



Australian Government

Department of Education, Employment and Workplace Relations

ICAGAM530A Develop and implement physics in a 3-D digital game

Release: 1

ICAGAM530A Develop and implement physics in a 3-D digital game

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 integrate a physics and rendering engine into a 3-D digital game.

Application of the Unit

This unit applies to game engine developers, gameplay programmers and other personnel 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. Develop a game world system	1.1 Develop code using a programming language and create instances of the rendering engine , the scene manager and physics engine's world objects 1.2 Constrain the frame rate to a desired frame rate
2. Develop components to support game world creation using physics	2.1 Create and implement game physics system 2.2 Implement game object primitives 2.3 Use or develop a physics debugger to show collision frames
3. Develop an understanding of constraint dynamics	3.1 Generate game objects or physics bodies constrained by joints 3.2 Build a ragdoll using the game physics system
4. Develop an understanding of physics interactivity	4.1 Use the game physics system to ray cast the scene 4.2 Implement a player controlled model 4.3 Implement and develop game physics system models
5. Develop an interactive scene	5.1 Use primitives to build a scene 5.2 Use trigger volumes to fire off events 5.3 Capture collision events with call-backs 5.4 Use a toggle for collision frames
6. Compile a report	6.1 Build class diagrams for all objects 6.2 Describe techniques used to create interactivity 6.3 Describe the objects and when they were used 6.4 Document the libraries used

Required Skills and Knowledge

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

Required skills

- analytical skills to:
 - interpret briefs, documents and conceptual information
 - analyse design requirements for game physics systems
- communication skills to:
 - check and confirm design requirements
 - collect, interpret and communicate in visual and written forms effectively for various audiences, including engineers and artists
 - communicate complex designs in a structured format drawn from industry standards, styles and techniques
 - communicate technical requirements related to software development, graphics requirements and code development to supervisors and other team members
 - provide practical advice, support and feedback to colleagues and management
 - translate design requirements into specifications
- initiative and enterprise skills to exercise a high level of creative ingenuity in creating a simulated physics environment with innovation
- literacy and numeracy skills to:
 - develop technical design documents
 - read briefs, work instructions, and technical and conceptual information
 - write instructions for the normal and competent operation and testing of all features and permutations
 - understand basic physics
 - understand game mathematics
- management skills to manage teams in order to effectively extract useful feedback
- 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
 - organise equipment and resources to achieve required outcomes
 - organise own time to meet milestones
- problem-solving skills to recognise and address potential quality issues and problems at design development stage
- research skills to undertake practical, technical and desktop research into advanced effects
- self-management skills to complete assigned tasks
- teamwork skills to:
 - realise a unified game-play vision
 - contribute to and work in a collaborative team
- technology skills to:
 - use correct file formats and archiving procedures

- resolve basic hardware, software and other technical issues associated with 3-D game environments.

Required knowledge

- basic programming techniques
- object-orientated language skills
- computer game development, including specific terminology
- current game-play hardware and software products
- environmental impact and sustainability considerations
- human resources required in the process of creating a game and their respective skills and technology requirements
- OHS requirements for:
 - ergonomics
 - electrical safety
 - materials handling
 - physical hazards, including lifting
- risk and critical path management
- technical constraints that hardware imposes on software development, graphics requirements, code development and creative visual design
- techniques for applying concept development skills
- techniques for applying concept visualisation skills.

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: • develop an interactive game that implements game physics • identify, evaluate and use a physics library.
Context of and specific resources for assessment	Assessment must ensure access to: • computer terminal • integrated development environment • word-processing application • the internet • physics libraries • computer hardware, software, games engines and file storage • copyright and intellectual property legislation • OHS legislation and enterprise policy • 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: • review of working executable that implements game physics • review of work samples or simulated workplace activities • review of fault-finding exercises • evaluation of reports and logbooks.
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.

<i>Develop code</i> may involve use of:	• code libraries • code objects • control loop • operating system code • programming language.
<i>Programming language</i> may include:	• ActionScript • C • C# • C++ • Flash Lite • Java • Object Pascal • Objective-C • Pascal • Python • VB.NET.
<i>Render engine</i> may include:	• game engines, such as: • BigWorld • Blender3D • Dunia • Half Life (Source) • Jade • Quake • Riot • Scimitar • Second Life • Unreal • Unity • OGRE • Irrlicht.
<i>Physics engine</i> may include:	• Ageia PhysX • Bullet • Havok physics

	• Newton game dynamics • open dynamics engine (ODE).
Desired frame rate may include:	• 120 fps • 30 fps • 60 fps.
Game physics system may include:	• call-backs, such as: • on collision effect game play • collision • combat: • bullets • debris • explosions • water • wreckage • gravity • movement: • attacking • dying • jumping • movement • swimming.
Primitives may include:	• 3-D models • animated 3-D models or characters • characters • compound shapes • convex hulls • serialisation.
Joints may include:	• ball and socket • corkscrew • hinge • slides • universal.
Game physics system models may include:	• player control, such as: • attacking • dying • jumping • movement • swimming • spline movement • wandering AI body.

Unit Sector(s)

Game development