

XYZ_Affair
Zicheng Zhen, Joel Ye, William Xiang
APCS2 pd10
2016-05-20

Final Project Proposal

Transit Trouble

Overview

Transit Trouble is an adaptation of the game Mini-Metro. You are the developer of a new mass transit system, and your goal is to keep the growing number of stations well serviced. You lose if the system fails to deliver passengers in a really really long time. There are various small elements of gameplay that will differ from Mini-Metro, elaborated on in features.

Mini Metro Website: <http://dinopoloclub.com/minimetro/>

Mini Metro Gameplay: <https://www.youtube.com/watch?v=jZnV-QYouvo>

Backend Structures

Nodes:

- Each train station is a single node. Attributes may include capacity, a list of lines (which the station is a part of), and shape (to determine which passengers board and exit).

(Priority) Queues:

- Each station (node) will contain a queue of passengers that continue to accumulate until a train car comes to pick them up.
- Passengers will be the same for the most part, but special characters may have priority while entering the queue.

Stacks:

- Train cars are stacks, in which the first passengers to arrive are the last to leave the train cars. While this may not be implemented in gameplay (all of the passengers that need to get off will get off, regardless of order in the stack), but it can be modified to create puzzle elements.

Linked Lists:

- Each train line is a linked list made from a series of stations (nodes). Train lines can be circularly linked and doubly linked.
- One station (node) may belong to multiple lines (linked lists), allowing for the creations of terminals where multiple

Search Trees:

- Each successive day will prompt the user to save the game.
- Saving will create a new leaf in the tree from one save state. This structure can be maintained using a search tree to store the save states. (This usage is iffy. Will only actually implement if we can track changes in the leaves as opposed to completely storing game data each time). Otherwise we will save in files.

Frontend Structures:

Hitboxes:

- Subway lines must be draggable at any point to connect to other stations
- Pausing/Saving (create a mechanism for storing map and station data)

Features:

- Random Events
 - Weather (rain, snow, ecological disasters(!)) can affect the game world.
 - VIP events disrupt parts of train lines
- Different upgrades that belong to a tech tree OR shop (for one time purchases). Different moments to give bonuses (as opposed to once per week). Upgrade points based on passage of time. Three main categories are: Capacity upgrades, More trains/lines/tunnels, Efficiency Upgrades.
 - Hiring personnel (faster passenger loading)
 - New train lines
 - Tunnel building (rivers!)
 - Customer patience
 - Station capacity
 - Train speed
 - System integrity (defense against weather events)

Algorithms:

- Train Line building. Minimal intersections but somewhat orderly layout of train lines. Detect when lines cross through stations and when they actually stop at stations.
- Detecting whether train lines needs to cross river.
- Closed path detection (special terminus needed)
- Map + station generation

Classes:

class Train	
Attributes	Methods
<ul style="list-style-type: none">- Associated train line- Stack composition of Passengers- int capacity- float velocity / physics	<ul style="list-style-type: none">- Accessors/Modifiers for attributes

class Line	
Attributes	Methods
<ul style="list-style-type: none">- LinkedList Composition of Stations	<ul style="list-style-type: none">- Accessors/Modifiers for attributes- getNextStation(int);

class Station	
Attributes	Methods
<ul style="list-style-type: none"> - int ID (integer associated with shape) - Node (underlying data structure) - int capacity 	<ul style="list-style-type: none"> - Accessors/Modifiers for attributes

class TrainLot extends Station	
Attributes	Methods
<ul style="list-style-type: none"> - Terminal lot for trains, gives direction to outgoing and receives incoming trains. 	<ul style="list-style-type: none"> - Accessors/Modifiers for attributes

class Passenger	
Attributes	Methods
<ul style="list-style-type: none"> - Shape (or whatever identification) - double waitTime 	<ul style="list-style-type: none"> - Accessors/Modifiers for attributes