Project Report: Milestone 1 and 2

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

December 5, 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          # initialy, 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.

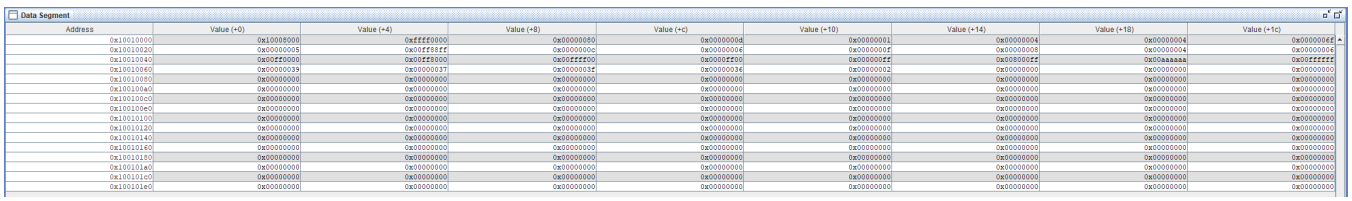

Figure 3: Screenshot of .data memory.

2 Part II: Milestone 1

3. Draw the scene (Milestone 1)

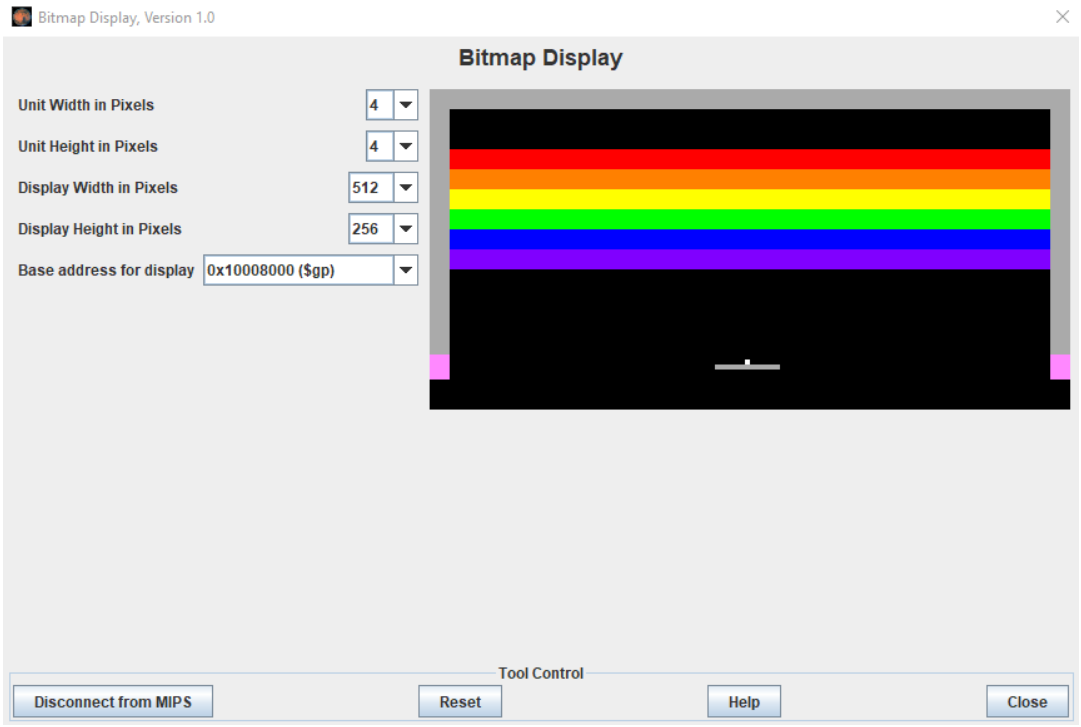

Figure 4: The static scene of Milestone 1 Drawing.

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.

5 Part V: DEMONSTRATION 2 Milestone 4 and 5

For our Milestones 4 and 5, we decided to do (a cumulative) 3 easy features and 2 hard features.
We chose the easy features: 1 (multiple lives), 5 (pause game), and 9 (launch ball) and hard features: 1 (track score) and 3 (bricks need multiple hits).

Detailed Changes Made:

1. (EASY) 1 - Multiple Lives: There are three hearts at the top left corner which correspond to the amount of lives you have. You start with three lives and once you lose all your hearts (lives), then the game ends. If you "lose" and still have lives left, then you will restart from the initial position where you will have to launch the ball again to CONTINUE playing with the same amount of bricks broken as before.
2. (EASY) 5 - Pause Game: This is a simple feature where any time the user clicks the exact key 'p', then the game will enter a 'paused' state where there is no movement (from player or game). In order to continue playing, the user must hit any other key, other than 'p', in order for the game to resume playing state and the ball will once again move.
3. (EASY) 9 - Launch Ball: At the beginning of every game, or at the beginning of every "new life" when the player has been initialized at the starting position (ball and paddle are in the middle of the screen), then they must launch the ball by pressing the spacebar in order to start the game.
4. (HARD) 1 - Track Score: For every brick hit by the ball, the score on the top right of the screen will increment by 1. In correlation to Multi-Hit Bricks, a 'brighter' coloured brick, once fully broken, awards the player 2 points. After a 'brighter' brick is hit by the ball, it's broken into a 'darker' brick [+ 1 points], and once that 'darker' brick is hit by the ball, it is completely destroyed [+ 1 points].
5. (HARD) 3 - Multi-Hit Bricks: There are 6 different colours of bricks in this game. There are red, green, and blue coloured bricks as well as their 'darker' counterparts. When the brick is 'brighter' it means that it requires two hits to break. When a 'brighter' brick is hit once, it turns into a 'darker' brick of that same colour. Once a 'darker' brick is broken, then it is deleted/destroyed from the game.

6 Part VI: HOW TO PLAY

GAME LAYOUT:

- Once you assemble and run the program, you will be met with the game screen as well as the 'STATS' section on the top of the game which contains your lives (hearts) as well as your score in numbers.

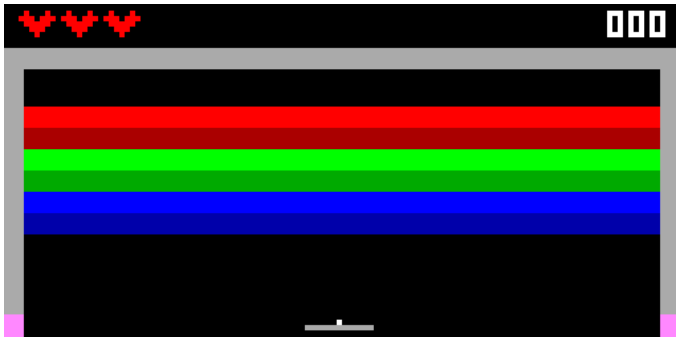

Figure 5: This is the Game's Initial Start Screen

START GUIDE:

- You begin with 3 lives represented by hearts (top-left of screen).
- In this game you will be chasing after the ball (white square) with your paddle (grey rectangle) and attempt to continue bouncing it off the paddle to aim and destroy the coloured blocks above you.
- To begin you must launch the ball from the paddle using the [Spacebar].
- Your paddle can move left and right using keys [a] for left and [d] for right, but CANNOT go past the walls OR go up and down, this is your paddle's boundary.
- The ball can go anywhere within the grey boundary, but if the ball reaches the bottom of the screen, which is past the paddle, then you lose a life.
- If you reach this bottom of the screen and lose a life, it will be reflected on the top left of your screen and you will lose a heart as well. If you have remaining lives, then you will restart from the middle of the screen just like when you first started the game.
  Note: You will continue your progress and the amount of bricks broken, as well as your score, will not be reset.
- There are two types of bricks, 'brighter' bricks and 'darker' bricks. For each hit on either brick, you will receive 1 point, which will be reflected in your score on the top-right of the screen. Once hit, a 'brighter' brick will be broken down (transformed) into a 'darker' brick. Once a 'darker' brick is hit, then that brick is 'destroyed' and will turn into empty space that the ball can move through.
- The game ends when you have broken every single brick leaving you with an empty space to freely practice your paddle skills with the ball.

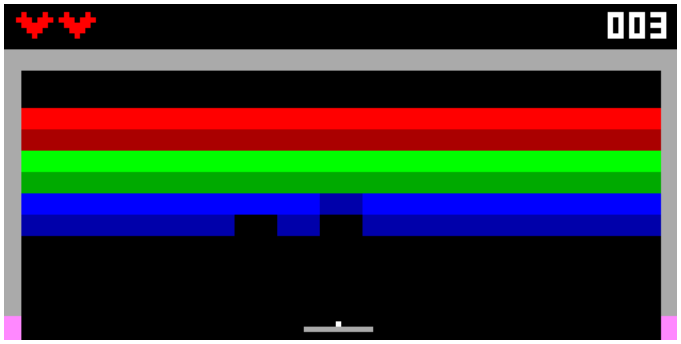

Figure 6: This is what it looks like if the player broke some blocks then lost a life.

FEATURES:

- Score: Increments by 1 for every block broken.
- Lives: 3 chances to complete the game, every time the ball touches the bottom of the screen, you lose a life. Lose all lives and the game ends.
- Collision: The paddle will collide with the wall and cannot go past it. The ball will collide with the paddle or wall, bouncing off of it.
- Pause Game: You can enter a paused game state pressing the key [p]. Exit pause with any key aside from [p].
- End Game: You can exit/terminate the game when you press either of the two keys, [q] or [x].

SUMMARIZED GAME CONTROLS:

PADDLE MOVEMENT

a Paddle Left

d Paddle Right

LAUNCH BALL

spacebar Launch Ball at start (Goes Straight Up)

PAUSE/UNPAUSE GAME

p Press Once to Set The Game into a Paused State

Any Key Any Key Aside from [**p**] will UnPause The Game and Continue the Ball's Movement.

EXIT GAME

q Press Once to Exit Game

x Press Once to Exit Game

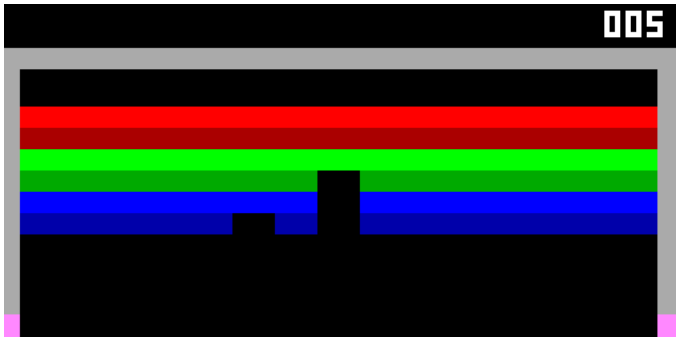

Figure 7: Game Over! The player lost all their lives and the game is finished. This is their final score.