GCE Computer Science (7517)

The Practical Project

|  |  |  |
| --- | --- | --- |
| Centre number |  | Centre name |
| 22133 | The Perse School |

|  |  |  |
| --- | --- | --- |
| Candidate’s full name |  | Candidate number |
| Abhinav Gupta | 2262 |

|  |
| --- |
| Project title |
| Travelers of Catan |

Contents

[1 Analysis *(9 marks)* 3](#_Toc451422998)

[1.1 Identification of the problem 3](#_Toc451422999)

[1.2 Research carried out 3](#_Toc451423000)

[1.3 Identification of the prospective user(s) 3](#_Toc451423001)

[1.4 Detailed background to the problem 3](#_Toc451423002)

[1.5 Numbered measurable, appropriate specific objectives of the project 3](#_Toc451423003)

[1.6 Modelling diagrams 3](#_Toc451423004)

[2 Design *(12 marks)* 4](#_Toc451423005)

[2.1 System design overview 4](#_Toc451423006)

[2.2 Algorithms 4](#_Toc451423007)

[2.3 Data structures 4](#_Toc451423008)

[2.4 File structure and organisation 4](#_Toc451423009)

[2.5 Database design 4](#_Toc451423010)

[2.6 SQL queries 4](#_Toc451423011)

[2.7 User interface design (HCI) 4](#_Toc451423012)

[2.8 Hardware selection/design 4](#_Toc451423013)

[3 Technical Solution *(42 marks)* 5](#_Toc451423014)

[3.1 Code listing 5](#_Toc451423015)

[4 System testing *(8 marks)* 6](#_Toc451423016)

[4.1 Test plan 6](#_Toc451423017)

[4.2 Annotated screenshots of test results 6](#_Toc451423018)

[5 Evaluation *(4 marks)* 7](#_Toc451423019)

[5.1 Comparison of project performance against the objectives 7](#_Toc451423020)

[5.2 Effectiveness of the solution 7](#_Toc451423021)

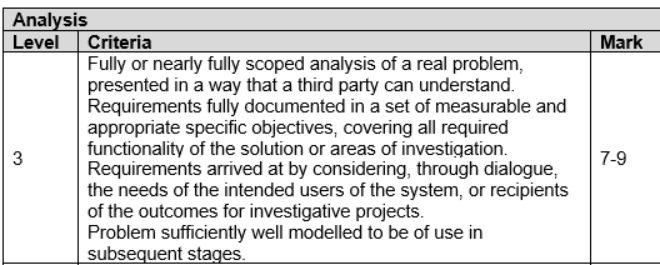
[5.3 Analysis of user feedback 7](#_Toc451423022)

[5.4 Possible improvements 7](#_Toc451423023)

[6 Appendix 8](#_Toc451423024)

[6.1 User feedback (authenticated by assessor) 8](#_Toc451423025)

# Analysis *(9 marks)*



In the analysis section we are looking for:

* A detailed description of the problem / investigation
* Clear evidence that research has been performed (dialogue)
* A clear set of objectives that will be useful across the later sections of the project
* Modelling of the proposed solution that will be of use to later design work

Having read the analysis stage:   
 Do you understand what the project is going to do?

Has the student set clear objectives that are detailed enough?

## Background to/ Identification of the problem



The problem is people want to play Settlers of Catan but don’t have enough time to either complete the game or get fully immersed in it. The game Settlers of Catan was created in 1995 in Germany, where players simulate settlers of new land, attempting to expand their individual empires. This is a turn based strategy board game between one and four players. Most notably, Catan is set on a hexagonal point-top tilemap, consisting of settlement pieces and resources. The Vanilla game rules require the player to roll …

However many of these rules could seem confusing at first which often repels newer players from immersing in this game.

Some other problems with the current design is that small road and village pieces can easily get lost, which can cause problems in longer games. In addition, dice can cause players to experience luck based events which may not be fair if they are tactically superior. There is also no way to improve at this game without carrying out independent research online.

Is this a detailed description of the background/ identification? Is there evidence of an understanding of how the current system contributes to the problem? Is there evidence that the user is well understood and the context is understood? If a game or puzzle, have you described the rules in sufficient detail along with any alternative rule options that may exist? Is there a problem related to having to carry a physical game around or not having time to play a full game in one sitting and the associated issues with packing away and restarting? If a puzzle is there an issue with getting access to enough different puzzles to become proficient? Is there any way to help become better at the game or puzzle?

Generic problems with board games

## Research carried out

You must include detailed evidence of research into the background of the project including any knowledge acquisition for the problem domain. Include screen shots or diagrams of the **current system** and interview **transcripts** with the current user. Ensure questions are **detailed** and really scope out the full complexity of the new system and show how it can solve the problem. Identify any **key algorithms** that you may need to write – either ‘complex user defined’ ones to implement the rules of your program or known algorithms e.g. Minimax. Include an analysis of how deep (how many levels) you are likely to be able to go in a minimax implementation based on your product. Does your game allow for **repeat turns** and so would minimax need to be modified from its pure form? Identify any key **data structures** that might be useful in building your solution. Are there any **research papers or web articles** that describe a **playing strategy** in your product that you will be able to implement. Note that you are not implementing the algorithm, but identifying it as ‘needing to be implemented’. Make specific links from the research of existing solutions to the project objectives. Will your product use any networking (client/server) and if so what networking technology will you use (websockets?)

Add diagram to talk about algorithms and data structures

### Board data structure

When storing the board in a data structure, a weighted undirected graph seems to be the most suitable option as it is easy to store cells that are connected together by a weight. This graph may be stored as a Dictionary of positions as keys and the cell objects as values or they may simply be stored as cell objects in a list with their positions as a public property, as it will need to be accessed and updated from the main game As for storing the board connection statuses (whether a road or wall exists and if so by which player), an adjacency list or matrix can be used. An advantage of an adjacency list is that it will take up less memory. However an adjacency matrix is very quick when looking up a specific connection between two nodes and that speed may be more important than memory if these lookups need to be made many times a second by my computer bot.

### AI algorithm options

When analysing multiplayer game computer bot algorithms, the main and most simple technique used is a Monte-Carlo Tree Search (MCTS). There are a few MCTS options that are considered to be the fastest and most optimized. The Max^n algorithm is an adaptation of the classic MiniMax algorithm that is extended to multiplayer by allowing the static evaluation of a position to be a n dimension vector. From extensive analysis it was proven that deep pruning is not possible in this algorithm due to the complexity of multiplayer game positions. This means that for the same time spent searching fewer positions get considered in comparison to other methods. An alternative method is the Paranoid method in which the computer is considered a maximising player and the other players are all minimising a single player’s score. It is shown that this approach can then undergo alpha beta pruning making it more efficient, despite it losing some precision as the best possible positions are often not searched for unless the correct minimising player is chosen. An algorithm that attempts to improve on both of these methods is the Best Reply Search algorithm. This algorithm approaches the problem in a similar way to Paranoid by classifying all other players as a single minimising player. However this time the algorithm searches moves from all opponent players on each turn, not just the moves of the current player. This means that many illegal positions can be reached which could be problematic for some games however should not affect Catan severely as positions are quickly changing. This means that BRS allows for alpha beta pruning like normal MiniMax which means it may be able to reach a depth of between 5 and 6 moves with one second of evaluation. In addition experimental data collected on ResearchGate in 2013 has shown that BRS beats the other two approaches in all 8 games used.

<https://www.researchgate.net/figure/Results-of-max-n-vs-paranoid-vs-BRS_tbl1_259655439>

### Storing files and online multiplayer

To store the board when the user is saving the game, the objects can be serialized into JSON or binary using built-in C# packages. After serialization to binary, the file can be encrypted using XOR encryption which will keep the data secure so players can not cheat by accessing the save file.

For networking, using HTTPS requests like GET and POST is an option for making a multiplayer game however this may end up being slow and therefore laggy. An alternative method would be to use web sockets which are much faster.

Things that I might have done

## Identification of the prospective user(s)

This is a short paragraph that identifies the user and describes them and their need. It should include detailed interviews with the prospective user (not superficial yes/no questions) where the user’s opinion is sought in detail. Make specific links from the user’s requirements to the objectives of the project.

### Interview

General Information

I: What annoys you about the games of the settlers of katan?

P: The fact that I can't move around the board. Like I want to be more immersed in the world and it often gets boring after ten rounds.

I: For the game and GUI, what colour schemes would you prefer to have.

P: More greens and yellows because that's less strain on eyes. The board itself should have a mix of colours depending on the resources, like lumber should be brown and metal should be silver.

I: How many players should it go up to?

P: It should be anywhere up to four players like the original game so it doesn’t become too cluttered.

General Game Rules

I: At the start of the game, where should each player’s capital city spawn?

P: Their capital city should spawn on a random location on the edge of the board to make it more fair and challenging.

I: What benefits are there from computer based games over board games?

P: There are pieces that you can lose and computer games are easier to save and come back to and play against people in a different location.

I: Would you think it would be better if you moved your entire empire or you had a piece which moves around and collects the resources?

P: I feel like it would be better to have a single piece that moves around, so that way you'd have to be able to prioritise your moves.

I: Should the gathering of materials still be based of a dice roll or since there is now a moving piece, should there be a different system?

P: There should be a new system where after rolling the dice, the resources spawn on the grid and the moving piece needs to go and collect them.

I: How long do you think each game should last and do you think there should be a time limit?

P: There shouldn’t be a time limit however I would like the games to last between ten to thirty minutes as it keeps the game

I: What new features would you specifically like the new game Travelers of Catan to do?

***Introducing the idea of the new moving piece***

P: The base game should be the same, with the same objectives. But there should definitely be more features in the game to make it more competitive and difficult, such as the ability to move around the map.

I: You mentioned moving around the map, how exactly should it work?

P: The moving piece should stop on a vertex on the hex grid so that it is next to three roads.

I: What resources should it collect from this position? All three?

P: I think so because it would also make the game go faster and the players need to strategize where the need to place their moving piece.

I: Should there be any challenges to movement?

***Introducing the new idea of walls***

P: It would be cool if you could buy walls or barriers along with villages, cities and roads. And when you place a wall the enemies can’t move past there. You shouldn’t be able to travel on the enemies territory.

I: So then should you be able to travel past the enemies villages, cities and roads.

P: Definitely not traveling on the enemies roads or through their villages and cities.

I: In that case what should walls offer instead of making a village.

P: Walls shouldn’t need to be connected to your main empire so you can place a wall around a specific recourse and get a monopoly because this would open a whole new gameplay mechanic because you can expand your empire’s territory without needing to buy loads of roads and villages.

I: How should they be implemented?

P: Make walls appear perpendicular to roads blocking all enemy travel between the two adjacent nodes.

I: When should you be able to buy a new road, city, village or wall.

P: So you should be able to make new roads, villages and cities when your piece is anywhere as long as its connected directly to your capital. The walls should only be made at the place where your piece currently is.

I: Should your own moving pieces be able to travel through your walls?

P: Yes but it should take two moves instead of one to discourage players making loads of walls.

I: Should there still be a robber piece like in Settlers of Catan?

P: There should be a Highwayman piece that you can move when you roll a 7.

I: What should it do?

P: It should sit at the junction between roads, though not on a wall, village or city and when an enemy comes past, it should take all of their resources and give it to the person who moved the Highwayman there.

I: As discussed there is currently the road, village, city and wall to buy. Should there be anything else?

P: Maybe as the game progresses you can buy another resource collector so you have two moving pieces to control

I: What should the cost be for that?

P: Maybe instead of buying it with resources to stop people from just gathering lots of material instead of progressing the game, you should unlock a second one after creating two cities so it’s more balanced between collecting and upgrading.

I: Could it instead be after getting a certain amount of victory points which then translates to getting a certain number of cities of villages.

P: Definitely that sounds better so instead of just getting two cities it could also be one city and two villages give you a new collector.

User Account System

I: Would you like to play with someone at the same computer as you or would it be nice to play remotely?

P: I mean, having options for both would be good, but if I'd have to pick one, then it would probably be same computer.

***This prompts the idea of multiplayer***.

I: What are your thoughts on having a leaderboard or a way to keep track of your progress as you play the game.

P: There should definitely be a leaderboard so I can see my skill improving and it will make the game competitive with friends.

I: Would you always want to finish a game of Travelers of Catan. What are your thoughts on saving a game’s progress?

P: Yes, I want to be able to leave it for a bit and then be able to come back.

***There should be a database to store the user’s progress and their score for the boardrbaord.***

bots, difficulty and additional features

I: Should the board generation have its own difficulty

P: I think it should generate a new board each game and there should be a separate difficulty for the board generation

I: You mentioned the board being a hex grid what should this look like?

P: The board is a grid of 25 hexagons like in the normal game with the recourse generators randomly distribute across the map.

I: What makes a board harder to play than another.

P: The resources should be more spread out so its harder to collect everything from just staying in one area of the board. There can also be fewer generators of one resource on the board so its more competitive to get. It would also make the game feel more natural and more competitive.

I: You mentioned the game should be up to four players. Should there be a one player mode or a general way to add bot opponents.

***This introduces the idea of the single player mode against the computer.***

P: Definitely, it would be fun to play with two players and two bots as well as the players can make an alliance to try and beat the bots. The bots should also have varying difficulty though.

I: What different factors should control the computer bot’s difficulty.

P: The harder bots should be able to strategize better which means planning moves ahead and knowing what resources it needs to collect next. It also should move the piece around the board in a more efficient way.

I: Should there be any way to team up in Travelers of Catan?

P: An alliance system would be pretty cool. You could then travel on the other person’s roads, villages and cities.

I: What are your thoughts on trading resources like in the original game?

P: It should definitely still be a feature.

I: What about deciding if a bot should accept or start a trade?

P: The bots shouldn’t be able to start trades as that could get annoying but if you trade with a bot it should calculate if it’s a good or equal deal and accept it.

I: What are your thoughts and talking back your moves?

P: I think that it’s probably not a good feature as it means you don’t need to think about your moves and games could get boring with people taking back their moves.

I: What factors should affect the spawn rates of a resource?

P: There should be a random chances of a recourse spawning at a location and this chances should be smaller if there is a city or village on its border.

I: Why does it need to be a smaller chance?

P: To discourage a player keeping their piece on their city and not moving.

I: Should there be any special features that occur during the gameplay?

P: Events that happen every five rounds would be cool like some resources spawning double or there being two Highwaymen.

I: How many events should there be per game?

P: They should keep happening until the game ends every five to ten rounds. Some ideas are changing the board like swapping resource regions.

## Numbered measurable, appropriate specific objectives of the project

These should cover all required functionality of the solution or areas of investigation ('appropriate' means the specific objectives are single purpose and at a level of detail that is without ambiguity)

NB These must be complex even if you are unable to code them all fully! They should be described in sufficient complexity as to meet the standard required of A level. Each user need should be identified as a measurable objective and then within that objective specific outcomes should be identified and listed. These should not include any add-ons of standard algorithms nor any code that will be generated by a GUI. This is the most important section for you to get right!!!!!!!!

### MVP

* 1. Running the program will launch the game in the command line.
  2. The game will write a map of each grid position and its randomly generate resource to the console.
  3. Two players will be instantiated in the game with an initial position. The players will then take turns entering the coordinates of the position they wish to move to in the terminal.
  4. After a player has made their move they will be given the option to make a purchase or continue.
  5. If they choose to purchase an upgrade, this will increment the players victory point count and decrease the relevant resources. However, If the player does not have enough resources to make a purchase, the program will not allow the player to complete this transaction and the menu to purchase an item or continue will be re-displayed.
  6. If the player chooses continue, the next player will get a turn.
  7. Between turns there should be an option to print the position of both players along with their victory points.
  8. After the second player has made their turn both players should receive resources based on the grid generated at the game start.
  9. After each players turn, if the player has reached 20 victory points the game will display information about the winner and loser's victory points to the console and should terminate.

### Advanced Terminal

* 1. When the game is launched, a menu should appear with the option to create a new game and close the game.
  2. If the user creates a new game, a menu will appear for the user to set the number of victory points required to win and an option to change the randomness of the generation of resources within the grid.
  3. When creating a game there will be an option to change the number of players from a default value of two to a maximum of four and there will be the option to change a player to a computer bot.
  4. This computer bot should make random moves and should always purchase anything it can afford.
  5. Once the game starts, both players will be informed the location of their starting capital city.
  6. When the player enters a new position to move to, their input will be validated and a message should be displayed if they have made and illegal move or eranous input. Once the player enters a correct input, their piece will move to this new position.
  7. Players will now be able to move through their own roads in addition to making a move and players will not be able to travel through walls built.
  8. If the player chooses to make a purchase with their resources, they will now be asked for the location of their purchased structure. Their input will be validated until there is a legal position entered.
  9. Between turns the players will be given another option to display the location of all of their structures.

### Basic GUI

* 1. The program should launch a Graphic Game which is set with 2 initial players and a randomly generated grid of resources.
  2. The Grid will be formed of 25 tesselated hexagons with an image of the randomly generated resource in the center of the hexagon. The edges between hexagons will have a small circular button element on top of them.
  3. Once it is the players turn they should now be able to click on a node and it should send them there. If the node they select is not adjacent to their current position the player should be indicated to make another move.
  4. After a player makes a move, a GUI will popup over the game asking them to make a purchase or continue. The four different things they can buy will be displayed next to a continue button.
  5. If the player continues, the GUI will close and the next player will be asked to make a move.
  6. If the player clicks on an item to purchase, their resources and victory points will be adjusted and the GUI will not close.
  7. Once either player reaches 20 victory points, a pop up over the game will display the winner and the game ends.

### AI

* 1. The player should play a single AI bot

### Advanced GUI

* 1. The program should launch a Graphic Game which opens on a home page with the option to create a game or exit.
  2. Once creating a game, a GUI popup will appear asking the user to enter the settings. The randomness input will now be a slider element.
  3. The game will load the hexagonal grid with an image of

### Accounts and Databases

We will see if it get here during Christmas or not but there should be enough without it

### Final Product

* 1. Working final product

## Modelling diagrams

E.g. E-R model, High level software object model, Data Flow Diagrams, mathematical models, the flow through the program.

General UML diagram outlining the key classes that will be required.

A screenshot of a diagram

Description automatically generated

Mathematical model of the hexagonal grid:

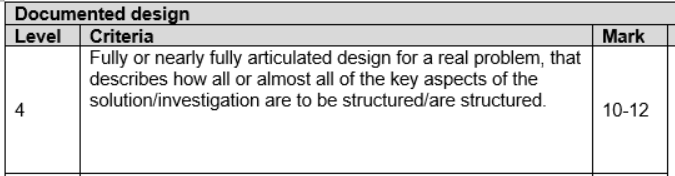
A hexagons with numbers

Description automatically generated

**Important read – Examiners Report**

Analysis There is encouragement that a student should gather details for the project from users via a dialogue of some form. Some of the interviews seen were very detailed and clearly gained relevant information for development of the project. Unfortunately, it was also common to see very short interviews which gathered no real requirements for the project to be assessed highly by centres. Students should be encouraged that in the interview it would be beneficial to ask probing questions to find out the real requirements of the user(s) and not just the kind of colours to be used or whether they like playing games. The analysis should contain a list of the objectives set by the student for their technical solution. It was pleasing to see many students provide a detailed list of objectives that indicated both the requirements to be met and the complexity that this might involve. Students who submitted vague and brief objectives would struggle to pick up high marks in the analysis section and it would also be common for the rest of the project to suffer slightly. Weak objectives also make awarding the completeness mark hard as consideration must also be placed into what an A-level student would be expected to achieve. The analysis section is to contain some modelling of the proposed system and it was pleasing to see students complete this in a variety of ways. Those projects that needed data processing usually included some discussion of the data required and DFD or ER diagrams. Students looking to produce a game sometimes struggled with the modelling section and also left the reader not understanding what their idea actually was. Students completing gaming projects could consider sketching out some ideas for the game and discussing the game flow as part of their modelling section.

# Design (12 marks)



In the documented design section we are looking for:

* An overview to the whole system design (module breakdown / objects / units / web pages …)
* Clear design & detail for some of the complex algorithms
* Detail for the data to be used (database design / data structure design)

Having read the documented design:

Do you understand how the project is going to work as a whole?

Do you have a clear understanding as to how some of the complex algorithms will work?

Do you understand how data will be processed / structured?

## System design overview

Description plus diagrams such as class diagram: For each class describe its purpose (high cohesion) and each public method (interface/low coupling)

Flowchart: Give general top-level flow of the system from running the program. May be more than one flow chart for different use cases.

Data flow diagram: Top level diagram showing how data moves between different parts of your software for different use cases e.g. logging in accesses database, saving access database.

Swim-lane diagrams for networking messages

Full UML diagram

One sentence description for every class (used in class comments)

Data flow diagram

## Data structures

The board will be stored in an Array of Node objects.

Lots of images

Different game states

Give examples of what data would be stored in the data structure in the context of your project. Do not be afraid to be verbose

Diagrams and descriptions of data structures

Include it all at different stages in the game (opening , middle game and end game)

## Algorithms

**Pseudocode** or similar for key algorithms essential to the success of the project. These MUST be linked into the how they fit into the project as a whole. Show how the algorithm would **affect** the **data structures** used in the context of your project. Give a **trace table/dry run** of sections of your key algorithms to **show** how they work.

Pseudocode for MINIMAX here:

Minimax(State, Player, PruningVector, Depth)

If (Depth == 0)

Return StaticEval(State).Normalized()

Else

Best = Vector4(0, 0, 0, 0)

For each Child in State

Value = Minimax(child, Player - 1 % 4, PruningVector, Depth-1)

Best = Max(Best, Value)

PruningVector = BetterForPlayer(Player, PruningVector, Value)

If (Value[Player] > PruningVector[Player]

and Value[Player - 1 % 4] < PruningVector[Player - 1 % 4])

Break

Return Best

StaticEval returns a vector4 containing the strength of each player in a given position by calculating their material advantage. By normalizing this vector and using the dot product to compare it with an axis, the returned angle indicates how good a position is for a specific player. Smaller angles between this vector and the axis would mean that the position is optimal for the current player as the vector is pointing mostly in their direction. From research papers published in 2015, it has been shown that the Max^n algorithm does not permit deep pruning meaning that pruning at a Depth higher than the pruning vector will not guarantee finding the optimal solution.

DJIKSTRAS TRACE:

Flow charts for anything important e.g checking for wins / legal moves etc

Make sure these algorithms match the UML diagram names

## File structure and organisation

Each game can be saved to a binary file that is encrypted using XOR encryption with a stored secret key. This means that players can not access a saved game state.

Show what files are going to be created and which software objects will be implemented in each file. Include folder structure for images, database file if used.

e.g JSON / bin serialization

## Database design

Provide schema for database. Give example data that would be stored in the database for your project.

Tables relationships

Data types

Restraints

## SQL queries

Give **all** SQL queries that would be needed to implement specific project objectives. (can be written after solution completed)

List all queries for free marks 😊(web based)

Add explanations for each one

## User interface design

Provide wire-frame diagrams of the graphical user interface, positioning of buttons, different screens. Give any menu commands that would be available in a terminal interface

Sketches made before the UI is created

A diagram of a diagram

Description automatically generated with medium confidence

## User guide

Run the executable file produced when compiling the Unity project. This may be run on the Windows operating system however Unity is capable of compiling to other platforms if necessary.

Write a user guide to say what software or modules the user will need to install or have available on their system, and how they should run your program from the command line

NOT A MANUAL

Instructions on how to start up the project ***not use it***

**Examiners report – an important read!**

It was pleasing to see good students carefully structure out the design of their technical solution. Effective use of diagrams to provide an overview of the whole system, key data requirements being identified and explained along with a breakdown of the complex parts leading to pseudo-code and/or code snippets would lead to a high mark. It was also common, however, to see a more random attempt at the design documentation including just pasting code across with no detail as to the design process or how it would link into the main system. So, for example, just providing stock algorithms for merge sort and binary search does not help the reader understand the design of the system.

Having a section titled ‘sample of SQL queries’ is not very beneficial in providing a reader an understanding as to how the system will work. Students would do better to design out a particular form/page and then discuss the algorithms required for that part of the system including the SQL queries to be used for that part. Students should be encouraged to think about the data to be used by the system. In a quiz system, for example, it would be beneficial to provide examples of the kind of question(s) to be asked. For a simulation it would be good to see how the formulas are to be used alongside, for example, a sketch of the trajectory of the projectile being modelled.

For a game a student could sketch out the grid or level and talk through, for example, the movement of any enemies.

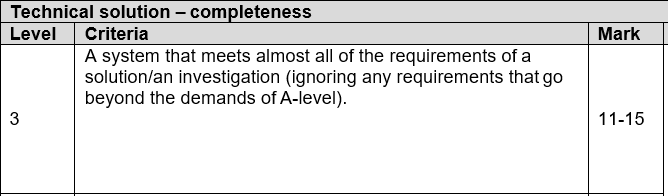
It was common to see algorithms appear without the reader having any real understanding as to how these fitted into the system and a few sketches or examples of the data to be used would help. So, for example, one student produced an excellent Sudoku solver which had some complex pseudo-code in the design section. This code was hard to understand but a few sketches of particular board layouts showing how the individual functions would perform would have really helped. It was common to see many students make use of well-known algorithms such as the merge sort. Just providing the pseudo code for this algorithm is not going to help their documented design mark. If the student talks about how this algorithm is going to be used by the system and integrated then this is beginning to pick up some credit. If the only pseudo-code or algorithm design a student attempts is based around merge sort, quick sort, binary search or other well-known algorithms without any attempt at looking at other parts of the system then the student should not be scoring highly in the documented design section.

# Technical Solution *(42 marks)*

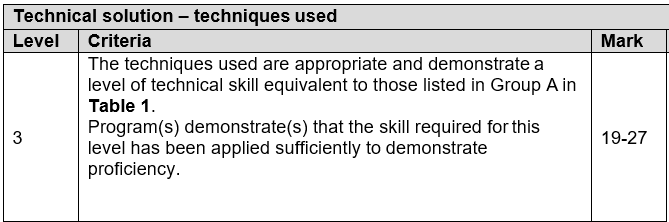
## Completeness Section

In the completeness section we are looking for a consideration as to:

* Has the project met the objectives set by the student in the analysis stage?
* How well have these objectives been met (consideration to HCI / features)?
* Does the technical solution match the original project background description?



## Technical Skills Section



In the technical skills section, we are looking for a marker to:

* Identify parts of the code where complexity is clearly evident and map to Group A/B/C  
  **[This can be helped by a student producing an overview guide]**
* Place consideration into the coding style and comment upon this
* Consider the overall effectiveness of the final solution (does it work how it should?)

### Overview Guide

You should produce a guide here that highlights techniques/ skills in sections A B and C with commentary. Your code should be commented to show where specific technical skills are being used e.g. # SKILL GROUP A – Graph Traversal

### Code listing

This should be fully commented and broken into suitable sections with subheadings. You can turn this section to landscape to make it easier to read. Ensure this is **fully annotated** with comments. Can use a website like hilite.me. Do not use ‘dark mode’ images or screenshots

# System testing *(8 marks)*

## Test plan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test number | Purpose of test | Test data | Expected outcome | Reference to test result |

## Testing Videos

Record your program being used, narrate each objective being shown to have been implemented. Give the objective in the audio of the recording. Provide bit.ly shortened links in the document to your videos which have been uploaded to youtube. Use a large monospace (e.g courier new) font with no underlining e.g.

https://bit.ly/abc0DEF12

# Evaluation *(4 marks)*

## Comparison of project performance against the objectives

## Effectiveness of the solution

## Analysis of user feedback

This should include the original user who was involved at the analysis stage

## Possible improvements

Be verbose about what could be improved – make a full statement. Get user feedback for improvement

# Appendix

## Git log

## References to web sites or other resources used

