BalatJack – A Scaling Roguelike Card Game

Summary

BalatJack builds on the simplicity of blackjack with the infinitesimal expansibility of Balatro to form a single-player, run-based rogue-lite game. Players chase ever-increasing chip goals by stacking card-modifying Jokers that escalate from modest perks to run-ending score explosions (targeting the maximum float in scientific notation, naneinf). This attempts to merge the pick-up-and-play clarity of Blackjack with the endless-combo depth of Balatro, ending up with a game that is:

- **Instinctive** – 21 is an easily recognizable target, displayed on-screen - **Replay-Ready** – There are 150 jokers, 11 special card styles & seals, 8 deck variants, and an endless mode – all randomized, with seeds.
- **Technically Light** – Builds off of many components present in the intro project, making development a simpler process.

One-Liner

BalatJack is a single-player rogue-lite that welds Blackjack's "21 or bust" snap-decisions to Balatro-style, exponentially scaling multipliers - delivering casino-tight rounds that explode into massive scores, with a feature set reasonable for our time limitations.

Problem/Opportunity

Background

- Our warm-up project proved we can render cards, handle input, and resolve Blackjack logic.
- Players already *know* Blackjack, but classic Blackjack plateaus quickly.
- Balatro showed there is a huge appetite for **scaling, numbers-go-crazy**

deck-builders.

The Opportunity

Create a minimal-learning-curve yet endlessly replayable desktop game that:

1. Showcases advanced state-management, UI, and RNG seeding
2. Provides a strong gameplay hook when the multiplier skyrockets
3. Fits comfortably inside the remaining course schedule if we keep the scope tight and expand only when time clearly allows.

Appetite

Time-box	**Calendar**	**Rationale**
5.5 weeks (1 Shape -> 3 Build cycles -> 1.5 Cool-down)	April 28 - June 6, 2025	Matches assignment deadline, allows tradeoffs

We will **cut down scope to fit this appetite** - not extend the appetite to fit the scope.

Solution Sketch

High-Level Gameplay

1. **Blind Loop** (some number of playable hands per blind)
   Hit/Stand/Split/Double/Discard -> hand resolves to chips * mult
2. **Shop**
   Buy / sell Jokers, apply Styles and Seals, take consumables, earn interest.
3. **Boss Blind** (every 3rd Blind)
   Rule twist + higher Chip goal -> beat to advance Ante.
4. **Win Condition**
   Clear Ante 8 or overflow float to roll credits and unlock next Stake.

Scope Boundaries (NWW)

Needs	**Wants**	**Would-Be-Cools**
Functional card interfaces, decks, card "display mats"	Endless mode	Daily Challenges
Score calculation	Seeded randomness	Achievements
~50 core jokers	More variants, jokers	
Single Stake tier	Consumables	
Boss Blind Mechanics	Animations	
Shop		

Systems Snapshot (within appetite)

1. **Joker Engine** - ~50 Jokers implemented; flat bonus, scaling counter, copy-trigger.
2. **Blackjack Core** - warm-up code + Split, Double, Trash actions, some logic improvements
3. **Economy** - chips -> cash -> shop, simple per-round interest mechanic.
4. **Boss Modifier Script** - data-driven rule tweaker evaluated at runtime.

Rabbit Holes & Unknowns

Risk	**Mitigation**
Balancing 50 Jokers could balloon	Stick with pre-shaped designs, limit mid-cycle tuning.
UI drag and drop edge cases	Fallback hot-key interactions via keyboard. Also improves accessibility.
Floating point errors, logical bugs	Unit tests + safe utilities such as safe_mul()
Scope creep	Postpone polish to cooldown, keep only necessary elements for roadmap progress
Conflicting code or access methods	Short meetings before branching to determine access APIs, coordinated documentation

Betting Recommendation

Place a **4-week bet** on the NWW described above. If we finish early, stretch goals (additional Jokers, extra Styles, cleaner animations/audio) can be tackled, as well as in cool-down as follow-up elements.

Open Questions

1. **Project milestones** - are we expected to regularly present milestones on the project, or build up the project for a larger end-of-class presentation?

Gameplay

The game plays out like a series of fast-paced casino “sets”, but with rogue-like escalation and deck-building strategy layered on top of classic Blackjack.

Step	What the player sees/does	Why it’s engaging
1. Set your table (run start)	Choose a deck variant, set up Stake. A 5-slot tray for Jokers sits beside the felt, initially empty.	Early decisions frame each run and hint at different playstyles.
2. Draw and decide (each hand)	The dealer gives you two cards face-up and one face down for themselves. You may Hit (draw), Stand, Split, Double, or Trash a card. Each action can trigger Joker effects (bonus chips, multiplier bumps, peeks, deck manipulation).	Blackjack familiarity keeps the learning curve minimal; Jokers add significant combo potential to each hand.3
3. Score Management	If your total is at or below 21, you earn Chips equal to that total (with some Special cases where Splits or a Blackjack are present. Those chips are then multiplied by your current Mult – a running value continually pumped up by scaling Jokers. A bust	Seeing chip values go up when multipliers chain together is the core thrill of the game, as typically called, “numbers-go crazy”-style gameplay.

	normally scores zero, but safety Jokers or card attributes can override that.	
Shop and Sculpt	After every blind, you access a shop: spend earned cash on new Jokers, apply Styles/Seals to cards, or by single-use consumables. Unspend cash earns interest, itself boostable by economy Jokers.	The shop phase is a short deckbuilding puzzle – choose to upgrade now or bank your cash to snowball later.
Blind progression	Each blind sets a Chip goal you must reach within a fixed number of hands. Boss blinds layer on rule twists (such as “No hitting on 17+” or “Glass cards shatter immediately”). Goals scale sharply through eight antes, then infinitely in Endless Mods.	Implicit time pressure to build up the deck before the Boss Blind, Blinds can showcase creative Joker counter-play and require planning in advance
Run Payoff	Meet every chip goal through Ante 8 and you win the run, unlocking new Jokers, Decks, and higher Stakes. Overshoot the goal to the point of overflowing the floating point unit and you simply win. The game is yours.	The common system of high scores drives people to learn the game and try new things, and the ability to go ‘naneinf’, or the scientific notation of the FPU having no idea what to do with that large of a number, gives a built-in bragging right.

For new players, if you can play basic Blackjack, you can start a Balatjack run in under a minute. Depth comes from experimenting with which Jokers to buy, when to split or double, and how to engineer wild scoring synergies.

Core Loop

Phase	**Player Actions**	**Key Hooks**
Blind Start	View Chip Goal and optionally Boss rule; Jokers with onBlindStart() fire	Anticipation & planning
Hands (<= H)	Hit, Stand, Split, Double, Trash	Quick tactical decisions
Resolve	Hand value -> Chips * Mult	Rising numbers gives positive feedback to the player
Shop	Buy jokers, styles, consumables, earn interest	Deck-building agency
Progress Check	Goal met – next blind; else Game Over	"One more hand" tension

For the base non-endless run, eight **Antes**, with a 1x, 2x, and typically 4x **blinds** make up the gameplay loop. Endless ante escalation unlocks after the first win.

User Experience & Visual Identity

The actual user interface is not yet defined, but several core tenets will be followed while creating this game:

- Keep zones separate. It should be clear what action occurs when a specific card is dragged into an area, how the deck is laid out, etc.
- Color-coded elements, such as blue chip values or red multiplier values, will allow for a more cohesive and intuitive play style
- Responsive drag/drop and clickable cards are the main complex interface – they may be dragged between Decks, dealt from Decks, added to a PlayedHand, or stored in a CardContainer box in the case of Jokers. Other than that, buttons are the main form of interaction in gameplay.

Development Path

1. Implement core Blackjack engine, card rendering, and basic UI based on warmup project
2. Joker framework, scaling counters, save JSON schema
3. Styles & Seals, Shop, Economy/Interest
4. Implement a subset of 50 jokers, stake logic, boss blind scripting
    a. Remaining jokers implemented as time allows.
5. Seed system, overflow handling, shop
6. Endless mode, art polish, analytics hook
7. Closed beta (shared among classmates/friends/family)
8. Submission

This is currently up for rework and revision, given the short timescale we have to complete the project.

User Personas

1. Casual Card Gamer (Jack, 28)

- Enjoys games like Solitaire, Poker, and Blackjack.
- Wants a fast-paced experience that feels familiar but offers surprises.
- Attracted to easy learning curves and recognizable rules.

2. Card combo chaser(steve, 24)

- Loves building combos and theory crafting
- Will try to break the game
- Play games like Bolatro and Slay the Spire

3. Gambler and card player(bob, 44)

- Loves gambling
- Blackjack is right up his alley
- Might be confused by new things
- Prone to overstimulation

Document Reference (for editability):

BalatJack