



Australian Government

Department of Education, Employment and Workplace Relations

ICAGAM508A Develop complex 3-D software for games and interactive media

Release: 1

ICAGAM508A Develop complex 3-D software for games and interactive media

Modification History

Release	Comments
Release 1	This Unit first released with <i>ICA11 Information and Communications Technology Training Package version 1.0</i>

Unit Descriptor

This unit describes the performance outcomes, skills and knowledge required to develop complex 3-D software for games or interactive media.

Application of the Unit

This unit applies to games programmers working in the game development industry.

Licensing/Regulatory Information

No licensing, legislative, regulatory or certification requirements apply to this unit at the time of endorsement but users should confirm requirements with the relevant federal, state or territory authority.

Pre-Requisites

Not applicable.

Employability Skills Information

This unit contains employability skills.

Elements and Performance Criteria Pre-Content

Element	Performance Criteria
<i>Elements describe the essential outcomes of a unit of competency.</i>	<i>Performance criteria describe the performance needed to demonstrate achievement of the element. Where bold italicised text is used, further information is detailed in the required skills and knowledge section and the range statement. Assessment of performance is to be consistent with the evidence guide.</i>

Elements and Performance Criteria

1. Build complex 3-D applications using multiple frameworks provided or engines	1.1 Employ integrated development environment facilities to include existing 3-D, audio and physics libraries suitable for games or interactive media production 1.2 Configure a 3-D environment compatible with a specified platform by using existing library facilities and appropriate language 1.3 Instantiate virtual objects in a complex 3-D environment 1.4 Import complex pre-constructed models retrieved from persistent storage into a 3-D environment using scripts or library routines 1.5 Use code to animate 3-D objects required by game play 1.6 Use code to handle collisions between objects in a 3-D environment 1.7 Use code to manipulate the texturing and other attributes of models during run-time execution 1.8 Incorporate environmental elements to enhance user experience 1.9 Select and use appropriate exception handling techniques to ensure program stability in a complex 3-D environment that uses multiple libraries
2. Create a complex graphical user interface (GUI) for a 3-D environment	2.1 Employ integrated development environment facilities to include existing 3-D compatible GUI controls suitable for complex games or interactive media production 2.2 Combine predefined GUI elements to create a complex interface for a 3-D environment 2.3 Modify scripts or code to customise existing GUI elements for a complex interface 2.4 Write code that processes events raised by a complex GUI in a 3-D environment 2.5 Use GUI events to modify the configuration of a complex 3-D environment
3. Debug a complex 3-D application	3.1 Use stand-alone debugging tools or tools provided by an integrated development environment to examine variables and trace running code across multiple libraries 3.2 Employ debugging facilities, such as log windows or files to detect logical and coding errors
4. Use documentation	4.1 Identify and choose suitable integrated or third-party

tools	documentation tools 4.2 Create and maintain code documentation for a complex 3-D project using selected integrated or third-party tools
-------	---

Required Skills and Knowledge

This section describes the skills and knowledge required for this unit.

Required skills

- analytical skills to interpret documentation and images to inform implementation of game specifications
- communication skills to:
 - check and confirm brief requirements
 - communicate clearly using speech and text
 - communicate technical requirements related to software development, graphics requirements and code development to supervisors and other team members
 - give constructive feedback
- literacy and numeracy skills to read briefs, game documentation, scripts, storyboards, scenarios, images, and technical and conceptual information
- planning and organisational skills to:
 - refer decisions to a higher project authority for review and endorsement
 - balance talent, experience and budget
 - delegate tasks and responsibility appropriately
 - establish clear roles and goals to achieve required game development outcomes
 - meet project deadlines
- problem-solving skills to recognise and address quality issues and problems
- teamwork skills to:
 - contribute to and work in a collaborative team
 - realise a unified game-play vision
- technical skills to:
 - implement programming solutions for specified problems
 - resolve basic hardware, software and other technical issues associated with game.

Required knowledge

- complex 3-D application development
- debugging techniques
- documentation techniques
- object-oriented 3-D programming concepts
- object-oriented 3-D programming language
- techniques for using a GUI to interact with a user
- techniques for using multiple games-oriented libraries.

Evidence Guide

The evidence guide provides advice on assessment and must be read in conjunction with the performance criteria, required skills and knowledge, range statement and the Assessment Guidelines for the Training Package.

Overview of assessment	
Critical aspects for assessment and evidence required to demonstrate competency in this unit	Evidence of the ability to: • employ multiple 3-D frameworks or libraries • build an appropriate GUI to develop complex 3-D applications with documentation generated by appropriate tools.
Context of and specific resources for assessment	Assessment must ensure access to: • suitable 3-D equipment, software and hardware • technical manuals • appropriate learning and assessment support when required • modified equipment for people with special needs.
Method of assessment	A range of assessment methods should be used to assess practical skills and knowledge. The following examples are appropriate for this unit: • evaluation of work samples or simulated workplace activities • observation of game production activities • verbal questioning or interview concerning aspects of game development, including: • capability of game engines and software tools to meet the requirements of the brief • evaluating game prototypes from technical, design and game-play perspectives • game testing and trialling procedures • maintaining integrity of the design brief and game design document • risk assessment and critical path planning • translating design and technical specifications into working game prototypes.
Guidance information for assessment	Holistic assessment with other units relevant to the industry sector, workplace and job role is recommended, where appropriate. Assessment processes and techniques must be culturally appropriate, and suitable to the communication skill level,

	language, literacy and numeracy capacity of the candidate and the work being performed. Indigenous people and other people from a non-English speaking background may need additional support. In cases where practical assessment is used it should be combined with targeted questioning to assess required knowledge.
--	---

Range Statement

The range statement relates to the unit of competency as a whole. It allows for different work environments and situations that may affect performance. Bold italicised wording, if used in the performance criteria, is detailed below. Essential operating conditions that may be present with training and assessment (depending on the work situation, needs of the candidate, accessibility of the item, and local industry and regional contexts) may also be included.

3-D, audio and physics libraries may include:	• bullet • direct sound • directX • FMOD • Irrlicht • newton • OGRE • openGL.
Configure a 3-D environment must include:	• device selection • game resolution • screen colour depth • output performance such as: • anti-aliasing • level of detail • filtering • caustics and refraction.
Language may include:	• C#.net • C++ • Java • Small Talk • VB.NET.
Virtual objects must include:	• audio managers • cameras • lights • physics managers • viewports.
Pre-constructed models refers to:	• 3-D mesh data containing predefined animation sequences that are retrieved from persistent storage using script or code • developed in-house • purchased third party.
Animate must include:	• animation selection/playback for pre-constructed models • based on user input • driven by code

	• dynamic camera movement as required by game play.
Objects may include:	• cameras • lights • pre-constructed meshes.
Environmental elements may include:	• audio • render filters • simulated physics • tactile feedback.
Integrated development environment may include:	• Code Warrior • Code::Blocks • Eclipse • J-Edit • Visual C++ • Visual Studio suite.
Documentation may include:	• architecture documentation • code comments • design documents • in-code documentation • internal module documentation • release documents • requirement documents • test documents • user manuals.

Unit Sector(s)

Game development