🎰 HARLEQUIN'S FORTUNE - SLOT MACHINE PROJECT SUMMARY

Portfolio Demo Slot Game
Developer: Alena Pumprová
Year: 2025
Status: Near completion

📊 PROJECT OVERVIEW

Fully functional slot machine game built with React and TypeScript, featuring professional casino mechanics, sound effects, and persistent storage.

✅ COMPLETED FEATURES

🎮 Core Game Mechanics

- **5x3 reel layout** with 10 paylines
- **Weighted random symbol generation** for realistic gameplay
- **Win calculation system** supporting:
  - Regular payline wins (3+ symbols from left)
  - Wild substitution (Harlequin)
  - Scatter wins (Diamond - pays anywhere)
- **Multiple bet options:** 10, 20, 50, 100, 200 credits
- **Auto-spin functionality** with automatic stop on low credits
- **Credit refill system** when balance runs out

🎨 Visual & UI

- Professional slot machine design with Harlequin theme
- Animated reel spinning with staggered stop timing (1.55s - 2.7s)
- **Win highlighting -** winning positions light up after spin
- Responsive bottom control panel
- Modal windows (Menu, No Credit)
- Sound toggle and menu buttons

🔉 Sound System

- **Complete audio implementation:**

    - Button click sounds
    - Spin sound effect
    - Win celebration sound
    - Error sound (insufficient credits)
    - Menu whoosh effect
    - Background music (waltz) at 30% volume
    - Refill credits sound
- Sound preloading for smooth playback
- Mute/unmute functionality
- Automatic sound cleanup

💾 Persistent Storage (Local Storage)

- **Credit balance** saved across sessions
- **Bet amount** saved
- **Sound preferences** saved
- Custom `useLocalStorage` hook implementation
- Automatic sync with React state

🎲 Game Balance & Math

Final Configuration (RTP ~91-92%):

Symbol Type	Payouts (3/4/5)	Weight	Notes
Cards (Spades, Clubs, Diamonds, Hearts)	4 / 8 / 16	90	Low value, frequent
Diamond Cards (4 variants)	6 / 14 / 35	65	Medium value
Hyena	12 / 30 / 90	40	High value
Diamond (Scatter)	3 / 8 / 25	15	Pays anywhere
Harlequin (Wild)	20 / 60 / 300	10	Substitutes all except scatter

Game Statistics (100k spin simulation):

- **RTP:** 91.40% ✅ (Above legal minimum of 85% in Czech Republic)
- **Hit Frequency:** 14.08% (1 win per ~7 spins)
- **Volatility:** Medium-High
- **Max Win:** ~90x bet

- **Average Win:** 6.5x bet when hitting

Paylines (10 total):

1. Middle row (straight)
2. Top row (straight)
3. Bottom row (straight)
4. V-shape
5. Λ-shape (inverted V)
6. Top zigzag
7. Bottom zigzag
8. Increasing diagonal
9. Decreasing diagonal
10. W-shape

🛠️ TECHNICAL IMPLEMENTATION

Tech Stack

- **React** with TypeScript
- **CSS** for styling and animations
- **Custom Hooks:**
  - `useSound` - Audio management
  - `useLocalStorage` - Persistent state
- **React Icons** (FaXmark for close buttons)

Key Components

- **GameField** - Main game logic and state management
- **Reels** - Reel display and animations
- **BottomPanel** - Control buttons and displays
- **MenuModal** - Game rules and symbol information
- **NoCreditModal** - Credit refill prompt

Critical Bug Fixes

1. **Autospin credit bug** - Fixed using `useRef` for real-time credit tracking
2. **State closure issues** - Resolved with proper ref usage in setTimeout
3. **Credit deduction in autospin** - Now correctly deducts each spin

File Structure

```
/src
 /components
  - GameField.tsx (main game logic)
  - Reels.tsx
  - BottomPanel.tsx
  - MenuModal.tsx
  - NoCreditModal.tsx
 /hooks
  - useSound.ts
  - useLocalStorage.ts
 /data
  - data.ts (symbols, paylines, game config)
 /img/symbols
  - [11 symbol images]
 /sounds
  - button.mp3
  - spin2.mp3
  - win.mp3
  - error.mp3
  - whoosh.mp3
  - waltz.mp3
  - refill.mp3
```

🎯 GAME BALANCE JOURNEY

Initial Problem

- Started with **RTP 221%** (player winning!) due to:
  - Scatter payouts too high (50/100/500)
  - Only 3 paylines (too few opportunities)
  - Requirement of only 2 symbols for win

Iterations

1. **Added 10 paylines** (from 3)
2. **Changed win requirement** to 3+ symbols (from 2+)
3. **Reduced scatter payouts** dramatically (to 3/8/25)
4. **Balanced all symbol payouts** through extensive testing
5. **Adjusted symbol weights** for proper frequency

Final Result

- **Legal RTP:** 91.4% (above 85% minimum)
- **Player experience:** Challenging but fair
- **Casino profit:** ~8.6% long-term

📚 LEARNING OUTCOMES

Concepts Mastered

1. **Local Storage API -** Persistent browser storage
2. **React useRef -** Handling state in closures/setTimeout
3. **Custom Hooks -** Reusable logic (useLocalStorage, useSound)
4. **TypeScript Generics -** `<T>` type parameters
5. **Audio in React -** Preloading, playback, cleanup
6. **Game Math -** RTP calculation, probability, volatility
7. **Monte Carlo Simulation -** Testing with 100k+ spins

Casino Math Insights

- Understanding RTP (Return to Player)
- Legal requirements (85% minimum in CZ)
- Volatility vs Hit Frequency trade-offs
- Symbol weighting and probability
- Scatter mechanics (pays anywhere vs paylines)
- Wild substitution logic

⚠️ KNOWN LIMITATIONS

1. **Hit Frequency:** 14% is relatively low
   - Would need more paylines (15-20) or different game mechanics to improve
   - Acceptable for medium-high volatility slot
2. **Mobile Optimization:** May need additional responsive work
3. **Animations:** Basic reel spinning, could be enhanced with:
   - Symbol anticipation effects
   - More elaborate win celebrations

        - Particle effects

🚀 POTENTIAL FUTURE ENHANCEMENTS

Gameplay

□ Free spins bonus round (triggered by scatters)
□ Multiplier mechanics
□ Gamble feature (double or nothing)
□ Progressive jackpot
□ Achievement system
□ Leaderboards

Technical

□ More paylines (15-25)
□ Better mobile responsiveness
□ Tablet landscape mode
□ Animation improvements
□ Loading screen
□ Tutorial mode

Analytics

□ Session statistics (total spins, total won/lost)
□ Biggest win tracking
□ Win history
□ Export game statistics

📖 MENU INFORMATION

Current Menu Content:

- About section (credits, year, demo warning)
- Symbol payouts table
- Game rules:
    - Match 3+ symbols from left to right
    - Wild substitution rules
    - Scatter pays anywhere
    - 10 paylines
    - Bet range: 10-200

  - Starting credit: 1000
  - Wins multiplied by bet

🎓 PROJECT SUMMARY

This project successfully demonstrates:

- **Full-stack game development** (frontend)
- **Complex state management** in React
- **Mathematical modeling** and balancing
- **Professional audio integration**
- **Data persistence** techniques
- **TypeScript** best practices
- **Custom hook development**
- **Problem-solving** (multiple critical bugs resolved)

Total Development Time: ~8-10 hours of active work
Lines of Code: ~1,500+ (estimated)
Testing: 100,000+ spins simulated for balance verification

📝 NOTES

- Game is **demo only -** no real money involved
- Complies with Czech gambling regulations (RTP > 85%)
- All assets and code are original work
- Suitable for **portfolio demonstration**

Status: ✅ **Production Ready** (with noted limitations)

Last Updated: January 20, 2026