

Course Assessment Specification (CAS)

Programme Title : Computer Engineering and Software Systems
Coursework Title : Programming Project
Module Name (UEL): Computer Networks and Distributed Systems
Course Name (ASU): Distributed Computing
Module/Course Code: EG7533/ CSE354
Level UEL/ASU : 5 / 3
UEL Credit Rating : 15 Credits **ASU Credit Rating:** 3 Credits
Weighting : 35%

Maximum mark available:

- 35 on software projects (1 project)

Lecturer : Prof. Ayman M. Bahaa-Eldin

Contact : If you have any issues with this coursework, you may contact your lecturer.

Contact details are: Email: ayman.bahaa@eng.asu.edu.eg

Hand-out Date : As shown in submission matrix

Hand-in Date : As shown in submission matrix

Hand-in Method : Submission through LMS

Feedback Date : Your work will be marked and returned within two weeks.

Introduction

This coursework is itemized into several parts to get the 35 marks associated to it.

You must use the templates provided by the instructor to prepare your work.

All assignments and projects will be handed-in electronically, while quizzes and exams are written

.

Learning Outcome to be assessed

7. Design a distributed computing model to solve a complex problem
9. Design and implement a distributed computing model
9. Configure a working environment for distributed computing
10. Work and communicate effectively in a team

Detail of the task

Attached separate file for each task.

89% and above:

Your work must be of outstanding quality and fully meet the requirements of the coursework specification and learning outcomes stated. You must show independent thinking and apply this to your work showing originality and consideration of key issues. There must be evidence of wider reading on the subject. In addition, your proposed solution should:

- illustrate a professional ability of drafting construction details,
- express a deep understanding of the in-hand problem definition,
- and applying, masterly, the learned knowledge in the proposed solution.

76% - 89%:

Your work must be of good quality and meet the requirements of the coursework specification and learning outcomes stated. You must demonstrate some originality in your work and show this by applying new learning to the key issues of the coursework. There must be evidence of wider reading on the subject. In addition, your proposed solution should:

- illustrate a good ability of drafting construction details,
- express a very good understanding of the in-hand problem definition,
- and applying most of the learned knowledge, correctly, in the proposed solution.

67% - 76%:

Your work must be comprehensive and meet all of the requirements stated by the coursework specification and learning outcomes. You must show a good understanding of the key concepts and be able to apply them to solve the problem set by the coursework. There must be enough depth to your work to provide evidence of wider reading. In addition, your proposed solution should:

- illustrate a moderate ability of drafting construction details,
- express a good understanding of the in-hand problem definition,
- and applying most of the learned knowledge, correctly, in the proposed solution.

60% - 67%:

Your work must be of a standard that meets the requirements stated by the coursework specification and learning outcomes. You must show a reasonable level of understanding of the key concepts and principles and you must have applied this knowledge to the coursework problem. There should be some evidence of wider reading. In addition, your proposed solution should:

- illustrate a fair ability of drafting construction details,
- express a fair understanding of the in-hand problem definition,
- and applying some of the learned knowledge, correctly, in the proposed solution.

Below 60%:

Your work is of poor quality and does not meet the requirements stated by the coursework specification and learning outcomes. There is a lack of understanding of key concepts and knowledge and no evidence of wider reading. In addition, your proposed solution would be:

- Illustrate an inability of drafting construction details,
- Failed to define the parameters, limitations, and offerings of the in-hand problem,
- Failed to apply correctly the learned knowledge for proposing a valid solution.

Academic Misconduct

The University defines Academic Misconduct as 'any case of deliberate, premeditated cheating, collusion, plagiarism or falsification of information, in an attempt to deceive and gain an unfair advantage in assessment'. This includes attempting to gain marks as part of a team without making a contribution. The department takes Academic Misconduct very seriously and any suspected cases will be investigated through the University's standard policy. If you are found guilty, you may be expelled from the University with no award.

It is your responsibility to ensure that you understand what constitutes Academic Misconduct and to ensure that you do not break the rules. If you are unclear about what is required, please ask.

A multi-player distributed 2D Car Racing Game along with chatting feature

This project should be done in teams from two to four students.

For this project, you will design, implement, and thoroughly test a distributed system,

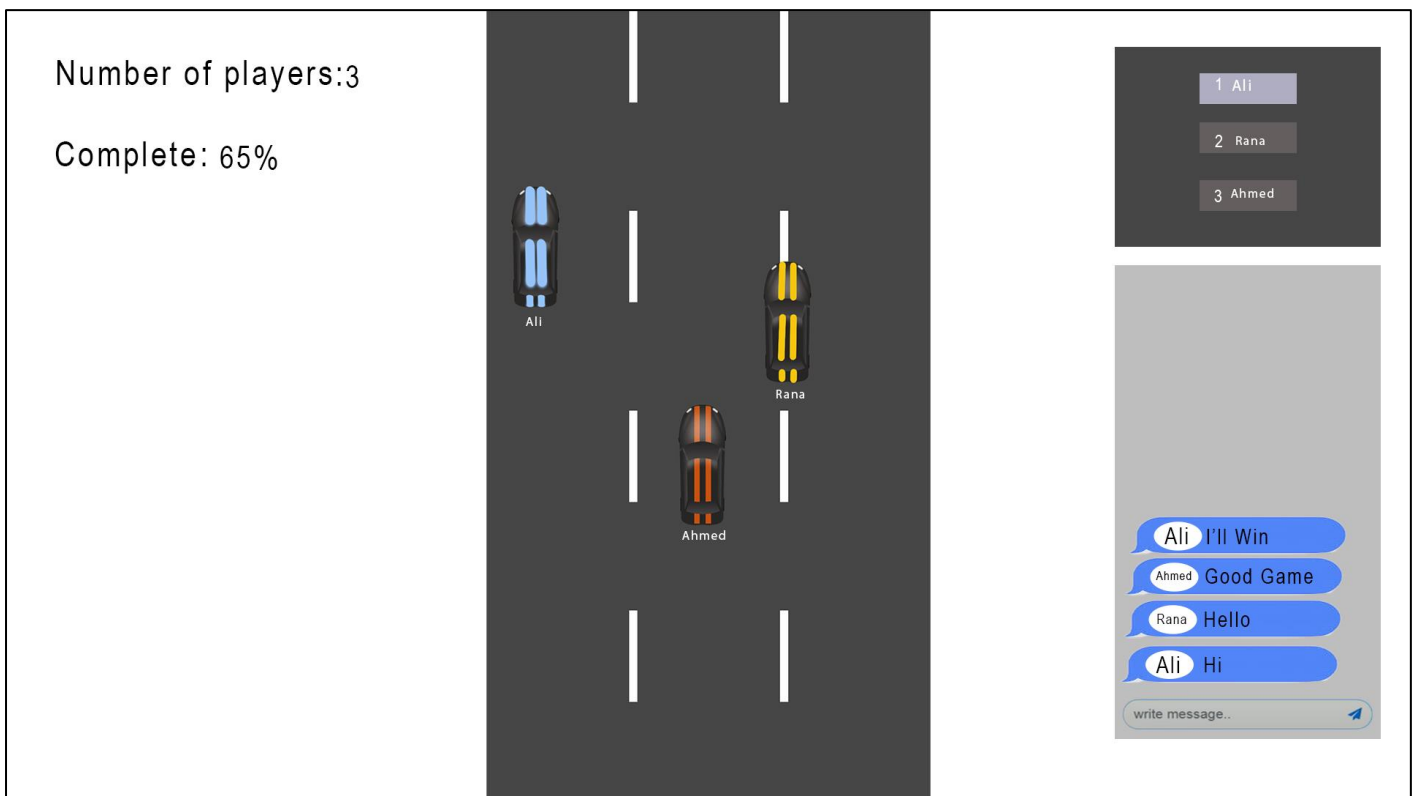
Requirements

Since this is a course in distributed systems, we want it to have “interesting” features from a systems perspective. Here are some important properties your system should have:

- The system must support multiple, autonomous agents (either human or automated) contending for shared resources and performing real-time updates to some form of shared state.
- The state of the system should be distributed across multiple client or server nodes.
- The system should be robust
- The system should be able to continue operation even if one of the participant nodes crashes.
- It should be possible to recover the state of a node following a crash, so that it can resume operation.
- You may use any of the packages in the Python (or comparable packages in whatever language you choose.)
- You are allowed to use external packages to support the application or user interface.

You are required to use the information you have learned in the course content, labs and using the tools provided, you are required to build a 2D car racing, in the simplified version.

1. The system should support real-time playing and viewing by multiple participants.
2. The system should support chat between participants during/before/after playing.
3. Multiple replicas would be maintained for fault tolerance.
4. Caching and/or copy migration would be useful to minimize application response time.



Note: You can choose your own design for the game.