Ticket to Ride Phase 0

Conceptual Model (UML Class Diagram). This is an individual project.

Objectives

1. Learn about conceptual modeling and UML class diagrams
2. Become familiar with the Ticket to Ride project

Patterns

None

Requirements

First, carefully study the rules for the Ticket to Ride game (available here). Then, create a conceptual UML class diagram for the game. Your model should include all major concepts of the game and the relationships between them. Major concepts would include things such as players, Ticket to Ride map, cities, trains, points, etc. Include multiplicity constraints on associations. Use generalization/specialization where appropriate. To enhance the clarity and readability of your conceptual model, rather than including everything in one diagram, you may create multiple smaller diagrams, each of which focuses on a subset of the model. If it helps, the same class may appear in multiple sub-diagrams or multiple times in the same sub-diagram.

Constraints

None

Help

To create, edit, and print a UML diagram you may wish to use a free tool such as lucid chart. There are many other such tools available on the internet.

- A complete overview of UML and its associated diagrams can be found here.
    - Specific information about class diagrams can be found here.
- Book: UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd Edition)
- Be careful with online tutorials. Many tend to be incomplete and focus only on design models

Deliverables

Submit a hardcopy printout of your conceptual UML class diagram(s) to the TAs.

Grading Tips

The information need not be organized in this manner but it should be present in the UML diagram.

1. Train card definitions (remember to define how locomotives differ from the others)
2. Route definition
3. Destination card definition
4. Longest Continuous Path
5. Player – Including User Name, Password, and things owned or claimed by players (there are several types of things)
6. Game – Things associated with the game including players, deck of train cards, face up train cards, and unused destination card deck
7. Syntax

- There are a few places where you should use generalization/specialization.
- Not every relation between two classes is an aggregation or composition. Aggregation and/or composition should be used sparingly.

- The relation between cities and routes should be represented using either an association class or an n-ary relation.
- Make sure you have enough constraints (multiplicity or general constraints).
- Be careful that all assumptions you are making should be captured by some kind of constraint. Avoid ambiguity. Consider using collection constraints (e.g. set, ordered set, sequence, multiset) on association ends. Some of the points in the 7 sections above are for expected constraints)
- If there is detail that cannot be explained in diagram syntax, add notes
- Your associations must be declarative, not imperative.
- Make your diagrams easy to read. Duplicate classes if necessary. You don’t have to fit the diagram on one page if that page is difficult to read and understand