Final Project

- Due Mar 20 by 11:59pm
- Points 100
- Submitting a file upload
- Available after Jan 13 at 12:01am

FINAL PROJECT

Due dates: See below

Description

- Design and implement an extensible Tile-Matching Game Environment (TMGE). We will adhere to the definition of a tile-matching game as used in this Wikipedia article (Links to an external site.).
- With a team of 8 or 9

Requirements

- The TMGE should accommodate any tile-matching game that involves a grid layout and game elements on this layout, including games such as Tetris, Klax, Bejeweled, Bust-a-Move, Puzzle Bobble, Candy Crush, Dr. Mario, Puzzle Fighter, etc.
- The TMGE should make it as easy as possible to create implementations of new games.
- The TMGE should provide a defined interface that all games built on top of the environment must follow.
- The TMGE should support two players running on the same local machine.
    - If your game supports networked multiplayer, you can get up to 5% extra credit for this project.
- The TMGE should support personal player profiles (the specifics of which are up to you). Login can be very simple and does not have to be secure.
- The TMGE need only support 2-player games (but you can support more players if you want to).
- The TMGE should work by providing a player with a list of games they can play and allow them to choose which one to start.
- The TMGE games should be GUI-based.
- The TMGE should be written in Java with modules specified and implemented using the Java Platform Module System.
    - Up to 5% extra credit for this assignment if you use Java modules that consume or provide services, by leveraging the uses and provides directives, in a sensible and effective manner

- The TMGE should only expose parts of itself necessary to build a game, hiding the internals of the TMGE that need not be used or extended by a game directly.
- The TMGE only needs to support running one game at a time.

Deliverables

- The TMGE itself
- Two or more games from the list above that are "built on top of" the TMGE.
- Documentation
- Instructions for running the game
- Code and runnable jar (via a CM repository like GitHub)
- Peer evaluations (will be made available to you)

Reuse

- Cannot pick up an existing game environment implementation
- You can reuse other components, but first, double-check with the professor

Grading Criteria

- Stakeholder: the player (how is the experience of playing a game?)
- Stakeholder: future developers of the TMGE (how is the understandability and quality of the code and design?)
- Stakeholder: game developers (how is the extensibility of the TMGE in supporting new board games? how well do you hide parts of TMGE that should not be exposed to game developers? how is the experience of building in a new game using your TMGE?)
- Stakeholder: you (what are your contributions to the project?)

Miscellaneous

- Mark clearly in your design/architecture the places that are variant (variable per game) and which are fixed
- Use a configuration management repository (this is good practice, but we will also use it to verify who wrote which code -- check in your own code!). Give the professor and the TA access via email.
- All group members do not have to present each time. You can split up the presentation work however you want.

Deadlines (all due at 11:59pm on the date specified except for finals week)

Feb. 26 and March 2:

- **Preliminary design presentations:** Present and explain your design at whatever level it currently exists. Also, discuss your progress to date and your planned next steps. Include high points, low points, and major challenges you experienced in this phase (about 10 minutes per team). These need not be formal presentations. For example, it can mostly involve discussion with instructors at your individual team tables.

March 3:

- **Preliminary design document,** including UML with description/explanation, turned into the "Final Project - Preliminary Design Document" assignment on Canvas.

March 4 and 9:

- **Detailed design presentations:** Present and explain a detailed design of your TMGE. Also, discuss your progress to date and your planned next steps. Include high points, low points, and major challenges you experienced in this phase (max 7 minutes per team).

March 9 and 11:

- **First demo:** Run your TMGE, as it currently exists. (max 7 minutes per team)

March 10:

- **Updated design document**, including UML with description/explanation, and step-by-step instructions for how to make a new game for your TMGE. Also, include a description of how and why this updated design evolved from your original design. Turn this document into the "Final Project - Updated Design Document" assignment on Canvas.

Finals Week:

- **Final demo:** Run your TMGE and implemented games in front of instructors (max 10 minutes per team)
    - **March 16, 7:00pm - 9:00pm, Room: ALP 1600 (Regular Classroom)**
    - **You can schedule times for presenting outside of the final exam time as well during finals week.**
- **Final (updated) design document and a retrospective on your design**, including UML with description/explanation, and step-by-step instructions for how to make a new game using TMGE. Also, include a description of how and why this updated design evolved from your original design. Include high points, low points, and major challenges you experienced in this phase and in the project as a whole. Turn this document into THIS assignment on Canvas.
- Peer evaluations (will be made available to you)

Suggestions

- **Split your team into sub-teams.** Because teams are large, it will be most efficient if you divide your team up based on the parts of the problem. Also, consider appointing a team lead (or two?) to make things more efficient. Be sure one of these sub-teams is dedicated to architecture/interface management between the separate parts.
- **Use a group communication/coordination tool.** g., Slack or something like it.
- **Think about what every game will have in common**, then abstract that with some kind of interface/abstract class.