CMP501 Network Game Development (2023-24)

Assessment Brief

Laith Al-Jobouri
School of Design and Informatics

Key Dates

Coursework due date: **12noon on Tuesday, 19th December 2023**
Feedback return date: **Tuesday, 23rd January 2024** (15 working days after submission)

Module Learning Outcomes

1. Critically evaluate network architectures and protocols for use in games development.
2. Assess and select development methodologies for networked computer games.
3. Assess and critically evaluate methods for dealing with the unpredictability of network conditions.
4. Design, develop and critically evaluate a fast-action network computer game application.

Introduction

In this coursework, you will demonstrate your ability to design, develop, test and critically evaluate a network application with a focus on the technologies used in fast-action networked computer games. You are tasked with developing a real-time graphical network application that makes use of appropriate techniques and architectures to ensure a smooth networked experience even under poor network conditions.

You must work on your own for this coursework.

If you have any questions, please contact the module tutor (l.al-jobouri@abertay.ac.uk).

Requirements

Your application must demonstrate:

- A simulated world containing two or more moveable objects, at least one of which must be controllable by the user;
- Network communication between machines (running either multiple copies of the same program, or two or more different programs);
- Synchronisation of object positions between the machines, so that all machines are capable of showing a consistent view of the game world;
- Using appropriate prediction and/or interpolation techniques to maintain smoothly synchronised movement even when the network connection is affected by latency.

Your application may run on any operating system or platform and may make use of external libraries, provided these are clearly acknowledged and don't prevent you from demonstrating the required skills. The networking solution should be built with WinSock or any other appropriate low-level Socket programming interface (e.g. SFML or .NET Sockets).

Choosing an application

You may implement any application you like provided it meets all of the requirements above and allows you to adequately demonstrate your knowledge. For example, you might choose:

- A multiplayer network game, where each player runs a copy of the game and controls their own character within the game world;
- A single-player game that can replicate its display onto other screens (e.g. for online observers, a tournament video wall or a flight simulator cockpit);
- An interactive simulation (e.g. Boids), where the user can influence multiple computer controlled actors distributed across several hosts.

Keep in mind that credit will not be given for clever game design or complex algorithms unless these features are part of the network solution. If in doubt, you are encouraged to discuss your choice of application with the module tutor and to ensure that it is appropriate for the assessment.

Report

You will submit a short report along with your code, in which you **explicitly justify your technical choices** using the knowledge you've gained from the module and additional research. For example, if you've chosen to use a peer-to-peer architecture, you should explain why that's an appropriate choice in the context of your particular application – e.g. you might talk about the expected number of players, the characteristics of the networks that players are likely to use, the requirements for efficiency of network communication, etc. You don't need to explain what a peer-to-peer architecture is.

Your report must be **no more than six A4 pages long**, including diagrams. It must be written in **formal academic style** and include **appropriate references** (from reliable sources) to peer-reviewed academic sources in Harvard Cite format. Your report should have the following section headings describing different aspects of your design:

- **Architecture** – the network architecture you've chosen (e.g. client-server, peer-to-peer);
- **Protocols** – the application-layer protocol(s) that you've designed, and the transport-layer protocol(s) you've chosen;
- **API** – the network API you've chosen, and any comments on its suitability for your application;
- **Integration** – how the networking code is structured, and how it's integrated with the rest of your application (e.g. using asynchronous I/O);
- **Prediction** – the prediction and/or interpolation techniques you've used;
- **Testing** – a practical evaluation of your program under poor network conditions using a tool like Clumsy to simulate latency/packet loss, and a critical discussion of the results of this testing (what can/can't it handle? how could you improve it?).

Please use diagrams where appropriate – e.g. when describing your interpolation strategy, and/or when presenting performance results.

Demonstration video

You need to record a short video demonstration of your application, showing what it does during normal operation, and how it performs under both good and poor network conditions. The video will be viewed by the marker, by the module's internal moderator, and by the module's external examiner – so please make sure it makes sense to someone who isn't already familiar with your project.

You can make use of the software like **OBS Studio** to capture your screen.

Please make sure to embed the video in the submission or include the video into the archive in your submission. Please ensure that the video is viewable via the link you provide (if you provide a link) and make sure that both Laith Al-Jobouri and Andrei Boiko can share it with the external examiners.

Submission

There will be 2 assignments on MyLearningSpace – one for the report and one for the application.

The report must be submitted in PDF format to the "**Report**" section. Note that it will be checked for originality using Turnitin, so don't forget to paraphrase any information from other sources and reference all sources in accordance with university regulations.

The ZIP archive containing the following must be submitted to the "**Application**" section:

- The complete source code for your application.
- A ready-to-run version of your application (e.g. a Windows .exe file. **Please test this!**);
- A short "read me" file (no more than a couple of paragraphs), in plain text or PDF format, describing how to run and control your application; and including a link to your demo video.

To reduce the size of your ZIP file, please ensure that you have cleaned out any temporary files from your application's source code before submission – if you've used Visual Studio, use "clean solution" option and then delete any remaining .obj, .ipch and .sdf files and the .vs folder. For external libraries, include the download link and short instructions on how to configure your project in the readme file, rather than including a copy of the library itself.

You must submit your report, ZIP file and link to the demo video through MyLearningSpace by **12noon on Tuesday, 19th December 2023**.

Feedback will be returned 15 working days after the submission.

Use of Generative AI in CMP501

Please Note The stipulations below apply only to CMP501 Assessment, You will normally be given specific guidance regarding Generative AI for each module that you study.

Generative AI (GAI), such as ChatGPT, Google Bard, and others, can be used to aid report writing, code development, data analysis, language and grammar correction, to name but a few. In this module we will integrate AI literacy focusing on its application in report writing, code development.

How Generative AI should be used

You may use Generative AI to:

a. understand concepts or questions related to lectures, tutorials, practical, seminars or the assessment for CMP501.
b. generate ideas for **code and reporting**.
c. get suggestions for **code improvement and report writing**.
d. assist in **debugging code**.

You may <u>NOT</u> use Generative AI to:

Generate any type of answers, code or presentation for the assessment of CMP501.

Any evidence of use to generate answers will be referred to the **<u>Student Disciplinary Officer</u>** within the School and may result in **<u>Academic Misconduct charge</u>**.

You may use specific tools, such as Grammarly for aiding Grammar and Structure.

a. It is recommended that you keep a record of your interaction with these tools.
b. It is recommended that you have evidence of drafts of your work.

You should adhere to the following:

1. Always reference AI use: Consider AI like any other source. If AI helps draft or edit a report, reference it appropriately.
2. Record Interaction: When using ChatGPT or a similar tool, include a transcript of your interaction in an appendix (for reports).
3. Summary: Include a summary in an appendix stating how you used the AI.
4. In-text Citation for AI: For each paragraph aided by AI, include an in-text citation.
5. Own thoughts: The majority of the insights and arguments in your work should be your own. Using AI for phrasing or structure is acceptable, but the core ideas must be original.
6. Be Responsible: Use AI responsibly to enhance your work. Critical thinking and creativity should come from you Speak with lecturer if you are unsure

Additional Points to Consider

a. Ethical AI Usage: When using AI in CMP501 assessment ensure the data and methods you employ adhere to privacy laws and ethical considerations.
b. Factual Accuracy: Consider the information generated by AI is factually accurate and scientifically valid. The information generated should be checked against peer-reviewed sources.