. These include object detection, recognition and tracking. These problems will serve as challenges as well as motivation to students, as they highlight the relevance and practicality of the material covered. As much as possible the unit will adopt a learn-by-doing approach, in that for most of the topics the underlying theory will be explained followed by a practical example or implementation. This will ensure comprehension and engagement of the student.

## Learning Outcomes

**On completion of this unit the student will be able to**

1. *Understand the basic concepts and relevance of Image Processing and Computer Vision.*
2. *Apply fundamental techniques in Image Processing.*
3. *Apply techniques to identify shapes and features in images.*
4. *Demonstrate Object Detection, Recognition and Tracking Techniques.*

## Mobile Game Development

**Unit level (MQF):** 6

**Credits:** 6

---

### Unit Description

This unit assumes students are already familiar with game design and development concepts completed in level 5 Game Level Design module. It is both a theoretical and a skills based unit and is designed to allow the students to understand and apply the necessary skills to create games for mobile devices.

This unit presents the theory and practice of mobile game development for mobile devices using the Unity game engine. This is a practical unit and the focus is to produce high quality production games.

Initially students will learn how to setup a mobile developer environment for Unity and how to use 'Unity Remote' app for testing. Following this setup, the unit exposes learners to mobile scripting and covers functionalities found in the Input and Handheld classes such as multi-touch, accelerometer, device vibration, location services and others.

Usability for gaming is an important issue and therefore students will learn how to build responsive GUI and optimised 2D textures in relation to game usability.

Social app development serves an important role for mobile games. Students will learn how to develop social media authentication and use 'share and send' dialogs. For example, the 'Facebook SDK for Unity' could be used to integrate mobile games with Facebook. Students will also learn how to develop web services integrated with Unity to store persistent player data.

Finally, students will learn how to manipulate the mobile player settings, create app icons, build the game for mobile and publish their own game on the app store.

### Learning Outcomes

On completion of this unit the student will be able to

1. *Plan and design usability for mobile games.*
2. *Plan and develop gameplay for a mobile game.*
3. *Manipulate a mobile game application based on device sensors.*

4. *Plan and integrate social networking in a mobile game.*
5. *Show the ability to prepare a mobile game for an online app store.*

MCAST

## Connected Gaming

**Unit level (MQF):** 6

**Credits:** 6

---

### Unit Description

In this unit, students will be applying their knowledge of programming to develop complex computer games which allow multiple users online to interact and communicate with each other in a structured and interesting way. Connected games allow for players connecting to each other over the internet or through social media. This unit will guide learners through the process of creating their own connected game, and building software which allows for decentralized networking to share information both for the multiplayer gaming aspect, as well as for data mining requirements. Some examples of connected games include games built to be published on online social platforms, as well as massively multiplayer online games which allow for players to interact on a network game server. Students will need to cater for the issues that can affect gameplay with respect to synchronization as well as communications between players in a lag-free way.

### Learning Outcomes

On completion of this unit the student will be able to

1. *Understand and implement a variety of infrastructures for connected games.*
2. *Build a connected game infrastructure.*
3. *Evaluate a connected game infrastructure.*
4. *Implement a connected game with a working data mining API.*

## Sound Engineering

**Unit level (MQF):** 6

**Credits:** 6

---

### Unit Description

This unit assumes that the students are already familiar with both sound design and game development.

In this course the students will learn concepts that are key in assisting a sound engineer during a sound project. Learners will embrace the concept of dynamic music and apply such techniques in their sound projects. This may involve the incorporation of several (smaller) sound projects that are blended together with sound effects to define the goals of a game. Such goals will be established by the use of a Digital Application Workstation (DAW), whereby it is assumed that learners already have prior knowledge about the basic aspects. Although learners would already have previous experience with a Digital Application Workstation (DAW), the unit still provides flexibility so that it is not linked to a specific DAW. For example, one may use Ableton Live or Prosonus Studio One in the creation and modification of sound projects.

The unit will start with a revision of Musical Instrument Digital Interface (MIDI)/Audio channels and effects in a DAW to ensure that all learners are on track with the required knowledge for this course.

A study of the frequency spectrum will follow where a number of frequency ranges are identified. Learners will understand the importance of each individual frequency range and its application in relation to a particular sound. Comprehension of gain amplification and reduction will enable learners to increase or decrease the amplitude of the particular waveform in that particular frequency range. Such knowledge can be applied with the use of Equalisation. Learners can therefore acquire know-how by applying Equalisation on sound.

Following topics include dynamic range compression and expansion and the use of plug-ins in a DAW. Compressors and Expanders in a DAW help empower the richness of an overall sound mix by defining certain sounds in the mix. The use of plug-ins in a DAW such as Virtual Studio Technologies (VSTs) enables learners to obtain a more realistic sound on a particular instrument.

Learner will apply the above tools via MIDI automation to achieve sound dynamics. The application of basic sound dynamics can be achieved by the use of MIDI automation.

Finally, the learner needs to identify and script the different stages in a game. An appropriate sound mini-project needs to adhere with the relevant game stage. For example, in a role-playing game if the player discovered a shortcut, higher-tempo matching music will reflect. Thus, an insight into interactive composition will allow learners to comprehend the various sound elements involved in dynamic music.

## Learning Outcomes

**On completion of this unit the student will be able to**

1. *Demonstrate an understanding of audio and frequency response making use of equalisation in order to obtain a quality sounding overall mix.*
2. *Produce and configure audio, MIDI and auxiliary channels.*
3. *Implement sound improvements with an improved dynamic range that result in a superior audio quality.*
4. *Design and apply proper sound dynamics in a game.*

## Programming for the Cloud

**Unit level (MQF):** 6

**Credits:** 6

---

### Unit Description

This subject will add on to the concept of cloud computing. The unit will introduce the motivating factors, benefits, challenges and the service models i.e., software-as-a-service, platform-as-a-service and infrastructure-as-a-service. Moreover, the unit will provide the learner hands-on experience with the commonly found tools within the cloud infrastructure industry such as storage technologies, security measures, highly elastic scalability in delivery of enterprise applications and software as a service (SaaS), caching techniques, and different hosting options.

Practical sessions will be the basis for this unit where a number of technologies will be explored, compared, analysed and then selected to be used within a much larger project to make use of the discussed advantages they will bring about in today's applications.

Moreover, the students taking this unit will be provided with a cloud account where they will undertake the task to configure the necessary settings to make use of such mentioned technologies. In the end their work should be deployed on this cloud account.

### Learning Outcomes

On completion of this unit the student will be able to

1. *Describe the main concepts and benefits behind Cloud Computing.*
2. *Use Cloud Storage solutions to store structured and large data.*
3. *Use other services provided by the Cloud services provider.*
4. *Use Cloud services available to host and consume web applications, APIs or other services.*

## IT Project

**Unit level (MQF):** 6

**Credits:** 6

---

### Unit Description

In this unit, students will be studying the research methods which may be applied to the results and discussions section in their thesis. They will be guided in the development of conclusions and inferring patterns from their results, and will learn how best to interpret and explain data gathered as part of their research. This module is intended to help students working on their final year projects and the practical assessments will be based in part on their project work.

### Learning Outcomes

On completion of this unit the student will be able to

1. *Define research scope.*
2. *Identify the correct approach for a given research problem.*
3. *Explain different research methodologies.*
4. *Present information and conclusions effectively.*

## Content Management Systems

**Unit level (MQF):** 6

**Credits:** 6

---

### Unit Description

When learning how to program students typically write short programs all by themselves. This contrasts with the reality of most workplaces which require employees to work in teams on larger projects. A content management system supports a number of users working in an environment that promotes synergies for people to achieve their goals. Enterprise-level software is more complex and typically requires a future-proof architecture that keeps maintainability and extendibility manageable.

The unit starts by exploring the anatomy of a Content Management system; a system that has stood the test of time. Furthermore, it considers how its design maintains the delicate balance between simplifying extendibility and keeping conflicts and bugs in check.

Next, learners can explore how this architecture can be extended through the creation of themes, plugins, and REST API applications.

Themes allow the programmer to modify the look and feel of the software. While it focuses on design, other development considerations such as responsiveness, and support for different layouts come into play.

Plugins, on the other hand, allow the developer to extend the functionality of the application via self-contained modules that can easily be activated or deactivated by a website administrator.

Finally, REST API application development allows one to write and integrate external applications that can have their own independent architecture and that can be written using an entirely different programming language. Although the unit is software intensive, it also covers administration topics such as authentication and authorization, configuration, and module management.

## Learning Outcomes

On completion of this unit the student will be able to

1. *Identify the features and functionality of an extendible content management system.*
2. *Activate and customise a versatile and responsive theme that supports multiple layouts*
3. *Install and set up several plugins including ones that incorporate settings and short-codes.*
4. *Integrate an external application capable of communicating with the content management system through a REST API.*

MCAST