CSEE 4840 Design Document: Arcade Poker Deck Builder Game

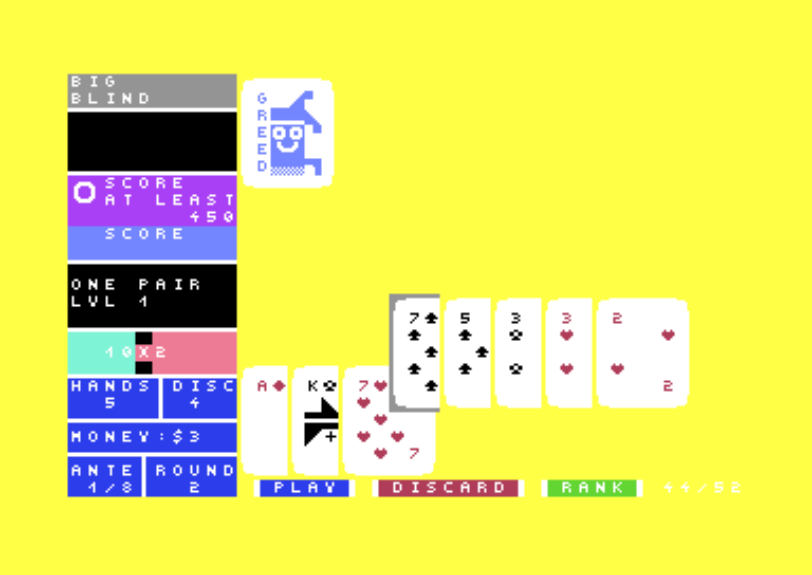
Julio Ramirez (jar2358), Mahdi Ali-Raihan (mma2268),
Timothy Melendez (tjm2196), Mario Carrillo (mc5132)

Spring 2025

Contents:

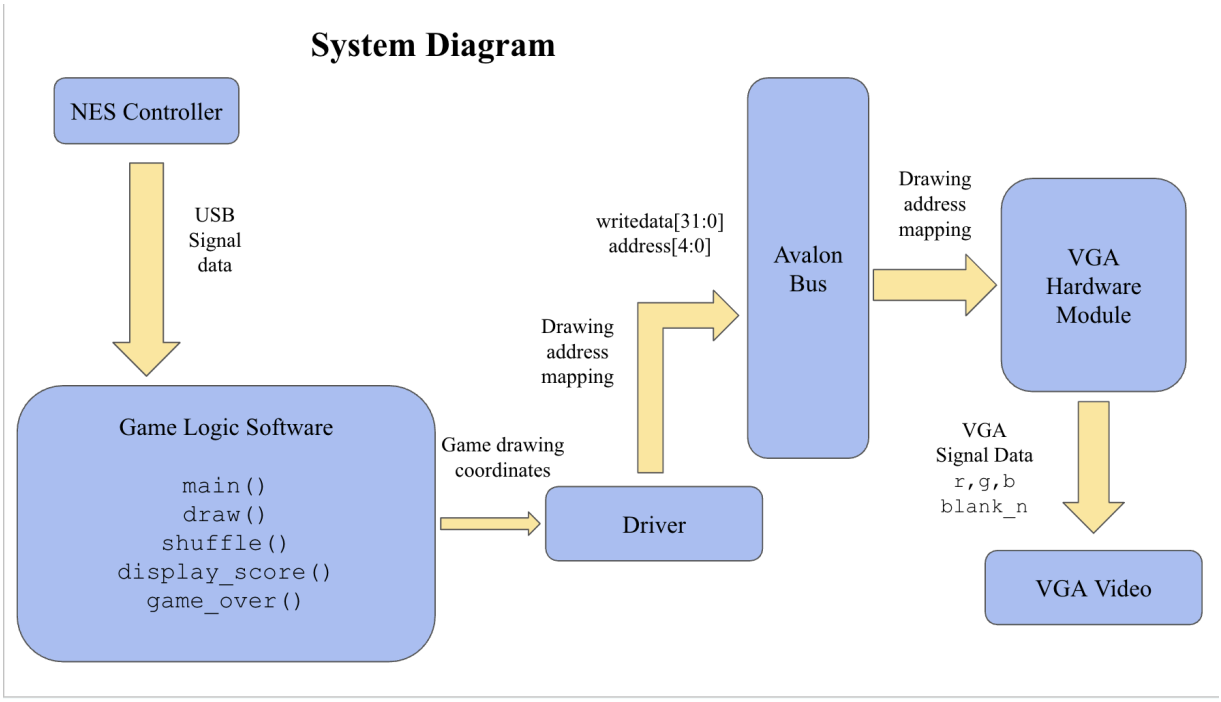
Introduction

This project aims to create an arcade version of poker, inspired by the roguelike card game *Balatro*. In this game, a hand of eight cards is dealt, from which the player may choose to play a poker hand or discard and draw. The task is to reach the required target score for each blind. As the user progresses through the game, the target score increases. To meet the target score, the user must form strong poker hands to earn more points. After beating the target score, the user will win a random ‘Joker’ card, which will help the user defeat much more difficult target scores down the road. The game will be displayed on a 640 x 480 VGA monitor and controlled via an NES controller.

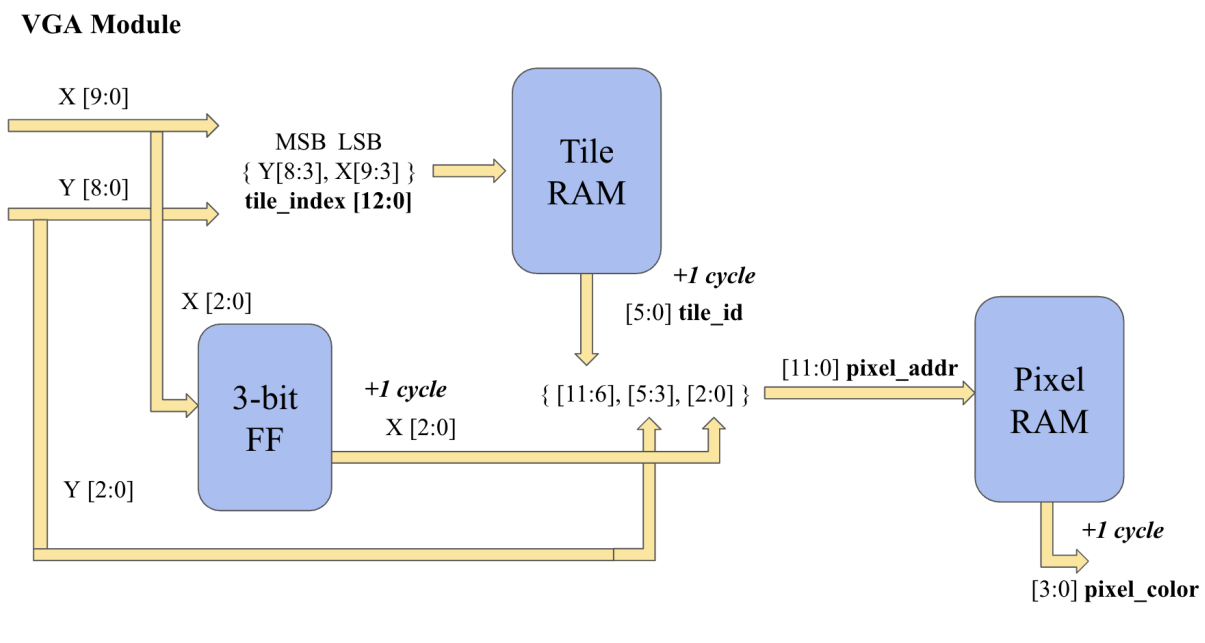


Pixelated version of Balatro for inspiration

System Block Diagram



VGA Video Rendering Peripheral:

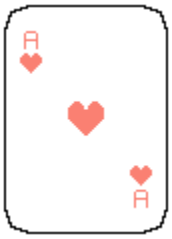


Algorithms

The following algorithms outline the functionality that will be implemented in the game.

- **shuffle()** - Shuffles the deck of cards before a play.
- **draw()** - Draws cards until the player has eight cards. This is called at the beginning of play, after discarding cards, and after playing cards, with a maximum of 52 cards.
- **discard()** - Discards the selected cards if enough discards remain.
- **play()** - Plays the player's selected poker hand.
- **dpad_input()**:
    - dpad left button → Moves the cursor to the left.
    - dpad right button → Moves the cursor to the right.
    - dpad ‘a’ button → Button that selects or deselects card (toggle effect)
    - dpad ‘b’ button → Button that discards selected card(s) - calls discard().
    - dpad ‘start’ → Button that plays selected card(s) - calls play().
- **evaluate_selected_cards()** - Evaluates the player's poker hand - identifies the name.
- **game_over()** - When a player runs out of playable hands, a game over occurs, and the player restarts from the beginning.
- **draw_joker()** - When the player beats a blind, the player has a 75% probability of getting a random joker.
- **display_score()** - Display the score the player has earned.
- **display_target_score()** - Displays the required score the player has to earn to continue.
- **evaluate_joker_cards()** - Checks to see if the player’s played poker hand matches the joker card(s) modifier requirements to receive a multiplier increase, chip count increase, or played hand increase level.
- **display_hands()** - Displays the number of available hands the player has left for the designated blind.
- **display_discards()** - Displays the number of available discards the player has left for the designated blind.
- **display_round()** - Displays the round the player is in.
- **display_hand_chip_and_mult()** - Displays the hands' chip and multiplier values.
- **evaluate_round()** - Evaluates the ongoing round to see if the game is in a valid state.

Resource Budgets

Category	Sprite	Size (Bits)	# of Sprites	Total Size (Bits)
Playing Card (Ace, 2-10, Jack, Queen, King)		88 x 128 x 3	13	88 * 128 * 3 * 13 = 439,296
Heart (small)		16 x 16 x 3	1	16 * 16 * 3 * 1 = 768
Spade (small)		16 x 16 x 3	1	16 * 16 * 3 * 1 = 768
Diamond (small)		16 x 16 x 3	1	16 * 16 * 3 * 1 = 768
Club (small)		16 x 16 x 3	1	16 * 16 * 3 * 1 = 768
Jester Face		40 x 40 x 3	1	40 * 40 * 3 = 4,800
King Face		24 x 24 x 3	1	24 * 24 * 3 = 1,728
Queen Face		24 x 24 x 3	1	24 * 24 * 3 = 1,728
Heart (Big)		24 x 24 x 3	1	24 * 24 * 3 = 1,728
Spade (Big)		24 x 24 x 3	1	24 * 24 * 3 = 1,728
Diamond (Big)		24 x 24 x 3	1	24 * 24 * 3 = 1,728
Club (Big)		24 x 24 x 3	1	24 * 24 * 3 = 1,728

Joker Card		88 x 128 x 4	10	88 * 128 * 4 * 10 = 450,560
Letters	A	8 x 8 x 2	26	8 * 8 * 2 * 26 = 3,328
Numbers (0-9)	0	8 x 8 x 2	10	8 * 8 * 2 * 10 = 1,280
			TOTAL	912,704

The Hardware/Software Interface

Controller

Our game will utilize the Nintendo NES controller to access in-game controls, which will interface with the DE1-SoC board via a USB connection.

VGA Monitor

For the VGA interface, we plan to use a 32-bit-wide address bus, with up to 6 bits of addressing.

Address Mapping

Address	32-Bit Wide Bus	Information
0	Bit 0: Game Running Bit 1: Game Over State Bit 2: Blind Completed Bit 3: Hand Being Played Bit 4: Cards Being Discarded Bit 5-31: Reserved for future use	Game State Control Register
1	[31:0]: Player Current Score	Player Current Score
2	[31:0]: Target Score for Current Blind	Target Score for Current Blind
3	[7:0]: Current Level/Blind Number	Current Level

4	[31:16]: Remaining Discards Counter, [15:0]: Remaining Hands Counter	Resource Counter Register
5	[31:0]: Random Number Generator Seed	RNG Register
6	[31:16]: Card 1 X Position, [15:0]: Card 1 Y Position	Card 1 Position Register
7	[31:16]: Card 2 X Position, [15:0]: Card 2 Y Position	Card 2 Position Register
8	[31:16]: Card 3 X Position, [15:0]: Card 3 Y Position	Card 3 Position Register
9	[31:16]: Card 4 X Position, [15:0]: Card 4 Y Position	Card 4 Position Register
10	[31:16]: Card 5 X Position, [15:0]: Card 5 Y Position	Card 5 Position Register
11	[31:16]: Card 6 X Position, [15:0]: Card 6 Y Position	Card 6 Position Register
12	[31:16]: Card 7 X Position, [15:0]: Card 7 Y Position	Card 7 Position Register
13	[31:16]: Card 8 X Position, [15:0]: Card 8 Y Position	Card 8 Position Register
14	[31:16]: Card 9 X Position, [15:0]: Card 9 Y Position	Card 9 Position Register
15	[31:16]: Card 10 X Position, [15:0]: Card 10 Y Position	Card 10 Position Register
16	[31:16]: Cursor X Position, [15:0]: Cursor Y Position	Cursor Position Register
17	[9:0]: Selected Cards Bitmap (1 = Selected)	Card Selection Register
18	[31:16]: Current Hand Chip Value, [15:0]: Current Hand Multiplier	Hand Value Register

19	Bit 0: D-Pad Left Bit 1: D-Pad Right Bit 2: 'A' Button Bit 3: 'B' Button Bit 4: 'Start' Button Bit 5-7: Reserved	Input Register
20	[2:0]: Action Request (0=None, 1=Play, 2=Discard, 3=Draw)	Action Request Register
21	[7:4]: Card 1 Suit, [3:0]: Card 1 Rank	Card 1 Value Register
22	[7:4]: Card 2 Suit, [3:0]: Card 2 Rank	Card 2 Value Register
23	[7:4]: Card 3 Suit, [3:0]: Card 3 Rank	Card 3 Value Register
24	[7:4]: Card 4 Suit, [3:0]: Card 4 Rank	Card 4 Value Register
25	[7:4]: Card 5 Suit, [3:0]: Card 5 Rank	Card 5 Value Register
26	[7:4]: Card 6 Suit, [3:0]: Card 6 Rank	Card 6 Value Register
27	[7:4]: Card 7 Suit, [3:0]: Card 7 Rank	Card 7 Value Register
28	[7:4]: Card 8 Suit, [3:0]: Card 8 Rank	Card 8 Value Register
29	[7:4]: Card 9 Suit, [3:0]: Card 9 Rank	Card 9 Value Register
30	[7:4]: Card 10 Suit, [3:0]: Card 10 Rank	Card 10 Value Register
31	[3:0]: Current Hand Type Evaluation Result (0=High Card, 1=Pair, 2=Two Pair, etc.)	Hand Evaluation Register
32	[31:16]: Joker Card 1 X Position,	Joker Card 1 Position Register

	[15:0]: Joker Card 1 Y Position	
33	[31:16]: Joker Card 2 X Position, [15:0]: Joker Card 2 Y Position	Joker Card 2 Position Register
34	[31:16]: Joker Card 3 X Position, [15:0]: Joker Card 3 Y Position	Joker Card 3 Position Register
35	[31:16]: Joker Card 4 X Position, [15:0]: Joker Card 4 Y Position	Joker Card 4 Position Register
36	[31:16]: Joker Card 5 X Position, [15:0]: Joker Card 5 Y Position	Joker Card 5 Position Register
37	[31:8]: Joker Card 1 Effect Parameters, [7:0]: Joker Card 1 Type	Joker Card 1 Effect Register
38	[31:8]: Joker Card 2 Effect Parameters, [7:0]: Joker Card 2 Type	Joker Card 2 Effect Register
39	[31:8]: Joker Card 3 Effect Parameters, [7:0]: Joker Card 3 Type	Joker Card 3 Effect Register
40	[31:8]: Joker Card 4 Effect Parameters, [7:0]: Joker Card 4 Type	Joker Card 4 Effect Register
41	[31:8]: Joker Card 5 Effect Parameters, [7:0]: Joker Card 5 Type	Joker Card 5 Effect Register
42	[31:0]: Tile Map Base Address	Graphics Base Address Register
43	[31:0]: Sprite Data Base Address	Sprite Data Register

44	Bit 0: VGA Enable Bit 1: Double Buffering Enable Bit 2-7: Reserved	VGA Control Register
45	[7:0]: Interrupt Status Flags Bit 0: VSync Interrupt - Set when a vertical sync occurs in the VGA signal (typically 60Hz), useful for timing game updates, Bit 1: Controller Input Interrupt - Set when a button press is detected on the NES controller, Bit 2: Timer Interrupt - Set when an internal timer expires (for animations, game events, etc.), Bit 3: Game State Change Interrupt - Set when the game state changes (new round, game over, etc.), Bit 4: Score Update Interrupt - Set when the player's score changes, Bit 5: Hand Completion Interrupt - Set when a poker hand is completed, Bit 6: System Error Interrupt - Set when a system error occurs, Bit 7: Reserved for future use	Interrupt Status Register
46	[7:0]: Interrupt Mask Bits Bit 0: VSync Interrupt Mask - When set to '1', the VSync interrupt is enabled, Bit 1: Controller Input Interrupt Mask - When set to '1', the controller interrupt is enabled, Bit 2: Timer Interrupt Mask - When set to '1', the timer interrupt is enabled, Bit 3: Game State Change Interrupt Mask - When set to	Interrupt Mask Register

	'1', the game state change interrupt is enabled, Bit 4: Score Update Interrupt Mask - When set to '1', the score update interrupt is enabled, Bit 5: Hand Completion Interrupt Mask - When set to '1', the hand completion interrupt is enabled, Bit 6: System Error Interrupt Mask - When set to '1', the system error interrupt is enabled, Bit 7: Global Interrupt Enable - When set to '1', all interrupts are enabled; when '0', all interrupts are disabled	
47 - 63	Future Use	Future Use