

2021 Subject & Assessment Guide

Game Development Basics

ICT30120

Certificate III in Information Technology

Game Development Foundations

Table of Contents

Game Development Basics	3
Units of Competency	3
Overall Learning Outcomes.....	3
Subject Description	3
Industry Relevance.....	3
Assumed Knowledge.....	3
Learning Components Guide	4
Assessment Criteria	4
Assessment Description.....	4
Software.....	7
Core.....	7
Appendix 1	9
Assessment Item 1 – Recommended Brief 1: <i>Tanks!</i>	9
Assessment Item 1 – Recommended Brief 2: <i>Ocean Rescue</i>	11

Game Development Basics

Units of Competency

The units of competency that are covered in this subject are as follows:

[ICTICT313](#) - Identify IP, ethics and privacy policies in ICT environments

[ICTPRG430](#) - Apply introductory object-oriented language skills

Assessment processes and competency evidence requirements are described in the *Assessment Criteria* section below. If you have prior or other evidence against competency you should discuss this with your trainer.

Subject Overview

Overall Learning Outcomes

- Apply knowledge of object-oriented design to write game scripts in the C# programming language.
- Demonstrate acquired skills in programming games using the Unity3D game engine.
- Demonstrate the application of current legislation and industry best practice with regards to IP, copyright, and ethics, when performing game development tasks.

Subject Description

This subject is designed to introduce the basic use of the Unity3D game engine and consolidate your knowledge of the C# programming language to create real time interactive applications.

Throughout this subject you will be using Unity3D to implement a simple game, following the lead of your trainer, to gain skills in using Unity3D.

Industry Relevance

Unity3D has become a popular engine for game development and the creation of interactive media. The skills you will learn in this subject can be applied to various industries that use C# and Unity3D including the games, advertising and simulation industries.

Assumed Knowledge

- Knowledge of computer use
- Knowledge of video games

Learning Components Guide

Your learning in this subject will be achieved through the following components. The study hours may vary.

Learning Component	Duration	Description
Classroom Activity	30 hours	Presentations, group work and tutorials
Individual Skills/Knowledge Development	5 hours	Self-paced practice exercises exploring Unity3D and basic skills in C#
Project Work	10 hours	Self-paced work developing a video game project, periodically checking-in with your trainer to show progress

Assessment Criteria

Assessment Description

Assessment Milestones

Please refer to your Class Schedule for actual dates on your campus

General Description

Throughout this subject, you will be learning the basics of game programming in Unity3D through the creation of a game under the instruction of your trainer. There are two game briefs for you to choose from. These are listed in *Appendix 1*.

You will complete a series of tutorials that will guide you through the completion of the game. Each tutorial will cover a specific programming concept, technique or aspects of how to use the Unity3D game engine. This includes using and navigating around the interface and its components as well as the basic functionality of the engine.

Your final game need not be an accurate reflection of these tutorials – you are able to customize the final project in any way you desire provided the assessment outcomes have been met. However, for programmers with minimal experience it is recommended not to deviate to far from the provided briefs.

To supplement the game project assessment task, you must also complete the Knowledge Questions (listed on <https://aie.instructure.com/>). These questions cover programming topics, as well as assessing knowledge in IP, copyright, and ethics.

You will also complete the Policy Review worksheet (available on <https://aie.instructure.com/>), which will guide you through workplace policies and procedures relating to IP, copyright, and ethics. After

reviewing the policy information provided you will apply these policies when working on your game project, and make recommendations for improvement at the end of your project.

You may supply additional, or alternative, evidence of competency as agreed on by your instructor.

Evidence Specifications

This is the specific evidence you must prepare for and present by your assessment milestone to demonstrate you have competency in the above knowledge and skills. The evidence must conform to all the specific requirements listed in the table below. You may present additional, or other evidence of competency, but this should be as a result of individual negotiation with your trainer.

Your Roles and Responsibilities as a Candidate

- Understand and feel comfortable with the assessment process.
- Know what evidence you must provide to demonstrate competency.
- Take an active part in the assessment process.
- Collect all competency evidence for presentation when required.

This table defines what you need to produce as evidence of competency.

Assessment Tasks & Evidence Descriptions
<p>1. Knowledge Questions</p> <p>Evidence that includes:</p> <ul style="list-style-type: none"> • Demonstration of requisite knowledge through the completion of the Knowledge Assessment Tasks (available on https://aie.instructure.com/).
<p>2. Create Digital Game</p> <p>Evidence that includes:</p> <ul style="list-style-type: none"> • Successful creation of a videogame as specified by your instructor that includes: <ul style="list-style-type: none"> ○ Use of current industry technologies ○ Demonstration of code syntax including: <ul style="list-style-type: none"> ■ Code commenting ■ Error checks ■ Use of variables and arrays ■ Use of logic and looping constructs ■ String manipulation ○ Program reads and writes to binary files ○ Source code follows a neat and consistent standard ○ Program submitted in a format that executes outside of any IDE or engine ○ Source code and assets also submitted
<p>3. IP, Copyright and Ethics Policy Review</p>

Evidence that includes:

- Completion of the Policy Review worksheet, identifying
 - Policies and procedures relating to your work
 - Your own role in adhering to policies
 - Identification of three different types of IP
 - The risk assessment and identification process

4. IP, Copyright and Ethics Policy Compliance Evaluation

Evidence that includes:

- Completion of the Policy Review worksheet, identifying
 - Observations of any IP, ethics and privacy policy infringements
 - At least two recommendations to improve existing policies

Assessment Instructions for Candidate

METHOD OF ASSESSMENT

Assessment is a cumulative process which takes place throughout a subject. A 'competent' or 'not yet competent' decision is generally made at the end of a subject. Your assessment will be conducted by an official AIE qualified assessor. This may be someone other than your trainer. The evidence you must prepare and present is described

above in this assessment criteria document. This evidence has been mapped to the units of competency listed at the beginning of this document. Assessments will be conducted on a specific milestone recorded above in this assessment guide document.

ASSESSMENT CONDITIONS

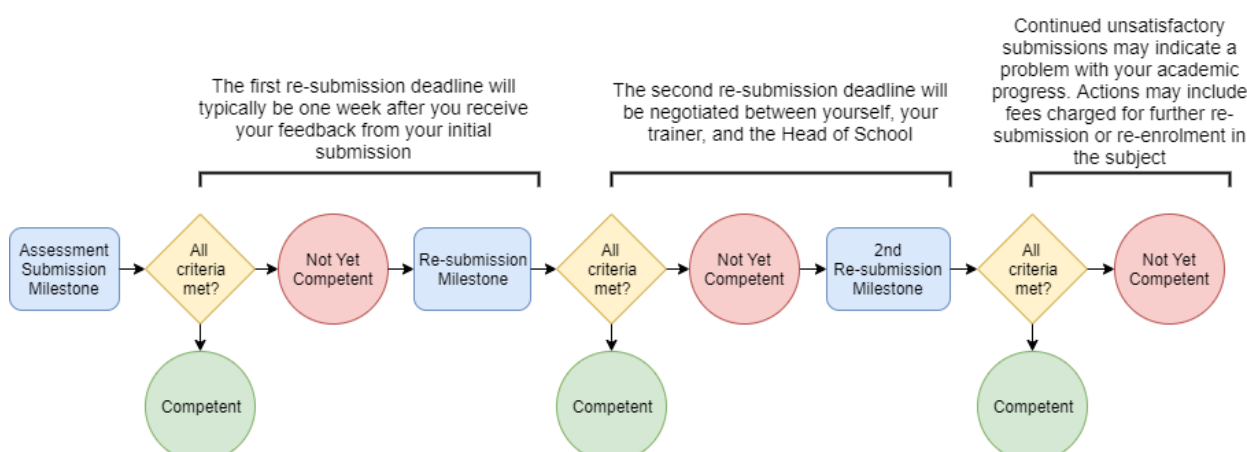
Formative assessment takes place as your trainer observes the development of your work throughout the subject and, although the assessor is likely to be aware of the evidence you are submitting, it is your responsibility to be prepared for the interview where a competency judgement is made (summative assessment). Forgetting something, or making a small mistake at the time of the milestone assessment, can be corrected. However, the assessor may choose to assess other candidates who are better prepared and return to you if time permits.

Upon completion of the assessment you will be issued with feedback and a record of the summative assessment and acknowledge that you have received the result. If you are absent for the nominated assessment milestone (without prior agreement or a sufficiently documented reason) you will be assessed as not yet competent.

GRADING

The assessment you are undertaking will be graded as either *competent* or *not yet competent*.

REASSESSMENT PROCESS



If you are assessed as being not yet competent you will receive clear, written and oral feedback on what you will need to do to achieve competence. You will be given a reassessment milestone no more than one (1) week later to prepare your evidence. If you are unsuccessful after your reassessment, you may be asked to attend a meeting with your Head of School to discuss your progress or any support you may need and further opportunities to gain competency.

REASONABLE ADJUSTMENTS

We recognise the need to make reasonable adjustments within our assessment and learning environments to meet your individual needs. If you need to speak confidentially to someone about your individual needs, please contact your trainer.

FURTHER INFORMATION

For further information about assessment and support at AIE, please refer to the assessment and course progress sections of your learner handbook.

Software

Core

Unity3D

Unity3D is a modern game engine used by many developers worldwide for developing games and interactive media. It is free to use, with paid premium options available. For this course you are able to use the free license.

- <http://unity3d.com>

Visual Studio

Microsoft's Visual Studio is the recommended Integrated Development Environment (IDE) for this subject. Other IDEs may be employed if desired as the content of this subject is designed to be cross-

platform and IDE agnostic. However, we cannot guarantee that all subject material will operate as intended on other IDEs and platforms.

- <https://visualstudio.microsoft.com/vs/community/>

Appendix 1

Assessment Item 1 – Recommended Brief 1: *Tanks!*

This brief has been designed around the specific, cumulative evidence you must prepare for and present by your assessment milestone to demonstrate you have competency in the requisite knowledge and skills for this subject. Submissions conforming to this brief will provide the specific evidence listed in the table titled *Assessment and Competency Requirements* at the end of this assessment item.

You may present additional, or other evidence of competency, but this should be as a result of individual negotiation with your trainer.

General Description

For this assessment item, you are tasked with making a single-player 3D tank shooter.

This is a single-player game where the player must destroy all enemy tanks in the game level to complete the game.

The *Tanks!* tutorials within this module will guide you through the development of this game, although you are free to customise or extend any aspect of the game, as long as you have all of the features listed in the *Requirements* section below.

Requirements

- Your game is to be implemented in the Unity 3D game engine
- It must be a single-player game with multiple (more than 1) targets (i.e., enemies)
- The player's tank can shoot to destroy all targets
- Destroying all targets will end the game (win state).
- Targets or other environmental obstacles may damage the player. The game will end with the player has no more health (lose state).
- Your code must:
 - Be commented
 - Make use of appropriate error checks
 - Use variables (at least once)
 - Use an array (at least once)
 - Use logic (if statements) and looping constructs (at least once)
- Display score and player health to the screen using User Interface components
 - Use string manipulation to convert numbers to strings and append strings
- Read and write high-score information to binary files

Submission

You will need to submit the following:

- A Release build of your application that can execute as a stand-alone program
- Your complete Unity 3D project

- Be sure to remove any temporary build folders, and the *Library* folder.
Only project files, source code files, and any resource files used should be included in your submission.

Package all files in a single compressed archive file (.zip, .7z, or .rar)

Submission Checklist

A single-player tank game in Unity 3D	
Game contains targets (enemies) that can be destroyed by the player	
Code shows use of variables, if statements, loops, and arrays	
Uses string manipulation to update and display score and/or health to the screen	
Game displays a persistent highscore table. Highscores are saved as a binary file and loaded on program launch	
Code compiles without errors or warnings	
Game executes without crashing	
A release executable has been made and included in the submission	
Project files and source code are included in the submission	
All files are packaged in a single compressed archive	

Assessment Item 1 – Recommended Brief 2: *Ocean Rescue*

This brief has been designed around the specific, cumulative evidence you must prepare for and present by your assessment milestone to demonstrate you have competency in the requisite knowledge and skills for this subject. Submissions conforming to this brief will provide the specific evidence listed in the table titled *Assessment and Competency Requirements* at the end of this assessment item.

You may present additional, or other evidence of competency, but this should be as a result of individual negotiation with your trainer.

General Description

For this assessment item, you are tasked with making a single-player 3D version of the classic game *Choplifter*, with a nautical theme.

This is a single-player game where the player must rescue all swimmers and return them to base. The player can only rescue one swimmer at a time. Unlike the classic game *Choplifter*, in this game there are no environmental obstacles or attacking enemies, although you are free to add these if you wish to add these elements.

The *Ocean Rescue* tutorials within this module will guide you through the development of this game. You are free to customise or extend any aspect of the game, as long as you have all of the features listed in the *Requirements* section below.

Requirements

- Your game is to be implemented in the Unity 3D game engine
- It must be a single-player game with multiple (more than 1) targets (swimmers)
- The player must collect each target and return it to the base (dock)
- Rescuing all targets will end the game (win state)
- The game can be a time-based challenge (collect all targets in the shortest time) with no loose state, or you can implement a loose state if you choose.
- Your code must:
 - Be commented
 - Make use of appropriate error checks
 - Use variables (at least once)
 - Use an array (at least once)
 - Use logic (if statements) and looping constructs (at least once)
- Display the game time to the screen using User Interface components
 - Use string manipulation to convert numbers to strings and append strings
 - If something other than a time-based challenge is implemented, display appropriate game information using User Interface components
- Read and write high-score information (best times) to binary files

Submission

You will need to submit the following:

- A Release build of your application that can execute as a stand-alone program
- Your complete Unity 3D project
 - Be sure to remove any temporary build folders, and the *Library* folder.
Only project files, source code files, and any resource files used should be included in your submission.

Package all files in a single compressed archive file (.zip, .7z, or .rar)

Submission Checklist

A single-player game in Unity 3D	
Game contains targets (swimmers) that can be collected by the player	
Code shows use of variables, if statements, loops, and arrays	
Uses string manipulation to update and display game time to the screen	
Game displays a persistent highscore table. Highscores (i.e., lowest times) are saved as a binary file and loaded on program launch	
Code compiles without errors or warnings	
Game executes without crashing	
A release executable has been made and included in the submission	
Project files and source code are included in the submission	
All files are packaged in a single compressed archive	