Logic-Based Solutions to Puzzles and Riddles
Deliverable 1: Research Report

Kyle Dick
F20PA Project
supervised by Kathrin Stark

October 26, 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 conists 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

- Aim 1: To derive a solution to Mastermind using logic and equational reasoning.
  - Objective 1A: Investigate solutions to similar problems. This will give an idea of how an implementation should be approached.
  - Objective 1B: Implement a base solution using simple logic without consideration to efficiency. This basic solution should only be concerned with giving a foundation that can be refined in later iterations.
  - Objective 1C: Iterate and improve on solutions using a declared evaluation process.

  Aim 2: Evaluate solutions and identify a desired solution.
  - Objective 2A: Create an evaluation strategy by identifying which aspects of the solution would best show improvement between iterations.
  - Objective 2B: Analyse an iteration using the chosen strategy.
  - Objective 2C: Report the results of the evaluation strategy and iterate if improvements are identified.
  - Objective 2D: 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. What follows 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 will relate to the field of functional programming and the specific area of logic and equational reasoning. 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 code-breaking board game designed by Mordecai Meirowitz originally manufactured by Invicta Toys and Games [1]. The original board game consisted of two players with different roles. Player A would be designated the code-breaker while Player B would attempt to solve the code.
The process of the game is as follows. Player A would first contruct the code from a set of six coloured pegs with the code being exactly four pegs in length. There are variants where the size of the set of choices and code length are variable however this describes the standard variant.
A variant of the original mastermind puzzle was explored

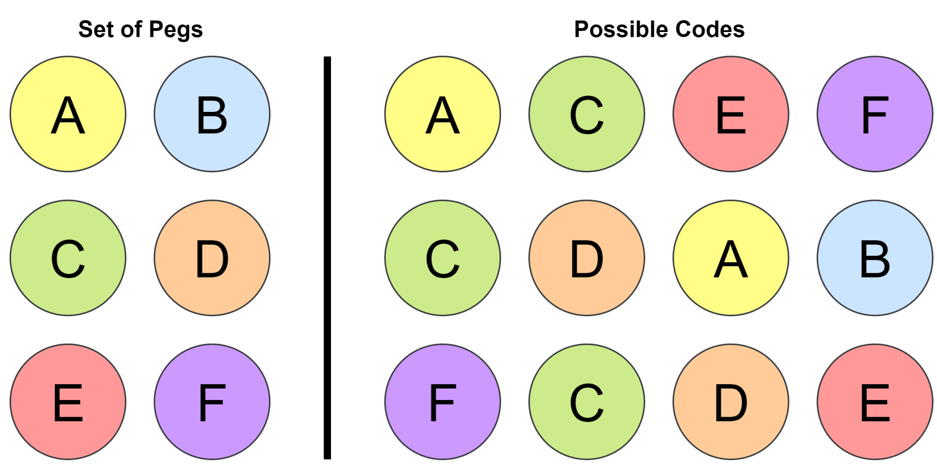

Figure 1: Example of code enumerations Player A could construct

The challenge for Player B is to correctly guess the code created by considering feedback given from Player A relating to how correct each guess was. A guess is evaluated with Player B given a number of pegs with two possible colours, one which represents a correct position and another which represents a correct colour.

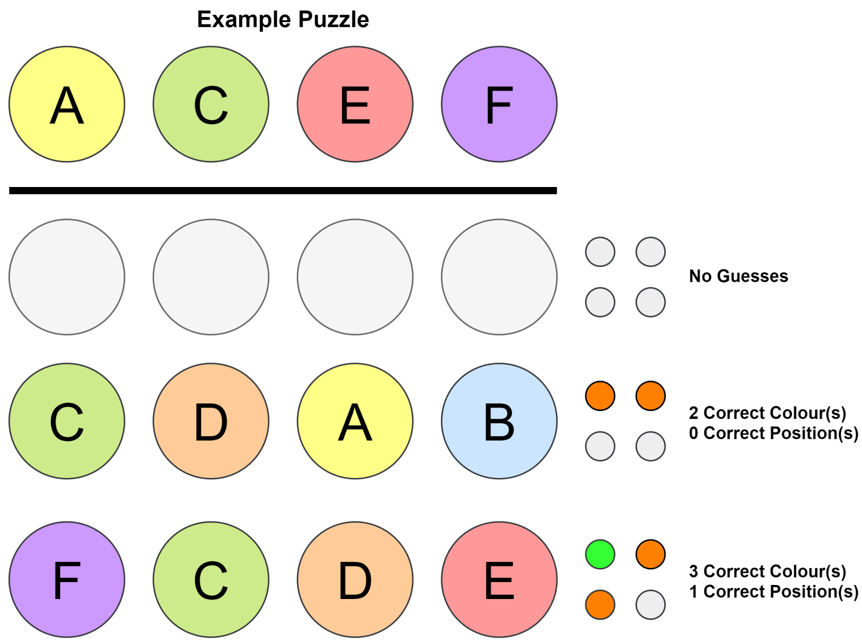

Figure 2: Simplified example of Player B guesses

The solution that this project aims to implement would assume the role of Player B attempting to correctly identify the generated code in the minimum number of guesses. To provide a solution which is not bound by the number of elements within the set of possible choices or the length of the code, the code will follow the following rules.

$$code = [c_i, ..., c_n]$$

$$pegs = \{c_i, ..., c_m\}$$

$$n, m = \mathbb{N} > 1$$

A solution which is agnostic of the code length and number of possible elements would support basic variants of Mastermind where the only difference are these restrictions. The base solution will solely focus on solving for a code of length four with six possible elements.
[This section should be extended with discussion of Clear Mastermind with focus paid to the foundation laid out by the researchers involved.

2.2 Functional Pearls

The solution that this project seeks to implement will utilise functional programming to acheive its goals. This required an analysis of similar problems which have seen investigation in the field. An area which proves useful for this area of research is the topic of Functional Pearls.

Functional Pearls are examples of programming which aim to teach important programming techniques and fundamental design principles [2]. Functional Pearls focus on brief but engaging examples that showcase either a guided explanation to a program calculation or proof or presenation of unique data structures. The encompassing feature of Functional Pearls is that they showcase educational material in a way that is digestable and entertaining.
A functional Pearl example which holds relevance to the current project is Richard Bird's 'A Program to Solve Sudoku' [3]. This paper focused on using equational reasoning over mathematics to implement a solution to sudoku puzzles written in Haskell. This solution is relevant to the current project due to the factor they share similar problems of code breaking. The approach of this solution was to start with a base solution which would generate a large volume of possible boards which would be matched to a given input. The conclusion of the study resulted in an elegant solution which would apply equational reasoning to derive a function capable of giving multiple possible solutions to the given input. The limitations were compared to similar solutions which showed a struggle with larger inputs but an increase over other implementations with simplified board states.

2.3 Logic and Equational Reasoning

3 Research Methodology

4 Evaluation Strategy

5 Project Management

5.1 File and Resource Management

5.2 Timeline and Deadlines

5.3 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 engineering of solutions.
- Technological (T) - Risks which result from the technology being used to engineer the solutions. This involves the software, hardware and frameworks used.
- Requirement (R) - Risks which result from changes to the requirements of the project. Examples of such being a requirement being dropped from the scope or a requirements priority undergoing a change.
- Estimation (E) - Risks which result from ill estimations of the project timing, capabilities of an individual or understanding of a certain technology.

ID	Risk	Risk Type	Description
P/T/R/E1	Textual Title of Risk	Class of Risk	Textual Description of the Risk

References

[1] 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)

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

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