SPECIALIST EDUCATORS IN
GAMES, ANIMATION & FILM VFX

2020 Subject & Assessment Guide

Cross Platform Development

ICT50215

Diploma of Digital and Interactive Games

academy of interactive entertainment
Sydney Melbourne Canberra Adelaide Lafayette Seattle Online

Table of Contents

Cross Platform Development

Units of Competency

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

ICTGAM510 – Prepare games for different platforms and delivery modes

ICTGAM507 – Develop intermediate 3-D software for games and interactive media

ICTGAM501 – Create design concepts for digital games and 3-D media

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 teacher.

Subject Overview

Overall Learning Outcomes

- Demonstrate an understanding of the needs and constraints of cross-platform development
- Apply knowledge of installers and deploying to target platforms
- Develop projects with the Unity 3D or Unreal Engine 4 game engines
- Apply knowledge of technologies enabling cross-platform development

Subject Description

This subject is designed to give you a working knowledge of the Unity 3D or Unreal Engine 4 game engines, as well as an understanding of the process of cross-platform development and deployment. There are a lot of concerns involved in deploying games to multiple platforms, and many of them are non-obvious. While modern pre-packaged game engines can take a lot of the grunt-work out of the process, there are still many steps involved in getting a game to run on multiple platforms.

This subject guides students through the process of developing projects within either the Unity 3D or Unreal Engine 4 game engines that are able to be deployed to target platforms easily. It also guides students through the creation of installers and uninstallers needed for distributing their games to users.

Industry Relevance

Many games are released on multiple platforms. Pre-packaged game engines such as Unity 3D and Unreal Engine 4 dramatically simplify the process. However even when using larger engines, there are many hurdles and considerations with getting the same game running on multiple platforms.

This subject delivers to learners the experience of targeting and developing for multiple dissimilar platforms while solving non-trivial problems relating to user experience, optimisation, interactivity and cross-platform deployment.

The Unity 3D and Unreal Engine 4 game engines are a widely used 3D game engines. Together they have powered thousands of games and are constantly being updated to add new features.

Assumed Knowledge

- Introductory knowledge of the C# programming language
- Knowledge of game systems and game loops

Subject Textbooks

The following is a selection of textbooks we recommend for this subject:

- Hocking, J, ***Unity in Action: Multiplatform Game Development in C# with Unity 5***, 1st Edition, Manning Publications

Assessment Criteria

Assessment Description

Assessment Milestones

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

General Description

This subject teaches you how to build and deploy games to multiple platforms and how to create an installer for your game projects.

For this assessment you are required to create a game using either Unity 3D or Unreal Engine 4 and build it for multiple platforms. Which platforms you support will depend on your available devices, but your teacher will advise you as to which platforms would be best supported.

The game does not have to be a detailed game project but must at least make use of loaded assets and a basic Graphical User Interface (GUI) demonstrating industry best practice. You must also demonstrate the use of your target platform's specific input devices. For example, on a mobile device you would need to demonstrate touch-screen input whereas on a PC the project would utilize controller input or keyboard.

This assessment also includes requirements for a pre-implementation planning document, internal code documentation, and a 5-8 minute progress presentation delivered mid-way through the project. These requirements are further outlined below.

Planning Document

The specific requirements for this piece of evidence are detailed in the item *1. Plan Cross-Platform Development* in the table *Assessment Tasks and Evidence Descriptions* below.

You are to create a document that states the target deployment hardware and outlines proposed solutions for cross-platform development issues that may arise. This may include:

- User interface mock-ups specifying input or display adjustments required across platforms ensuring that the UI and HUD design is aligned with industry best practice.
- An identification and discussion of player control issues across platforms,
- Screen size and aspect ratio differences,
- API or software version requirements,
- Deployment methods,
- Platform-specific features or constraints,
- Industry standard tools, APIs, or methods for handling cross-platform development issues, and
- Identification of any environmental considerations involved with using the technologies used in the project creation, and/or with the final product.

Your planning document need not be a complete technical design document, although it does need to thoroughly identify expected issues you will face when developing and deploying your game, along with proposed strategies for dealing with these issues.

Implementation

The specific requirements for the pieces of evidence covered by the implementation are detailed in the items *2. Develop Cross-Platform Development Project*, *3. Deploy to Multiple Platforms*, and *4. Create Installer for a Project* in the table *Assessment Tasks and Evidence Descriptions* below.

You are to implement a cross-platform game using either the Unity 3D or Unreal Engine 4 game engine.

It must be possible to build each version of your game from the same game project (i.e., you will have one project that can be built for two or more different devices). How you handle any platform specific code within your codebase is up to you.

Your application must:

- Be developed for at least 1 PC platform (including Web), and 1 non-PC platform.
- Use platform-specific input devices,
- Use custom scripts, written in C# or C++, to manipulate in-game objects and handle GUI events. (You must be able to demonstrate competency in writing code, even if using visual scripting for some of your game logic),
- Include a graphical user interface (GUI) suitable for a game, and aligned with industry best practice,
- Make use of loaded assets, including 3D models,
- Use exception handling techniques.
  (For example, using a try/catch when loading data files)

A suggested project brief has been included in *Appendix 1*, although you are free to explore and develop your own game ideas for this assessment.

In addition to creating your project for multiple platforms you will need to create an installer / uninstaller for a target PC platform (Windows, OSX, Linux).

Finally, you are to use an integrated or third-party documentation tool (such as Doxygen) to create and maintain code documentation.

Progress Presentation and Feedback

The specific requirements for the pieces of evidence covered by the implementation are detailed in the items *5.Progress Presentation and Feedback* in the table *Assessment Tasks and Evidence Descriptions* below.

Approximately half way through your project you are to deliver a presentation covering the progress of your project. The exact timing of this presentation will be specified by your teacher.

Your presentation is to include a demonstration of your project (at it's current state of development), and should include the following information:

- A description of the game or application you are making
- A status update, outlining the work you have done and the work remaining
- A brief live-presentation or video showing current gameplay or application execution
- A self-evaluation of the project's usability

At the end of your presentation you are to seek and record feedback from your audience. You must demonstrate acceptance of this feedback by reflecting on it and incorporating any relevant feedback into your initial planning documentation.

Where your planning documentation has been updated as a result of feedback, you must clearly indicate this. Your planning documentation must be updated with at least one piece of feedback, even if this is not implemented in your final project submission.

Updates to your planning documentation must demonstrate:

- Acceptance and incorporation of feedback into the final design
- Assessment and refinement of the concept, and
- Reflection on possible changes to improve the visual design and capabilities of the components

Changes as a result of feedback to the planning document do not need to be implemented in the submitted project.

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 teacher.

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

1. Plan Cross-platform Development

Evidence that includes:

- Creation of a planning document containing:
    - Screen and GUI mock-ups specifying input of display adjustments made across platforms,
    - An identification and discussion of player control issues across platforms,
    - Screen size and aspect ratio differences,
    - API or software version requirements,
    - Deployment methods,
    - Platform-specific features or constraints, and
    - Industry stand tools, APIs, or methods for handling cross-platform development issues, and
    - Identification of any environmental considerations involved with using the technologies used in the project creation, and/or with the final product.
- Planning document created prior to implementation
- Access and use of appropriate information sources, with references cited in planning documentation

2. Develop Cross-platform Project

Evidence that includes:

- Creation of a cross-platform project, negotiated with your teacher, that supports multiple target platforms.
    - Your teacher may provide a project brief for you, or you may design your own

  The project that must:
    - Support multiple platforms and their specific input devices
    - Use multiple source files
    - Contain code written in either C# or C++
    - Make use of binary assets
    - Implement a Graphical User Interface (GUI)
    - Execute code in response to GUI events
    - Handle input correctly across different devices
    - Contain game world objects manipulated by code
    - Uses exception handling techniques
- All project source files submitted for review
- Code written and commented to an acceptable industry standard as agreed to by your teacher
- Internal code documentation created and maintained using an integrated or third-party documentation tool, such as Doxygen.

3. Deploy to Multiple Platforms

Evidence that includes:

- Submission of deployable builds of your Cross-platform Project for each platform that you support, packaged within compressed files
  - Allowed compressed file types are zip or 7zip
  - You must submit a compressed file for each platform supported

4. Create Installer for a Project

Evidence that includes:

- Creation of an installer for one of your Cross-platform Project builds
  - Your project must be able to be successfully installed and uninstalled from its target platform
  - A user of your installer must be able to change install options, for example, change the target install directory
  - Once installed your Cross-platform Project must be able to run as expected

5. Progress Presentation and Feedback

Evidence that includes:

- A 5 – 8 minute presentation, delivered at the project mid-point milestone, covering a demonstration of the project implementation and a self-evaluation of the project's usability
- Updating the initial planning documentation to include the following:
  - Acceptance and incorporation of feedback into the final design
  - Assessment and refinement of the concept, and
  - Reflection on possible changes to improve the visual design and capabilities of the components
- Updates and additions to initial planning documentation must be clearly marked as such

6. Application Handover

Evidence that includes:

- A game engine project, containing all source code and resource files, that compiles without errors
  - All temporary and built executable files and folders, and unused assets have been removed
- A "readme" or client document explaining how to compile, run and operate the program

All submitted material archived in a single compressed file (zip, rar, or 7z)

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 teacher. 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

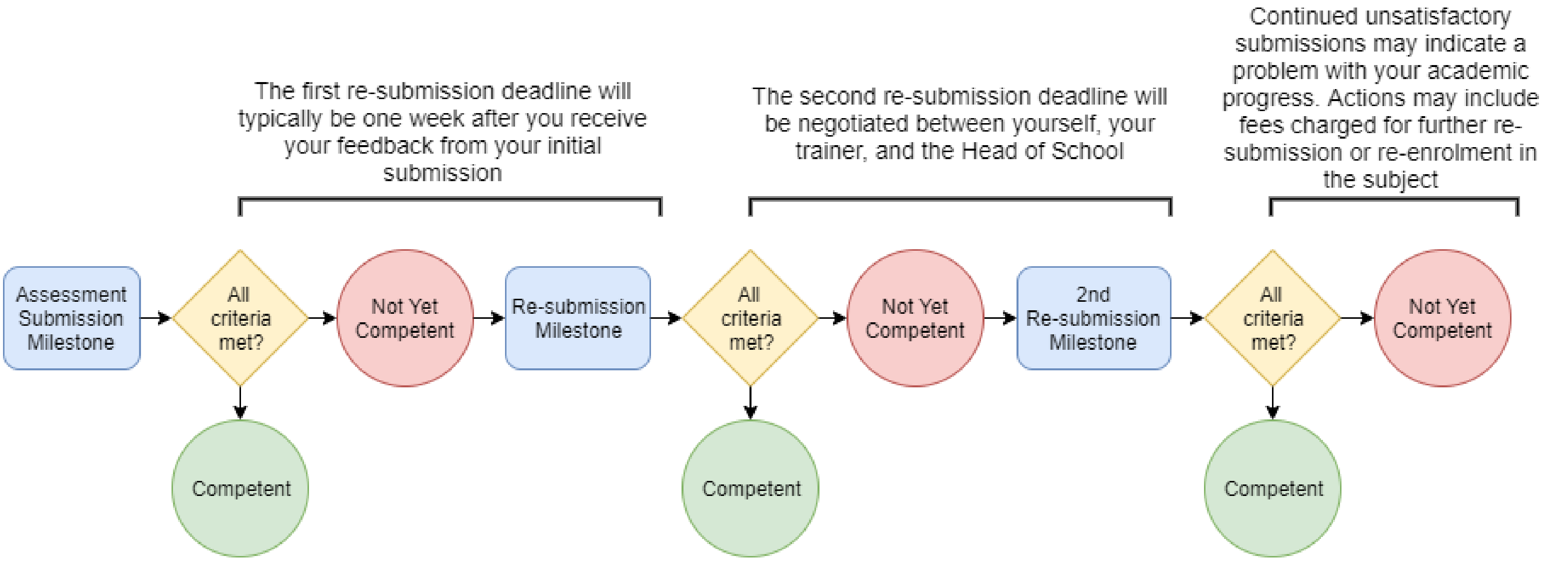
Formative assessment takes place as your teacher observes the development of your work throughout the subject and, although the assessor is likely to be aware of the evidence you are submitting, it is your responsibility to be prepared for the interview where a competency judgement is made (summative 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


The first re-submission deadline will typically be one week after you receive your feedback from your initial submission
The second re-submission deadline will be negotiated between yourself, your trainer, and the Head of School
Continued unsatisfactory submissions may indicate a problem with your academic progress. Actions may include fees charged for further re-submission or re-enrolment in the subject
Assessment Submission Milestone
All criteria met?
Not Yet Competent
Re-submission Milestone
All criteria met?
Not Yet Competent
2nd Re-submission Milestone
All criteria met?
Not Yet Competent
Competent
Competent
Competent

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. Failing to submit an assessment will result in you being assessed as not yet competent. 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 teacher.

FURTHER INFORMATION

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

Software

Core

Microsoft Visual Studio

Microsoft's Visual Studio is the recommended 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://www.visualstudio.com/

Unity3D / Unreal Engine 4

Projects need to be done in a game engine of choice. Learners can use the engine of their choice, no restrictions.

Unity3D is a widely used 3D game engine. It has powered many financially and critically successful games. It has a wide array of features that aid with development, especially for a small team. Games made with Unity can be built to a large array of devices.

- http://www.unity3d.com

Unreal Engine 4 is a complete suite of game development tools used to make games from 2D mobile games to console blockbusters and VR. Unreal 4 is a 3rd party development tool used in many game studios and offers professional development experience.

- https://www.unrealengine.com/

InnoSetup

Inno Setup is a free installer for Windows programs. First introduced in 1997, Inno Setup today rivals and even surpasses many commercial installers in feature set and stability.

- http://www.jrsoftware.org/isinfo.php

Doxygen

Doxygen is an industry standard tool for generating documentation from annotated source code, and supports popular programming languages including C++ and C#. HTML documentation is extracted directly from the source code, making it much easier to keep the documentation consistent with the source code.

- http://www.doxygen.nl/

Appendix 1

Suggested Project Brief – The Floor is Lava

Overview:

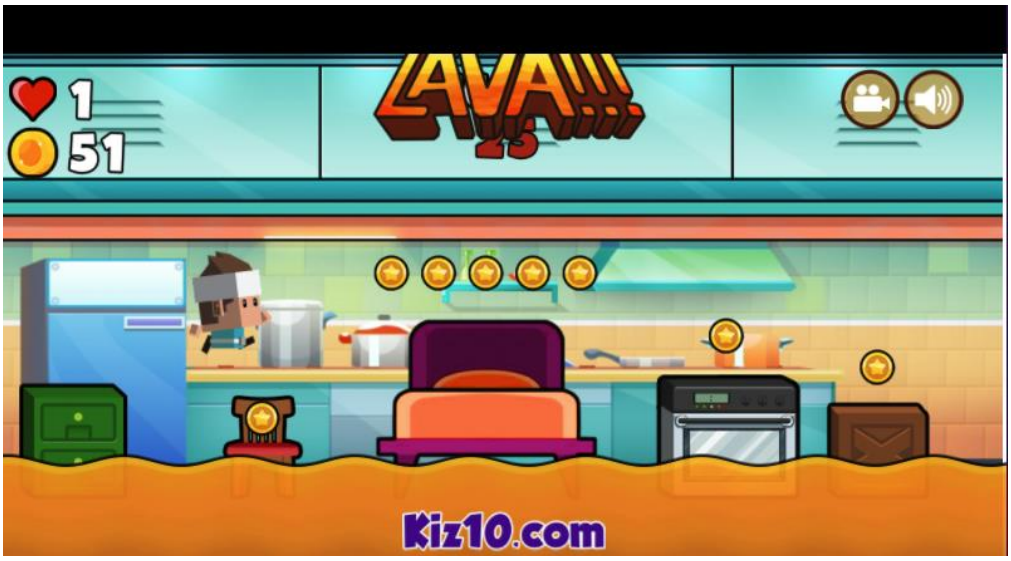
As the number of game platforms continues to grow, many studios and independent developers find it beneficial to develop for, or port their games to as many platforms as possible as a means of maximizing the return on their investment. In fact, when developing casual games, it is common for developers to intentionally design their game around the various features and limitations of all the platforms the game is intended to be released on.

Ensuring your game functions well and is fun to play on all release platforms can sometimes be challenging. Differences in processor speed, input methods, screen aspect ratio and resolution are just some of the hurdles that must be considered and planned for.

In this assignment you are tasked with developing a fun, casual game for PC and mobile.

Implementation:

The Floor is Lava! is a simple endless runner in which the player-controlled character must keep running across the screen, jumping from one piece of furniture to another while never touching the floor. The aim is to run as far as possible before touching the lava (floor). The score of each run is measured by the number of coins collected.



Play *The Floor is Lava!* online at the following URL to familiarise yourself with its mechanics: http://kiz10.com/the-floor-is-lava-online/

Alternatively, you can watch a video of the gameplay here: https://youtu.be/qJ-s__OEHMw

You are tasked with creating your own version of this game. Ensure your game has the following features:

- Your game must be made using a 3D game engine, like *Unity 3D* or *Unreal Engine 4*.
- Your game must use 3D primitives, positioned in a 3D world. If you are making a completely 2D game, use 3D planes positioned in the 3D game world and displayed with an orthographic camera. (I.e., don't create your game using only GUI components).
- Your game must include a GUI showing the player score, or other game information. Consider adding a main menu, high-scores screen, or other screens.
- Your game must run on at least 2 different platforms (i.e., PC and mobile), and must take advantage of the different features of those platforms (for example, touch input on mobile devices).
- It must be possible to build all versions of the game from the same source code/project.
- Create an installer for your PC version, and ensure your mobile build is packaged appropriately (i.e., create an installable .apk for an Android build). The PC installer must also contain an uninstaller.

In addition to the game, you also need to create some planning documentation. Create a document that:

- Outlines the design of your game;
- Identifies any issues in cross-platform development. This may include:
    - User interface mock-ups specifying input or display adjustments required across platforms,
    - An identification and discussion of player control issues across platforms,
    - Screen size and aspect ratio differences,
    - API or software version requirements,
    - Deployment methods,
    - Platform-specific features or constraints, and
    - Industry standard tools, APIs, or methods for handling cross-platform development issues.
- Lists any third-party tools or libraries used, or the sources for any artwork or animations not created yourself;

Submission

You will need to submit the following:

- A Release build of your gamer for each target platform that can execute as a stand-along program
- An installer / uninstaller for the PC build of your game
- A planning document identifying and discussing cross-platform development issues and solutions
- Play testing feedback and evaluation document
- Source code documentation, created with a third-party documentation tool (such as Doxygen)
- Your complete game project, including source code

Be sure to remove any temporary build folders (i.e., the Debug and Release folders, and Library folder) and unused assets included in your project. 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

This submission checklist is used to assist your assessor in marking your assessment.

A copy of this checklist can be downloaded from https://aie.instructure.com/ and must be submitted with your project.

General

Description	**Y/N**
All submitted projects compile without errors *Programs that don't compile cannot be assessed*	
The program includes a "readme" or document explaining how to compile, execute and operate the program	
The program performs as described in the general description	
The program contains no logical errors	
The code is sufficiently commented and clean	
An attempt has been made to increase the program's efficiency	
Code compiles without no warnings	
Program executes without crashing	
Program has no memory leaks, and closes all files after use	
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	

Estimate the number of hours taken to complete this assessment task	
How many times have you submitted this assessment task (including this time)?	

Required Features

Complete the following table by providing the class name or file name, along with the line number, to show where you have implemented each feature.

Feature	**Class/File**	**Line Number**
Player input is handled in an industry standard way across different devices		
Game world objects are manipulated using code		

Feature	**Y/N**
The game can be built for multiple platforms from the same project	
You have written a planning document that identifies and discusses cross-platform development issues	

The game compiles and runs error free	
The game executes without crashing	
One game project is used to build release builds for all target hardware	
You have written game code in a programming language such as C# or C++	
The game includes a GUI	
Code is executed in response to GUI events	
The GUI is handled correctly across different devices	
You have given a 5-8 minute presentation and integrated feedback into your documentation/game	
A release executable has been made for each target device and included in the submission	
An installer/uninstaller has been included for the PC build	
Source code documentation has been created (using a program like Doxygen) and included in the submission	