A'yaan Abdul-Mughis

Dulwich College | 2025

Logic Express

OCR A Level Computer Science Component 3

Table of Contents

[Analysis 3](#_Toc170641220)

[Problem Identification 3](#_Toc170641221)

[Computational Methods 3](#_Toc170641222)

[Stakeholders 3](#_Toc170641223)

[S Interview 4](#_Toc170641224)

[Form 1 4](#_Toc170641225)

[Justification 6](#_Toc170641226)

[Requirements 7](#_Toc170641227)

[Must 7](#_Toc170641228)

[Should 7](#_Toc170641229)

[Could 7](#_Toc170641230)

[Won’t 8](#_Toc170641231)

[Research 9](#_Toc170641232)

[OpenTTD 9](#_Toc170641233)

[Gamesnacks Strategy Games 11](#_Toc170641234)

[YouTube Videos 12](#_Toc170641235)

[Python Games 12](#_Toc170641236)

[Themes 14](#_Toc170641237)

[Solution 14](#_Toc170641238)

[Success Criteria 14](#_Toc170641239)

[Design 15](#_Toc170641240)

[Interface 15](#_Toc170641241)

[Data Structures 15](#_Toc170641242)

[Algorithms 15](#_Toc170641243)

[Decomposition 15](#_Toc170641244)

[Solution 15](#_Toc170641245)

[Testing Approach 15](#_Toc170641246)

[Implementation 16](#_Toc170641247)

[Iterative Process & Comments 16](#_Toc170641248)

[Iteration 1 16](#_Toc170641249)

[Error 1 16](#_Toc170641250)

[Iteration 2 16](#_Toc170641251)

[Iteration 3 16](#_Toc170641252)

[Testing 17](#_Toc170641253)

[Executed Test Plan 17](#_Toc170641254)

[Evaluation 18](#_Toc170641255)

[Testing 18](#_Toc170641256)

[Solution 18](#_Toc170641257)

[Product 18](#_Toc170641258)

[Maintenance & Development 18](#_Toc170641259)

Analysis

# Problem Identification

There is an innumerable selection of games available for teenagers to play. Although I do not necessarily enjoy playing the most popular games (notably first person shooters or action-adventure games) I do enjoy the puzzle and problem-solving aspect of other games such as sudoku. It would make sense, therefore, that I develop a game which suits this preference, and find suitable stakeholders who agrees—motivation is a huge factor to consider when developing a project.

Problem solving as a skill is also widely applicable to other aspects of life. Logic and solving problems is a valuable skill for any career.

# Computational Methods

There are a number of ways a puzzle game would suit a computational approach.[[1]](#footnote-1)

1. **Procedural Content Generation**

Levels may be randomly generated based on a set of rules.

Levels can also be made manually, and this would be easier. Making them auto-generate would mean the game would be more varied.

1. **Graphics Rendering**

Makes animations more realistic and visually appealing.

Basic animations are fine, but visual appeal matters during gaming.

1. **Collision Detection**

Detect when an object is touching another, or whether the mouse is touching an object.

Very important part of the game mechanics.

1. **Sound & Music Integration**

Play sound throughout the game, and when actions occur within the game.

This will help make the game feel more relaxing to the player.

1. **Game State Management**

Manage screens in the game like menu, start page, and tutorials.

Helps to structure the game into different sections, to make it feel more professional.

1. **Scoring**

Store scores in a persistent text file, or database.

Adds an element of competition to the game.

Without these computational elements, the game would not be very complex and would not be a significant challenge.

# Stakeholders

Finding a stakeholder for my game was relatively simple. One of my peers (S) plays many many strategy and puzzler games and they have agreed to be the stakeholder for my project. He is 18 years of age and considers himself to be an experienced gamer. He will be playing this game for leisure after he has completed his school work, or during the holidays. While S does have a proper gaming PC at his desk, he often travels abroad, and his laptop is rather underpowered. This game should provide him some enjoyment as it will not require an expensive graphics card.

This game will be developed for casual gamers, like S, who enjoy problem solving.

## S Interview

To begin with, I asked S a series of questions to gauge his interest and develop an initial idea of the requirements.

**What, in your opinion, makes a game interesting?**

A unique twist—a game has to have something to make it truly stand out. So many game concepts have already been made a good game needs more than just to be fun, but also to give the audience an experience they would find nowhere else.

**What is your favourite problem-solving-oriented game, and why?**

Personally it's Chants of Sennaar, it is a puzzle game structured around the way written language is formed. It gets a little boring towards the end as it tries to oversimplify more complex grammatical techniques, but it is overall an ingenious way of forcing the player to think outside the box and use knowledge of language to solve traditional puzzles.

**What makes a game stand out for you?**

The two things that make a game stand out to me the most at a first glance are the art style and the primary mechanic. If both look interesting the game is bound to be interesting.

**What does gaming mean to you? Why do you game?**

I usually only game in my spare time to relax, meaning I am not usually as interested in competitive games and rather a more casual playstyle. I am also a big fan of games that make the player feel powerful while still being challenging. These are a great way to let loose while still having to prove my skill.

**What’s the main emotion you feel after you have finished gaming—is it mainly enjoyment or sense of achievement?**

What I usually look for after I have finished gaming is a sense of accomplishment, especially when they get recorded in the form of achievements or progression bars.

As well as the initial interview, S and I will have regular weekly meetings to update him on the progress of the project. His feedback will inform successive stages of development, and this ensures that all work carried out it in line with the requirements.

## Form 1

In early 2024, I sent out a form to gauge the interests of some other potential stakeholders to help inform my gaming decisions and narrow down my choices.

I asked the following questions:

1. What type of game would be most appealing to you?
2. What games do you play?
   * If a stakeholder answered with games I hadn’t played before, I then disregarded their other answers
3. How do you like games to make you feel?
4. What is your preferred gaming platform?

The results were inconclusive, but I did find that the majority of people answered that the game should be:

* Exciting and mentally challenging (Figure 1)
* Get progressively harder (Figure 2)
* Based on strategy (Figure 2)
* Should be on desktop (Figure 3)

Forms response chart. Question title: How would you like my game to make you feel?
Select as many options as you wish.
. Number of responses: 6 responses.

Figure 1

Forms response chart. Question title: What type of game would be most appealing to you?
Select as many options as you wish. Feel free to write anything else in the Other area.
. Number of responses: 6 responses.

Figure 2

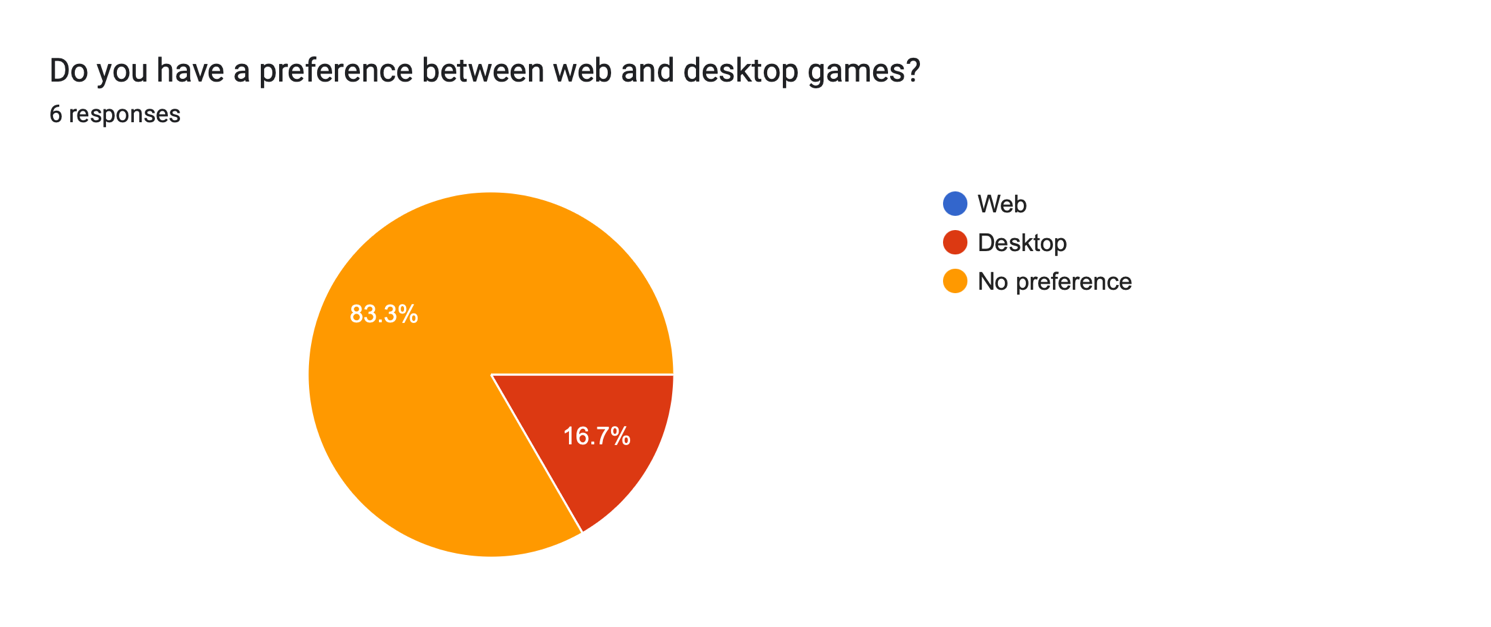


Figure 3

## Justification

As shown above, the data shows S, as well as the other potential stakeholders, would prefer a game of strategy, which aligns well with something I would enjoy designing and implementing.

It is important that I am motivated to complete the project, and the process is enjoyable.

## Requirements

### Must

|  |  |
| --- | --- |
| Requirement | Justification |
| 1. Function offline | Stakeholder requirement |
| 1. Use object-oriented programming concepts | Exam requirement |
| 1. Be a game | Stakeholder requirement |
| 1. Have an original and recognisable name | Personal interest (unique & memorable game) |
| 1. Run on any computer with a Python interpreter installed | Basic software requirement |
| 1. Be controlled with a keyboard and mouse | Stakeholder requirement, other forms of interaction are infeasible to implement |
| 1. Have a graphical user interface | Stakeholder requirement |
| 1. Use computational methods | Exam requirement |

### Should

|  |  |
| --- | --- |
| Requirement | Justification |
| 1. Be written in Python | Personal requirement (experienced in Python) |
| 1. Use the Pygame library for the interface and game loop | Personal requirement (experienced in Pygame) |
| 1. Use comments | Ease of programmer and gamer |
| 1. Be easy to understand and to play | Ease of gamer |
| 1. Have a 800 lines of code, including comments | Exam requirement |
| 1. Have a main menu | Stakeholder requirement |
| 1. Have multiple levels with increasing difficulty | Stakeholder requirement |
| 1. Be humorous with intelligent puns | Personal requirement (unique & memorable game) |
| 1. Have a scoring mechanism | Stakeholder requirement |

### Could

|  |  |
| --- | --- |
| Requirement | Justification |
| 1. Have sound effects | Research conclusion (relaxing gaming experience) |
| 1. Use text files to save and load game progress | Motivation |
| 1. Have paid and professional game art | Look & feel of game |
| 1. Have a parallax effect | Look & feel of game |
| 1. Have a parental controls feature that limits time on the game | Showcase programming expertise (not a requirement) |
| 1. Have an elaborate colour scheme which matches the game story | Look & feel of game |
| 1. Have a start screen the first time the game is opened to explain the game | Ease of gamer |
| 1. Have a dedicated download website with updates, a help page and more information | Ease of gamer |
| 1. Support multiple languages | Ease of gamer |

### Won’t

|  |  |
| --- | --- |
| Requirement | Justification |
| 1. Be in 3D, or try to simulate it | Infeasible |
| 1. Have any major bugs that impact the enjoyment of the game | Enjoyment of game |
| 1. Have unintentional loopholes that bypass game logic | Enjoyment of game |
| 1. Be violent, graphic, or destructive | Personal distaste |
| 1. Not include any AI elements | Infeasible |
| 1. Connect to the Internet for software updates | Infeasible |

# Research

As mentioned above, I am not familiar with gaming and needed to research a wide range of games that I could potentially build. The main type of game genres I found (that were realistic to replicate) were platformer and puzzler. Most other game types like first-person shooter (e.g.: Fortnite) and sandbox (e.g.: Minecraft) require complicated 3D graphics outside of the scope of this project.

During my research, I was pleased to be proven wrong that not all games are designed to be addictive. In fact, I found a large community of games much more thoughtfully designed, and so wished to make a game in this genre.

The most basic template for a game is an arcade game, which is what I have decided to go ahead with. I already have experience in coding Space Invaders, and a basic attempt at PAC-MAN.

## OpenTTD

I found one particular open source game called OpenTTD.Une image contenant carte, capture d’écran, texte, Logiciel multimédia

Description générée automatiquement

Figure 4: OpenTTD Home Screen

Une image contenant carte, capture d’écran

Description générée automatiquement

Figure 5: OpenTTD Game

Une image contenant carte, capture d’écran

Description générée automatiquement

Figure 6: OpenTTD Gameplay

Une image contenant texte, capture d’écran, logiciel, Site web

Description générée automatiquement

Figure 7: OpenTTD Wiki

At first, the game was difficult to play, but I grew to enjoy the complexity as it meant there were so many possibilities, and that the game is designed for exploration more than anything else. The game is clearly very technical and I am aware that coding something of this nature would be a struggle for me. The design of the game felt nostalgic, which I appreciated.

From this research, I learn that “old” games are still popular, and that complexity is not necessarily a bad thing. There should also be some scope for adventure, as this was a successful element of the game.

## Une image contenant texte, capture d’écran, multimédia, Appareil électronique Description générée automatiquementUne image contenant capture d’écran, Système d’exploitation, texte, multimédia Description générée automatiquementUne image contenant texte, capture d’écran, multimédia, logiciel Description générée automatiquementGamesnacks Strategy Games

Figure 8: Dice Puzzle

Figure 9: Block Puzzle

Figure 10: Jigsaw Puzzles Hexa

<https://gamesnacks.com/>

Gamesnacks provides simple games that can be played with a bad Internet connection on simple hardware. The main outcome from these games was realising that relaxing music and very basic graphics can also make a game enjoyable. My favourite of these three was in Figure 10: Jigsaw Puzzles Hexa which had a reward element included, as well as a time limit to increase the speed at which the player completes the puzzles to add a competition aspect. There is scope in my game to have a timer-based scoring system.

From this research, I learned that music can add a lot to the game, and I will also be adding music to my game for a relaxing element. I also have learnt that a reward element is essential, and the best way to implement this is time-based scoring. Scores can then be stored in a text file.

## YouTube Videos

**You Are Given Strange Objects, And Must Learn Their Rules** <https://youtu.be/RDp33e3ttTE>

Figure 11

**How Have I Never Played These Iconic Games Before?** <https://youtu.be/Ti02T7X9oPo>

Figure 12

I researched some different ways I might implement the game. A popular choice was Unity, but I am most comfortable coding in Python, and am aware of the Pygame module used by other Python developers to make games.

I also found YouTube to be a good source of game reviews and gameplay videos. The Figures 8 & 9 helped me understand how puzzle games are built and what makes a successful puzzle game. I had initially started implementing an escape room game for this project, but this research revealed why it would not have been successful.

This research was most helpful for identifying what is possible with Pygame.

## Python Games

All other Python games that I had the opportunity to play were simple.

Une image contenant capture d’écran, texte, Logiciel multimédia, affichage

Description générée automatiquement

Figure 13: <https://fusevolt.itch.io/dungeons-and-caves>

I didn’t enjoy this game, as it was very basic and also difficult to understand, but this is the kind of graphics level I will be aiming for.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Figure 14: <https://ohsqueezy.itch.io>

I thoroughly enjoyed playing this game and was a little disappointed that the source code was so complex. This is the style of game I wish to replicate: I liked that it was easy to get started, and there was an element of puzzle in it that makes the user think, though the second puzzle in Figure 11 was significantly more challenging than the first! There is a clear objective.

From this research, I learnt that my game should have a clear objective (and that coding a game isn’t easy!).

## Themes

There were a couple of personal interests that I could feasibly apply to the game, computing and trains. Trains, however, is the much more unique option and I saw an opportunity to add humour by having cheesy National-Rail-related puns throughout the game that would appeal to my audience.

# Solution

After the above research, it is very clear that my stakeholder would enjoy playing a simple train puzzler, developed in Pygame.

Bringing together all of the elements discussed, the game would ideally also:

* Be mentally challenging
* Increase in difficulty with every level
* Adventurous
* Have relaxing music
* Time-based scoring that can be saved to a text file
* Have a clear objective

## Success Criteria

I would consider this game to be a success if:

1. The game meets all Must and Should requirements (testing)
2. The stakeholder enjoys playing the game (interview stakeholder again)

Design

# Interface

* Draw, label, explain and **justify** choices made

# Data Structures

* Classes with attributes and their methods, **justify**

# Algorithms

* All algorithms made or used need their own pseudocode

# Decomposition

# Solution

# Testing Approach

* Any information here about how you plan to test the project and who you plan to test it with

Implementation

# Iterative Process & Comments

## Iteration 1

* Go back through GitHub log and take screenshots of code development and show the story of the code
* Screens, objects, classes and game loop changes all require a **justification**

### Error 1

* Show error
* Explain error
* Fix error
* Show error fix

## Iteration 2

* Movements, game logic etc.

### Iteration 3

* “AI” and how enemies react

Testing

# Executed Test Plan

* Use Computer Science textbook and approach testing formulaically

Evaluation

# Testing

# Solution

# Product

# Maintenance & Development

* How the product can be improved and maintained
* How effective was the development technique

1. ChatGPT used to generate a list of possible computational methods

   Accessed: 29 Jun 2024 2 PM

   Prompt: give some examples of computational methods that can be applied to a Python game [↑](#footnote-ref-1)