

<p style="text-align: center;"><b>Transport Empire</b> <b>Design Document</b> <b>Ver 0.1.03 Date: 15.september.2005</b> <b>Uzurpator DD + Wiki FRD merged</b></p>
---

### About this document

This document describes the game as it was discussed up to 15rd September 2005.

Normal describing text is written like this

*Options and comments are in italics*

Modifications are marked with an asterisk -> \*

New content is marked with an exclamation mark -> !

Currently the DD keeper is Uzurpator (that is me). If you have any comments, or updates - please contact me so I can track version changes.

BEFORE any change is included in the document it FIRST has to be approved by the rest of the development team.

The default format of the file is MS Word DOC. If you are a Linux Nazi you either have to deal with it or convert it to ps/pdf on your own.

Uzio out

*Hyro: Linux Nazi is someone who demands .ps or .pdf despite the fact that the provider of the original document might not have technical possibilities to make the conversion. Besides the rule is: I made the document - I decide the format. I convert .ps to .pdf - you can deal with converting .doc to .ps.*

*Linux users will understand that.*

---

### Table of contents:

#### 1.Project Overview

- 1.1 What is TE
- \* 1.2 Basic Agreements
- 1.3 Configuration
- \* 1.4 Folder Structure
- 1.5 Scale
- 1.6 Sets

#### 2.Map

- 2.1 Size
- 2.2 Height
- 2.3 Water
- 2.4 Rivers
- 2.5 Terrain types

#### 3. /deleted/

#### **4.Transporting, general**

- \* 4.1 Cargo Types
- ! 4.2 Type Definition
- ! 4.3 Cargo Definition
- 4.4 Loading Rules
- \* 4.5 Cargo Class - Raw materials
- \* 4.6 Cargo Class - Semi finished products
- \* 4.7 Cargo Class - Goods
- \* 4.8 Cargo Class - Manifest
- \* 4.9 Cargo Class - Express
- \* 4.10 Warehouses
- 4.11 Route Planning

#### **5.Game World**

- \* 5.1 Industries - basics
- \* 5.2 Industries - production chains
- \* 5.3 Industries - distribution centres
- \* 5.4 Industries - setting, removal, production level changes
- ! 5.5 Industries - Definitions
- \* 5.6 Seaports
- 5.7 Cities - General
- \* 5.8 Cities - Local Authorities
- \* 5.9 Cities - Business district
- \* 5.10 Cities - Residential district
- \* 5.11 Cities - Industrial district
- \* 5.12 Returnee
- \* 5.13 Buildings - Definition

#### **6.Trains**

- \* 6.1 Track - Definition
- \* 6.2 Track - Directions and Building
- \* 6.3 Track - Grades
- 6.4 Track - Switches
- \* 6.5 Track - Curves
- \* 6.6 Track - Signals
- \* 6.7 deleted
- \* 6.8 Stations - General
- \* 6.9 Stations - Platforms
- \* 6.10 Stations - Workshops
- 6.11 Stations - Movement
- 6.12 Rolling Stock - Definitions
- 6.13 Rolling Stock - Building

#### **7.Road Vehicles**

- 7.1 Roads - Definition
- 7.2 Roads - Directions and Building
- 7.3 Roads - Grades
- 7.4 Roads - Intersections
- 7.5 Roads - Curves
- 7.6 Roads - Freeways
- 7.7 Roads - State and Private Roads
- 7.8 Stations - Loading bays

- 7.9 Stations - Workshops
- 7.9 Vehicles - Definitions
- 7.10 Vehicles - Building
- 7.11 Vehicles - Special Consists

## **8.Aircraft**

- 8.1 Airports
  - 8.1.1 Runways
  - 8.1.3 Hangars
  - 8.1.4 Heliports
  - 8.1.5 Loading Bays
- 8.2 Aircraft

## **9.Marinal Transport**

- 9.1 Infrastructure
  - 9.1.1 Wharves
  - 9.1.2 Docks
  - 9.1.3 Channels
  - 9.1.4 Traffic organisation
- 9.2 Tugs
- 9.3 Barges
- 9.4 Ships

## **10. Engineering**

- 10.1 Bridges
- 10.2 Tunnels
- 10.3 Elevations

## **11. Vehicles Misc.**

- 11.1 Reliability
- 11.2 Life-span
- 11.3 Wrecks

## **12. World Misc.**

- 12.1 Stock Market
- 12.2 Bonds
- 12.3 Loan
- 12.4 Cargo Distribution
- 12.5 Fares
- 12.6 Ratings - station
- 12.7 Ratings - industry
- 12.8 Ratings - city
- 12.9 Scenarios - plot side of the game, goals
- 12.10 High Scores
- 12.11 Game Setting
- 12.12 Weather & Disasters
- 12.13 Pollution
- 12.14 Tourist attractions
- 12.15 Difficulty settings

## **13. Interface**

#### 14. Wild Idea Space

```
//-----//  
//      Chapter 1 : Project Overview      //  
//-----//
```

Transport Empire is a game about transporting goods and passengers on a player built network of roads, tracks etc.

The game is roughly based on the concept of Transport Tycoon

```
// ----      1.1 Title ---- //
```

Transport Empire

```
// ---- 1.2 Basic agreements ---- //
```

Game is placed in full 3D environment.

Internally The Game uses such measurement units:

- Length is expressed in metres (m)
- Time is expressed in seconds (s)
- Weight is expressed in kilograms (kg)
- Velocity is expressed in meters/second (m/s)
- Volume is expressed in cubic meters (m<sup>3</sup>)
- Force is expressed in Newtons (N)
- Currency is expressed in Credits (TEC)
- Position is expressed in Tiles (TET)
- Angle is expressed in radians (rad)
- Power is expressed in Watts (W)

The Game will provide a conversion of the internally used units into human friendly units - including possibility to define user - units. Default displayed units may include

- Kilometres (km)
- Hours/Days/Weeks (h/d/w)
- Tonnes (t)\* (1 tonne = 1000 kg)
- Tons (t) (1 ton = 2000 lb)
- Kilometres per hour (km/h, kph)
- Kilonewtons (kN)
- British Miles (m)
- Us Miles (m)
- Knots (Kt)
- Pounds (lb)
- Miles per hour (mph)
- Pound Force (lbf)
- British Pound (£)
- Dollar (\$)
- Degree
- Kilowatt (kW)

*Currency is expressed in Credits (C) - meaning - that for example 1 British pound = 10000 credits, 1 dollar = 7000 credits etc*

```
// ---- 1.3 Configuration ---- //
```

The Game is configured via a set of spreadsheets. These spreadsheets are tab delimited text files. Configuration sheets are placed in folders. The Game assumes that files in certain folders are configuration files (of a given type) and will parse them accordingly (this way config files can be added without modifying existing ones)

For example - all files in 'te/vehicles/rail/models' are definitions of models for train vehicles.

*Tab files are easy to edit and parse - MS Excel can edit them - also there are other editors - Diablo II mod base uses them since Diablo II is also configured by a set of spreadsheets*

// ---- 1.4 Folder Structure ---- //

Folder structure shows where all files are stored by default - the game will seek certain objects (models, configs etc) in predestined folder.

```

TE
|-visuals
| |-shaders          // textures and shaders are stored here
| \-models           // models (geometry) for all objects
|-sfx
| |-music            // music
| \-sound            // sfx
\ -conf
| -types              // definitions of terrain and cargo types
| -industry
| | -cargoes          // cargo definitions
| | -buildings        // building definitions (warehouse, processor etc)
| | \-entities        // industry definitions (composition of buildings, names etc)
| -vehicles           // stores cargo carrier type definitions (hopper, etc)
| | -ducts
| | | -canals         // definitions of canals
| | | -tracks         // definitions of railroad tracks
| | \-roads           // definitions of roads
| -rail
| | -vehicles         // definitions of rail vehicles visible on purchase list
| | \-models          // invisible on purchase list, are called upon by vehicles
| -road
| | -vehicles         // definitions of road vehicles visible on purchase list
| | \-models          // invisible on purchase list, are called upon by vehicles
| -air
| | -aircraft         // definitions of aircraft visible on purchase list
| | \-models          // invisible on purchase list, are called upon by vehicles
| \-water
| | -vessels          // definitions of aircraft visible on purchase list
| | \-models          // invisible on purchase list, are called upon by vehicles
| -citybuildings     // definitions of city buildings
| -warehouses         // definitions of storage units
| -stations
| | -platforms        // definitions of platforms
| | -workshops        // definitions of workshops
| | \-loaders         // definitions of loading/unloading apparati
| -rvbays
| | -bays             // definitions of loading bays
| | -platforms        // definitions of platforms
| | -ramps            // definitions of ramps
| | \-workshops       // definitions of workshops
| -airfields
| | -taxiways         // definitions of taxiways
| | -loaders          // definitions of loading buildings (terminals etc)
| | \-hangars         // definitions of hangars
\ -ports
| -docks              // definitions of docks
| -wharves            // definitions of wharves
| \-locks             // definitions of locks

```

*Note - this is a quick mock-up - showing the concept*

// ---- 1.5 Scale ---- //

One tile is a square with size of 16m x 16m (256 sqm). Default map is 1024x1024 tiles - total of 16384m x 16384m (16 square km!)

There are around 20-40 cities. Average city has population of around 3000-5000 people. Biggest cities are ~500000. Biggest cities take around 64 tile radius (3.29 square kilometres).

*Overall - the transports and roads are the same as in TTD, but the world is much bigger - forcing bigger distances. Also bigger space will allow better maps based on the real world. Better slope-to-distance and better scenery. One thing though - 16m tile is only a reference for the sizes of all the objects - not a reference of actual distances/speeds etc.*

// ---- 1.6 Sets, groups, random choice ---- //

Sets - each object in the game is grouped in sets. Player chooses certain sets (like: american trains set + truck set + africa terrain set) before making a game.

Groups - some objects are identified by groups. For example buying Scania Truck will actually open a window with a few types of this truck (Like: Scania Hopper Truck, Scania Refrigerated Truck etc).

Random Choice - some objects will be built and then the actual object will be selected randomly from a cache of objects (which are available in the given period) - for example buying a boxcar will randomly assign a livery to it.

```
//-----//  
//          Chapter 2 : Map          //  
//-----//
```

The Map is an rectangle area on which most of game's activity takes place. The Map shows roads, tracks, trees, rivers and all other objects game supports. The Map is composed of square tiles.

Tile edge is 16 metres long. It is composed of four triangular areas (used for placement of some objects).

Tile vertices cannot differ in height more then 10 meters. Landscaping tools will not allow building 'pillars' of rock. Also - tiles are joint - eg 4 tiles can share 1 vertex. It is impossible to create a "cliff" by using tile edges.

Landscaping is achieved in two ways - either automatic while building a track or road, or manual - when the player will carve the land using a set of designated tools. Landscaping costs money.

Landscaping can be up to the smallest 0.1m resolution in height difference.

Some things (like rivers, hard bedrock or shoreline) that cannot be landscaped.

*idea: maybe the game should store origin heights of all vertices and base costs of landscaping on how much deviation form origin height occurred - this way landscaping would be much more limited thus forcing to manage hills rather*



*then just levelling them on sight as it is now in TTD/LOMO*

// ---- 2.1 Size ---- //

The Map can have any size beginning at 128x128 vertices. Size has an upper limit of 65536x65536 vertices. Both sides of The Map can be of any value between these two (so 170x2021 is a viable size).

*mental note - soft limit around 4096 or 8192 would probably have to be imposed.*

// ---- 2.2 Height ---- //

Height is stored as a mesh of all vertices with 0,1 m resolution. 65536 values are possible. 0 being the lowest, 65535 being the highest.

*mental note - soft limit of around 1024 would probably have to be included.*

*idea: laying track/road/canal will automatically landscape to a slope chosen by the player (0%, 0.5%, 1% etc)*

*idea: erosion - from time to time the game will randomly modify certain vertices causing the terrain to change*

*idea: earthquake - random modification of vertices in a secluded territory destroying everything*

*idea: while building things such as railways automatic terra-forming would be used*

// ---- 2.3 Water ---- //

Up to height of 32 m all terrain is under water by default. If a terrain has connection to outside sea (or any other water) than it will be flooded by water.

*This one is very tricky - it can make the life of a map makers and (especially) coders a living hell.*

// ---- 2.4 Rivers ---- //

Certain tiles will use 'river' property - these tiles will check their surroundings and 'flood' them if possible. Once river has settled are tiles adjoining river are marked 'unmoveable' for landscape tools.

*mental note - ships can travel rivers if maximum slope is 0,1%.*

// ---- 2.5 Terrain Types ---- //

Terrain types define how expensive/cheap it is to clear/terra-form it and which buildings can be built on a tile of this type.

```
//-----//
//      Chapter 3 : deleted      //
//-----//
```

```
//-----//
//      Chapter 4 : Transporting General      //
//-----//
```

#### // ---- 4.1 Cargo Types ---- //

Cargo type sets which kind of a vehicle can carry given cargo. The Game may include such cargo types.

**Hopper** - used for cargoes that provide a shader for representation (for example coal or ore)

**Platform** - used for cargoes that that need a model on a carrier - like steel or machinery.

**Generic** - used for cargoes that are invisible in their carrier - like goods.

*And also probably 10000000 other types would have to be included if higher discrepancy of cargo needs to be achieved. Passengers for instance would need separate type.*

Vehicle definitions use this as a common denominator.

Each vehicle that has capacity must be assigned to one of these types *(or to a new type that can ofc be introduced)*.

Each cargo must provide data needed for the chosen type vehicle to work:

Open Hopper needs texture of how the cargo looks like.

Platform needs a model that is carried.

External vehicle types might need other things.

Cargo types for instance set fares for certain cargoes.

*Mental note - provide an example of a spreadsheet of data of vehicle type*

#### // ---- 4.2 Type Definition ---- //

**Name** - name of the cargo type

**Visuals** - What kind of visuals are used for the cargo on the carrier vehicle (values are: **model**, **shader**, **none**)

*Also here preferences of service type should be included - like - hopper cargoes like to go slow, but steady etc.*

#### // ---- 4.3 Cargo Definition ---- //

**Name** - name of the cargo

**Type** - name of a cargo type it belongs to

**startyear** - from which year cargoes of this type may start to appear (industries will start to pop up)

**endyear** - from which year cargoes of this type will start to cease to appear (industries will close down)

**Unit** - what is the unit of measurement (volume, weight, quantity, load)

**UnitWeight** - weight of the one unit of this cargo

**Manifest** - the name of the list (with data) of vehicles this cargo will provide (relevant only for trains) to transport itself. Empty for non-manifest cargo

*It works this way - the shipper gives you the car to deliver to the destination*

**Destined** - yes/no - does this cargo choose its own destination, or is it decided on industry level (used for passengers)

**Returnee** - Does this cargo create a returnee? If so - name of the returnee definition, also the origin will not produce another unit until it consumes a returnee.

**InterCity** - is this cargo required to travel to different city to be accepted (other city = consumer must be in another town's city limits)

**Group** - cargoes of the same group may be transported simultaneously in the same carrier (*so you can carry returnees, inter city passengers, commuters etc in the same car - on a multi-stop train*)

// ---- 4.4 Loading Rules ---- //

Refitting is generally impossible - each vehicle can carry all cargoes that are of its base type (eg - all cargoes that are carried in 'hopper' types can be carried in all vehicles of 'hopper' type).

Although - interface must provide an easy way of loading certain amount of cargo 'per vehicle'

Eg - if a station produces ore + coal + sand. All hopper type of cargo.

The interface must provide of ordering:

load 3 cars of coal, 5 cars of sand and 2 cars of ore

*idea: three modes should be available:*

- *load all cargoes fitting transport*
- *load certain cargoes with auto balanced amount*
- *load given amount*

*idea: leave when all/one cargo is loaded to capacity*

// ---- 4.5 Cargo Class - Raw materials ---- //

Raw materials are cargoes that are produced 'from nothing'. These include coal, ores or grain.

These cargoes are usually available in high amounts and require not-so-fast service.

*Note - this is arbitrary distinction - internally the game works solely on types and subtypes*

// ---- 4.6 Cargo Class - Semi finished products ---- //

This category includes cargoes that are accepted by industries and are produced by industries from raw materials (or optionally - from other semi finished goods). For example steel or paper are in this category. They are

typically in smaller amounts than raw materials. Excess SMP can be accepted by industrial buildings in a city.

*Note - this is arbitrary distinction - internally the game works solely on types and subtypes*

// ---- 4.7 Cargo Class - Goods ---- //

Goods are final stage of all industry chains. They are available in smaller amounts than semi finished products. They are accepted by distribution centres. Usually are produced in smallest amount of all industrial cargo.

*Note - this is arbitrary distinction - internally the game works solely on types and subtypes*

// ---- 4.8 Cargo Class - Manifest ---- //

Manifest cargo is miscellaneous cargoes (grouped under 'Manifest' type of cargo) produced by industrial buildings in a city. It has its own destinations - like passengers - and will follow it if possible. It is usually transported in boxcars and tankers.

*Note - this is arbitrary distinction - internally the game works solely on types and subtypes*

// ---- 4.9 Cargo Class - Express ---- //

Express cargo are cargoes that are transported in coaches/vans like mail or passengers (both commuters and inter-city). It has its own destinations and will follow them if possible.

*Note - this is arbitrary distinction - internally the game works solely on types and subtypes*

// ---- 4.10 Warehouses ---- //

Warehouses are buildings adjoining stations which store items meant for transport. Warehouses have a cargo type assigned to them (hopper, generic whatever) and a cargo designated to store - designation might be changed.

*Example - warehouse is of 'hopper' type and is designated for 'coal'*

Warehouse definition may include multiple cargo types of cargo to be stored.

Warehouse provides catchment area for the stations of all kinds.

All warehouses that are a part of a single station will share its storage space. Catchment area has strength that has diminishing returns - the further away, the less effective it is.

Warehouses MUST connect to the station they serve.

Warehouse definition includes:

**Model** - model(s) used for visuals  
**startyear** - year from which this WH is available  
**endyear** - last year of availability of this WH  
**cost** - cost of rising the WH

**type** - cargo types stored (like 'hopper', 'generic etc') up to 8 types  
**capacity** - maximum capacity in kg  
**price** - cost of building  
**raze** - cost of removing the WH  
**maintenance** - cost of monthly upkeep  
**range** - range of catchment area provided by this warehouse  
**transporttype** - is this warehouse for rail/air/road/ships (eg - certain warehouses are optimised for ships or road vehicles).

// ---- 4.11 Route Planning ---- //

**Concept 1** - route is assigned to the vehicle in a form of orders (like in TTD). Vehicle consists can be copied (instead of order lists - to save on clicking) when building.

**Concept 2** - route is planned as a set of orders and vehicles are bound to it (or bought with the route in mind).

**Concept 3** - route is planned like above, but with consists that are built along with it in a set of exact copies.

*Idea: vehicles that share their order lists (and are of the same type - eg rail only) could have their commands (stop, start, goto depot etc) shared. Maybe even automatic modification could be used eg: if one vehicle is replaced then all shared vehicles will be. If locomotive was added then all trains on the same order lists would have eventually a locomotive added.*

```
//-----//  
//          Chapter 5 : Game World          //  
//-----//
```

// ---- 5.1 Industries - basics ---- //

Industries are entities scattered along the map. Industries are composed of three elements:

Input storage  
Processing facility  
Output storage

**Input storage** - It is a building that warehouses all input cargoes before they are processed. If an industry accepts no cargo then input storages are not present. Station/Loading area will accept given cargo if it has input storage within its catchment area. Industry can randomly increase and decrease input storage by building/razing input warehouses.

**Output storage** - this is building that warehouses output cargoes of an industry. All industry activity will cease once all output storages full. A station will list industry products (for pick up by transports) if the output storage is within station's catchment area. Industry may chose to build extra storage space.

**Processing facility** - these facilities process input cargoes into output cargoes. Once per day they use their 'production function' to put output into output storage. Industry can increase production capacity/rate by building new processing facilities.

Production function is an algebraic expression which sets how much input cargoes are used and for what. This function will only be executed if there is sufficient amount of input cargoes in input storage and sufficient amount of free storage in output warehouse. Maximum inputs is 8, maximum outputs is 8.

*Example:*

*One coal mine shaft produces 5 tons of coal per day.*

*One steel mill 'processor' uses 3 tons of ore/scrap metal(processed) and 2 tons of coal to make 2 tons of steel and 3 tons of ash (waste).*

*One factory hall uses 4 tons of steel and 2 ton of plastics to make 4 tons of goods and 1 ton of waste and 1 ton of scrap metal.*

*One scrap yard uses 1 ton of scrap metal and produces 1 ton of scrap metal(processed)*

*Idea: some industries might be seasonal - like farms may produce more during summer and very little (if none) in winter*

// ---- 5.2 Industries - production chains ---- //

Production chain is an entity produced by connection between industries and their produced cargoes.

Production chain should be closed - eg. all cargoes circulating within should all have destinations and producers. Cargoes That do not have output/input industry will be accepted/produced by seaports/customs warehouses (default industry always present on map edges). Chains may produce manifest or goods. A Scenario maker may choose to override the default behaviour (automatic balancing of cargo) or/and designate certain ports/warehouses to produce certain cargoes.

*Mental note - provide an example of production chain*

// ---- 5.3 Industries - distribution centres ---- //

Distribution centres are special entities that receive cargo such as goods and distribute it among cities. They have storage for 'goods' and their 'production' rate is dependant on how much population they cover. This places are warehouses of several trade companies and alike.

*See - there is a distribution centre on the outskirts of a town servicing given town. No more trainloads of goods dropped off in the middle of the city*

// ---- 5.4 Industries - setting, removal, production level changes ---- //

Occasionally a new industry will pop up - It happens randomly although:

- if there is a gross oversupply of some goods then industries will either increase their capacity or new industry will pop up (also excess freight might travel to seaports where it conveniently disappears)
- if there is a gross under supply of some raw materials then new supply

industry (mine or farm or whatever) will show up (also - seaports might provide extra raw materials)

Industry will close if:

- randomly (once again - raw material industries can do this), after the production will drop below certain point
- it didn't produce its 6 month capacity in the last 3 years (read: if there was no supply (factories) or demand (all) for at least 6 month worth of production then the industry goes bankrupt  
for example: max production of a coal mine is 1000 tonnes per month - so the mine must produce at least 6000 tons of coal in the last 3 years to avoid closing)

*Mental note - some sort of protection from closing should be provided for the beginning part of the game - when players/AI cannot service all industries*

*Mental note - the counting is on the 'processor' level - if the processor is not utilised then it will be razed down. If it was the last processor then the industry will close.*

*Idea: a counter showing how much the industry produced in the last x years might come in handy*

*Idea: instead of removing the industry unused buildings may go into 'condemned' state and could be revived if someone pays \$\$\$.* Condemned building would disappear after 3-5 years.

*Idea: condemned buildings would gradually deteriorate costing more money to revive on each passing day. Cool model might change showing how the building rusts to the ground.*

Player may fund prospecting for a new industry - like looking for a place to build a farm or new coal seam. Also player can build processing/accepting industries and sponsor expansion of production/processing power/storage of existing industries.

Map designer may disallow building of certain industries in a given area, or force building certain industries in a given area.

*Meaning: map designer may state:  
coal mines cannot be built in "Western part"  
or  
coal mines can be built only in "Eastern Part"  
this will result in a fact, that the game will prospect for new industries only in certain parts of the map*

// ---- 5.5 Industries - Definitions ---- //

Industry definitions come in two parts, part 1 is "Industry" part, part 2 is "Models" part.

Industry - defines global industry parameters

Models - defines parameters that are on the level of buildings that make the industry

Industry part

**Name** - name of the industry  
**startyear** - year from which the industry will start to pop up  
**endyear** - year from which the industry will start to close  
**cost** - cost prospecting for the industry  
**special** - is this industry a seaport, customs warehouse, distribution centre (overrides all below)  
**product** - up to 8 cargoes this industry produces (inactive if an industry is a seaport or customs warehouse)  
**input** - up to 8 cargoes this industry needs to start producing (inactive if an industry is a seaport or customs warehouse)  
**density** - are the buildings of this industry built one next to another, or scattered around  
**roads** - are the buildings of this industry forced to adjoin a road (also built by the industry)  
**connected** - are the buildings of this industry connected via something (pipes, conveyor belts etc)

Models part:

**Name** - name of the building  
**Owner** - name of the industry that can build this building  
**startyear** - year from which this building will start to be built  
**endyear** - year from which this building will cease to be built  
**cost** - cost of funding this building  
**productionfunction** - an algebraic expression showing how and what is produced by this industry (8 expressions possible)  
**inputstorage** - capacity of input storage in this building - up to 8 for the all cargoes of the industry  
**outputstorage** - capacity of output storage in this building - up to 8 capacities for all products of this industry  
**connected** - does this building require connection via pipes/conveyor belts etc  
**range** - active only if an industry is a distribution centre - sets how big is the radius in which the centre will distribute cargo accepted)

// ---- 5.6 Seaports ---- //

Seaports are special industries that serve as safety valve for cargoes that are not produced or accepted, are in short supply or low demand. Also seaports provide certain amount of passengers and manifest cargo. Seaports are built on the coast near the edge of the map.

Default behaviour of seaports may be overridden by a scenario author - by either forcing them off, or designating certain seaports/customs warehouses to produce certain amount (automatic) of certain cargoes.

// ---- 5.7 Cities - general ---- //

Cities are special entities that are composed of houses, commercial buildings and industrial buildings. Cities house most of the world's population as well as a good share of small industries. Cities grow if they are properly serviced by transport.

// ---- 5.8 Cities Local Authorities ---- //

Each city has local authority. This governing body may forbid player company from building anything in the city authority limits and collects taxes for property on the authority limits.



There are two types of local authority limits. Canton limits and city limits. City limits constitute area taken by all buildings in the city + 2 tiles around them. Canton limits is all area around cities up to canton limit of another city. These limits are visible through LA interface.

Authority rating increases when player provides help for the city (lobbying :p) or bribes the authority. Player can also set permanent/periodic advertisement campaigns.

Authority rating decreases if the player went on destruction derby and destroyed too many buildings in the city.

#### // ---- 5.9 Cities - Business district ---- //

Business district is a district that mostly houses office buildings. These buildings accept mail, commuters and inter-city passengers and provide those cargoes (returnees in a case of passengers). Business district houses banks that provide valuables.

#### // ---- 5.10 Cities - Residential district ---- //

Residential district houses all city population. It produces commuters, mail and inter-city passengers and also accepts them. Residential districts also accept (through distribution centres) finished goods and water.

#### // ---- 5.11 Cities - Industrial district ---- //

Industrial district is a part of town that houses small industries. It produces and accepts manifest cargo, also produces and accepts commuters (returnees in a case of production).

#### // ---- 5.12 Returnees ---- //

Returnee is a passenger that was 'produced' by 'consuming' non-returnee passenger by a building that has no population (commercial/industrial).

- Returnee has the destination which was the origin of the passenger that created returnee
- Returnee has the same type as the passenger that created returnee (eg - commuter remains a commuter. Inter city remains inter-city).
- Passenger always creates a returnee when 'consumed' by commercial/industrial building.
- Passenger usually creates a returnee when 'consumed' residential building.
- Only residential buildings can create passengers - rest only creates returnees.

#### // ---- 5.13 Building Definition ---- //

Buildings are defined by such fields:

**startyear** - year of first presentation

**endyear** - obsolete year

**radiusmin** - minimum distance from the district centre the building will be built - in tiles

**radiusmax** - maximum distance from the district centre the building will be

**built** - in tiles  
**production** - production function  
**accepts** - names of the cargoes it accepts  
**inputstorage** - capacity of the input storage  
**outputstorage** - capacity of the output storage  
**raze** - cost of demolishing this building (-1 for not possible to raze)  
**type** - type of terrain this building can be built upon (special: ALL flag means all terrain types), up to 4 types.  
**district** - industrial, business, residential, none (none used for trees for instance).

```

//-----//
//      Chapter 6 : Trains      //
//-----//

```

## // ---- 6.1 Track - Definition ---- //

Track is the utility that allows trains to move (:D). Track is defined by several properties - which are:

**name** - name of the track  
**next** - name of the next track in the series

*(tracks are defined in series - these are used for track allocation for vehicles - they can ride on all tracks that are lower in the series of track definitions)*

*example:*  

<i>name</i>	<i>next</i>
<i>Track1</i>	<i>Track2</i>
<i>Track2</i>	<i>UltraTrack</i>
<i>UltraTrack</i>	<i>Turbotrack</i>
<i>TurboTrack</i>	

*Trains assigned to Track1 will be able to ride all tracks. Trains assigned to Track2 will be able to ride track2, ultratrack and turbotrack etc.)*

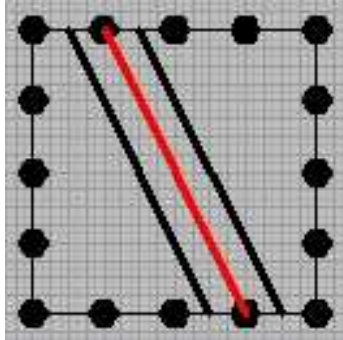
**speed** - maximum speed that can be attained on this track  
**price** - price that needs to be paid for 1 meter of single track  
**maintenance** - cost of monthly maintenance of 1m of track  
**raze** - cost of taking 1m of track off  
**age** - how many years it will take before the track starts to deteriorate (deterioration - causes max speed to drop up to 10 km/h after some time - around 5 years)  
**startyear** - year of presentation  
**endyear** - year of obsoletion  
**shader** - name of the shader script of this track  
**masts** - boolean, does it use electric masts for the catenary  
**incline** - maximum slope this track can be built on

*mental note - make a tool to distinguish track types - like colour clause, also showing track age/deterioration/max speed)*

*mental note - provide an example spreadsheet of track definition*

## // ----- 6.2 Track - Directions and Building ----- //

Internally The Game stores track on a single tile as a stretch between two points distributed along the edges of the tile - like in the picture below.



Thick red - track axis  
Thick black - track right of way  
Dots - tile border snap points

Building track is first done in 'survey' mode. All landscaping and track laying is first projected, and then it is approved (by player and local authorities in areas the track will run through) and construction starts

Track is laid using drag-and-drop method.

Curves are created by smoothing track elements using bezier curves or b-splines.

// ---- 6.3 Track - Grades ---- //

Grades are used in speed calculations. Height difference between two ends of a track (on a single, one square section of track) sets grade for speed calculations:

- 0.5 m rise - 1%
- 1 m rise - 2%
- 1.5 m rise - 3%
- 2m rise - 4%

The game uses realistic speed calculations - alike those from TTDpatch. These calculations are used for braking and accelerating.

*The formulas for tractive/braking effort will be provided in the time to come.*

// ---- 6.4 Track - Switches ---- //

Switches are places that allow trains to split a path. Speed is restricted depending on track and curvature involved. If a track is a switch then it causes extra maintenance - depending on a type of switch.

// ---- 6.5 Track - Curves ---- //

Curves are places where trains change direction. Curves are built automatically and smoothed by using bezier or b-spline curves (:D). Curves cause extra drag and impose speed limits - depending on curvature.

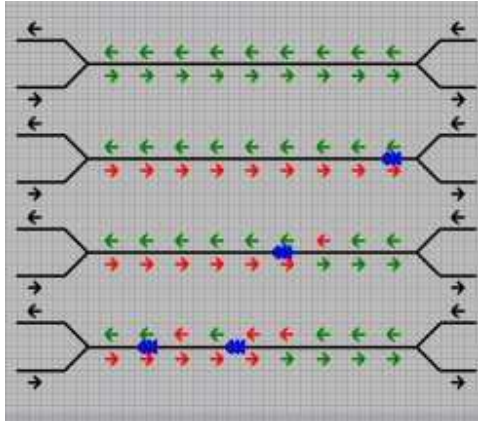
```
// ---- 6.6 Track - Signals ---- //
```

Signals are used to steer traffic.

There are two types of signals:

### **Two way signals.**

Signals allowing two way entry.



How does it work?

Arrow shows a signal for the traffic in the direction of the arrow. Two way signals will block traffic that is opposing current direction to the nearest junction. If there are two 'two way' stretches of track then trains will try to occupy the one that is free both ways (3-4 aspect signalling - trains will choose the lowest aspect available?)

### **One way signals.**

These signals allow entry only from the one direction. They work like 1-way signals in TTD.

Presignals - presignals are signals that protect a line before a train enters. If a train before presignal cannot proceed on one of the routes it wants (of which come out of the given junction) to use then it will get 'red' presignal.

Junctions - junctions are places where 3 or more tracks meet. Junction will always try to secure 'exit' - eg green path out of the junction. This will ensure that for instance 3 trains will not crowd 3 way junction by approaching from all directions.

Path Based Signalling - a train will enter a junction despite having 'stop' signal if it has a free path to its destination through the junction.

```
// ---- 6.7 Deleted ---- //  
// ---- 6.8 Stations - General ---- //
```

Stations are entities that load/unload and service trains. Stations are composed of: platforms, loading/unloading installations, workshops and warehouses.

// ---- 6.9 Stations - Platforms ---- //

Platforms are facilities that allow trains to load. Just like warehouse a platform services certain types of cargo. Platforms are built as a structure parallel to the track (using track Right-of-way as a reference). Platforms MUST be level.

Platform may be for loading or unloading only - or for both.

Platform is for a certain type and then specified for a certain cargo - just like warehouses.

// ---- 6.10 Stations - Movement ---- //

Trains need to change direction on a station (reverse that is).

Normally trains cannot 'warp' back like in TTD. The train will slowly go into 'reversing' mode in which leading locomotives are re-coupled to the forward and rest of the locomotives have reversed facing (if they are single direction - such as steam locos). The whole reversing takes around 2-5 seconds for each locomotive that needs to be re-coupled.

If there is a locomotive on both ends then the train does not need to reverse. Train also do not need to reverse if the whole consist is made of push-pull equipment and there is a cab-car or locomotive on both ends of the train.

If a train can continue without reversing then it will do so (so a RoRo setup is possible).

Ghost mode: To relieve traffic jam a train can be put into 'ghost mode' when it becomes translucent and other trains can pass through it at 20 km/h.

// ---- 6.11 Stations - Workshops ---- //

Workshops are special buildings which accompany stations. If a train needs a service and the station has the workshop, then the train will enter 'service' mode in which it will serviced.

If there is a need to change the consist of a train (like replace a locomotive, or cars) or there is a need to store the train then the train can enter the depot - where those operations are possible.

Workshops are built as a separate building and may store up to 3 times their length of trains. A workshop requires to have a connection to the station and a track(s) leading into it. Several workshops can be built next to each other creating a bigger entity.

Trains can be moved between two workshops even if workshops are not connected with a track. The movement speed is about 15-50 km/h depending on the year.

// ---- 6.12 Rolling Stock - Definitions ---- //

Vehicles are defined in two parts. First part is 'vehicle' part that stores data that is global for the vehicle (like price or running cost). Models part is the part that defines 'per vehicle' properties - like power or tractive

effort.

#### Vehicle part

**name** - text name of the vehicle  
**year** - Year of design  
**price** - base price  
**maintenance** - cost of yearly maintenance (fuel excluded).  
**fuel\*** - type of fuel used (coal, heavy oil, diesel, petrol, LPG, kerosene, electric (normal), electric (regenerative), fuel cell)  
**time** - time in seconds (real-time) it takes to deliver a ready vehicle from the manufacturer to the workshop it was ordered from  
**speed** - maximum speed of the vehicle  
**accelmodel\*\*** - model of acceleration for this vehicle (steam, diesel/electric, linear, none)  
**service cost** - cost of single service  
**service time** - time of servicing  
**track** - up to 8 types of track this vehicle can run on (and all of it's derivatives)  
**type** - type of this vehicle (hopper, platform, van, tanker, carriage, etc - type must be defined before)  
**locomotive** - boolean, is this vehicle a locomotive (a locomotive is required on the tip of the train to achieve max speed)  
**singledirection** - boolean, is this locomotive a single direction one (thus needing reversing when train facing changes)  
**pushpull** - boolean is this vehicle suitable for push-pull operation

#### Models part

**sndlight** - sound when vehicle is at or below 1/4 power (blank for vehicles without power)  
**sndmedium** - sound when vehicle is between 1/4 and 1/2 power (blank for vehicles without power)  
**sndheavy** - sound when vehicle is between 1/2 and 3/4 power (blank for vehicles without power)  
**sndfull** - sound when vehicle is between 3/4 and full power (blank for vehicles without power)  
**sndover** - sound when vehicle is above full power (blank for vehicles without power)  
**sndlowspeed** - sound when vehicle is running at low speed (0 for cars - only 'rail song' for instance, or sanding).  
**Loopsound** - sound that is supposed to be looped (like steam puffs - it is dependant on speed - the faster it goes - the faster the puffing - also length of this sound is used for determining steam puffs)  
**sndhorn** - sound of the whistle  
**power** - power (in kW) of the vehicle  
**capacity** - what amount of goods this vehicle can carry  
**weight** - mass associated with this model  
**next** - Next model in class (used for multi-part vehicles - none for single or last vehicle in a consist)  
**maxTE** - maximum tractive effort (0 = do not display, for wagons for instance)  
**maxBE** - maximum braking effort (0 = do not display, for wagons for instance)  
**reverse** - is the model reversed or normal (0/1)  
  
**\*fuel** - type of fuel used by the vehicle. Fuel prices vary during the game. Fuel used is proportional to power produced.

**\*\*accelmodel:** accelmodel sets type of the locomotive.

Steam: Power starts at minimum\*1.5 at 1 m/s to achieve max TE, then linearly grows to full value at 3/5 of max speed. Then it drops to 3/5 max value. When final speed is reached then power is cut to minimum value needed to keep that speed.

Diesel/Electric: Power is at the beginning throttled to minimum required for max TE until it reaches max. Then it is kept until max desired speed is achieved. Then The power is throttled to the minimum needed to keep that speed.

Linear: acceleration is linear to power.

None: this vehicle has no acceleration model and is idle (used for wagons).

// ---- 6.13 Rolling Stock - Ordering ---- //

Vehicles are ordered, delivered and modified in the Workshops (Depots).

```
//-----//  
//      Chapter 7 : Road Vehicles      //  
//-----//
```

// ---- 7.1 Roads - Definition ---- //

Road is a stretch of pavement that trucks ride on (*surprise!*).

Road(s) has the following properties:

**name** - name of the road  
**next** - name of the next road in the series  
**speed** - maximum speed that can be attained on this road  
**price** - price that needs to be paid for 1 meter of single road  
**maintenance** - cost of monthly maintenance of 1m of road  
**raze** - cost of taking 1m of road off  
**age** - how many years it will take before the road starts to deteriorate  
(deterioration - causes max speed to drop up to 10 km/h after some time - around 5 years)  
**startyear** - year of presentation  
**endyear** - year of discontinuation  
**shader** - name of the shader script of this road

// ---- 7.2 Roads - Directions and Building ---- //

Roads are internally saved the same way as track is.

Roads are built using the same principle as trackage -> first survey mode to plan all the building, then the works commence. The same drag-n-drop method is used.

Roads are by default double direction and require twice the Right-of-way the trains need

Single width ROW roads are possible, but they are always single direction ones (this way complex intersections are possible)

// ---- 7.3 Roads - Grades ---- //

Road vehicles use the same curve-to-rise as railroads.

// ---- 7.4 Roads - Intersections ---- //

Roads can intersect - these intersections can be even at 90 degree angle. Up to 6 pieces of tarmac can be laid on a single tile. Roundabout will be created if too many stripes of tarmac will be laid.

// ---- 7.5 Roads - Curves ---- //

Roads - like tracks - will be a subject of curve smoothing. Just like with tracks curves will restrict speed and cause extra drag.

// ---- 7.6 Roads - Motorways ---- //

Motorways are roads that have two lanes - each for different direction. Freeways are built like roads, but are more restricted:

motorways allow only two parts of the road on a single tile, but also cannot turn in a 90 degree angle.

Motorways cannot cross on a single level.

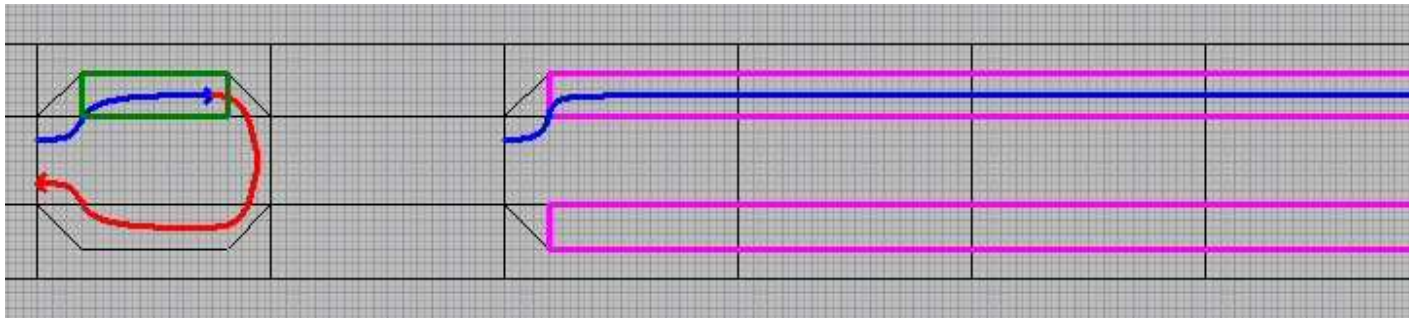
// ---- 7.7 Roads - State and Private Roads ---- //

Government will build roads that are for public use. Government roads impose 105 km/h speed limit for buses and 90 km/h speed limit for trucks. (120/100 for govt motorways)

Private roads are roads owned by transport companies - these roads have no speed limits beside the construction ones. Also private roads are usable only for trucks/buses owned by the owner of the road.

Owning a truck imposes certain amount of road tax - which is 5-250% (*random?*) for the fuel costs when vehicle travels on govt road.

// ---- 7.8 Stations - Loading bays ---- //



Loading Bays are structures that allow trucks to load (*surprise!*). They have a form of bays on the side of the road. If there is a platform on the one side of the bay then the vehicle may reverse. If not (loading platforms are on both sides) then the vehicle will continue.



Leftmost:

Single bay on a single tile.

- green - loading bay
- blue - empty vehicle path
- red - full vehicle path when reversing

Right:

Long 5 tile loading bay (automatically happens when several bays are built one after another. Can house several trucks, as well as long trucks) - two sided - thus no reversing.

- blue - path of a vehicle (assuming empty bay)
- Pink - loading areas.

Note - vehicles will use both sides to load, regardless of the direction they came from.

Loading bay is defined by these properties:

**Model** - model(s) used for visuals

**startyear** - year from which this bay is available

**endyear** - last year of availability of this bay

**price** - cost of rising the bay

**type** - cargo types supported (like 'hopper', 'generic etc') up to 8 types

**cost** - cost of monthly upkeep

**raze** - cost of removing the bay

**throughput** - the maximum rate of which a cargo can be loaded/unloaded - in tons per second (Realtime).

**load** - boolean, bay for loading

**unload** - boolean, bay for unloading

// ---- 7.9 Stations - Workshops ---- //

Workshops are special 'bays' that are built just like them. They house a full truck - or amount of trucks that a bay of that length would store.

// ---- 7.9 Vehicles - Definitions ---- //

Trucks are composed of these fields (as usual there is vehicle and models part)

Vehicle:

**name** - text name of the vehicle

**year** - year of design

**price** - base price

**maintenance** - cost of yearly maintenance (fuel excluded).

**fuel** - type of fuel used

**time** - time in seconds (real-time) it takes to deliver a ready vehicle from the manufacturer to the workshop it was ordered from

**speed** - maximum speed of the vehicle

**service cost** - cost of single service

**service time** - time of servicing

**road** - up to 8 types of roads this vehicle can run on  
**type** - type of this vehicle (hopper, platform, van, tanker, carriage, etc - type must be defined before)  
**vehicle** - is this vehicle: rig (single, self contained, may tow a trailer), tractor (carries semi-trailers), trailer or semi-trailer.

Models:

**sndlight** - sound when vehicle is at or below 1/4 power  
**sndmedium** - sound when vehicle is between 1/4 and 1/2 power  
**sndheavy** - sound when vehicle is between 1/2 and 3/4 power  
**sndfull** - sound when vehicle is between 3/4 and full power  
**sndover** - sound when vehicle is above full power  
**sndlowspeed** - sound when vehicle is running at low speed  
**sndhorn** - sound of the whistle  
**power** - power (in kW) of the vehicle  
**capacity** - what amount of goods this vehicle can carry  
**weight** - mass associated with this model  
**next** - Next model in class (used for multi-part vehicles - none for single or last vehicle in a consist)  
**maxTE** - maximum tractive effort (0 = do not display, for wagons for instance)  
**maxBE** - maximum braking effort (0 = do not display, for wagons for instance)

// ---- 7.10 Vehicles - Ordering ---- //

Vehicles are ordered from a warehouse, and they are serviced and sold there also.

// ---- 7.11 Vehicles - Special Consists ---- //



It is possible to attach several semi-trailers to a single tractor and create a road-train

Road trains have these limitations:

- They can only be used on private roads and, if scenario allows, outside city limits
- They are much more speed restricted on curves
- Maximum length
  - 56 m for state roads and 120m for private roads (if state roads allow RT)
  - 56m for private roads (if state roads disallow RT)

```
//-----//  
//      Chapter 8 : Aircraft      //  
//-----//
```

## // ---- 8.1 Airports ---- //

Airports are structures that service, load and depart aircraft. Airports are built by towns, but can be expanded by transport companies.

### // ----- 8.1.1 Runways ---- //

Runways are available in 3 sizes. Short (5 squares), Medium (10 squares) and Long (15 squares). Short runways service only small planes. Medium services small and medium planes and Long services all planes.

Runways are connected by one-way "roads" - taxiways - built by the player.

Runways are universal - for starting and landing.

- Plane will use a runway for taking off if can reach it by a taxiway and it is long enough for this plane.
- Plane will use a runway for landing only if there is a taxiway connected at the end (allowing to exit the runway) and the runway is long enough.

Rules of movement on the airport:

- Plane will land only if there is a free runway of proper length (with taxiway leading off)
- Plane will choose a free loading bay of those it can reach by the taxiways
- Plane will attempt to start only if there is a runway of proper length the plane can reach.
- Plane will dynamically try to choose free runway it can reach while attempting to take-off

### // ----- 8.1.2 Taxiways ---- //

Taxiways are one-way roads that connect various part of an airport. They are built just like roads, but follow the rule that only two 'strips' may be laid on a single tile. Taxiways are 15 m wide and take all tile.

### // ----- 8.1.3 Hangars ---- //

Hangars are connected to airports via taxiways. Aeroplanes are ordered (built) in hangars. Hangars also upgrade and service aeroplanes. Hangars include a helipad for helicopters.

### // ----- 8.1.4 Heliports ---- //

Heliports are pads that serve a purpose of a loading bay for helicopters. They are also "runways" for them. Helicopter will choose a free helipad of all available for it at a given airport.

Heliports can also be built within towns on a top of a commercial building (for a fee).

### // ----- 8.1.5 Terminals ---- //

Terminals are places where aeroplanes are loaded and unloaded.

Terminals consist of a few tiles of free space (for the aircraft to land) and - a building accompanying it. Overall a terminal may take several tiles - depending on size.

Terminal is defined by this properties:

**Model** - model(s) used for visuals  
**startyear** - year from which this terminal is available  
**endyear** - last year of availability of this terminal  
**price** - cost of rising the terminal  
**type** - cargo types supported (like 'hopper', 'generic etc') up to 8 types  
**cost** - cost of monthly upkeep  
**raze** - cost of removing the terminal  
**throughput** - the maximum rate of which a cargo can be loaded/unloaded - in tons per second (Realtime).  
**load** - boolean, bay for loading  
**unload** - boolean, bay for unloading  
**size** - planes of which size this terminal can service (small, medium, large)

// ---- 8.2 Aircraft ---- //

Aircraft are entities that make the air transport possible. They are described by such properties:

**name** - text name of the vehicle  
**startyear** - Year of design  
**endyear** - year of discontinuation  
**price** - base price  
**time** - time in seconds (real-time) it takes to deliver a ready vehicle from the manufacturer to the hangar it was ordered from  
**speed Max** - Maximum possible speed  
**thrust** - in kg used in speed/acceleration calculations  
**maintenance** - cost of a single service at depot  
**fuel** - type of fuel used  
Capacity - what amount of goods this aircraft can carry  
**Type** - type of this aircraft (hopper, platform, van, tanker, carriage)  
**vtol** - does this vehicle have vertical landing/take off  
**size** - specifies which runway length or terminal type this aircraft requires  
**weight** - weight of the aircraft  
**model** - model used for visuals  
**capacity** - capacity in tonnes

sounds

**idle** - aircraft does nothing  
**on ground** - aircraft is on a taxiway heading to takeoff  
**takeoff** - aircraft is on the runway starting  
**landing** - aircraft is landing  
**loop** - looping sound - like propellers or somesuch.

//-----//  
// Chapter 9 : Marinal Transport //  
//-----//

// ---- 9.1 Infrastructure ---- //



*I was thinking, that only inland water transport should be included – such as tugs*

*I don't think that TE is big enough to accommodate full blown ocean liners...*

// ----- 9.1.1 Wharves ---- //

Wharf is a "platform" built parallel to the coastline next to which ships load/unload. They must be as long or longer than the ship they are supposed to service.

// ----- 9.1.2 Docks ---- //

Docks are structures that build and repair ships. they have to be long enough to service a single ship and can hold only one at a time.

// ----- 9.1.3 Channels ---- //

Channels are structures that allow ships to travel through the map where there is no water. Channel must be level. To increase the elevation Sluices must be used. Sluice must be as wide and long as the serviced ship.

// ----- 9.1.4 Traffic organisation ---- //

Water Traffic has got to be organised by setting 'corridors' in the port. Corridors are one-way. Properly set up port must contain: one entry, one exit and one wharf.

// ---- 9.2 Tugs ---- //

Tugs are ships that push/pull a string of barges. They are defined by such properties:

**name** - text name of the tug  
**startyear** - Year of design  
**endyear** - year of discontinuation  
**price** - base price  
**time** - time in seconds (real-time) it takes to deliver a ready vehicle from the manufacturer to the dock it was ordered from  
**speed Max** - Maximum possible speed  
**power** - maximum power of the engine  
**maintenance** - cost of a single service at a dock  
**fuel** - type of fuel used  
**weight** - weight of the aircraft  
**model** - model used for visuals

// ---- 9.3 Barges ---- //

Barges are parts of the system that carries cargoes. They are defined by:

**name** - text name of the barge  
**startyear** - Year of design  
**endyear** - year of discontinuation  
**price** - base price  
**time** - time in seconds (real-time) it takes to deliver a ready barge from the manufacturer to the dock it was ordered from  
**maintenance** - cost of a single service at a dock  
**type** - type of the vehicle (hopper etc)  
**capacity** - capacity in tonnes  
**fuel** - type of fuel used  
**weight** - weight of the aircraft  
**model** - model used for visuals

// ---- 4.4 Ships ---- //

Ships are self contained carriers. They are defined by such data:

**name** - text name of the ship  
**startyear** - Year of design  
**endyear** - year of discontinuation  
**price** - base price  
**time** - time in seconds (real-time) it takes to deliver a ready vehicle from the manufacturer to the dock it was ordered from  
**speed Max** - Maximum possible speed  
**power** - maximum power of the engine  
**type** - type of the vehicle (hopper etc)  
**capacity** - capacity in tonnes  
**maintenance** - cost of a single service at a dock  
**fuel** - type of fuel used  
**weight** - weight of the aircraft  
**model** - model used for visuals

```
//-----//  
//      Chapter 10 : Engineering      //  
//-----//
```

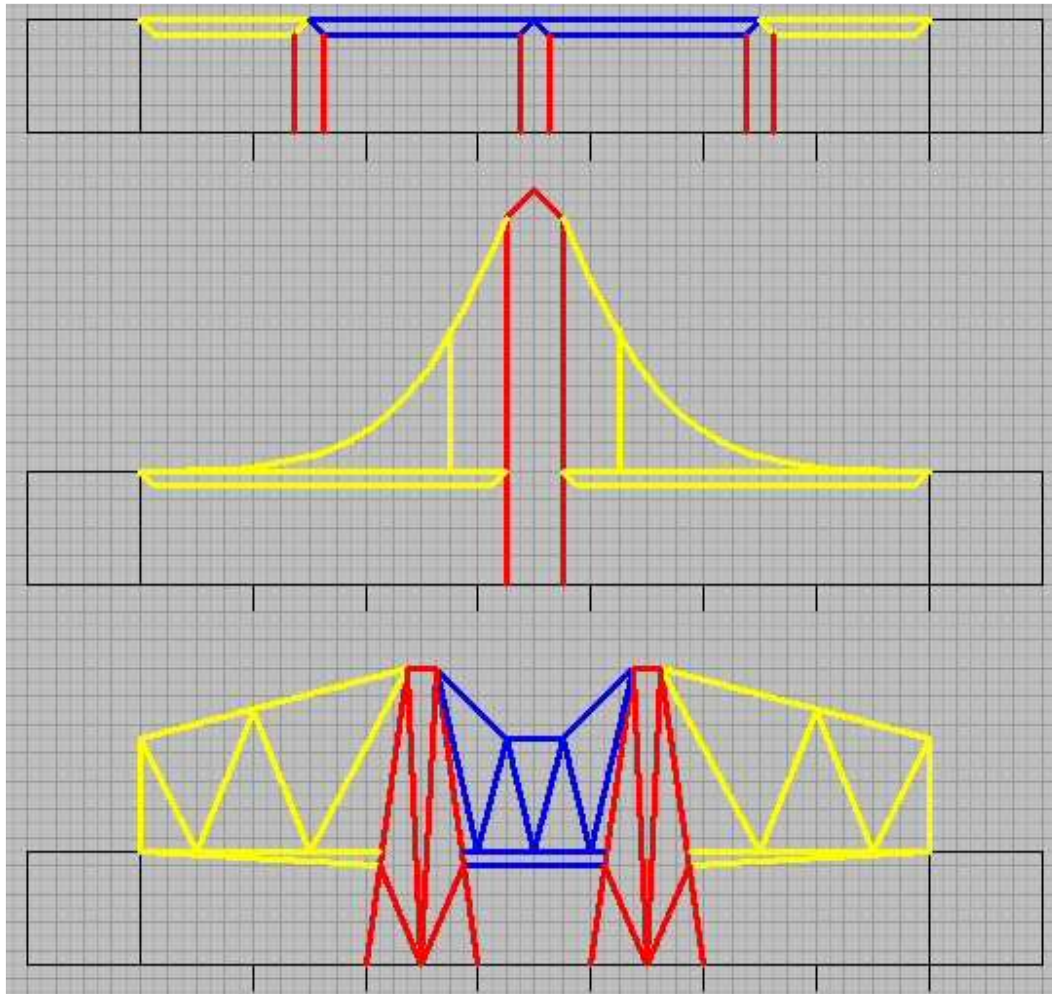
// ---- 10.1 Bridges ---- //

Bridges are structures used for crossing rivers and areas that are impossible to lay track on. Bridges begin with heads which take track to minimum depth this bridge can be built over. Heads are block of concrete/brick.

Heads are default for every bridge. Track or road is what player will lay on the head top.

Bridges are made of piers and spans.

There are two types of a span - first span and middle span. First span connects piers to the heads. Middle spans connect piers themselves.



The scheme is:

Yellow - 1'st span

Red - pier

Blue - middle span

Definitions are

Bridge as a whole:

**name** - text name of the bridge

**startyear** - year of first availability

**endyear** - year of discontinuation

**lengthmin** - minimum length of the bridge

**lengthmax** - maximum length of the bridge

**width** - width of the bridge (in tiles) (half and quarter tile also possible)

**maxspeed** - maximum speed attainable on this bridge

first span definition

**length** - length of the span (in tiles)

**cost** - cost of building the span

**maintenance** - cost of maintaining the span for a year

**raze** - cost of removing the span  
**time** - time of building the span, in seconds real-time

middle span definition is the same

pier definition includes

**cost** - cost of building the pier  
**maintenance** - cost of maintaining the pier for a year  
**raze** - cost of removing the pier  
**time** - time of building the pier, in seconds real-time

Bridges are built symmetrically - piers and spans are mirrored from both directions (eg the same pier or span is used on both sides).

// ---- 10.2 Tunnels ----- //

Tunnels are an effect of auto-landscaping. If the landscape exceeds certain amount the game will enter a tunnel and also will enter 'underground' mode. Then the player may commence building (all structures and tiles below above certain height will cease to be shown, also 'hidden' tiles are showed for those that will cause exiting underground mode, and those that will allow further building of the tunnel).

All structures that can be built over the ground, can be built underground (that is stations, etc) but one thing to note - catchment area is also limited vertically.

// ---- 10.3 Elevations ---- //

Elevations are 'negative' tunnels. They allow to rise track/road without landscaping. Elevations allow twisting, turning etc. Maximum elevation height is around 4-5 metres above the ground.

Elevation has either a form of a brick support (no speed limit), a wooden trestle (cheap, but limited speed) or a steel trestle (speed limited, but not severely, cheaper than brick, more expensive than wood).

```
//-----//  
//      Chapter 11 : Vehicles Misc.      //  
//-----//
```

// ---- 11.1 Reliability ---- //

Each vehicle has a 'reliability' rating. It corresponds to the probability of a breakdown.

Reliability = 100 points when vehicle has been freshly serviced. It will drop slowly to 0.

It will drop:

- 1 per 10 days when vehicle is idling (stopped or loading)
- 1 per 5 days when vehicle is working in normal specs
- 1 per 1 day when vehicle was overstrained (for instance there was more TE required of a locomotive than the locomotive can handle)



Each service reduces max-reliability by 1, but no more than 4 points per year and minimum 1 per year.

Below 50 a real possibility of a breakdown starts to kick in.

At 51+ reliability there is 0.25% chance for a breakdown

At 50 reliability there is 2% chance for breakdown

At 25 reliability there is 50% chance for breakdown

At 1 reliability there is 98% chance for breakdown

*Idea: Max reliability may be randomised - ranging from 75 to 125*

*Idea: New models may have reduced max reliability which will rise in the first 2-3 years*

Service of a vehicle takes 2 days game time.

// ---- 11.2 Life-span ---- //

Vehicles don't have 'life' but they are limited by dropping maximum reliability (thus increasing chance for a breakdown).

100 reliability will give approx. 20 years of useful (breakdown free) vehicle life.

// ---- 11.3 Wrecks ---- //

Wrecks happen when vehicles when they collide or fall off the track or road.

Trains:

Failed signals - randomly a signal may malfunction causing a train to override it and possibly crash onto another train.

Derailment, Catastrophic - locomotive derails, mayhem ensues, destroying loco and random amount of wagons (rest will appear at the nearest depot, cargo remains on train, passengers are lost)

Road Vehicles:

Train wreck - a vehicle is destroyed by a train on a level crossing. Road vehicle is destroyed, loco loses 10 reliability points. Train must stop for 1 game day.

Random crash - one vehicle loses traction and falls off the road, possibly crashing into another vehicle

Aircraft:

Fatal Breakdown/Bad dispatching etc - plane makes 'kaboom' and drops to the ground (probably destroying buildings below)

Ships:

Ship catches fire, or is pirated or whatever and sinks.

//-----//  
// Chapter 12 : World Misc. //

//-----//

#### // ---- 12.1 Stock Market ---- //

Just like in RT2/RT3 the player (the chairman) is a separate entity from the company and amasses his own cash.

The player starts the company with 10-20% investment in the company and the rest is 'outside' investment.

Each company has about 40%-60% of its stock on the market. Only this stock is available for purchase.

Stock price is determined by company value. If a stock goes above \$100 per share then a split 2 for 1 happens (number of all stock doubles, its price drops to \$50).

Stock price goes up when somebody (player, other players, AI) buys them. Stock price drops when somebody sells it. These are temporary changes and the stock will return to normal value after some time.

Buying on margin. It is buying on 'credit' - eg above current cash level of the buyer. Money is borrowed to buy stock in hope for it to rise - and thus make some profit

Selling Short - reverse of above, borrowing stocks to sell them in hope to drop and thus make profit on this.

In both cases a margin call might happen - when a broker requests either buying back short-sold stock or selling margin bought stock.

#### // ---- 12.2 Bonds ---- //

Bonds are a form of borrowing money for future profits. Depending on bond rating (from C - lowest to AAA - highest) are differently yielded.

C - 21%  
CC - 19%  
CCC - 17%  
B - 15%  
BB - 13%  
BBB - 11%  
A - 9%  
AA - 7%  
AAA - 5%

Depending how high is the rating the bonds may attract more investors.

Bonds are usually for 25 years, but player can set any amount. Also the sum for which the bond is taken is set by the player

Bond may be repaid at any moment with +10% penalty which is not considered if the bond is repaid when it matures.

#### // ---- 12.3 Loan ---- //

Loan is a form of debt when a company takes a sum of money (no more than 75%

of company value) and then repays it in regular intervals until all is repaid.

// ---- 12.4 Cargo Distribution ---- //

Industries have a pre-set group of receivers.

In the beginning of the game receivers are close to the source, but during the play there will be shift towards longer distances.

Receivers list is not static - each change will be reported in the newspaper in 6 month notice.

// ---- 12.5 Fares ---- //

Fares for transporting are set per type. Eg. All hopper type cargo are \$0.43 per tile-tonne.

Player can set percentage from which it will deviate for a single cargo (like coal or ore). Lowering fares will attract more cargo to the station (in a case of competition with other carriers). Making it higher will make reverse effect.

If there is a more then 75% difference between the lowest offer and the current offer then the current is discarded (station rating = 0%)

*Maybe fares could be controlled on station/single industry level to actively compete for cargo on station level.*

// ---- 12.6 Ratings - station ---- //

Station ratings define how much cargo is transferred to a given station. Station ratings may depend on:

- Vehicle reliability (passengers may spot junky car and refuse to ride it)
- Vehicle age (likewise)
- Mode of average speed in last 10 rides
- Frequency of service
- Amount of cargo stored on station (perishables for instance)

Overall - ratings show how efficient is the station at luring cargo to it.

Station rating of 39% shows that the station lures 39% of cargo produced by an industry. Ratings are per cargo (Eg. 25% of iron ore etc)

Which of above rules is applied is set through types (eg hopper: freq of service & amount of cargo)

// ---- 12.7 Ratings - industry ---- //

Industry ratings show how much of the total possible production is actually produced. This production is proportionally spread among stations.

Amount of produced cargo is calculated on efficiency of transport around the industry.

// ---- 12.8 Ratings - city ---- //

City ratings show how much the local authority likes/dislikes the current

transport corporation.

If the ratings are not high enough then the authority might not allow the corporation to do certain things (like build track/stations etc) within city or district limits.

City ratings are dependant on efficiency of commuter, mail and inter-city traffic in the city.

If there are no other companies moving stuff (passengers, mail) within city limits then starting ratings are maximum. If there are other corporations then drop to 50% of maximum ratings.

// ---- 12.9 Scenarios - plot side of the game, goals ---- //

Scenarios are maps with certain goals in mind. These are one (or few) of the following (and a few other may be added ofc):

Achieve certain company value (CNW - company net worth)

Achieve certain personal cash (PNW - personal net worth)

Move A tons of cargo B from C to D

Hold control X% of transport of commodity Y for at least Z years

Be one of X top performing companies

Also - during the game (at date or because other dialog box was selected). These dialog boxes allow to modify certain game data like running costs or company overhead.

Once the objectives are completed the player will be graded and will be presented a chance to either continue playing or finish the game.

// ---- 12.10 High Scores ---- //

High Scores are kept separately for each scenario and difficulty. Only top 25 names are displayed. Players can merge their high score lists.

// ---- 12.11 Game Setting ---- //

A 'setting' is a set of configuration files that provides certain elements to the game. A 'setting' may contain land tiles, vehicles, buildings etc.

A 'setting' is complete if it contains all data the game needs to run (vehicles, carriers, buildings, road and track, land tile definitions etc). A setting is open if it only contains certain parts (like only buildings or only tiles).

*A complete setting is an equivalent of a 'climate' in TTD. An open 'setting' is more of a vehicle set - like a grf Usset or such.*

// ---- 12.12 Weather & Disasters ---- //

The map and certain aspects of the game can be randomly modified by external sources. These include natural disasters and influences.

A list (not complete) looks somewhat like this

- Rain - a beautiful rain - an eye candy
- Storm - possibility for a lightning to strike a land based object and cause

- fire (destroying a building or forests)
- Heavy rain - causing mud slide and small terrain deformations
- Earthquake - random, major changes of the terrain along a "fault" (the game will locate faults on map generation). Destruction and mayhem ensues
- Tidal wave - from a certain direction (from map edge) certain shore tiles exposed to that direction are destroyed, sunk or risen.
- Snow - fresh snow (despite eye candy) will cause the vehicles to lose grip

// ---- 12.13 Pollution ---- //

Pollution is a stat that may slowly build up on certain tiles of the map (industrial zones, industries, rail/road yards, roads, tracks etc). Nobody actually know WTF it is supposed to do in a transport game, but alas it was included in the FRD

// ---- 12.14 Tourist attractions ---- //

Tourist attractions are special buildings that appear for no apparent reason and purpose to piss off the DD writer.

// ---- 12.15 Difficulty settings ---- //

The player has control over what difficulty the game will be by setting several options that include (incomplete list)

- amount of water/hills
- financial aspects (max loan, bond percentages, presence or absence of stock market etc)
- maximum money to borrow/start
- local authority "friendliness"
- certain vehicle performance aspects
- other

```
//-----//
//      Chapter 13 : Interface      //
//-----//
```

Interface of the game includes such properties:

- All player buildings are built first in survey mode (planning), then the player acquires land, and latest construction ensues.

```
//-----//
//      Chapter 14 : Wild Idea Space      //
//-----//
```

- waypoints - each and every vehicle can have a waypoint on its route, despite how strong path-finding is.
- Ability to import a manager photo and company logo
- Subsidies - randomly chosen multiplication for income for given route, also for limited amount of time
- Speed limits on the road.
- Each town will have its own automatic policeforce, firefighters and ambulance team, who will interfere if road vehicles break speed limits, or when accidents occur.

- Heavy load traffic (loud noise) will decrease company rating (or make nearby houses disappear) unless sound-walls are built.
- If a player owns an industry, they can set the amount of productions for different types of end products.
- Logging industries own parts of the forests they cut down. Rating of the transport company varies with the amount of pollution the industry produces
- Roads have civil traffic near and between towns. In case of heavy traffic, this might hinder a company's road vehicles. (Road) junctions should have an explicit rule about who may go first. This can be done by giving right-of-way to traffic from the right or putting traffic lights at the junction. City's will automatically place traffic lights at 'big' junctions. On ramps and expressways can be built to enter highways.
- Frequent flyer miles: This extension will make an airport attract more passengers, but the profit per passenger will drop a little bit. This extension should be interesting for larger towns.
- Installing a depot at the bottom of a steep mountain (or hill), with a couple of locomotives inside, will help passing trains, by automatically letting the extra locomotives help get the train to the top.
- Locomotives are a separate vehicles that can pick up bundles of wagons for example at a farm, and drop them then at the factory, where the locomotive then picks up some goods wagons to bring them to a city.
- Multiple Units (MU's) should be able to combine with other MU's (of the same type) but should also be able to split again. This splitting should only happen when traffic is low.