Project Report: Milestone 1

Janssen Myer Rambaud (1008107004), Felix Zhang (1007650212)

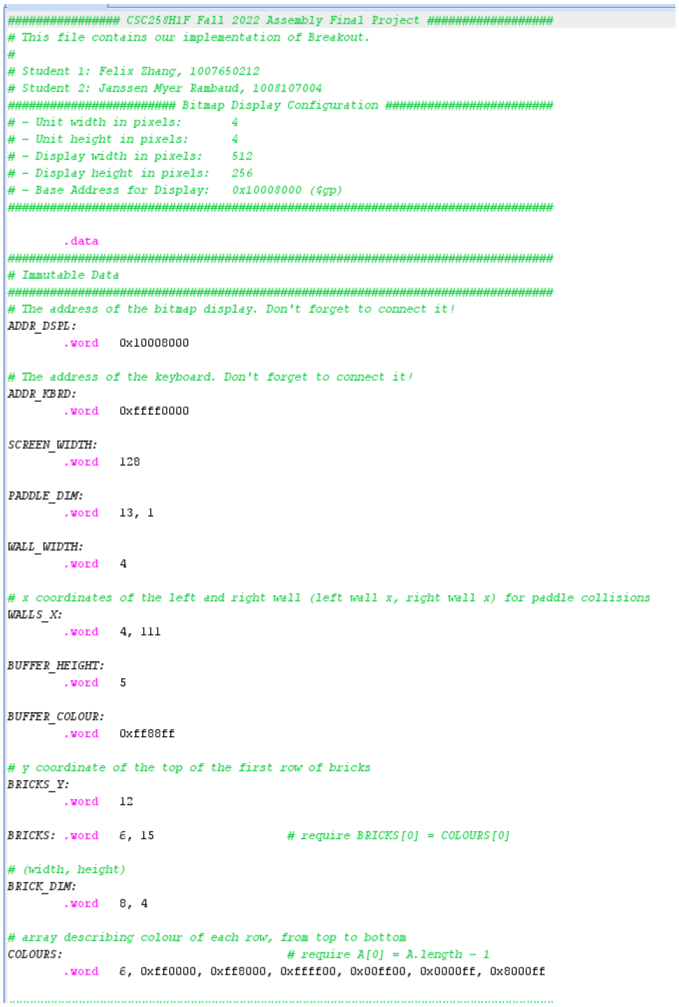
November 29, 2022

1 Part I: Planning and Configuration

1. **Breakout Bitmap Display and Memory Map Plan**: Our plan is to have a 512 pixel (wdith) x 256 pixel (height) display while keeping the same base address 0x10008000 ($gp). We also plan on having unit widths be 4 pixels x 4 pixels because it remains even while being a better size than say 2 x 2, which is too small. A table of our memory map plan is given below.

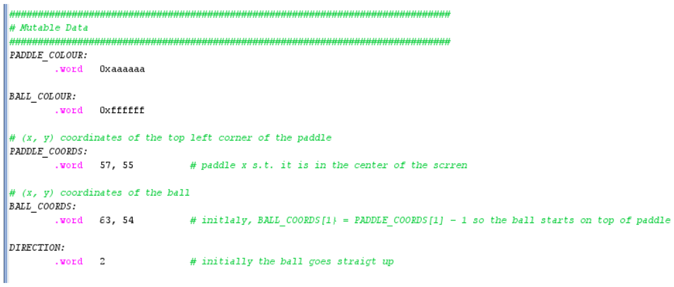
Data Segment	Address	Variable Names
IMMUTABLE DATA		
	0x10008000 (Address of Bitmap Display)	ADDR_DSPL
	0xffff0000 (Address of Keyboard)	ADDR_KBRD
	Takes in 1 address (Display width divided by unit width [in pixels])	SCREEN_WIDTH
	Takes in 2 addresses for (length, height) of the paddle	PADDLE_DIM
	Takes in 1 address for width of wall	WALL_WIDTH
	Takes in 2 addresses (left wall x-coordinate, right wall x-coordinate)	WALLS_X
	Takes in 1 address for (height) of buffer	BUFFER_HEIGHT
	Takes in 1 address buffer colour (used for drawing walls, bricks)	BUFFER_COLOUR
	Takes in 1 address for the y coordinate of a brick	BRICKS_Y
	Takes in 2 addresses (2-D array that stores colour)	BRICKS
	Takes in 2 addresses (width, height) for a brick	BRICKS_DIM
	Takes in 7 addresses (1 for array length, 6 for elements/colours)	COLOURS
MUTABLE DATA		
	Takes in 1 address for colour of paddle (gray)	PADDLE_COLOUR
	Takes in 1 address for colour of ball (white)	BALL_COLOUR
	Takes in 2 addresses (x, y) coordinates	PADDLE_COORDS
	Takes in 2 addresses (x, y) coordinates	BALL_COORDS
	Takes in 1 address for the movement direction of the ball	DIRECTION

2. Translate your plan into the .data section of your breakout.asm program. Assemble your program in MARS and inspect memory to ensure it matches your plan. Include a screenshot (or multiple screenshots) of memory demonstrating that it has been laid out according to your plan.



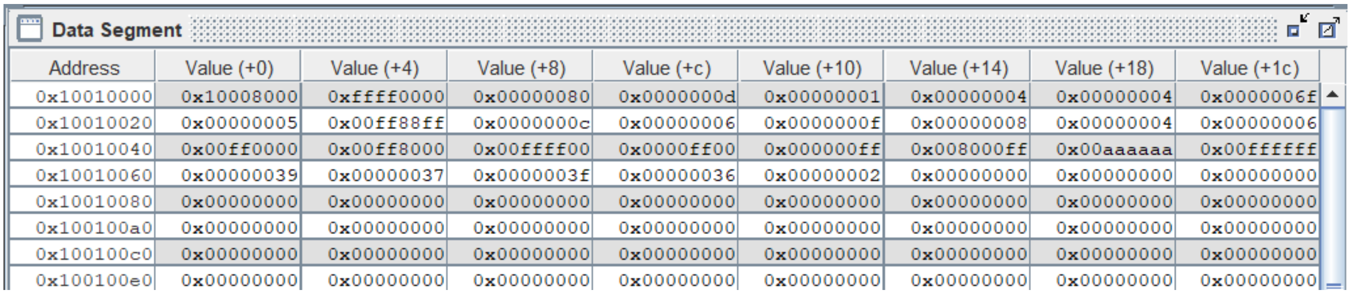
```
################ CSC258H1F Fall 2022 Assembly Final Project ##################
# This file contains our implementation of Breakout.
#
# Student 1: Felix Zhang, 1007650212
# Student 2: Janssen Myer Rambaud, 1008107004
######################## Bitmap Display Configuration ########################
# - Unit width in pixels:       4
# - Unit height in pixels:      4
# - Display width in pixels:    512
# - Display height in pixels:   256
# - Base Address for Display:   0x10008000 ($gp)
##############################################################################

        .data
##############################################################################
# Immutable Data
##############################################################################
# The address of the bitmap display. Don't forget to connect it!
ADDR_DSPL:
        .word   0x10008000

# The address of the keyboard. Don't forget to connect it!
ADDR_KBRD:
        .word   0xffff0000

SCREEN_WIDTH:
        .word   128

PADDLE_DIM:
        .word   13, 1

WALL_WIDTH:
        .word   4

# x coordinates of the left and right wall (left wall x, right wall x) for paddle collisions
WALLS_X:
        .word   4, 111

BUFFER_HEIGHT:
        .word   5

BUFFER_COLOUR:
        .word   0xff88ff

# y coordinate of the top of the first row of bricks
BRICKS_Y:
        .word   12

BRICKS: .word   6, 15                   # require BRICKS[0] = COLOURS[0]

# (width, height)
BRICK_DIM:
        .word   8, 4

# array describing colour of each row, from top to bottom
COLOURS:                                # require A[0] = A.length - 1
        .word   6, 0xff0000, 0xff8000, 0xffff00, 0x00ff00, 0x0000ff, 0x8000ff
```

Figure 1: Screenshot of immutable variables in the .data code section.



```
##############################################################################
# Mutable Data
##############################################################################
PADDLE_COLOUR:
        .word   0xaaaaaa

BALL_COLOUR:
        .word   0xffffff

# (x, y) coordinates of the top left corner of the paddle
PADDLE_COORDS:
        .word   57, 55          # paddle x s.t. it is in the center of the scrren

# (x, y) coordinates of the ball
BALL_COORDS:
        .word   63, 54          # initlaly, BALL_COORDS[1] = PADDLE_COORDS[1] - 1 so the ball starts on top of paddle

DIRECTION:
        .word   2               # initially the ball goes straigt up
```

Figure 2: Second screenshot of mutable variables in the .data code section.

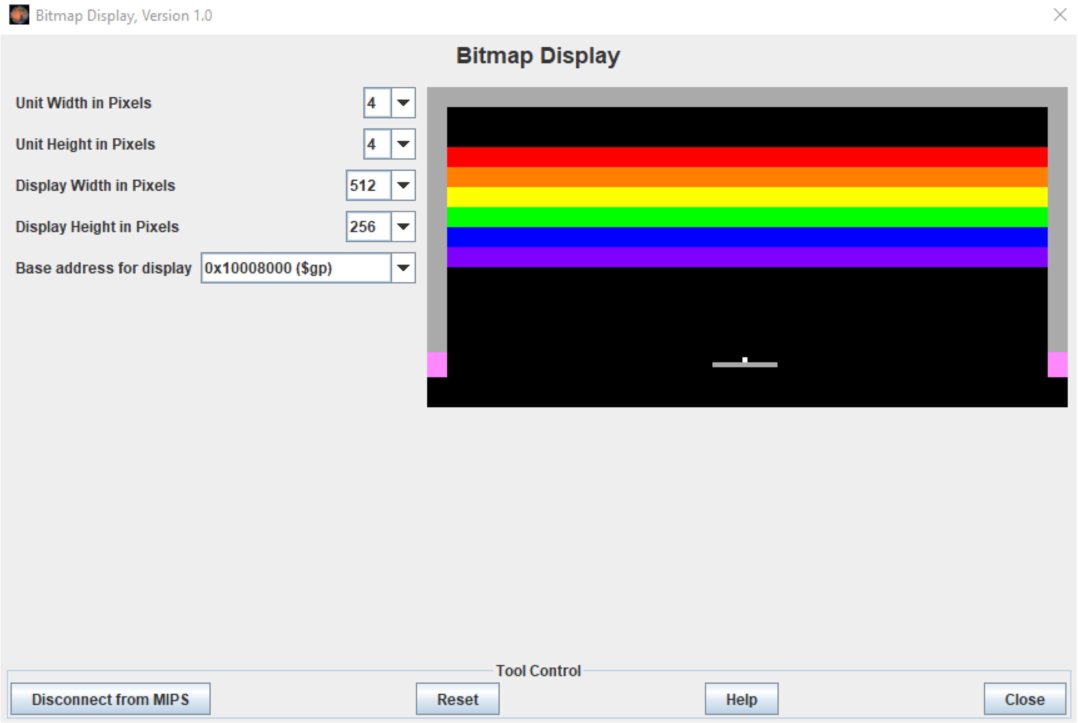


Address	Value (+0)	Value (+4)	Value (+8)	Value (+c)	Value (+10)	Value (+14)	Value (+18)	Value (+1c)
0x10010000	0x10008000	0xffff0000	0x00000080	0x0000000d	0x00000001	0x00000004	0x00000004	0x0000006f
0x10010020	0x00000005	0x00ff88ff	0x0000000c	0x00000006	0x0000000f	0x00000008	0x00000004	0x00000006
0x10010040	0x00ff0000	0x00ff8000	0x00ffff00	0x0000ff00	0x000000ff	0x008000ff	0x00aaaaaa	0x00ffffff
0x10010060	0x00000039	0x00000037	0x0000003f	0x00000036	0x00000002	0x00000000	0x00000000	0x00000000
0x10010080	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100a0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100c0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000
0x100100e0	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000	0x00000000

Figure 3: Screenshot of .data memory.

2 Part II: Milestone 1

3. Draw the scene (Milestone 1)



Bitmap Display, Version 1.0

Bitmap Display

Unit Width in Pixels 4

Unit Height in Pixels 4

Display Width in Pixels 512

Display Height in Pixels 256

Base address for display 0x10008000 ($gp)

Figure 4: The static scene of Milestone 1 Drawing.

3 Part III: Milestone 2

QUESTION: How will the ball change directions when it collides?

The rules governing how the ball changes directions when it collides are as follows.

If it collided with the paddle, then the resulting direction depends on which section of the paddle it hit. Hitting the leftmost third results it travelling north-east (at a 45° angle), the middle third results it travelling north, and the rightmost third results in it travelling north-west (at a 45° angle).

Otherwise, if it collided with a non-paddle object then the resulting direction depends on the original direction. If the original direction of ball forms the angle θ with the surface it collided with, then the resulting direction is the reflection of the original direction across the line perpendicular to the collision surface away from the collision point. For example, if the ball is travelling north-west (at a 45° angle) and collides with the left wall, then it begins travelling in the north-east direction (at a 45° angle). Note this also means if the ball was travelling perpendicular to surface with which it collided, then after the collision, it is moving in the reverse direction. So, if the ball was travelling north before colliding with the bottom of a brick, then it is travelling south after the collision.

Upload breakout.asm to MarkUs so that you have a snapshot of your progress so far.

4 Part IV: Milestone 3

Upload breakout.asm to MarkUs so that you have a snapshot of your progress so far.