Instructions:

1. Go to tools, and connect both “Bitmap Display” and “Keyboard Display and MMIO Simulator” to MIPS.
2. Make sure you give pixel unit height and width if 4, and display width and height of 64.
3. Go to the main function and run the program. You will be asked in a pop up to be Player 1, Player 2, or exit the game (Note: if you exit the game, you will have to recompile).
4. The first pop up will tell you “Player 1?”. Select “Yes” if you want to be player 1, select “No” if you want to be player 2, select “Cancel” if you do not want to play at all.
5. Play the game: move up, down, left, and right with keys W, A, S, and D. The console will output your score each time you capture a bit.
6. If you win or lose, the console will display a message telling you your result. There will also be a pop up that does the same (here just click OK).
7. You will be asked if you want to play again. Select “Yes” to play again, select anything else to quit.

Overview:

- In this game, the objective is to use the bit with the marquee effect (this is your character) to catch non-moving bits across the bitmap display. Therefore, we will be making use of the bitmap and keyboard.
- My program starts off with a box with 3 options, asking if the player wants to be player 1, player 2, or not play the game at all. If the player wishes to play the game, we proceed with the code. If the player wishes not to play the game, a pop up will tell him or her “Goodbye” and end the program.
- Regardless of whether player chose to be player 1 or player 2, the run of the game starts with a countdown. We will call a macro function to do this countdown. The countdown is “3..2...1”, where the numbers will flash once in yellow before fading away. This was done by drawing the relevant number in yellow, and fading it by delaying and drawing over it in black
- The code later branches out to player 1 call or player 2 call, based on user selection. If the user chose player 1, we call the macro that will output player 1’s scenario. If the user chose player 2, we call the macro that will output player 2’s scenario.
- In both player 1 and player 2 scenarios, the user will move up, down, left, and right with WASD keys. The user can hit space for a special call, which will happen out of luck. Here, the user can either win or lose the game by chance.

- In both player 1 and player 2 scenarios, there will be a bit placed on the left, right, top, or bottom edge of the bitmap display. It will be randomly placed based on the number of iterations passed so far.
- If the user chose to be player 1, he or she will have to mainly capture the white bits. Every time the user hits a white bit, his or her score will increase by 1. Every now and then, there will be a red bit which will double score, triple score, add 100, or take off 100 from the current score. If the user's score exceeds 100, he or she wins the game. If the user's score falls less than -100, he or she loses the game.
- If the user chose to be player 2, he or she will have to mainly capture the red bits. Every time the user hits a red bit, his or her score will decrease by 1. Every now and then, there will be a white bit which will multiply score by -2, multiply score by -3, subtract 100 from score, or add 100 to score. If the user's score falls below -100, he or she wins the game. If the user's score exceeds 100, he or she wins the game.
- Every time the user's score is updated, it will be printed in the console on its new line. Keep in mind that result from hitting space can alter the win or loss of the game if it does by any chance. The user can also have a certain number of points and still lose the game or vice versa, win the game.
- If the user has won or lost the game, the macro function of displaying a win or loss result will be shown in the console. Furthermore, the same message will be displayed in a pop-up, with its score following the line.
- After the win or loss message, the user will be asked if he or she wants to play the game again. If yes is selected, it will reiterate, asking the user which player to play the game as, or not play the game at all. If any other button is selected, a pop-up tells the user "Goodbye" and then the program ends. The program as a whole keeps looping until the user no longer wishes to play.

Pseudocode:

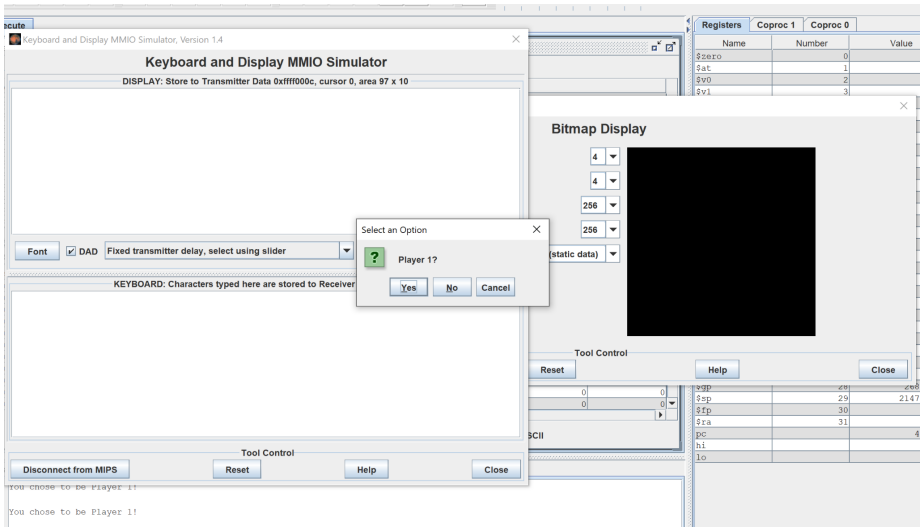
```
Start the program
Ask user to be player 1
If "Player 1":
        Show the countdown intro
        While Player 1 didn't win or lose game
                W is pressed:
                        Move player up
                A is pressed:
                        Move player left
                S is pressed:
                        Move player down
```

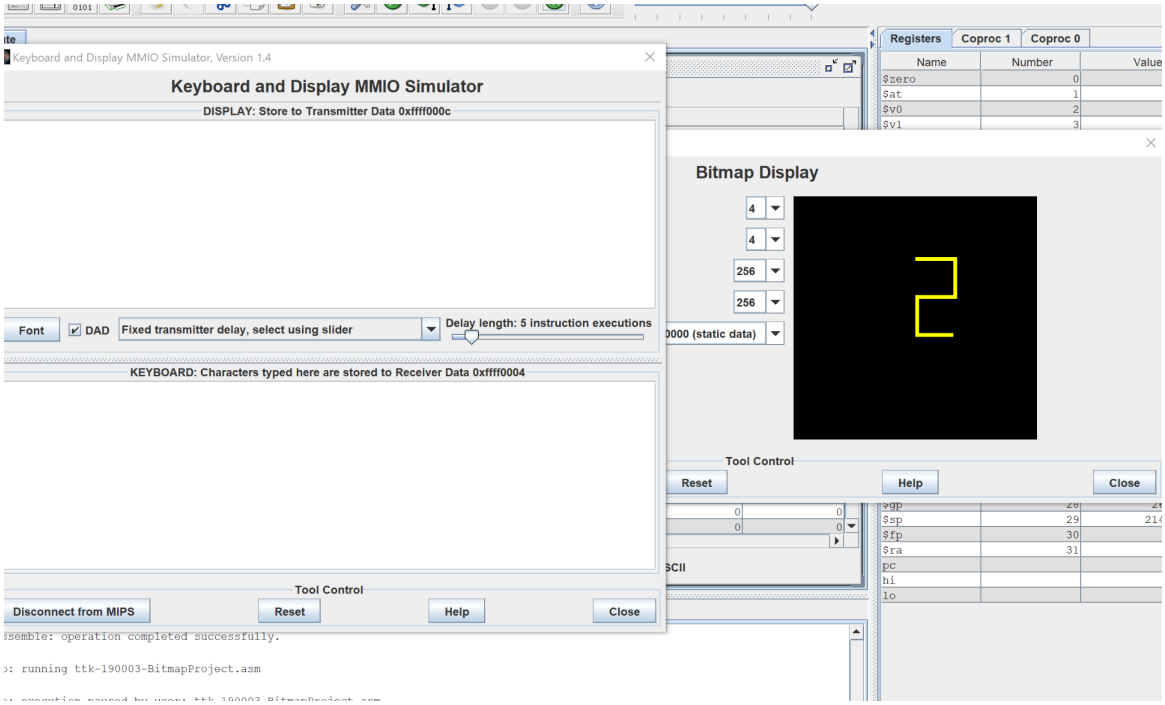
```
            D is pressed:
                Move player right
            Space is pressed:
                If score <= 5:
                    Jump back to while loop
                If score % 5 == 0:
                    Player 1 has won the game!
                    Jump out to ask to play again
                Else:
                    Player 1 has lost the game!
                    Jump out to ask to play again
            White Pixel is hit:
                Increment score by 1
                If score >= 100:
                    Player 1 has won the game
                    Jump out to ask to play again
            Red Pixel is hit:
                If count % 10 == 0:
                    Double score
                Else if count % 5 == 0:
                    Triple score
                If score < 100
                    Score += 100
                Else if score > 100
                    Score -= 100
                If score >= 100:
                    Player 1 has won the game
                    Jump out to ask to play again
                If score <= -100:
                    Player 1 has lost the game
                    Jump out to ask to play again
            Increment count of captured bits by 1
Else if “Player 2”:
        Show the countdown intro
        While Player 2 didn’t win or lose game
            W is pressed:
                Move player up
            A is pressed:
                Move player left
            S is pressed:
```

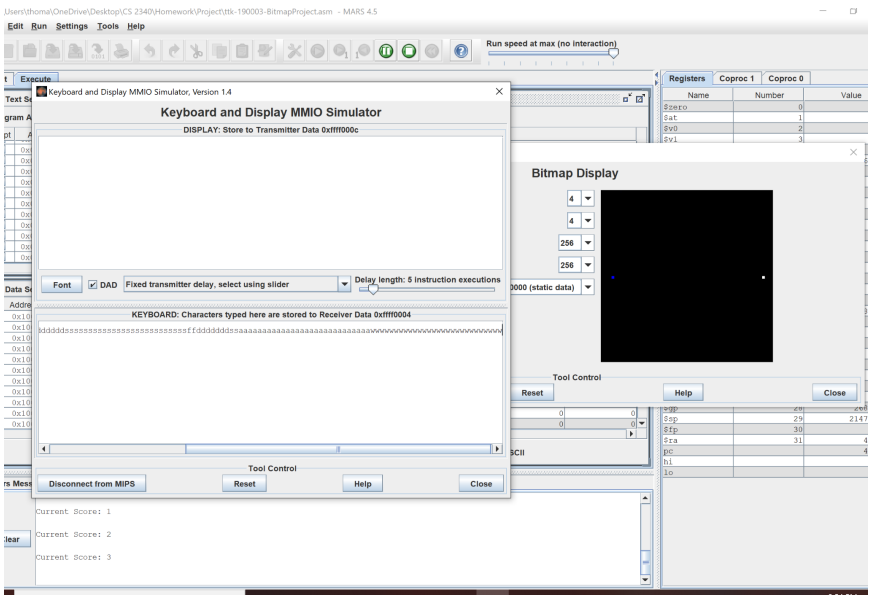
```
            Move player down
        D is pressed:
            Move player right
        Space is pressed:
            If score <= 5:
                Jump back to while loop
            If score % 5 == 0:
                Player 1 has lost the game!
                Jump out to ask to play again
            Else:
                Player 1 has won the game!
                Jump out to ask to play again
        Red Pixel is hit:
            Increment score by -1
            If score <= -100:
                Player 1 has won the game
                Jump out to ask to play again
        Red Pixel is hit:
            If count % 10 == 0:
                Score *= -2
            Else if count % 5 == 0:
                Score *= -3
            If score < 100
                Score += 100
            Else if score > 100
                Score -= 100
            If score <= -100:
                Player 1 has won the game
                Jump out to ask to play again
            If score >= 100:
                Player 1 has lost the game
                Jump out to ask to play again
        Increment count of captured bits by 1

Else:
    Jump to end program
If user wants to play again:
        Jump to start of program
    Else:
        Jump to end
```

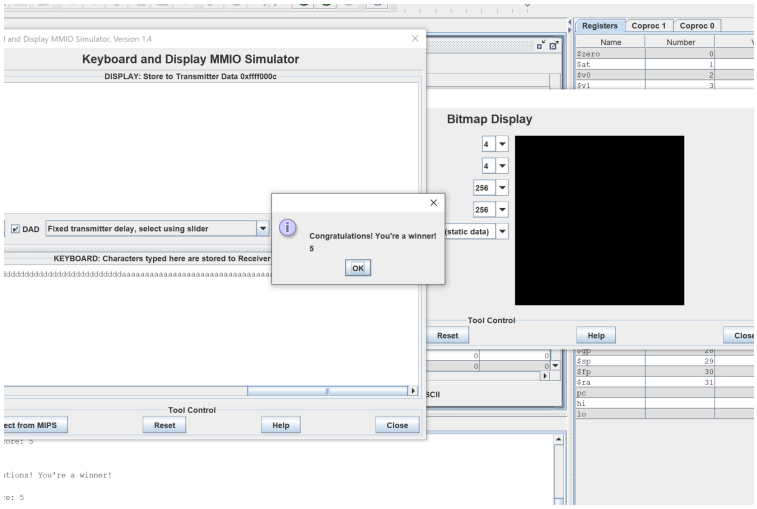
End the program

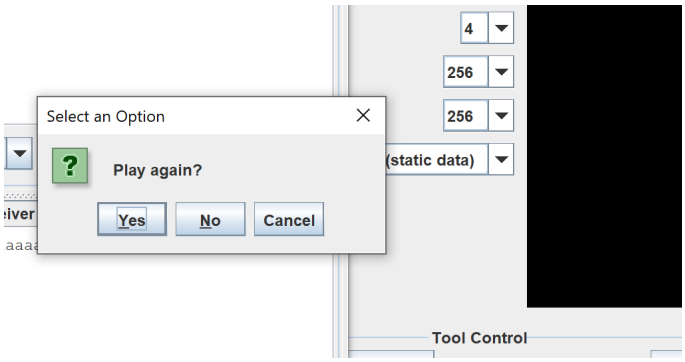
Sample Run Screenshots:

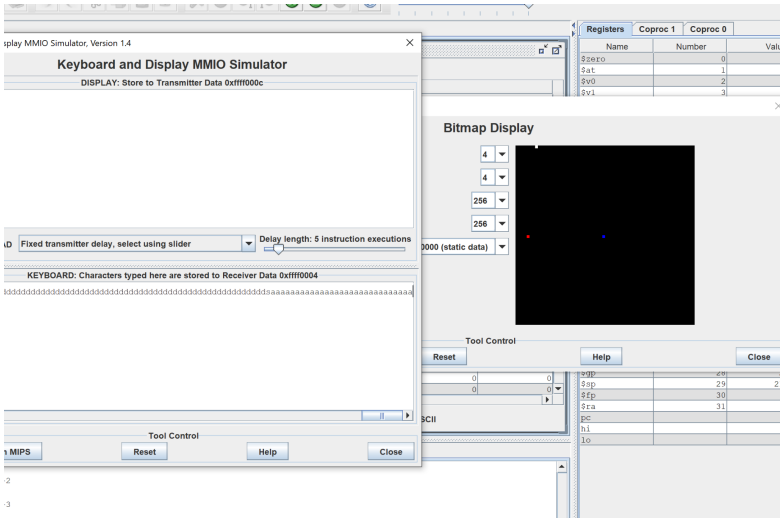


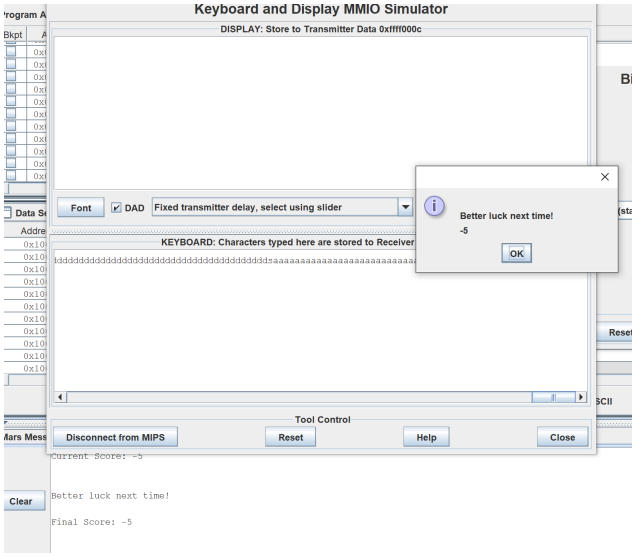


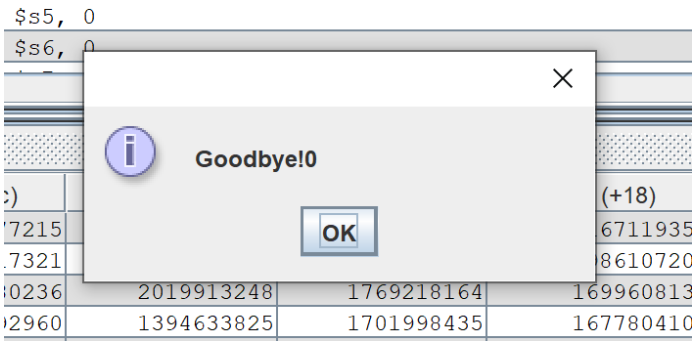

Keyboard and Display MMIO Simulator, Version 1.4
Keyboard and Display MMIO Simulator
DISPLAY: Store to Transmitter Data 0xffff000c
Font
DAD
Fixed transmitter delay, select using slider
Delay length: 5 instruction executions
KEYBOARD: Characters typed here are stored to Receiver Data 0xffff0004
Tool Control
Disconnect from MIPS
Reset
Help
Close
Bitmap Display
Current Score: 1
Current Score: 2
Current Score: 3


Keyboard and Display MMIO Simulator
DISPLAY: Store to Transmitter Data 0xffff000c
Congratulations! You're a winner!
5
OK
Bitmap Display
Tool Control


Select an Option
Play again?
Yes
No
Cancel
Tool Control







Warnings:

- Make sure the “Bitmap Display” and “Keyboard Display and MMIO Simulator” are all connected to MIPS. If not, the game cannot be played properly.
- Make sure you give pixel unit height and width if 4, and display width and height of 64. Otherwise, there will be absurd displays in the bitmap display.

- If a user has elected to play as “Player 2”, make sure he or she captures every bit. Extra bits can overlap in future runs of the game. Some bits may or may not be fully blacked out due to miscellaneous inconveniences that do not interfere with the program’s functions, but may cause a slight user disconvenience.
- At one point, the current score may stop displaying on its own line. Normally it is supposed to be displayed (for example: “Current Score: 3”), but sometimes it will just display the number, and all other following numbers of the score on the same line. Again, this only causes a user inconvenience.
- In the worst cases, pop ups may not display strings at all, they may only display numbers in win or loss pop up messages at the end.
- This is a friendly reminder but at the beginning, in the pop up displaying “Player 1?” string, hit “Yes” to choose to play as player 1, “No” to choose to play as player 2, or “Cancel” to not play the game at all.
- If you click X on pop ups with “Yes”, “No”, “Cancel” options of buttons, it will automatically take the answer as if “Yes” was clicked.
- Keep in mind that playing as player 2 has the exact opposite game objectives as that of player 1. When playing as player 1, white bits are usual bits whereas red bits are special bits; and this is vice versa for player 2. 100 points wins the game as player 1 and loses the game as player 2, -100 points loses the game as player 1 and wins the game as player 2.