CSC258 Assembly Project Report

Jason Wang (1008584649)

29 March 2023

Contents

Milestone 1

Constants and Variables

The following are the constants defined in memory:

1. `DISPLAY_WIDTH = 256` (width of display in pixels)
2. `DISPLAY_HEIGHT = 256` (height of display in pixels)
3. `ADDR_DSPL = 0x10008000` (address of the bitmap display)
4. `ADDR_KBRD = 0xffff0000` (address of the keyboard)
5. `BG_COLOUR = 0x000000` (colour code of the background)

6. `FRAME_TIME = 60` (sleep time between frames in ms)
7. `HEART_ADDR = [520, 556, 592]` (offset from `ADDR_DSPL` of the top left corner for each heart)
8. `HEART_FULL_COLOUR = 0xe54b4b` (colour of heart when full)
9. `HEART_EMPTY_COLOUR = 0x000000` (colour of heart when life is lost)
10. `SCORE_ADDR = [716, 732, 748]` (offset from `ADDR_DSPL` of the top left corner for each digit of the score)
11. `WALL_COLOUR = 0x555555` (colour of the walls)
12. `TOP_WALL_THICKNESS = 36` (thickness of the top wall)
13. `SIDE_WALL_THICKNESS = 8` (thickness of the side walls)
14. `BRICK_COLOUR = 0x82d2c8` (colour for the breakable bricks)
15. `BRICK_ROWS = 28` (thickness of the row of bricks)
16. `BRICK_START_HEIGHT = 56` (starting height of the brick row)
17. `BRICK_WIDTH = 16` (width of an individual brick)
18. `BRICK_HEIGHT = 8` (height of an individual brick)
19. `PADDLE_COLOUR = 0xeeeeee` (colour code of the paddle)
20. `PADDLE_WIDTH = 36` (width of the paddle)
21. `PADDLE_HEIGHT = 4` (height o the paddle)
22. `BALL_COLOUR = 0xe54b4b` (colour code of the ball)
23. `BALL_SIZE = 4` (width and height of the ball)

The following are the variables defined in memory as well as their initial values:

1. `PADDLE_X = 112` (x coordinate of the top left corner of the paddle)

2. `PADDLE_Y = 252` (y coordinate of the top left coordinate of the paddle)
3. `BALL_X = 128` (x coordinate of the ball)
4. `BALL_Y = 228` (y coordinate of the ball)
5. `BALL_VX = -4` (x velocity of the ball)
6. `BALL_VY = -4` (y velocity of the ball)
7. `IS_PAUSED = 0` (boolean indicating whether the game is paused or not)
8. `IS_LAUNCHING = 1` (boolean indicating whether the user is currently positioning the ball)
9. `LIVES = 3` (number of lives)
10. `SCORE = 0` (current score measured by bricks broken)

Scene

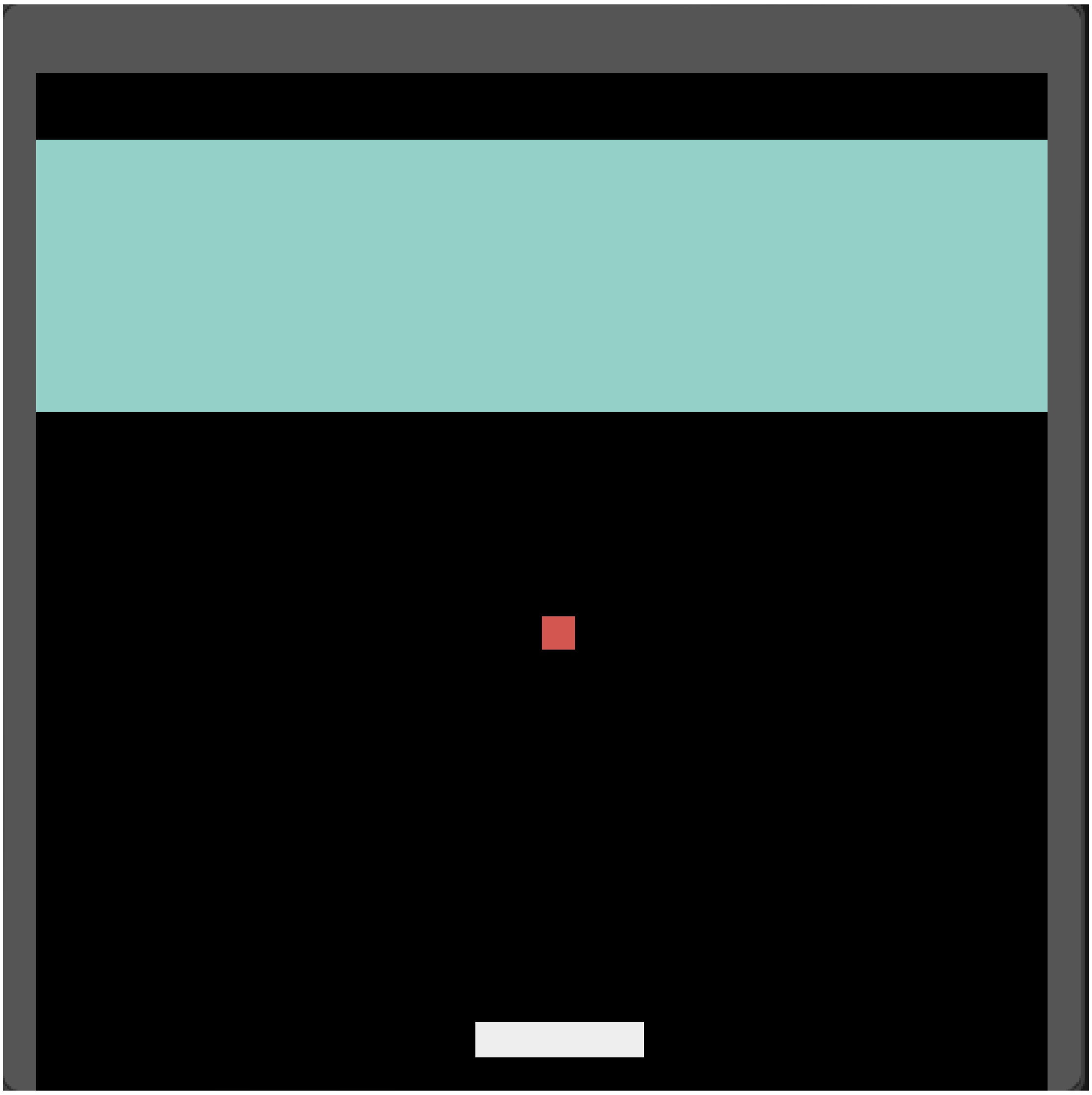

Figure 1: Screenshot of the static scene developed in Milestone 1

Milestone 3

How will the ball change directions when it collides?

Depending on the direction in which the collision occurs, the ball's parallel velocity component will be inverted. That is, if the pixel ($\texttt{BALL\_X} + \texttt{BALL\_VX}$, $\texttt{BALL\_Y}$) has a colour determined to be collidable, the ball reacts to a collision in the x-direction by reversing the x-velocity

$$\texttt{BALL\_VX} = -\texttt{BALL\_VX}$$

and if ($\texttt{BALL\_X}$, $\texttt{BALL\_Y} + \texttt{BALL\_VY}$) has a colour determined to be a collidable object, the ball reacts to this collision in the y-direction by reversing the y-velocity

$$\texttt{BALL\_VY} = -\texttt{BALL\_VY}$$

If the ball collides with a corner (whether it be the inward or outward corner), which occurs when both of the above cases are true, the ball will reverse direction completely:

$$\texttt{BALL\_VX} = -\texttt{BALL\_VX};\ \texttt{BALL\_VX} = -\texttt{BALL\_VX}$$

How to Play

Ensure that the bitmap display is configured to 256x256 with 4px unit size.

Controls:

- <a> + <d>: move paddle left and right
- <space>: launch ball when starting an attempt
- <p>: pause the game
- <q>: quit the game

Launch the ball with <space> and then destroy all the bricks to win! If the ball reaches the bottom of the screen, a life will be lost. After three lives lost, the game will end.