Number Tower User Manual

**SETUP**

* To start the program, navigate to the ‘src’ folder and open our custom MARS program simply named ‘Mars.jar’.
* Go to the File tab and open ‘pyramids.asm’.
* Go to the Tools tab and open the Number Pyramids extension.
  + Click Connect to MIPS

Build the program and run.

**GAMEPLAY**

* Upon running the program, the boxes in the number tower will populate with known and unknown values, displayed in both the graphical extension as well as the console.
* Each box will contain the sum of the two boxes underneath it. The goal of the game is to fill in all the empty boxes and complete the number tower.
* The console will prompt you for an input location. This input will only accept two capital letters, denoting the row and index of the box you wish to fill in.
  + Input is processed immediately without the need to press enter.
* Once a valid location has been selected, you will be prompted to input the answer you wish to place in the box. Your answer must be a positive integer and can be anywhere from one to three digits.
  + Input your answer by pressing enter.
  + Three-digit answers are processed immediately.
* If your answer is correct, a jingle will play and the number will be filled into the number tower.
* If your answer is wrong, an error sound will play and you will go back to choosing a location.
* Once all boxes in the number tower have been filled, you win the game!

**EXITING**

* To exit the program safely, stop the program and then close MARS.

Number Tower DEVELOPER Manual

**SETUP**

* To start the program, navigate to the ‘src’ folder in terminal and enter ./createJar.sh to run the packaged version of Mars with custom source code. You will need to run this every time you wish to launch mars and see output for debugging or whenever you make changes to the source code.
* Under mars/tools/ you will find the extensions. Look for NumberPyramids.java, and then scroll down to the processMIPSUpdate() method. This is where the magic happens.
* You can see the custom annotations for @DRAW and @MUSIC-START/MUSIC-STOP.
* You can make your own annotations here for whatever MIPS command you want to implement that a higher-level language like java offers such as networking or music or graphics.

**GRAPHICS API**

* To use the graphics API we provided, you must do three things. One, push the value that the pyramid block should contain such as 120 or 335, the number inside it to $a0. Two push the block location from an index 0-27 with 0 being the top block being index 0 and the bottom right being index 27 to $a1. Then finally place any random MIPS command with # @DRAW attached to the end of the command.  
    
  Ex. move $a0, $s1, # $s1 contains the number like 120 or 335

move $a1, $s2, #$s2 contains the number 0-27

li $s1, 0, # @DRAW

**SOUND API**

* To use the sound API we provided, you must do three things. One place a .wav file containing the game music you want to use under src/mars/tools and call it bg.wav. Then similar to the Graphics API you just have to attach @MUSIC-START to the end of any MIPS command when you want to start the song and @MUSIC-STOP when you want to end the music or exit the program.
* To use the sound functionality of the MARS simulator, simply call the “playsound” subroutine after loading a pitch and duration of the sound desired into $a0 and $a1 respectively. More detail can be found in the project report, under the “Bonus > Sound” heading within the algorithms discussion.