csc258 proj

Nicholas Au, Future Hu

December 2022

1 Introduction

2. For the data part, We have 9 different kinds of colours storing in our memory, including red, green, blue, grey, black, white, yellow, light green and purple. These are stored under the label MY_COLOURS. 4 different kind of labels for locations, which are some mutable data of some object components in our game. This includes the x_axis for the paddle, the x and y axis or the ball location, the ball current direction etc. These data are stored under the label DYNAMIC_LOCATION. We also save up space to store up location address around the ball. And store up space to store the level.

3. https://www.overleaf.com/project/6385e7e50c77d60d154ae448

```
    .data
    .eqv RED 0xff0000
    .eqv GREEN 0x00ff00
    .eqv BLUE 0x0000ff
    .eqv GREY 0x808080
    .eqv WHITE 0xffffff
    .eqv YELLOW 0xffea00
    .eqv BLACK 0x000000
    .eqv TEST_COLOUR 0xf5bc42
    .eqv LIGHTGREEN 0x90EE90
    .eqv PURPLE 0x8A2BE2

    .eqv FRAME_DELAY 3000
    .eqv SLEEP_DELAY 15
##############################################################################
# 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
```

Figure 1: RGB Colour Data Stored in Memory.

```
##############################################################################
# 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

MY_COLOURS:
        .word   RED
        .word   GREEN
        .word   BLUE
        .word   GREY
        .word   WHITE
        .word   YELLOW
        .word   BLACK
        .word   LIGHTGREEN
        .word   PURPLE

# TODO: can change back the initalization data here.
```

Figure 2: RGB Colour Data Stored in Memory.

```
##############################################################################
DYNAMIC_LOCATION:
    .word   13  # x_paddle address
    .word   16  # x_ball location
    .word   28  # y_ball location
    .word   1   # ball_direction
# The surrounding tiles of the ball
TILE_UP:
        .space 4
TILE_DOWN:
        .space 4
TILE_LEFT:
        .space 4
TILE_RIGHT:
        .space 4
TILE_UPPER_LEFT:
        .space 4
TILE_UPPER_RIGHT:
        .space 4
TILE_LOWER_LEFT:
        .space 4
TILE_LOWER_RIGHT:
        .space 4
LEVEL:
        .space 4
```

Figure 3: Dynamic Location Stored in Memory

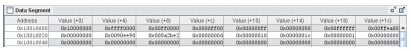

Figure 4: Data Segments with values in hexadecimal.

4. For Milestone 1, most of the components can be represented by using line functions. We hence chose to create a drawing horizontal line function and a drawing vertical line function. We also used a get_location_address function, which it can help us to transform a x and y coordinate with each having 32 units, into a specific address inside the bitmap display. By using these functions, we can draw lines easily by only filling in a starting address, the width/height of the line both in 32 units form, and the colour of it (stored in memory) to create it. This design suits the data in our memory as we can then store coordinates and translate to address easily.

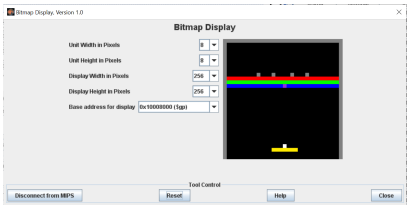

Figure 5: Initial position of the game: Level 1

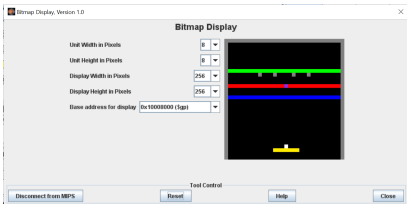

Figure 6: Initial position of the game: Level 2

6. For colliding, we designed it just as physics in real life. When the ball hits an object, the ball will bounce off at the same angle but at an opposite direction. In more detailed description, when the ball is going in a straight line towards the up, down, left, or right direction. It will only collide when there is an object in front of it. If the ball is going towards a diagonal direction, then it may collide with any objects in front of it along the diagonal path. The ball then will bounce off at the same angle but at an opposite direction.

2 Features

We have accomplished 3 easy features and 3 hard features. Including easy feature number 3, 5 and 7. Hard feature number 3, 4, 7. For easy feature number 3, for each collision the ball will increase a bit. For easy feature number 5, the user can press "p" to pause and press "p" to resume the game. For easy feature number 7, we added 4 unbreakable bricks in the map. For hard feature number 3, green color needed to hit twice to break it it will turn into light green after one hit, then disappears after two hit. For hard feature number 4, we have 2 different levels that allow the user to choose before start. For hard level number 7, we have a purple brick, if it breaks, the blocks the left 3 bricks and the right 3 bricks of it will also break.

3 How To Play

On top of the screen are some break lines, and the objective of the game is to break all of them by using the paddle to repeatedly bounce the ball onto them. If the ball gets pass the paddle, the game ends immediately and the player losses if the player hasn't break all the breaks by that time.

To start the game, the player should press either "1" or "2" to choose between level 1 or level 2 layout.

The player than can control the paddle can be controlled by the player by using the "a" and "d" button, "a" to move paddle to the left and "d" to move paddle to the right. The player can also use the "p" button to pause anytime during the game. Pressing the "p" button again will resume the game. Use the "q" button to quit the game at any point of the game.

There are also some things to keep aware of. Anything in grey are unbreakable. You may see some unbreakable bricks in the middle of the map. Purple bricks are "power-up" blocks, if the purple block is hit, it destroys 3 bricks on it's left and 3 bricks on it's right. Green bricks have two layers, which you need to hit two times on it to break it entirely. The green break will turn into light green brick to indicate that it has been hit once. The ball with get quicker as the ball collides more and more. Lastly, the ball will bounce off straight up if it hits the middle of the paddle. The ball will bounce off diagonally if it hits the tail of the paddle.