Film Title Prediction Game

A JavaFX game predicting film titles from actor pairings.

Agenda

- Introduction to the Film Title Game
- Technical Architecture: JavaFX & MySQL
- Interactive Gameplay & Hint System
- Development Environment & Deployment
- Strategic Hint Design Discussion

Actors' interaction on stage, hinting at film title prediction.

Film Title Prediction Game

This JavaFX application challenges users to deduce film titles from actor pairings. It features a strategic hint system and a responsive UI for an engaging interactive experience.

Technological Foundation

- JavaFX: Dynamic, cross-platform GUI for interactive gameplay.
- MySQL: Robust relational database for actor-film data.
- Data Storage: Efficient retrieval of film datasets.
- Scalability: Supports expanding actor and movie information.

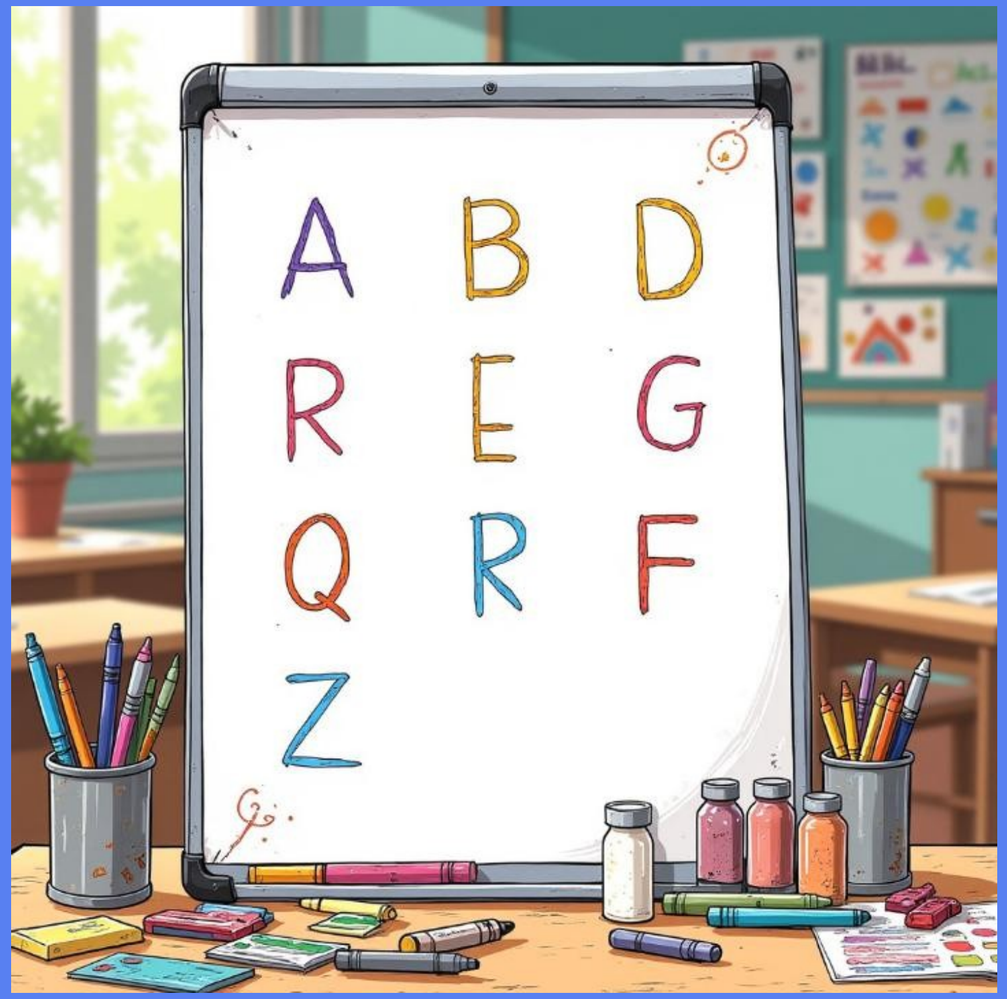
Interactive User Interface with JavaFX

Our JavaFX UI leverages FXML for responsive layouts, ensuring smooth interaction. Components like `Button` for submission and `TextField` for guesses provide intuitive navigation. Visual feedback, such as 'Correct!' messages, enhances gameplay.

Persistent Data Storage with MySQL

MySQL provides persistent storage for our game's actor-film data. We define tables like `Movies (movie_id, title, genre)` and `Actors (actor_id, name)`, linked by a `MovieActors` junction table, enabling efficient querying for game mechanics.

Actors and crew filming a cinematic scene with period costumes.



Colorful letters on a whiteboard in a classroom.

Strategic Hint System

- Reveals a letter, e.g., 'S_eepless in Seattle'.
- Provides movie genre, e.g., 'Romantic Comedy'.
- Lists another co-star for context.
- Balances game challenge with essential support.

Optimized Game Usability

Optimized usability ensures an engaging player experience. Features include clear on-screen instructions, immediate feedback for guesses, and robust error handling for invalid inputs. This design prevents frustration and maintains player flow.

Interactive Sudoku game design with players and urban background.

Financial data analysis on laptop, reflecting startup data.

Application Launch & Data Load

- JavaFX application initializes.
- Establishes MySQL database connection.
- Fetches curated actor-film data.
- Loads data into memory for gameplay.

User Interaction Cycle

The system displays a random actor pair, like 'Leonardo DiCaprio' and 'Kate Winslet'. Users then input their movie guess, such as 'Titanic', into a text field, initiating the validation process.

Guess Validation & Feedback

The system validates user guesses against the MySQL database, providing immediate feedback like "Correct!" or "Try again!". Players can also request hints, which incur a coin cost.

Website hosting and data analysis with server infrastructure and cloud.

Development Environment Prerequisites

- JDK 11+: Core Java runtime and development tools.
- JavaFX SDK 17+: UI toolkit for rich client applications.
- MySQL Server 8.0+: Database for actor-film data.
- Configured MySQL database schema for project data.

Organizational structure diagram with color gradient.

Database Schema Configuration

The MySQL schema requires `movies`, `actors`, and `movie_actors` tables. Define primary and foreign keys, like `movie_id` in `movies` and `actor_id` in `actors`, to establish relationships. Populate these tables with initial film and actor data for game functionality.

Deployment and Execution

Deployment involves updating database connection parameters like `db.url` and `db.username` in the source. Compile the JavaFX application into an executable JAR file. Finally, run this JAR directly from the command line or within an IDE.

Game Launch & Actor Display

Players start by launching the JavaFX application. The game immediately presents actor names, such as "Ryan Gosling" or "Emma Stone," prompting the user for a film title. This initial view sets the stage for the guessing challenge.

Submitting Your Guess

Users input their movie title guess, e.g., "La La Land," into the on-screen text field. They then click the "Submit" button to validate their answer. This interactive process drives the core gameplay loop.

Simple interest growth: coins accumulating over time.

Coin System & Hint Economy

- Initial balance: 50 game coins.
- Correct guess: Earn +15 coins.
- Hint request: Costs 5 coins.
- Encourages strategic hint usage.

Strategic Hint Design: Balancing Engagement and Challenge?

Given a 5-coin hint cost, how can we design diverse hint types—e.g., revealing a letter, genre, or co-star—to optimize player engagement and learning, while preserving game challenge and fostering thoughtful deduction?