

The Design Document for HuarongDao

Haobo Liu (hl3645), Rui Chu (rc3414), Jiusheng Zhang (jz3444), Jingwei Zhang (jz3555), Nina Hsu (hh2961)

CSEE 4840 Embedded System Spring 2023

Contents

1	Introduction	2
2	System Block Diagram	3
3	Project Algorithm	4
4	Resources Budget	5
5	Hardware/Software Interface	6

Introduction

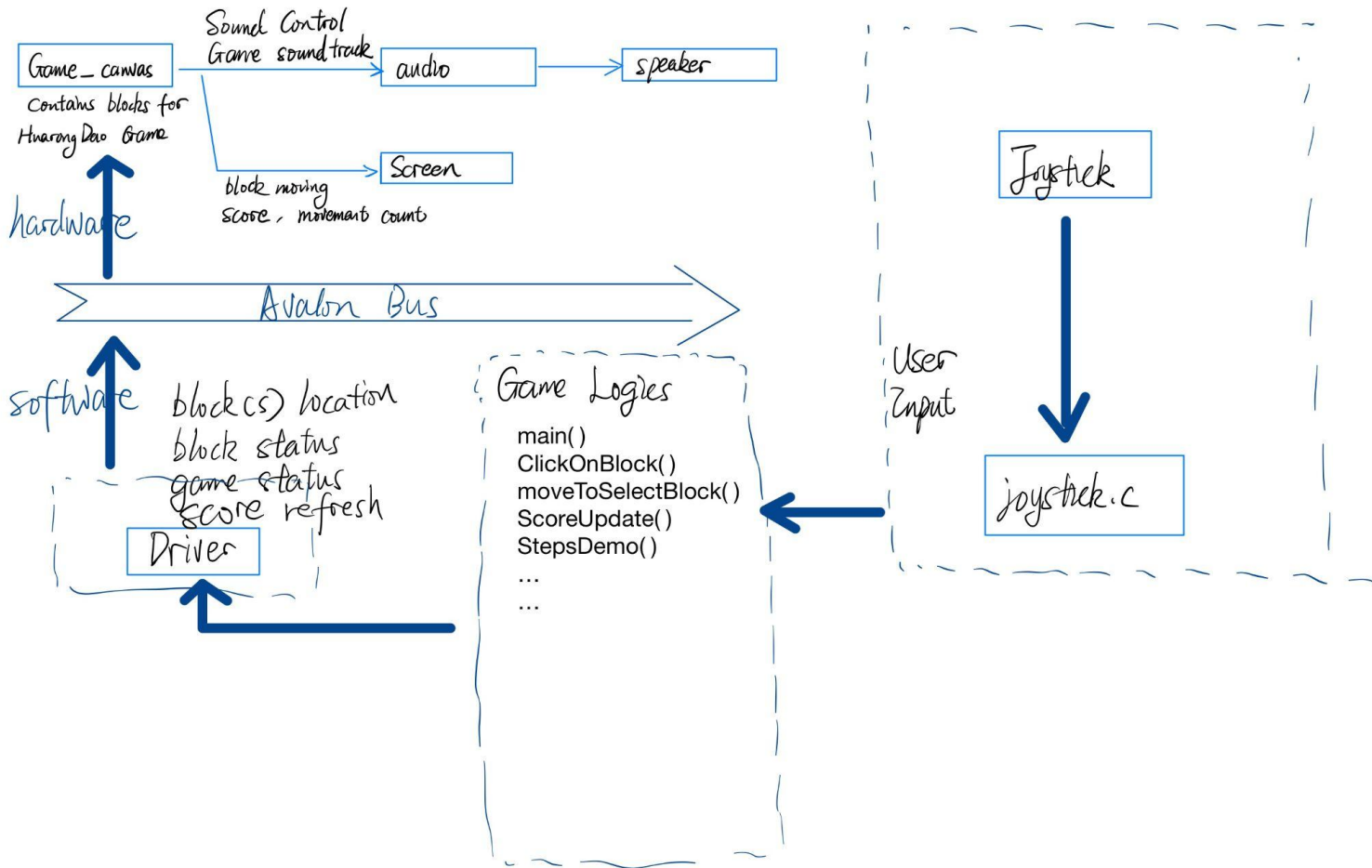
Our plan is to implement a 2D game that takes a mouse and a keyboard as input, and outputs a VGA display. We choose *Huarong Dao*, a Chinese variation of *Klotski*, as a potential game logic. We might switch to other game logics, but we will stick to the overall logic:

Mouse/Keyboard input → *2D Game Logic* → *VGA Display*



Klotski Layout and Huarong Dao

System Block Diagram



Project Algorithm

- game logic
 1. Board representation: The game board is represented as a 3x3 grid of squares, with each square either containing a tile or being empty.
 2. Tile movement: When a tile is selected, it can be moved horizontally or vertically to an adjacent empty square. The movement of the tiles is constrained by their size and shape, and the position of the other tiles on the board.
 3. Win condition: The player wins the game when the largest tile, representing the general or king, is moved to the center of the bottom row and out of the board through the exit.
 4. Game state management: The game needs to keep track of the current board state, including the position of each tile, and check for valid moves and the win condition.
- algorithm used to solve the puzzle
Brute Force Search or A*

Resources Budget

Graphics Memory Budgets

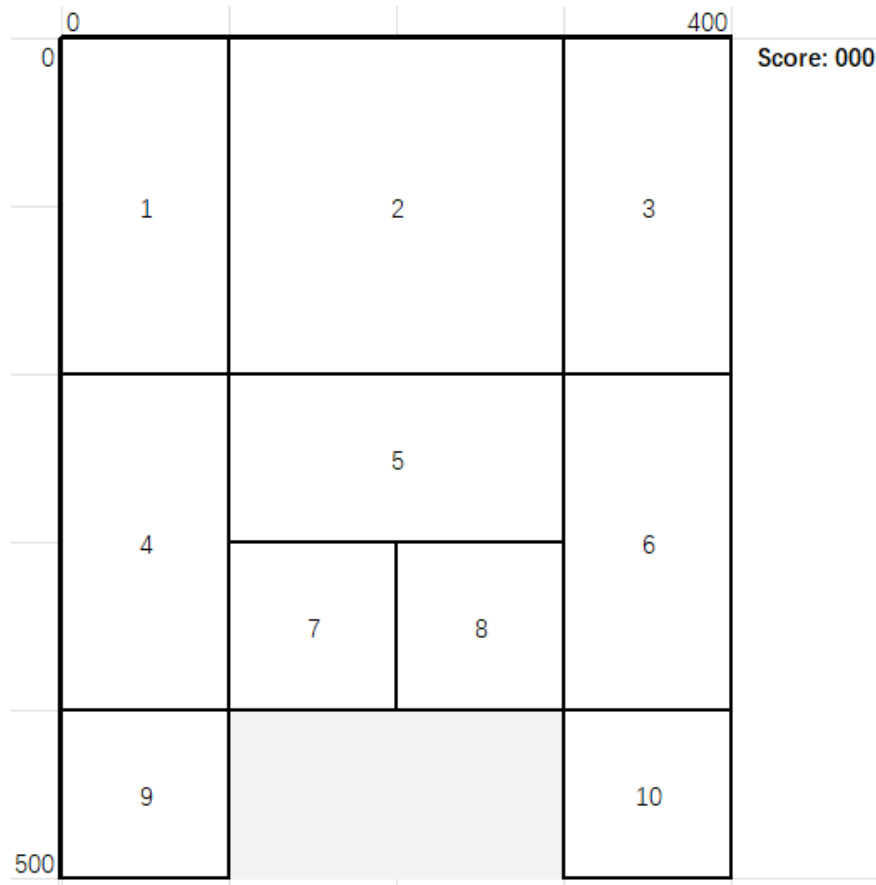
Category	Size (bytes)	#	Total size (bytes)
100x100 block	100x100	4	40,000
100x200 block	100x200	4	80,000
200x100 block	200x100	1	20,000
200x200 block	200x200	1	40,000
Score	100x10	1	1000
Memory budgets (bits) 1,448,000			

Audio Memory Budgets

BGM	Movement	Win
Try to limit to ~1,000,000 bits in total		

Hardware/Software Interface

VGA output layout



```
module h_vga(  
    input logic clk,  
    input logic reset,  
    input logic write,  
    input logic [15:0] writedata,  
    input  chipselect,  
    input logic [3:0] address,  
    output logic [7:0] VGA_R, VGA_G, VGA_B,  
    output logic  VGA_CLK, VGA_HS, VGA_VS,  
                VGA_BLANK_n,  
);
```

output logic VGA_SYNC_n);

writedata: 2-byte data received from the software. The software will send the x_location and y_location of each block, the location of cursor, the location of selector, location of selected block and the score data.

address: the location where the writedata should be written to. There are 14 addresses.