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.

Architecture – the network architecture you’ve chosen (e.g. client-server, peer-to-peer)

a

Protocols – the application-layer protocol(s) that you’ve designed, and the transport-layer

protocol(s) you’ve chosen

a

API – the network API you’ve chosen, and any comments on its suitability for your application

a

Prediction – the prediction and/or interpolation techniques you’ve used

a

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?)

a