Train Movement

Current

If there exist a path for one train the algorithm first checks if the path is valid, meaning all tracks are connected with each other and in the right order. If it concludes that the path is valid it sets all points that connect the tracks on the path to the correct status (straight or branch) and powers all tracks to full speed in the appropriate direction.Once the train reaches the beginning of the destination track the power of the last and second to last track gets halved. Once the train reaches the end of the destination track the power to this track is cut.

Planned

Instead of powering the tracks directly we abstract and set the speed of a train instead. For this to work we need a general train class and an instance of it for each train.

This class shall contain the status of the train (parking, halting, waiting or moving) as well as the tracks occupied by said train. A train can occupy up to three tracks:

- If it is parking, halting or waiting it occupies only the track it sits on. (1 track)
- If a train is moving and reaches the end of the current track it occupies this and the next track until it reaches the end of said track. (2 tracks)
    - if the next track has no signals on it the train occupies this and also the following track (3 tracks)

Now every time a contact on the track changes state the environment updates all train instances so that they in return may update the trains position and change the power state of some tracks.

> ⚠ This has as prerequisite that a contact will always change state when a train drives over it.