

Slot Machine by Akos and Itmam

A simple, interactive slot machine game implemented in Oracle PL/SQL. It uses three packages to separate concerns: `slot_pkg` (coordinator/state), `menu_pkg` (UI/menu), and `game_pkg` (game mechanics). Data for payouts and rarity lives in the `symbols` table.

Quick Start

1. Install objects (tables + packages):

- Run `setup.sql`

2. Play the game:

- Run `game.sql`.

Example:

```
-- In SQL*Plus or SQLcl
@setup.sql
@game.sql
```

How To Play

- Start from the main menu in `game.sql`: choose options 0–6.
- Place a bet (default: 1 coin). Change it with:

```
EXEC slot_pkg.change_bet(5);
```

- Choose "Play" to spin the reels. Your balance updates based on the result.
- View payouts, rules, balance, or statistics from the menu.
- Exit with option 0.

Starting balance is 1000 coins. Bet must be between 1 and 100 and not exceed current balance.

Payouts (Summary)

- Triple match: `winnings = bet × symbol base_value`. Diamonds 💎 pay 100x, Clover 🍀 50x, etc.
- Double match: at least $2 \times \text{bet}$ (reduced payout based on symbol value).
- Three Stop 🚫 symbols: 0 payout.

See the full table via:

```
EXEC slot_pkg.show_payouts;
```

Core Functions

`slot_pkg` (Coordinator)

- `print_menu_gui()`: Shows main menu.

- `handle_menu_choice(p_choice)`: Routes menu actions; maintains state.
- `show_payouts()`, `show_rules()`, `show_balance()`, `show_statistics()`.
- `change_bet(p_new_bet)`: Validates and updates current bet.
- `reset_game()`: Restores balance (1000), bet (1), and clears stats.
- `get_balance()`, `get_current_bet()`: Accessors.

menu_pkg (Menu/UI)

- `print_menu_gui()`: Menu ASCII UI.
- `handle_menu_choice(p_choice, p_balance IN OUT, p_current_bet IN OUT, p_stats IN OUT)`: Validates funds; calls spin.
- `show_payouts()`: Prints payout table from symbols.
- `show_rules()`: Summarizes rules and special symbols.
- `show_balance(balance, bet, total_won, total_bet)`: Current status.
- `change_bet(new_bet, balance, current_bet IN OUT)`: Range and balance checks.
- `show_statistics(balance, bet, stats)`: Win rate, biggest win, net result.

game_pkg (Mechanics)

- `print_slot_machine_gui(r1, r2, r3)`: Slot UI for a spin.
- `get_random_symbol()`: Random symbol (🍒, 🍉, ..., 💡, 💬).
- `pad_symbol(sym)`: UI spacing helper.
- `evaluate_win(r1, r2, r3, bet := 1) RETURN NUMBER`: Computes winnings using `symbols.base_value` and match type.
- `spin_reels(bet, balance IN OUT, stats IN OUT)`: Deducts bet, spins, evaluates, updates balance and stats.

Data Model

- [tables/symbols_table.sql](#)
 - `symbols(symbol_char, symbol_name, base_value, rarity)` controls payouts.
 - `win_combinations(combo_type, multiplier, description)` is provided for future extensions.