Evaluation of Mastermind Solutions
Deliverable 1: Research Report

Kyle Dick
F20PA Project
supervised by Kathrin Stark

November 21, 2022

Abstract

Describe the problem that is to be approached.

Contents

1 Introduction

The main focus of this project is to investigate solutions to the puzzle game known as Mastermind. The approach will be to develop an initial base solution in a functional programming language which can be iteratively improved with the goal of creating a solution which is logically sound and efficient. Mastermind is a simple code-breaking game in which one player will create a secret code which consists of four coloured pegs from a choice of six. This is the standard version of the game however there are multiple variations of the game, some of which will be discussed and examined within the scope of this project.

1.1 Aims and Objectives

The aims and objectives of this project can be surmised in two specific goals. The first goal is to derive a solution to the logical puzzle Mastermind. The second goal is to evaluate solutions implemented during this project with the aim to find methods to improve later iterations. An in depth explanation of the chosen aims and objectives are as such:

• Aim 1: To derive a solution to the puzzle Mastermind.

The goal to find a solution to the problem which minimizes the number of guesses required to discover the correct combination of pegs which comprise the code. Intially the goal will be to derive a base solution to the Mastermind puzzle. The base solution being the most intuitive solution to the problem which does not aim to be the most efficient but to provide a foundation on which improvements can be made. The following objectives are associated with this aim:

• Objective 1.A: Investigate the problem space.

The Mastermind puzzle has previously been the subject of similar research regarding the ability to efficiently find the correct code combination. This stage of the project will concern itself with investigating these previous implementations to guide the project. Through exploring the methods utilised in other investigations into this problem the areas in which these solutions are lacking or could be improved can be found.

• Objective 1.B: Implement a base solution.

The ideal base solution should acheive the basic goal of arriving at the correct code combination but should not be the most elegant solution at this point. Instead the base solution should be the foundation for which improvements are made

in later iterations. The base solution will be guided through the research conducted through objective 1.A.

• Objective 1.C: Evaluate implementation and iterate.

This objective is a continous process which will be referred to throughout the life cycle of the project. Each iteration of the solution, starting with the base solution, should be evaluation against a set of criteria defined later in the document. After this review possible improvements can be documented and worked towards for the next iteration.
Aim 2: The goal of this project is to arrive at a solution which can be evaluated in terms of its soundness and efficiency. This means that the solution should be one that is easily understood with the appropriate explanation. The steps in which the solution is arrived at should follow a logical progression that is reflected in its documentation. The efficiency should be measured through how quickly the correct code combination can be found by the solution.
Objective 2.A: Designing an evaluation process for this project should be foremost conernced with how well the solution is able to solve the problem. The Mastermind puzzle is most commonly restricted by the amount of guesses allowed to the player. From this the metric in which a solution should be measured against is how quickly the correct code combination can be found and its minimum amount of guesses to do so. A solution which does not reach the correct code combination within a set amount of allowed guesses will be considered a failed solution. Objective 2.B: Analyse an iteration using the chosen strategy. Objective 2.C: Report the results of the evaluation strategy and iterate if improvements are identified. Objective 2.D: Document results taken to arrive at a desired solution.

2 Background

This section will provide background material which aims to give context to the aims and objectives of this project. The project was inspired by a paper exploring sudoku solutions from a series of problems known as functional pearls, the processes used to derive those solutions will be used to guide this project in this current stage. Following this brief introduction is an explanation of the game Mastermind which the solution will be derived from along with references to previous work by others. The previous work examined will focus on the specific area of search spaces in regards to finding the optimal best move at each position in the puzzle. At the conclusion of this section the goal is that the reader has an understanding of the important concepts relating to this project such that the aims and objectives are clear in their feasability and relevancy.

2.1 Mastermind

Mastermind is a two-player code-breaking game centered around a code-maker (CM) who creates a secret code and a code-breaker (CB) who attempts to discover the code [1].The physical board game was designed by an Isreali Postmaster named Mordecai Meirowitz and originally manufactured by Invicta Toys and Games [2].
The standard variant of the game plays as follows:

- The CM will create a secret code of length L unknown to CB from a set of symbols S, in the physical game these were represented by plastic pegs. In the standard variant $L = 4$ and $S = 6$ represents a code of four symbols in length constructed from a set of six possible symbols.

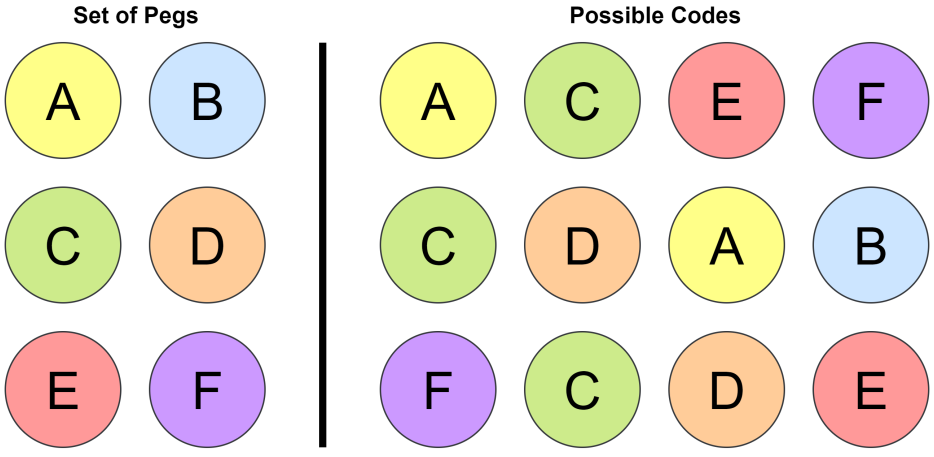

Figure 1: Example of code enumerations the CM could construct in the standard variant.

- The CB is tasked with solving the code that the CM has constructed by creating test patterns from S. In the physical board game the number of guesses that the CB is allowed is limited to 10 [3], the size of the plastic board. The test patterns are evaluated by CM through a pair of indicators. The first indicator, represented by a white plastic peg in the physical game, denotes that a correct symbol was used but in an incorrect position. The second indicator, represented by a black plastic peg, denotes that a correct symbol was used in the correct position.

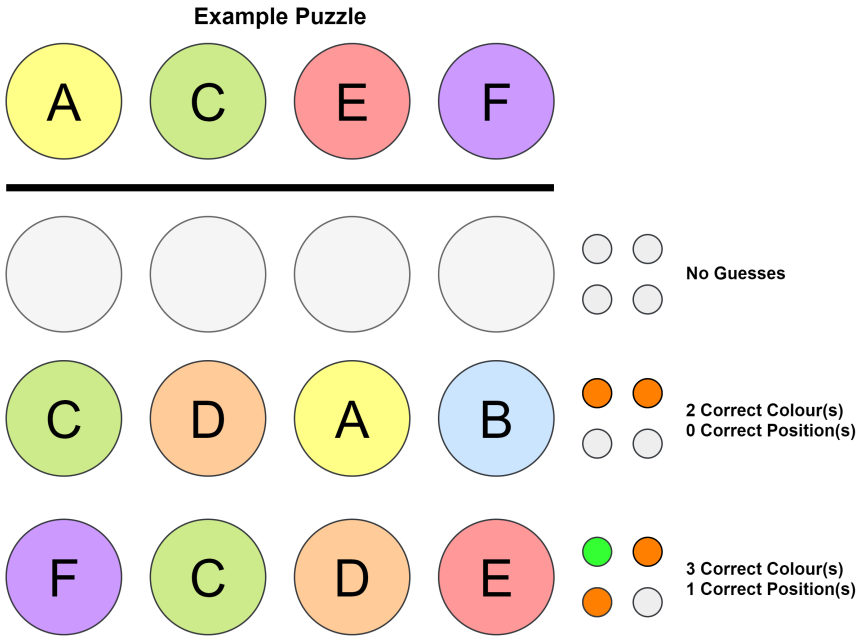

Figure 2: A simplified representation of CB attempting to break the code set by CM

- The game concludes when either the CB is able to correctly guess the code set by the CM or they exceed the allowed number of guesses.

The game can be treated as a individual puzzle in the absence of a human CM through randomly generated code combinations. This project aims to implement a solution to this puzzle that can search for the correct code combination that is comparable with exisiting solutions. The act of searching for the correct code in this context is known as a combinatorial optimization problem with a wide range of applications [4] and it is because of this applicability that the search for a solution is subject to continued investigation.

2.2 Donald Knuth's Solution and Search Spaces

One of the earliest proposed solutions to the Mastermind puzzle was a paper by Donald E. Knuth in which the claim is made that it is possible in all cases for the CB to find the correct code within five attempts [1] [5]. The solution introduces a concept which is important to solving search problems known as the *consistent search space*, a term which would be defined further in later work [6]. The standard variant of Mastermind assumes that the code is of length four with six possible elements it can be constructed from. This definition would mean the code constructed by the CM would be one of $6^4 = 1296$ possible combinations, this set of combinations is referred to as the *s*earch space of the puzzle.
The algorithm that Knuth proposes seeks to restrict this search space after each guess using the feedback provided by the CM. The algorithm proceeds after each guess by picking the best possible choice for the subsequent guess at each step, this greedy strategy [1] is possible as the feedback provided can eliminate combinations which are not consistent with previously guesses.
The first stage of Knuth's investigation was to define the rules of the game in a way that a computer would be able to understand. First defining the two important variables being the code $x_1x_2x_3x_4$ and the test pattern, or guess, $y_1y_2y_3y_4$
The initial rules that were derived were as follows:

- The number of "black hits (B)", i.e., the number of positions j such that $x_j = y_i$
- The number of "white hits (W)", i.e, the number of positions j such that $x_j =!y_j$ *but* $x_j = y_k$ for some k and y_k which has not at this point been a hit. [5]

The process of restricting the search space can be seen in the example below with the symbols represented as integers (1, 2, 3, 4, 5, 6):

No. of Guesses	Test Pattern	Hits	Possible Combinations
0	no guess	N/A	1296
1	1122	WWW	16
2	1213	BWW	4

At the initial point where no guess had been made the search space is still comprised of all 1296 possible code combinations. This space of possible combinations is restricted using the result of the first guess which states that three of the chosen symbols were correct however their positions were incorrect. This information allows a great amount of restriction to only 16 possible code combinations which the code could belong to. The second guess restricting this space even further to just four possible combinations. Through this method Knuth asserted that it should be possible to achieve a correct guess within five test patterns. The starting pattern discovered by Knuth to give proof to this claim was found to be 1122.
A possible progression of stages from this initial test pattern would be: Another example is as follows:

No. of Guesses	Test Pattern	Hits	Possible Combinations
0	no guess	N/A	1296
1	1122	B	256
2	1344	W	44
1	3526	W	7

This situation would also allow for the next test pattern to distuingish the final possible combinations using the test pattern 1462. The algorithm places a high value on test patterns which coax the process towards the fourth guess being able to distinguish between the possible combinations. This method of solving the problem however is admitted within the paper to not be the most optimal solution. This can be shown by the way that the algorithm processes the following situation:

No. of Guesses	Test Pattern	Hits	Possible Combinations
0	no guess	N/A	1296
1	1122	BWW	16
2	1213	BB	4

which results in a possible four remaining code words (2212, 4212, 5212, 6212). The algorithm would select the test pattern 1145 as the next guess however when compared to the test pattern 4222 it can be shown that it in actuality results in less distinguishing results:

Test Pattern	Hits	Code
4222	BWW	2212
	BBB	4212
	BB	5212
	BB	62122
1145	W	2212
	WW	4212
	WW	5212
	W	62122

It is clear that should 4222 be used at the test pattern there are two possibilities in 2212 and 4212 where the code can be known by the third guess. This is a better result compared to the test pattern 1145 where two possibilities will always be the result.

2.3 Scoring Algorithms in Heuristics / Scoring Possible Combinations

2.4 Functional Pearls

The inspiration for this project was a paper by Richard Bird which implemented a solution to the puzzle game Sudoku as an example of a series of problems known as functional pearls. This sections will explore the topic of functional pearls and their relevance to the current project. Functional Pearls begin as small problems

which programmers wish to explore, focusing on brief but engaging examples that showcase either a guided explanation to a proof or presenation of unique data structures. The goal of a functional pearl is to teach important programming techniques and fundamental design principles [7]. Richard Bird described functional pearls in a speech as 'elegant, instructive exampels of functional programming' while showcasing his implementation of a sudoku solution[8].
The solution which this project aims to explore would enter the scope of a functional pearl due to its aim to document an example of functional programming. The project would however deviate from a traditional functional pearl as the scope would be expanded to consider similar solutions in the problem space to evaluate the different methods of solving Mastermind.
The functional pearl which inspired the project is Richard Birds paper 'Functional Pearl: A Program to Solve Sudoku' [9]. The aim of the solution was to define the function:

$$Sudoku :: Board \rightarrow [Board]$$

This function would take an input in the form of a Sudoku board, represented by a matrix of characters, and output a list of possible completed boards. Following this declaration the paper would proceed to use logic and equational reasoning to define the function using the functional language haskell to achieve the solution. Finally Richard Bird was able to arrive at the following definition for his solution:

$$Sudoku :: Board \rightarrow [Board]$$

$$Sudoku = map(maphead) \bullet search \bullet prune \bullet choices$$

where search, prune and choices representing functions declared earlier in the implementation. This sudoku solutions provide an example of how the mastermind solution this project aimst to implement may be approached. Aiming for a solution that resembles this process of declaring an initial function or set of functions then deriving a defintion for them that can be evaluated through comparison with similar solutions. The next section will examine the problem that this project will aim to solve and the work that has been explored in the area previously.

3 Research Methodology and Requirements Analysis

The process of implementing a solution to Mastermind should consider the specifics of how the solution will be optimized. The problem space of solutions to Mastermind puzzles has been explored in multiple over investigations which necessitates that the requirements for this implementation be clearly defined and exhaustive. Guiding the declaration of how the solution should function are the set of research questions which state the purpose of the solution and what it hopes to accomplish.

3.1 Research Methodology

The question that this project aims to answer is 'How can the combinatorial problem of Mastermind be solved'. The scope of this question is quite large so instead it will be broken down into a set of smaller questions that will help to define the research methodology. Those questions are as such:

- How can the search space of consistent combinations be minimized
- Which scoring algorithms provide the best possible chance of selecting the correct code from a set of consistent solutions.

4 Evaluation Strategy

The evaluation of the solutions implemented in this project relate strongly to the research questions that we have defined in the previous section. Following from this means that the metrics on which our solutions will be evaluated on are as such:

- A solution which is able to restrict the consistent combination search space is considered more valuable.
- A solution which is more accurate in selecting the correct code from the consistent search space is more valuable.
- A solution which is able to scale elegantly with both an increase in possible symbols which comprise the code and an increase in the length of the code.

5 Preliminary Work

In here talk about the work done with the current base solution. It will be originally done based on the donald knuth implementation.

6 Project Management

6.1 File and Resource Management

6.2 Timeline and Deadlines

6.3 Risk Analysis and Mitigations

To manage the progress of this project efficiently it was important to identify possible risks that could prevent the realisation of the goals laid out earlier in this document. To aid in the identification of the risks the following key was used to classify the associated risks:

- People (P) - Risks which are the result of issues related to those individuals involved in the project. This relates to the wellbeing, scheduling and personal issues that can be encountered.
- Technological (T) - Risks which can result from the technology being used to engineer the solutions. This relates to the technologial constraints that could be encountered or the hardware required.
- Requirement (R) - Risks which can result from changes to the requirements of the project. This relates to problems encountered with the work being implemented for the project. Logical problems and issues with the material involved would be included in these risks.

ID	Risk	Description
P/T/R1	Textual Title of Risk	Textual Description of the Risk

ID	Risk	Description
P.1	Illness and Health Complications	The situation in which an individual involved in the project is unable to contribute due to illness or health complications. This risk is slightly heightened as the world recovers from the effects of the Covid-19 pandemic however it also considers other conditions which would require time away from the project.
T.1	File Loss	This is the situation in which files or documents relating to the project is lost. This is a low priority issue as precautions have been taken to mitigate this such as using version control through Github.
R.1	Insufficient Base Solution	The success of this project is dependant on having a solid base solution. The success of the base solution is assessed by a set of conditions which deems it suitable for solving the problem. The base solution being unable to meet this conditions is a serious situation and as such should be of a high priority.

References

[1] Weisstein, Eric W. *'"Mastermind." From Mathworld – A Wolfram Web Resource'* https://mathworld.wolfram.com/Mastermind.html (Online; accessed 9-November-2022)

[2] Invicta Toys and Games ltd. *'History of Mastermind'* https://web.archive.org/web/20070812104420/http://dspace.dial.pipex.com/town/road/gbd76/toys.h (Archived 2007)

[3] Nelson, Toby. *'Investigations into the Master Mind Board Game'* https://web.archive.org/web/20150906043015/http://www.tnelson.demon.co.uk/mastermind/index.h (1999, Archived 2015)

[4] Merelo-Guervós, J.J., Castillo, P. and Rivas, V.M. *'Finding a needle in a haystack using hints and evolutionary computation: the case of evolutionary MasterMind', Applied soft computing* **6(2)** (2006) pp. 170–179. doi:10.1016/j.asoc.2004.09.003.

[5] Knuth, Donald E. *'The Computer as Master Mind', Recreational Mathematics* **9(1)** Stanford University (1976, Baywood Publishing Co., Inc.)

[6] Merelo, J.J., Mora, A.M., Cotta, C. and Runarsson, T.P. *'An experimental study of exhaustive solutions for the Mastermind puzzle'.* (2012)

[7] Bird, Richard. *How to Write a Functional Pearl* International Conference on Functional Programming, Portland (2006)

[8] Gibbons, Jeremy. *University of Oxford, Functional Pearls* http://www.cs.ox.ac.uk/people/jeremy.gibbons/pearls/ (2009)

[9] Bird, Richard. *Functional Pearl, A Program to Solve Sudoku* Cambridge University Press, Cambridge, 2006