

LEVEL 6 ASSESSMENT SPECIFICATION

Student name:	James Craddock-Jones	Student ID number:	256515
Programme:	BSc Software Engineering		
Module:	Socket Programming and Concurrency		
Module code:	ACCA6028	Contribution to Overall Module Assessment (%):	100%
Lecturer:	Andrew Green	Internal Verifier:	John Williams
Assignment Title:	Socket programming report and implementation.	Word count (or equivalent):	3000 1000 + implementation
Submission deadline:	20/5/2022	Return date of provisional marks & written feedback:	10/6/2022
Submission method:	All written assessments, where practical and possible, must be submitted via Turnitin unless otherwise instructed by the Lecturer. (Please DO NOT put this assessment specification into Turnitin or it will match many similarities with other students' submissions.) Alternative submission method (if applicable): Late submission of the assessment will result in a late penalty mark. Penalties for late submission: Up to one week late, maximum mark of 40%. Over one week late, 0%. Only the Extenuating Circumstances Panel may approve a change to submission dates.		
Academic honesty / referencing:	Academic honesty is required. In the main body of your submission you must give credit to authors on whose research and ideas your work is based. Append to your submission a reference list that indicates the books, articles, etc. that you have used, cited or quoted in order to complete this assessment.		

**Module Learning Outcomes
(from module syllabus)**

- Critically appraise the major concepts, principles and practice to design web services.
- Demonstrate a comprehensive knowledge and understanding of complex client/server concepts, principles and tools to build, test, debug and critically evaluate software components in scalable, distributed applications.

TASK DESCRIPTION

Task 1 – report (50%)

You are to write a research-driven essay of 3000 words which explores the following topics:

- How the Transmission Control Protocol operates and endeavours to guarantee delivery of data. Refer to the three-way handshake and so-called “Two Generals’ Problem”.
- The impact of the maximum transmission unit (MTU) size on the transfer of data. You should consider associated factors such as the size of the protocol data unit (PDU), their relative positions in the OSI model and the practicalities of IP fragmentation.
- Synchronicity and asynchronicity in network protocols. Consider the sorts of applications you might use with each. Consider I/O dispatch models.
- The meaning of interoperability for networked applications and how it is best achieved.

Consider how these are relevant to us as programmers and how they may impact on our design decisions for development of networked applications.

Task 2 – implementation (50%)

You are to develop a networked application using sockets, to build a networked application of your choosing. You may pick the application you wish to implement, however should be careful that:

- 1) It is of an appropriate scope. You may wish to discuss your planned implementation with your module lecturer before you start.
- 2) You are not relying too heavily on resources you yourself did not create. It is acceptable to use external sources (including those developed in class) with attribution, however you will gain very few marks for doing so and must show significant independent development.

This task is intentionally open-ended – you may make the specific decisions about the application you wish to build, how it is to be implemented, the precise features you will include and what external protocols it might support.

Finally, you should include a written report of 1000 words which explains the features and design decisions of your application, including how it adheres to best practice.

Below are three example scenarios to give you an idea of the anticipated scale of the application. You may select one of these 3 if you'd like or can pick something entirely different.

Example 1: Chat Application

Construction of a chat application (e.g. [IRC](#)).

This should be an application which supports multiple simultaneous users and avoids I/O blocking through multithreading. You must successfully incorporate low-level data transport through sockets.

This should include some form of interface, but this may be a web frontend, console application, Windows form or similar. The precise features to be included are up to you, but as a starting point, it would be wise to consider how a user can:

- 1) Join the server.
- 2) Send a message.
- 3) Leave the server.

There is significant scope for extended functionality, such as setting a nickname, viewing users connected to the server, including images or emojis (though this will be interesting if your frontend is a console application), sending private messages, etc.

Example 2: Web Server

Construction of a web server, e.g. [Apache](#) or [Nginx](#).

To construct a web server, you will need to investigate an existing protocol – http. You will need to consider how you can respond to the requests a http client (i.e. a web browser) might make. This should include the ability to transfer a html file as part of the body of a http response.

This server should be multithreaded and support an appropriate i/o dispatch model to allow the multiple simultaneous connections.

Much of the challenge with this sort of application is that you are heavily constrained by the existing protocols. Be careful to analyse them carefully in creating your implementation.

Example 3: Multiplayer Game

Construction of a networked multiplayer game, implementing the frontend game of your choosing. This might be something turn-based (likely easier) or something real-time (likely harder).

Through the interoperability principles we've discussed during this module, you may create a socket server which uses a frontend that isn't necessarily using the same environment. This might be console-driven, web-based, or uses a game engine (e.g. Unity).

Build a game which has more than one player, be it a text-based adventure game, a digital form of a boardgame, a real-time strategy game or an action game. You should carefully consider the right communication protocol for such a game (e.g. TCP vs UDP) and how you structure the data which is sent and received.

GUIDANCE FOR STUDENTS IN THE COMPLETION OF TASKS

NOTE: The guidance offered below is linked to the five generic assessment criteria overleaf.

1. Engagement with Literature Skills

Your work must be informed and supported by scholarly material that is *relevant* to and *focused* on the task(s) set; you should make use of scholarly reviews and primary sources, where appropriate (for example, refereed research articles and/or original materials appropriate to the discipline). You should provide evidence that you have accessed a wide *range* of sources, which may be academic, governmental and industrial; these sources may include academic journal articles, textbooks, current news articles, organisational documents, and websites. You should consider the *credibility* of your sources; academic journals are normally highly credible sources while websites require careful consideration/selection and should be used sparingly. Any sources you use should be current and up-to-date, mostly published within the last five years or so, though seminal/important works in the field may be older. You must provide *evidence* of your research/own reading throughout your work, using a suitable referencing system, including in-text citations in the main body of your work and a reference list at the end of your work.

Guidance specific to this assessment:

Task 1: Breadth of evidence of suitable and authoritative reading beyond that specified in the unit materials. Correct use of Vancouver referencing.

2. Knowledge and Understanding Skills

At level 6, you should be able to demonstrate coherent and detailed knowledge and a systematic understanding of the subject area, at least some of which is informed by the latest research and/or advanced scholarship within the discipline. You should be aware of the uncertainty, ambiguity and limits of knowledge. Your work must demonstrate the growing extent of your knowledge and systematic understanding of concepts and underlying principles associated with the subject area. *Knowledge* relates to the facts, information and skills you have acquired through your learning. You demonstrate your *understanding* by interpreting the meaning of the facts and information (knowledge). This means that you need to select and include in your work the concepts, techniques, models, theories, etc. appropriate to the task(s) set. You should be able to explain the theories, concepts, etc. meaningfully to show your understanding. Your mark/grade will also depend upon the *extent* to which you demonstrate your knowledge and understanding; ideally each should be complete and detailed, with comprehensive coverage.

Guidance specific to this assessment:

Task 1 and 2: Evidence of understanding of key concepts and technologies involved in this area. Evidence to be demonstrated through a comprehensive report and a suitable development project.

3. Cognitive and Intellectual Skills

You should be able to: critically evaluate evidence, arguments, assumptions, abstract concepts and data some of which are at the forefront of a discipline (and that may be incomplete) to devise and sustain arguments, to make judgements and/or solve problems; describe and comment upon particular aspects of current research, or equivalent advanced scholarship, in the discipline. Your work must contain evidence of logical, analytical thinking, evaluation and synthesis. For example, to examine and break information down into parts, make inferences, compile, compare and contrast information. This means not just describing what! But also justifying: Why? How? When? Who? Where? At what cost? At all times, you must provide justification for your arguments and judgements. Evidence that you have reflected upon the ideas of others within the subject area is crucial to you providing a reasoned and informed debate within your work. Furthermore, you should provide evidence that you are able to make sound judgements and convincing arguments using data and concepts. Sound, valid conclusions are necessary and must be derived from the content of your work. Where relevant, alternative solutions and recommendations may be proposed.

Guidance specific to this assessment:

Task 1: Evidence of your ability to evaluate sources of information and form your own well-reasoned responses.

Task 2: Evidence of your ability to develop the concepts covered into a wholly or partly original development project.

4. Practical Skills

At level 6, you should be able to apply the methods and techniques that you have learned to review, consolidate, extend and apply your knowledge and understanding, and to initiate and carry out projects. You will deploy accurately established techniques of analysis and enquiry relevant to the discipline, and apply them in complex and unpredictable contexts, to devise and sustain arguments and/or to solve problems. You should be able to frame appropriate questions to achieve a solution - or identify a range of solutions. You should be able to demonstrate how the subject-related concepts and ideas relate to real world situations and/or a particular context. How do they work in practice? You will deploy models, methods, techniques, and/or theories, in that context or circumstances, to assess current situations, perhaps to formulate plans or solutions to solve problems, or to create artefacts, some of which may be innovative and creative. This is likely to involve, for instance, the use of real world artefacts, examples and cases, the application of a model within an organisation and/or benchmarking one theory or organisation against others based on stated criteria. You should show awareness of the limitations of concepts and theories when applied in particular contexts.

Guidance specific to this assessment:

Task 2: Evidence of your ability to utilise the concepts, techniques and methods covered along with any skills already developed to create a suitably original and functional development project.

5. Transferable Skills for Life and Professional Practice

Your work must provide evidence of the qualities and transferable skills necessary for graduate-level employment requiring the exercise of initiative and personal responsibility and decision-making in complex and unpredictable circumstances. This includes demonstrating: the learning ability for professional development to advance existing skills and acquire new competences of a professional nature that will enable you to assume significant responsibility within organisations; that you can initiate and complete tasks and procedures, whether individually and/or collaboratively; that you can use appropriate media to effectively communicate information, arguments and analysis in a variety of forms to specialist and non-specialist audiences; fluency of expression; clarity and effectiveness in presentation and organisation. Work should be coherent and well-structured in presentation and organisation.

Guidance specific to this assessment:

Task 1: Evidence of your ability to produce a well-formed and well-presented report which progresses logically and draws appropriate conclusions.

Task 2: A well-presented development project with use of consistent naming conventions, coding style, indentation and fully commented source code.

STUDENT FEEDBACK FORM

This section details the *extent* to which the assessment criteria are demonstrated by you, which in turn determines your mark. The marks available for each category of skill are shown. Lecturers will use the space provided to comment on the achievement of the task(s), including those areas in which you have performed well and areas that would benefit from development/improvement.

Task 1:

Generic Assessment Criteria	Marks available	Marks awarded
1. Engagement with Literature Skills		
	20	
2. Knowledge and Understanding Skills		
	30	
3. Cognitive and Intellectual Skills		
	30	
4. Practical Application Skills		
	0	
5. Transferable Skills for Life and Professional Practice		
	20	

Assessment Mark (Assessment marks are subject to ratification at the Exam Board. These comments and marks are to give feedback on module work and are for guidance only until they are confirmed.)	Late Submission Penalties (tick if appropriate)		%
	Up to 1 week late (40% Max)		
	Over 1 week late (0%)		

Task 2:

Generic Assessment Criteria	Marks available	Marks awarded
1. Engagement with Literature Skills		
	0	
2. Knowledge and Understanding Skills		
	10	
3. Cognitive and Intellectual Skills		
	10	
4. Practical Application Skills		
	70	
5. Transferable Skills for Life and Professional Practice		
	10	

Assessment Mark (Assessment marks are subject to ratification at the Exam Board. These comments and marks are to give feedback on module work and are for guidance only until they are confirmed.)	Late Submission Penalties (tick if appropriate)		%
	Up to 1 week late (40% Max)		
	Over 1 week late (0%)		

GENERIC ASSESSMENT CRITERIA

Level 6

In accordance with the Framework for Higher Education Qualifications, at the end of Level 6 students should have coherent and detailed knowledge and a systematic understanding of their subject area, at least some of which is informed by the latest research and/or advanced scholarship within the discipline. They will be able to accurately deploy established techniques of analysis and enquiry within a discipline, using their conceptual understanding to devise and sustain arguments and/or to solve problems. They should be aware of the uncertainty, ambiguity and limits of knowledge. They should be able to critically evaluate evidence, arguments, assumptions, abstract concepts and data (that may be incomplete), to make judgements, and to frame appropriate questions to achieve a solution - or identify a range of solutions. They will apply the methods and techniques that they have learned to review, consolidate, extend and apply their knowledge and understanding, and to initiate and carry out projects. They will have the ability to manage their own learning, and to make use of scholarly reviews and primary sources (for example, refereed research articles and/or original materials appropriate to the discipline). They will demonstrate the qualities and transferable skills necessary for employment requiring: the exercise of initiative and personal responsibility; decision-making in complex and unpredictable contexts; the learning ability needed to undertake appropriate further training of a professional or equivalent nature.

Level 6	FAIL	MARGINAL FAIL	SATISFACTORY (3 rd / Pass)	GOOD (2.2 / Pass)	VERY GOOD (2.1 / Merit)	EXCELLENT (1 st / Distinction)	EXCEPTIONAL (1 st / Distinction)
Category	0-29%	30-39%	40-49%	50-59%	60-69%	70-84%	85-100%
Engagement with literature (including reading, referencing, academic conventions and academic honesty)	Little or no evidence of reading and/or reliance on inappropriate sources. Views and findings mostly unsupported and non-authoritative. Referencing conventions used incoherently or largely absent.	Poor engagement with essential reading. No evidence of wider reading. Reliance on inappropriate sources, and/or indiscriminate use of sources. Heavily reliant on information gained through class contact. Inconsistent and weak use of referencing.	Engagement with a limited range of mostly relevant and credible sources. Some omissions and minor errors. Referencing conventions evident though not always applied accurately or consistently.	Engagement with an appropriate range of research-informed literature, including sources retrieved independently. Some over-reliance on texts. Referencing may show minor inaccuracies or inconsistencies.	Engagement with a wide range of research-informed literature, including sources retrieved independently. Selection of relevant and credible sources. Very good use of referencing, with no/very few inaccuracies or inconsistencies.	Engagement with an extensive range of relevant and credible literature, informed by the latest research. Consistently accurate application of referencing.	Exceptional engagement with an extensive range of relevant and credible literature, informed by the latest research. High-level referencing skills consistently and professionally applied.
Knowledge and understanding (Coherent and detailed)	Major gaps in knowledge and systematic understanding of the subject	Gaps in knowledge, with only superficial systematic understanding.	Limited knowledge and systematic understanding of the relevant	Knowledge is reasonably detailed, accurate with a good systematic	Knowledge is reasonably extensive coherent and detailed.	Excellent coherent and detailed knowledge and systematic	Exceptionally coherent and detailed knowledge and systematic

Level 6	FAIL	MARGINAL FAIL	SATISFACTORY (3 rd / Pass)	GOOD (2.2 / Pass)	VERY GOOD (2.1 / Merit)	EXCELLENT (1 st / Distinction)
Category	0-29%	30-39%	40-49%	50-59%	60-69%	70-79%
knowledge and systematic understanding of the subject area, at least some of which is informed by the latest research and/or advanced scholarship within the discipline.)	matter. Substantial inaccuracies. No awareness of knowledge of the latest research and/or advanced scholarship within the discipline.	Some significant inaccuracies and/or irrelevant material. No awareness of knowledge of the latest research and/or advanced scholarship within the discipline.	concepts and principles within the subject area which to some limited extent, is informed by current research and scholarship.	understanding of the field of study and to some extent, current research and scholarship.	Exhibits very good understanding of the breadth and depth of established views, and the work is, at least in part, well-informed by current research and scholarship.	understands the principles and the current and scholarly views. Clear awareness of challenges and established views and limitations of knowledge.
Cognitive and intellectual skills (Conceptual and critical thinking, analysis, synthesis and evaluation of research, assumptions, abstract concepts and data (that may be incomplete); logic, argument and judgement.)	Wholly or almost wholly descriptive work. Little or no analysis, synthesis or evaluation. Failure to develop arguments, leading to illogical or invalid judgements. Unsubstantiated generalisations, made without use of any credible evidence.	Largely descriptive work, with superficial use of critical evaluation. Weak development of arguments and judgements. Information accepted uncritically, uses generalised statements made with scant evidence and unsubstantiated opinions. Ideas sometimes illogical and contradictory.	Limited attempt at critical thinking, analysis, synthesis and evaluation, tending towards description. Some evidence to support emerging arguments and judgements but these may be underdeveloped or with a little inconsistency / mis-interpretation. Asserts rather than argues a case.	Some critical thinking, analysis, synthesis and evaluation. Can analyse new and/or abstract concepts and data without guidance. An emerging awareness of different stances and ability to use evidence (that may be incomplete) to support the argument. Mostly valid arguments and logical judgements. Some tendency to assert/state opinion rather than argue on	Sound critical thinking, analysis, synthesis and evaluation demonstrating critical thinking. Ability to devise and sustain persuasive arguments, and to review the reliability, validity and significance of evidence (that may be incomplete) to make mostly appropriate and valid judgements.	Excellent critical thinking, analysis, synthesis and evaluation demonstrating critical thinking. Ability to devise and sustain persuasive arguments, and to review the reliability, validity and significance of evidence (that may be incomplete) to make mostly appropriate and valid judgements. Some evidence of independent thought and ability to go beyond questions suggested by the text to grasp broader issues and wider context.

Level 6	FAIL	MARGINAL FAIL	SATISFACTORY (3 rd / Pass)	GOOD (2.2 / Pass)	VERY GOOD (2.1 / Merit)	EXCELLENT (1 st / Distinction)
Category	0-29%	30-39%	40-49%	50-59%	60-69%	70-79%
				the basis of reason and evidence.		
Practical skills (Apply/deploy accurately established tools and techniques; initiate and carry out projects; formulate solutions to solve problems in complex and unpredictable contexts.)	Limited or no use of methods, materials, tools and/or techniques. Little or no appreciation of the context of the application. Limited understanding of the application of theory to practice or making appropriate links between the two. Very weak problem-solving skills in complex and unpredictable contexts.	Rudimentary application of methods, materials, tools and/or techniques but without consideration and competence. Flawed appreciation of the context of the application. Weak understanding of the application of theory to practice, with only occasional evidence of making appropriate links between the two. Weak problem-solving skills in complex and unpredictable contexts.	An adequate awareness and mostly appropriate application of well-established methods, materials, tools and/or techniques. Basic appreciation of the context of the application. Theoretical knowledge and understanding applied in practice, but not always making logical links between the two. Can identify problems and propose basic solutions in complex and unpredictable contexts without fully appreciating the complexity.	A good and appropriate application of standard methods, materials, tools and/or techniques. Clear appreciation of the context of the application. Mainly consistent, accurate and logical application of theory to practice, making appropriate links between the two. Can identify problems and propose mostly appropriate solutions in complex and unpredictable contexts.	A very good application of a range of methods, materials, tools and/or techniques. Very good consideration of the context of the application, with perceptive insights. Can identify problems and propose appropriate solutions in complex and unpredictable contexts. Evidence of some innovation and creativity.	An advanced application of a range of methods, materials, tools and/or techniques. The context of the application is well considered and insightfully applied. Appropriate deployment of theory to extend understanding of established conventions. Can identify complex problems and propose excellent solutions. Innovative and creative solutions evident.
Transferable skills for life and professional practice (Exercise of initiative and	Communication medium is inappropriate or misapplied. Work is poorly structured, disorganised	Communication medium is poorly designed and/or not suitable for the audience. Work is poorly presented in a	Can communicate in a suitable medium but with some room for improvement.	Can communicate effectively in a suitable format, but may have minor errors.	Can communicate well, confidently and consistently in a suitable format.	Can communicate professionally, confidently and consistently in a suitable

Level 6	FAIL	MARGINAL FAIL	SATISFACTORY (3 rd / Pass)	GOOD (2.2 / Pass)	VERY GOOD (2.1 / Merit)	EXCELLENT (1 st / Distinction)
Category	0-29%	30-39%	40-49%	50-59%	60-69%	70-79%
personal responsibility; professional development; initiate and complete tasks and procedures: individually and/or collaboratively; use appropriate media to communicate effectively; fluency of expression; clarity and effectiveness in presentation and organisation.)	and/or confusingly expressed. Very weak use of language and/or very inappropriate style. Little or no evidence of autonomy (or collaboration, where relevant) in the completion of tasks. Little or no evidence of the skills required in graduate employment.	disjointed manner. It is loosely, and at times incoherently, structured, with information and ideas often poorly expressed. Weak use of language and/or inappropriate style. Weak independent initiative (or collaboration, if relevant). Limited evidence of the skills required in graduate employment.	Mostly ordered presentation and structure in which relevant ideas / concepts are reasonably expressed. Work may lack coherence in places. Can work as part of a team, but with limited involvement in group activities. Demonstrates the basic skills required in graduate employment, with some areas of minor weakness.	Mostly coherent, organised work, in a suitable structure and is for the most part clearly expressed. Can work effectively independently and/or as part of a team, with clear contribution to group activities. Demonstrates the skills required in graduate employment, with some areas of strength and some of minor weakness.	Work is coherent, fluent, well-structured and organised. Can work very well autonomously and/or as part of a team, with very good contribution to group activities. Demonstrates very good graduate employment skills, with just occasional minor weakness.	Work is coherent, fluent and well-structured. Can work very well autonomously and/or as part of a team, with very good contribution to group activities. Demonstrates excellent graduate employment skills and further development.

THIS DOCUMENT IS ALSO AVAILABLE IN WELSH

Assignment 1

-

Socket Programming and Concurrency

James Craddock-Jones

256515

20/05/2022

Contents

Task 1: Search Report.....	16
What is Socket Programming?	16
What is TCP/IP?	17
TCP/IP Layers	18
What is the Three-Way Handshake?.....	20
Three Steps of the Three-Way Handshake.....	21
The Two Generals' Problem	21
What Are Packets?	22
Why Use Packets?	23
Packet Structure	23
Maximum Transmission Unit	24
MPU and Fragmentation	25
What is the Protocol Data Unit?.....	25
Encapsulating a PDU	26
Asynchronous and Multithreading.....	26
Asynchronous Programming.....	26
Multithread Programming.....	27
Interoperability in Networking	28
What is Interoperability.....	28
Why Interoperability is Important.....	28
Task 2: Application Development	29
Intended Design	29
Introduction.....	29
Mock-Ups	29
Assets.....	29
Game Flow	32

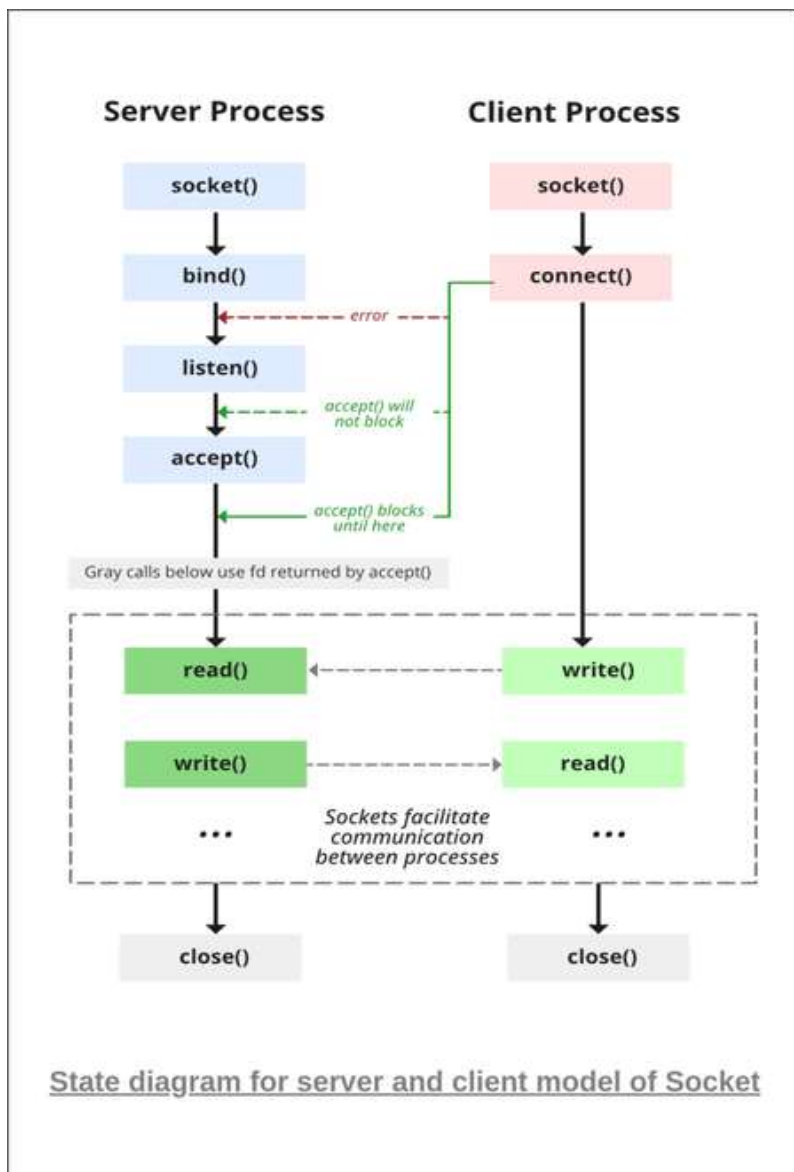
Game Network Design	33
Game Maintenance, Extension, and Best Practices.....	33
Final Design.....	33
Demo	33
Game Maintenance, Extension, and Best Practices.....	34
Pitfalls and Drawbacks	36
References	36

Task 1: Search Report

What is Socket Programming?

Socket programming means communicating among multiple nodes connected to the network. While one socket listens on a given port associated with a particular IP address, the other socket connects. Client applications connect to the server, and the server then establishes a listener socket to monitor incoming and outgoing communications. (1)

The diagram below shows how this works with the flow of data transferred between the client and server and how the data is processed.



(1)

What is TCP/IP?

TCP/IP stands for Transmission Control Protocol/Internet Protocol and is called after its two most significant elements. TCP/IP consists of 5 layers. These are the physical layer, the data link layer, the IP layer, TCP/UDP layer, and finally, the application layer. (2)

Sending data over TCP/IP is like sending somebody a message printed on a completed jigsaw piece through the post. The message has been written down, and the jigsaw has been dismantled. Each jigsaw component can then take a separate postal

route, with some taking longer than others. They could be out of sequence when the jigsaw pieces reach their final location after journeying along their numerous paths. TCP/IP ensures that the jigsaw pieces reach their destination. (3)

TCP/IP can be compared to a jigsaw assembler on the endpoint whose job is to arrange the jigsaw pieces correctly, request that missing parts be resent to the endpoint, and inform the transmitter that the jigsaw has been solved. From the very beginning, the first jigsaw piece is transmitted until the endpoint receives the last bit. TCP maintains a link with the transmitter. (3)

Internet Protocol is connectionless. Each chunk of data is identified and transmitted separately from the sender to the targeted system, with no recognition from the destination. (3)

Internet Protocol

Internet Protocol is a network-layer protocol that supports transport protocols such as TCP and UDP and delivers packet data services between programmes. (2)

Transport Control Protocol

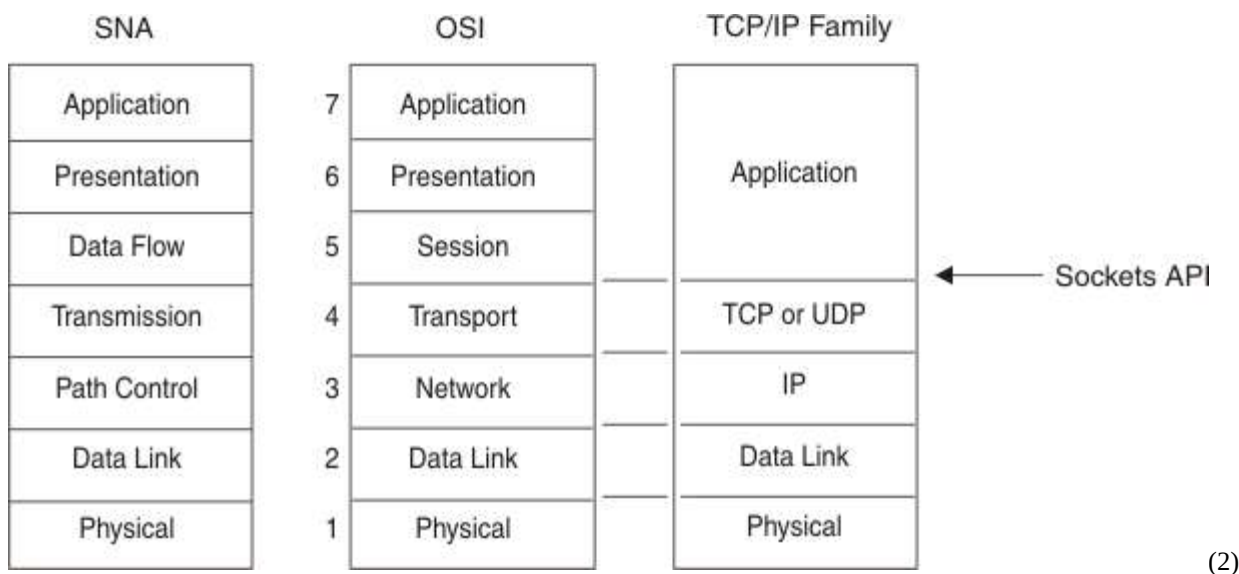
TCP is a protocol for the transport layer. Before data transfer begins, it creates a solid digital relationship among apps. Data using this protocol is given without misjudgements or repeats, while being recovered in the same sequence as it was sent. TCP considers data to be an infinite flow of bytes. (2)

User Datagram Protocol

A further transport-layer protocol that may be used rather than TCP is UDP, which stands for User Datagram Protocol. It links apps over a shaky connection. There is no end-to-end link. As a result, information is transmitted link by link. This protocol comes with no guarantees. Data packets may be received out of order, and data may be lost or replicated when delivered to the final destination. (2)

TCP/IP Layers

Below is a diagram that compares the various transmission protocols to one another, depicting the many layers each contains.



Application Layer

The application layer allows for the standardised flow of data between many different applications. HTTP, FTP, POP3, SMTP, and SNMP are some of the many protocols used daily within the application layer. The actual data that is used within the application layer, which is presented to the end-user, is derived from the payload. (4)

Transport Layer

The transport layer has the responsibility of ensuring network-wide end-to-end connection. TCP provides various features for host-to-host interactions, such as flow control, multiplexing, and reliability. TCP and UDP are transport protocols that are frequently used instead of TCP for specific purposes, such as videogames, where connection speed is more critical than data integrity. (4)

Network Layer

The network layer, commonly called the internet layer, is used to manage packets and connect many different networks to send data over network boundaries. Network layer protocols utilised while sending and receiving data include IP (Internet Protocol) and ICMP (Internet Control Message Protocol), which would then be used for error monitoring all through this process. (4)

Physical Layer

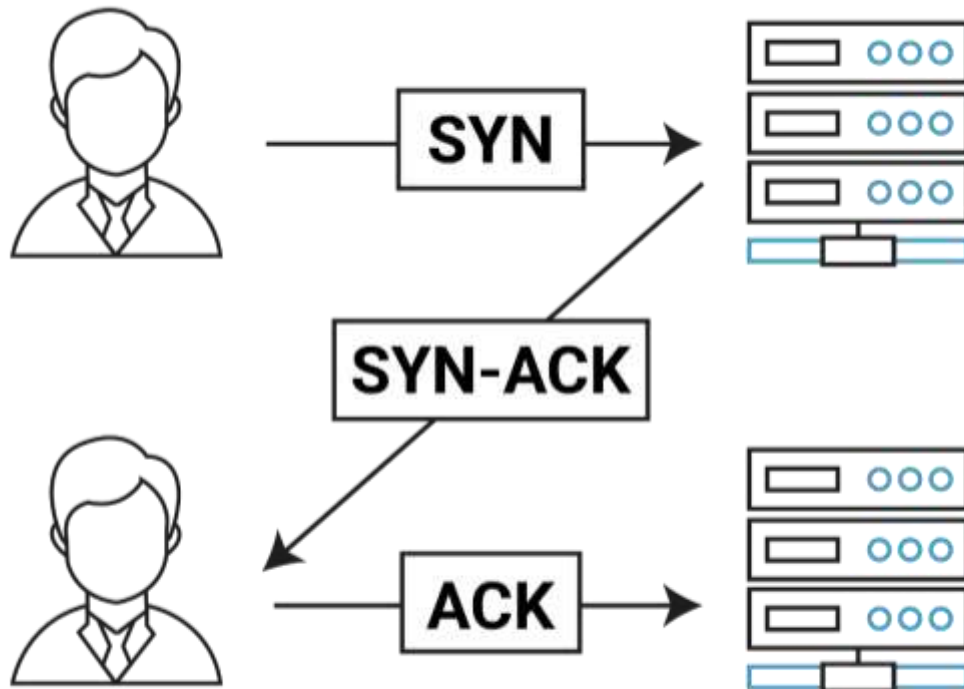
The physical layer may be often referred to as the network interface layer. This layer contains protocols that can only operate on a single connection. The network component is one of the primary components used to link network nodes or hosts. (4)
Ethernet is one example protocol that uses this layer to connect to other devices like switches or routers to computers to allow those computers to access the local network and extend into the wider internet. (4)

What is the Three-Way Handshake?

The three-way handshake is a method of communication between a server and a client used by the TCP/IP protocol. This involves three steps, as the name suggests. These include initiating a connection between a client and server and negotiating any parameters for the connection via the TCP socket connection before either party transmits any data to one another. (10)

The three-way handshake allows for data transmission to and from a device simultaneously through the network connection. The three-way handshake must witness both the client/s and server synchronising and acknowledging via SYN and ACK packets before any data transmission communications can be established between both parties. (10)

The three-way connection is a necessary process that is taken out when connecting a client and server together. This is because this process ensures that the relationship between both parties is stable enough that data can be transmitted between each other. Any connection instabilities can be found easily by any data missing between the various SYN and ACK data packets sent between the server and client. (10)



Three Steps of the Three-Way Handshake

Step 1: the connection between both the server and client is established

The client sends an SYN data packet to the server to establish a connection between each other to transmit data to and from one another. For this to happen, the server must be open to new connections. The SYN data packet will include all the vital information for the server to communicate back to send its response as well as how the client expects data to be sent back to it. (10)

Step 2: The server receives and processes the request that was sent from the client

Whenever the server node receives SYNC packet data from the client node, it analyses it and sends back a confirmation message in the shape of an ACK packet with two separate number sequences. These number sequences are the ACK one and another SYN, which are sent back to the client. (10)

The ACK in this transaction will add an extra bit onto the end of the SYN that was sent from the client to the server. The SYNC that is being sent to the client is new altogether. (10)

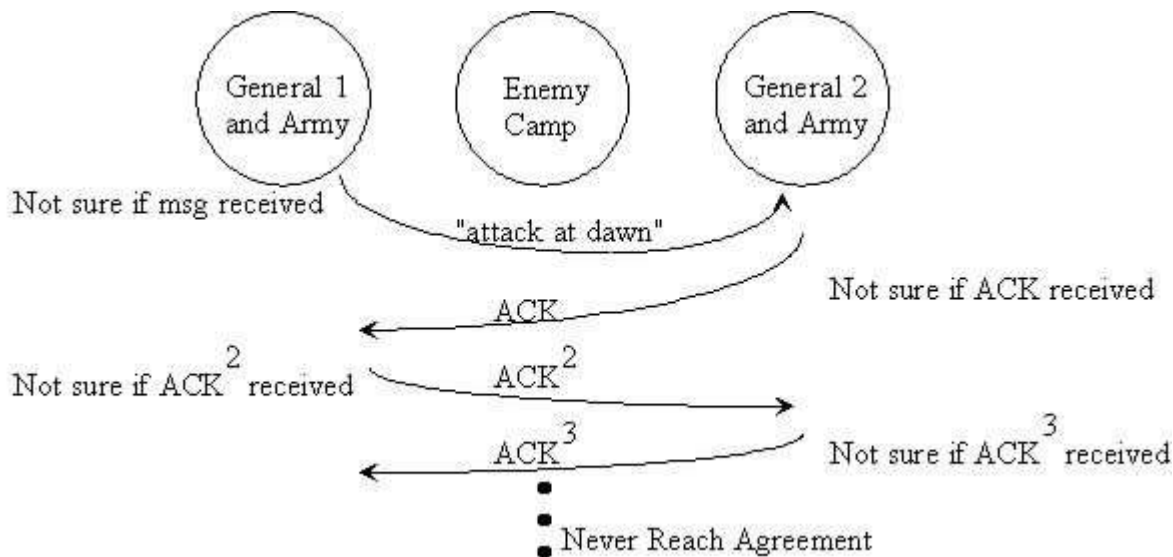
Step 3: The client receives SYN and ACK from the server

The third and final step to this handshake sees the client receiving the server's new SYN and ACK packets. The client adds a further bit to the SYN that was sent from the server like the server also did from the client. That updated SYN packet is then transmitted back to the server, and once that is done, the connection has been established between both parties. (10)

The Two Generals' Problem

The two generals' problem refers to the communication between two computers over the internet. There is no way to guarantee that communication can be made between both computers or that transmission will ensure the integrity of the data being sent through the network. Communication containing information such as packets can become lost when travelling through the various routes to the end destination. (11)

The below shows this using two different armies trying to communicate with each other while communicating past the enemy camp. No matter how many communications are sent in the form of an ACK, the sender general will never be sure if the ACK has been sent to the other general. (11)



(11)

For example, if you were to buy a game off a game store like the PlayStation Store or Steam, you will be expecting a confirmation of purchase back from the server when you press buy and expect the game to then show up in your library. If the communication is lost between the client and the server, the client or the server will not know if the game has been purchased successfully, leading to an error. This can cause the money to be still taken from the user's bank account, which can be stressful for them to get back.

There is, however, a workaround that can be somewhat made when handling the lost connection. The most common workaround is to always expect data to be lost when transferring data over the internet and use, for example, connection protocols such as TCP to mitigate loss. (11)

On the other hand, there is an almost perfect solution to this issue. Get computer A to assume how long it takes to transmit the data through the connection route and spam messages for however long it takes until computer B sends a confirmation message back. (11)

After sending the message, computer B then waits for further communication from computer A with the absence of messages being sent back from computer A allowing computer B to build up knowledge on what has been agreed by computer A. (11)

What Are Packets?

A network packet is a small broken-down component of a more extensive transmission sent over a network. Data is divided into small packets and transmitted over computer networks like a local area network or the internet. When the packets are received at the endpoint, the endpoint rebuilds the packets in their correct order back into the extensive transmission. (5)

An excellent example of this is provided by Cloudflare, which states, "Suppose Alice is writing a letter to Bob, but Bob's mail slot is only wide enough to accept envelopes the size of a small index card. Instead of writing her letter on normal paper and then trying to stuff it through the mail slot, Alice divides her letter into much shorter sections, each a few words long, and writes these sections out on index cards. She delivers the group of cards to Bob, who puts them in order to read the whole message." (5)

Why Use Packets?

Packets are not necessarily needed when sending data from one computer to another. When only two computers communicate with one another, they can transmit that data in one long line of bits to each other without issues. (5)

The issue with this comes in when communicating with many computers at one time instead of just two. Doing it this way with more than two computers starts encountering bandwidth issues. The connection speed is limited by hardware such as Ethernet cables, routers, switches, and external factors such as your ISPs provided speeds. Sending long bits of data, in this case, saturates those connections. (5)

Using packets reduces this burden as data can be broken down into more minor elements of data that can be transmitted down the most effective routes through the internet or network to the end destination. There is no order in which these must be sent, allowing multiple connections over the same networking equipment. (5)

Packet Structure

The header is the most significant portion of a packet. This is due to the number of elements it includes to ensure that all data is transferred correctly. (6)

These elements include the checksum, which is the part of the packet that detects any errors with the data transmitted, a 16bit identification number, any flags that tell hardware such as routers if the packets are intended to be fragmented, and the fragmentation offsets. In addition, the destination address, number of hops the packet has made or can make, the IP address, length of the packet, and many more such as the packet number that is used when reassembling the data. (6)

The payload includes the actual data being sent with the trailer, including further error checking and mitigation. (6)

Structure of a packet

Header	■ Internet protocol ■ Size of the header and payload ■ Source and destination IP address ■ 16 bit identification number ■ 96 bits
Payload	■ Content or data of the packet ■ 896 bits
Trailer	■ Signature of the packet ■ Error checking ■ 32 bits

©2019 TECHTARGET. ALL RIGHTS RESERVED

(6)

Maximum Transmission Unit

A maximum transmission unit is a unit of quantification used when transmitting data across the internet. It represents the biggest data packet that a networked device can handle. *“Imagine it as being like a height limit for freeway underpasses or tunnels; Cars*

and trucks that exceed the height limit cannot fit through, just as packets that exceed the MTU of a network cannot pass through that network.” (7)

Any packet that exceeds the limit of the MPU can, on the other hand, be further broken down into smaller packets to fit within the MPU limit set by the network device. MPUs use bytes as a unit of measurement. One thousand five hundred bytes is the max MPU limit that is allowed. (7)

MPU and Fragmentation

When two computer devices establish a link and communicate data, the data is routed over numerous networks. Not only the MPU limit of all devices in-between the two computers must be considered. Fragmented packets are those that surpass the MPU anywhere along the network route that the packets will travel across. It is a mix of control and user data described in a specific OSI model layer. (7)

What is the Protocol Data Unit?

Protocol Data Units are the units of information sent across network connections containing control information, address information, or data transmitted from one computer to another. (8)

In the OSI model, the PDU takes many forms:

layer 1: the PSU is just a bit of data ---> **Layer 2:** the PDU is now a frame of data ---> **Layer 3:** the PDU is now a packet that will be transmitted over the Internet ---> **Layer 4:** the PDU is now a segment of data if using TCP and a datagram if using UDP. (8) (9)

After Layer 4, the DPU is the data used within those layers, such as the application and presentation layers. (9)

GeeksforGeeks

PDU Name

Data
Data
Data
Segments
Packets
Frames
Bits

OSI Model Layers

Application Layer
Presentation Layer
Session Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer

(9)

Encapsulating a PDU

Encapsulating a PDU involves wrapping one PDU within another PDU. This happens when one PDU is sent lower than another PDU. For example, if that PDU was sent in the data field. A way of looking at it is if a PDU is acquired from a higher protocol stack and includes information containing pertinent lower layer header information. This data has been readied for transportation to the network layer. The PDU indicates its status as data moves from one plane to another. (9)

Asynchronous and Multithreading

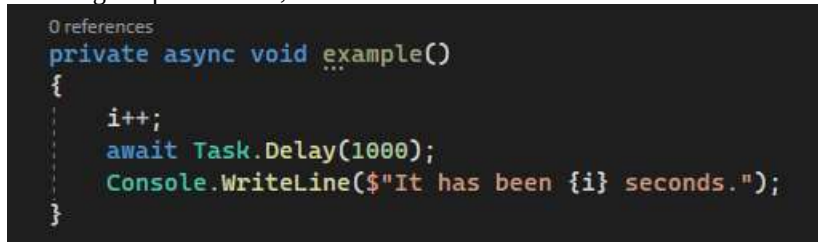
Asynchronous Programming

Asynchronous programming is the process of parallelisation in which a sequence of instructions runs separately from the primary programming process. (12)

Suppose a stalling action were to occur in the programme, but we wish to proceed with the processing without waiting for a response. In that case, we utilise asynchronous programming to work around this issue. It enables programmers to design processes that can execute concurrently without waiting for other processes to be completed. (12)

How to use Asynchronies Programming

Within C#, asynchronous programming is done by using `async` and `await` functions. For example, you may have set up a private void that you wish to become asynchronous. This can be done by using the `await` function within the void and using `async` when declaring the private void, as shown below.



```
0 references
private async void example()
{
    i++;
    await Task.Delay(1000);
    Console.WriteLine($"It has been {i} seconds.");
}
```

When to use Asynchronies Programming

Asynchronous programming should be utilised while executing processes that could stall or delay the process tree across the many methods being conducted. *“When performing blocking operations between the methods, we should use asynchronous programming. In scenarios where we have a fixed thread pool or when we need vertical scalability in the application, we use asynchronous programming.”* (12)

Multithread Programming

A thread is a singular steady stream of management inside a programme. Multithreading is just a mechanism in which the microprocessor within the computer employs many threads to run numerous tasks simultaneously. (12)

Multithreading is mainly employed when programmers wish to take advantage of multi-threaded computers by having many workers operate independently from the main thread, usually the GUI thread. (12)

How to use Multithreading

Within C#, multithreading is slightly more complex compared to asynchronous programming. You must create a new thread that is linked to a task and then start that thread for the job to be processed.

```

int i = 0;

0 references
public void StartNewThread()
{
    Thread t1 = new Thread(new ThreadStart(example));
    t1.Start();
}

1 reference
public void example()
{
    id++;
    Thread.Sleep(1000);
    Console.WriteLine($"It has been {i} seconds");
}

```

To stop a running thread, you would then do `t1.Abort()`. Note that it is important to make sure a thread is thread-safe before merging a thread outcome with the main thread. For example, updating the GUI from a thread. Not making an independent thread thread-safe, results in a program crash.

When to use Multithreading

Multithreading should be used when many independent tasks are being excited at the same time. *“When we have independent functions performing independent tasks, we should use Multithreading. One good example of this will be downloading multiple files from multiple tabs in a browser. Each download will run in its own thread.”* (12)

Interoperability in Networking

What is Interoperability

Network interoperability permits connected networks to continue sending and receiving data while retaining the acceptable level of quality required by the endpoint with no adverse effect on the broadcasting and acquiring platforms. (13)

Network interoperability mainly alludes to the effective interconnectivity of telecommunication services across multi-vendor, multi-carrier interconnections under intensive or possibly even routine settings while adhering to the necessary standards and criteria. (13)

Why Interoperability is Important

In order to accomplish end-to-end communication, network interoperability is a critical process that needs to be undertaken. If the networks are more varied, the more critical it will be to guarantee that they can interconnect to enable end-to-end transmission. (13)

Many benefits are brought from using network interoperability, as stated below:

“The user benefits as he can communicate with whom he wants to with a single terminal.

The telecom operator benefits as it can select the best equipment available from different manufacturers based on best performance and prices.

The product/service manufacturer benefits as it can extend the same equipment set to different operators.” (13)

Task 2: Application Development

Intended Design

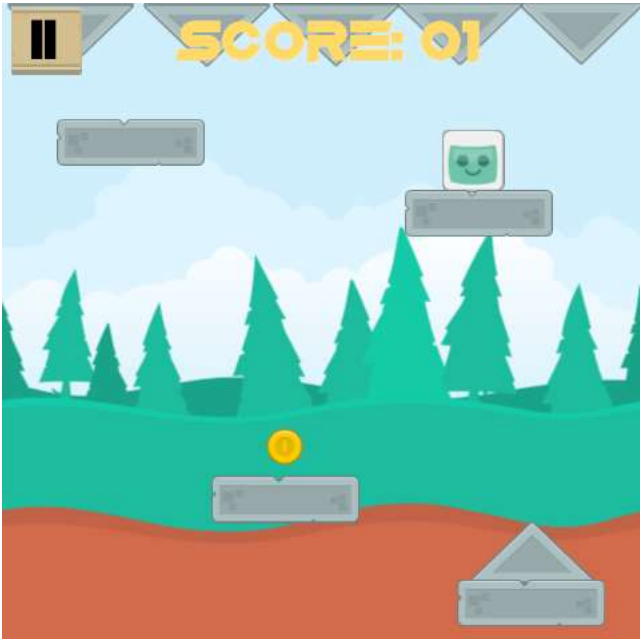
Introduction

Gravity is a hyper-casual game in which the player must travel a randomly generated terrain as far down as possible without dying by falling into the abyss, onto a spiked platform, or be transported to the top by the platforms only to be slain by even more lethal spikes.

You have been transported to a dimension that consists of a continuous flow of platforms moving up into space. To survive in this dimension, you will have to overcome complex challenges such as dodging spiked platforms, which could result in your death by falling into the void beneath the earth.

Mock-Ups

The mock-up depicts the game itself, in which the player must either fall to the next platform and maybe die or stay on the current platform and perish when it hits the spikes at the top of the screen.

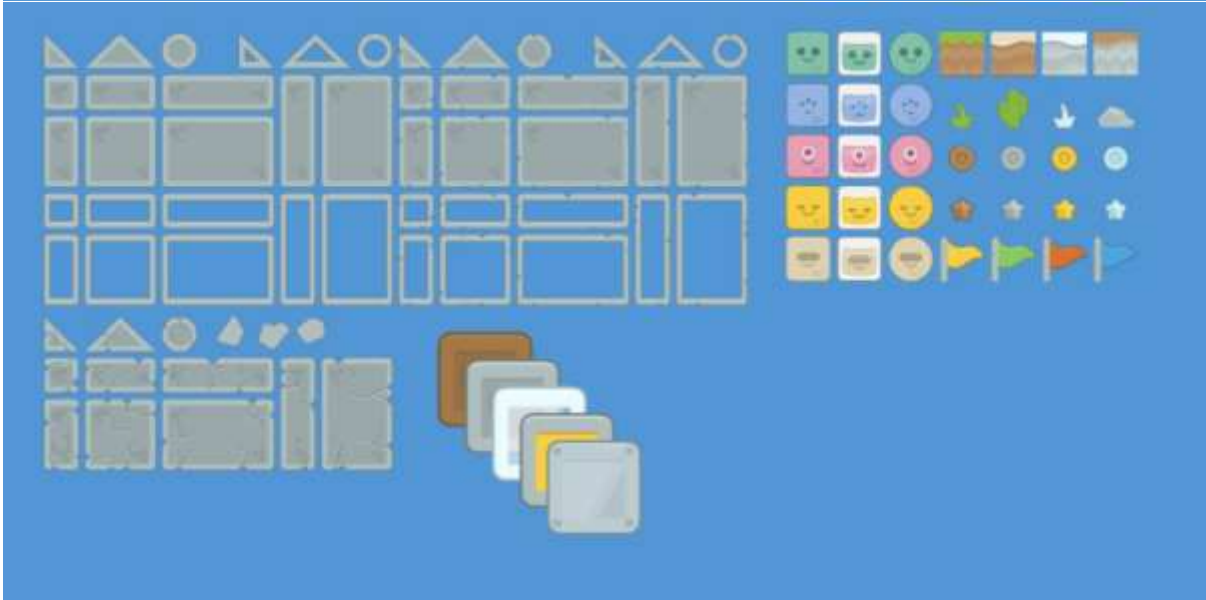


Assets

I have selected to use the assets from kenney.nl, a website that offers free copyright and license-free materials for game development.

The physics components will be used for the player sprites and some of the platforms that the players will be able to drop down into.

The RPG UI bundle will be used for the game's interface because it matches the game's design and feel.





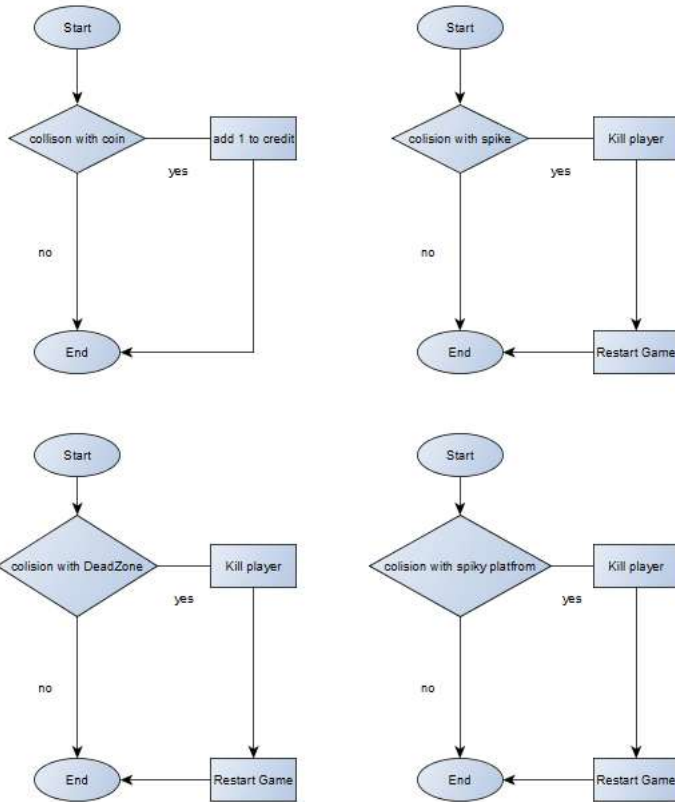
Links to assets:

<https://www.kenney.nl/assets/background-elements-redux>

<https://www.kenney.nl/assets/physics-assets>

<https://www.kenney.nl/assets/ui-pack-rpg-expansion>

Game Flow



Flowchart 1 is used to determine whether or not the player has collected a coin. If the player has, they will receive 1 point; if they do not, the game will not add any points to the score.

According to the second flowchart, the player has reached the spikes at the top of the map. If the player reaches them, the game will pause, and a text menu will show before restarting.

The third flowchart checks to determine if the player has entered a dead zone, where platforms spawn. If they go below this level, the game item will be destroyed, and the game will end when a menu appears.

According to the fourth and final flowchart, the player has plummeted onto a platform with a spike on top of it. If the player lands on a menu, a message will show for a brief period before the game restarts.

Game Network Design

The game will be using both TCP and UDP protocols to transfer data to and from the game and game server. Essential data such as user IDs will be sent over TCP to ensure data relating to such types reach the server and client 100% complete. On the other hand, movement data will be sent via UDP for the data to be sent, received, and displayed in real-time. It is also the least important data being transmitted.

The server will be developed within a console application rather than being developed in Unity like the game's client. Player movement will be handled on the client-side and sent to the server rather than the server handling movement entirely. This will be done as the game was intended to be a single player. It also ensures player gravity works as intended when falling without being simulated on the server.

Game Maintenance, Extension, and Best Practices

Commenting on the code is a fundamental process that will be done while the game is being developed. Depending on the coding language, it can be as easy as typing `/` for a single line remark or `/**` for multi-line comments.

Another way they will abide by best practices is that the files and folders will be organised logically. This allows folders to be divided into parts, such as the assets that would be kept within them, such as sprites and scripts in this case.

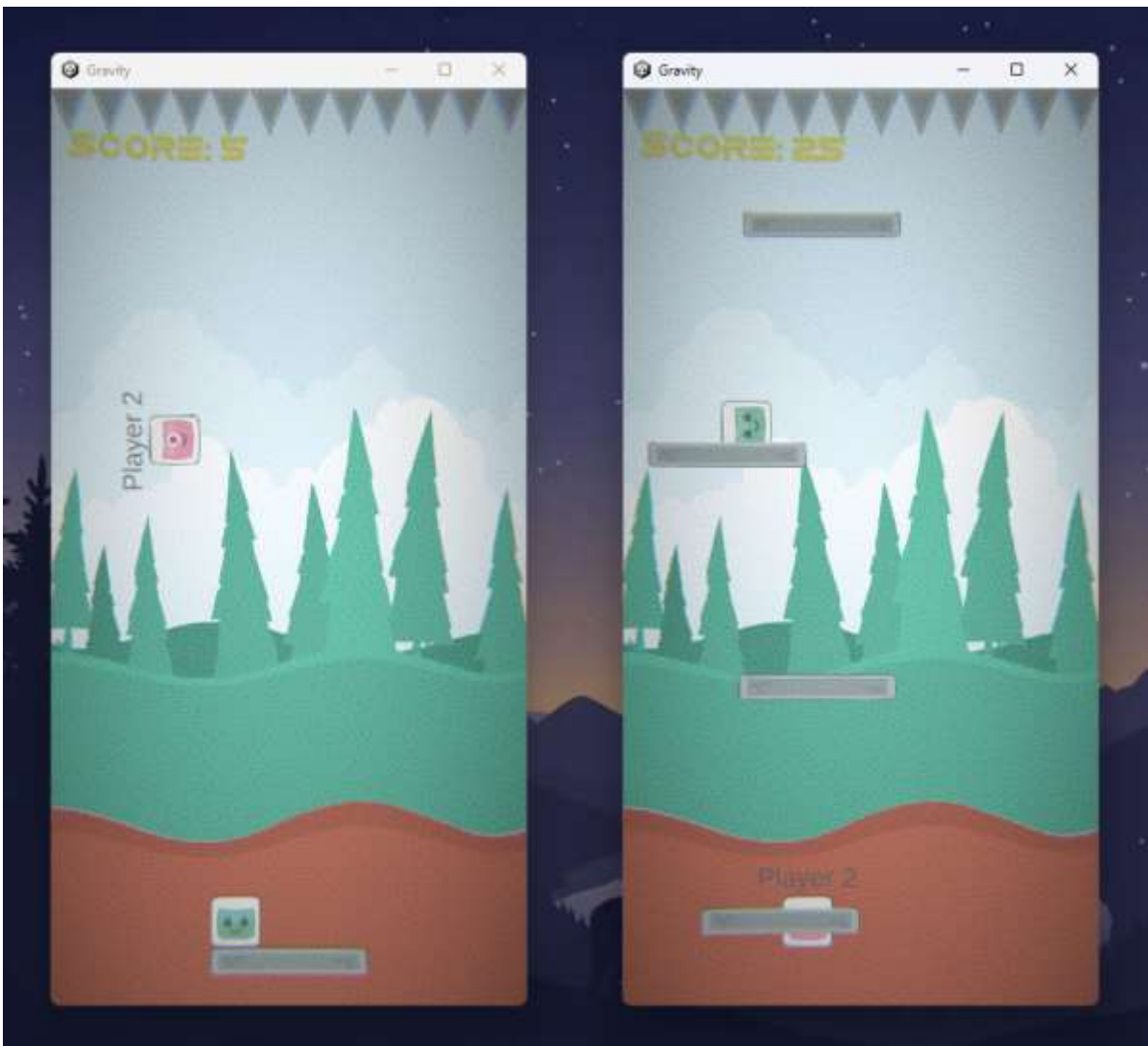
Furthermore, appropriately identifying variables will also be used to aid maintenance and withstand best practices. For example, if one was titled `moveSpeed`, it shows clearly that it is a variable that regulates the movement speed of the object to which the script is tied.

Final Design

Demo

Link to demo video: <https://youtu.be/lin08tsM-j0>

Link to code: [Development](#)



This is the final product. The right image is the game as it should be, while the left is used as a demonisation of working networking as it would be impossible to have two instances of the proper game running as the player would die straight away. The platform spawning and movement are done locally, so the players are battling to last longer than each other.

Game Maintenance, Extension, and Best Practices

The game was mostly commented throughout both the client and the server projects that have been developed. Below is an example of this with the connection and send data methods where the player data is managed when making a connection and where data will be sent to the player/s.

```

1 reference
public void Connect(TcpClient _socket)
{
    //Get socket
    socket = _socket;
    //Set buffer size
    socket.ReceiveBufferSize = dataBufferSize;
    socket.SendBufferSize = dataBufferSize;

    //Get socket stream
    stream = socket.GetStream();

    dataRecieved = new Packet();

    //Collect buffer data
    receiveBuffer = new byte[dataBufferSize];

    stream.BeginRead(receiveBuffer, 0, dataBufferSize, ReceiveCallback, null);

    ServerSend.Welcome(id, "Welcome to the server!");
}

```

3 references

```

public void SendData(Packet _packet)
{
    try
    {
        //Check is socket is empty
        if (socket != null)
        {
            //Begin stream write to send to player/s
            stream.BeginWrite(_packet.ToArray(), 0, _packet.Length(), null, null);
        }
    }
}

```

In addition, all the assets used throughout the development have been organised into folders and subfolders that are relevant and easy to understand for future development and maintenance. The images show this with the main assets folder with all the major components split into their own folder categories and subcategories.



In the following image, the scripts have been split up into their own subcategories, socket pro for the socket programming scripts, player for the player scripts, platform for the platform scripts, and MainMenu for the main menu scripts.



Pitfalls and Drawbacks

There are many drawbacks and pitfalls with the current implementation. One is that the platform spawning system is done at a local level, meaning that the game will differ for each player and vary in difficulty.

In addition, the current implementation does not handle disconnections properly and requires the whole game executable to be restarted every time the player dies for the disconnection to occur. This causes an error on the client when trying to re-join without closing and reopening the whole game.

Furthermore, the current implementation continuously spams the server and clients with player locations when not needed. A work around could be used on the local player's side to only send data when movement is detected.

References

1. GeeksforGeeks, *Socket Programming in C/C++*, [Socket Programming in C/C++ - GeeksforGeeks](#) Data accessed: 14/04/2022
2. IBM, *TCP/IP TCP, UDP, and IP protocols*, [TCP/IP TCP, UDP, and IP protocols - IBM Documentation](#) Data accessed: 14/04/2022
3. Cloudflare, *What is TCP/IP?*, [What is TCP/IP? | Cloudflare](#) Data accessed: 16/04/2022
4. M.E Shacklett, *TCP/IP*, [TCP/IP: What is TCP/IP, and How Does it Work? \(techtargert.com\)](#) Data accessed: 20/04/2022
5. Cloudflare, *What is a packet?* | *Network packet definition*, [What is a packet? | Network packet definition | Cloudflare](#) Data accessed: 22/04/2022
6. A. Zola, *network packet*, [What are network packets, and how do they work? \(techtargert.com\)](#) Data accessed: 22/04/2022
7. Cloudflare, *What is MTU (maximum transmission unit)?*, [What is MTU \(maximum transmission unit\)? | Cloudflare](#) Data accessed: 25/04/2022

8. Techopedia, *Protocol Data Unit (PDU)*, [What is a Protocol Data Unit \(PDU\)? - Definition from Techopedia](#) Data accessed: 26/04/2022
9. GeeksForGeeks, *Protocol Data Unit (PDU)*, [Protocol Data Unit \(PDU\) - GeeksforGeeks](#) Data accessed: 29/04/2022
10. N. Shalom, *Three-Way Handshake*, [What is a Three-Way Handshake? - Definition from Techopedia](#) Data accessed: 29/04/2022
11. S. Rout, *The Two Generals Problem*, [The Two Generals Problem. And why two computers can never... | by Siddharth Rout | Geek Culture | Medium](#) Data accessed: 29/04/2022
12. Code Maze, *Difference Between Asynchronous Programming and Multithreading in C#*, [Difference Between Asynchronous and Multithreading in C# \(code-maze.com\)](#) Data accessed: 10/05/2022
13. Comviva, *All You Need To Know About Network Interoperability*, [All You Need To Know About Network Interoperability - Comviva](#) Data accessed: 10/05/2022