



Peter Mansour (phm2122)  
 Shabhari Saravanan(ss5913)  
 Pratyush Agrawal(pa2562)  
 Aaron Pickard(ajp2235)

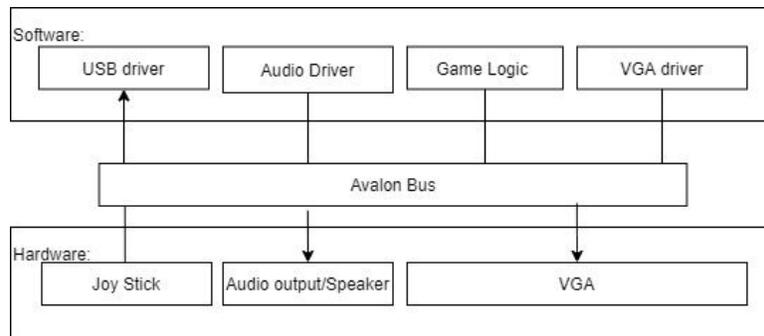


Fig 1. Software & Hardware Interface

Display:

Background:

We will have 3 background layers

Layer 1\* (back): black background with stars

Layer 2\*: image of trees that is 2x or 3x the width of the screen

Layer 3 (front): obstacle sprites generated by converting .jpg or .png using matlab to .mif and then to .v files using qsys.

\*Using python's PIL library, we will generate 3D numpy matrices for layers 1 and 2 where length and width correspond to x and y dimensions of the screen and the 3rd dimension corresponds to the R, G, and B layers of the pixel.

#### Audio:

convert the audio files to .v files, then use CODEC WM8731 to output the sounds to the speaker after configuring it using I2C multiplexer. We will trigger the audio signals by writing to a register in /dev/vga\_ball using ioctl

#### Sounds:

Shooting  
Explosion

#### Sprites:

Player	
Obstacles	
Spaceship	
Stars	

Bullet (If there is sufficient memory, we can set pixel colors instead)

\*\*\* Instead of explosion sprite, we will make player and spaceship blink when destroyed

#### Registers in vga\_ball.v:

3 registers for each pixel in the background storing R, G, and B values obtained from the software side.  
2 location registers for the player  
2 location registers per obstacle  
2 location registers per spaceship  
2 location registers for bullet  
Register for triggering each audio

#### Communication with JoyStick:

We will be using the usb driver used for connecting the keyboard or SDL JoyStick library.

#### Game Logic:

##### Thread 1 (background layer 2):

Shift starting column left to right by 1 every couple of seconds to create background moving effect  
Iterate through the 3d matrix storing background pixels' colors from start column to column before it and write to corresponding registers on hardware side

##### Thread 2 (obstacles & spaceships):

Always start at the right-hand side of the screen and move to the left at a constant speed

The time between spawning obstacles should be random

Main Thread (Player):

Spawns 3 static spaceships at the beginning of the game

Once the player presses “Up Arrow” Key, the player’s position is shifted up and the location on screen is updated

Check if the player hit an obstacle; trigger explosion sound and sprite accordingly by writing to appropriate register using ioctl

GameOver if all 3 spaceships are destroyed

Thread 4 (Bullet):

Limit number of bullets fired at the same time to 1

The position of the bullet is updated every couple of seconds

Check if bullet exited the screen or hit a spaceship

Trigger sound effects and the blink spaceship sprite if bullet collided with it

Questions:

How do we manage/order multiple layers on the hardware side to get the depth effect?