CSEE 4840 Project Proposal: Roguelike Poker Deckbuilder Game

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

Spring 2025
A Roguelike Poker Deckbuilder on an FPGA with VGA Output, Controller Support, and Audio
Inspired by Balatro

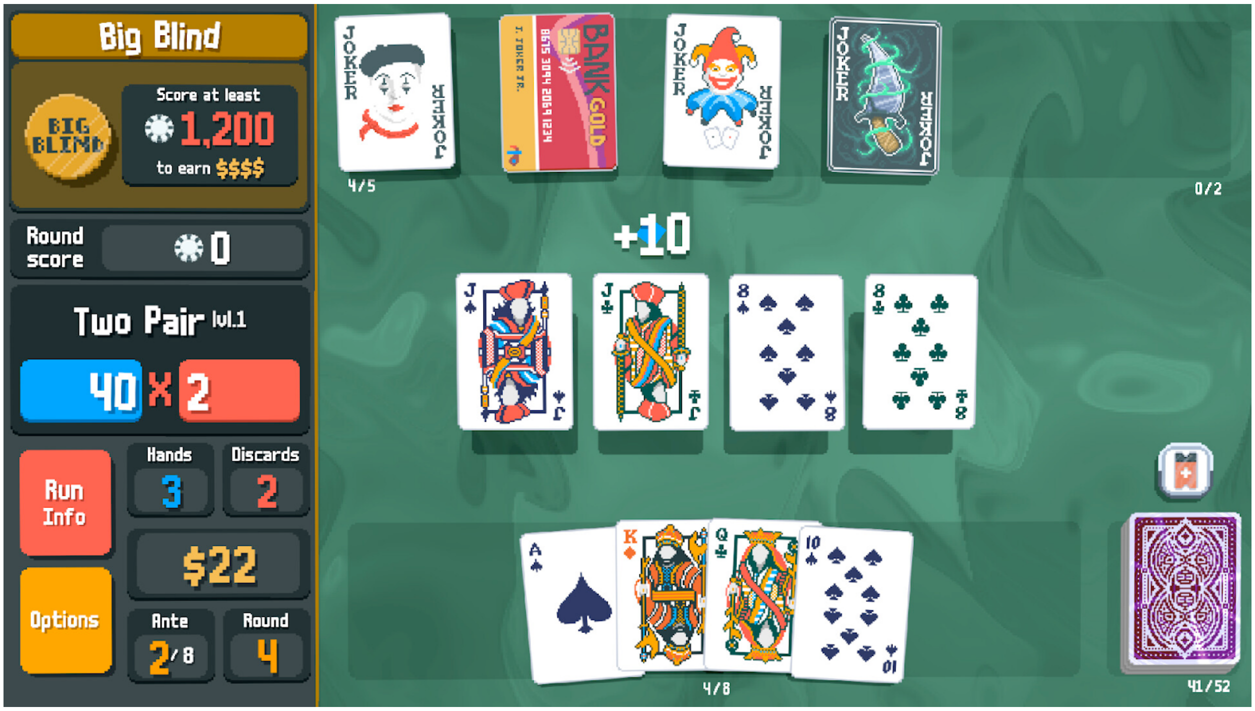

Figure 1: Balatro - Two Pair

Overview

This project aims to design, develop, and implement a 2D roguelike poker deckbuilder game (similar to Balatro) on the provided FPGA. We plan to generate video output via the VGA monitor at a target frame rate of 30 FPS, support controller input through USB, and audio output.

Materials

1. FPGA Altera Cyclone V
2. VGA Monitor
3. USB Controller: NES-Style or Similar (USB Support)
4. SD Card with Ubuntu or Similar Linux Image
5. Speaker for Audio/Music

Game Flow & Mechanics

1. Start/Menu Screen
    a. Player can select a ‘Deck’ that will give them a modifier
        i. Such as:
            1. An additional draw for the ‘run’
            2. An additional hand for the ‘run’
            3. Additional money for the ‘run’
2. Deck & Cards
    a. Uses poker suits and values
    b. Includes special ‘modifier cards’
        i. Consumables:
        ii. Jokers, Tarots, Planet Cards
            1. These can affect multipliers and or chip counts.
    c. Basic actions:
        i. Shuffle
        ii. Draw
        iii. Discard
        iv. Hand Evaluation: (Pairs, Straight, Flush, etc.)
3. Encounters / “Blinds”
    a. Each ‘blind’ is an encounter that increases with difficulty
    b. Beat the blind by creating strong poker hands, winning chips, and meeting a chip threshold
4. Progression & Shop
    a. After defeating each blind, earn money
    b. Use money to buy jokers and consumables
    c. Progress to more challenging blinds until the final boss or run fails.
5. Victory / Defeat
    a. Survive all blinds or up until you lose to a blind
    b. Summary of performance upon conclusion

Scope

1. Game Logic
    a. Implement basic deck logic: Shuffle, Draw, Discard, Evaluate, Card Effects
        i. Card Effects Include:
            1. Multiplier Effects: e.g., a Joker doubles final chip count if in hand
            2. Chip Count: each card has a ‘chip’ value that will be added per hand
    b. Implement a rudimentary roguelike enemy and progression system
        i. Earn money by defeating a blind
        ii. Allowed to buy ‘Jokers’, and Consumables: ‘Tarots’ and ‘Planet Cards’
    c. Simple/Basic asset, sprite, and text system
2. Video Output
    a. VGA Monitor at a resolution allowed by the memory available on the FPGA running at 30 FPS
    b. Render game elements
        i. Cards, Text, Sprites, Colors
3. Audio Output
    a. Music to Accompany the Game
4. User Input
    a. Support a Programmable USB Controller
    b. Interpret button presses to select cards, navigate menus, and progress through the game
5. Hardware/Software
    a. We will preferably use the Cyclone V’s SoC to run the game logic on the ARM processor while the FPGA handles VGA, input, and any acceleration we might add.

Hardware-Software Interface

1. Game Logic on ARM
2. FPGA for VGA
3. USB Controller Input Handling
4. Audio Output
5. Memory and Data Storage

Milestones

1. Basic VGA Sprite Displayed
2. Card Rendering and Controller Input
3. Core Game Logic
4. Audio Integration
5. Final Demo

Challenges

- Framerate
- Memory Handling
- Controller USB Input
- Implementing Audio
- Correct pixel sizing