

Movie Name Game Project Instructions

1. Project Introduction

Project Title: Movie Name Game

Main Objective: Replicate the behavior of the game hosted at cine2nerdle.app/battle using movie data from TMDb.

General Skills:

- Practice with version control systems
- Group work collaboration
- Implementation of software design patterns
- Open-ended project planning

Specific Skills:

- Building data indexes
- Working with Model-View-Controller design pattern
- Working with text-based user interfaces (TUIs)

Optional "Stretch" Skills:

- Working with APIs
- Implementing caching mechanisms
- Managing state for multiplayer games

2. General Requirements & Restrictions

Team Structure

- Teams will have 3 members
- All members must contribute, verified by peer evaluations and GitHub history

Version Control

- Teams must use GitHub project management tools
- Projects should be broken down into discrete tasks. These tasks will be managed & assigned using GitHub and tracked from planning through completion
- Regular commits/pull requests from feature branches expected

Technical Requirements

- All code must be written in Java (firm requirement)

Timeline

- First proposal due: March 7
- Feedback provided by: March 19
- Required check-in meeting by: April 11 (to review planning & design choices)
- Project code due: May 12
- Project presentations to TAs: Until May 13 (can be scheduled after May 1 with group & TA discretion)

Documentation

- All code must be documented with comments and tests
- Fully featured JavaDocs included as optional "completeness" feature
- Work with outside resources (books, websites, LLMs/Copilot) is permitted but must be cited

3. Specific Requirements for this Project

Data Sources

- Data sourced from TMDB (themoviedb.org)
 - Basic implementation can use "TMDB 5000" dataset from Kaggle (CSV download)
 - Optional extension: Direct API access to TMDB

User Interface

- Simple copy of the cine2nerdle game
- Layout decisions are up to you but should be readable and clean
- When typing movie names, autocomplete suggestions should be provided
- Minimum display requirements during gameplay:
 - Both users' names, win conditions, and progress
 - History of most recent five movies played (including links between them, title, release year, and genres)
 - Number of rounds played

Game Mechanics

- Game starts with an arbitrarily selected movie
- Players designated as player 1 and player 2
- Active player has 30 seconds to name a movie "connected" to the previous movie
- Connections can be: shared actor, director, writer, cinematographer, or composer

- Each movie can be used only once. Any specific connection can only be used up to three times
- Players choose win conditions before starting (default: genre-based)
 - e.g. Player wins by naming five horror movies or five romance movies
- A player loses if they cannot connect to another movie within the time limit

Required Project Features

- Implementation of Model-View-Controller design pattern
- Implementation of one additional design pattern (team choice)
- Development and documentation of data indexes
- Program must be executable with simple command (e.g., `java MovieNameGame`)
- Program must be *fast* after startup
 - Autocomplete must be real-time
 - User plays must be accepted & processed or rejected instantly.

Optional Features

- Win conditions related to specific actors/directors
- "Items" or "Powerups" that can be deployed on a user's turn, e.g.
 - a "block", which prevents the other player from naming a movie that contributes to their win condition
 - a "skip", which passes the movie back to the other player for them to connect from
 - an "escape", which selects another popular movie for the other player to connect from
- Dynamic API queries to TMDB
- Caching scheme to prevent repeated lookups/API calls
- Local networking for multiplayer functionality
- Other team-proposed game features

4. Evaluation Scheme

Code Quality

- 80% testing coverage required
- Commented code required

Documentation

- Design document with class diagrams for all included classes

Presentation

- Required sections:

- Project design
- Initial project plans
- Choices made for optional requirements
- Project demo
- Challenges faced/project retrospective
- TA Q&A

Team Assessment

- Peer evaluation via Google form

5. Other Resources

- [The game itself](#)
- [Library for text-based applications](#)
- [Overview of MVC architecture and an influential paper about the idea](#)