2023-2024 - BSc. Digital Games Development / Creative Computing - Year 1

OOPMM – Multimedia Object Oriented Programming

Develop an OOP Multimedia Game

A03 (Home Assignment)

Assignment Guidelines

This assignment is a **home assignment** that is to be completed by **Monday 22rd January 2024 9.00am**.

The completed work is to be submitted through OOP Multimedia VLE course.

Some parts of the assignment require a certain amount of **research** to be completed successfully. It is entirely your responsibility to do so.

There are 4 sections in this assignment: **Section 1,2,3** and **4**.

Even if certain portions of the answers may be found in the unit material, you are to look up material yourself and to provide your original solutions.

Your submission should include **FOUR** files:

1. Research tasks: A **document** in .**docx** format with answers to Research tasks.
2. Practical tasks: A **compressed** .**zip** file containing your project folder.
3. Short video**: Screen recording with voice over** explaining your design approach and important own scripts.
4. AI assist (e.g. ChatGPT) histories: **If AI assistance was used**, demonstrate the process of game development and your problem-solving skills.

The College operates a cheating/**plagiarism policy** and any copied work will be penalized according to this policy.

All the institute procedures rules and regulations apply to this assignment.

Project Background and Design

You have been hired to design and develop a simple fun 2D/3D game. You may choose a Single Screen, Platformer, Scroller, Side-Scroller, Arcade, Retro or Adventure Game Types. (https://itstillworks.com/12342409/types-of-fun-2d-games) You are free to choose any genre and theme you wish, as long as Game Functionality List and Code Design Requirements are met.

You may make use of ready-made assets at: https://kenney.nl/assets?q=2d to develop your idea and implement your game.

Section 1 – Game Design

1. Prepare a Game Design Document, which outlines the following:

    a. Game Title and Description.

    b. Overview of the Game:
        i. Genre
        ii. Inspirational references
        iii. Basic Narrative
        iv. Include at least one piece of concept or Inspirational art.

    c. Game Loop and Storyboard

        i. How long is the game?
        ii. How many levels are there?
        iii. What is the average play time?
        iv. What are the objectives?
        v. How many playable characters? Can you customize or upgrade them/what can you customize or upgrade?
        vi. **Storyboard (minimum four sketches/elements)**:
            1. How will the player move throughout the game? What is the locomotion style? What are the major gameplay events?
            2. Screen Flow -- A graphical description of how each screen is related to every other and a description of the purpose of each screen.

    d. Mechanics

        i. Physics – How does the physical universe work?
        ii. Movement in the game
        iii. Objects – how to pick them up and move them
        iv. Actions, including whatever switches and buttons are used, interacting with objects, and what means of communication are used
        v. Combat – If there is combat or even conflict, how is this specifically modelled?
        vi. Economy – What is the economy of the game? How does it work?
        vii. Game Options – What are the options and how do they affect game play and mechanics?

e. Art Style / Assets

Include a description of your art style and supplement with art concepts or inspirational concepts (include a minimum of **two references**). Describe your style for the environment, characters, UI, etc. You can also link to a different area/scene it lives in.

SE2.2 - Design an Object-Oriented game or multimedia application		
	Maximum	Awarded
1a. Game title and description	1	
1b. Game type choice and description	1	
1c. Game Loop and Storyboard	4	
1d. Mechanics	2	
1e. Art style & assets	2	
Total	10	

Section 2 – Functionality Requirements

Implement the following functionalities and game features:

	Functionality	**Details**
2.1	**Minimum of 4 scenes:** Welcome Screen Level 1 Level 2 End of Game Scene	• Setup scene and orthographic camera. • Scene Background. • Scenes in Build settings. • Scene Management.
2.2	**Event driven Menu System.**	UI Menu including minimum of 3 options: • Start Game • Set Difficulty Level • View High Score
2.3	**Maintain persistence of Game Data and/or objects between scenes.**	Implement DontDestroyOnLoad(), statics and/or singletons to maintain game data and objects between scenes.
2.4	**Implement Scriptable Objects**	Research and implement Scriptable Objects to act as data containers for prefabs which have unchanging data an then maintained as Assets also reducing memory overhead.
2.5	**Load and Save Game Data and/or State.**	Research Data Serialization to store your current level, health, lives and score. https://learn.unity.com/tutorial/persistence-saving-and-loading-data

SE4.4 - Create robust code through the implementation of event handling techniques.		
	Maximum	Awarded
2.1 Minimum of 4 scenes	2	
2.1 Scene Transitions	2	
2.2 Event-Driven Menu System Buttons	2	
2.2 Three Event Triggers (correct events triggered in UI) (1 mark for event listener, 1 mark for each correct event trigger)	4	
Total	10	

KU4.1 - Show how persistence between different scenes and/or levels can be maintained according to set requirements.		
	Maximum	Awarded
2.3 Appropriate code for maintaining data between scenes (1 mark for partial functionality)	3	
2.3 Objects and/or data accurately maintained (1 mark for partial functionality)	2	
Total	5	

AA4.3 - Make use of persistent and secure data storage.		
	Maximum	Awarded
2.4 Correct use of Scriptable Objects	2	
2.5 Create Serialized Data Objects	1	
2.5 Loading Function	2	
2.5 Saving Function	2	
Total	7	

Section 3 – Code Requirements

You are required to make use of the following OOP concepts and code design patterns:

	Requirements	Details
3.1	Make use of S.O.L.I.D. Design principles.	Code can easily be extended, modified, tested, and refactored without any problems.
3.2	Implement a GameManager script.	Script to take care of game states, handle game data, and scene management.
3.3	Implement a GameData script.	Script to hold important game data and properties.
3.4	Use of try/catch, RequireComponent, checking for nulls etc. to capture runtime errors or to handle disruptions in normal flow.	e.g.1 Use try and catch blocks to handle incorrect user input and/or File IO errors e.g.2 RequireComponent() to ensure components referenced by a script are present and created if not.

AA3.3 - Produce a well-structured and scalable game or multimedia application.		
	Maximum	Awarded
3.1 Good use of OOP concepts learnt (access modifiers, inheritance, polymorphism, abstract classes, interfaces, getters/setters) to create robust, extendable code.	3	
3.2 GameManager script	2	
3.3 GameData script	1	
3.4 Robust code design including exception handling / require component etc.	1	
Total	7	

Section 4 – The Final Game

4.1 Design a class diagram to describe your OOP application.

4.2 Build and deploy your game to PC or mobile device.

4.3 Write a short report detailing:

a. The code design patterns used in your game. (refer to notes and/or 'Gang of four')
b. Suggestions for improvement.

4.4 Record a short **5 minute video** explaining your **design approach** and important **own scripts**. In your final project ZIP file, submit the **video clip** and any **AI assist (e.g. ChatGPT) histories** that demonstrate the process of game development and your problem-solving skills.

SE3.4 - Evaluate and defend the final outcome with a justification of techniques used.		
	Maximum	Awarded
4.1 Class Diagram	3	
4.2 Game Build	1	
4.3 Report detailing game design patterns and suggestions	2	
4.4 Video explanation	4	
Total	10	