2021-2022 - BSc. Multimedia Software Development - Year 1

OOPMM – Multimedia Object Oriented Programming

Develop an OOP Multimedia Game

A02 (Home Assignment)

Assignment Guidelines

This assignment is a **home assignment** that is to be completed by **Monday 24th January 2022 9.00am**.

The completed work is to be submitted through Moodle.

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 **TWO** 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.

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 game. You may choose a Single Screen, Platformer, Scroller, Side-Scroller, Arcade, Retro or Adventure 2D Game Type. (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. Choose and describe a Game Type (Single Screen, Platformer, Scroller, Side-Scroller, Adventure)

    c. Game Loop
        i. What will the player be doing in your game?
        ii. How will they do it?
        iii. How does the player progress through the game?
        iv. How is the narrative delivered?

    d. Scope
        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?

    e. Art Style / Assets
        Include a description of your art style and supplement with art concepts or inspirational concepts. 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	2	
1b. Game type choice and description	2	
1c. Game Loop	2	
1d. Scope	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	**Design one Animation using state machine system.**	Design a simple UML state diagram to describe your animation. https://docs.unity3d.com/Manual/StateMachineBasics.html
2.4	**Implement one Animation using state machine system.**	EXAMPLE: idle>>walk>>idle
2.5	**Maintain persistence of Game Data and/or objects between scenes.**	Implement DontDestroyOnLoad(), statics and/or singletons to maintain game data and objects between scenes.
2.6	**Implement High Score feature and save it using PlayerPrefs.**	Research and implement PlayerPrefs to store Highest Score recorded.
2.7	**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	

AA2.3 - Make good use of a state-machine system.		
	Maximum	Awarded
2.3 UML state diagram. (1 mark for states, 2 marks for correct transitions)	3	
2.4 Implement one animation. (2 marks for animation, 2 marks for correct transitions)	4	
Total	7	

KU4.1 - Show how persistence between different scenes and/or levels can be maintained according to set requirements.		
	Maximum	Awarded
2.5 Appropriate code for maintaining data between scenes (1 mark for partial functionality)	3	
2.6 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
Saving PlayerPrefs	2	
Create Serialized Data Objects	1	
Loading Function	2	
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	Research and describe Exception Handling.	
3.5	Use of try and catch blocks to handle disruptions in normal flow.	e.g. use try and catch blocks to handle incorrect user input and/or File IO errors

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 GamaData script	2	
Total	7	

KU4.2 - Describe how exception handling is used to address unexpected events.		
	Maximum	Awarded
3.4 Describe Exception Handling	3	
3.5 Demonstrate an example of Exception Handling	2	
Total	5	

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 pattern and OOP concepts used in your game.
b. Suggestions for improvement.

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