**Use case:** Identifying if a train is on a junction

**Scope:** Junction Control system

**Level:** Sub-function

**Intention in context:** The train intends to cross the junction safely.

**Multiplicity:** Only one train can be on the junction at any time

**Primary actor:** Trains (Because they want to cross safely(= goal))

**Secondary actor:** Controller (train uses controller to reach goal)

**Maybe facilitator actor**: Lights (used by a controller to communicate with other trains)

**Scenarios:**

**Main success:**

1. Train has arrived on the junction
2. Lights remain red
3. Train leaves junction, no other trains arrived
4. Traffic Lights remain red

**Extensions & exceptions:**

1. Train has arrived on the junction
2. Lights remain red
3. Train leaves junction, other train has arrived in meantime (Don’t keep in mind multiple trains, part of 4)
4. Other train gets green light

**Use case:** Priority for multiple trains

**Scope:** Junction control system for multiple trains

**Level:** Sub-function

**Intention in context:** Trains want to cross the junction fairly

**Multiplicity:** There must be two trains to have priority

**Primary actor:** Trains

**Secondary actor:** Controller

**Facilitator actor:** Lights

**Scenarios:**

**Main success:**

1. Two trains are waiting at the junction
2. Traffic light for earliers arriving train turns green
3. Earlier Train enters junction
4. Earlier train’s light turns red
5. Earlier train leaves junction
6. Later train’s light turns green
7. Later train enters junction
8. Later train’s light turns red
9. Later train leaves junction

**Extensions & exceptions:**

1. Two trains are waiting at the junction
2. They arrived at the exact same time (Low Presicion)
3. Assign “earliest” status to random train
4. Execute main scenario