

IM21 Program Electronic Games To Develop Functionality			
Example job titles: Game Programmer			
Overview	Knowledge and Understanding	Awareness	Performance Statements
<p>This unit is about your ability to use a programming language to develop the product's functionality. Note that this is typically an iterative process involving constant feedback and revision, so patience and a willingness to be objective about your work are vital.</p> <p>Some industry-standard languages you might use are:</p> <ul style="list-style-type: none"> ▪ C++ ▪ C# <p>Some valuable foundation languages that individuals might be familiar with are:</p> <ul style="list-style-type: none"> ▪ Java ▪ Pascal ▪ Smalltalk 	<p><i>This is what you must know</i></p> <ol style="list-style-type: none"> a. How to interpret, follow and write specifications or other briefs; b. How, and to whom, to ask questions to clarify requirements or raise issues in response to the specification or brief; c. Relevant programming principles and best practice (such as object-oriented and/or procedural programming); d. The functions and syntax of the programming language you are using; e. How to write efficient code that is also readable and maintainable; f. Why it is important to clearly and professionally document your code so that others can read and understand it; g. How to test and debug your code efficiently, effectively and thoroughly; h. How and when to write custom debugging code; i. How to use appropriate software for asset management, code version control and bug tracking; <p>In addition you may need to specialise in one of the following areas:</p> <ol style="list-style-type: none"> j. Physics programming; k. Graphics programming; l. Network programming; m. Artificial Intelligence programming; n. User interface programming; o. Audio programming; p. Logic/mathematical skills. 	<p><i>This is what you must be aware of</i></p> <ol style="list-style-type: none"> i. Project parameters and constraints including target platforms and their capabilities; ii. The overall structure of the game being developed; iii. Sources of information for help, tips and tricks for making most effective use of the programming language and development environment you are using; iv. The implications of later requests for changes to functionality or other aspects of the product; v. The technical and logistical issues surrounding development for cross-platform delivery; vi. The expectations of others who may be involved in debugging, using or modifying your code; vii. Who wrote the specification, who designed the relevant part of the game, and who is responsible for decision making; viii. An understanding of the nature of a code-base, which is subject to constant revision and ongoing development; ix. The commercial value of software code. 	<p><i>This is what you must be able to do</i></p> <ol style="list-style-type: none"> 1. Use the specified development environment or coding tool effectively; 2. Code programs or program components to provide specified functionality; 3. Produce modular code; 4. Clearly document and comment your code so that others can understand it; 5. Create efficient code that is easy to read and maintain; 6. Liaise with colleagues to ensure designs and specifications are correctly implemented; 7. Respond positively to requests for changes to work schedules, timescales and product features; 8. Use version control and asset management systems to ensure full back-up of your work.