Assembly Project: Breakout

Fatimah Oyarekhua

April 27, 2023

1 Instruction and Summary

1. Which milestones were implemented? Milestones 1, 2, 3, 4 and 5 were implemented. H = hard, E = easy
2. Features implemented: Milestone 4
   (a) add multiple lives (E)
   (b) display game over (E)
   (c) hit brick multiple times (H)
3. Features implemented: Milestone 5
   (a) add sound effect (E)
   (b) add pause (E)
   (c) add unbreakable brick (E)
   (d) add paddle physics (ball hits left paddle it goes left, middle paddle stays middle and right paddle goes right) (H)
   (e) blue brick power up adds life (E)
   (f) allow player to launch ball at beginning (E)
4. How to view the game:
   (a) Unit width in pixels: 8
   (b) Unit height in pixels: 8
   (c) Display width in pixels: 256
   (d) Display height in pixels: 128
   (e) Base Address for Display: 0x10008000 ($gp)
5. How to play the game/ game controls:
   (a) launch ball: space bar
   (b) pause game: p
   (c) play game: l
   (d) quit game: q
6. Game Summary:
   - It uses the draw rectangle function to create 3 grey walls (one at the left of the screen, one on the right of the screen, and one at the roof), the paddle and the bricks.
   - The ball initially bounces off the paddle in the middle right direction specified by my direction_x and direction_y variables. If the ball hits something to the right the direction x will go the opposite way.

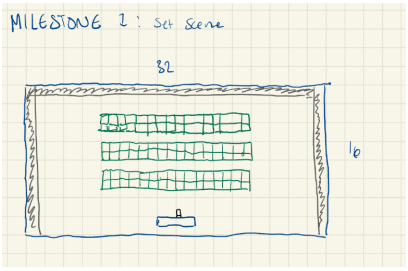

Figure 1: module 1: Planning the scene

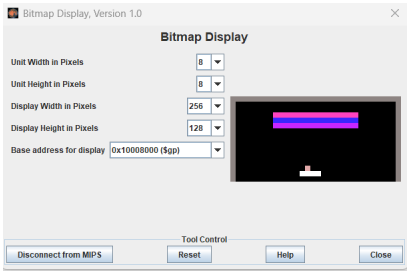

Figure 2: module 1: The scene

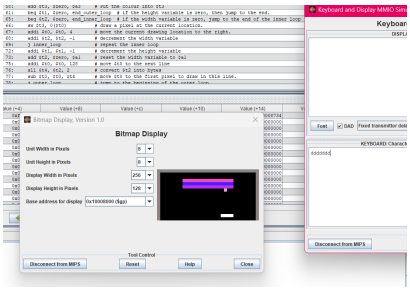

Figure 3: module 2: The keyboard

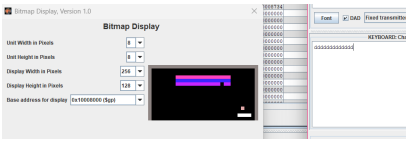

Figure 4: module 3-a: Collision example

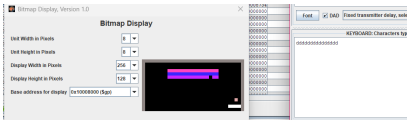

Figure 5: module 3-b: Collision example 2

Direction y works similarly and both are binary/boolean values. For the y-values, up and down are 1 and 0 respectively. For the x-values left, neither and right are 0, 1, and 2 respectively.

- I used the keyboard asm file provided to implement functionality with the keyboard
- The game detects whether a collision has occurred by checking the colour of its surroundings. If it is not the wall colour or background colour, it's a brick and it breaks it. The life of the brick depends on the colour of the brick. Once the brick is red, the brick will fully break. If it is the wall it simply bounces off of it

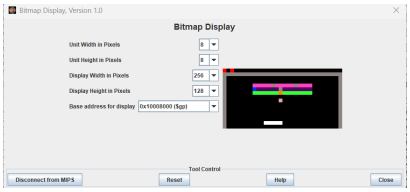

Figure 6: module 4-5

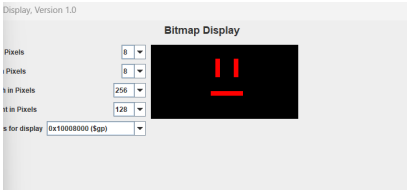

Figure 7: game over screen

- paddle physics were added by checking how much of the paddle is on the left side or the right side. The paddle is 5 pixels so when the ball hits the paddle (checked by colour), the colour of the pixel underneath and 2 pixels to the right of the ball is checked. If it's black then the ball is on the right side of the paddle. The same is done on the left side. If both pixels checked are white then the ball is centered on the paddle
- lost life was detected by checking if the location of the ball was greater than a particular pixel. If so, a life was lost, and the ball and paddle were reset. If all lives were done then game over screen was shown and the entire screen was reset.