# Project assessment: Studio Simulation

## Criteria

### Unit code, name and release number

| |
|---|
| ICTICT433 - Build graphical user interfaces (Release 1)<br>ICTPRG535 - Build advanced user interfaces (Release 1)<br>ICTGAM535 - Develop complex 3-D software for games and interactive media (Release 1)<br>ICTGAM554 - Create games for mobile devices (Release 1)<br>CTGAM537 - Prepare games for different platforms and delivery modes (Release 1) |

### Qualification/Course code, name and release number

| |
|---|
| ICT50220 – Diploma of Digital and Interactive games |

## Student details

### Student number

| |
|---|
| |

### Student name

| |
|---|
| |

## Assessment declaration

*Note: If you are an online student, you will be required to complete this declaration on the TAFE NSW online learning platform when you upload your assessment.*

This assessment is my original work and has not been:

- plagiarised or copied from any source without providing due acknowledgement.
- written for me by any other person except where such collaboration has been authorised by the Teacher/Assessor concerned.

### Student signature and Date

| |
|---|
| |

| | |
|---|---|
| Version: | 20220809 |
| Date created: | 09 August 2022 |
| Date modified: | 09 August 2022 |

For queries, please contact:

St Leonards Tafe, J Block

# Assessment instructions

| Assessment details | Instructions |
|---|---|
| **Assessment overview** | The objective of this assessment is to assess your knowledge and performance in:<br>• object oriented programming concepts applicable to building graphical user interfaces<br>• application programming interface (API) features<br>• key testing procedures and documentation techniques required when building graphical user interfaces<br>• the client system, and the server system<br>• client-side programming<br>• object-oriented programming<br>• Animation, Audio, Cameras, Lighting, Physics, Mesh and Textures |
| **Assessment Event number** | 1 of 3 |
| **Instructions for this assessment** | This is a project-based assessment that assesses your knowledge and performance of the unit.<br>This assessment is in six parts:<br>1. Review and clarify requirements<br>2. Design the system<br>3. Develop the system<br>4. Develop test plan<br>5. Test the application<br>6. Evaluation and handover<br>And is supported by:<br>• Assessment feedback<br>• Supporting documents as provided |

| Assessment details | Instructions |
|---|---|
| **Submission instructions** | On completion of this assessment, you are required to submit it to your Teacher/Assessor for marking. Where possible, submission and upload of all required assessment files should be via the TAFE NSW online learning platform.<br><br>Ensure you have included your name at the bottom of each page of documents you submit.<br><br>It is important that you keep a copy of all electronic and hardcopy assessments submitted to TAFE and complete the assessment declaration when submitting the assessment. |
| **What do I need to do to achieve a satisfactory result?** | To achieve a satisfactory result for this assessment you must answer all the questions correctly.<br><br>If a resit is required to achieve a satisfactory result it will be conducted at an agreed time after a suitable revision period. |
| **What do I need to provide?** | • TAFE NSW student account username and password. If you do not know your username and password, contact your campus or service centre on 131601.<br>• Computer or other device with word processing software and internet access<br>• Writing materials, if required<br>• USB drive or other storage method with enough free space to save work to.<br><br>If assessment is completed off campus:<br>• Integrated development environment<br>• Libraries and re-use components and their licensing agreements<br>• Other software as required |

| Assessment details | Instructions |
|---|---|
| **What the Teacher/Assessor will provide** | • Access to this assessment and learning resources and any supporting documents or links.<br>• User requirements<br>• System and design specifications<br>• Integrated development environment<br>• Other software as required |
| **Due date**<br>**Time allowed**<br>**Location** | Friday of Week 18<br>Indicative time to complete:<br>• In class: 12 hours<br>• Out of class: 8 weeks<br>Assessment is to be completed both in and out of class. |
| **Supervision** | Part of this is an unsupervised, take-home assessment. Your Teacher/Assessor may ask for additional evidence to verify the authenticity of your submission and confirm that the assessment task was completed by you.<br>You may access your referenced text, learning notes and other resources. |
| **Assessment feedback, review or appeals** | In accordance with the TAFE NSW policy *Manage Assessment Appeals,* all students have the right to appeal an assessment decision in relation to how the assessment was conducted and the outcome of the assessment. Appeals must be lodged within **14 working days** of the formal notification of the result of the assessment.<br>If you would like to request a review of your results or if you have any concerns about your results, contact your Teacher/Assessor or Head Teacher. If they are unavailable, contact the Student Administration Officer.<br>Contact your Head Teacher/Assessor for the assessment appeals procedures at your college/campus. |

## Specific task instructions

The instructions and the criteria in the tasks and activities will be used by your Teacher/Assessor to determine if you have satisfactorily completed this assessment event. Use these instructions as a guide to ensure you demonstrate the required knowledge and skills.

You may have the option to record your participation and submit as video evidence. If you are submitting video evidence, you must:

- provide a video clearly meeting all requirements
- ensure you have access to the equipment and resources required to participate in the demonstration
- follow the Video recording instructions (pdf), which includes useful tips, links to resources and a demonstration video.

**Note: For some tasks you are required to take screenshots.**

- You can use the Snip & Sketch Tool (Windows), print screen or an equivalent tool to record your actions.
- To show this is your device, use Sticky Notes (Windows) or an equivalent tool to record your name and ensure you capture the sticky note in the screenshot.

## Project scenario

You are working as a programmer as part of the development team at Rubber Ducky Studio. The project manager, Odin Evans, has asked you to lead up with a client who is interested in funding his own game at the studio. You have been tasked with communicating with the client and building his dream game.

To whom it may concern,
I am looking to fund a development team to create a multi – platform, multi – player, 3D game with UI/2D card based unlockable elements for units/ items. The game must have Animation, 3D mesh and textures that can be changed. I am willing to fund a small team to get the prototype working over the next 8 weeks, once I have a design brief and if the project shows promise we can develop it further.
I have a few ideas and I want to know what idea you think is possible in that time frame with the team you have at Rubber Ducky Studios.

Idea 1: **Tower Defence Game**. Players will have the ability to unlock new characters or towers with different attack rates, damage, and damage types.

Idea 2: **Racing Game**. Players will have the ability to unlock new cars or characters with different speeds, acceleration, and handling.

Idea 3: **Battle Royal** or **MOBA Brawler**. Players will have the ability to unlock new weapons or characters with different abilities, damage, and range.

Idea 4: **Strategy Base Builder**. Players will have the ability to unlock new buildings or raiders with different damage, attack range, and damage types.

Idea 5: **DCCG** or **Digital Collectable Card Game**. Where players can unlock new cards or skins with different abilities, cost, and damage types.

Idea 6: **Rogue-lite, RPG strategy co-op.** Where the players can unlock new cards or weapons with different abilities, damage, and attack types.

Please let me know what Idea your team is planning on developing. I am planning on being hands on during the planning and development stages.

Regards
Ambros Chaos

The studios Lead programmer James-Rae is heading up another project that Odin has signed off on, but he will be providing mentoring as you and your team pick up your first client project at the studio.

## Part 1: Review and clarify requirements

The studios lead programmer and your team supervisor,  James-Rae, wants you to get started on planning which project to peruse for the client. James-Rae has said that client projects allow the studio to make extra money to fund other projects for the studio and that this client is returning to build another project with the employees of Rubber Ducky Studios. James-Rae suggests that you and your team create multiple mind-maps and base concepts of each idea to determine which one your team will be capable of creating. Once you have determined what idea shows promise with in the timeframe and with the skills of your team that you complete the following sections of the studios technical design document.

- Introduction
    - Rational
    - Background
    - Terminology
    - Non-Goals
    - Proposed Design
    - Software and Hardware Requirements
- Research
    - Platforms
        - What platforms are available
        - What 2 platforms could you build this game to work on
    - Frameworks
        - What Game engines exist to build on these platforms
        - What Networking frameworks exist for that game engine
    - Libraries
        - What libraries are needed for the following
            - Platforms
            - Networking
            - Game Engine

Collaborate with the Client and team leader to discuss the task, share ideas and confirm requirements. Record your discussion via video or writing (such as screenshots of emails or chats messages) and submit this discussion as evidence within your Technical Design Document as part of requirements.

**Submit the following for Part 1:**

- Requirements Technical Design Document

## Part 2: Design the application

The Lead programmer and team leader, James-Rae, wants you to help with the technical design document of the game application's systems, extending the work previously done.

James-Rae strongly suggests looking into using things such as Inheritance, Interfaces and Scriptable Objects. Below is a few resources that your team leader suggested looking at:

https://learn.unity.com/tutorial/inheritance

https://bitesizedtech.com/post/how-to-make-a-game-inheritance/

https://learn.unity.com/tutorial/interfaces#

https://circuitstream.com/blog/learn-c-for-unity-lesson-6-inheritance-and-interfaces/

https://learn.unity.com/tutorial/introduction-to-scriptable-objects

https://docs.unity3d.com/Manual/class-ScriptableObject.html

https://unity.com/how-to/architect-game-code-scriptable-objects

The Lead programmer and team leader, James-Rae, has also stated that you don't need to use all elements that he suggests, interfaces are the easiest way to achieve functionality in a system like this and he wants you , he has also told you that understanding how they work will help with how you design your system.

The following tasks will include filling in the technical design document and project requirements for the following sections

1. System Architecture
2. GUI/UI Layouts and Events
3. Required Components
4. Persistent storage
5. Pseudocode
6. UML Diagrams

Once you have made the technical design document James-Rae will need to see and review the plans to sign off so that you can start development.

### Task 1 System Architecture

To plan your game systems, you will need to work out what C# programming techniques you will be using. With that in mind, you need to develop and design the architecture of your overall system and create a visual architecture flow diagram. For this flow diagram James-

Rae suggested using https://app.diagrams.net/ and creating a mind map of how your game is going to flow and connect, or another mind map tool you are familiar with.

## Task 2 GUI/UI layouts and Events

Planning GUI/UI layouts and the events that run them are super important as your UI is how you communicate to the player what is happening while they are playing the game. Use https://app.diagrams.net/ to block out some mock GUI/UI menus and player interactions.

## Task 3 Required Components

Using the game engine of your choosing you will need to work out what components are built into the engine that you can use in your game. Built in components reduce pressure and time developing different systems.

## Task 4 Persistant storage

A core mechanic of your game will be tracking your players unlock progress. You will need to plan and develop a system that will track and store the players persistent data and progress.

## Task 5 UML

Develop and document the following UML diagrams for the scenario using an appropriate UML diagramming tool. The diagrams should cover all use cases applicable to the scenario.

1. Create an initial UML **class diagram** based on the requirements. Include the classes and their relationships.
2. Develop a UML **activity diagram**. *(behaviour diagram to describe dynamic aspects of the system) https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/*
3. Develop a UML **sequence diagram**. *(sequence diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration.) https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-sequence-diagram/*
4. Develop a UML **communication diagram.** *(communication diagram is an extension of object diagram that shows the objects along with the messages that travel from one to another.) https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-communication-diagram/*

5. Develop a UML **state diagram**. *(state diagram shows the different states of an entity.)*[https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-state-machine-diagram/](https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-state-machine-diagram/)

6. Refine your design and create a new **detailed UML class diagram**.

   *(Class diagram is a graphical notation used to construct and visualize object-oriented systems. A class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's)*

   *https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/*

## Task 6 Psudeocode

Plan and document your pseudocode system so that James-Rae can see the direction you are planning on taking for each class within your system.

## Task 7 Client Approval

**Submit the following for Part 2:**

- Requirements Technical Design Document

# Part 3: Develop the system

Develop the system according to the requirements and design.

## Task 1 Prototype

The team leader, James-Rae, has signed off and approved for you to start development and implementation of your game.

Complete the following tasks using the IDE and Game Engine that you have chosen. James-Rae also informs you that the implementation is using the canvas system and not the IMGUI system, but you can use IMGUI to prototype your behaviours.

1. Create a prototype of your system to test different ways you can implement your system. You should make this in a test scene for James-Rae to see your workflow,
2. Document any errors (bugs) found during the development of the system along with actions taken to resolve them and the status (whether the error/bug has been fixed).

You must ensure that any errors are fixed. Include one or more screenshots of the error and the debugging tool used.

### Task 2 Canvas

Incorporate canvas GUI components into your game systems.

1. Document any errors found and resolve any dependencies. Include one or more screenshots of the error and the debugging tool used.

### Task 3 Final Implementation

Once your system is working in a final project scene, implement your finished working system.

**Submit the following for Part 3:**

- Zipped folder containing:
    - IDE project files
    - Game Engine Project

## Part 4: Develop test plan

You need to design and develop the test planning for the system developed in Part 3. You will develop and perform test cases to ensure that the **syntax is correct**, system logic **meets user requirements,** and the **re-use components work correctly** within the project.

Use the technical design documentation template provided to document the following items.

### Task 1: Test planning

Develop your test plan with consideration of the software development life cycle. It must include:

a. test objectives

b. test types

c. deliverables

d. testing tools

## Task 2: Test cases

Develop three **test cases for the systems**. These must include the following testing techniques:

- One **UI error testing** test case covering input and operation of the system UI.
- One **use case testing** test case for testing your re-usable component is working as expected within the system.
- One **data condition testing** test case for testing the classes and their methods.

Each test case must include the following:

- Test Case Identifier (Test ID), name and description
- Priority (Low/Med/High)
- Test type/technique
- Test data and expected results
- Steps to be taken

## Task 3: Test method

Identify a **test method** including the **IDE** that you will use for testing the core game systems, with consideration for the organisational guidelines. Explain why you have chosen your testing method compared to other options available. Add this to the Appendix of the testing document.

## Task 4: Automated testing scripts

Using pseudocode, design and document a test script algorithm (including data structures) for your **data condition testing** test case in Task 2 to test the classes and methods. Consider your selected testing method from Task 3.

Describe the process to be undertaken, ensuring that you provide the steps of the algorithm using the table provided in the testing document.

Review the test data and expected results identified in Task 2 to ensure your test script algorithm covers the test case as described, including a full range of input values that complete the test case sufficiently.

**Submit the following for Part 4:**

- Testing document.

# Part 5: Test the application

Test your system by carrying out your test plan, using the IDE debugging tool and test method you identified in Part 4. Test and document the following tasks in your Test plan document from Part 4.

## Task 1: Implement test script algorithm

Write a script that implements the **data condition testing** test case outlined in Part 4 above to test the application. Confirm the test script builds and runs as expected within your chosen testing method.

## Task 2: Undertake testing

1. Follow your test plan and complete testing of **ALL** your test cases in Part 4, including executing the test script developed above. Document and log all test results including the script execution. You must modify the application where necessary to ensure that any errors are fixed, and it meets the requirements. Include one or more screenshots of the testing script being run.
2. Complete the Test results table then write a progress report analysing and summarising your test results.

## Task 3: Finalise testing report

Discuss your test findings with a development team member. This may include discussion of errors found, how to resolve the errors or how you resolved them. Record your discussion via video or in writing (such as screenshots of emails or chat messages) and submit this recording as evidence within the finalised testing report.

Your team member may include another student enrolled in the same unit.

**Submit the following for Part 5:**

- Finalised testing documentation.

## Part 6: Evaluation and handover

Now that your application is finished, you need to check that it meets the user requirements and hand it over to the team leader. You will need to submit your application to another person acting as the team leader to obtain sign-off, indicating their approval.

1. Sign off

**Submit the following for Part 6:**

Submit

# Assessment Feedback

*NOTE: This section must have the Teacher/Assessor and student signature to complete the feedback. If you are submitting through the TAFE NSW online learning platform, your Teacher/Assessor will give you feedback via the platform.*

## Assessment outcome

☐ Satisfactory

☐ Unsatisfactory

## Assessor feedback

☐ Has the Assessment Declaration for this assessment event been signed and dated by the student?

☐ Are you assured that the evidence presented for assessment is the student's own work?

☐ Was reasonable adjustment in place for this assessment event?

*If yes, ensure it is detailed on the assessment document.*

| *Comments*: |
| --- |
| |

## Assessor name, signature and date:

| |
| --- |
| |

## Student acknowledgement of assessment outcome

| *Would you like to make any comments about this assessment?* |
| --- |
| |

## Student name, signature and date

| |
| --- |
| |