The program implements the 2048 game and display the value using colorful blocks with distinct sizes.

# How to Play

## Goal

The goal of the game is to generate the highest level of block by merging from those smaller blocks.

## Rule

Each turn a new level-1 block will be generated, and player can enter the moving direction of the current distribution. Blocks will move to the specified direction if there is not a different level block or frame on their way. If two same level blocks collide, the two blocks will merge into a higher-level block.

The first-time creation of each level of merged blocks, an audio will be play as a congratulating feedback to player. The player wins if the highest level of block is generated. On the contract, if there is no available moving directing can apply on any of block, the game will end with failure.

The actual value of blocks corresponding to the 2048 game will be printed into the I/O console during the gaming process.

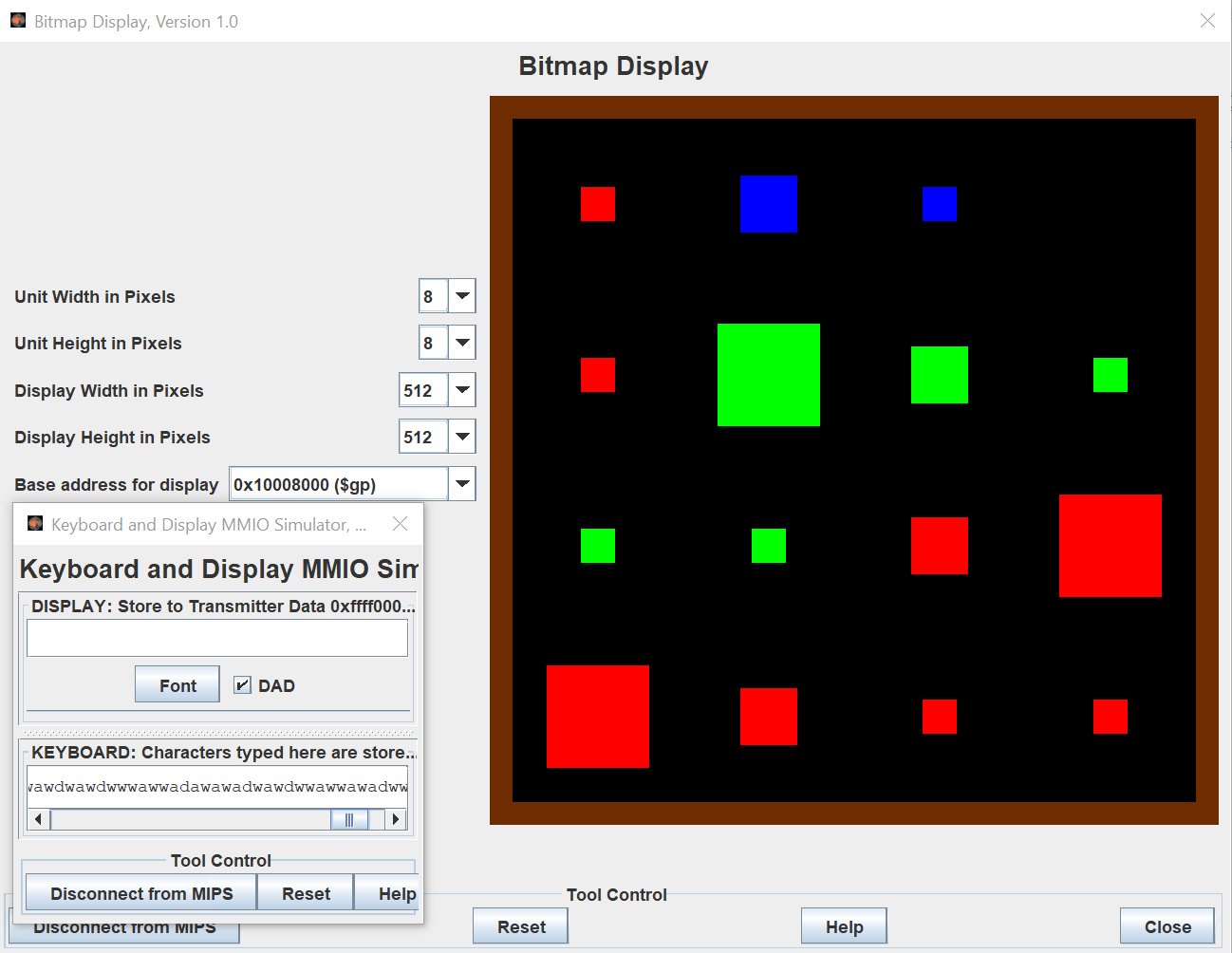
# How to Run

1. Bitmap Display
   1. unit width: 8
   2. unit height: 8
   3. display width: 512,
   4. display height: 512,
   5. base address: $gp
2. Keyboard and Display MMIO Simulator: control the motion of the blocks
   1. up: "w",
   2. down: "s",
   3. left: "a",
   4. right: "d"

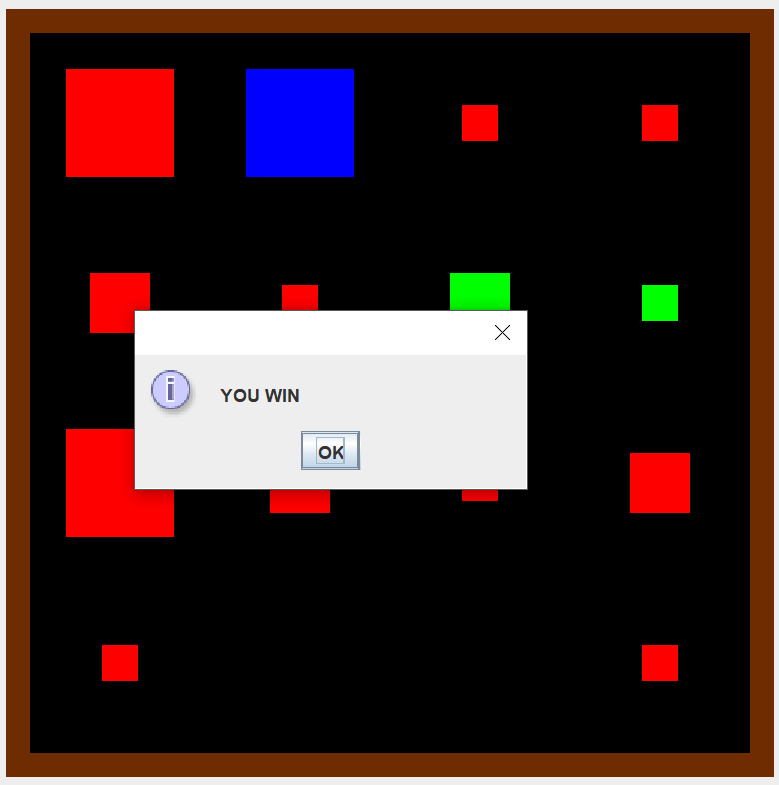
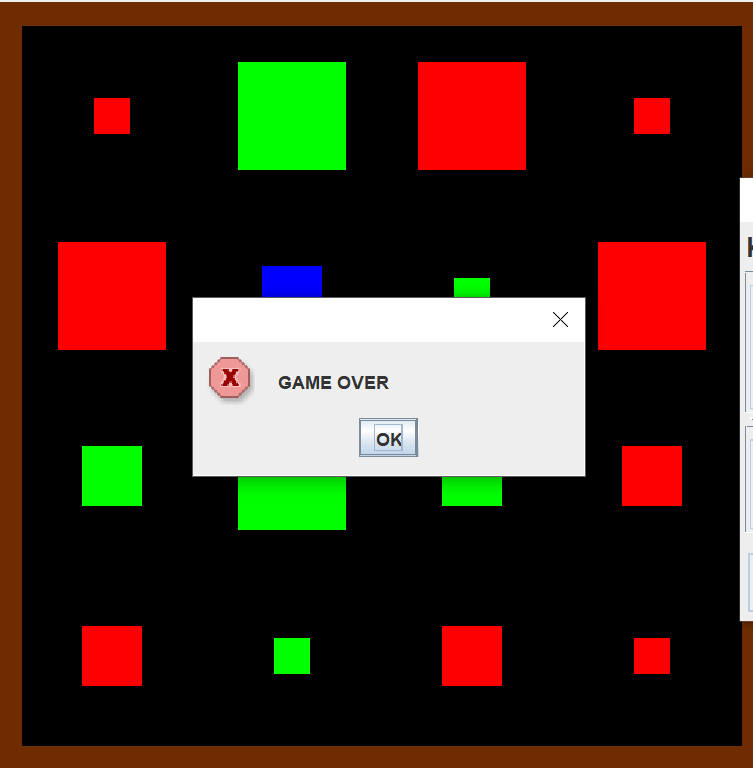
Note: Speed up the animation by adjusting the “moving\_step” in data section to “3” if Monitor flashing rate is too low

# Program Screen Shots

The blocks display in the bitmap display. The keyboard and display MMIO simulator is used to type input of moving direction.



Two different game ending situations. Left figure shows the highest level blue block is generated. Thus, the play win in the condition. Right figure shows no more direction can make any motion for the any of blocks. It means the player fail in the game.

The actual value record is printed into the console I/O panel.

