

# JFreerails 1.0 Functional Specification

Luke Lindsay

last updated 31. August 2006

## Overview

FreeRails a is real time multi player strategy game where players compete to build the most powerful railroad empire.

This spec is incomplete and what is here will be revised several times before it is finalized.  
This spec does not discuss in detail how features will be implemented.

## Scenarios

In designing software, it helps to imagine a few real life stories of how actual (stereotypical) people would use it. We'll look at three scenarios.

### *Scenario 1: Jeff.*

Aged 38, splits his time between contract web development work and (real) surfing (i.e. He's an aging hippy). Played RT1, disliked RT2 and RT3. Sceptical about the quality of open source software.

### *Scenario 2: Andy.*

Young programmer, 19, rides a moped. First language not English. Never played RT1 but has played RT2 and RT3. When software doesn't work or works in a way he doesn't expect, he is quick to blame the software. Wants to get involved in the development of jfreerails and to write is own AI.

### *Scenario 3: Claudia.*

Recently left university and started working for the government (as an economist). Never played any of the RT games before. Wants something to play on her computer to distract her from her research.

## Non Goals

Realism - its a game not a simulation..

## **Postponed for later releases**

This version will not support the following:

Customisablity

Difficulty levels

Advanced graphics (including animations other than movement).

Performance

Cheat proof: don't play it with your house-mate who keeps stealing your milk

Prority Shipments

Scripting events and AI using Jython

Internet high score table

Custom Look-and-Feel

Undo building track

Rate wars

Save game compatibility between releases.

## ***Specification***

Glad you're still reading. Now lets look at the features this version will actually have..

### ***Key***

Implemented features

[Planned features](#)

### ***Sections***

Game Model

Non player changes to the Game Model

User\_Interface

Reports and Dialogue boxes

Launcher

AI

# Game Model

## Time

Each year is a representative day. So, for instance, a train travelling at an average speed of 10 mph will make two (and a bit) 200 mile trips in a year.

## Map

Two maps will be available. A map of south America - used for proper games -and a small map - used for a tutorial. The maps are divided into square tiles. Tiles are 10 miles across. Rather than let the scale vary between maps of different sized regions, different sized maps should be used.

The exact shape of mountain ranges and the distribution of terrain types will vary between games.

## Terrain

### Terrain type

Each tile on the map has a 'terrain type' e.g. Farm, desert etc.

<u>Right-of-way cost</u>	<u>Before you build track on a square, you need to purchase the right of way. Different terrain types have different ROW costs.</u>
Category	All terrain types fall into one of the following categories: Urban, River, Ocean, Hill, Country, Special, Industry, or Resource.
Cargo production	Some types of terrain produce cargo (of one or more types), e.g. A Cattle Ranch produces livestock.
Cargo consumption	Some types of terrain consume cargo, e.g. 'City' tiles consume 'Food'.
Cargo conversion	Some types of terrain convert one type of cargo to another, e.g. A Stock-Yard [vegetarians might want to stop reading..] converts livestock to food.

### Terrain Heights

Different terrain types will have different heights, I'll call this the type-height. The height of a tile will be the weighted average of the type-heights of the surrounding tiles.

## Cargo

There are a number of cargo types, e.g. Mail, Passengers, Livestock etc. Cargo types fall into one of the following categories: Mail, Passengers, Fast Freight, Slow Freight, Bulk Freight. Mail is the most sensitive to speed of delivery; Bulk Freight is the least.

## Cities

Cities of random size are added at predefined locations on the map when the game starts. [As time passes, cities grow and shrink depending the amount of cargo picked up and delivered.](#)

## Track

There are a number of track types, e.g. standard track, double track, tunnel etc. A tile can only have track of one type.

You can only built track if you have cash  $> 0$ . The cost of building track depends on the track type, the track configuration, [and the right of way cost of the terrain](#). Some track types can only be built on certain terrain, e.g. an Iron Girder bridge can only be built on river. If some track has already been built, all new track must connect to the existing track.

Track can be upgraded. The cost of upgrading track from type X to type Y is less than the cost of building track of type Y but more than the cost of Y less the cost of X.

Track can be removed. When you remove track you get some money back.

Different track types have different maintenance costs. The maintenance cost must be paid once per year.

## Bridges

	<a href="#">Price</a>	<a href="#">Train Speed</a>	<a href="#">Track Type</a>
<a href="#">Wooden Trestle</a>	<a href="#">\$50.000</a>	<a href="#">slow</a>	<a href="#">single</a>
<a href="#">Iron Girder</a>	<a href="#">\$200.000</a>	<a href="#">fast</a>	<a href="#">single</a>
<a href="#">Stone Masonry</a>	<a href="#">\$400.000</a>	<a href="#">fast</a>	<a href="#">double</a>

## Trains

Once you have built some track and a station you can build a train. You get a choice of engines and the option to add up to 6 wagons. You can build trains even if you have cash  $< 0$ . (This is to stop people getting stuck without any trains since you need trains to make money.)

Trains can be scheduled to stop at 2 or more stations.

The pseudo code below describes the behaviour of trains.

```
if(train is moving){
    if(train has reached a new tile){
        if (there is a station here){
            unload cargo demanded by station
            if(this is a scheduled stop){
                if(consist needs changing){
                    upload cargo that won't fit after changing consist
                    change consist
                }
            }
            load cargo
            setStatus(stoppedAtStation)
            departTime = currentTime + stopTime
        }else{
```

```

        start moving to next tile
    }
    }else{
        keep moving
    }
} else if(train is at station){
    if(waiting for full load){
        load any cargo
        if(full){
            update next station on schedule
            if(currentTime > departTime){
                setStatus(moving)
                start moving to next tile
            }
        }
    }else{
        if(currentTime > departTime){
            if(this was a scheduled stop){
                update next station on schedule
            }
            setStatus(moving)
            start moving to next tile
        }
    }
}
}

```

## Track Contention

On single (double) track, only one (two) train(s) can move on a tile at a time. Trains 'book' the next tile they want to enter. A train enters a tile when it is first in the queue of trains who have booked the tile and there is space on the tile for the train to move. Gridlock can never occur because when trains stop moving they do not block other trains.

## Engines

Two types of engine are available when you start the game. Three other types become available later. A train's engine can be upgraded.

## Wagons

There is one wagon type for each cargo type. A wagon can carry 40 units of cargo. Wagons are free and can be added at any station (since moving empty wagons about would be boring).

## Train Schedules

The stations a train stops at and whether it changes its wagons when it stops are governed by a train's schedule.

## Train movement

The more wagons a train is pulling, the slower it moves. The greater the amount of cargo, the slower the train. The gradient of the track also affects speed. When trains arrive at a station it stops for a few moments to load and unload cargo. Trains stop instantly (this is a simplification so we don't need to

look ahead) but speed up slowly.

## Stations

Supply and demand at a station is determined by the tile types within the station's sphere-of-influence. Different station types have different sized spheres-of-influence. The spheres-of-influence of two stations cannot overlap.

### Station Improvements

<i>Improvement type</i>	<i>Effect</i>	<i>Price</i>
<a href="#">Engine Shop</a>	<a href="#">Lets new trains be built at station</a>	<a href="#">\$100,000</a>
<a href="#">Switching yard</a>	<a href="#">Cuts time taken the change wagons by 75%</a>	<a href="#">\$50,000</a>
<a href="#">Maintenance shop</a>	<a href="#">Cuts yearly maintenance of trains that stop at station by 75%</a>	<a href="#">\$25,000</a>
<a href="#">Cargo Storage</a>	<a href="#">Prevents a certain cargo from wasting away</a>	<a href="#">Variable</a>
<a href="#">Revenue booster</a>	<a href="#">Increases revenue from a cargo t. by X%</a>	<a href="#">Variable</a>

## The Economy

[The economy alternates between 5 states.](#)

<a href="#">Economic Climate</a>	<a href="#">Base interest rate</a>	<a href="#">Worsens to</a>	<a href="#">Improves to</a>	<a href="#">Effect on track price</a>
<a href="#">Boom</a>	<a href="#">2%</a>	<a href="#">Panic</a>	<a href="#">N/A</a>	<a href="#">+33%</a>
<a href="#">Prosperity</a>	<a href="#">3%</a>	<a href="#">Recession</a>	<a href="#">Boom</a>	<a href="#">+17%</a>
<a href="#">Moderation</a>	<a href="#">4%</a>	<a href="#">Recession</a>	<a href="#">Prosperity</a>	<a href="#">No change</a>
<a href="#">Recession</a>	<a href="#">5%</a>	<a href="#">Panic</a>	<a href="#">Moderation</a>	<a href="#">-17%</a>
<a href="#">Panic</a>	<a href="#">6%</a>	<a href="#">N/A</a>	<a href="#">Recession</a>	<a href="#">-33%</a>

## Stock and Bonds

The value of a bonds is \$500k.

[The interest rate for new bonds = \(base interest rate\) + \(number of outstanding bonds\). Bonds can only be issued if this figure is <= 7.](#)

New railroads issue 1,000,000 shares at \$5 per share.

Shares are traded in 10,000 share bundles.

If the share price is >\$100 at the end of the year, stocks are split 2 for 1.

Stock price = [Net worth at start of year + 5 \* profit last year] / [ shares owned by public + 0.5 shares owned by other players]

Let profit last year = 100,000 in the first year.

When a player buys or sells shares, the price used is the price calculate after the shares have changed hands.

A transaction fee \$25,000 applies each time a bond is issued or repaid and each time a bundle of shares is bought or sold.

### **Competition between Railroads**

You can take over a rival by buying over 50% of its stock. When you have done this, you have indirect control over the other railroad. You can transfer money between the two railroads, tell the other railroad where to build track to next, and tell the other railroad to repay its bonds.

If you go on to buy 100% of the stock, you have the option to merge the with the other railroad. If you do this, you gain complete control over the other railroad. I.e. the other railroad's track, trains, and stations are added to your railroad. Once a merger has taken place, there is no way to undo it.



## Non player changes to the Game Model

### ***City Growth and Decay***

As time passes, Urban (e.g. Village, City), Industry (e.g. Factory, Steel Mill), and Resource (e.g. Coal Mine, Sugar Plantation) tiles are added and removed from cities. Example, a factory tile will relocate from New York to Boston if Boston's utility gain exceeds New York's utility loss. Utilities are calculated as follows. The routine that updates cities should run once per month.

<b><i>Category</i></b>	<b><i>Marginal Utility</i></b>	<b><i>Motivation</i></b>
Industry	<u>Number of Urban Tiles / (1+Number of Industry Tiles)^2</u>	<u>Urban tiles supply labour to industries. There are decreasing returns to scale.</u>
Resource	<u>(Units of Resource Picked Up + c) / (1 + Number of Resource Tiles)</u>	<u>Resources grow when they are exploited. There are decreasing returns to scale.</u>
Urban	<u>Units of demanded cargo delivered - k * Number of Urban Tiles / (1+Number of Industry Tiles)</u>	<u>Urban tiles value employment and delivery of cargo but are adverse to overcrowding.</u>

Industries owned by Railroads do not enter the utility calculations, so when you build an industry, it stays put!

### ***Payments for delivering cargo***

...

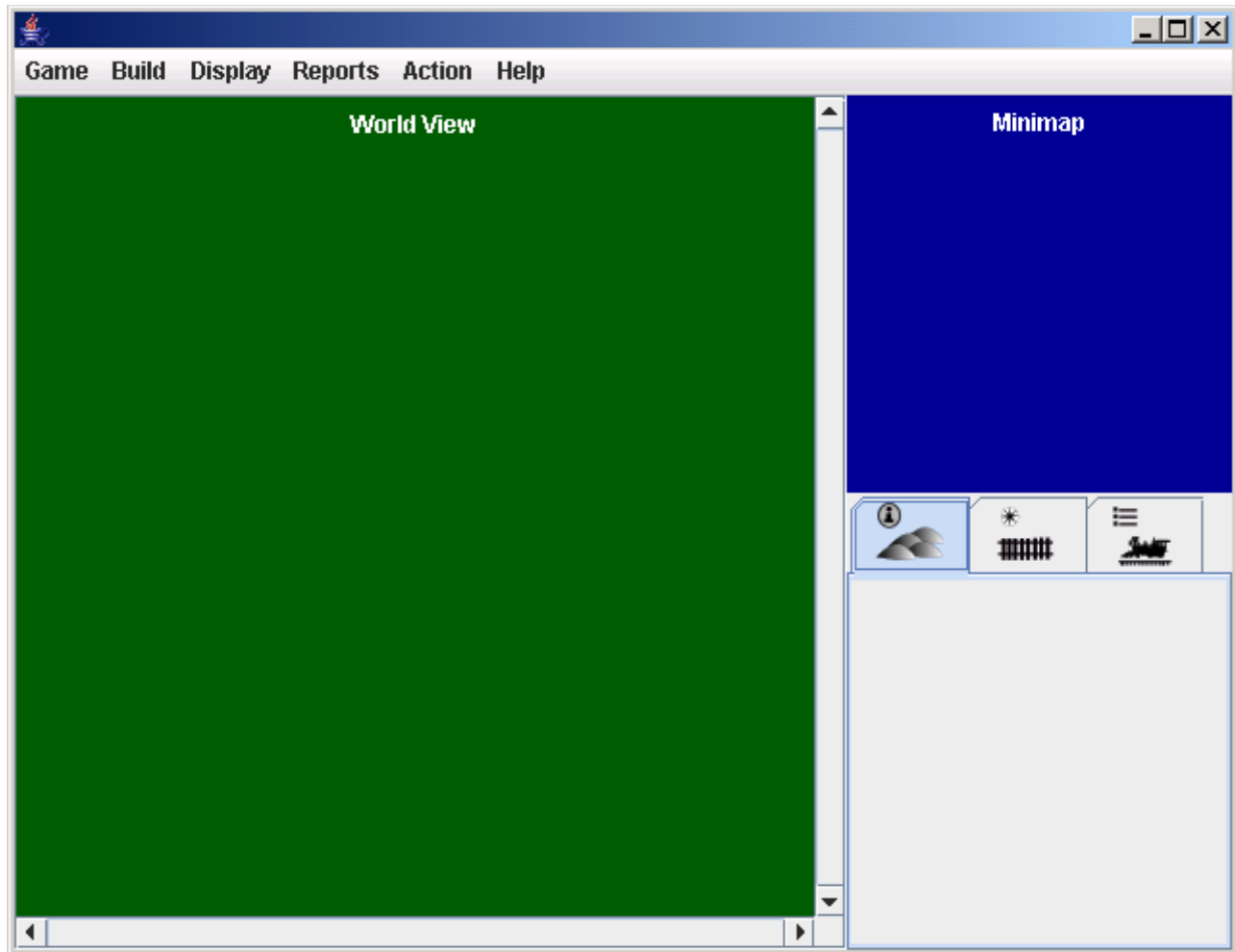
### ***Addition and removal of cargo to stations***

...

# User Interface

## Main Window

The main window has a menu bar, the world view, the mini map, and a tabpane.



The GUI components should display properly when the main window is 640 \* 480 pixels or bigger. The table below shows the dimensions of the components in terms of the width (W) and height of the main window. The figures do not include space taken up by borders, scroll bars, tabs, menus etc.

<i>Component</i>	<i>Width</i>	<i>Height</i>
Minimap	200	200
Tab's content	195	H - 300
World View	W -230	H - 70

Pressing the tab key toggles keyboard focus between the world view window and the tabpane's content.

## The menu bar

<i>Menu</i>	<i>Items</i>	<i>Details</i>
Game		
	New Game	
	Game Speed	Paused, Slow, Moderate, Fast
	Save Game	Shows prompt for filename
	Load Game	Shows list of saved games
	Exit Game	
Build		
	New train	
	<a href="#">Build industry</a>	
	<a href="#">Improve Station</a>	
<a href="#">Display</a>		
	<a href="#">Regional display</a>	
	<a href="#">Area</a>	
	<a href="#">Detailed Area</a>	
	<a href="#">Options</a>	
Reports	Balance Sheet	
	Income Statement	
	Networth Graph	
	<a href="#">Stocks</a>	
	Leaderboard	
	<a href="#">Accomplishments</a>	
Broker		
	Call Broker	
Help		
	Controls	
	Quick Start	
	<a href="#">Manual</a>	
	Report Bug	
	About	

## World View Window

The world can be displayed at 4 zoom levels:

	<i>Tile Size</i>		<i>Minimap</i>
Local detailed	30*30 pixels		Visible
<a href="#">Local</a>	<a href="#">15*15 pixels</a>		<a href="#">Visible</a>
<a href="#">Network</a>	<a href="#">Scaled so that all the player's stations are visible</a>	<a href="#">Shows trains, stations, and track but not geography</a>	<a href="#">Visible</a>
<a href="#">Regional</a>	<a href="#">Scaled so the whole map is visible</a>		<a href="#">Hidden</a>

## The Cursor

The cursor can be in one of the following modes. The cursor should only be visible if the world view has keyboard focus. The cursor's appearance should indicate which mode it is in.

The initial cursor position is 0,0. However, if a game is loaded or a new game is started and the map size is the same as the last map size, then the cursor should take the position it had on the last map.

### ***Place station mode***

The cursor gets put into place-station-mode when the player selects a station type from the build tab. Shows the radius of the selected station type.

Red when station cannot be built on selected square, white otherwise. This should be determined by whether building the station is possible, not merely whether there is track on the selected tile.

Pressing the LMB attempts to place the station. If the station is built, the cursor is returned to its previous mode; if the station is not built, the cursor remains in place-station-mode.

Pressing the RMB or pressing Esc cancels placing the station and returns the cursor to its previous mode.

### ***Build track mode***

Track can be built by dragging the mouse (moving the mouse with the LMB down). As the mouse is dragged, the proposed track is shown. Releasing the LMB builds the track. Pressing the RMB or Esc cancels any proposed track.

Track can be built by pressing the number pad keys.

### ***Remove track mode***

Track can be removed by moving the cursor with the number pad keys.

Info mode

## Components on RHS

### Minimap

## Current Cash

## Date

Shows the current year and month.

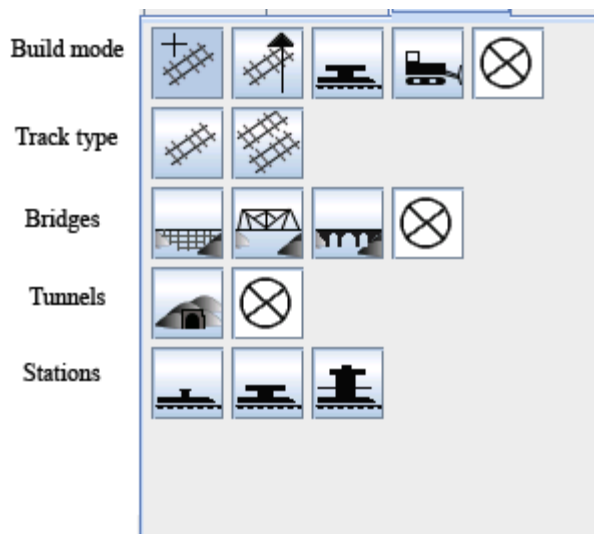
## Train Roster Tab

Shows wagons in each train and whether they are full or empty, the trains relative speed and destination. Double clicking a train on the roster (or pressing enter when the train roster has focus) or on the map opens the train report for the train.

## Build Tab

There are 5 build modes (see the table and screenshot below).

<i><b>Build mode</b></i>	<i><b>Options visible when mode is selected</b></i>	<i><b>Action</b></i>
build track	Track type, bridges, and tunnels.	When the cursor is moved, track is built. On clear terrain, the selected track type is built. On rivers, the selected bridge type is built (if a bridge type is selected.) On hills and mountains, tunnel is built if build tunnels is selected.
upgrade track	Track type and bridges	Track and bridges are upgraded to the selected type when the cursor enters a tile.
build station	Stations	
bulldoze	None	When the cursor moves from a tile to a neighbouring tile, any track connecting the two tiles is removed.
off	None	Nothing is built or removed when the cursor moves.

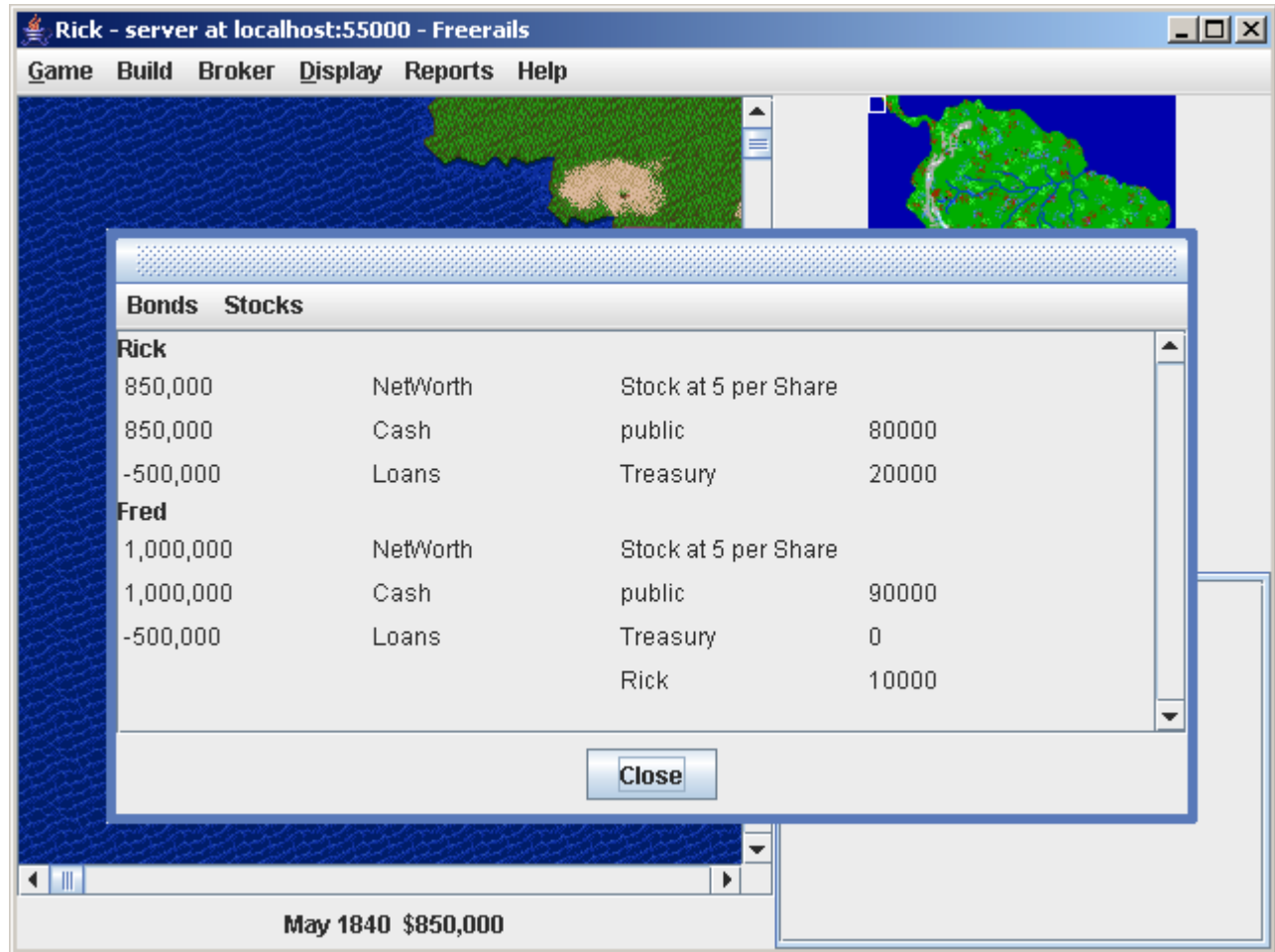


The Build tab should not accept keyboard focus when the mouse is click on it. This is because doing so would cause the world view window to lose focus which is annoying when you are building track using the keyboard.

When a new game is started or a game is loaded, the build mode should default to 'build track' with single track, wooden trestle bridges, and tunnels selected.

# Reports and Dialogue boxes







## Broker screen



## Station report






Shows information on a station. There will be 3 tabs: 'supply and demand', 'trains', and 'improvements'

### Supply and Demand Tab

	<i>Will pay for</i>	<i>Supplies /cars per year</i>	<i>Waiting for pickup /car loads</i>
<i>Mail</i>	<i>Yes</i>		
<i>Passengers</i>	<i>Yes</i>		
<i>Food</i>	<i>Yes</i>		
<i>Oil</i>	<i>No</i>		
...			

### Trains Tab

[The trains tab will list all trains that are sheduled to stop at this station. Note, if a train is scheduled to stop at the stations several times, there will be a row in the table for each sheduled stop.](#)

<i>Train</i>	<i>From</i>	<i>Delivers</i>		<i>Picks-up</i>	<i>To</i>
<i>#1</i>	<i>Rio De Janeiro</i>				<i>Barquisimeto</i>
<i>#1</i>	<i>Barquisimeto</i>			<i>automatic</i>	<i>Rio De Janeiro</i>
<i>#2</i>	<i>Barquisimeto</i>			<i>No change</i>	<i>Rio De Janeiro</i>
<i>#5</i>	<i>Barquisimeto</i>	-			<i>Barranquilla</i>
...					

### Improvements

[Shows the station improvements that have been built at this station and lets you buy additional ones.](#)

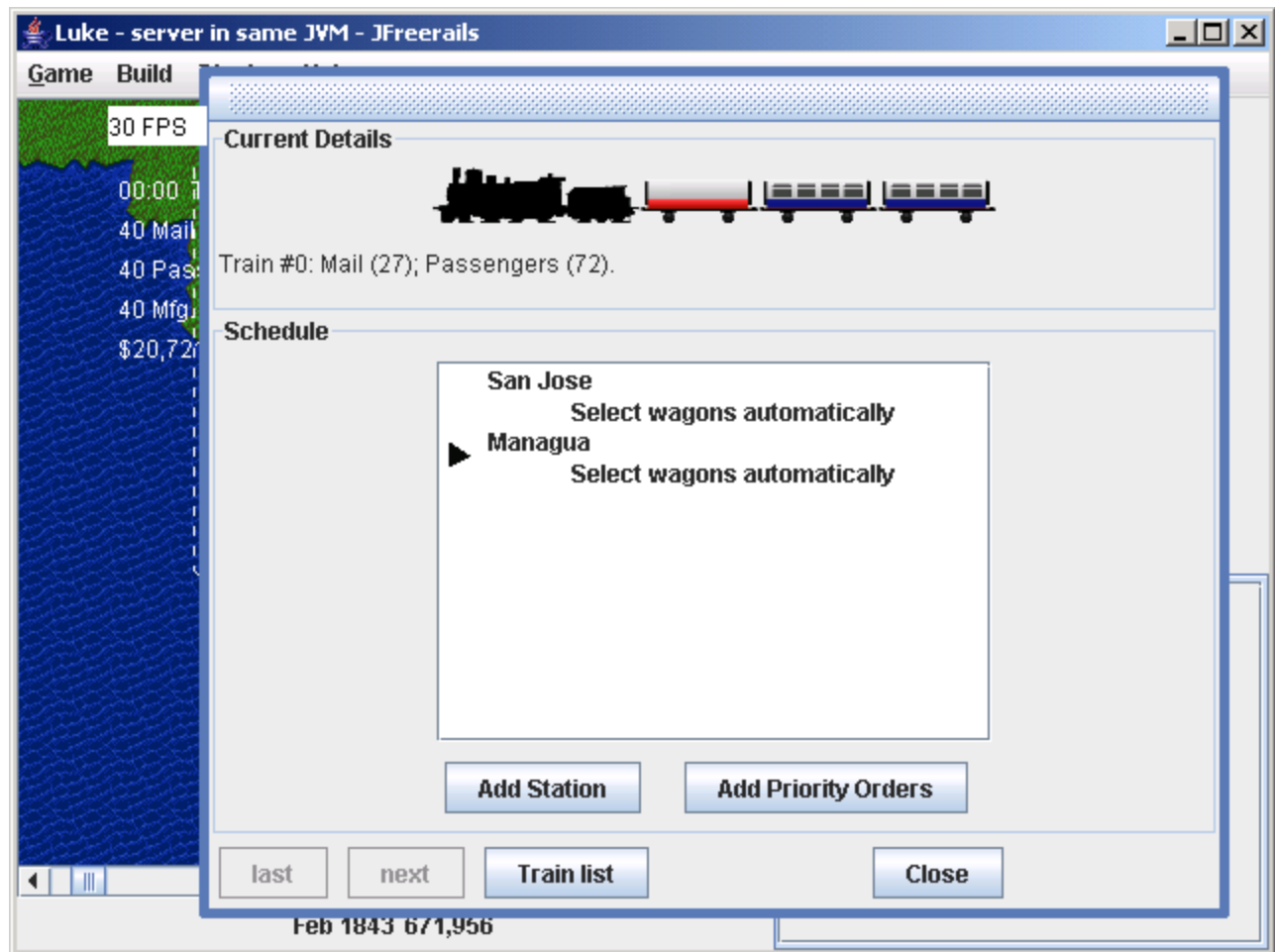


## Station List

[Shows summary details for each of the stations: name, type, cargo waiting, revenue this year.](#)

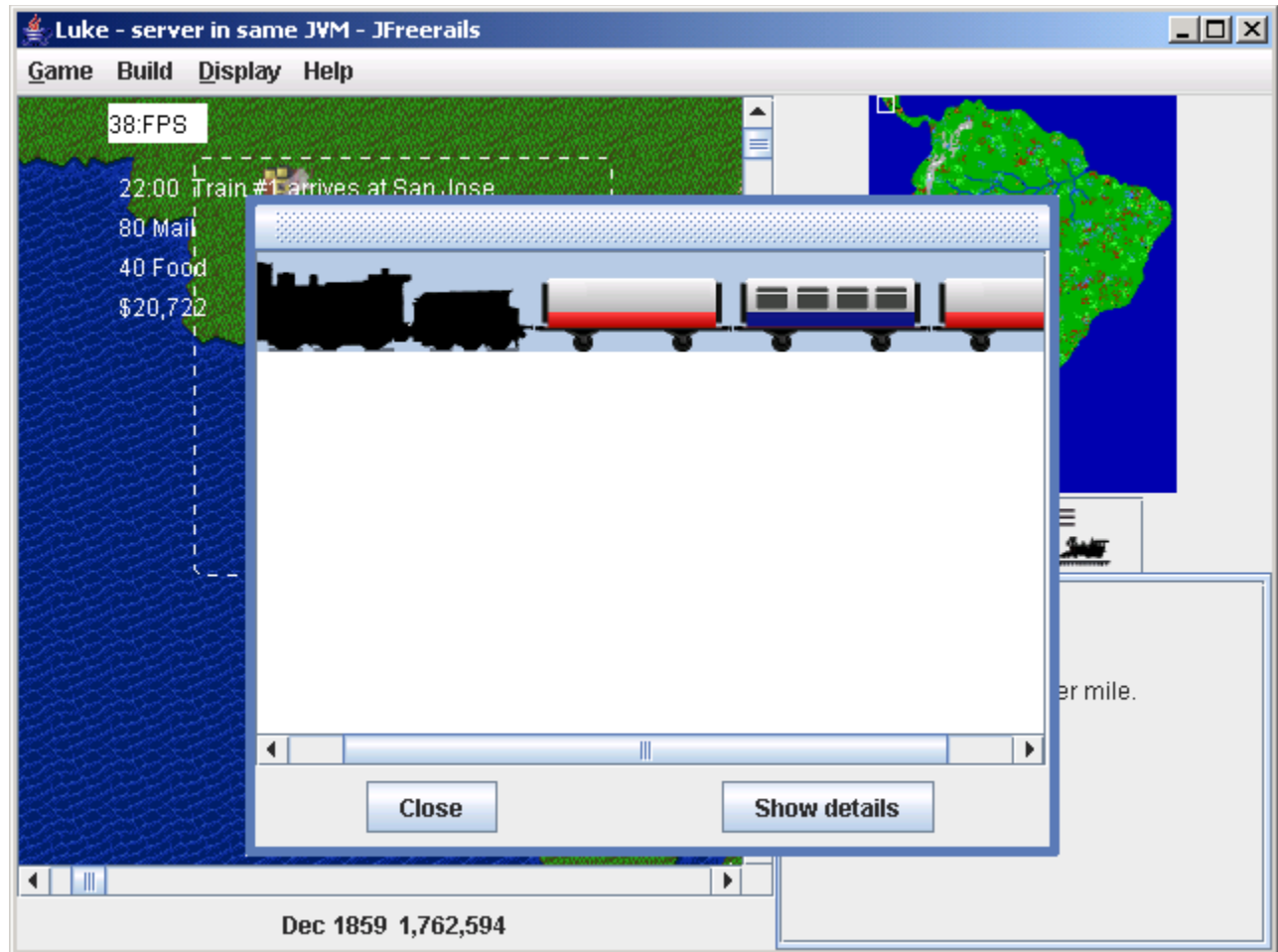
## Train report

The report shows the following information on a particular train. Its schedule, it wagons and what it is carrying.

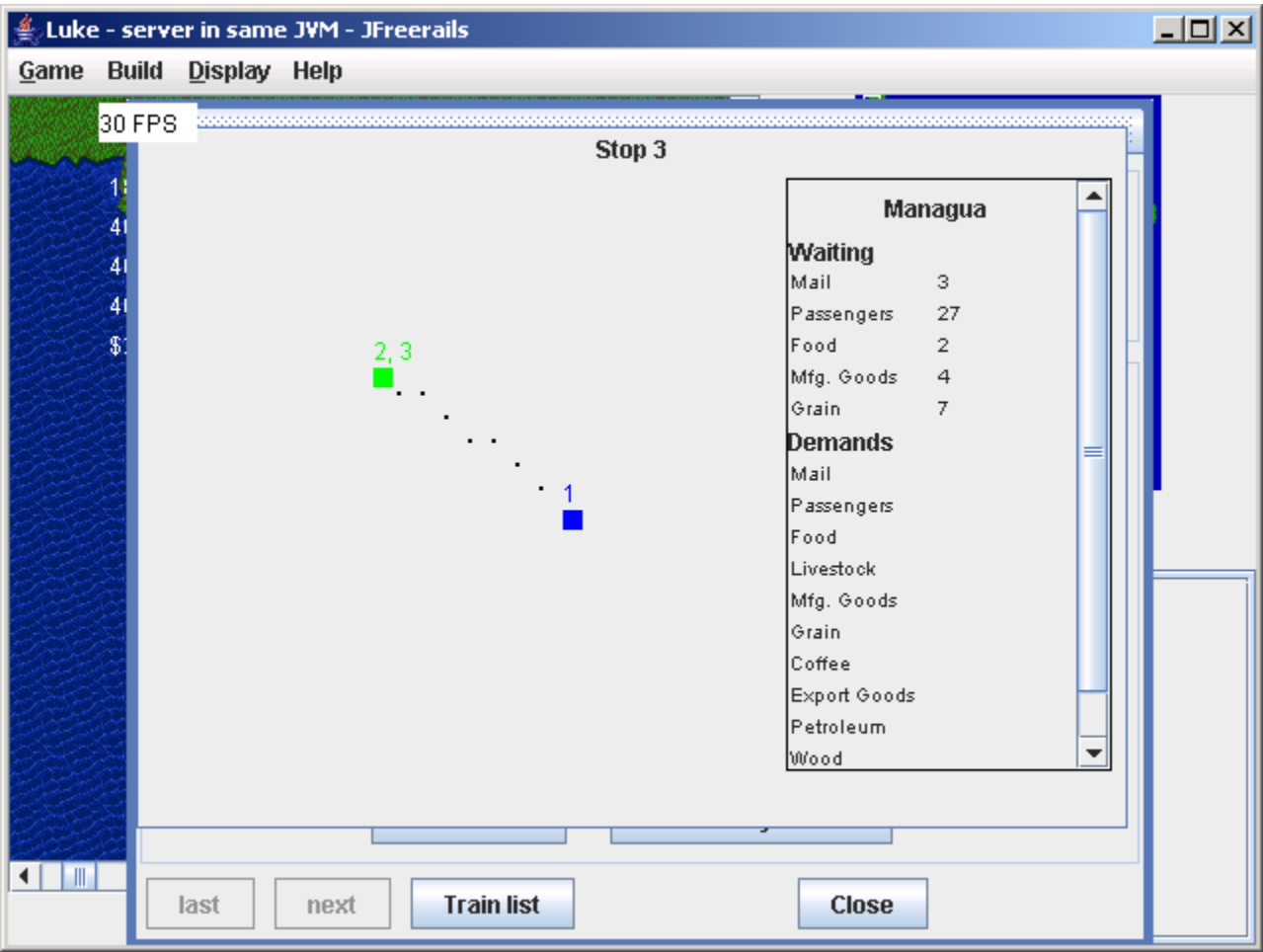


## Train List

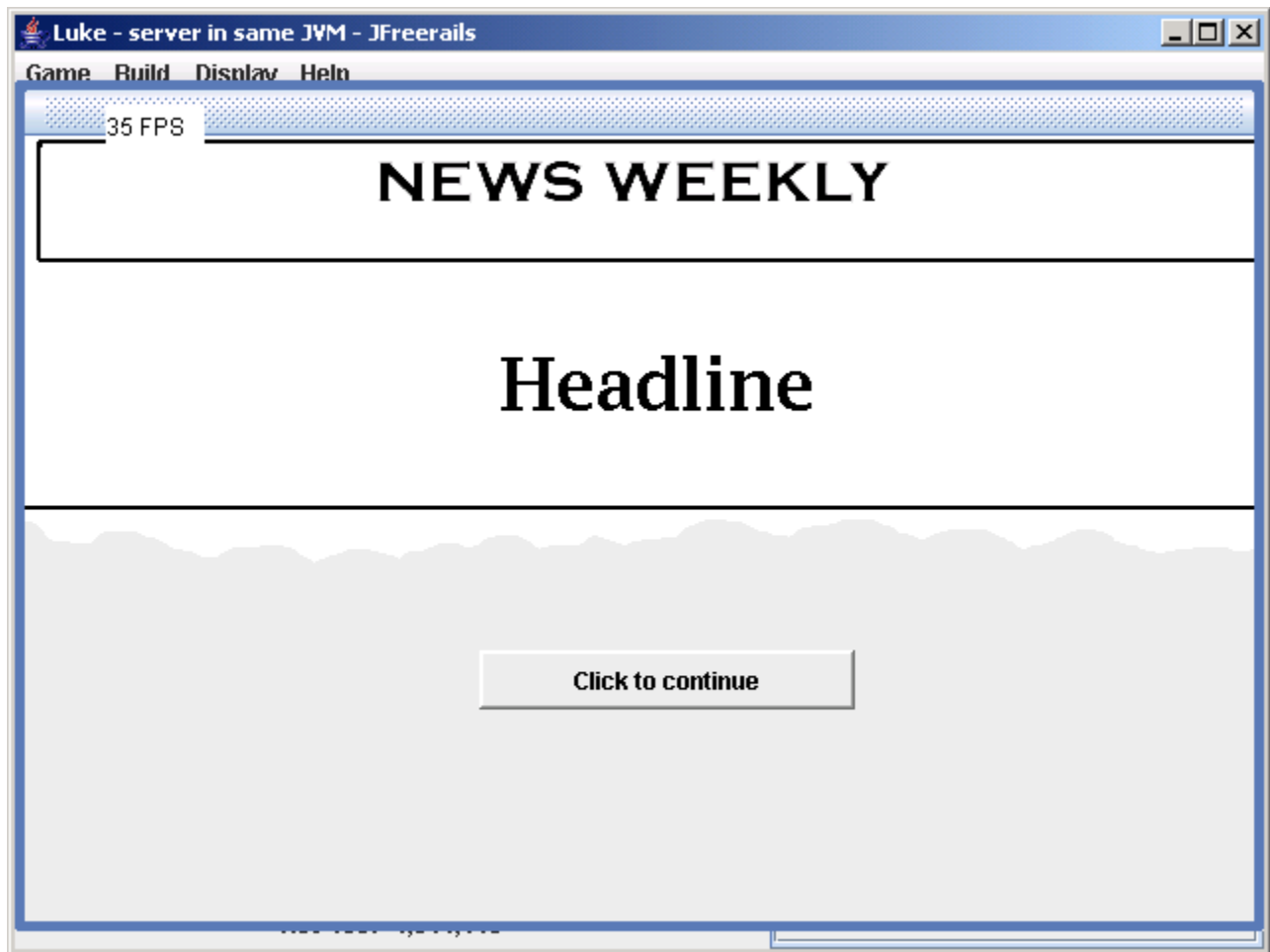
Shows summary details for each of the trains:



Select Station



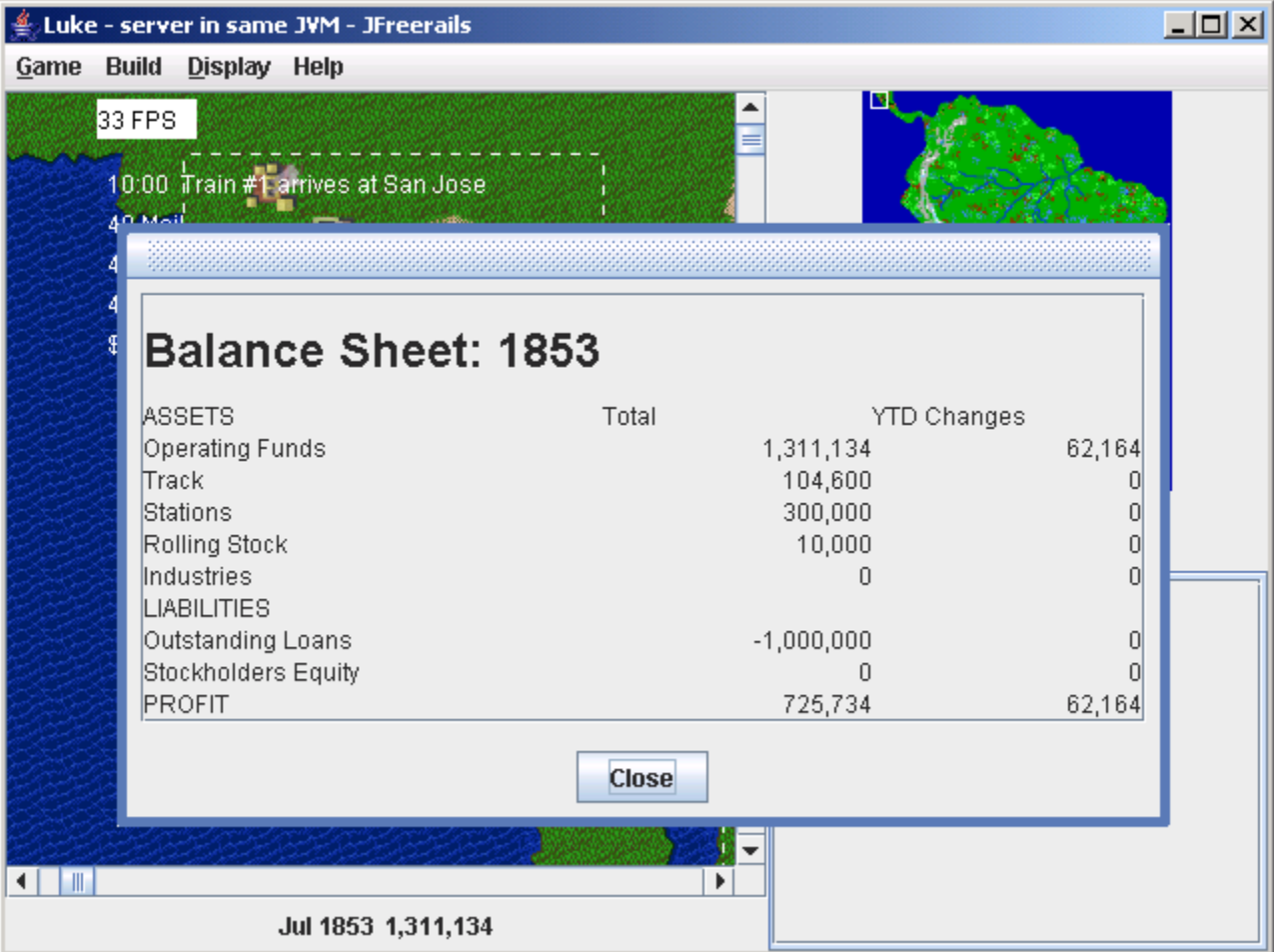
## ***Newspaper***



## ***Leaderboard***



## Balance Sheet

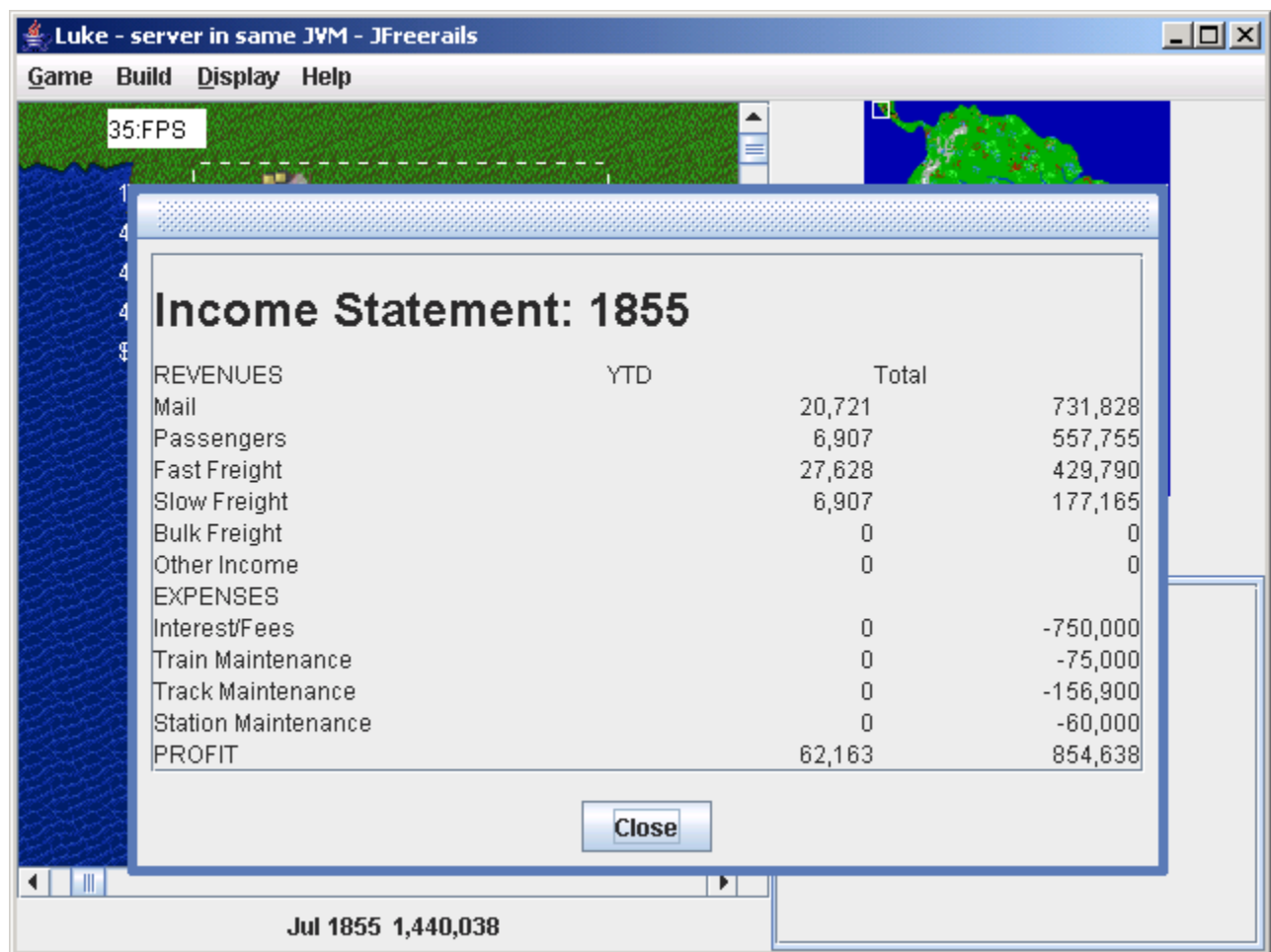


The screenshot shows the JFreerails game window titled "Luke - server in same JVM - JFreerails". The menu bar includes "Game", "Build", "Display", and "Help". The game map in the background shows a train arriving at San Jose at 10:00. A "Balance Sheet: 1853" dialog box is open, displaying the following financial data:

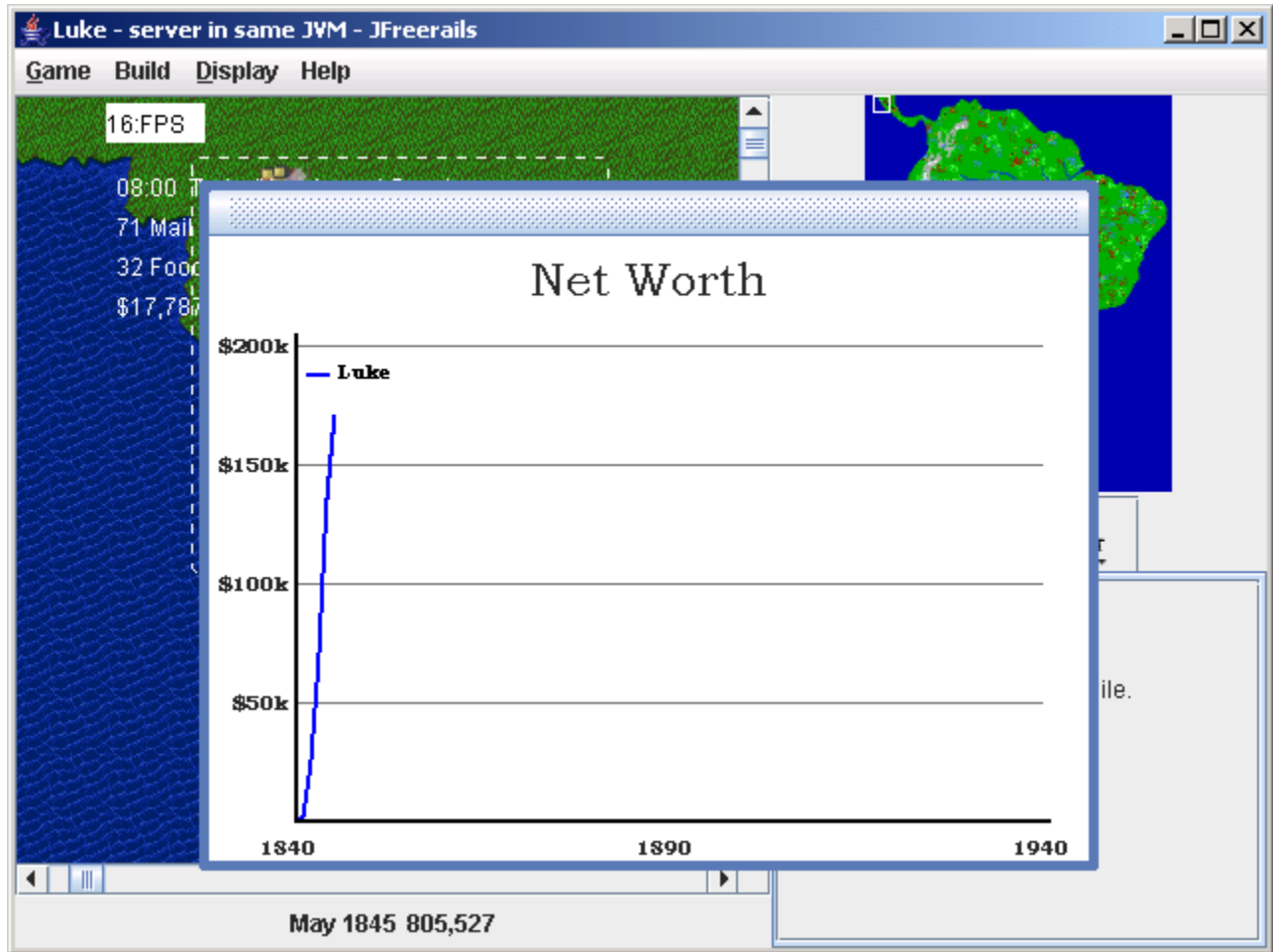
	Total	YTD Changes
<b>ASSETS</b>		
Operating Funds	1,311,134	62,164
Track	104,600	0
Stations	300,000	0
Rolling Stock	10,000	0
Industries	0	0
<b>LIABILITIES</b>		
Outstanding Loans	-1,000,000	0
Stockholders Equity	0	0
<b>PROFIT</b>	<b>725,734</b>	<b>62,164</b>

A "Close" button is located at the bottom of the dialog box. The status bar at the bottom of the game window shows "Jul 1853 1,311,134".

## Income Statement



## Networth Graph



## Report Bug Dialogue

The report bug dialoguebox is accessible from the help menu. It is also shown when there is an unexpected exception.

It should list the following information and it should be possible to copy and paste the details to the clipboard.

<i>Property</i>	<i>Value</i>
tracker.url	<a href="http://sourceforge.net/tracker/?group_id=9495&amp;atid=109495">http://sourceforge.net/tracker/?group_id=9495&amp;atid=109495</a>
java.version	Java System Property
java.vm.name	Java System Property
os.name	Java System Property
os.version	Java System Property
jfreerails.build	The timestamp generated by the ant script.
jfreerails.compiled.by	The username of the crazy person who ran the ant compile target

The how to report bug dialogueshould appear as follows...

**How to report a bug**

Use the sourceforge.net bug tracker at the following url:  
{tracker.url}

Please include:

1. Steps to reproduce the bug (attach a save game if appropriate).
2. What you expected to see.
3. What you saw instead (attach a screenshot if appropriate).
4. The details below (copy and past them into the bug report).

{os.name} {os.version}  
{java.vm.name} {java.version}  
Freerails build {jfreerails.build} compiled by {jfreerails.compiled.by}



And the “Unexpected Exception” version should read ...

### Unexpected Exception

Consider submitting a bug report using the sourceforge.net bug tracker at the following url:  
{tracker.url}

Please:

1. Use the following as the title of the bug report:  
{Exception.type} at {filename} line {line.number}
2. Include steps to reproduce the bug (attach a save game if appropriate).
3. Copy and paste the details below into the bug report:

{os.name} {os.version}






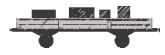






{java.vm.name} {java.version}

Freerails build {jfreerails.build} compiled by {jfreerails.compiled.by}

{stacktrace}

## Cargo Chart

The cargo chart will show the sources of supply and demand for each of the cargo types. The information will be presented in a table as below. There should be a 'print' button which should... well, its pretty obvious what it should do.

<i>Raw materials needed</i>	<i>Source of Supply</i>	<i>SupplyRate /cars per year</i>	<i>CargoType</i>	<i>Source of demand (Number needed to create demand)</i>	<i>Cargo Produced</i>
 Oil	 Refinery	-	<b>Petroleum</b>  	 Factory	 Export Goods
-	 Harbour		<b>Mfg. Goods</b>  	 City(2)	-
				 Village(4)	-

## Load Games

Displays a list of saved games. The list comprises all files ending in '.sav' in the directory from which the game was run. If the current game is a network game, the relevant directory is the directory from which the server was run. All players, not just the host, can access the dialogue.

Please select a game to load.		
<div>Game1.sav Game2.sav Game3.sav</div>		
OK	Cancel	Refresh

The 'OK' button is only enabled when a game is selected.

Pressing the 'OK' button loads the selected game.

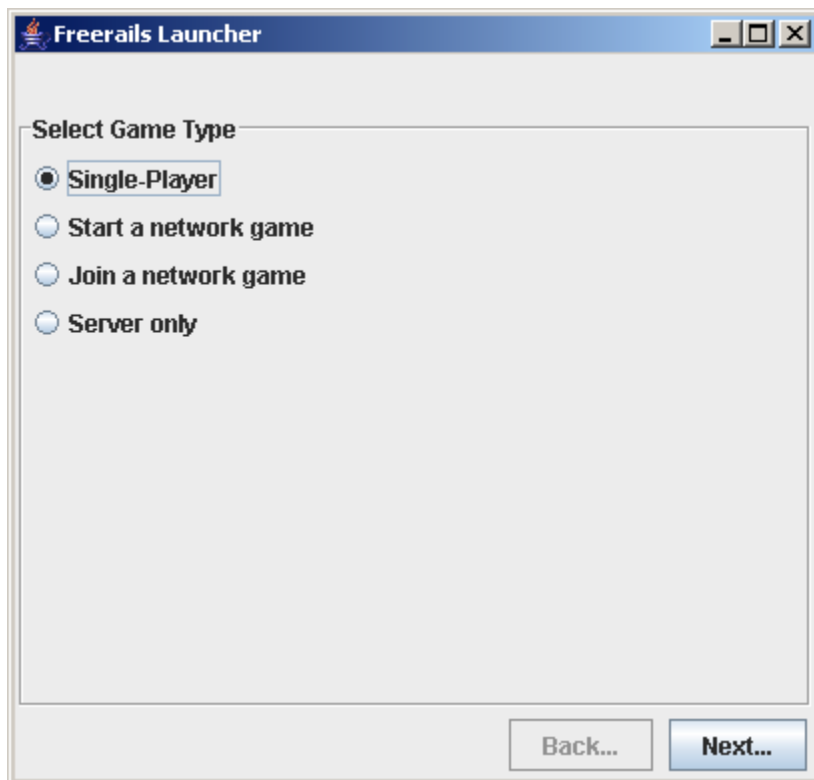
Pressing the 'Cancel' button closes the dialogue box.

Pressing the 'Refresh' button updates the list of saved games, taking into account any changes to the filesystem (e.g. any files that have been added, removed, or renamed.)

# Launcher

## ***Panel 1: Select Game Type***

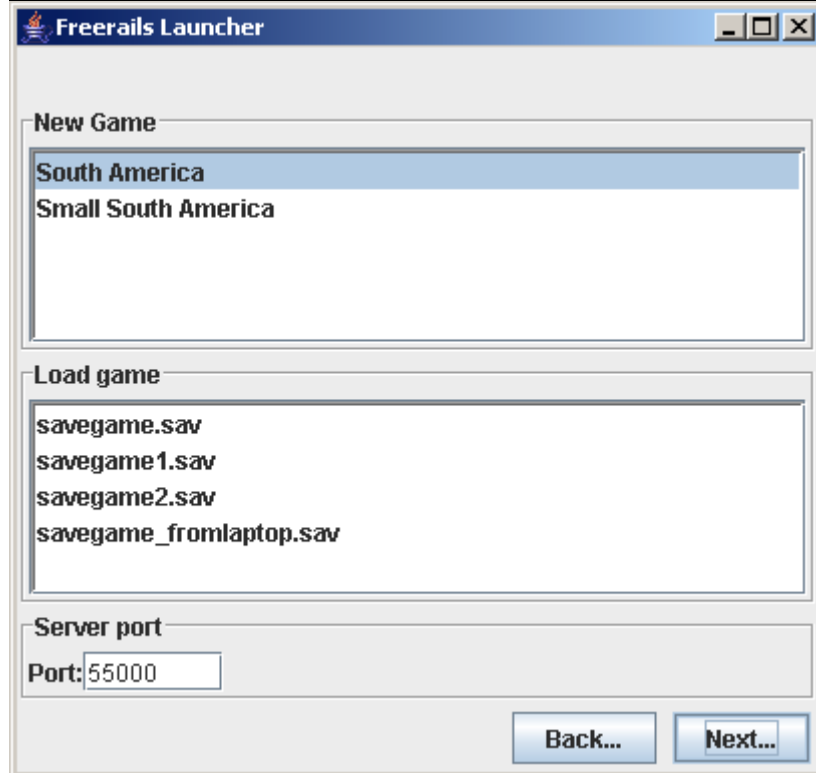
<i>Selection</i>	<i>Next Screen</i>
Single Player	Select Map (without server port input box)
Start a network game	Select Map (with server port input box)
Join a network game	Client details (with remote server details showing)
Server only	Select Map (with server port input box)



## Panel 2: Select Map (and server details)

The value of the field “Server port” should be the value entered last time the launcher was run. On the first run it defaults to 55000.

<i>Selection</i>	<i>Next Screen</i>
Single Player	Client details (without remote server details showing)
Start a network game	Client details (without remote server details showing)
Server only	Connected players



<i>Condition</i>	<i>Message or result</i>	<i>When checked</i>
No saved game available.	The item "Load a saved game" should be disabled	When the panel is created.
Port field does not contain a number between, inclusive 0 and 65535	"A valid port value is between between 0 and 65535." and disable "next" button.	As text is entered.
" Start a new map" is selected but no map is selected."	"Select a map". The "next" button should be disabled.	When the radio button selection changes and when the selected map in the map list changes.

<i>Condition</i>	<i>Message or result</i>	<i>When checked</i>
Can't start server on specified port	Use the message from the exception. The next button should still be enabled.	When the next button is pressed.

### ***Panel 3: Client details (and remote server details)***

The following fields should be recalled from the last time the launcher was run.

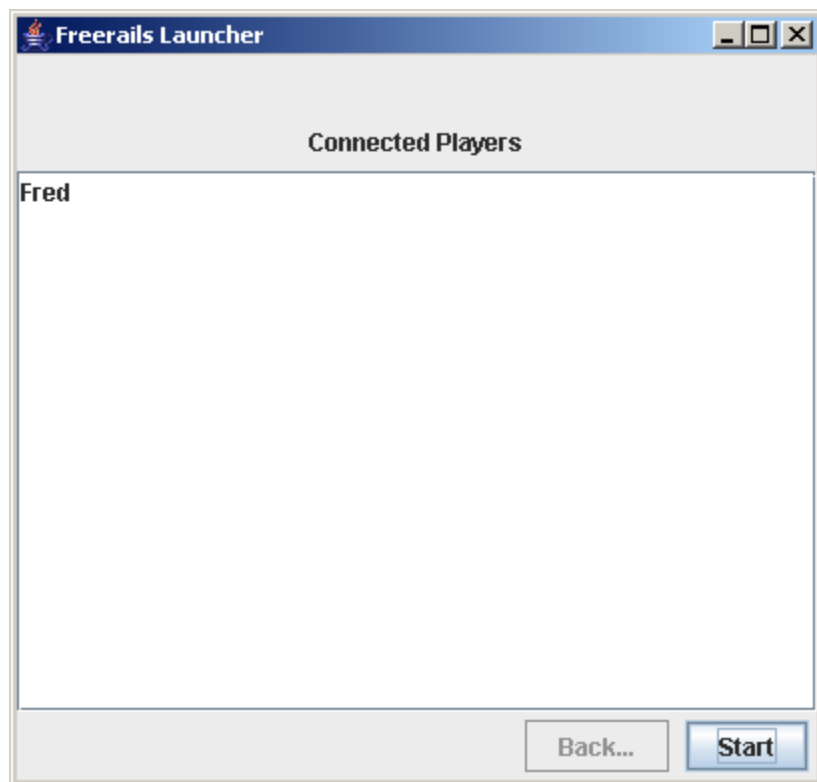
<i>Field</i>	<i>Default</i>	<i>Notes</i>
Player name	The value of system property "user.name"	If a game is being loaded, the text box should not be appear. Instead there should be a dropdown list with the names of the players from the saved game.
IP Address	127.0.0.1	
port	5500	

<b>Selection</b>	<b>Next Screen</b>
Single Player	Progress bar
Start a network game	Connected players
Join a network game	Progress bar

The screenshot shows the 'Freerails Launcher' window. It has three main sections: 'Player Details' with a 'Player name' text box containing 'Fred'; 'Remote server address' with 'IP Address' (169.254.182.62) and 'port' (55000) text boxes; and 'Select Display Mode' with three radio buttons: 'Windowed' (selected), 'Windowed (fixed size 640\*480)', and 'Full screen'. Below the radio buttons is a list box showing '640x480 16 bit 60Hz', '640x480 32 bit 60Hz', and '640x480 16 bit 70Hz'. At the bottom are 'Back...' and 'Start' buttons.

<b><i>Condition</i></b>	<b><i>Message or result</i></b>	<b><i>When checked</i></b>
The " Player name" field is empty.	" Enter a player name" and disable " next" button.	As text is entered.
Port field does not contain a number between, inclusive 0 and 65535	" A valid port value is between between 0 and 65535." and disable " next" button.	As text is entered.
" Full screen" is selected but no map is selected."	" Select a display mode" . The next button should be disabled.	When the radio button selection changes and when the selected display-mode in the display-mode list changes.
The " IP address" field is empty.	" Enter the address of the server" and disable " next" button.	As text is entered.
Can't resolve host.	"Can't resolve host."	When next button is pressed.
Can't connect to server.	" Can't connect to server."	When next button is pressed.
Load game was selected.	The player name textbox should be replaced with a dropdown list of players in the saved game.	When the form is displayed.
We are connecting to a remote server which has loaded, but not started a game, and the player name we entered is not a player in the saved game.	"New players can't join a saved game."	When next button is pressed.
We are connecting to a remote server but the game has already started.	"New players can't join a game in progress."	When the next button is pressed.

## ***Panel 4: Connected players***





## Panel 5: Progress Bar



# AI

Disclaimer - the notes below are very incomplete. It might make sense to do something simpler for the first version of the AI.

Deciding which cities should be connected to each other  
Create a table of the distances between cities. E.g.

	City A	City B	City C	City D
City A	x	20km	30km	25km
City B	x	x	35km	15km
City C	x	x	x	40km
City D	x	x	x	x

```
For every pair of cities i, j {  
    For every city k where i != k and j != k {  
        Let A = the distance between i and j.  
        Let B = the distance between i and k.  
        Let C = the distance between k and j.  
        If (A < B and A < C) then remove the value at i, j from the table.  
    }  
}
```

We can now construct a graph from the values remaining in the table. It will have the following properties. First, every city is connected to its nearest neighbour. Second, we can get from any city to any other city. Third, not too much track will be wasted.

Deciding the order in which to connect cities

Lets assume the profitability of a line between 2 cities, A and B is given by the following condition. Profitability = (Cargo supplied by A and demanded by B + Cargo supplied by B and demanded by A) / Distance between A and B.

Implementation Note: a natural way to analyze supply and demand and cargo conversions would be using matrix algebra. E.g. supply and demand at a station could be represented by  $n \times 1$  matrices and cargo converted by an  $n \times n$  matrix (where  $n$  is the number of cargo types). There is a public domain java matrix package available at: <http://math.nist.gov/javanumerics/jama/>.

The simplest strategy for building track would be starting with most profitable connection. Note, that on the first move, we can pick any connection, but on subsequent moves, we are restricted to connections involving at least one city we have already connected to. Call this restricted set of connections S. A reasonable strategy for subsequent moves would be repeatedly picking the most profitable connection from S.

A more sophisticated strategy would take into account the restriction that new track must connect to

existing track when picking the first pair of cities to connect. We could approach the problem as follows. Assume we build one connection per year and the game continues until we have built all possible connections. Suppose our payoff for building a connection is the profitability of the connection times the number of years remaining. For simplicity, assume that once we have built the first connection, we revert to just picking the most profitable connection from  $S$  as before. Now, we can solve the problem of which connection to start with by comparing the payoff over the complete game for each of the possible starts.

Obviously, to formally solve the problem above, we would need to consider strategies other than picking the most profitable connection from  $S$  for moves after the first one. However, unless the number of cities is relatively small this would likely take a long time to solve. What is more, we have not even considered what other players may be doing, so even if we could formally solve the problem above, we would still have a lot of work to do.