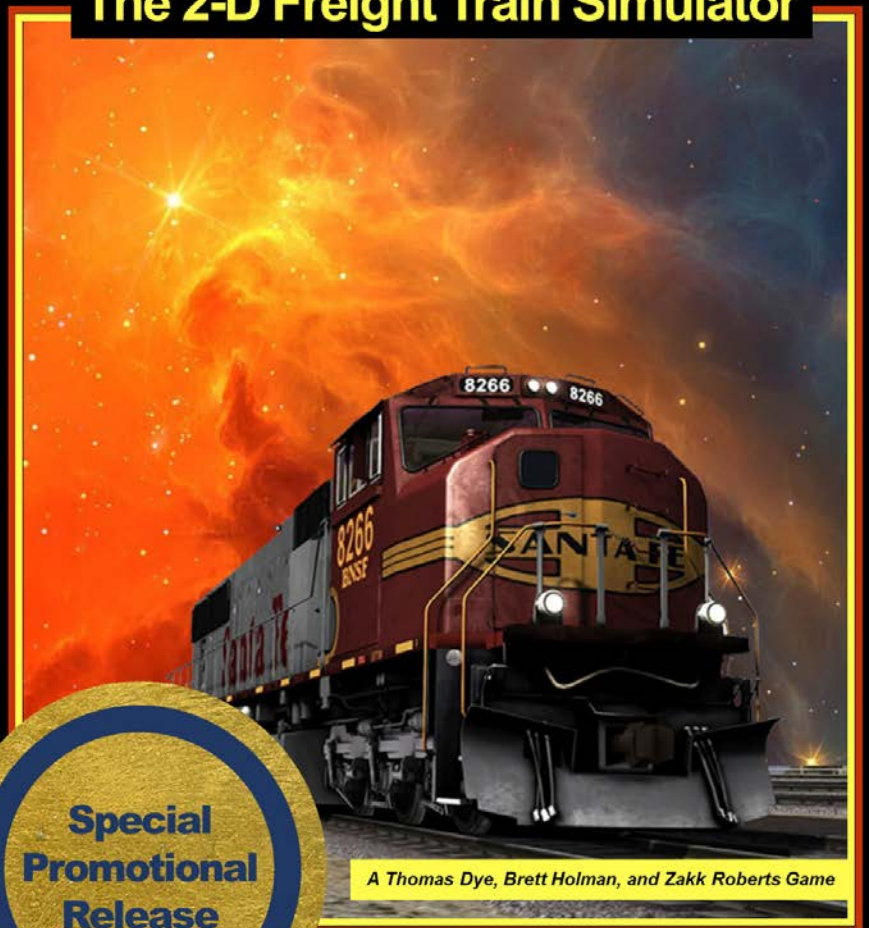
FREIGHT COMMANDER

The 2-D Freight Train Simulator



Special Promotional Release

Hip Replacements

English Version contains: -
Carte de reference et guide d'installation en Francais

PROJECT OVERVIEW

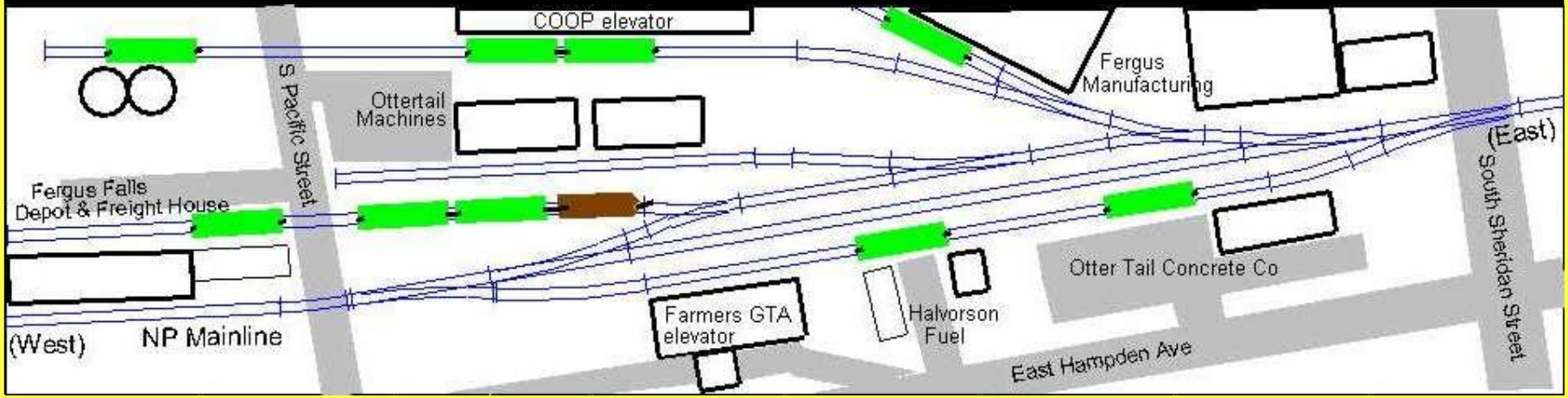
Model railroading is more than just running a train around the Christmas tree.

Realistic scenery model railroad layouts can be huge!

PROJECT OVERVIEW

Freight train operations are a great way to interact with a layout as a linear puzzle game.

The goal of the game is to take trains to local industries to pick up and deliver goods.



A FREIGHT CAR'S LIFECYCLE

1. Shipping orders are received at a yard for a product delivery.
2. Waybills are used to associate freight cars to shipping orders.
3. Trains are blocked together in "consists" of freight cars.
4. A first player crew drives an empty train to industries and drops off cars to be loaded.
5. A second player crew picks up cars and moves them to other industries to be unloaded.
6. A third player crew collects empty cars at industries and returns them to the yard.

WHY A SIMULATION?

As a formal game, all these steps are hard to track manually.

Fortunately, we know SQL!

```
#GetNextDestination
DROP FUNCTION IF EXISTS GetNextDestination;
DELIMITER $$
CREATE FUNCTION GetNextDestination(
    ShippingOrder INT)
RETURNS VARCHAR(255)
DETERMINISTIC
BEGIN
    DECLARE Destination VARCHAR(255);
    CASE
        WHEN ShippingOrder IN (SELECT ShipmentID FROM ShipmentsPickedUp WHERE Shipment
            AND ShippingOrder IN (SELECT ShipmentID FROM ShipmentsDelivered WHERE Ship
            THEN
                SET Destination = (SELECT ReturnToYard FROM Waybills WHERE UsingShipme
        WHEN ShippingOrder NOT IN (SELECT ShipmentID FROM ShipmentsPickedUp WHERE Ship
            THEN
                SET Destination = (SELECT FromIndustry FROM Shipments WHERE ShipmentID
        WHEN ShippingOrder IN (SELECT ShipmentID FROM ShipmentsPickedUp WHERE Shipment
            THEN
                SET Destination = (SELECT ToIndustry FROM Shipments WHERE ShipmentID =
    END CASE;
```

RELATIONAL MODEL

27 different relations, tracking:

- Modules
- Industries
- Rolling Stock
- Product Type Associations
- Shipments
- Crews
- Trains

ENTITIES

Similar entities need to be grouped together.

NORMALIZATION

From conception to delivery, the database has maintained a similar design.

Minor 2NF and 3NF improvements:

- Split Modules table into Modules and ModulesAvailable.
- Split Industries table into Industries, IndustriesAvailable, and IndustryActivities.
- Split IndustrySidings table into IndustrySidings and SidingsAvailableLength.
- Split Trains table into Trains and TrainLocations.
- Split Shipments table into Shipments, ShipmentsPickedUp, and ShipmentsDelivered.

PROJECT EVALUATION

Good design early on prevents major reworking.

Real data allows for realistic testing.

Stretch goals align with core functionality without major modification.

DEMO

Having a database is great, but how do I play the game?