
FreeRails Documentation

Release 0.4.0

The FreeRails team

Nov 14, 2022

Contents

1	Introduction	3
2	Player Manual	5
2.1	How to play	5
2.2	Game controls	5
3	Specification	7
3.1	Sid Meier's Railroad Tycoon (1990)	7
3.2	Railroad Tycoon II (1998)	10
3.3	Introduction	11
3.4	User interface	11
3.5	Game Model	11
4	Development	33
4.1	Brief overview of the architecture	33
4.2	Server - client communication	33
4.3	Track implementation	35
4.4	Computer controlled (AI) players	36
4.5	Coding Guidelines	36
5	History	37
5.1	FreeRails (2000-2005)	37
5.2	Railz (2004-2005)	38
5.3	FreeRails2 (2007-2008)	38
5.4	Railz2 (2012-2014)	38
5.5	FreeRails3 (2016)	38
5.6	FreeRails continuation (2017-)	38
6	Changelog	39
6.1	FreeRails continuation (2017-)	39
6.2	FreeRails3 (2016)	39
6.3	FreeRails (2000-2005)	39

FreeRails is a real-time, multi player railway strategy/management game where players compete to build the most powerful railroad empire. It is based on the RailRoad Tycoon I and II games.

CHAPTER 1

Introduction

2.1 How to play

- (1) Look at what different tiles supply and demand by moving the cursor around and pressing 'I'.
- (2) Pick two cities that each have at least 2 city or 4 village tiles. Ideally, the cities should be 10-20 tiles apart.
- (3) Build track between the cities (by using the number pad keys with num lock on or by dragging the mouse).
- (4) Build stations at each of the cities (press F8). Make sure the station radius surrounds most of the city tiles.
- (5) Build a train (press F7). Don't add any wagons (they will be selected automatically).

2.2 Game controls

2.2.1 On the map view

- Build Track: Mouse or number pad with Num Lock down
- Move Cursor: Hold [control] and use Arrow Keys
- Call Broker: 'M'
- Save Game: [control] + 'S'
- Load Game: [control] + 'L'
- Build Station: 'F8' (on existing track)
- Build Train: 'F7' (after building a station)
- Terrain Info: 'I'
- Build industry: 'B'

2.2.2 On the train orders screen

- Goto Station: 'G'
- Change station: 'S'
- Automatic schedule: 'A'
- Remove station: 'Delete'
- Toggle wait until full: 'W'
- Remove last wagon: 'Back space'
- Add station: 'N'
- Set priority orders: 'O'

2.2.3 On the select station screen

- Change selection: Mouse over or number pad
- Accept selection: Click mouse or 'Enter'

3.1 Sid Meier's Railroad Tycoon (1990)

This is a summary of the game model of the original Railroad Tycoon (RRT).

3.1.1 Start condition

- \$1 million (half in equity, half as loan)
- Lay track
- Build stations
- Buy trains
- Schedule trains
- Sell, buy bonds
- Build additional industries
- Buy or sell shares

3.1.2 Stations

Max. 32 stations per player. First stations gets an engine shop for free. Stations can be up- or downgraded. During an accounting period built or re-built stations pay double for freight.

- Signal Tower
- Depot (3x3 squares influence area)
- Station (4x4 squares influence area)
- Terminal (5x5 squares influence area)

3.1.3 Station improvements

- Engine shop
- Store
- Hotel
- Switching yard
- maintenance shop
- cold storage
- livestock pens
- goods storage
- post office
- restaurant

3.1.4 Trains

Max. 32 trains per player.

- Change “consist” (number and type of cars/wagons)
- Types include: mail, passenger, freight

Engines

- 0-4-0 Grasshopper
- 4-2-0 Norris
- 4-4-0 American
- 2-6-0 Mogul
- 4-6-0 Ten-Wheeler
- 2-8-0 Consolidation
- 4-6-2 Pacific
- 2-8-2 Mikado
- 2-6-6-4 Mallet (Challenger class)
- F3-A Diesel-Electric
- EMD GP Diesel-Electric
- 2-2-0 Planet
- 2-2-2 Patentee
- 4-2-0 Iron Duke
- 0-6-0 DX Goods
- 4-2-2 Stirling
- 0-8-0 Webb Compund

- 4-2-2 Johnson Midland Spinner
- 4-4-0 Claud Hamilton
- 4-6-2 A1 Class
- 4-6-2 A4 Class
- 6/6 GE Class Crocodile
- 1-Do-1 Class E18
- 4-8-4 242 A1
- V200 B-B
- Bo-Bo-Bo RE Class 6/6
- TGV

Cars

- mail
- passenger
- beer
- livestock
- goods
- hops
- textiles
- steel
- chemicals
- cotton
- coal

3.1.5 Supply and demand

3.1.6 Stock market

3.1.7 Time

- 100 years running time (starting year depends on the scenario)
- Accounting period two years long

3.1.8 Difficulty levels

The chosen difficulty level remains in effect for the whole duration of a game.

Levels are “Investor”, “Financier”, “Mogul”, “Tycoon”. The level of difficulty affects revenue earned by each delivery as well as the tycoon rating at the end.

3.1.9 Reality levels

- “No Collision Operation/Dispatcher Operation” In the dispatcher operation, the movement of trains is controlled by block signals and collisions are possible.
- “Friendly competition/Cut-Throat competition” In friendly competition, they do not buy your stock, attempt to take you over or start rate wars.
- “Basic economy/Complex economy”

3.1.10 Economy

- Simple: Stations with two or more cities will buy everything
- Complex: Specialized rules

3.1.11 Other features

- If the share price of a competing railroad falls below \$5 and stays there for too long, it can be dissolved and be removed from the game
- For each bankruptcy that the player declares, the interest for selling new bonds is increased by 1%. After a certain number of bankruptcies, the player is unable to sell any bonds.
- Each car that is placed on a train costs \$5,000. When the consist changes, one is only charged if the total number of cars increases.
- There is a Find City option

3.1.12 Supplied scenarios

- Western United States
- Northeast United States
- Great Britain
- Continental Europe

3.1.13 Links

- https://en.wikipedia.org/wiki/Railroad_Tycoon

3.2 Railroad Tycoon II (1998)

This is a summary of the game model of the Railroad Tycoon 2 (RRT2).

This specification describes the various elements of the game and their interactions. It may not discuss in detail how features will be implemented.

3.3 Introduction

Let's imagine a few real life stories of how actual (stereotypical) people would use it.

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 his 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 - it's a game not a simulation.

3.4 User interface

All dialogs and screens should have a help button showing localized help. The help is just a locally stored html page.

3.4.1 Start screen

Must enable the following actions

- change preferences (modal dialog)
- start the scenario editor (full screen)
- exit the game
- show help information (modal dialog)
- start playing a game (modal dialog)

3.4.2 Start playing dialog

Must enable the following actions

- start a new local game
- continue a stored local game
- connect to an external server and participate in a network game hosted on that server
- host network game (and participate in it)
- close the dialog

3.5 Game Model

3.5.1 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) 100 mile trips in a year.

3.5.2 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.

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

3.5.3 Terrain

Terrain Type

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

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 live-stock.

Cargo consumption Some types of terrain consume cargo, e.g. 'City' tiles consume 'Food'.

Right-of-way cost Before you build track on a square, you need to purchase the right of way. Different terrain types have different ROW costs.

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.

3.5.4 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.

3.5.5 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.

3.5.6 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 sufficient cash. 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 minus the cost of X. Track can be removed. When you remove track you get a little amount of money back.

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

Bridges

Name | Price | Train Speed | Track Type

- Wooden Trestle | \$50,000 | slow | single
- Iron Girder | \$200,000 | fast | single
- Stone Masonry | \$400,000 | fast | double

Track Contention

On single (double) track, only one (two) train(s) can move at a time. Gridlock shouldn't occur because when trains stop moving they do not block other trains.

3.5.7 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){
          unload cargo that won't fit after changing consist
          change consist
        }
      }
      load cargo
      setStatus(stoppedAtStation)
      departTime = currentTime + stopTime
    }else{
      setStatus(readyToMove)
    }
  }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(readyToMove)
      }
    }
  }
}
```

(continues on next page)

(continued from previous page)

```
    }
  }else{
    if(currentTime > departTime){
      if(this was a scheduled stop){
        update next station on schedule
      }
      setStatus(readyToMove)
    }
  }
}
if(train status is readyToMove){
  find next track section
  if(number of trains on next track section < number of tracks){
    setStatus(moving)
    move train onto next track section
  }
}
```

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.

Path finding is automatic. That means that trains take the slowest perceived route and priority trains having priority. (This may change at some point with the player being able to choose between several possible routes between two stations. It would be implemented using way points, maybe even changing routes depending on the traffic.)

3.5.8 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

Improvement Type | Effect | Price

- Engine Shop | Trains can be built | \$100,000
- Switching Yard | Cuts time taken the change wagons by 75% | \$50,000
- Maintenance Shop | Cuts yearly maintenance of trains that stop at station by 75% | \$25,000
- Cargo Storage | Prevents a certain cargo from wasting away | variable
- Revenue booster | Increases revenue from a cargo X by Y% | variable

3.5.9 Economy

The economy alternates between 5 states, with an order and states can only change to adjacent states.

Economic Climate | Base Interest rate | Effect on track price

- Boom | 2% | +33%
- Prosperity | 3% | +17%
- Moderation | 4% | -
- Recession | 5% | -17%
- Panic | 6% | -33%

Stocks 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 = $[\text{Net worth at start of year} + 5 * \text{profit last year}] / [\text{shares owned by public} + 0.5 \text{ shares owned by other players}]$

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.

3.5.10 Non player effects on 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.

Category | Marginal utility | Motivation

- Industry | $\text{Number of Urban Tiles} / (1 + \text{Number of Industry Tiles})^2$ | Urban tiles supply labour to industries. There are decreasing returns to scale.
- Resource | $(\text{Units of Resource Picked Up} + c) / (1 + \text{Number of Resource Tiles})$ | Resources grow when they are exploited. There are decreasing returns to scale.
- Urban | $\text{Units of demanded cargo delivered} - k * \text{Number of Urban Tiles} / (1 + \text{Number of Industry Tiles})$ | Urban tiles value employment and delivery of cargo but are adverse to overcrowding.

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 at stations

3.5.11 User Interface

Main Window

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

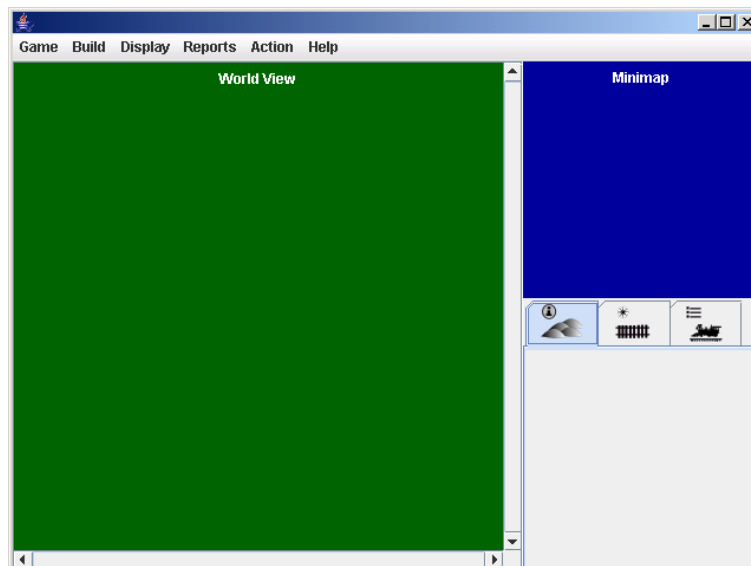


Fig. 1: Scheme of main window.

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.

Component | Width | Height

- 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.

Menu bar

Game - New Game | Game Speed (Paused, Slow, Moderate, Fast) | Save Game | Load Game | Exit Game

Build - New train | Industry | Improve Station

Display - Regional Display | Area | Detailed Area | Options

Reports - Balance Sheet | Income Statement | Networth Graph | Stocks | Leaderboard | Accomplishments

Broker - Call Broker

Help - Controls | Quick Start | Manual | Report Bug | About

World View Window

The world can be displayed at 4 zoom levels:

- Local detailed: 30x30 px
- Local: 15x15 px
- Network: Scaled so that all the player's stations are visible, Shows trains, stations, and track but not geography
- Regional: Scaled so the whole map is visible, Minimap hidden

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

3.5.12 Components on right hand side

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).

Table 1: Build modes

Build mode	Options visible when mode is selected	Action
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.
up-grade 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.

3.5.13 Reports and dialog boxes

Broker Dialog

Station report

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

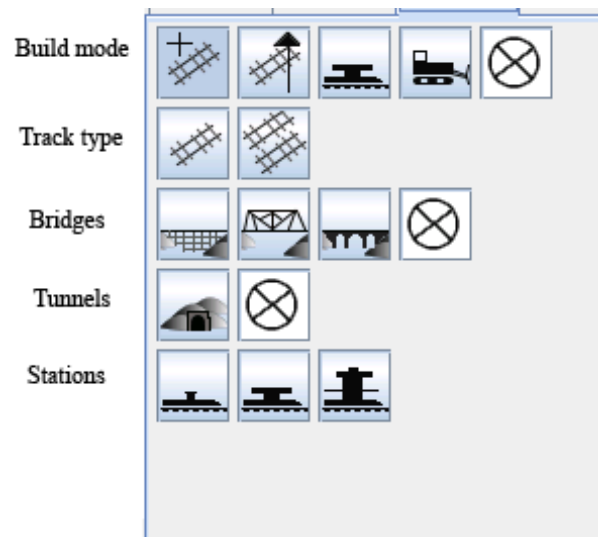


Fig. 2: Scheme of build tab.

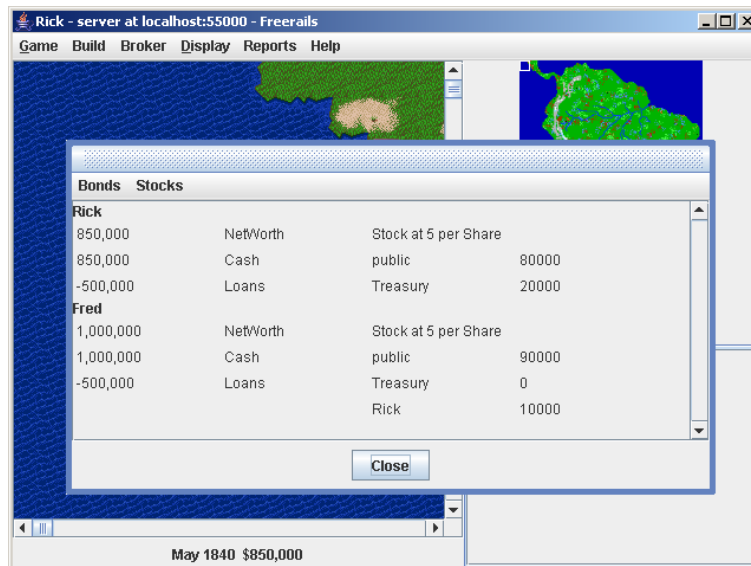


Fig. 3: Example of the broker dialog.

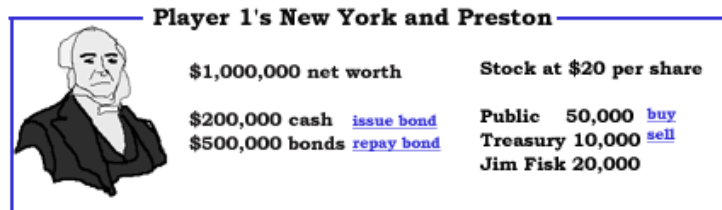


Fig. 4: Another example of the broker dialog.







	<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>		
...			

Fig. 5: Supply and Demand Tab

The trains tab will list all trains that are scheduled 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 scheduled stop.

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

Fig. 6: Trains Tab

Improvements tab 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

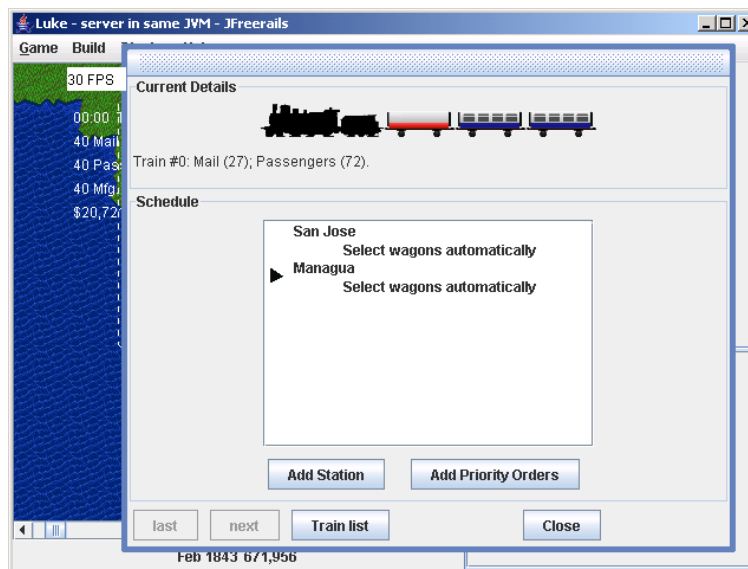


Fig. 7: Trains report

Train list

Shows summary details for each of the trains

Select station

Newspaper

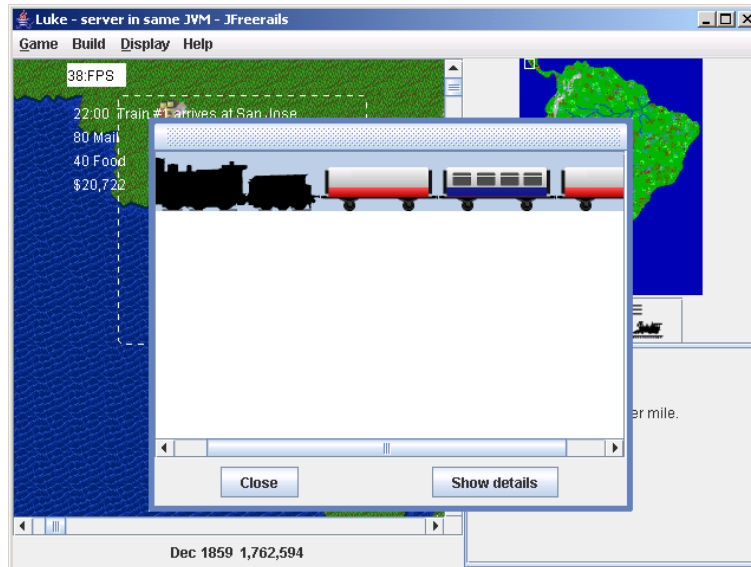


Fig. 8: Trains list

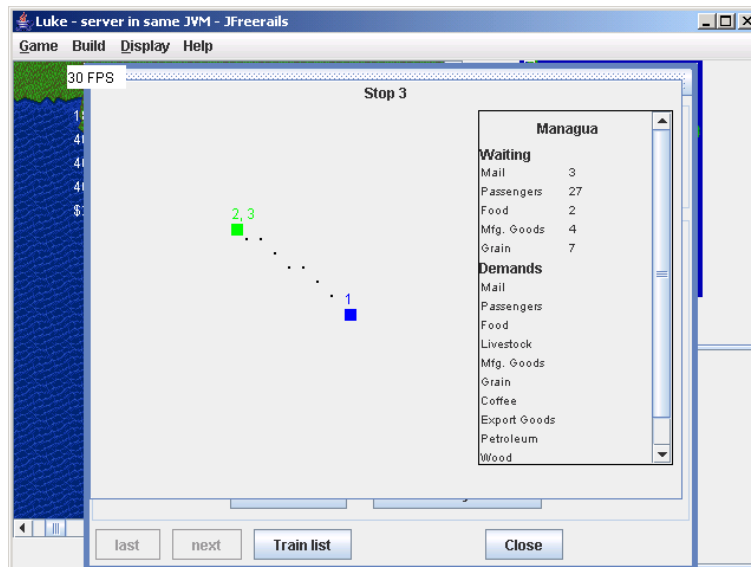


Fig. 9: Select station



Fig. 10: Newspaper

Leaderboard



Fig. 11: Leaderboard

Balance sheet

Income statement

Networth graph

Report bug dialog

The report bug dialog box 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. Property | Value

- tracker.url http://sourceforge.net/tracker/?group_id=9495&atid=109495
- java.version Java System Property
- java.vm.name Java System Property
- os.name Java System Property
- os.version Java System Property

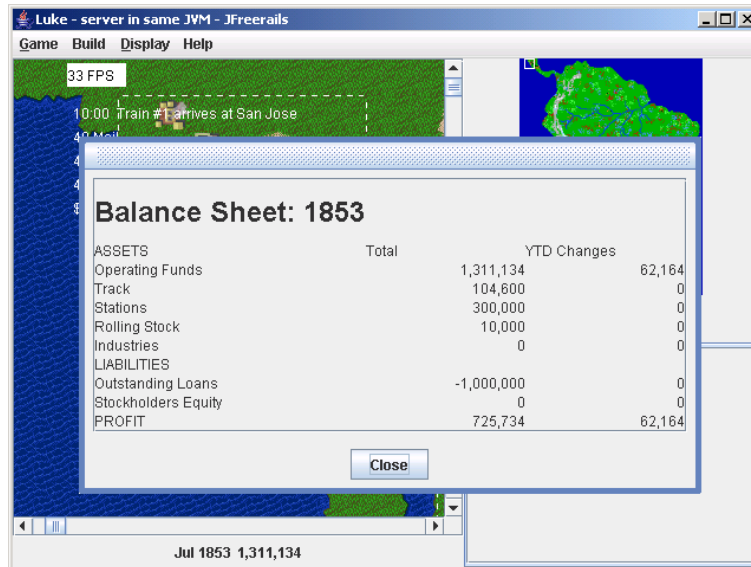


Fig. 12: Balance sheet

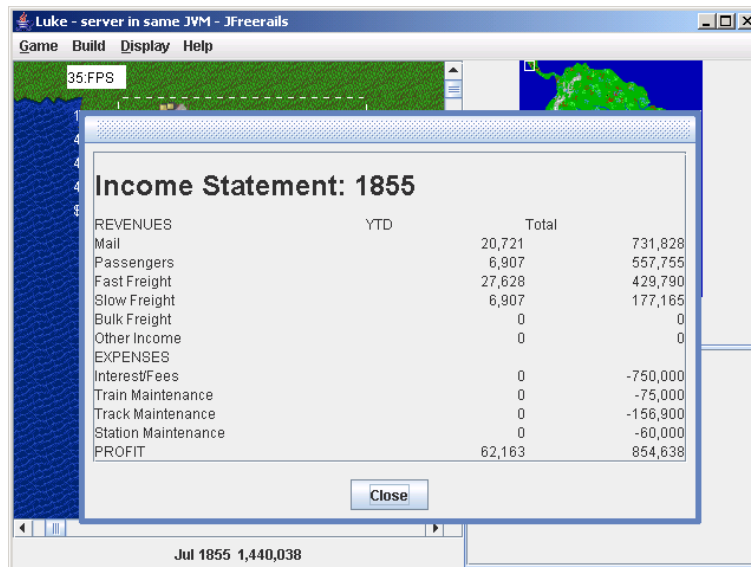


Fig. 13: Income statement

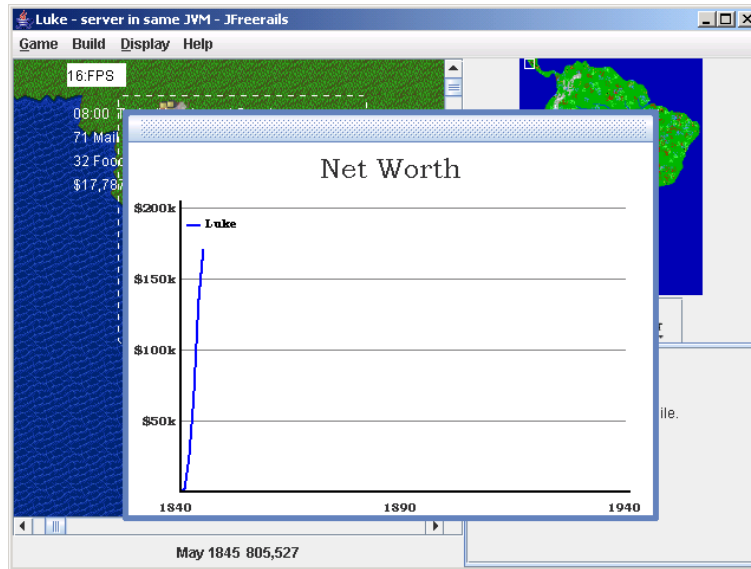


Fig. 14: Networth graph

- `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 dialog should 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** paste 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}
```

(continues on next page)

(continued from previous page)

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

Fig. 15: Cargo chart

## 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.       |  |  |
| Game1.sav<br>Game2.sav<br>Game3.sav |  |  |
|                                     |  |  |
|                                     |  |  |

Fig. 16: Load game

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

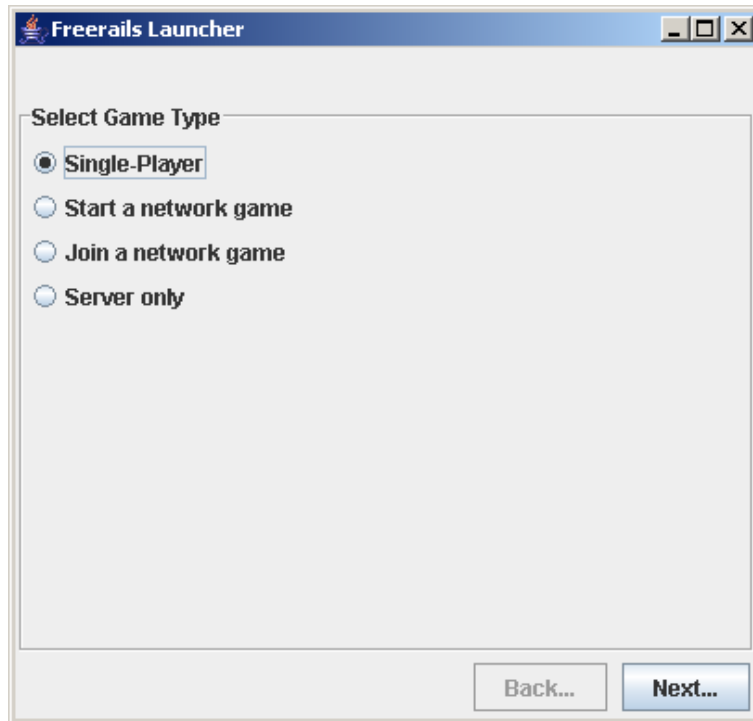


Fig. 17: Launcher1

### Selection | Next Screen

- 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.

### Selection | Next Screen

- Single Player Client details (without remote server details showing)
- Start a network game Client details (without remote server details showing)
- Server only Connected players

### Condition | Message or result | When checked

- 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 0 and 65535.” and disable “next” button. | As text is entered.

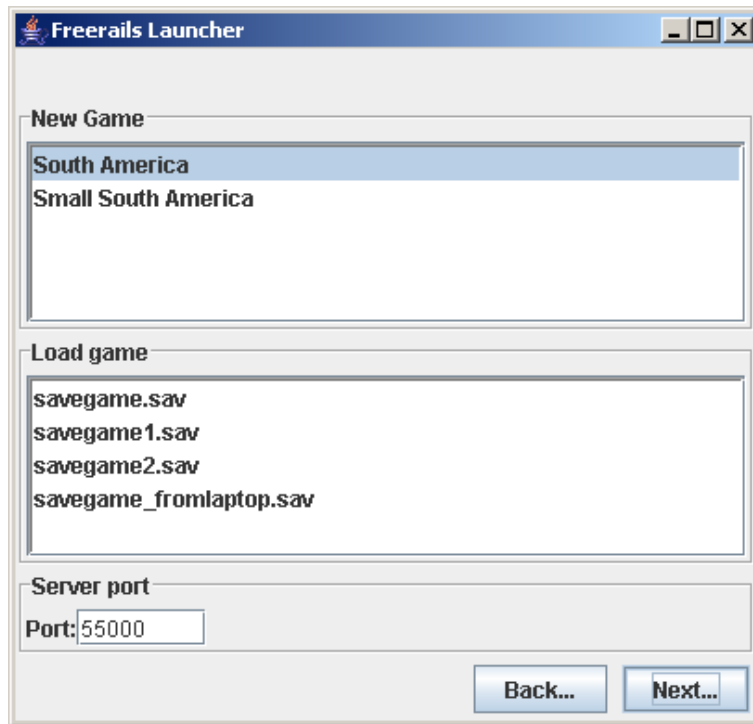


Fig. 18: Launcher2

- ” 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.
- 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. Field | Default | Notes

- 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 |

Selection | Next Screen

- Single Player | Progress bar
- Start a network game | Connected players
- Join a network game | Progress bar

Condition | Message or result | When checked

- 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.



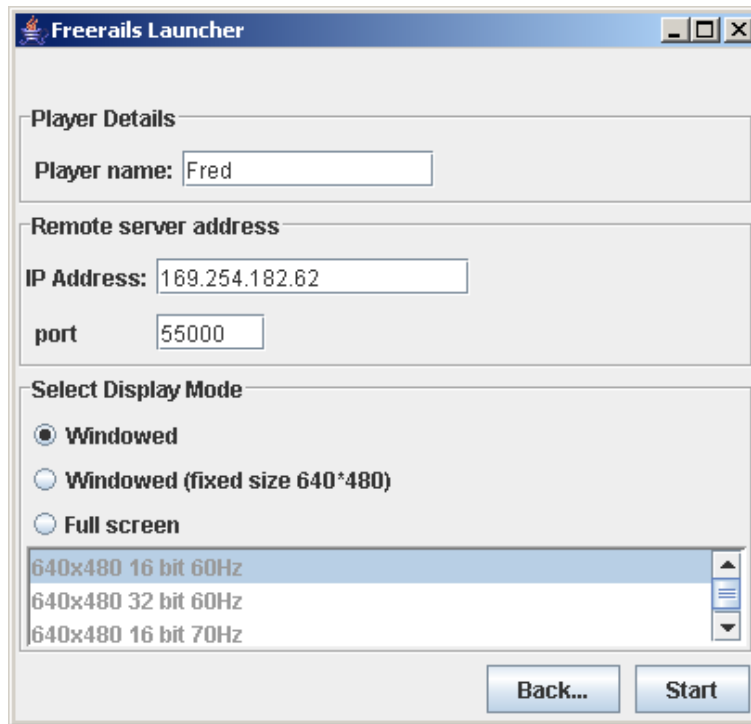


Fig. 19: Launcher3

- 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

### 3.5.14 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.

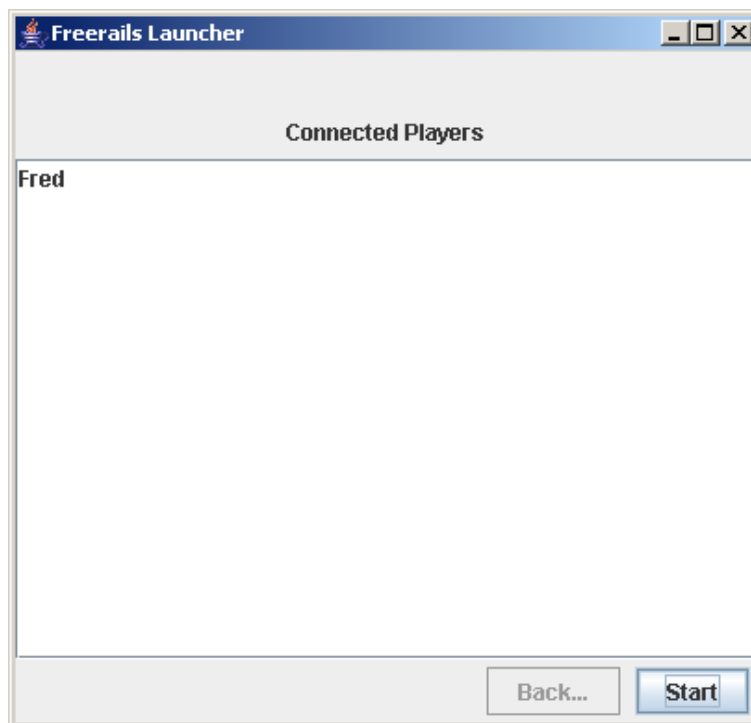


Fig. 20: Launcher4



Fig. 21: Launcher5

Table 2: City distances example

|        | City A | City B | City C | City D |
|--------|--------|--------|--------|--------|
| City A | x      | 20 km  | 30 km  | 25 km  |
| City B | x      | x      | 35 km  | 15 km  |
| City C | x      | x      | x      | 40 km  |
| 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  $n * 1$  matrices and cargo converted by an  $n * n$  matrix (where  $n$  is the number of cargo types). There is a public domain java matrix package available at: <http://math.nist.gov/janumerics/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.



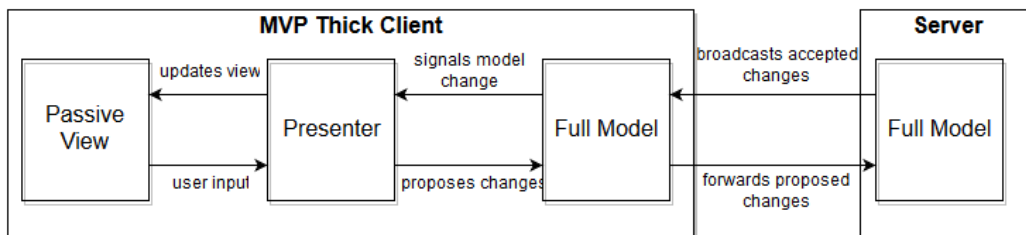


Fig. 1: Client/Server design overview.

## 4.1 Brief overview of the architecture

The model and the view are separated. The classes that make up the model, referred to as the game world, are in the `freerails.model.*` packages. The view classes are in the `freerails.view.*` packages. The state of game world is stored by an instance of a class implementing the interface `World`. This class is mutable - one can think of it as a specialised hashmap. Changes to the game world involve adding, removing or changing their properties.

The client and server are separate. They communicate by sending objects to each other. This is done by sending serialized objects over a network connection.

When a new game starts or a game is loaded, the server sends the client a copy of the `World` object. All changes to the game world that occur after the game has started, referred to as moves, are done using the classes in the package `freerails.move.*`.

## 4.2 Server - client communication

The server is the entirety of computational processes that are concerned with evaluating the game mechanics and distributing the game updates, which can be physically separated from the players.

The server knows logged in users and communicates with them via connections which can read and write serialized Java objects over the internet. However, connections can get lost and reestablished, so the server needs to have the ability to re-associate a connection to a player and the player needs the ability to re-identify itself within a new connection.

Solution: Identity is an uuid, given by the server to the client, together with a display name, that the client chooses. Identification is by uuid. If there is a player on the server side without connection and a new connection presents the UUID of this player, instead of creating a new player, the connection is re-associated.

Before that connections have to send their game version id and the version id has to be equal to the version id of the server or the connection will be closed immediately.

Furthermore: Connections not presenting a UUID within certain a timeout or presenting something else are automatically closed.

For this we need handler, that actually process messages and allow to act on the various things as well as do something in the future. (see also Java Executors <https://stackoverflow.com/a/2258082/1536976>)

Proposed structure of identify: Identity (UUID id, String name)

Identified players can send commands/message to the server and may receive responses.

Examples are: chat message (are echoed to all) , chat log request (returns with the last X chat messages), display available inbuilt scenarios, display saved games ready for loading on the server, create random scenario with options, start hosting a game (preparation phase), start game, stop game.

- ChatMessage: String content, String author, String date (certain format)
- SimpleMessage (Request): Enum type of request (chat log request, display available scenarios, )
- ChatLog: List<ChatMessage>
- Scenario/SaveGameInformation: String title, String description, Image preview
- RandomScenarioCreationOptions:
- ...

The server is represented with a list of <id, message> and processes them. The first player on the server to decide to host a game becomes the game master. He decides when to start the game, when to stop it running on the server. However, the players can save the game locally as well (see later)

Stopping the server is done externally by the local client, not through messages. The local server starts at start of the local client and stops when the local client stops.

When a server stops, the current running game is stopped, all connections are closed.

A game consists of a certain number of players and additional data, all contained in a model object. In a game the players are either humans or AI. Access to a player may be restricted by password. Users can decide to use their uuid as password.

If no passwords are stored, the scenario is said to be “pure”, otherwise it is said to be “restricted”. Pure scenarios can always become restricted. Restricted scenarios can only become pure if all participating human players agree. Passwords are part of the game model.

Client options:

- Internal (program version acting as option version)
- Server limited to local connections (default: on, changes are reflected on next start)
- UUID (chosen at first start, may be set)
- Default password (chosen randomly at first start, may be set)
- Use default password (otherwise the player is prompted before sending)

- Start in windowed/fullscreen mode (changes are reflected on next start)
- Music mute (immediate, default: off)
- Music volume (immediate)

Implementation (stored as JSON as key-value pairs, or as properties maybe?)

During the game, only Chat Messages (not part of the game, chat messages not persistent when server stops) and stop game messages (by game master).

The game has a status (running, paused). It pauses if the majority of players hit the pause button (message) or if a player lost a connection. The game master determines the game speed (stored in the game model).

Everyone can send a Move, Moves are collected in a list in the order of arrival at the server. Every move is processed sequentially and first tested for applicability. If so, it gets a running number (applied moves since start of game) and is applied and sent to the client. The client then checks if the current running number is the successor of the last received such move and applies it. If not, a Message (Invalid Move or so) is returned.

A Move should always encapsulates a single atomic game action for each player.

Move: void apply(World), Status applicable(Read-Only-World)

In case of inconsistencies every player can always send a request to obtain the whole world).

## 4.3 Track implementation

The map consists of an rectangular grid consisting of square-sized tiles. Each tile (except those at the border) have exactly eight neighbors which can be uniquely identified by compass points (north, north-east, ...) given the current center tile or by grid positions (row, column).

The total track on the map consists of many track pieces, where each piece connects two neighboring tiles (diagonal connections have  $\sim 1.41$  (square-root of 2) times the length of horizontal or vertical connections).

Path finding of the trains works on a graph where the tiles are nodes and the track pieces connecting neighboring tiles are the edges.

Building and removing track works by adding and removing track pieces. In particular, the planning of a longer piece of newly built track is done by path finding again.

The track itself is visualized by rendering each tile according to its track configuration. The track configuration is an 8 bit value indicating if there is a connecting to one of the 8 neighbors from the current tile. See the attached image for some examples.

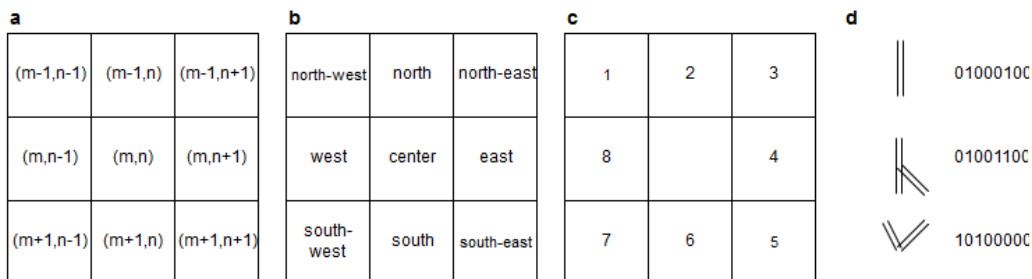


Fig. 2: Track configurations.

The track configurations are not independent from each other. For every connection on a tile towards a neighboring tile, this neighboring tile must also have a connection to this tile. This invariant must be obeyed by not allowing to change track configurations directly, but only by allowing adding or removing of track pieces at a time.

Track pieces have no direction, any train can go on them both ways. However, a train can change direction at every tile at most by 90 degree, effectively inducing some kind of directionality.

Track pieces can be single tracked or double tracked. There can only be one running train on each track piece (and trains have a certain extent, also measured in track pieces). However, stopped trains do not count as obstacles.

The track configuration of a tile is sufficient to draw it uniquely on the tile. Not all possible 8 bit values are valid.

Bridges and stations are a special case. Stations have a orientation and only allow track parallel to their orientation. Bridges span a water tile (other track cannot be put on water) and consist of two track pieces resulting in a parallel configuration.

## 4.4 Computer controlled (AI) players

Is built exactly like a client (communicates by Moves) but without UI or presenter. Lives locally on the server. Shutdown automatically if the game ends.

## 4.5 Coding Guidelines

- Follow package dependency rules (utils does only depend on java libraries, model only on utils, move only on model and utils, ..).
- Avoid circular dependencies between packages. I.e. if any classes in package A import classes from package B, then classes in package B should not import classes from package A.
- Run all unit tests after making changes to see whether anything has broken. You can do this using the check gradle target.
- Javadoc comments. Add a couple of sentences saying what the class does and the reason for its addition.
- Use the Code Formatter with care. Avoid reformatting whole files with different formatting schemes when you have only changed a few lines.
- Consider writing junit tests.
- Consider using assertions.
- Add //TODO comments if you spot possible problems in existing code.
- Use logging instead of System.out or System.err for debug messages. Each class should have its own logger, e.g. `private static final Logger logger = Logger.getLogger(SomeClass.class.getName());`

### Reading

- Effective Java (<http://java.sun.com/docs/books/effective/>) (sample chapters online)
- User Interface Design for Programmers (<http://www.joelonsoftware.com/uibook/chapters/fog0000000057.html>) (available online)



### 5.1 FreeRails (2000-2005)

The original [FreeRails](#) project was registered on Sourceforge on 2000-08-09 with the aim to create a fun game based off the RailRoad Tycoon and RailRoad Tycoon II games. It was described as a real time multi player strategy game where players compete to build the most powerful railroad empire.

The introductory [news message](#) by Luke Ehresman reads:

My friends and I have long loved the game Railroad Tycoon, and would love to see a open source version, very similar to FreeCiv. However, none of us have time to work on this due to dedication to other open source projects. We would love to see people run with the idea, and in a hopeful attempt at pushing the idea off to the open source community, I have created a SourceForge project. To find out more about the FreeRails project (sister project to FreeCiv), visit the web page: <http://freerails.sourceforge.net> and be sure to join our mailing list as we discuss the future and feasibility of this project.

It was programmed in Java and used a [CVS repository](#) for source control. Regular [releases](#) from 2001-07-25 (version 0.0.2) to 2005-09-23 (version 0.2.7) brought it to a quite extensive, playable tech demo including play over network. The source code is under the GPL-2.0 license. Main Java programmer seems to have been Luke Lindsay.

- Last binary release: [FreeRails 0.2.7](#)
- Authors: Luke Ehresman (Original project proposal), Luke Lindsay ((Administrator)), Alexander Opitz, Chris Phillips, Samuel Benzaquen, Dennis M. Sosnoski, Rob, MystiqueAgent, Scott Bennett, Robert Tuck, Sonny Zubia, Steven Mackay, Jan Tozicka, Moises Duarte, [adam@jgf](mailto:adam@jgf), Patrice Espie, Jon Lund Steffensen, Karl-Heinz Pennemann, David Magnuson (Testing), Dave Brosius

#### 5.1.1 C++/C# client for FreeRails

A client for Freerails using C++ was [announced](#) in 2001-08, but no official releases remain. Code can be found in the CVS repository. They shared the same graphics resources.

## 5.2 Railz (2004-2005)

On 2004-02-23, Railz was registered on now defunct [BerliOS](#) as a railway strategy/management game based upon the freerails source code base. It was created by Robert Tuck. Until the beginning of 2005 it saw several releases and the code was considerably changed from the FreeRails project on Sourceforge. An archived version of the [project page](#) is available as well as a automated [exported project page](#). The source code of Railz was also released under GPL-2.0.

- Last binary release: [Railz 0.3.3](#)
- Authors: Those of FreeRails (2000-2005), especially Robert Tuck

## 5.3 FreeRails2 (2007-2008)

On 2007-11-05, [FreeRails2](#) was registered on Sourceforge with the aim to create an optimized version of the FreeRails project. A web start version was included on the [web site](#). It used a [SVN repository](#) to extend the code base of the FreeRails (2000-2005) project. The development mainly focused on improving the infrastructure. The main contributor was Roland Spatzenegger.

- Last binary release: [FreeRails2 0.4.0](#)
- Authors: Those of FreeRails (2000-2005) and Roland Spatzenegger ([cymric\\_npg](#))

## 5.4 Railz2 (2012-2014)

On 2012-04-08, LukeYJ (Luke Jordan?) created the [Railz2](#) project on Sourceforge as a follow-on to the freerails and Railz projects. The code was stored in a [SVN repository](#) and the starting point was the Railz 0.3.3 release. Again the Java code base was extended.

- Last binary release: [Railz2 0.4.0](#)
- Authors: Those of Railz and LukeYJ

## 5.5 FreeRails3 (2016)

On 2016-12-24, LukeYJ created the [FreeRails3](#) project on Sourceforge as a fork from the inactive Freerails2 project. The code was stored in a [GIT repository](#) and the history FreeRails2 was imported. Within a few weeks some issues were fixed and a new release was made with the version number 0.0.4 (smaller than the version of FreeRails before). Finally the code was also placed on [Github](#).

- Last binary release: [FreeRails3 0.0.4](#)
- Authors: Those of FreeRails2 and LukeYJ

## 5.6 FreeRails continuation (2017-)

On 2017-12-12, Trilarion created the [FreeRails](#) project on Github as a continuation of FreeRails 1, 2 and 3 in Java. The history of FreeRails 1, 2 and 3 as well as Railz 1 and 2 was investigated, stored in git branches ([FreeRails on Sourceforge](#) and [Railz on Sourceforge](#)) including the last releases of each of the previous projects.

No release has yet been made.

---

**Note:** The change logs for the years 2000 to 2006 are more like commit messages and are kept here for historical reasons. From 2017 on this change log describes the publication date and changes of every released version.

---

### 6.1 FreeRails continuation (2017-)

Jan 2, 2018, FreeRails 0.4.1

- No changes in functionality, artwork or user interface
- Code of version 0.4.0 made runnable on Java 8 and with a slight round of code cleanup

### 6.2 FreeRails3 (2016)

Dec 25, 2016, lukeyj

- Re-added station info panel
- Only show good supplied or demanded
- Modernised build through maven
- Pulled out file paths into constants

### 6.3 FreeRails (2000-2005)

Sep 3, 2006, Luke

- Give moving trains priority over stationary trains.

- Fixed bugs
- 1551106 Trains with 'auto consist' set don't pickup cargo

Sep 1, 2006, Luke

- Features implemented:
- 987520 Track Contention

Aug 31, 2006 Luke

- Brought back 'cheat' menu.
- Fixed bugs
- 1537413 Exception when building station.

Jul 24, 2006 Luke

- Fixed bug where list of saved games didn't get updated when files were added, removed, or renamed.

Jul 16, 2006 Luke

- Fixed bugs
- 1384250 Wait until full slows frame rate.
- 1384249 Unexpected Exception when removing last wagon.

Dec 5, 2005 Luke

- Fixed bugs
- 1313227 Remove last wagon doesn't work with more than 2 wagons
- 1313225 Harbour: Conversion makes no sense
- 1341365 Exception when calculating stock price after buying shares
- 1303162 Unexpected Exception in SquareTileBackgroundRenderer

Sept 21, 2005 Luke

- Improve exception reporting

Sept 19, 2005 Luke

- Fixed bugs
- 1289010 Net worth on broker screen is wrong
- 1289014 Can buy 100% of treasury stock
- 1289012 No limits on issuing/repaying bonds
- 1295855 Can buy stock when can't afford to
- 1289008 Stock price never changes

Sept 11, 2005 Luke

- Fixed bugs
- 1266577 Broker screen layout

Sept 10, 2005 Luke

- Fixed bugs
- 1269664 Can't tell other player has my stock

- 1266575 Stock holder equity not shown properly on the balance sheet.

Sept 9, 2005 Luke

- Fixed bugs
- 1269676 Stack traces often lost

Sept 8, 2005 Luke

- Fixed bugs
- 1266581 Underscores in map names
- 1223231 “waiting” message on launcher unclear
- 1266582 Progress bar
- 1269679 Train Info doesn’t update

Aug- 31, 2005 Luke

- Fixed bugs
- 1266584 Can’t select which saved game on launcher
- 1269683 Load game when no saved games.
- 1269688 Load game in full screen
- 1269689 Load game across network

Aug- 24, 2005 Luke

- RFEs implemented
- 1223234- Add show FPS to ‘Display’ menu.
- 1223235- Save launcher input
- Fixed bugs
- 1266695 Unexpected exception during network game
- 1266637 OutOfMemoryError
- 1110270 Reproducible Crash Bug
- 1223230 Incorrect station price on popup
- 1223228 NotSerializableException when saving game with trains

Jul 03, 2005 Luke

- More code cleanup.

Jul 03, 2005 Luke

- Reorganisation of existing code.

May 22, 2005 Luke

- Code cleanup
- Improve pathfinder: finding paths for track is now up to 20 times faster.
- More of the same

Apr 10, 2005 Luke

- More work on new train movement classes

Apr 04, 2005 Luke

- More work on new train movement classes

Apr 01, 2005 Luke

- More work on new train movement classes

Feb 20, 2005 Luke

- More work on new train movement classes

Feb 18, 2005 Luke

- Refactoring existing train movement classes in
- preparation to use new classes.

Feb 05, 2005 Luke

- Update website to use SSI
- Work on new train movement classes
- Added AI page to functional spec.

Feb 04, 2005 Luke

- Add new train movement classes.

Jan 27, 2005 Luke

- Added toString() to KEY classes.

Jan 27, 2005 Luke

- Added serialVersionUID field to serializable classes.

Jan 26, 2005 Luke

- Bugs Fixed:
- 1105499- Word wrapping in Html components
- 1105494- Load game with wrong player
- 1105488- Attempting to join game in progress

Jan 25, 2005 Luke

- Work on bug 1105494- (Load game with wrong player).

Jan 24, 2005 Luke

- Second attempt at fixing bug 1103632 (Sound on Linux)

Jan 17, 2005 Luke

- Note, some of theses changes occurred at earlier dates but were not entered into this change log.
- Bugs Fixed:
- 1103632- Sound on Linux
- 1103633- Build station mode
- 1103634- 'P' sets priority orders
- 1102801- keys on train orders

- 1102803- Blank schedule after adding stations
- 1102797- Pause 1st time track is built
- 1103154- Building track quickly with keyboard fails
- 1103150- Can build track in station placement mode
- 1102804- Cursor on map edges
- 1103155- Can't upgrade station with F8
- 1102800- Turbo game speed does nothing
- 1102806- Newspaper does nothing
- 1102798- Building track out of station too expensive
- 1102799- "Can't afford to remove station"
- 1087429- Same icon for info, no tunnels, no bridges
- 1096168- No tooltips on build tab
- 1087428- Wrong cursor message
- 1087431- Message "Illegal track config.."
- 1087373- Stations influence should not overlap
- 1087427- Terrain info dialogue close button
- 1087409- java.io.InvalidClassException
- 1087414- Upgrade track on Ocean -> ArrayIndexOutOfBoundsException
- 1087425- NullPointerException
- 1087426- Can see stations boxes for other players
- 1087433- Can't tell that train roster has focus
- 1087422- Pressing 'I' on other's station ->crash-
- 1005144- java.lang.IllegalArgumentException: Tried to add TrainPosition
- Features implemented:
- 927146- Display natural numbers for trains, stations, etc
- Other changes:
- New track graphics

Jan 14, 2005 Luke

- Updated build.xml
- Minor javadoc updates

Jan 13, 2005 Luke

- Bugs fixed:
- 1098769 Blinking cursor
- 1098767 Can't remove bridges when 'no bridges' selected
- 1099095 Remove track not cancelled
- 1099093 Upgrade track starting at station fails

- 1099083 Remove train, then click train list-> Exception
- 1099091 Station placement cursor wrong colour-
- 1099092 Station influence remains after station removed

Jan 09, 2005 Luke

- Bugs fixed:
- 1087432- Can't remove or upgrade track using mouse

Jan 04, 2005 Luke

- Bugs fixed:
- 1087437- java properties window should word wrap.
- 1087434- Building track out of station
- Other changes:
- Code cleanup

Dec 18, 2004 Luke

- RFEs Implemented:
- 1055501- Automatically build bridges & tunnels
- 931570- Improve Cursor
- 915941- Bridge types GUI
- 915940- Tunnels options GUI

Dec 15, 2004 Luke

- More on track build system. Its almost complete.

Dec 14, 2004 Luke

- Work on track build system. Appropriate track for the terrain
- is now automatically selected. Still some bugs.

Dec 12, 2004 Luke

- Updated functional specification.

Nov 16, 2004 Luke

- Work on GUI to select track type and build mode.

Nov 15, 2004 Luke

- Started using java 1.5 language features
- Updated build.xml to use 1.5 and removed 'format' and 'ConstJava' ant targets.

Oct 27, 2004 Luke

- Bugs Fixed:
- 1054729- Can't build bridges using mouse

Oct 19, 2004 Luke

- Bugs Fixed:
- 1046399- No supply and demand at new stations



Oct 18, 2004 Luke

- RFEs Implemented:
- 1048913- Option to turn off sound
- Bugs:
- Work on 1046399- No supply and demand at new stations

Oct 17, 2004 Luke

- RFEs Implemented:
- 972863- Launcher: progress bar should be on new page
- Bugs Fixed:
- 1047435- Can't rejoin game
- 1047445 Invalid port but next button enabled-
- 1047440 Progress bar not visible when starting network game
- 1047431- No server but no error message.
- 1047422- java.net.SocketException: Connection reset
- 1047412- 2 players, same name -> Exception

Oct 13, 2004 Luke

- Bugs Fixed:
- 1047428 "no players" message goes away
- 1047414 Connected players list should auto update
- 1047439 Shutting down remote client crashes server
- 1047425 2 servers, same port -> Exception
- 1046385 pressing Backspace causes IllegalStateException

Oct 12, 2004 Luke

- Made map scroll when mouse is dragged outside the view port
- when building track.

Sep 18, 2004 Luke

- RFEs Implemented:
- 931581 Build Industry.
- 931594 Show which player is winning.
- 915955 Automatic Schedules.
- 931597 Graph showing total profits over time.
- 915957 Build track by dragging mouse.-
- 932630 Change speed from network clients.

Aug 14, 2004 Luke

- Added ConstJava ant target

- Note, ConstJava adds the keyword ‘const’ to java. It can be typed `/*=const */` so that the files remain valid java files.
- Fixed some mutability problems that it identified.

Aug 10, 2004 Luke

- Implemented City growth
- Work on deadlock and unexpected exception bugs.

Jul 26, 2004 Luke

- Apply Jan Tozicka’s patch for bug 997088 (IllegalArgumentException in OneTileMoveVector.getInstance)

Jul 21, 2004 Luke

- Remove some circular dependencies.

Jul 07, 2004 Luke

- Fixed problem with unit tests in freerails.controller.net

Jul 07, 2004 Luke

- Bugs fixed:
- 972866 Build track by dragging - only when build track selected

Jul 06, 2004 Luke

- RFEs Implemented:
- 915943 Sounds!
- Bugs fixed:
- 984510 freerails.world.player.player; local class incompatible

Jun 25, 2004 Luke

- Bugs fixed:
- 979831 Stack traces printed out when running unit tests

Jun 17, 2004 Luke

- Apply Vincenzo Di Massa’s station distance patch.
- Fixed DisplayModesComboBoxModels.removeDisplayModesBelow(.) so
- that it does not remove display modes when displayMode.getBitDepth() returns DisplayMode.BIT\_DEPTH\_MULTI

Jun 15, 2004 Luke

- Bugs fixed:
- 972869 Crash when track under train removed.
- 972867 Signal towers do nothing - I’ve removed them!
- 972864 Deselect place-station-mode when track selected

Jun 14, 2004 Luke

- Bugs fixed:
- 948668 Building Station on Curve - Cursor changes function -
- 948671 Map City Overlays incorrect

- 967675 No trains/stations but train & station menus selectable
- 972738 Crash when station removed
- 967662 Bottom of terrain info tab cut off in 640\*480 res.
- 972869 Crash when track under train removed.

Jun 13, 2004 Luke

- Bugs fixed:
- 948651 IP Address input should be checked immediately.
- 948649 Dialogue Box Behavior
- 967668 No supply & demand at new station
- 948672 Large numbers of active trains slows performance -

Jun 12, 2004 Luke

- Bugs Fixed:
- 967667 Cannot close multiple dialogue boxes.
- 967664 Fullscreen res. below 640x480 16bit selectable.
- 967666 Selected fullscreen resolution ignored.
- 967713 FPS counter obscures build menu
- 967660 Debug text sent to console
- 948679 Delete/Rebuild single section of track doesn't cost anything

Jun 9, 2004 Luke

- Bugs Fixed:
- 967673 Crash when building track close to edge of map

Jun 6, 2004 Luke

- Bugs Fixed:
- 967677 OutOfMemoryError after starting several new games

Jun 6, 2004 Luke

- RFE implemented:
- 915960 Logging

Jun 5, 2004 Luke

- Bugs Fixed:
- 967129 Main map white on 1.5.0 beta 2
- 941743 Build train dialog closes without building train.
- 967214 EchoGameServerTest hangs

May 31, 2004 Luke

- Bugs Fixed:
- 948653 Crash after loading a saved game when one is not available.-
- 948665 "Show Details" on Train List doesn't work if no train is selected.

- 948659 Dialogue Box Behavior not deterministic
- 948663 Extra Close Button on Station List tab
- 948661 No Formal Specification (see /src/docs/freerails\_1\_0\_functional\_specification.html)
- 948656 Non Movable Dialogue Boxes
- made dialogue boxes movable
- added option to show/hide station names, spheres of influence, and cargo waiting.

May 30, 2004 Luke

- Bugs Fixed:
- 948666 Crash when Building Train with Money < 0 and only one station

May 28, 2004 Luke

- Bugs Fixed:
- 948655 Can't see consist when there are more than 6 wagons
- 948675 Can't upgrade station types
- 948680 No way to tell sphere of influence for a station type

May 27, 2004 Luke

- Bugs Fixed:
- 948676 Waiting list is cut off
- 948673 Cost of Building track/stations not shown
- 948670 Removing non-existent track
- 948654 Locomotive graphic backwards

May 24, 2004 Luke

- Bug fixes for freerails.world.top.WorldDifferences

May 24, 2004 Luke

- Added class freerails.world.top.WorldDifferences - may be useful for RFE 915957!

May 10, 2004 Luke

- Applied Jan Tozicka's first patch for 915957 (Build track by dragging mouse)

May 5, 2004 Luke

- Fix bug in SimpleAStarPathFinder spotted by Jan Tozicka.

Apr 30, 2004 Luke

- Applied Jan Tozicka's patch
- Implements 927165 (Quick start option)

Apr 21, 2004 Luke

- Fix DialogueBoxTester
- Tweak build.xml

Apr 11, 2004 Luke

- Added some javadoc comments.

- Added hashcode methods to classes that override equals.
- Code cleanup
- Let track be built on terrain of category 'Industry' and 'Resource'

Apr 9, 2004 Luke

- Fixed bug 891452 (2 servers same port, no error message)
- Fixed bug 868555 (Undo move by pressing backspace doesn't work)
- Fix for bug 910132 (Too easy to make money!)
- More work on bug 910902 (Game speed not stored on world object)

Apr 8, 2004 Luke

- Added website to CVS
- Added website deployment targets to build.xml

Apr 7, 2004 Luke

- Implemented 930716 (Scale overview map) by
- incorporating code from Railz.

Apr 6, 2004 Luke

- Fix selection of track type and build mode that was broken by the game speed patch.

Apr 6, 2004 Luke

- Implemented 915945 (Stations should not overlap)
- Increased the quality of scaled images returned by ImageManagerImpl

Apr 5, 2004 Luke

- Implemented 915952 (Boxes showing cargo waiting at stations)

Apr 5, 2004 Luke

- Fixed 910134 Demand for mail and passengers
- Updated javadoc comments in freerails.server.parser.

Apr 4, 2004 Luke

- Implemented 927152 Show change station popup when add station is clicked

Apr 3, 2004 Luke

- Apply Jan Tozicka's 2nd patch for 910902

Apr 2, 2004 Luke

- Fixed bug 910130 (Placement of harbours)

Apr 1, 2004 Luke

- Made trains stop for a couple of seconds at stations.
- 915947 Implement wait until full.

Apr 1, 2004 Luke

- 910138 After building a train display train orders
- 910143 After building station show supply and demand

- Started rewriting freerails in C#!

Mar 30, 2004 Luke

- Implemented 915949 (Balance sheet)
- Fixed bug where an exception was thrown if you moved the cursor when ‘View Mode’ was selected on the build menu.

Mar 29, 2004 Luke

- Implemented 915948 (Income statement)

Mar 27, 2004 Luke

- Updated coding guidelines.

Mar 15, 2004 Luke

- Added ‘Show java properties’ to about menu.

Mar 14, 2004 Luke

- Implemented 910123 (Add/remove cargo to cities more frequently).

Mar 13, 2004 Luke

- Fixed various bugs where exceptions were getting thrown.
- Stopped the client window getting displayed before the world is loaded from the server.

Mar 13, 2004 Luke

- Implemented 910126 (Train list on RHS panel)
- Started 915303 (Icons for buttons and tabs) - the tabs on the RHS now have icons instead of titles.

Mar 12, 2004 Luke

- Apply Jan Tozicka’s patch for 910902 (Game speed not stored on world object).

Mar 9, 2004 Luke

- Increase client performance. 93FPS to 111FPS on my machine.
- Note, I get much higher FPS when the client and server are in different JVMs.

Mar 8, 2004 Luke

- Readied 640x480 fixed size windows mode. It is useful for taking screen shots and making sure the dialogue boxes work in 640x480 fullscreen mode.

Mar 6, 2004 Luke

- Added Scott Bennett’s terrain randomisation patch.

Mar 6, 2004 Luke

- Remove ‘never read’ local variables.
- Fixed bug 910135 Trains jump when game un paused
- Fixed bug 891360 Trains don’t get built while game is paused

Mar 5, 2004 Luke

- Applied Jan Tozicka’s patch for bug 900039 (No clear indication game is paused).

Mar 4, 2004 Luke

- Minor changes to coding guidelines.

- Fixed stale serialVersionUID problem in freerails.world.player.Player
- Made ant script insert build id into README and about.htm

Mar 3, 2004 Luke

- Apply Scott Bennett's removal\_of>Loading\_text patch.

Mar 3, 2004 Luke

- Implemented Request 905446 Track should be continuous
- Implemented Request 905444 Multi player support: different track

Mar 2, 2004 Luke

- Implemented Request 905443 Multi player support: different trains

Mar 1, 2004 Luke

- Implemented Request 905441 Multi player support: different bank accounts
- Note, presently some of the dialogue boxes are not working. This will be fixed as adding multi player support continues.

Feb 27, 2004 Luke

- Some fixes for DialogueBoxTester.

Feb 27, 2004 Luke

- Refactoring in preparation for multiplayer support.

Feb 26, 2004 Luke

- Applied Jan Tozicka's 'Shortcuts for game speed' (patch 904903).

Feb 21, 2004 Luke

- Fix 891359 - Javadoc package dependencies out of date
- Tidy up javadoc

Feb 20, 2004 Luke

- Fix 839371 - Goods & livestock wagons appear the same on train orders

Feb 20, 2004 Luke

- Fix bugs 867473 and 880450 (Intermittent deadlocks).

Feb 18, 2004 Luke

- Fix bug 839331 - set initial game speed to 'slow' instead of paused
- Fix bug 874416 (station icon hides after track-upgrade)
- Fix bug 839361 (Several industries of the same type in same city)
- Fix bug 891362 (Cancel button on select engine dialogue doesn't work )
- Fix bug 891431 No link between train list and train orders screens

Feb 18, 2004 Luke

- Removed unreachable code.
- Fix build.xml

Feb 17, 2004 Luke

- Apply move infrastructure patch.
- Apply OSX work around.

Feb 16, 2004 Luke

- Add new select station popup to train orders dialogue (fixes bug 891427).
- Add ‘About’ dialogue (fixes bug 891377)
- Add ‘How to play’ dialogue (fixes bug 891371)

Feb 6, 2004 Luke

- Apply Robert Tuck’s patch to fix bug 880496 (User stuck after connection refused)

Feb 5, 2004 Luke

- Apply Robert Tuck’s Mac OS X fixes.
- Uncomment out code in TrackMaintenanceMoveGenerator

Feb 4, 2004 Luke

- Add testDefensiveCopy() to WorldImplTest

Jan 19, 2004 Luke

- Applied Robert Tuck’s launcher patch.

Dec 31, 2003 Luke

- Remove some unused code.
- Fix some things jlint moaned about - perhaps slightly pointless!

Dec 30, 2003 Luke

- Refactoring to change the threads in which moves are executed.
- (i) Moves are pre-committed on the client’s copy of the world by the thread “AWT\_EventQueue.”
- (ii) All moves are now executed on the server’s copy of the world in freerails.server.ServerGameEngine.update() by the thread “freerails server”.
- (iii) Moves received from the server are now executed on the clients copy of the world in freerails.client.top.run() by the client thread by the thread “freerails client: ...”
- Moves are passed between threads using queues.
- Currently starting new games and loading games does not work.
- Removed most of the passing of mutexes between classes.

Dec 29, 2003 Luke

- Apply Robert Tuck’s patch to BufferedTiledBackgroundRenderer.
- Make the client keep its own copy of the world object even when it is in the same VM as the server.

Dec 24, 2003 Luke

- Prepare for release.

Dec 23, 2003 Luke

- Refactoring to remove some cyclic dependencies.

Dec 20, 2003 Luke

- Apply part of Robert Tuck’s performance patch.



- Update side on wagon graphics.
- Fix for bug 839355 (User not told why track cannot be built)

Dec 18, 2003 Luke

- Fix for bug 855729 (Game does not start on pre 1.4.2 VMs)

Dec 17, 2003 Luke

- Move UNITS\_OF\_CARGO\_PER\_WAGON constant to WagonType.

Dec 17, 2003 Luke

- Applied Robert Tuck's patch to fix apparent network lag.
- Tweaked 'format' ant target so that it does not format files that are up to date.

Dec 13, 2003 Luke

- Fix bug: stations on the trains schedule can now be changed again.

Dec 13, 2003 Luke

- Fixed bug: passengers are now demanded by cities and villages.
- Fixed bug: track maintenance cost is no longer equal to the build cost.
- Fixed bug 839366 (No feedback when trains arrive)

Dec 12, 2003 Luke

- Add Robert Tuck's new train graphics.

Dec 8, 2003 Luke

- Deprecate methods that take a mutex as a parameter.

Dec 6, 2003 Luke

- Apply source code formatting.

Dec 5, 2003 Luke

- Apply Robert Tucks move ahead patch.

Nov 30, 2003 Luke

- Fixed bug 839376 (Harbours are not painted properly)

Nov 30, 2003 Luke

- Fixed bug 839336 (Removing station train heading to causes Exception)

Nov 29, 2003 Luke

- Fixed bug 839392 (After F8 to build station, position still follows mouse)
- Added jalopy 'format' target to build.xml

Nov 18, 2003 Luke

- Applied Robert Tuck's patch to fix the bug that occurred with 1 local client and 1 networked client in a 2nd VM.

Nov 10, 2003 Luke

- Made MoveExecuter non-static.
- Fixed bug 835337.
- Remove debug console output.

Nov 9, 2003 Luke

- Applied Robert Tuck's to fix bug 835241.

Nov 3, 2003 Luke

- Added Scott Bennett's enhanced city tile positioner.

Nov 03, 2003 Luke

- Applied Robert Tuck's patches to update the launcher gui.
- Added Scott Bennett's extra Cities

Oct 18, 2003 Luke

- Applied Robert Tuck's patch adding comments to ServerGameEngine.
- Other javadoc updates.

Oct 13, 2003 Luke

- Applied Robert Tuck's network patch.

Oct 06, 2003 Luke

- Fixed, I think, bug where trains went off the track.

Oct 04, 2003 Luke

- Update CVS write permissions.

Sep 12, 2003 Luke

- Add Robert Tuck's 'build' tab patch.

Sep 07, 2003 Luke

- Added progress bar to show what is happening while the game is loading.

Sep 03, 2003 Luke

- Added GUI to select display mode and number of clients.

Aug 28, 2003 Luke

- Made train speed decrease with no of wagons.
- Made fare increase with distance travelled.
- Made CalcSupplyAtStations implement WorldListListener so that when a new station is added, its supply and demand is calculated by the server.

Aug 25, 2003 Luke

- Added new Train orders dialogue.
- Made changes to train consist and schedule use Moves instead of changing the DB directly.
- Lots of other changes/fixes.

Aug 23, 2003 Luke

- Removed cruft from the experimental package.
- Added a simple train list dialogue, accessible via the display menu.
- Made the engine images have transparent backgrounds and flipped them horizontally.

Aug 19, 2003 Luke

- Applied Robert Tuck's patches that separated the client and server and allow you to start up two clients in the same JVM.
- Fixed painting bug that occurred when you started two clients.
- Major refactor to get the checkdep ant target working again.

Aug 11, 2003 Luke

- You are now charged for track maintenance once per year.
- Cargo conversions occur when you deliver cargo to a station if an industry that converts the relevant cargo is within the station radius.

Aug 07, 2003 Luke

- Applied Robert Tuck's patches to:
  - (i) Stop the Terrain Info panel from setting its preferred size to a fixed value.
  - (ii) Fix the issue with starting a new map and being unable to lay track.
  - (iii) Update remaining classes to use MoveExecutor.
  - (iv) Add the station info panel to the tab plane.
  - (v) Add the train info/orders panel to the tab plane.

Aug 06, 2003 Luke

- Applied Robert Tuck's patch to stop the split pane divider getting focus when you press F8.
- Added the field 'constrained' to AddTransactionMove. When this is set to true, the move will fail if you don't have enough cash.
- Made the building and upgrading track cash constrained.

Aug 04, 2003 Luke

- Added 5 patches contributed by Robert Tuck
  - (i) Changes to build.xml
  - (ii) Added 'View mode' to build menu.
  - (iii) Update to train schedule so that stations can be added and removed.
  - (iv) Changes to MoveChain and Addition of MoveExecutor.
  - (v) Adding TabbedPane to the RHS with a tab to show terrain info.
- Made build.xml copy the game controls.html file.

Aug 02, 2003 Luke

- Increased the number of resource tiles that are placed around cities.
- Fixed bug where cargo was added to trains before wagons were changed.

Aug 01, 2003 Luke

- Fixed failure in DropOffAndPickupCargoMoveGeneratorTest.

Jul 30, 2003 Luke

- The player gets paid for delivering cargo, simply \$1,000 per unit of cargo for now. See `freerails.server.ProcessCargoAtStationMoveGenerator`
- Fixed bug where 40 times too much cargo was being produced by changing figures in `cargo_and_terrain.xml`

Jul 27, 2003 Luke

- Got DropOffAndPickupCargoMoveGeneratorTest running without failures.

Jul 21, 2003 Luke

- The player now gets charged for: building stations, building trains, upgrading track
- The text for the 'Game controls' dialogue box is now read in from a file rather than hard coded into the java.

Jul 08, 2003 Luke

- Added initial balance of 1,000,000.
- Added prices to the track types defined in track\_tiles.xml
- Updated the track XML parser to read in the track prices.
- Updated the build track moves that you get charged when you build track and get a small credit when you remove track.

Jul 07, 2003 Luke

- Wrote 'Move' class to add financial transactions.
- Changed the class that adds cargo to stations so that- it adds 40 units per year if the station supplies one carload per year.

Jun 30, 2003 Scott

- Cargo is now transferred correctly

Jun 28, 2003 Luke

- Moved 'show game controls' menu item to the Help menu.
- Removed 'add cargo to stations' menu item from the game menu. Now cargo is added to stations at the start of each year.
- Set the initial game speed to 'moderate'.
- Added junit test for DropOffAndPickupCargoMoveGenerator

Jun 28, 2003 Luke

- Moved classes to remove circular dependencies between- packages and updated the 'checkdep' ant target.

Jun 27, 2003 Luke

- Added 'station of origin' field to CargoBatch and updated- the classes that use CargoBatch as appropriate. It lets us check whether a train has brought cargo back to the station- that it came from.

Jun 27, 2003 Luke

- Added 'no change' option to train orders - it indicates that a train should keep whatever wagons it has when it stops at a station.
- Made 'no change' the default order for new trains.

Jun 15, 2003 Luke

- Improved the train orders dialogue to show- the current train consist and what cargo the train is carrying.

Jun 15, 2003 Luke

- Fixed a load of problems with station building.
- stations can now only be built on the track
- building a station on a station now upgrades the station rather than adding a new one.

- building stations is now fully undoable in the same way as building track.

Jun 15, 2003 Luke

- The map gets centered on the cursors when you press 'C';
- Pressing 'I' over a station brings up the station info dialogue box.
- Station radii are defined in track xml.
- The radius of the station type selected is shown on the map when the station types popup is visible.

Jun 14, 2003 Luke

- Fixed bug where train went past station before turning around.

Jun 12, 2003 Luke

- Improved javadoc comments.

Jun 11, 2003 Luke

- Add change game speed submenu to game menu.

Jun 11, 2003 Scott

- Implemented the Train/Station cargo drop-off and pickup feature, trains currently only pickup cargo. Its playable!

Jun 05, 2003 Luke

- Added loadAndUnloadCargo(..) method to freerails.controller.pathfinder.TrainPathFinder

Jun 04, 2003 Luke

- Updated freerails.world package overview.

Jun 01, 2003 Luke

- The game times passes as real time passes.

Jun 01, 2003 Luke

- Rewrote ClientJFrame using Netbeans GUI editor.
- Added JLabels to show the date and available cash to ClientJFrame.

May 31, 2003 Luke

- Pressing backspace now undoes building/removing track.

May 31, 2003 Luke

- Make build track moves undoable.

May 31, 2003 Luke

- Cargo gets added to stations based on what they supply, currently this is triggered by the 'Add cargo to stations' item on the game menu.

May 19, 2003 Scott

- Fixed the problem and deviation from the design ;-) of the station cargo calculations, there's now a temporary menu item on the display menu. Use this to manually update the cargo supply rates.

May 18, 2003 Luke

- Uses the new engine and wagon images on the select wagon, select engine, and train info dialogue boxes.

May 18, 2003 Scott

- The cargo supplied to a station can now be viewed from the menu, although some more work is needed.

May 16, 2003 Luke

- Now loads tile sized track images instead of grabbing- them from the big image.

May 12, 2003 Luke

- Now prints out the time it takes to startup.

May 11, 2003 Luke

- Track is shown on the overview map again.
- Rules about on what terrain track can be built have been added, this is driven by terrain category.

May 10, 2003 Luke

- Rejig track and terrain graphics file names following discussion on mailing list.
- Generated side-on and overhead train graphics.

May 05, 2003 Luke

- Added station info dialogue.
- Fixed some bugs related to loading games and starting new games.

May 05, 2003 Luke

- Changed map view classes to use a VolatileImage for a backbuffer.

May 05, 2003 Luke

- Added terrain info dialogue.

May 03, 2003 Luke

- Fixed river drawing bug.

May 02, 2003 Luke

- The terrain graphics now get loaded correctly although there is a bug in the code that picks the right image for rivers and other types that are drawn in the same way.

May 01, 2003 Luke

- Split up track and terrain images.

Apr 28, 2003 Luke

- Integrate new terrain and cargo xml into game. Temporarily lost terrain graphics.

Apr 19, 2003 Luke

- More work on schedule GUI, you can set change the station that a train is going to.

Apr 19, 2003 Luke

- Work on train schedule GUI.

Apr 16, 2003 Luke

- Added NonNullElements WorldIterator which iterates over non-null elements
- Stations now get removed when you remove the track beneath them
- Station name renderer and train building and pathfinding classes updated to handle null values for stations gracefully.

Apr 10, 2003 Scott

- Added City Names
- Added Random City Tile positioning.
- Cities are now no longer related to the image map. Positions are determined by the data in the south\_america\_cities.xml file.

Apr 04, 2003 Luke

- Simple train schedules, set the 4 points on the track that trains will travel between by pressing F1 - F4- over the track.

Apr 04, 2003 Luke

- Added package comments for javadoc.

Mar 22, 2003 Luke

- Got the game running again!

Mar 19, 2003 Luke

- Refactored to use the new world interface, does not run yet.

Mar 10, 2003 Luke

- Fixed bug [ 684596 ] ant build failed

Mar 10, 2003 Luke

- Added the MapViewJComponentMouseAdapter in MapViewJComponentConcrete.java contributed by Karl-Heinz Pennemann - it scrolls the main map while pressing the second mouse button.

Mar 10, 2003 Luke

- Added mnemonics contributed by Scott Bennett

Jan 24, 2003 Luke

- Release refactorings.

Jan 12, 2003 Luke

- Fixed javadoc errors.

Jan 12, 2003 Luke

- Major refactoring
- added ant target, checkdep, to check that the dependencies between packages are in order. What it does is copy the java files from a package together with the java files from all the packages that it is allowed to depend on to a temporary directory. It then compiles the java files from the package in question in the temporary director. If the build succeeds, then the package dependencies are ok.

Jan 11, 2003 Luke

- Refactoring and removing dead code.

Jan 10, 2003 Luke

- Added package.html to freerails.moves
- refactoring to simplify the move classes.

Dec 22, 2002 Luke Lindsay

- Added 'Newspaper' option to 'game' menu to test drawing on the glass panel. The same technique can be used for dialogue boxes.

Dec 04, 2002 Luke Lindsay

- The classes from the fastUtils library that are needed by freerails have been added to the freerails source tree, so you no longer need fastUtils.jar on the classpath to compile and run freerails.

Dec 01, 2002 Luke Lindsay

- Prepare for release.

Dec 01, 2002 Luke Lindsay

- The trains no longer all move at the same speed.

Nov 30, 2002 Luke Lindsay

- Load, save, and new game now work again.

Nov 30, 2002 Luke Lindsay

- The path finder now controls train movement. Press t with the cursor over the track and all the trains will head for that point on the track.

Nov 27, 2002 Luke Lindsay

- Wrote SimpleAStarPathFinder and a unit test for it. It seems to work. The next step is use it together with NewFlatTrackExplorer to control train movement.

Nov 26, 2002 Luke Lindsay

- More or less finished NewFlatTrackExplorer and incorporated it into the main game code.

Nov 26, 2002 Luke Lindsay

- Wrote NewFlatTrackExplorer and NewFlatTrackExplorerTest, in preparation for writing a pathfinder.

Nov 24, 2002 Luke Lindsay

- Rewrote PositionOnTrack and added PositionOnTrackTest. track positions can now be store as a single int.

Nov 24, 2002 Luke Lindsay

- Organise imports.

Nov 09, 2002 Luke Lindsay

- Changes to how the mainmap's buffer gets refreshed.vInstead of the refresh being driven by the cursor moving, it is now driven by moves being received. This means that it it will refresh even if the moves are generate by another player.

Nov 08, 2002 Luke Lindsay

- Stations can be built by pressing F8.
- The station types no longer appear with the track types on the build menu.

Nov 06, 2002 Luke Lindsay

- Fixed 'jar\_doc' task in build.xml

Nov 05, 2002 Luke Lindsay

- Moving trains: the class ServerGameEngine has a list of TrainMover objects, which control the movement of individual trains. Movement is triggered by calls to ServerGameEngine.update() in the GameLoop's run() method.

Nov 03, 2002 Luke Lindsay

- Improvements to TrainPosition and ChangeTrainPositionMove classes



Oct 28, 2002 Luke Lindsay

- Fix javadoc warnings
- Add 'upload to sourceforge' task to build.xml
- Add world\_javadoc task to build.xml.

Oct 27, 2002 Luke Lindsay

- Wrote ChangeTrainPositionMove and ChangeTrainPositionTest

Oct 27, 2002 Luke Lindsay

- Wrote TrainPosition and TrainPositionTest to replace Snake class.

Oct 16, 2002 Luke Lindsay

- Removed cyclic dependencies from the rest of the project.

Oct 16, 2002 Luke Lindsay

- Refactored the freerails.world.\* packages so that (1) freerails.world.\* do not depend on any other freerails packages. (2) there are no cyclic dependencies between any of the freerails.world.\* packages. This should make it easier to maintain.

Oct 13, 2002 Luke Lindsay

- Added trains! They don't move yet. Hit F7 when the cursor is over the track to build one.

Oct 13, 2002 Luke Lindsay:

- Add a task to build.xml that runs all junit tests.
- Change build.xml to work under Eclipse.

Sep 29, 2002 Luke Lindsay:

- Reorganised package structure.
- Changed files that were incorrectly added to the cvs as binaries to text
- Small changes to build.xml so that the ChangeLog, TODO, and build.xml files are included in distributions.
- Changed DOMLoader so that it works correctly when reading files from a jar archive.

Sep 24, 2002 Luke Lindsay:

- Updated TrainDemo, it now draws wagons rather than lines.

Sep 23, 2002 Luke Lindsay:

- Wrote a simple demo, TrainDemo, to try out using FreerailsPathIterator and PathWalker to move trains along a track. To see it in action, run: experimental.RunTrainDemo

Sep 22, 2002 Luke Lindsay:

- wrote PathWalkerImpl and PathWalkerImplTest

Sep 19, 2002 Luke Lindsay:

- wrote SimplePathIteratorImpl and SimplePathIteratorImplTest
- removed the method boolean canStepForward(int distance) from the interface PathWalker so that looking ahead is not required.

Sep 16, 2002 Luke Lindsay:

- Updated and commented FreerailsPathIterator and PathWalker interfaces.

- build.xml written by JonLS added. (Sorry, I - forgot to add it to the change log earlier.)

Sep 08, 2002 Luke Lindsay:

- Wrote 'Snake' class that represents a train position.

Aug 26, 2002 Luke Lindsay:

- Games can now be loaded and saved.
- New games can be started.

Aug 18, 2002 Luke Lindsay:

- More work on active rendering fixes for linux.

Jul 28, 2002 Luke Lindsay:

- Partially fixed active rendering under linux.

Jul 04, 2002 Luke Lindsay:

- Rotate method added to OneTileMoveVector

21 Jun, 2002 Luke Lindsay:

- Fullscreen mode
- GameLoop, freerails now uses active, rather than passive, rendering.
- Work on separating the model and view.
- Tilesets can be validated against rulesets - ViewLists.validate(Type t)
- FPS counter added.

Mar 04, 2002 Luke Lindsay:

- Rearrange dependencies in freerails.world...

Mar 02, 2002 Luke Lindsay:

- Reorganisation of package structure.

Feb 16, 2002 Luke Lindsay:

- Unrecoverable FreerailsExceptions replaced with standard unchecked exceptions.
- Changed CVS directory structure.
- This ChangeLog started!
- genindex
- search

Built on Nov 14, 2022