MCAST
MALTA COLLEGE of ARTS, SCIENCE & TECHNOLOGY

Assignment Cover Sheet

Please read carefully

- This assignment sheet is to be returned back to the lecturer by the student with the completed work. Work handed in after the deadline date will be penalized.
- Students caught copying from other students or plagiarizing (copying from lecturers' notes, handouts, slides, internet, books or any other printed or digital media) will be disqualified and will get a REFERRAL for their assignment or a FAIL if it is the last resit.
- An assessor has the right to ask the student to attend an interview without prior notice if the assessor wishes to confirm that the work submitted has been clearly understood by the student.
- It is the students' responsibility to keep a copy of the assignment for revision.
- Students should not share assignment sheets between different classes.

Student's Name	**Michael Cassar**		
Programme	**B.SC (Hons.) in Software**	Academic Year	**2013/2014**
Assessor's Name	**Gerard Said** **Paul Pulis**	Group/s	**1BSC3S 1BSC4S** **1BSC5S**

Unit No	**10**	Unit Name	**Entertainment and Interactive Software Development**		
Assignment No	**2**	Sit	**FirstSit**	Type	**Home**
Assignment Title	**Create functional interactive software**				
Issue Date		Deadline Date		Date returned to students	
Assignment IV	**Mark Anthony Farrugia**		Date	**23 Jan 2014**	

Pass Assessment Criteria			Merit Assessment Criteria			Distinction Assessment Criteria		
Criteria	Met	Not Met	Criteria	Met	Not Met	Criteria	Met	Not Met
Unit 10-EIS : P2.1			Unit 10-EIS : M2.1			Unit 10-EIS : D1.1		
Unit 10-EIS : P2.2			Unit 10-EIS : M3.1			Unit 10-EIS : D2.1		
Unit 10-EIS : P3.1			Unit 10-EIS : M4.1			Unit 10-EIS : D3.1		
Unit 10-EIS : P3.2								
Unit 10-EIS : P4.1								
Unit 10-EIS : P4.2								

Note : Computation of final grade for the unit will take into consideration each individual outcome as per assessment criteria.

(C*) denotes that the criteria was carried from a previous sit.

Assignment Status	

Assessment Criteria Description	
Unit 10-EIS : P2.1	Demonstrate knowledge of conceptual interactive software design
Unit 10-EIS : P2.2	Use scripting languages to create functional interactive software
Unit 10-EIS : P3.1	Demonstrate the capability of translating conceptual design to code
Unit 10-EIS : P3.2	Show understanding of advanced coding concepts
Unit 10-EIS : P4.1	Use complex coding techniques to create functionally interactive software
Unit 10-EIS : P4.2	Show the ability to deploy your interactive software on different platforms and environments
Unit 10-EIS : M2.1	Apply a correct strategy to storing game information in your conceptual design
Unit 10-EIS : M3.1	Apply appropriate techniques to generate advanced and maintainable code
Unit 10-EIS : M4.1	Present your interactive application appropriately
Unit 10-EIS : D1.1	Demonstrate creative thinking in interactive software creation
Unit 10-EIS : D2.1	Use critical reflection to comment on your conceptual design
Unit 10-EIS : D3.1	Write an innovative and entertaining interactive application

Assignment 2 – Design a computer game

Assignment Guidelines

- Read the following instructions carefully before you start the assignment.
- This assignment is a **home** assignment.
- Your submission should be **burned** to a blank CD.
- You must submit a CD which should contain your work in a folder named using the following format: **Name_Surname_Home_Assignment2_Sit1**
- Any documentation must be bound in a soft folder, along with the CD

Task 1 (P2.1) – Describe your game

Write a report describing the game you wish to implement. You need to include the following sections in your report:

- Background story of the game.
- Player task(s)
- How level difficulty increases.
- How scores are calculated.
- What the ultimate goal of the game is.
- How to play.

As an appendix to your report, please include a screenshot of what you believe your game will look like. You may create your own mockup or use sprites that you may have found on the internet laid out on a mockup of the screen using Adobe Photoshop.

P2.1 Demonstrate a knowledge of conceptual interactive software design

Task 2 (P3.1) – Create your game mechanics

Create the game you described in Task 1. You will need to implement the following features:

- Changing materials depending of the position of your game object.
- Changing materials depending on collision.
- Use of built in arrays.
- Use of general arrays of objects.
- Use of foreach loops.
- Use of scripts embedded in other GameObjects by referencing them.
- Using Unity's tag system
- Physics enabled collisions
- Use of the AddRelativeForce method
- Use of at least two different physic materials
- Creation of at least two different levels (with different backgrounds and object properties)
- Use of the keyboard as an input device

P3.1 Demonstrate the capability of translating conceptual design to code

Task 3 (P3.2) – Create your customized game menu

Create a menu to load the game as well as to view high scores. The menu must use custom fonts and buttons

- Full use of the GUI skin functionality with respect to input boxes and settings dialog boxes.

P3.2 Show understanding of advanced coding concepts

Task 4 (P2.2) – Deploy your game

Build your game application as an executable. Ensure that all resources are loaded correctly. The game must run without error.

P2.2 Use scripting languages to create functional interactive software

Task 5 (P4.1) – Refine your game

The game programming must have a clear structure. During your demonstration, you must explain how to create an additional level within the application and show the general outline of how such an additional level may be integrated in your software.

P4.1 Use complex coding techniques to create functionally interactive software

Task 6 (P4.2) – Explain how to deploy your game

Write a short paragraph explaining how to optimize your game to run on different platforms. Which code may be tweaked to allow for support on multiple platforms?

P4.2 Show the ability to deploy your interactive software on different platforms and environments

Task 7 (M2.1) – Implement additional functionalities in your game engine

Implement additional functionalities in your game engine, namely:

- At least three levels with different backgrounds, objects and environments
- A customized heads up display with at least one custom font and different colors for scores and any information that needs to be displayed to the player.
- Your game must use physics in an innovative way. You must therefore cater for multiple objects colliding with each other (three objects colliding, not simply two objects colliding with each other), and they must affect each other programmatically.

M2.1 Apply a correct strategy to storing game information in your conceptual design

Task 8 (M3.1) – Implement database support using a library
Implement a database enabled player profile system, allowing the player to save his/ name and personal details within the application, and load his name and past highest score. Use the appropriate database libraries and package the database with your application.

M3.1 Apply appropriate techniques to generate advanced and maintainable code

Task 9 (M4.1) – Present your interactive application appropriately
Enhance the description you presented in Task 1 with additional screenshots and formatting. It should be designed to encourage a person to play the game, with detailed instructions outlining the goal of the game

M4.1 Present your interactive application appropriately

Task 10 (D3.1) – Implement enhanced database support
Implement a system to save your game state at any point during the game using a shortcut key. You must save all the variables necessary to reload that game state in the database. You must also implement a system to reload that exact game state through the game menu.

D3.1 Write an innovative and entertaining interactive application

Task 11 (D2.1) – Comment on your implementation
Write a report on your implementation. Answer the following questions:

- Do you consider your game fun to play?
- How would you improve your game mechanics to make it more interesting to play?

D2.1 Use critical reflection to comment on your conceptual design

Task 12 (D1.1) – Demo your game

Play your game in front of the lecturer. You need to explain how the game is played in a clear way. The game must work correctly and clearly

D1.1 Demonstrate creative thinking in interactive software creation