COLLEGE OF CREATIVE INNOVATION

Programme

Bachelor of Software Engineering – 360 Credits

Year Two

Course

201 Introduction to Game Programming

(Level 6, 15 Credits)

Assessment

Project

Weighting within course:

100%

Assessment information

- Ask your tutor if you need further explanation or if the instructions are not clear.
- The purpose of this assessment is to assess your knowledge. As part of your academic and professional integrity, you must work alone on this assessment. In the event Yoobee suspects collusion, this will be addressed. For more information on plagiarism, please refer to the Student Handbook.
- Submit your completed assessment online (Blackboard) in the correct space provided.
- Percentage and feedback will be returned within 15 days of the submission date.

Learning outcomes (LO)

- **LO1:** Define a field of inquiry through the mapping of personal learning objectives to a specialist area of practice within software development.
- **LO2:** Demonstrate understanding in a specialist area of software development through practice-based inquiry.
- **LO3:** Differentiate individual learning strategies as a practitioner in a specialist area of software development.

Objective

The objective of this assignment is to develop an Integrated Design Document (IDD) and build a functional prototype for a game. The IDD will serve as a plan to guide the project's development process and task orientation. The focus of the prototype will be on functionality and understanding of the framework, with a test-level of aesthetics.

[LO1] Task 1: Approval: Game idea (Due week: 2nd week)

Provide a concise overview of your chosen game idea. The overview should cover game concept, gameplay mechanics, target audience, and potential market. Once your tutor approves the idea, proceed with completing the remaining tasks (Task 2 and Task 3). Note that Task 1 does not carry any marks. The purpose of this task is to assess the viability and potential of your game idea, as well as to provide constructive feedback on the concept, gameplay, marketability, and overall appeal. The feedback may be given orally or in written form, depending on your tutor's preference. Actively listen to the feedback and take notes for self-reflection and future improvement.

[LO1 & L02] Task 2: Integrated Design Document Creation (Due week: Final week)

1. Each student is required to create an Integrated Design Document, IDD in short, (also known as Game Design Document) for their game project.
2. Include the following components in your IDD:
    a. Game concept and description
    b. Target audience and platform(s)
    c. Unique Selling Point
    d. Game mechanics and gameplay overview
    e. Levels, objectives, reward and progression system
    f. User interface design and controls
    g. Schedule with completion due dates for each feature/component
    h. References to any external sources or inspirations
3. You may use a provided template of a Game Design Document or source one from reputable online resources. Make sure to customize it according to your game project.

[LO2 & LO3] Task 3: Prototype Development (Due week: Final week)

1. Using the IDD as a guide, students will develop a functional prototype of their game.
2. Focus on implementing the proposed functionalities and features, ensuring they are fully functional.
3. A test-level of aesthetics is expected at this stage. Concentrate on functionality rather than graphical polish.
4. Version control practices should be followed throughout the project, and the source files must be stored in a dedicated git repository along with the IDD.
5. Regular commits are expected to demonstrate familiarity with version control and project progression.

Deliverables

1. Integrated Design Document (IDD) – a wiki document stored in a git repository
2. Prototype - a functional prototype of the game, along with the source files, stored in a git repository.
3. A video recording demonstrating the working of your game and showcasing all implemented features.

Submission

Submit the following online (Blackboard) in the correct space provided, no later than the due date outlined for your class.

1. A link to the git repository containing your project along with the IDD
2. A video recording (or a link to the video)

Performance Criteria

Your assignment will be evaluated based on the following criteria:

	A-, A, A+	B-, B, B+	C-, C, C+	D
IDD (40%)				
Game Introduction (15%)	A comprehensive introduction to the game including the following: • Game concept and description • Target audience and platform(s) • Unique Selling Point	A detailed introduction to the game including the following: • Game concept and description • Target audience and platform(s) • Unique Selling Point	An introduction to the game including the following: • Game concept and description • Target audience and platform(s) • Unique Selling Point	Provides minimal or unclear information.
Game Features (15%)	All features including the following • Levels, objectives, reward and progression system • Game mechanics and gameplay overview • User interface design and controls are described with excellent details.	All features including the following • Levels, objectives, reward and progression system • Game mechanics and gameplay overview • User interface design and controls are described with good details.	All features including the following • Levels, objectives, reward and progression system • Game mechanics and gameplay overview • User interface design and controls are described with basic details.	Provides minimal or unclear information.
Timeframe (10%)	Provided a schedule with completion due dates for each feature/component with excellent details.	Provided a schedule with completion due dates for each feature/component with good details.	Provided a schedule with completion due dates for each feature/component with basic details.	Provides minimal or unclear information.
Prototype (60%)				
Persistence (15%)	The build has not pivoted or otherwise deviated from the submitted IDD at all.	The build pivoted or otherwise deviated from the submitted IDD with minute changes.	The build pivoted or otherwise deviated from the submitted IDD with small changes.	The build pivoted or otherwise deviated from the submitted IDD with major changes.
Performance (15%)	Implemented all proposed functionalities and features in the prototype with excellent quality. The build runs seamlessly without errors on the platforms specified in the IDD.	Implemented all proposed functionalities and features in the prototype with good quality. The build runs without errors on the platforms specified in the IDD.	Implemented most of the proposed functionalities and features in the prototype. The build runs with error on the platforms specified in the IDD.	Implemented few of the proposed functionalities and features in the prototype. The build runs with errors on the platforms specified in the IDD.
Code and Version Control Practices (15%)	Demonstrated the following with excellent details: • Version control practices should be followed throughout the project • Regular commits are expected to demonstrate familiarity with version control and project progression. • Code contributions are well-formatted, followed the don't-repeat-yourself rule (DRY), well-commented and make the best use of OOP practices where possible. Clever optimization, structure and intermediate level techniques used.	Demonstrated the following with good details: • Version control practices should be followed throughout the project • Regular commits are expected to demonstrate familiarity with version control and project progression. • Code contributions are well-formatted, followed the don't-repeat-yourself rule (DRY), well-commented and make the best use of OOP practices where possible. Clever optimization, structure	Demonstrated the following with basic details: • Version control practices should be followed throughout the project • Regular commits are expected to demonstrate familiarity with version control and project progression. • Code contributions are well-formatted, followed the don't-repeat-yourself rule (DRY), well-commented and make the best use of OOP practices where possible. Clever	Demonstrated the following poorly: • Version control practices should be followed throughout the project • Regular commits are expected to demonstrate familiarity with version control and project progression. • Code contributions are well-formatted, followed the don't-repeat-yourself rule (DRY), well-commented and make the best use of OOP practices where possible. Clever optimization,

		and intermediate level techniques used.	optimization, structure and intermediate level techniques used.	structure and intermediate level techniques used.
Video Evidence (15%)	A video-recording demonstrating the working of your game and showcasing all implemented features, with excellent quality.	A video-recording demonstrating the working of your game and showcasing all implemented features, with good quality.	A video-recording demonstrating the working of your game and showcasing all implemented features, with basic quality.	A video-recording provides minimal or unclear information.