

OOSE 2017 – Project

Marks: 60% of overall marks for module.

Note: Details below are provisional and may change slightly based on the feedback from external examiner.

Project deliverables should be completed in groups of three (3) people. Please agree the groups within a week. The Project needs to be completed and returned to the lecturer on paper and on Moodle in the lab of 29th April 2017.

The main objective of this project is the development of a computer system for a **Minecraft Game System**.

- The project is composed of multiple sections or deliverables.
- The project work should be carried out in as homework and in specified labs.
- You may use any diagramming tool (Rational Rose, Visio, Gliffy, Creately etc.) to carry out the project but the final work must be presented on paper.
- A UML tool must be used to complete at least three deliverables.
- An interim project comprising of deliverables **i – iv** should be submitted in class on **28th March 2017**.

Minecraft

The problem is to design a cloud based game for children age 5+. The trigger for the problem is based on Minecraft.

<http://www.youtube.com/watch?v=FP5p0FDL83E>

From a technology viewpoint the system should be compatible with PCs & mobile devices and deployed via the Cloud.

Anything novel to do with Minecraft software will be acceptable for this project.

More information on Minecraft is available at [the Minecraft Wiki](#).

Deliverables

Group = **GREEN** Individual = **BLUE**

For the above Problem description:

1. **Identify the actors.** (5 Marks)
2. **Construct a Use Case Model.** (5 Marks)
3. **Describe in detail any use case from the use case model. The use case must contain an alternate flow.** (10 Marks)
4. **Create a Project Plan to deliver your application. The plan must include a minimum of 3 nesting levels and include all the major tasks and deliverables.** (10 marks)
5. **Create a conceptual class diagram of the chosen use case. The conceptual class diagram should demonstrate the use of many of the following: attributes, relationships, navigability, association class, multiplicity and composition.** (10 Marks)
6. **Create a glossary that lists and defines all the terms that require clarification.** (5 marks)
7. **Draw a System Sequence diagram from the conceptual class diagram.** (10 Marks)
8. **Develop Contracts for a minimum of two of the system operations in the system sequence diagram.** (10 marks)
9. **Draw Communication diagrams based on the above 2 contracts. The communication diagrams should demonstrate the use of design patterns.** (10 marks)

10. Presentation (how well does the package of models look?). (5 marks)
11. Use of a UML tool. (10 marks)
12. Put together a Testing Plan outlining how you propose to validate the application and verify that it is free of defects. (10 marks)

Learning Outcomes Covered in project

On successful completion of this module the learner will be able to:

LO1 Demonstrate the conceptual, practical and technical skills of planning and monitoring a project plan using an appropriate CASE tool

LO3 Describe in detail the theory, concepts and methods pertaining to the Unified Modelling Language (UML).

LO4 Create requirements using use case modelling concepts.

LO5 Demonstrate conceptual and technical skills in the analysis, design and implementation of a software system using Object Oriented Concepts.

Notes

UC Modelling Principles

- Relevant Actors are Identified
- Relevant UC's are identified
- UC's provide value to the actor
- UC descriptions are based on a template
- The UC is described in great detail
- The activation is correctly written
- Mainflow is a sequence of logical transactions
- The mainflow is enumerated
- Alternate flows are described and labelled correctly

Class Diagram Concepts

The conceptual class diagram should demonstrate the use of

- names from the problem domain as documented in the use case,
- attributes and methods,
- relationships,
- roles,
- constraints,
- association class,
- multiplicity
- composition

Principles for Interaction Diagrams and Patterns

- Demonstrate interaction diagrams: communication diagrams, sequence diagrams, statechart diagrams
- Communication diagrams demonstrate the use of
 - Link
 - Messages
 - Return type
 - Message to itself
 - Iteration

- Creation of instances
- Number sequencing
- Conditional messages
- Collections
- Message to a class object
- Patterns (Controller , Creator, Expert)