CSC10002 - PROJECT
THE MATCHING GAME

March 16, 2022

I Introduction

The Matching Game (commonly known as Pikachu Puzzle Game) includes a board of multiple cells, each of which presents a figure. The player finds and matches a pair of cells that contain the same figure and connect each other in some particular pattern. A legal match will make the two cells disappear. The game ends when all matching pairs are found. Figure 1 shows some snapshots from the Pikachu Puzzle Game.
In this project, we will develop a simplified version of this Matching Game by remaking the game with characters (instead of figures).

Figure 1: The Pikachu Puzzle Game [1]

[1] google.com

II Detailed Description

II.1 Standard Mode

This mode contains the essential steps to make the game possible.

1. Game starting: Initialize a board with the given size while satisfying the following conditions
   - The total number of cells must be even.
   - The number of distinct characters is specified in advance.
   - For each character, determine an even number to define the number of occurrences for that character.
2. Any matching pair must satisfy the following conditions
   - Characters on the two cells must be identical.
   - When the cells disappear, their positions are replaced with blank spaces.
   - The matching pattern must be one of the below motifs
     - I Matching

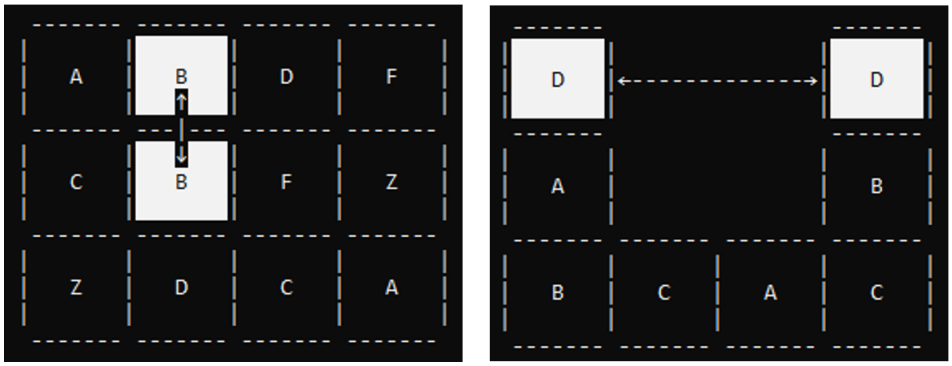

Figure 2: I Matching

     - L Matching

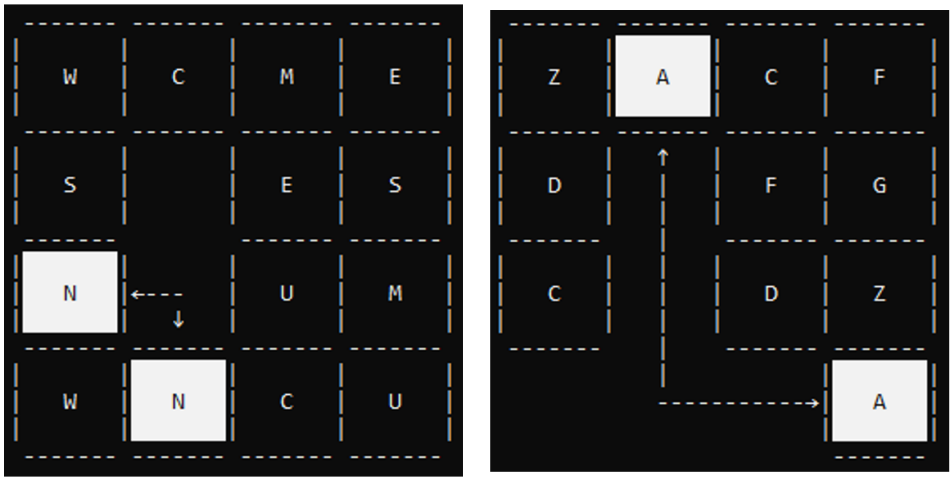

Figure 3: L Matching

– U Matching

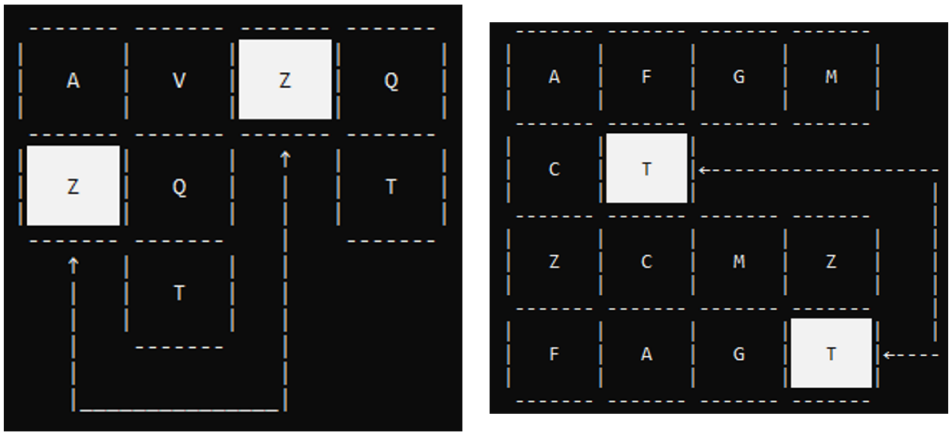

Figure 4: U Matching

– Z Matching

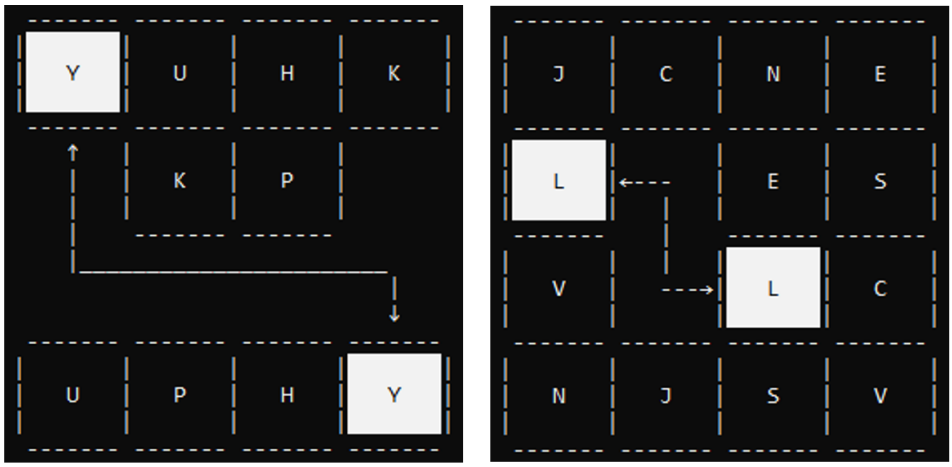

Figure 5: Z Matching

3. Game finishing: check the following conditions
   - Are there any cells left?
   - Are there any valid pairs left?

II.2 Difficult Mode

In this mode, the neighboring cells should slide into the newly emptied spaces in a particular direction (left to right, up to down, etc.) Students may decide to use Pointer, LinkedList, or both for implementation. If you choose to use both data structures, only one is needed to make the game run, and the other should help with the report (The details will be given below)

II.3 Advanced Features

Players will experience the game better if you can add one (or more) of the following extra features to the game.

- Color and sound effects
- Visual effects (Figure 6)

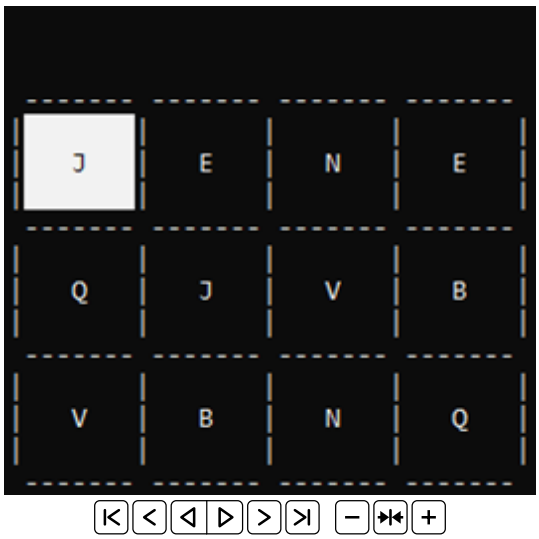

Figure 6: Visual Effect

- Background: You can design anything for a background. Then, when a matching pair disappears, the background content corresponds to those emptied cells.

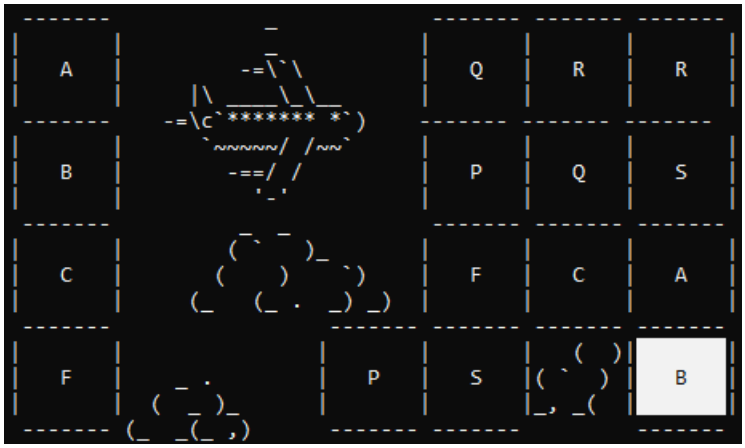

Figure 7: Game with background [2]

- Leaderboard: The top N players that finished their games with the shortest time will register their information to the Leaderboard.
- Move suggestion: Show player a valid match when they press the "Help" key.

[2]Ref: `http://www.cplusplus.com/forum/general/58945/`

III Submission and Grading

III.1 Submission

Your submission **ID1_ID2.zip**, which will be uploaded to Moodle, must contain the following files and folders:

- A `Source` folder contains all "*.cpp" and "*.h" source files
  - These files must be executable via the ".exe" file built from g++. Any other compiler used must be note int the report, but not not recommended.
  - There should be explication comments for your source code.
- A `report.pdf` report which includes all the below points:
  - Your name, ID, and class
  - A tutorial of how the game works
  - An explanation of how to complete all the requirements in Standard Mode by using verbal descriptions or pseudo-code
  - A comparison of running time and the complexity of source code in the Difficult Mode when using Pointer or LinkedList. (If students chose to implement both data structures)
  - Any other remarks about your design and implementation for the Advanced Features.
  - All references, e.g., weblinks and books, must be appropriately cited. If you discuss with your classmates or some high-level mentor, their names must be listed too.
  - **DO NOT** include you source code in the report. That is pointless.
- In your report, there must be an explanation for the necessity of these files and instructions on using them.
- This project description is associated with a checklist. Please make sure that you complete the checklist and include it in your submission.

III.2 Grading

- The lectures will evaluate your submission based on the implementation, report, and what is provided in the checklist.
- Your submission will get a zero score if it commits one of the following issues.
  - No checklist or report included
  - Regulations violated
  - The program is not runnable or malfunctioning.
- Plagiarism and Cheating will result in a 0 point for the entire course and will be subject to appropriate referral to the Management Board for further action.