**Mind Reader Game Project Report**

Course: CS3340.501 Professor: Nhut Nguyen

Team: The Shield Team Member: Chaoran Li, Yuer Jiang, Xue Cheng

Report Submitted by: Pengfei Tong

**Description**

The team project this semester is the Mind Reader game. The game is for a human player and our program is the 'mind reader'. The program will ask player to think of a number between 1 and 63. Then display the 6 cards, on each card there are 32 number displayed. Player asked to input ‘y’ or ‘n’ based on the whether the guessed number been displayed in the card. After the last card displayed, this program will get all the valid input from the player and correctly display the number that player has in mind. At the end of a game, the program will ask player if he/she wants to play another game and then repeat or end the program.

**Challenges and How We Overcame the**

**What I Have Learned**

**Algorithms and Techniques**

**Algorithms**

The mind read game seem magic but it has a very simple algorithms behind it. Computer displays six sets of numbers. For every i from 0 to 6, it displays the numbers with the same (either 0 or 1, selected randomly) binary digit in the i-th position counting from right. Answering "Yes" the player actually tell the program whether the digit is 0 or 1. Thus digit after digit, the program collects the valid information about the binary representation of the number that player had in mind. Since (111111)2 = 64 > 63, any number below 64 needs at most 6 bits to be written in the binary system.

For example, assume game player choose number 38. Its binary representation is 100110. I have to guess 6 binary digits. All numbers split into two groups: those that have the rightmost digit 0 and the rest that have the rightmost digit 1. Number 38 belongs to the first group. The program randomly select which group (below 64) to display. The player answer will be either "Yes" or "No", depending on which of the two groups actually shows up. But, regardless of which group is shown, once you press one of the buttons, The program can determine the rightmost digit of your number. With 38 in mind, if you see the first group, you will press the "Yes" button. If you see the second group, your response will be "No".

Other digits are determined in the exactly same manner. All numbers split into two groups: those that have their 3rd (from the right) digit equal to 0 (1, 2, 3, 8, 9, 10, 11, 16, 17, ...) and the rest that have 1 in the 3rd position (4, 5, 6, 7, 12, 13, 14, 15, 20, ...). The program select one of the groups randomly. Player answer will depend on which of the two groups has been selected. But regardless, pressing one of the buttons player tell the program which digit it is. For 38, if program display the first group, the answer must be "No". For the second group, the answer must be "Yes." In both cases we received the information to realize that the 3rd digit of your number is 1. Same thing, after 6 input we able to reveal the 6 binary digits which is the specific decimal number.

**Extra credit graphical display**

We used the bitmap pixel display, use pixel to draw number cube 45 x 225 pixel and word 8 x 16 pxiel for digits from 0 to 9. We able to call specific digits when we need to display it. Additional graphical display background is made and adjust to the proper pixel size to fit in bitmap. Different graphical picture content game information that ask game player for user input to make the game flow. Modified mars added Syscall 60 that made inserted picture background displayed properly.

**Extra credit Background music**

The purpose of the read number notes added in the program is to make the computer beep according to a text note file that is named 'Music.txt'. The first line in the text file is a three-digit number that corresponds to the number of notes going to be played, from 001 to 100. For each note it represent a different pitch and instruments tone. It knows the notes D, E, F, G, B, C, and C #. The sequence of different notes made up a background music been insert after the first and last graphical background are displayed. In the assembly code, Syscall 33 used that plays a note and waits to continue until after the beep.

**Ringtones**

The ringtones are carried out using the MIDI out system call by MIPS. Different pitches and instruments were used to distinguish between the ringtones of ‘y’ input, ‘n’ input and other invalid input.

**Contributions of Group Members**

Chaoran Li:

* + Peer evaluation: 10/10
  + Contributions of the Project: %
  + Contribution for Related work:
    - Write the algorithm for the random shuffle
    - Main algorithm in MIPS
    - Assemble all the modules and function from other teammates had finished

Yuer Jiang:

* + Peer evaluation: 10/10
  + Contributions of the Project: %
  + Contribution for Related work:
* Initiated the main algorithm in java
* Bitmap digits display
* Check code and debugging

Xue Cheng:

* + Peer evaluation: 10/10
  + Contributions of the Project: %
  + Contribution for Related work:
    - Main algorithm testing and debugging
    - User Manuel
    - Video demo

**Suggestion**