Logic Based Solution to Mastermind

Author Kyle Dick

MEng (MA) Software Engineering

4th Year Dissertation

Supervised by Kathrin Stark

Heriot-Watt University

School of Mathematical and Computer Sciences

Department of Computer Science

May 2023

Abstract

Here is a summary of the work presented in this dissertation.

To be updated with each main body draft.

Declaration

I, Author Kyle Dick confirm that this work submitted for assessment is my own and is expressed in my own words. Any uses made within it of the works of other authors in any form (e.g., ideas, equations, figures, text, tables, programs) are properly acknowledged at any point of their use. A list of the references employed is included.

Signed:

Date:

Acknowledgements

I thank this and that person for their help.

Table of Contents

List of Figures

List of Tables

List of Equations

List of Algorithms

List of Codes

Chapter 1 Introduction

1.1. Motivation

Mastermind is a codebreaking game played between two players each with opposing goals. The game was originally a board game which would utilise plastic pegs on a board to represent codes and other elements of the gameplay, a brief description of how the game functions is as follows. The code-maker (CM) is tasked with constructing a hidden-code (HC) within the boundaries of set parameters. These parameters can be defined as a length l which the HC must exactly match in it's number of present symbols and a set of symbols s which is used to construct the HC. In the standard variant of the game the length is four symbols whilst the set consists of six symbols of which repeats can be present within the HC.

The code-breaker (CB) is the role designated to the player that is tasked with discovering the contents and arrangement of the HC. This processs involves the CB attemtping guesses at what they believe the HC should be while refining their guesses based on responses given by the CM as to the accuracy of each guess. These responses utilise two metrics to judge the accuracy of a guess. The first metric in a response are white pegs which symbolise that a symbol in the current guess is present within the HC but has been arranged in an incorrect position. The second metric are black pegs which represent a symbol in the guess which is present within the HC and is also in an identical position in both the guess and the HC.

1.2. Aim and Objectives

So in this dissertation we aim to address this aspects of Y. In particular we want to achieve the following objectives:

- Something;

* Another thing with different bullet;

- The last thing.

1.3. Methodology

Here are briefly the main problems, and what approach we used to tackle them.

1.4. Contributions

In order, what this dissertation contributes:

1. First item.
2. Second item.
3. Third item.

1.5. Organisation

Here is how this dissertation is organised. After motivating and introducing our work (this chapter), we investigate the literature to present the state-of-the-art in Chapter 2. We then present our great solution in Chapter 3, before evaluating it in Chapter 4. Finally we conclude in Chapter 5, highlighting limitations, and possible future work.

Chapter 2 Background

Some text to explain what's to come. In Section 2.1 we explore concept A, then we continue with concept B in Section 2.2

2.1. Concept A

Doe introduces the context of Y [2]. Smith et al. discuss the implementation of X [3]. We implement W, as describe by Alice [1].

2.2. Concept B

More references.

2.3. Conclusion

Here we conclude the background, recap concepts explored and key notions for rest of document.

In next chapter, Chapter 3, we do some implementation for concept A. Chapter 4 will detail our evaluation of concept B.

Chapter 3 Development

Here be some development specifications. We follow the steps for concept A, as highlighted in Chapter 2, Section 2.1. In Section 3.1 we discuss the back-end, while Section 3.2 will focus on front-end development.

3.1. Back-end

Server stuff here. See Equation (3.1) and Equations (3.2) and (3.3) that explains the maths behind application.

$$B = \sum_{\alpha \in A} \alpha \times \epsilon \tag{3.1}$$

$$\begin{aligned} X &= a^{\epsilon} \\ Z &= \frac{\sqrt{A \times B}}{\sum_{\gamma \in \Gamma} \gamma} \end{aligned} \tag{3.2}$$

$$x(r) = \begin{cases} a^2 + \sqrt{3}b & \text{if } r \text{ is even} \\ a^2 + \frac{\sqrt{3}}{2}b & \text{if } r \text{ is odd} \end{cases} \tag{3.3}$$

The algorithm for X (Algorithm 1) has such and such advantages. The Pyhton implementation is shown in Code 3.1.

```
import library

# GCD of a and b
def my_gcd(a, b):
    t = b
```

Algorithm 1 Euclidean Algorithm

```
function EUCLID(a, b)                          ▷ Finding the GCD of a and b
    while b ≠ 0 do
        t ← b
        b ← a mod b
        a ← t
    end while
    return a
end function
```

```
    while not b == 0:
        t = b
        b = a % b
        a = t
    return a
```

Code 3.1: Python example of GCD

3.2. Front-end

Interface specs here. See the mockup for interface in Figure 3.1[1].

In Figure 3.2a we detail the data flow for process A and while process B is shown in Figure 3.2b.

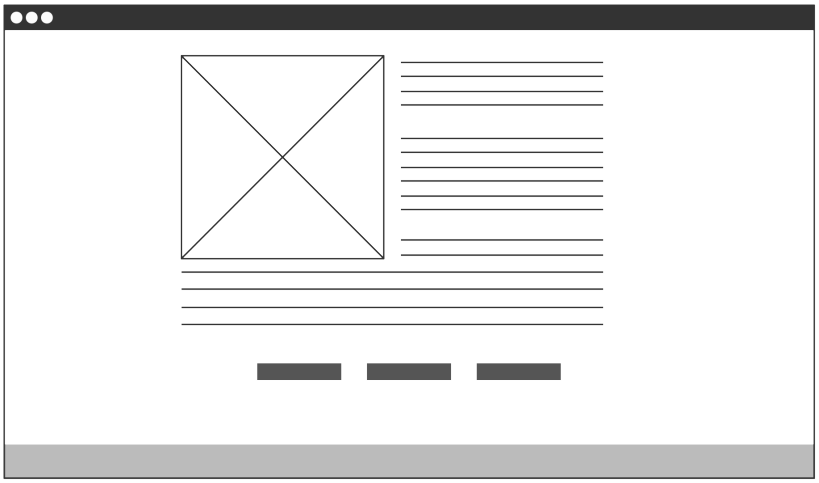
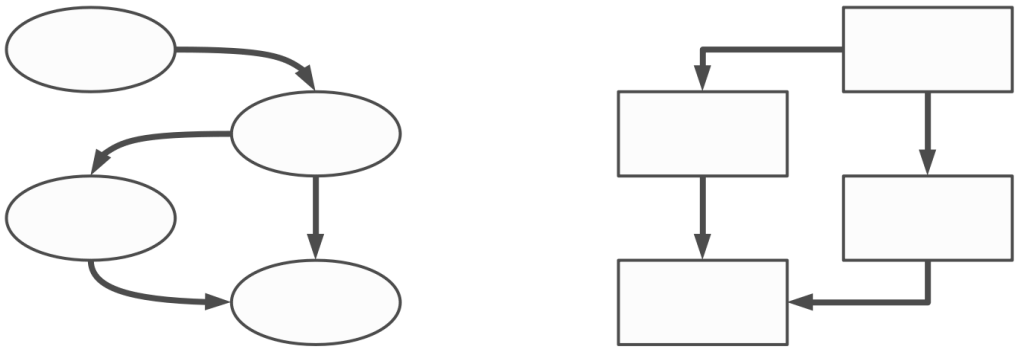


Figure 3.1: Front-end mockup. With an added description to help the reader (won't appear in list of figures).

[1]Complete details in Appendix B, page 11



(a) Flowchart A. **(b)** Flowchart B.

Figure 3.2: Flowchart for some processes.

3.3. Conclusion

This chapter has detailed our system implementation. In particular, we have divided the processes into the back-end (Section 3.1) and the front-end (Section 3.2).

In the next chapter, Chapter 4, we evaluate our system.

Chapter 4 Evaluation

In this chapter, we evaluate the system developed in Chapter 3 with two tests, first Test 1 Section 4.1 and then Test 2 Section 4.2. We then explore concept B with users in Section 4.3, as laid out in the Background (Chapter 2, Section 2.2).

4.1. Test 1

4.2. Test 2

4.3. User Study

Table 4.1 listed the many properties highlighted by users.

A	b	c	1
X	lorem	ipsum	2
S	lor	*em ispum*	3.2

Table 4.1: Table of some properties

4.4. Conclusion

In this chapter we have carried the evaluations. Tests 1 and 2 (Sections 4.1 and 4.2) have proved this and that, but highlighted this limitations due to that aspect of the algorithm.

Presented to users, the interface was said to be such and such.

This chapter concludes the work carried out during the project. In the next chapter, Chapter 5, we present the conclusion of this dissertation.

Chapter 5 Conclusion

This is the conclusion of this dissertation. We re-contextualise the motivation for the project (Section 5.1), list our contributions Section 5.2, and finally discuss the limitation of the project and suggest future work Section 5.3.

5.1. Motivation and Goals

Here we will summarise briefly what were the motivations for the project, and the main objectives we tried to achieve.

5.2. Contributions

Here we will summarise the main contributions and achievements of this project, in relation to the initial objectives, and in contrast with similar work found in the background chapter (Chapter 2).

We will also summarise some of the methodology used to achieve goals and tackle problems.

5.3. Limitations and Future Work

Here we will highlight the main limitations of the project, and suggest possible corrections, extensions, or uses for our work in future projects.

References

[1] Bob Alice. *Handbook of this method*. Great Publisher, 2016.

[2] Jane Doe. Exploring y. In *Proceedings of that conference*, 2017.

[3] John J Smith and Jane Doe. Implementing x in z context. *Communications of this association*, 2018.

Appendix A UML Diagrams

Some text introducing the diagrams.

The diagrams.

Appendix B Screenshots of my application

The screenshots.

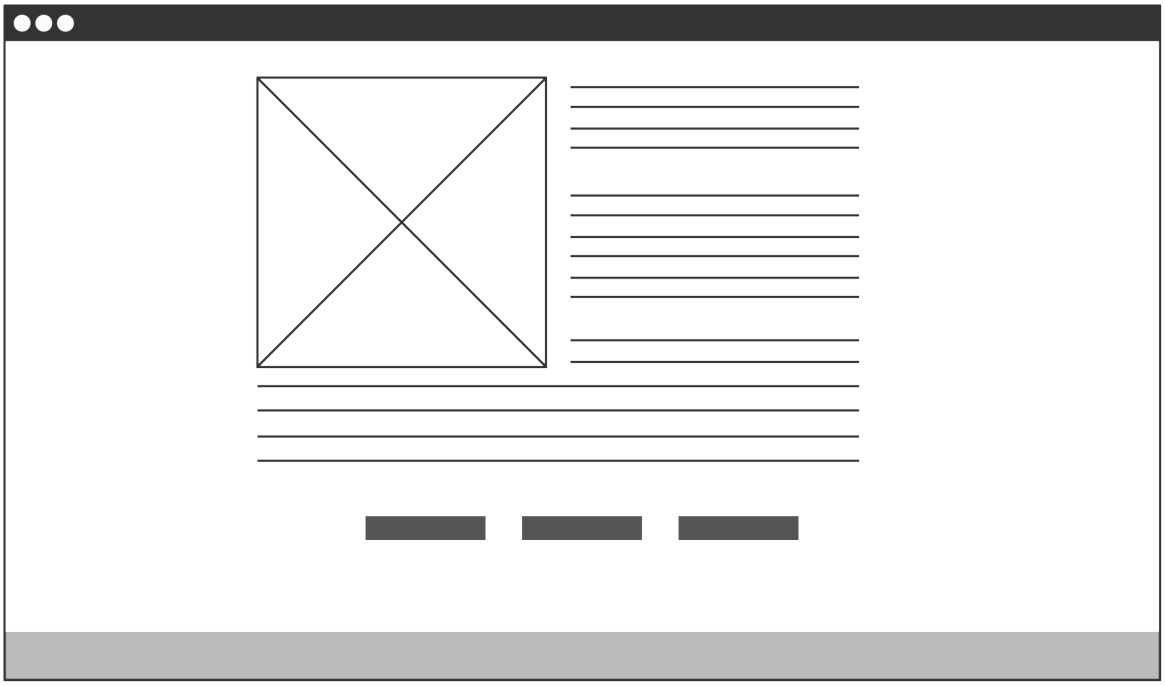


Figure B.1: Mockup of front-end.