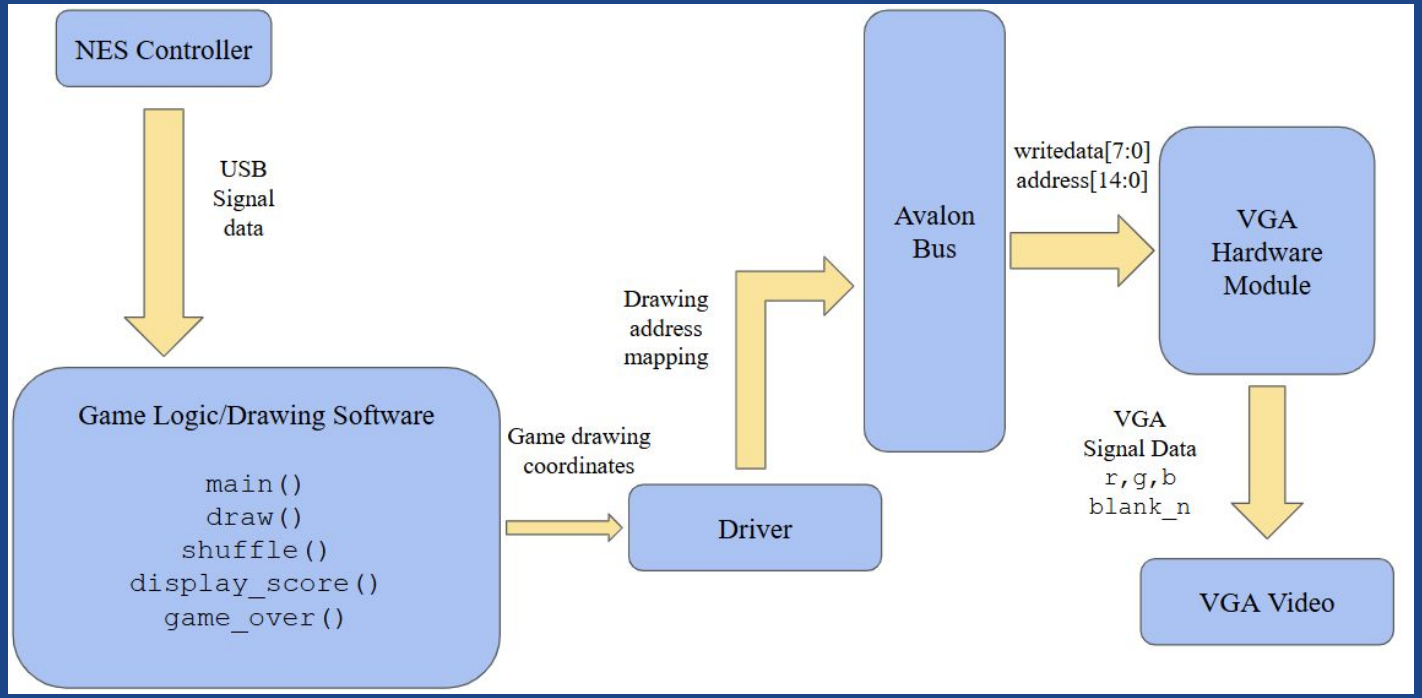
Arcade Poker Deck Builder Game: Balatro Minus

Mahdi Ali-Raihan, Timothy Melendez, Mario Carrillo, Julio Ramirez

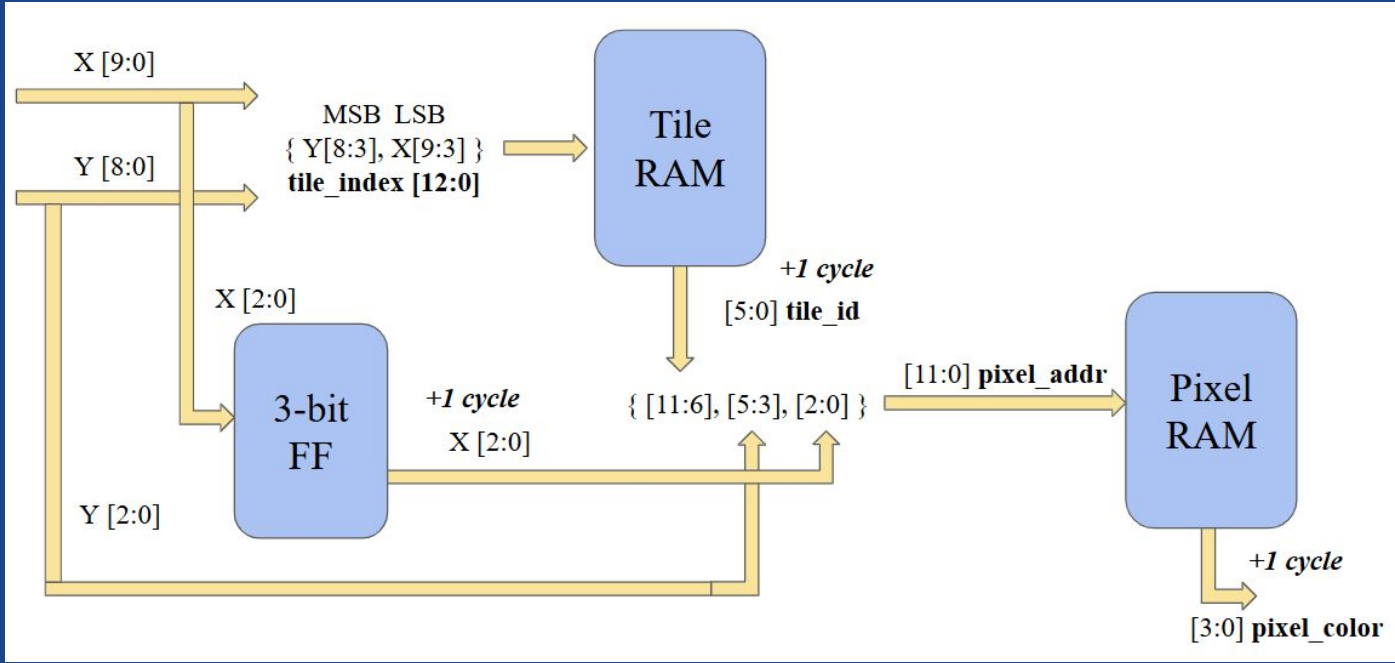
Game Overview

- Build poker hands using cards in your hand
- Reach a threshold score to beat the round
- Three successive rounds conclude an Ante, up to eight Antes
- You win if you beat the last Ante
- You lose if you fail to reach the threshold score in a round with the number of hands provided

System Overview

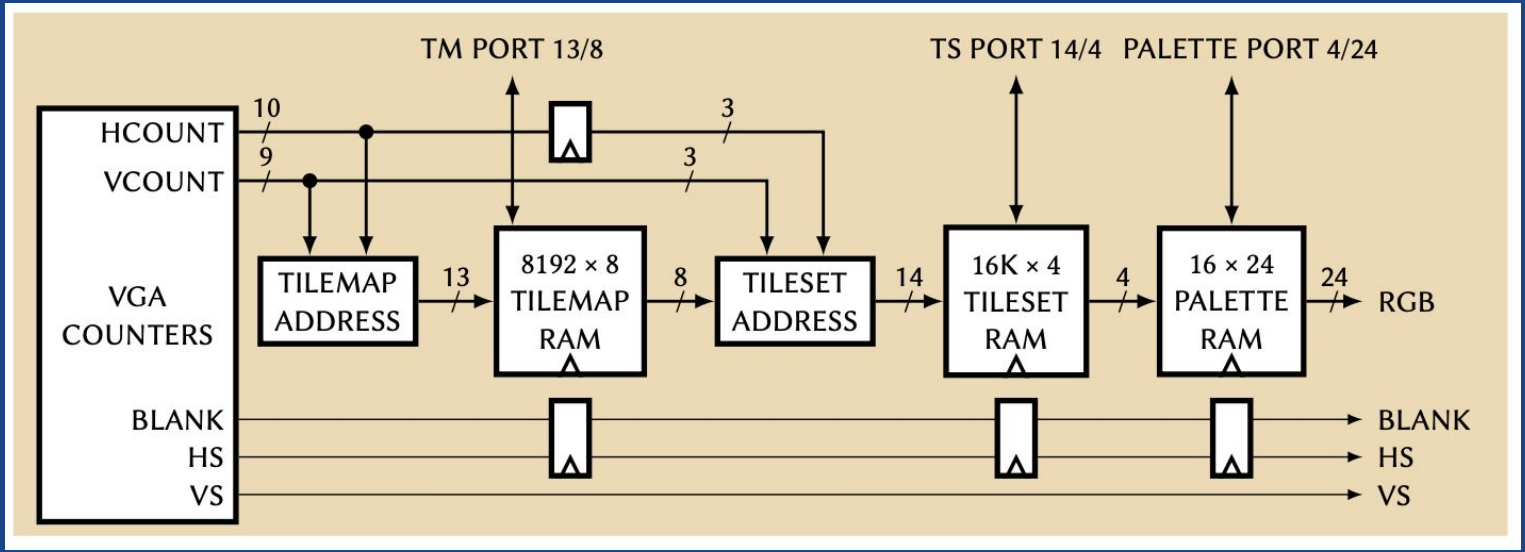


VGA Control: Original Design

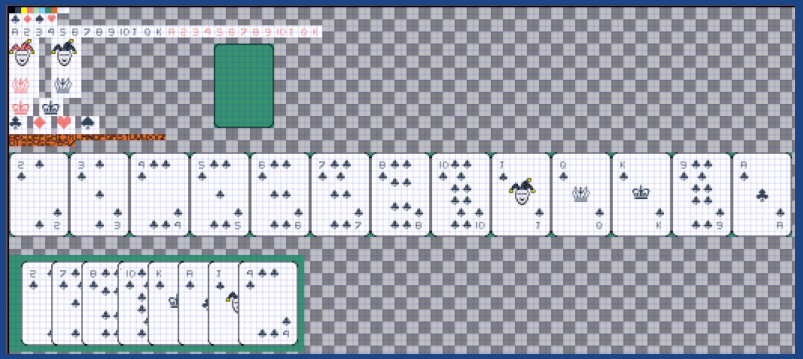


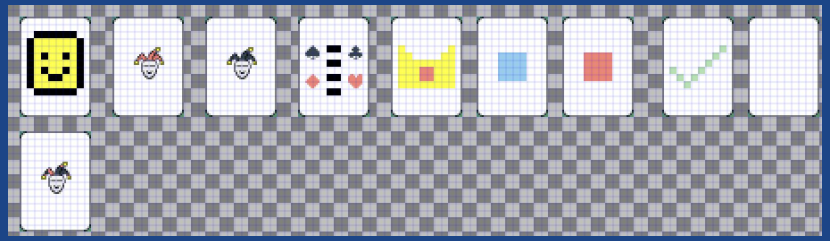
VGA Control: Actuality

Courtesy of Stephen A. Edwards



Tile Graphics





- Used 255 distinct tiles in our Poker game, which are all created in Asesprite

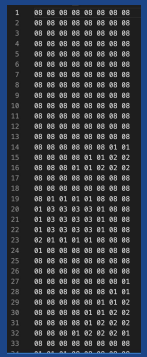
Convert Tileset png into hex file

↓ Tileset in indexed png form (tiles in one row)

```
00000000
000000FF
36454FFF
FFFF00FF
FA8072FF
AFE1AFFF
89CFF0FF
F0F0F0FF
FFFFFFFF
37946EFF
DF7126FF
00000000
00000000
00000000
00000000
00000000
```

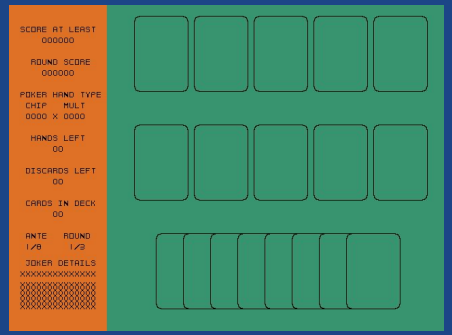
Palette.hex
(12 colors + padding)

→

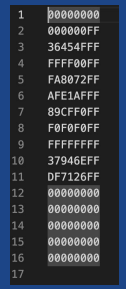


tileset.hex

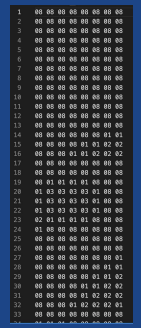
Converting from .png to .hex



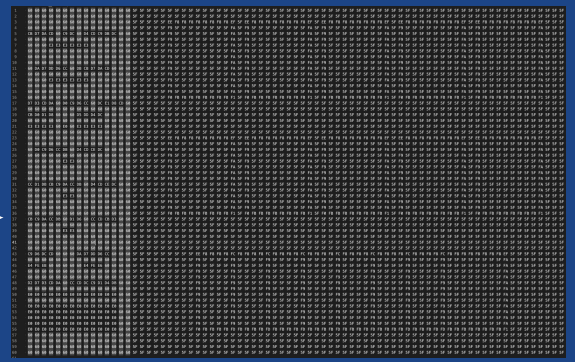
Game UI in indexed PNG form



palette.hex

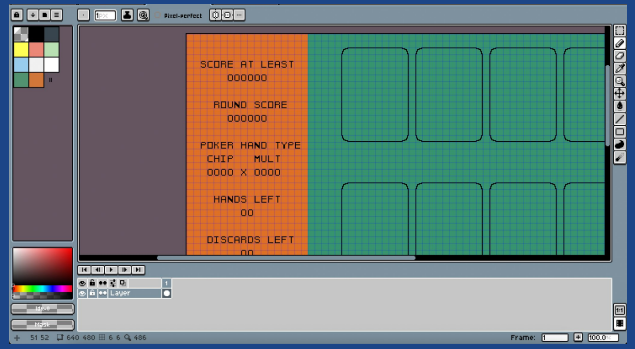


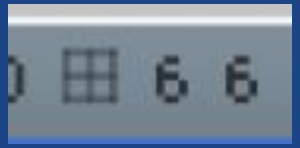
tileset.hex



Game UI indexed png is now in .hex format

Retrieving Tile Row/Col to Modify from SW → HW





VGA HW-SW Interface

```
/* Device registers */
#define TILEMAP_BASE    0x0000  /* 8 KiB: 0x0000-0x1FFF */
#define PALETTE_BASE    0x2000  /*  64 B:  0x2000-0x203F */
#define TILESET_BASE    0x4000  /* 16 KiB: 0x4000-0x7FFF */
```

```
static void set_tile(vga_poker_tile_coords_t *tile_coords)
{
    int row;
    int col;
    int tile_id;
    unsigned short new_tile_addr;
    row = tile_coords->row;
    col = tile_coords->col;
    tile_id = tile_coords->tile_id;
    new_tile_addr =  TILEMAP_BASE + (row << 7 | col);
    pr_info("addr: %u\n", new_tile_addr);
    iowrite8(tile_id, dev.virtbase + new_tile_addr);

}
```

```
static void set_pixels(vga_poker_pixels_t *pixel_coors)
{
    int tile_id;
    int i;
    tile_id = pixel_coors->tile_id;
    for (i = 0; i < 64; i++) {
        int color;
        int new_addr;
        color = pixel_coors->pixels[i];
        new_addr = TILESET_BASE + (tile_id << 6) + i;
        pr_info("addr: %u\n", new_addr);
        iowrite8(color, dev.virtbase + new_addr);
    }
}
```

```
static void set_palette(vga_poker_palette_t *palette)
{
    int base;

    if (palette->index >= 16) return;

    base = PALETTE_BASE + (palette->index << 2);
    pr_info("addr: %u\n", base);
    /* write into creg[7:0], [15:8], [23:16] */
    iowrite8(palette->red,   dev.virtbase + base + 0);
    iowrite8(palette->green, dev.virtbase + base + 1);
    iowrite8(palette->blue,  dev.virtbase + base + 2);
    /* commit into palette[pl->index] */
    iowrite8(0, dev.virtbase + base + 3);
```

NES Controller


```
/* Controller Input Functions */
void initialize_debounce();
int is_debounce_elapsed();
int
check_controller_input(unsigne
d char *report);
void toggle_card_selection();
void move_cursor(int direction);
void
process_controller_input(unsig
ned char *report);
```

Left: moves cursor left

Right: moves cursor right

A: select card

B: deselect card

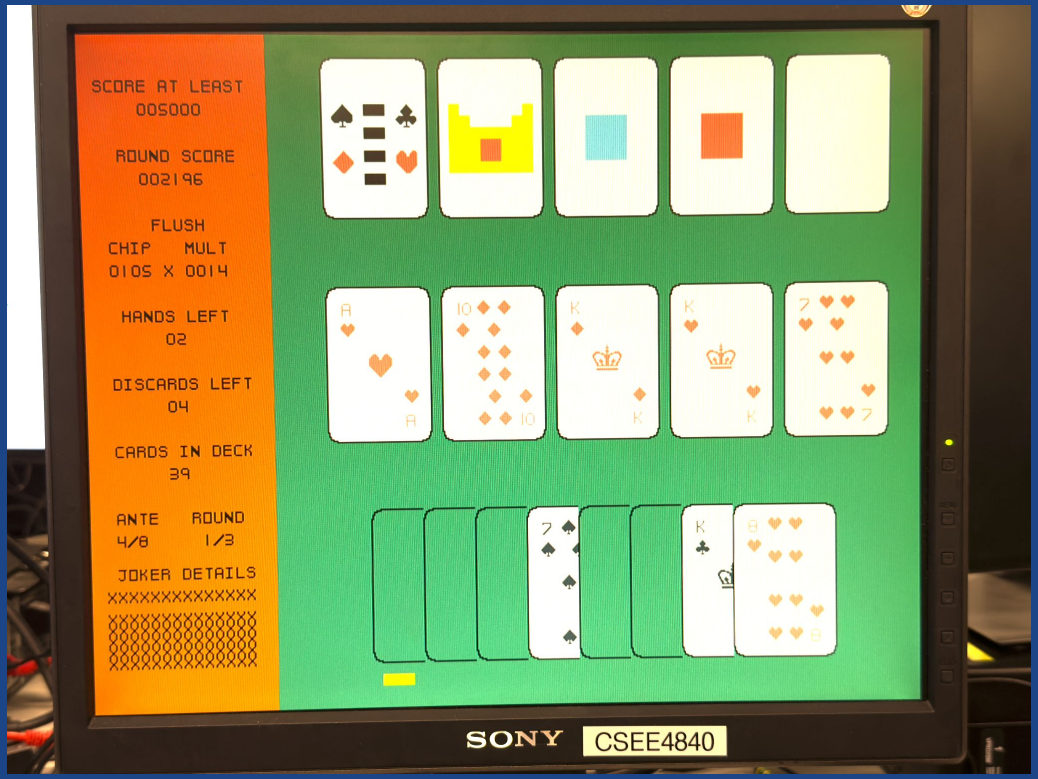
Select: discard selected cards

Start: play selected cards

Game Logic

- Init Deck, Shuffle, Hand Generation
- Select/Discard Cards
- Chip and Mult Values Based on Poker Hand Type (Pair, Two Pair, Flush, Straight, etc...)
- Current Score, Target Score, Discards, Hands,
- Game States (Win, Lose, Next Level, etc...)
- Joker Probability Drops

Image of Game



Demonstration