



Australian Government

Department of Education, Employment and Workplace Relations

ICAGAM507A Develop intermediate 3-D software for games and interactive media

Release: 1

ICAGAM507A Develop intermediate 3-D software for games and interactive media

Modification History

Release	Comments
Release 1	This Unit first released with <i>ICAI1 Information and Communications Technology Training Package version 1.0</i>

Unit Descriptor

This unit describes the performance outcomes, skills and knowledge required to develop intermediate 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

<p>1. Build a 3-D application using a provided framework or engine</p>	<p>1.1 Employ <i>integrated development environment</i> facilities to include existing 3-D libraries suitable for games or interactive media production</p> <p>1.2 Use existing library facilities and appropriate <i>language</i> to facilitate <i>configuration of a 3-D environment</i> compatible with a specified platform</p> <p>1.3 Instantiate <i>virtual objects</i> in a simple 3-D environment</p> <p>1.4 Create <i>mesh primitives</i> using 3-D library routines</p> <p>1.5 Generate code to manipulate 3-D objects, including cameras, lights and mesh primitives</p> <p>1.6 Import pre-constructed meshes from persistent storage into a 3-D environment using scripts or library routines</p> <p>1.7 Apply class inheritance to modify or extend existing 3-D class</p> <p>1.8 Select and apply exception handling techniques to ensure program stability in a simple 3-D environment</p>
<p>2. Create a graphical user interface (GUI) for a 3-D environment</p>	<p>2.1 Employ integrated development environment facilities to include existing 3-D compatible GUI controls suitable for games or interactive media production</p> <p>2.2 Combine predefined <i>GUI elements</i> to create a simple interface for a 3-D environment</p> <p>2.3 Modify scripts or code to customise existing GUI elements</p> <p>2.4 Write code that processes events raised by a GUI in a 3-D environment</p> <p>2.5 Create GUI events to modify the configuration of a simple 3-D environment</p>
<p>3. Debug a 3-D application</p>	<p>3.1 Use stand-alone debugging tools or tools provided by an integrated development environment to examine variables and trace running code</p> <p>3.2 Use debugging facilities, such as log windows or files, to detect logical and coding errors</p>
<p>4. Deploy documentation tools</p>	<p>4.1 Investigate and select integrated or third-party <i>documentation</i> tools</p> <p>4.2 Deploy integrated or third-party tools to create and maintain code documentation</p>

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:
 - appropriately 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
- resolve basic hardware, software and other technical issues associated with games
- teamwork skills to:
 - contribute to and work in a collaborative team
 - realise a unified game-play vision
- technical skills to:
 - build a GUI to interact with a user
 - use correct file formats and archiving procedures
 - use 3-D libraries or frameworks.

Required knowledge

- 3-D application development
- debugging techniques
- documentation techniques
- object-oriented 3-D programming concepts and language.

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	<p>Evidence of the ability to:</p> <ul style="list-style-type: none"> • create at least one 3-D framework or library that includes an appropriate GUI • create documentation generated by appropriate tools.
Context of and specific resources for assessment	<p>Assessment must ensure access to:</p> <ul style="list-style-type: none"> • suitable 3-D IT equipment hardware, software and other technical material, such as manuals • 3-D software applications • appropriate learning and assessment support when required • modified equipment for people with special needs.
Method of assessment	<p>A range of assessment methods should be used to assess practical skills and knowledge. The following examples are appropriate for this unit:</p> <ul style="list-style-type: none"> • evaluation of work samples or simulated workplace activities • observation of game production activities • verbal questioning or interview concerning aspects of game development, including: <ul style="list-style-type: none"> • 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	<p>Holistic assessment with other units relevant to the industry sector, workplace and job role is recommended, where appropriate.</p> <p>Assessment processes and techniques must be culturally</p>

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

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.

<i>Integrated development environment</i> may include:	<ul style="list-style-type: none"> • Code Warrior • Code::Blocks • Eclipse • J-Edit • Visual C++ • Visual Studio suite.
<i>Language</i> may include:	<ul style="list-style-type: none"> • C#.net • C++ • Java • Small Talk • VB.NET.
<i>Configuration of a 3-D environment</i> must include:	<ul style="list-style-type: none"> • device selection • game resolution • screen colour depth • output performance, such as: <ul style="list-style-type: none"> • anti-aliasing • level of detail • filtering • caustics and refraction.
<i>Virtual objects</i> must include:	<ul style="list-style-type: none"> • cameras • lights • viewports.
<i>Mesh primitives</i> may include code generated:	<ul style="list-style-type: none"> • cubes • cylinders • other vertex-based constructs • spheres.
<i>GUI elements</i> may include:	<ul style="list-style-type: none"> • buttons • checkboxes • item lists • option buttons • overlay panels • text input fields.
<i>Documentation</i> may	<ul style="list-style-type: none"> • architecture documentation

include:	<ul style="list-style-type: none">• code comments• design documents• in-code documentation• internal module documentation• release documents• requirement documents• test documents• user manuals.
----------	---

Unit Sector(s)

Game development