

PROJECT REPORT

MIND READER: [YOUTUBE](#)

MODIFIED MARS ASSEMBLER CODEBASE: [GITHUB](#)

PROGRAM DESCRIPTION:

Our project consists of four parts :

1. CALLER FUNCTION :

To generate cards and show in the random way we are using a random function which will generate a random number between 1 to 6 and sends the random number to the card generator function.

2. CARD GENERATOR:

We are using card generator function to generate cards to display. This generates a card and displays it to the user. We also receive input from the user whether the card has a number or not and save the answer.

3. CONVERTER:

Once six cards are displayed to the user and corresponding input for each card is stored. This module will convert the bits stored so far to create the decimal.

4. UI:

While displaying cards we have a colourful display for the numbers and implemented sound and music for different selections between yes and no buttons. Giving the user a nice experience while playing the game.

EXTRA CREDIT FEATURES:

1. Added GUI to the game
2. Added background music, that plays during the game

CHALLENGES:

1. Limited UI Capabilities:

- As MARS presents significant challenges in provision of custom UI and music
- I took a routed approach to provide these features by modifying the code base of MARS Assembler written in JAVA.
- I created three new syscalls that enable custom GUI and background music.
- I added a couple of class files to the mars assembler java codebase in order to provide these features.

FOLLOWING ARE THE DETAILS OF THE CLASS FILES ADDED :

SyscallGameDialog.java, SyscallWelcomeDialog.java, SyscallResultDialog.java

Please find the codebase here: [GITHUB](#)

2. Card Generator :

- Designed the algorithm to generate the numbers to be displayed in each card
- Refined the algorithm to generate the numbers without using recursion.

3. Aligning to Register Preservation principles:

- We learnt that register preservation is essential during subroutine calls to avoid any overwriting of register values, which may mess up the entire program
- However, with prior exposure solely to high-level programming, preserving and restoring register values between caller and callee routines was a challenge to implement.

LEARNINGS:

- Explored whole MARS Codebase and understood details of its multitudinous features
- Learned about how an assembly code written in mars editor is processed
- Explored MIPS architecture and its intricacies
- Became proficient in Assembly Language Programming
- Understood the design principles and conventions while using different registers.
- Learnt the proper use of Stack in MIPS.
- Helped me to gain insight into how MIPS code look.

ALGORITHMS

CallerFunction:

- Using syscall 42 we will give a range for which random number to be generated. We will do a modulus of 6.
- Simultaneously we will have an array of six positions all initiated with -1.
- The remainder we generated we will check corresponding positions in the array if it is not yet called it will then pass that number to card generator to display the card.

CardGenerator:

- **Core Logic:** In each card, we will display the numbers which have a particular bit set to 1 and the rest of the bits can be anything.

For example, out of 6 bits if you are picking a card which has a first bit set to 1. Then the numbers which have first bit 1 should be displayed like all the numbers from 32 to 63.

- How to get all the numbers having a particular bit to 1?

If we are doing for 3rd bit we will skip the first 8 elements and then add next 8 elements and then skip the Graphic/colour display

- Extra features of the programs (e.g. background music, sounds or music to indicate invalid input, pleasant display etc...) implemented and documented.

next 8 elements. Similarly, we do for other bits.

- Once we display the card we will record user reply and send it to converter function.

Converter Function:

- This function converts the bits generated by the user to the number which we need to display.
- The bits which need to be converted will be passed as an argument to the function. We will create a stack with numbers which are the powers of 2 which we will use for converting the Bits
- Then, we will iterate through both the bit stack and number array simultaneously and multiply to give the value for that particular Bit and add to sum which is initially 0. At the end of the loop, we will have the Bits converted to a value, which we will print to the user

CONTRIBUTION OF EACH TEAM MEMBER

1. **Dileep Siva Sai Nallabothu (DXN180025):** Card Generator, Documentation, Debugging.
2. **Rohith Peddi (RXP190007):** UI, Card Generator, Documentation(Lead the team for the entire project).
3. **Manideep Reddy Miryala (MXM190033):** Converter, Documentation, Debugging.
4. **Subhajit Chatterjee (SXC190070):** Random number generator, control flow, Video Making.

PEER GRADING

1. We worked as a team, all of them were very excited about doing this project and implementing new features to MARS.
2. We learnt a lot and felt that we can do many more wonderful things as a team