

Assessment Brief

Laith Al-Jobouri

School of Design and Informatics

Key Dates

Coursework due date: Tuesday, 14th December 2021

Feedback return date: Tuesday, 18th January 2022 (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. Describe common security problems in network game applications and apply basic secure software engineering practices to mitigate them.
5. 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 engage with the project proposal exercise to ensure that it is appropriate for the assessment.

Presentation

You will do a short presentation, in which you will 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 presentation should last no longer than 10 minutes; there will be approximately five minutes for questions and discussion afterwards.

Your presentation will cover the following topics:

- The network architecture you've chosen (e.g. client-server, peer-to-peer);
- The application-layer protocol(s) that you've designed, and the transport-layer protocol(s) you've chosen;
- The network API you've chosen;
- How the networking code is structured, and how it's integrated with the rest of your application (e.g. using asynchronous IO);
- The prediction and/or interpolation techniques used;
- Any other design or development decisions that relate to the networking capability;
- A critical discussion of the effectiveness of your solution (e.g. what kinds of network conditions will cause it not to work well).
- The results of any testing carried out utilising tools (e.g. [Clumsy](#)).

You will be assigned a presentation time slot **during the week beginning Monday 10th January 2022**; a list of these times will be available to you on MyLearningSpace beforehand. If the time assigned to you isn't suitable, please get in touch with the module team as soon as possible to arrange an alternative.

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 upload your completed video to MS Stream and include a link to the video in your submission. Please ensure that the video is viewable via the link you provide and make sure that both Andrei Boiko and Laith al-Jobouri are set as “owners” of the video to allow us to share it with external examiners.

Submission

There will be 2 assignments on MyLearningSpace – one for the application and one for the slides of your presentation.

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 ZIP file and link to the demo video through MyLearningSpace by **23:59 on Tuesday, 14th December 2021**.

Your presentation slides must be submitted to the “**Presentation**” section **on or before Monday, 10th January 2022**. You are allowed to make minor adjustments to the slides before the actual presentation, but you will need to resubmit them if that is the case.

*Feedback will be returned 15 working days after the submission of the application, unless your presentation is delayed or rescheduled – in which case the feedback will be returned after the presentation takes place.