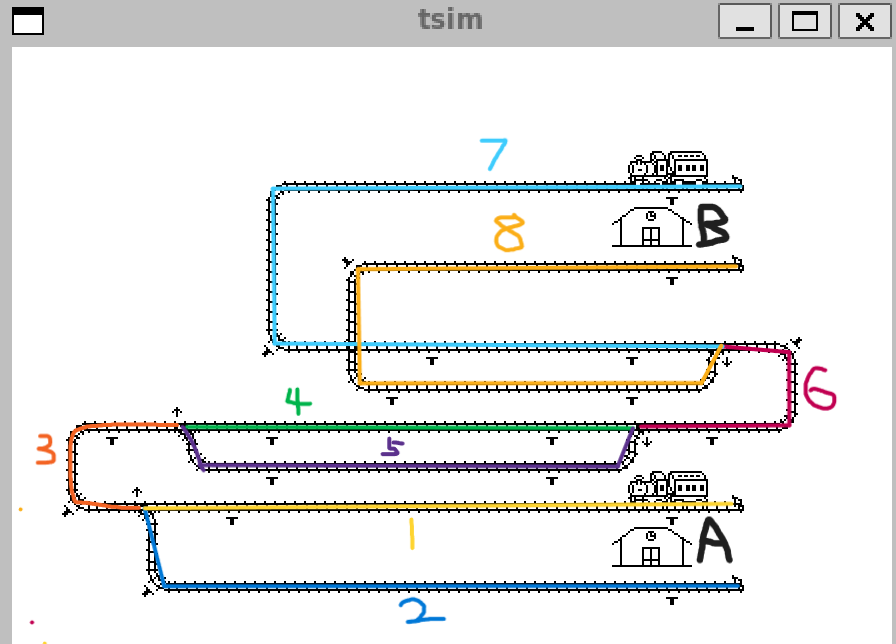
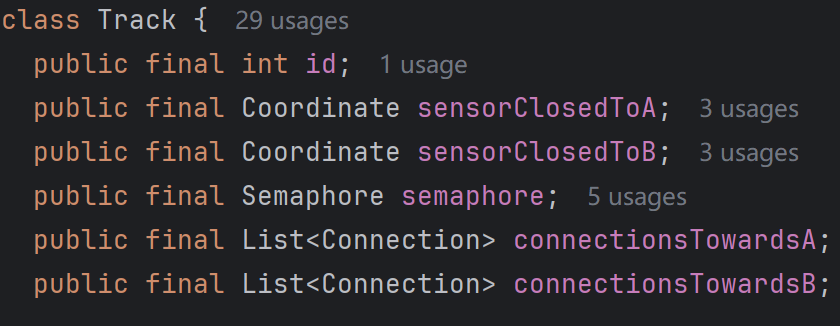
**Map design:**  
**Track:**



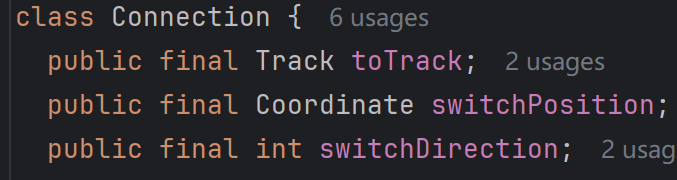


The map is divided into 8 tracks by switch.

Each track has two sensors, one near station A and the other near station B.

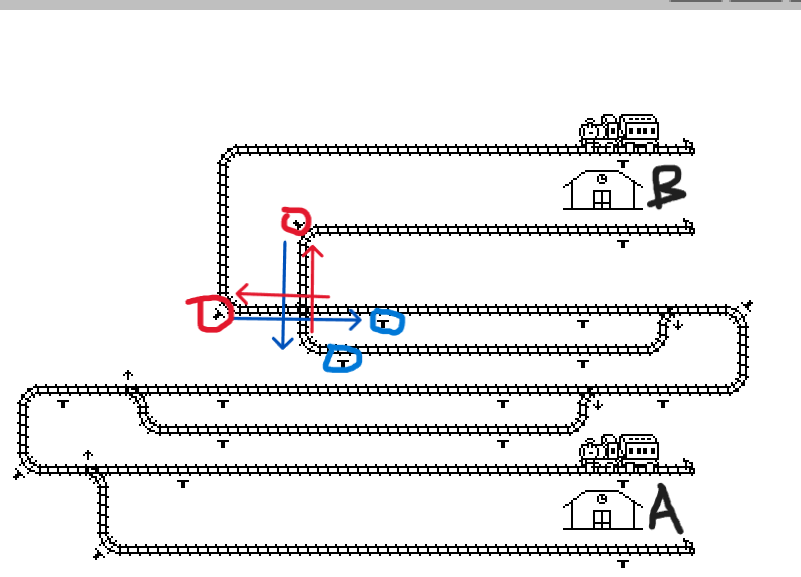
Each track has a list of connections in each direction, which stores all possible next paths in the corresponding direction. Take track 3 as an example, track 4 and track 5 are connected in direction B, and track 1 and track 2 are connected in direction A

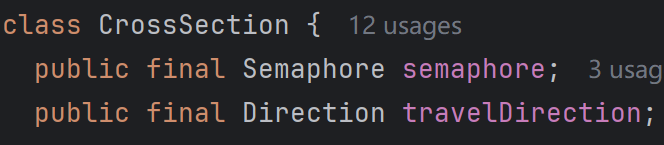
**Connection:**



Each Connection records the next track to be connected, the switch position, and the direction the switch is changed

**Crossroads:**

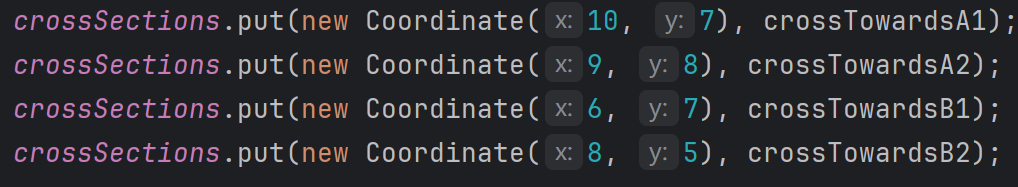




Two red arrows indicate sections in direction B, two blue arrows indicate sections in direction A, and four sections share a semaphore.

Each section has a one-to-one mapping sensor at the end, and we define a data structure to describe the mapping relationship between the sensor and the section:



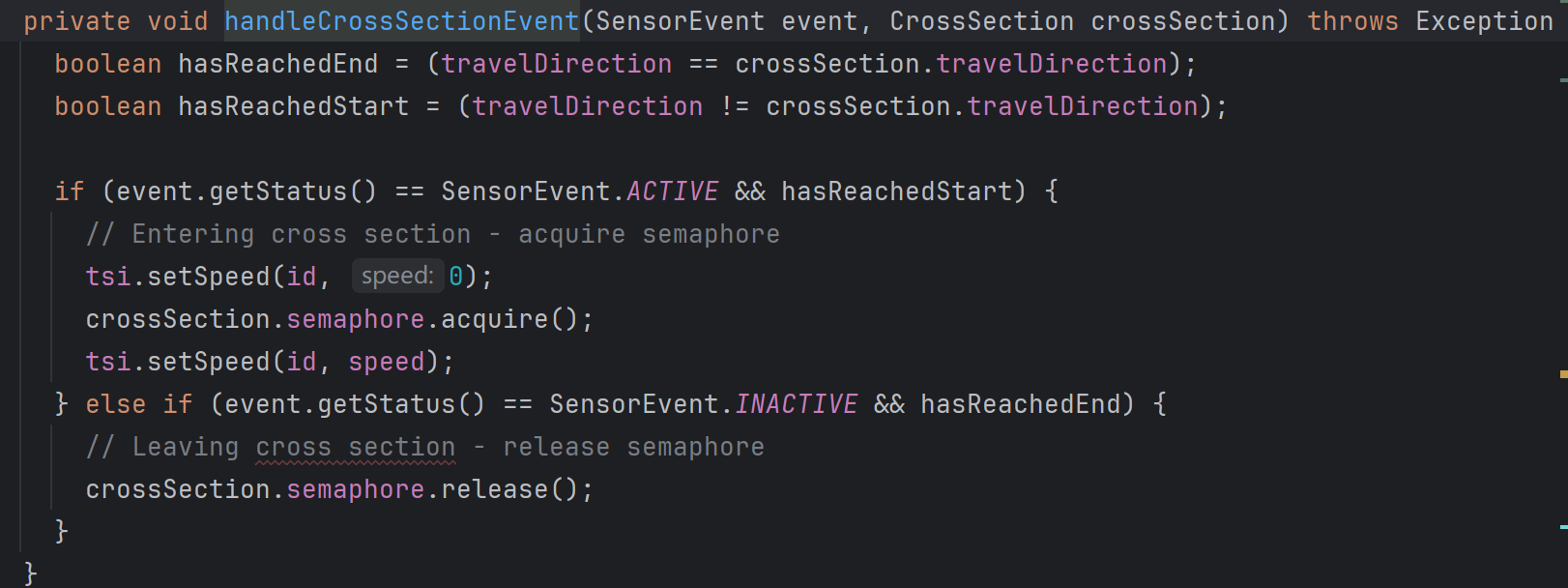


**Train behavior:**

During the train running, a SensorEvent will be triggered, from which we can obtain the sensor coordinates, we need to determine whether the sensor is in the Cross section or the normal track, the two have different processing logic.

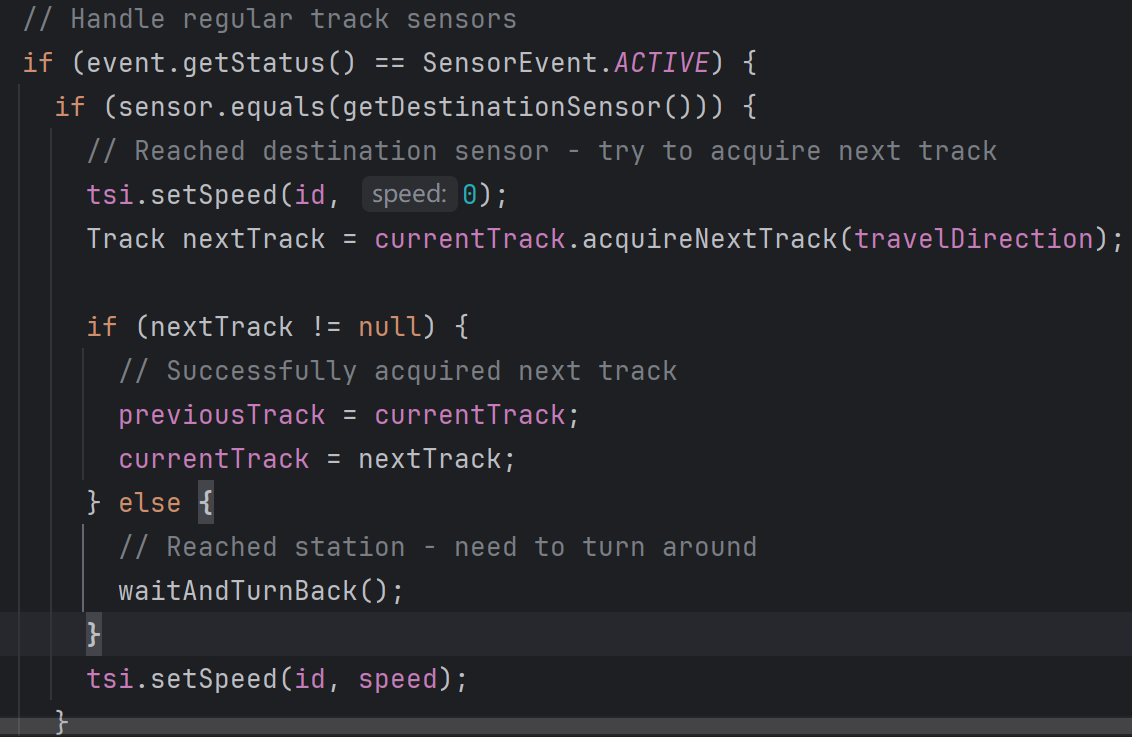
**Cross Section:**

As mentioned earlier, each section has a corresponding sensor, and if we obtain the sensor coordinates, we can find the corresponding section according to the mapping relationship, and then take out the direction of the section. If the train is traveling in a different direction than the direction of the section, it means that the train has just arrived at the intersection and needs to obtain the semaphore at the intersection to continue moving forward. If the two are in the same direction, it means that the train has left the intersection and needs to be released.

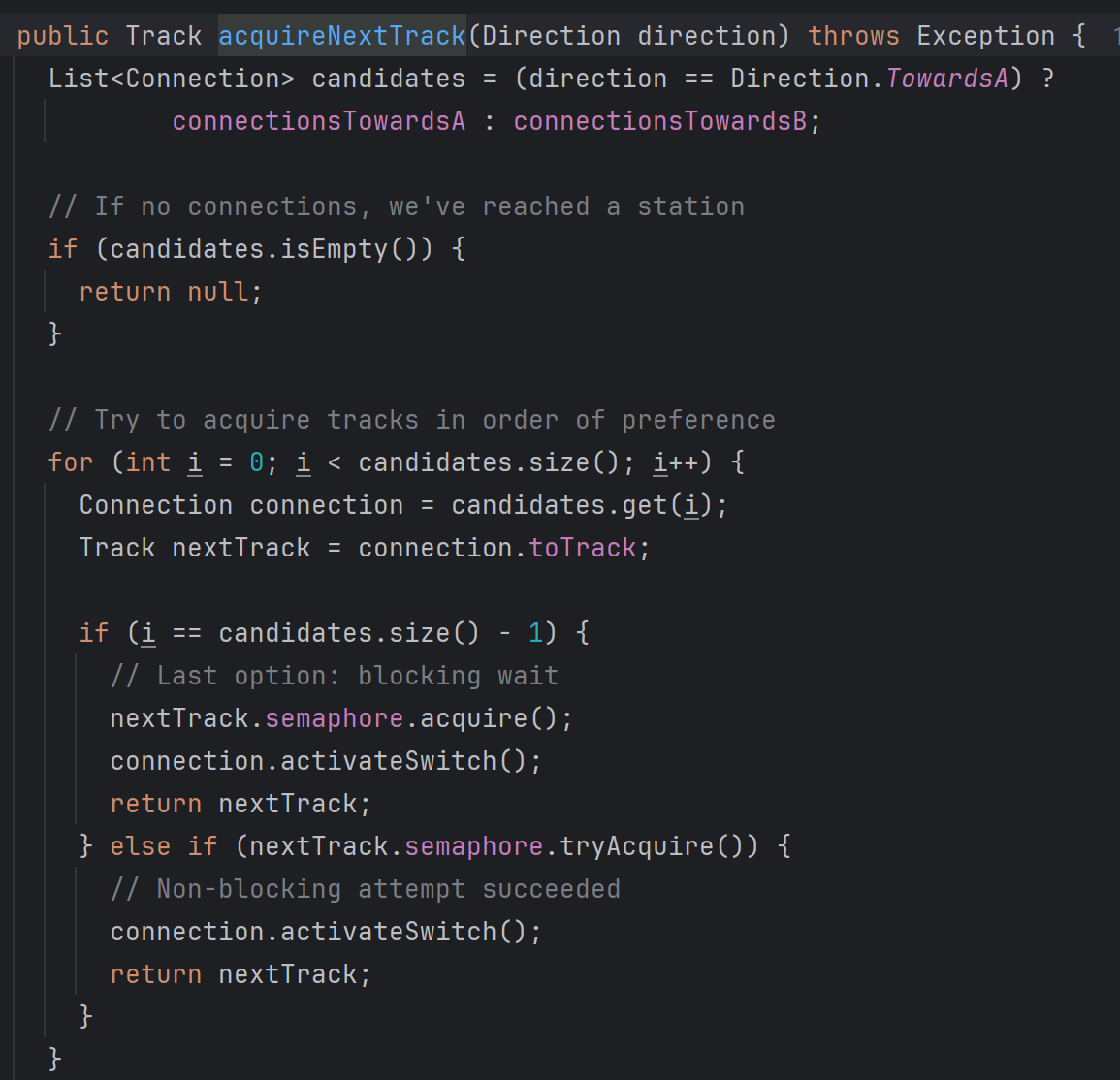


**Track:**

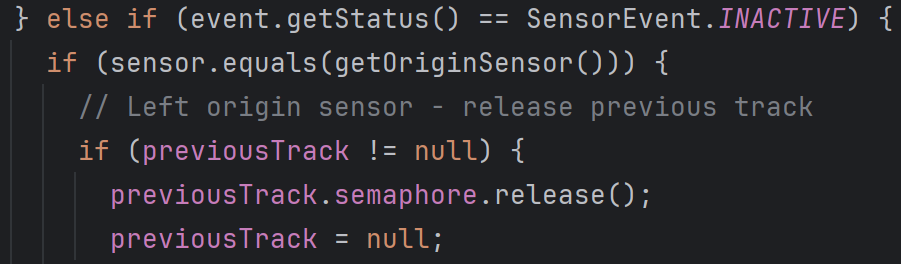
When the triggered sensor is on the track, we can give an example: on each track, the sensor close to A represents the end of the track, and the sensor close to B represents the start point of the track. A train is sent to station A, when the train travels to a sensor on a certain track close to A, triggering the "active" signal, the train stops and tries to obtain the semaphore of the next track;



Since there may be multiple tracks to choose from, it will try to obtain the first track non-blocking first, and if not, it will block to obtain the second track. After the acquisition is successful, update the currentTrack to the next track, and the previous track to currentTrack



When the train passes the sensor close to B on the next track, the "inactive" signal is triggered, and the previous track is released



**maximum speed**

The maximum speed was set at 20, and the two trains quickly collided at the intersection. We think the possible reason is that it takes some time for tsim to report the sensor event. And even if the active event is triggered, the train slows down and requires braking distance, not stopping immediately. So we think the possible scenarios are:

The slow train obtained the signal of the intersection and thought it was safe to pass. The section where the express train was located knew about Semaphore, but the collision occurred due to the tsim delay and braking distance that caused the express car to stop in time. We also tested at a speed of 19 and there were no collisions, but probably because the run time was not long, so there were no accidents. To be on the safe side, the maximum speed should be set to 15