VGA output:

* struggled to find driver that was suitable for esp32-s3 (most were only for esp32), found suitable one made by bitluni.
* Used a resistor ladder with 200R and 100R resistors for each colour output pin (r,g,b) which allows for 4 colour levels for each colour (0, low, medium, high) which is not much but allows for 4\*4\*4=64 different colours
* Initially tried running with resolution MODE\_800x600x60 (hRes, vRes, frequency) but this lead to screen flickering between each refresh, so I experimented with different timing settings but eventually moved down to MODE\_640x480x60
* The buffer count was also reduced from 2 to 1 for same reason as above
* Kept getting a “watchdog timer” error and the program kept rebooting, also the screen alignment would be off and change everytime the screen was redrawn. At first I thought this was due to infinite while loops, so I inserted “yield()” into loops which calls the esp32 to do any background maintenance that it would normally do with the normal setup() and loop() structure. This did decrease the alignment issues but it was still wrong and would shift lots when complex things were drawn.
* Next, I thought it was due to the large font size taking up memory on the board so I used a simple font which again reduced the screen shifting but not perfectly
* Finally, I reduced the resolution further to MODE\_320x240x60 which fixed all alignment issues. Then I could select a larger font and drawing complex graphics was no issue. Although in cases where the whole screen is redrawn there is some flicker between transitions (Still working on this issue).
* Maybe write something about during our first makerslab session, having to connect the monitors VGA port to board directly using little cables.

Memory storage:

* Due to not needing too much memory (only to store highscores), and limited available pins I selected to use a EEPROM chip that can be controlled with just 2 pins and has 256Kb storage which is plenty for our use. These chips have limited write cycles before wearing out, but this isn’t too much of an issue since highscores may be regulary read but aren’t regularly written (i.e. beaten).

Audio:

* Opted to use a mini 8ohm speaker that can be controlled using just 2 pins
* Considered an amplifier for louder music but the speaker on its own had ample volume for our use, also the amplifier is relatively expensive and would require addition power supply so is not necessary in our design.

Consol casing:

* Opted for a retro arcade machine look to go with the matching retro controller
* Browsed old arcade machines before tracing a rough shape and drawing it with CAD to 3d print
* It was difficult to Fit parts exactly, so multiple sections had to be printed separately as prototypes to confirm dimensions. E.g. The holes for nuts to sit in (to join panel), bed for circuit board, and holes for potentiometer and vga port had to be printed multipe times to ensure the right size.
* Had to get a secure way to store ttgo in console whilst having screen visible, chose have a rectangular beam that precisely fit the ttgo dimensions, then when closing case this clamps ttgo to prevent moving. (I’m still working on final case design so will update this later)

Console circuitry:

* Chose to solder all console circuitry to strip board as it is secure, and easy to customize if the design changes. Soldering took far longer than expected (about 4 hours) dues to tip of soldering iron not heating up so had to use side, this made it difficult to be precise.
* For wires, used female-female cables and cut them in half and stripped. Then the stripped end could be soldered, and the female port could connect to the TTGO or VGA pins as necessary. (Most likely will secure these further for the finished product by stripping end and soldering, or by gluing each port to pin to prevent them popping out).

Console code (Menus and gameplay):

* Although I was still facing issues with VGA output as mentioned above, wanted to at least have some game with graphics implemented. No online libraries would work so had to write the code myself to avoid the VGA issues I was facing.
* This code (PONG and the moving block thing) worked fine on their own, but still wouldn’t work properly when called through the menu due to the issues above
* When the vga output was all fixed this code had to be tweaked but then worked fine