


# (Game) Networking

Term 2.3



Publication date	25rd of February 2021
Version	1.7
Module coordinator	J.C. Wichman (HWI05)
Lecturers	J.C. Wichman (HWI05)
Tutors	Paul Bonsma (PBO06)
CMGT roles	Engineer 

## 1 General overview

Module Name	Networking
Unit code	L.26032
Year and Term	2.3
CMGT roles	Engineer
Credits	3 ECTS (84 hours)
Lessons	5 lectures, 6 labs
Study load	48-84 hours (based on 8-12 hours per week) (1.5 hours of lecture + 2.5 hours of lab + 4-8 hours of homework) * 5
Responsible lecturer	Hans Wichman
Lesson structure	1.5 hours lecture, 3 hours labs
Module summary	In this course you'll learn about basic networking protocols and how to build a game with them.
Industry relevance	Almost every game and/or app has some sort of networking functionality build into them these days.
Type of exam	Assessment (based on assignments)
Exam code	T.51758
CMGT Competencies	1. Technological research and analysis 2. Designing, prototyping and realizing 3. Testing and rolling out
Required prior knowledge and skills / conditions for enrolment	Basic knowledge of C#, Unity and software architecture principles.
Preparatory for:	Your professional multiplayer networking career.

## 2 Why this module?

This cluster teaches the basics of network programming using low level socket programming in C#. We will use a practical approach with lots of examples and exercises.

This course focuses on the core principles underlying any networking system, whether it be Unity, Unreal or the GXP Engine. This means you no longer depend on what such a tool offers you, instead you can depend on the core foundations of network programming, which are not subject to change for any time soon: **sockets**. In addition, this approach is a better match to the demands of the industry, which isn't looking for "button pushers", but for people who actually know what they are doing.

In addition this course will provide valuable networking insights that will also benefit you, should you choose to go with packages such as Photon or Mirror in the future instead of Sockets.

### 2.1 What happens in the labs and lectures?

Each lecture dives into explaining a specific learning goal/topic for the course including the assignment that goes with it. You can work on the assignments during labs and get feedback on your work before you sign it off.

### 2.2 How does this module relate to other modules in the CMGT study programme?

We expect that you are proficient in C#, have a working knowledge of Unity and understand a given software architecture (after some explanation).

## 3 What are you going to learn in this module (learning objectives)?

After this course you will be able to:

1. Recall basic TCP/IP theory and principles
2. Use reliable (object-based) messaging to create a networked application, meaning you can:
  1. communicate correctly between application instances using TCP
  2. employ non-blocking IO principles to avoid tying up the render loop while networking
  3. employ error handling to ensure server robustness
  4. implement strongly typed (de)serializable protocol messages
  5. design an application protocol and supportive client/server architecture

## 4 Which resources do you need?

- Unity / Visual Studio

*Please install/update to the latest 2019 LTS version on the day of the first lecture. This ensures we are all using the same Unity version for the duration of the course and can easily exchange Unity projects back and forth.*

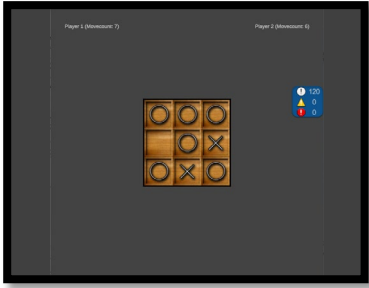
## 5 What does the programme of this module look like?

week	Lecture/Lab	Topic(s)
1.3	Lecture	Basic networking principles
1.3	Lab	Assignment 1 - Getting started with TCP <i>(Download the starting code for this assignment + criteria from blackboard)</i>
1.4	Lecture	TCP Challenges & Solutions
1.4	Lab	Assignment 2 - Create a textbased chatbox in Unity <i>(Download the starting code for this assignment + criteria from blackboard)</i>
1.5	Lecture	Object based protocols
1.5	Lab	Assignment 3 - Create a 3D avatar based chatlobby in Unity <i>(Download the starting code for this assignment + criteria from blackboard)</i>
1.6	Lecture	Finite state machines, UIManagers & GameRooms: Building a game
1.6	Lab	Assignment 4 - Create a simple turnbased game in Unity <i>(Download the starting code for this assignment + criteria from blackboard)</i>
1.7	Lecture	<... to be announced ...>
1.7	Lab	Finish your assignments
1.8	Lecture	<i>None (Easter Monday)</i>
1.8	Lab	Finish your assignments
1.9	Assessment	16th of April

## 6 How is this module assessed?

### 6.1 Assessment

For this course, you have to complete 4 assignments:

	
<p><i>Assignment 1. Getting started with TCP</i></p>	<p><i>Assignment 2. Text based chatbox</i></p>
	
<p><i>Assignment 3. 3D Avatar chatbox</i></p>	<p><i>Assignment 4. Turn based game</i></p>

You can work on these assignments at home and during scheduled lab hours. The assignments are individual and also the student achievements are individually assessed. As usual you have to produce your own code and should be able to explain every single line of code and theory behind it. Grading is done based on a rubrics form and each assignment including its grading criteria will be released on a per week basis. **Starting code & projects are provided for every assignment.**

### 6.2 Procedure

- All your assignments (1-4) should be handed in, in 1 zip file, through blackboard.saxion.nl before the assessment in week 3.9.
- Each student gets 20 minutes for the assessment.
- To be allowed to participate in the final assessment, the sufficient parts of **assignments 1, 2 & 3 must be finished and graded before week 3.9** (we will assess and grade them during the online lab in a breakout room).
- During the final assessment, you have to be able to answer questions related to everything taught during the *first 4* lectures and all of your assignments.
- The grades will be published through bison.saxion.nl.

### 6.3 Criteria & Assessment form

The criteria can be found on blackboard under the "Course assignments" section (soon).

### 6.4 Resit

Redo: will take place in 4.9 (one quartile later)

## 7 Who are the contact persons for this module?

### Module coordinator:

Hans Wichman [j.c.wichman@saxion.nl](mailto:j.c.wichman@saxion.nl)

### Lecturers:

Hans Wichman [j.c.wichman@saxion.nl](mailto:j.c.wichman@saxion.nl)  
Paul Bonsma [p.s.bonsma@saxion.nl](mailto:p.s.bonsma@saxion.nl)

## 8 References

- TCP/IP Sockets in C# - Practical Guide For Programmers, David B. Makosfke & Michael J. Donahoo  
*Covers all of the information presented during the lectures and more (although in a different format and form).*
- Multiplayer Game Programming: Architecting Networked Games, Josh Glazer & Sanjay Madhav  
*Recommended for further study after the course.*