Question part - introduction

The binary question description for every worker is the same (i.e. Which specials does the creature in the image belong to?), but the two choices are varying sequentially. The answers of all 10 questions in each round would be collected and stored in an array. The answer array would be used as MatLab input. The MatLab algorithm is based on Ulam-Renyi game approach. The MatLab exe file was provided by Qiyu. The output of MatLab is a txt file consisting of two index stated the size of the matrix and a matrix. The ten questions, more precisely, ten pairs of choice, for the next round would be generated according to the MatLab output matrix. All of these is done by PHP files and a MatLab exe file.

Question part – flow chart

Use PHP to create a txt file and write three indexes in it → Use PHP to call MatLab and pass the txt file to it as input → Use MatLab to initialize the dataset →

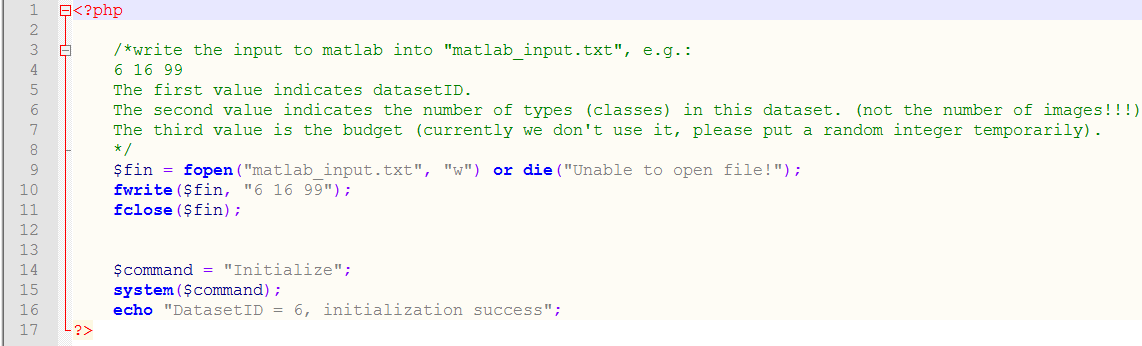
Use PHP to write two indexes and the answer array in a txt file (for the first round, the answer array is void) → Use PHP to call MatLab and pass the txt to file to it as input → Use MatLab to generate a matrix based on the input → Use MatLab to write the output in an txt file → Use PHP to read the matrix → Use PHP to generate a 2-D array of string type contains both the choices (A and B) for all 10 questions → Send to the front end for display → Get the answer array from 10 workers

Question part – details explanation

1. Use PHP to create a txt file and write three indexes in it → Use PHP to call MatLab and pass the txt file to it

The three indexes here are dataset ID (image ID) , number of classes in this dataset and budget for this dataset.

The PHP code is as following, while the explanation is included as comments.

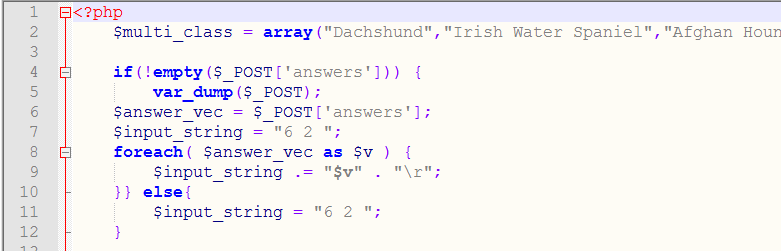


2. Use MatLab to initialize the dataset.

This part is done by a MatLab exe file named as “Initialize” provided by Qiyu.

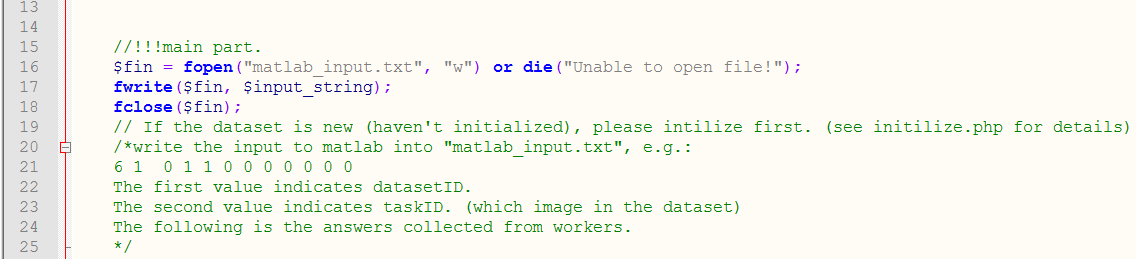


3. Use PHP to write two indexes and the answer array in a txt file (for the first round, the answer array is void).

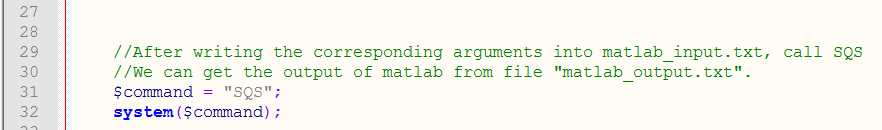


Firstly, store the name of 16 species, not fully listed in the screenshot, in an array variable called “multi\_class”.

The if/else statement is to pass the data in answer vector (“answer\_vec”) collected from the workers in the last round to a variable called “input\_string”. “input\_string” is started with two indexes stating the dataset ID and the task ID, followed by the answer vector. If it is the first round, answer vector would be void. Thus, “input\_string” will only contain the two indexes.



4. Use PHP to call MatLab and pass the txt to file to it as input.



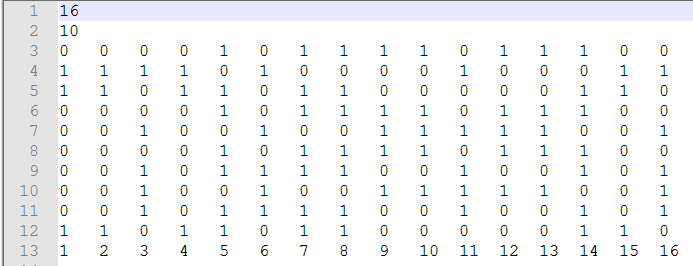
5. Use MatLab to generate a matrix based on the input → Use MatLab to write the output in an txt file.

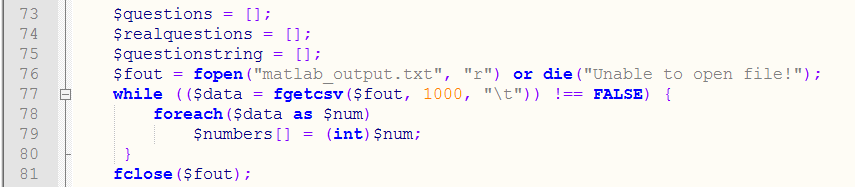
This part is done by a MatLab exe file named as “SQS” provided by Qiyu.



6. Use PHP to read the matrix.

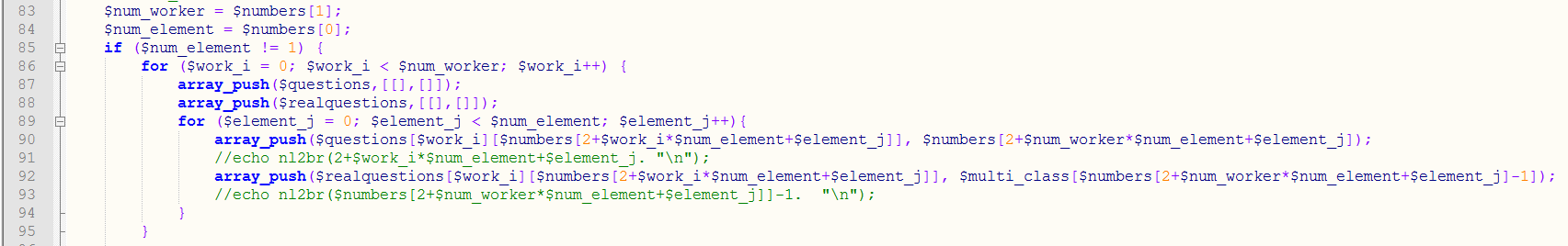
This is how the MatLab output looks like:





Open the txt file and read all numbers in sequence. All numbers are saved in a 1-D array named “numbers”.

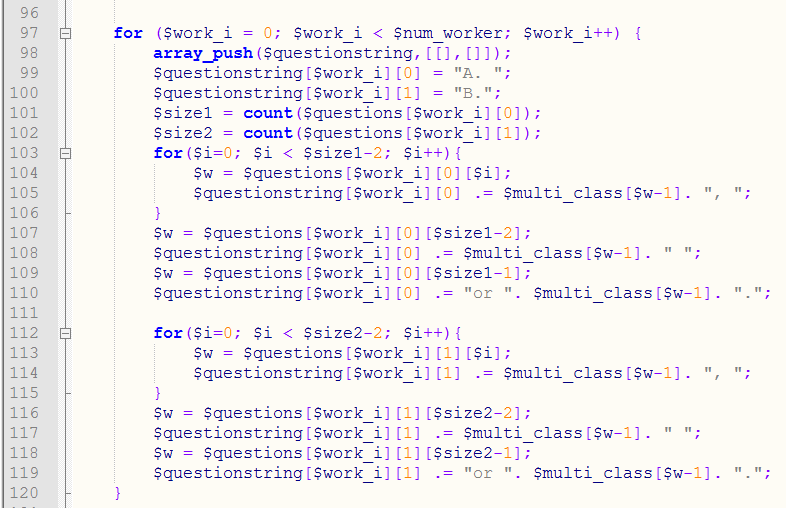
7. Use PHP to generate a 2-D array of string type contains both the choices (A and B) for all 10 questions.



First two elements of array indicates the number of species (16 in this example) ad the number of workers (10). They are stored in variable “num\_element” and “num\_worker” respectively. The nested “for” loop full in the two 3-D arrays “questions” and “realquestions”. The meaning of 3-D array “questions” is as following:

|  |  |  |  |
| --- | --- | --- | --- |
|  | i | j | Question[i][j] |
| Meaning | Order of worker  Range:0-9 | Option A (0) or B(1)  Range: 0-1 | An array contains species codes appears in the corresponding option |
| Example 1 | 0 | 0 | 1,2,3,4,6,11,15,16 |
| Example 2 | 0 | 1 | 5,7,8,9,10,12,13,14 |
| Example 3 | 1 | 0 | 5,7,8,9,10,12,13,14 |
| Example 4 | 1 | 1 | 1,2,3,4,6,11,15,16 |

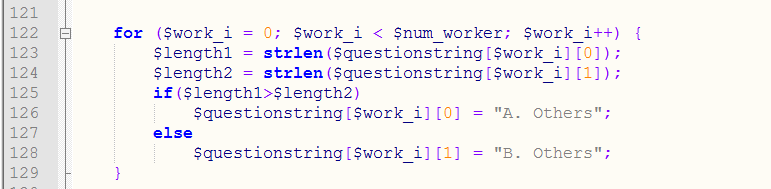
The array “realquestions” is similar to array “questions”. The only difference is that the species code are changed into real species names.



Create a 2-D string type array called “questionstring” containing the string for display on the workplace. The content is similar to 3-D array “realquestions”. Their difference is as following:

|  |  |  |  |
| --- | --- | --- | --- |
| Array name | i | j | Array\_name[i][j] |
| realquestions | Order of worker  Range:0-9 | Option A (0) or B(1)  Range: 0-1 | An array contains species names appears in the corresponding option |
| questionstring | Order of worker  Range:0-9 | Option A (0) or B(1)  Range: 0-1 | A finalized string of the option for display |

The string prototype: A. (B. )xxx, xxx, xxx, xxx, or xxx



Replace the content of longer option of every question by “Others”.

8. Send to the front end for display → Get the answer array from 10 workers

The codes shown following is only a demo for display, the real workplace webpage would be beautified by front end peers.



