Danz: Impermanent Loss

Product Requirements Document
Version 1.0 | December 2024

Overview

Impermanent Loss is a gamified trading experience within the Danz ecosystem that transforms DeFi mechanics into a spectator-friendly "dance battle" between creator coins. Users pair two tokens, stake them together, and watch them compete over a set duration. At resolution, the losing token is liquidated into the winner—turning impermanent loss from a DeFi pain point into the core game mechanic.

Platform: Farcaster Mini App (Base ecosystem)
Foundation: Built on danz-web layout and design system

Core Concept

The name "Impermanent Loss" is intentional. In traditional DeFi, impermanent loss is something users try to avoid. Here, it's the point—you're guaranteed to end up concentrated in whatever won the dance.

User Types

Pool Players (Dancers)
Stake capital by pairing two creator coins. Higher risk, direct skin in the game. Exit with winner tokens before betting treasury executes.

Bettors (The Crowd)
Bet ETH/USDC on dance outcomes. Lower barrier to entry. Receive winner tokens + $DANZ points after treasury execution.

Game Flow

Phase 1: Dance Setup

A dance is created with two creator coins from the Base ecosystem. Parameters defined at creation:

- **Duration**: Fixed time (e.g., 3 days) OR volatility threshold (e.g., 50% price divergence)
- **Entry window**: How long users can join after dance starts
- **Minimum stake**: Floor for pool entry
- **Betting limits**: Min/max bet sizes

Phase 2: Entry Period

For Pool Players:

1. User selects an active dance (or creates one)
2. Deposits ETH or USDC
3. System auto-purchases equal value of both creator coins
4. Tokens staked into the dance pool
5. User receives pool position NFT/receipt

For Bettors:

1. User views active dances with live stats
2. Picks a side (Token A or Token B)
3. Deposits ETH/USDC into betting treasury
4. Bet recorded, added to visible treasury counter

Phase 3: The Dance (Active Period)

Live Display Elements:

- Real-time price charts for both tokens (the "dance")
- Current winner/score based on price movement
- Total pool size (staked liquidity)
- Betting treasury size (visible and growing)
- Bet distribution (% backing each token)
- Time remaining or distance to volatility threshold

What Users See: Two tokens charting against each other with a live treasury counter ticking up. The spectator energy of knowing that whole pot becomes buy pressure the moment it resolves.

Phase 4: Resolution

Trigger: Time expires OR volatility threshold hit

Execution Sequence:

1. **Winner Determined**
   Token with higher % price appreciation wins
2. **Pool Player Settlement** (Immediate)
   - Losing token in each position liquidated
   - Swapped to winning token via DEX
   - Users now hold only winner tokens
   - Optional: Early exit fee if user wants to exit pre-resolution
3. **Betting Treasury Execution** (Immediate after pool settlement)
   - Treasury buys winning token on market
   - Creates visible buy pressure post-resolution
   - Winner tokens distributed proportionally to correct bettors
   - $DANZ points distributed to all bettors

The Cascade Effect:
Pool players know the betting treasury is coming. Even after resolution, there's reason to hold because that buy pressure hasn't hit yet. Bettors are telegraphing future demand.

$DANZ Points System

Points accumulate in user's database record for future token launch.

Earning Points:

Action	Points	Notes
Enter pool	Base + stake multiplier	Higher stake = more points
Win pool position	Bonus multiplier	Correct side gets bonus
Place bet	Base points	Participation reward
Win bet	Bonus + bet size multiplier	Scaled to risk taken
Lose bet	Small consolation	Keeps engagement
Streak bonus	Multiplier per consecutive win	Caps at some max

Database Schema (User Record):

```
user_id: wallet_address
total_points: number
dance_history: [
  {
    dance_id,
    type: "pool" | "bet",
    side: token_address,
    amount_in,
    outcome: "win" | "lose",
    points_earned,
    timestamp
  }
]
tier: "bronze" | "silver" | "gold" | "diamond"
streak_count: number
```

Technical Architecture

Smart Contracts (Base)

DanceFactory.sol

- Creates new dance instances
- Manages global parameters

- Registry of active/completed dances

Dance.sol (per dance instance)

- Holds staked token pairs
- Tracks pool positions
- Executes resolution swaps
- Emits events for frontend

BettingTreasury.sol

- Holds betting deposits per dance
- Executes post-resolution buys
- Distributes winner tokens to bettors

Backend Services

Dance Indexer

- Listens to contract events
- Updates dance states in real-time
- Calculates live scores

Price Oracle Integration

- Pulls creator coin prices (likely via DEX reserves or existing oracle)
- Feeds live data to frontend

Points Service

- Processes resolution events
- Calculates and stores $DANZ points
- Manages user records

Frontend (danz-web extension)

New Views:

1. **Dance Browser**
    - Grid/list of active dances
    - Filter by: time remaining, pool size, token pairs
    - Quick stats: current leader, treasury size, entry status
2. **Dance Detail**
    - Live dual chart visualization
    - Pool entry interface
    - Betting interface
    - Real-time treasury counter
    - Bet distribution breakdown
    - Countdown/progress to resolution
3. **Create Dance**
    - Token pair selector (from Base creator coins)
    - Duration/threshold configuration
    - Entry parameters
4. **Portfolio**
    - Active positions (pools + bets)
    - Historical dances
    - $DANZ points balance and history
    - Tier/rank display

Farcaster Integration

Cast Moments (Distribution Vectors)

On Pool Entry:

> "🕺 Just entered the dance floor—$CREATOR1 vs $CREATOR2
> Pool: X ETH | Ends: [date/threshold]
> [Join the dance →]"

On Bet Placed:

> "📣 Betting on $CREATOR1 to win
> Treasury: X ETH and growing
> [Place your bet →]"

On Win:

> "🏆 Called it! $CREATOR1 took the dance
> Won: X tokens + Y $DANZ points
> [See results →]"

Frame Actions

- Quick bet directly from cast
- View live dance stats in-frame
- Share dance to followers

UI/UX Specifications

Design System Reference

Inherits from danz-web:

- Color palette
- Typography scale
- Component library
- Animation patterns
- Spacing system

Key Visual Elements

The Dance Visualization

- Two price lines in brand colors
- Animated when crossing/diverging
- Winner highlight effect

Treasury Counter

- Prominent, always visible during active dance
- Animated increment on new bets
- Pulse effect approaching resolution

Entry Interfaces

- Clear two-panel layout for pool vs betting
- Amount input with token balance display
- Confirmation with fee/slippage preview

Mobile-First Considerations

- Bottom sheet for actions
- Swipe between dances
- Haptic feedback on bets/entries
- Push notifications for resolution

Edge Cases & Rules

Early Exit (Pool Players)

- Allowed pre-resolution with fee (e.g., 2-5%)
- User receives current value of both tokens
- Exits the game, no resolution payout

Ties

- If tokens end within X% threshold, both sides win
- Pool players keep original tokens (no swap)
- Betting treasury splits proportionally

Insufficient Liquidity

- If winning token has insufficient DEX liquidity

- Resolution swaps execute in tranches
- Users notified of delayed settlement

Dance Cancellation

- If both tokens rug or delist
- All funds returned minus gas
- No points awarded

Success Metrics

Engagement

- Daily active dances
- Unique users (pool + bettors)
- Repeat participation rate
- Average session duration

Volume

- Total value locked in pools
- Betting treasury volume
- DEX volume generated at resolution

Social

- Casts generated per dance
- Frame interaction rate
- Follower growth from shares

Points

- Total points distributed
- User tier distribution
- Points velocity (rate of earning)

Development Phases

Phase 1: MVP

- Single dance type (time-based, 3 days)
- Manual pool creation (admin)
- Basic betting
- Points tracking
- Core UI views

Phase 2: Expansion

- User-created dances
- Multiple duration options
- Volatility threshold triggers
- Enhanced visualizations
- Farcaster frames

Phase 3: Advanced

- Dance series/tournaments
- Crew-based pools
- Predictive analytics
- Advanced point mechanics
- Governance preparation

Open Questions

1. **Creator coin source**: Which creator coin launchpad/protocol on Base? (friend.tech successors, Wow, etc.)
2. **Oracle reliability**: How to ensure accurate, manipulation-resistant price feeds for newer creator coins?

3. **Fee structure**: Platform fees on pool entry, betting, resolution swaps?
4. **Points conversion**: Ratio/mechanism for $DANZ points to eventual token?
5. **Regulatory**: Any compliance considerations for betting mechanics?

Appendix: User Stories

As a degen, I want to pair two creator coins I'm bullish on and let them battle, so I end up concentrated in the stronger one without having to manually trade.

As a spectator, I want to bet on dance outcomes with friends, so we have something to watch and root for together.

As a creator coin holder, I want to see my token in dances, so it gets exposure and trading volume.

As a points farmer, I want to participate in as many dances as possible to accumulate $DANZ before the token launch.

Document maintained by Danz team. Last updated December 2024.