The game is called Section Controller.

In railway parlance, a *section* is that part of a railway route that exists between any two stations. The two stations may be consecutive or they may have multiple stations in between them. A railway route is said to be made up of one or many sections.

A *Section Controller* is one who controls a section. He/she ensures that no trains collide with each other. He/she does this by ensuring that there is sufficient “gap” between two trains, and that the paths taken by trains do not end in collisions. He/she also decides which train should go on which track, which train should go first, whether a late train can overtake an on-time train, etc. He/she also ensures on-time running of trains.

Actually, these decisions are made in collaboration with various railway men, such as the station master, the points man, the loco drivers, the guard, etc. Thus, this game should be a multi-player game.

However, initially, the game shall be designed as a single-player game, wherein the section controller shall play the various roles. Here, the section controller himself shall set the points, change the signals, etc. The game shall later on be made as a multi-player game. Hence for the purposes of this document, only the single-player game is considered now.

The objectives of this game are:

1. To provide a training platform for section controllers.
2. To provide scenario analysis for section controllers, so that they can analyse the impact of new trains on their section, and if required, recommend changes in timings, stoppages, etc.
3. Point 2 holds for rail enthusiasts too.
4. To analyse past ‘mistakes’ if necessary. That is, if a section controller handled a train in a particular situation in a particular manner, but later on realizes/feels that it could have been handled in a different manner so that the trains are on time, then he can use this game for verifying his intuition.

All players need to have a username, so that scores can be matched against players.

*Scores?* Oh yes, scores. The user shall be given 100 points when he first creates his username. The following actions decrease his score.

1. A train has its departure station on the section. It leaves the departure station early/on-time but leaves the section late.
2. Having entered the section early/on-time, a train leaves the section late.
3. Having entered the section late, a train is further delayed in the section and leaves the section much later than it should have.
4. The above scenarios deal with determining the points when the train leaves the section. However, for a train with its terminating station in the section, the points shall be calculated when the train reaches the station. Just to avoid confusion later, I’ll reiterate the points above with the scenario of a train terminating within the section.
   1. A train has its departure station and terminating station on the section. It leaves the departure station early/on-time but reaches its terminating station late.
   2. Having entered the section early/on-time, a train reaches its terminating station late.
   3. Having entered the section late, a train is further delayed in the section and reaches its terminating station much later than it should have.
5. Note that the user is not penalized for the train entering the section late. The user is only penalized for making the train *more late*. This also means that if the train leaves the section with the same time lag as it had when entering the section, then the user is not penalized at all. In such a case, the train is considered ‘on-time’.

The following actions increase his score.

1. A train has its departure station on the section. It leaves the departure station late but leaves the section early/on-time.
2. Having entered the section late, a train leaves the section early/on-time.
3. Having entered the section early/on-time, a train is able to further decrease its time lag in the section and leaves the section much earlier than it should have.

The points decreased/increased shall be proportionate to the number of minutes the train was made late/early. For every minute that the train was made late, 10 points are to be deducted. For every minute that the train was made early, 10 points are to be added.

In the single player mode, as soon as the user logs in, he can go to two areas: game area, or administration area. In the game area, the user performs actions that enable him to play the game.

* 1. A section has a set of stations. Some stations are major stations, while some are minor.
  2. For now, the game will only deal with single-line sections. Single-line sections are those where there is only one railway track, and that track is used for trains in both directions. Multiple-line sections will come in future versions.
  3. The game area will have a horizontal line that represents the single-line section. There will be various vertical lines of a small size drawn on this horizontal line. These are the stations on the section. The home station will be the leftmost station, while other stations will be drawn starting from the home station towards the right. The rightmost station will be the final station of the section.
     1. The stations will be drawn on the section proportional to the real-life distances from the home station.
     2. The stations at the end of each section are called “*section entry stations*”.
  4. Each section has a “home” station which is where the section controller sits. The section starts from the home station and continues until it reaches another station where another section controller sits.
  5. Each section controller has under his control, only that portion of the route that starts from his home station to just before the next home station, *but not the next home station*. This has two implications;
     1. The section controller can control his home station, all stations on the section but not the next home station, and all tracks in his home station, as well as at each station on the section, but not the tracks at the next home station.
     2. When a section controller wants to send a train into the portion of the track that is controlled by another section controller, the first section controller must confirm with the other section controller before doing so. This is because the first section controller does not have the next station under his control (*it is in the control of the second section controller*) & in a single-line section, the next signal that the train will face is only at a station, and planning must be done for that train at that station to ensure a safe route is available.
  6. Stations are those places where there are multiple tracks. Each station has a set of tracks, some of which may have platforms. Usually major stations are those where all tracks have platforms, while minor stations are those where only a few tracks have platforms and other tracks don’t.
     1. In a single-line section, the only way to make trains in opposite directions cross each other safely is by performing the crossing at a station, where there are multiple tracks.
     2. In minor stations, it is preferable to use those tracks without platforms to route trains that don’t stop at the station. This isn’t a compulsion, but a best practice. The game will not remove points if the user doesn’t follow this.
  7. Each station has 1 point at either end which sets the track that a train will take when it enters the station, & when it exits the station.
  8. The user can change the points at stations so that trains can be routed through.
  9. There is a signal at each end of the station, for each track through the station. The two signals on a track face each other from either end. Thus, for a train entering a station, only those signals on the same track but on the other end of the station are useful. If a train is moving through the station to another station, then the user must set the aspect for the correct signal.
     1. In a single-line section, only stations have signals. This means a green aspect given at a station indicates that the train can move from the current station to the next station without any risk to it.
  10. The user can set the signal aspect so that a train can stop/move through the station.
      1. Note that if a train is scheduled to stop, the controller can still give a green aspect. The train will stop for the required duration and then move on.
  11. The signal aspects at each station are: Red, Amber & Green.
  12. Every section has a Maximum Permissible Speed (MPS) which is the maximum speed any train can travel on the section. No train shall exceed this speed even if it’s late.
  13. Trains run on the section. Some trains may be non-stop in the section, while some trains may have scheduled stops at various stations in the section.
  14. All trains will have a section entry time & a section exit time. This will help in determining when non-stop trains that travel trough the section must appear in the game. This will also help in determining when trains with a scheduled timetable, but without a scheduled stop at the first station on the section must appear in the game.
  15. A train will always aim to be on-time at its next stop. Thus, the following scenarios apply:
      1. A train will move from the current stop to the next stop only at a speed that will ensure it reaches the next stop at the scheduled arrival time. This means if a train has its route clear till the next stop, it will not try to move fast & reach early.
      2. If a train is late while starting from its current stop, it will still aim to reach its next stop on-time. This means that it will move at a higher speed through the section than it would have if it had started on-time from its current stop. However, the train’s speed shall never exceed MPS.
      3. If a train is on-time while starting from its current stop, but it has to make a few unscheduled stops before its next stop, then while starting from the unscheduled stop, it shall move at a speed that will ensure it reaches its next stop on-time. However, the train’s speed shall never exceed MPS.
  16. Note that the point above refers to trains with scheduled stops on the section. For trains that do not stop anywhere on the section, the rules in the point above read as follows:
      1. A train will move through the section only at a speed that will ensure it exits the section at the scheduled time. This means if a train has its route clear through the section, it will not try to move fast & reach early.
      2. If a train enters the section on-time, but had an unscheduled stop on the section, it will still aim to exit the section on-time. This means that it will move at a higher speed through the section than it would have if it did not have an unscheduled stop. However, the train’s speed shall never exceed MPS.
      3. If a train enters the section late, it will still aim to exit the section on-time. This means that it will move at a higher speed through the section than it would have if it did not have an unscheduled stop. However, the train’s speed shall never exceed MPS. This also applies if the train enters the section late & had unscheduled stops on the section.
  17. Now that we understand the concept of MPS and train speeds, here are the meanings of each aspect:
      1. Red: The train shall come to a complete stop just before the signal.
      2. Green: The train can travel at any speed as long as it does not exceed MPS.
      3. Amber: The train can travel at a speed that does not exceed half of the MPS.
      4. All of the above rules apply whether a train is early/on-time/late. There are no exceptions.
  18. When a train has an unscheduled stop at a station, it can stop on any track in the station.
  19. When a train has a scheduled stop at a station, it must stop only on tracks with platforms. Stopping on a non-platform track will reduce the user’s score by 10 points. This is to ensure passengers aren’t inconvenienced.
  20. A train can depart from the scheduled stop only at the scheduled departure time. Thus, if a train is early, it should wait until the scheduled departure time to depart. This is because passengers plan their arrival at the station based on the scheduled departure time.
  21. A train must stop for the duration of the scheduled stop. The “duration” is defined as the time difference between the scheduled departure & arrival times for the train at that station. This has two scenarios:
      1. If a train arrives late, but before the scheduled departure, it must stay for the duration even if that means it will stay beyond the scheduled departure time. This is to avoid inconvenience to passengers in situations where a train arrives 1 minute before its scheduled departure time.
      2. If a train arrives beyond its scheduled departure time, then it must wait at the station for the scheduled duration.
  22. Thus, from all the above scenarios, it’s hopefully clear that a late train can cover up its lag only in the area of the section in between scheduled stops. The more a train lags, the more it will find it difficult to cover up the lag, especially in the face of trains going in opposite directions or having more stops. Efficient handling of conflicting trains in these areas is the only way to ensure a good score.

In the administration area, the user can do administrative tasks. These are:

* 1. He can add new trains,
     1. This means both passenger and goods trains. So there should be a provision, like a checkbox, for saying whether the train is a passenger or goods train. If a passenger train, get the stoppages. If not, just get the origin and destination and get the timings at the starting and end stations of the section.
  2. He can delete existing trains,
  3. He can add stoppages to trains,
  4. He can delete stoppages from trains,
  5. He can change the time a train arrives at a station,
  6. He can change the time a train leaves a station,
  7. He can change the MPS of the section,
  8. He can add new tracks
     1. to a station,
     2. as a new route
        1. He can create the new route by holding the mouse down and dragging the mouse to create the new route.
        2. This is very complex; let us not have this in the first version.
  9. He can remove tracks from a station,
  10. He can remove a route entirely,
  11. He can add a platform to a track without one,
  12. He can remove the platform for a track, thus making it platform-less,
  13. He can add points to the tracks at a station.
  14. He can remove points to the tracks at a station,
  15. He can add signal posts to the tracks at a station,
      1. Did I just mention *signals*? Sorry, for the initial purposes of this game, all signals can be found only at stations, not in between them.
  16. He can remove signal posts from the tracks at a station,
      1. Is this safe? How about a warning message here?
  17. Yikes, I forgot!! He can add a station,
  18. By corollary, he can delete an entire station,
      1. If he does so, then the information about this stopping is removed for all trains.

The administration area changes must be stored somewhere, and for that, we need a storage mechanism. In order for the game to be widely usable with the least amount of installation required, we can use XML files (*which are human-readable as well as computer-processable*) which we bundle along with the game. This section describes what files will exist & how each file’s XML structure is.

All XML files are stored in the data/ folder.

Each section is described using an XML file, whose structure is as follows:

<section mps=”” home-station=””>

<stations>

<station code=”” name=”” distance-from-home=”” no-of-tracks=”” no-of-platforms=””/>

</stations>

<trains>

<train number=”” name=”” day-of-arrival=”” direction=”” section-entry-time=”” section-leaving-time=”” />

</trains>

</section>

The <section> tag describes the entire section. The mps attribute is the maximum permissible speed of the section expressed in km/hr.

For all the stations in a given section, the <section> tag contains multiple <station> tags, inside the <stations> tag. One of these stations must be the home station for the section, and that station’s code must be specified in the home-station attribute of the <section> tag.

Each <station> tag contains a code attribute (which is the station code), a name attribute (which is the name of the station), a distance-from-home attribute (which is the distance of that station from the home station of the section), a no-of-tracks attribute (which describes how many tracks exist at the station), & a no-of-platforms section (which describes how many platforms exist at the station).

The Section Controller game will understand that the section controller’s control starts at the home station and proceeds in ascending order of the distance-from-home attribute with the exception of the last station, i.e. the station with the maximum value of distance-from-home.

The name of each section XML file is the name of the first station followed by a hyphen followed by the name of the last station. Here, first station should not be the first <station> tag immediately below <section>, but the home station’s station tag. Similarly, the last station should not be the last station tag, but the station with the maximum distance-from-home value.

Some constraints in the section file: 1) Because a section must contain atleast two stations, the <section> tag cannot be empty, nor can it contain only 1 <station> tag, 2) In a <station> tag, the no-of-platforms attribute value must be less than or equal to the no-of-tracks attribute value, 3) The distance-from-home attribute of the home station specified in the section tag must be zero.

For all the trains that will pass through the given section, the <section> tag contains multiple <train> tags, inside the <trains> tag.

Each <train> tag contains a number attribute (which is the train number), a name attribute (which is the name of the train), a direction attribute (which is the direction of travel of the train in the station – this is used to determine if a train is travelling towards the home station or away from the home station), a day-of-arrival attribute (which is the day of the week that the train enters the section), a direction attribute (which describes the direction of the train, either towards the home station or away from the home station), & section-entry-time & section-leaving-time attributes (which describe the time of day when the train enters/leaves the section).

The value of day-of-arrival attribute is “Daily” for Daily trains. For non-daily trains, the value must be “M” for Monday, “Tu” for Tuesday, “W” for Wednesday, “Th” for Thursday, “F” for Friday, “Sa” for Saturday, & “Su” for Sunday, if the trains enter the section on those days. Any other values are invalid. Combinations of these days can be specified using comma-separated values, eg: “M,F”. Such combinations cannot have “Daily” in them – if they do, it is an invalid value. Eg: “Daily,F” is invalid.

The value of the direction attribute can only be one of “TowardsHome” or “AwayFromHome”. Any other values are invalid.

The section-entry-time & section-leaving-time may appear to be unnecessary, since each train will also have its own XML file that lists the stations it stops at on the section. However, they may be situations where a train travels through a section, but does not stop at any station on the section. In such cases, the only way to identify that such a train exists & is scheduled to travel through the section is through these attributes.

One implication of having all trains on the section listed inside the <trains> tag for that <section> is that some trains which can potentially be in a section entry station may not be known purely from the data in the <trains> tag. Consider two sections, A-B & B-C. Here, A, B & C are major stations with several minor stations in between. If the user is playing on section B-C, and B has a train departing from it towards A, then the Section Controller game will never know about it, because A-B & B-C sections are separate XML files. This will also mean that the game is not “real” as in real-life. This implication can be left as-is for now. Future versions of the game will fix this.