

# OOSE 2018 – Project

**Marks:** 60% of overall marks for module.

- Project deliverables should be completed in groups of three (3) people.
- Team disruptions should be managed within the team, if agreement cannot be reached please contact the lecturer as soon as possible.
- The main objective of this project is the development of a computer system for a **Clash of Clans Game System**.
- The project is composed of multiple sections or deliverables (12 in all).
- The project work should be carried out in **as homework and in specified labs**.
- The final work must be presented on **paper** as well as via **Moodle**.
- **IBM Rational Rose** or **Visual Paradigm** must be used to complete the relevant deliverables.
- The Final Project needs to be completed and returned to the lecturer **on paper and on Moodle** in class on **Tuesday 17 April 2017**.
- An interim version of the project containing deliverables i, ii, iii, iv and xi should be submitted on Moodle / TurnItIn only by **Saturday 24 March**.
- This interim project will be marked and worth **25%** of the project while the full project will be worth **35%**. The marks indicated next to the deliverables represent the percentage marks for the full project. Any diagramming tool can be used for the interim project.



# Clash of Clans

The problem is to design a cloud based game for children age 7+. The trigger for the problem is based on Clash of Clans <https://www.youtube.com/user/OfficialClashOfClans>

Clash of Clans is about developing a strategy, building defences and attacking the enemies.

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

**Anything novel** to do with Clash of Clans software will be acceptable for this project.

More information on Clash of Clans is available at the [Clash of Clans Wiki](#).

## Deliverables

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

(Ensure that it is very clear which person carried out each of the individual tasks.)

**For the above Problem description:**

- (i)    Identify the **actors**.  
(4 Marks)
- (ii)    Construct a **Use Case Model**.  
(6 Marks)
- (iii)    Describe in **detail any use case** from the use case model. The use case must contain an alternate flow.  
(10 Marks)
- (iv)    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)
- (v)    Create a **glossary** that lists and defines all the terms that require clarification.  
(5 marks)
- (vi)    Draw a **System Sequence diagram** from the conceptual class diagram & the detailed Use Case Description.  
(10 Marks)

- (vii) Develop **Contracts** for a minimum of **two** of the system operations in the system sequence diagram. (10 marks)
- (viii) Draw a **Communication diagram** based on the above contracts. The communication diagram should demonstrate the use of **design patterns**. (15 marks)
- (ix) **Presentation** (how well does the package of models look?). (5 marks)
- (x) Use of **Rational Rose / Visual Paradigm**. (5 marks)
- (xi) 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)  
Describe how you monitor your project plan as you develop your project. Use Microsoft Project to produce the Gantt chart and any diagramming tool of your choice to produce the Work Breakdown Structure (WBS).
- (xii) Put together a **Testing Plan** outlining how you propose to validate the application and verify that it is free of defects. (10 marks)

# 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)