

AiEdit

In this help file you will find an explanation on how to make a set of ai files for your track with use of AiEdit.

1 .Ip Files

First of all some explaining of the files you're about to create is necessary.

Every track has a set of .lp files which guide the ai around the track.

Every .lp file consist of parameters which GPL reads to position the ai on the track and give them a heading and speed. Keep in mind that the values for positioning a car are all related to the centreline of the track.

1.1 Speed and Positioning:

For instance all the .lp files have the following speed and positioning parameters:

Longitude = the number in meters from the s/f line.

Latitude = the amount in meters from the centreline of the track (- is to the right and + is to the left of the centreline)

Longitudinal speed = the speed in mph the car will have at that point (ideally)

Lateral speed = the speed in vertical direction to the longitudinal speed.

Yaw speed = the rotation about a vertical axis that passes through the car's centre of gravity.

1.2 The complete set:

The complete set of the .lp files make up the boundaries in which the ai normally would behave.

Were:

Race.lp = the normal race line an ai car would try to follow.

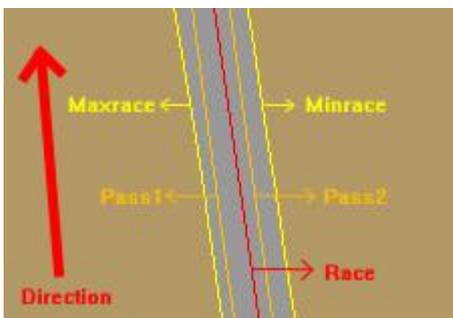
Min/Maxrace = the boundaries of the track surface. Also used for the shift-R and defect cars returning to the pits.

Pass1/2 = passing line which can (situation dependant) be used by an ai car.

Pit.lp = the line ai cars will follow when going into pit mode.

So if you would look at a track section the line up of the files would be something like this:

Maxrace – Pass1 – Race – Pass2 – Minrace (the pitline would follow the race line for 90% so forget that one at this point)



AiEdit

1.3 Flags and modifiers:

You've probably noticed that there are some other parameters as well.

There's one speed modifier and several Flags in a .lp file.

These flags and the speed modifier are only needed in the race.lp. GPL ignores all other .lp flags.

Below you'll find a brief description of these flags and what they're used for. I've put them in approximate order of decreasing importance:

1.3.1 Speed modifier:

The speed modifier is actually stored along with the flags, but follows different rules. I've called it a speed modifier, but it also effectively increases the grip levels. So adding a speed modifier makes the AI go faster and be less likely to understeer off. The numerical value is the proportional speed increase, i.e. 0.1 means 10% faster than the speeds would normally indicate.

Flag rules:

First some basic rules concerning the use of flags in an .lp file:

1. You can only ever put one flag on a given lp record.
2. Do not put a flag on the last record. GPL could ignore the last record.

1.3.2 Flags:

Tire Warmup:

You need to have at least one of these or else the AI tires will never get up to temperature. Space them roughly evenly around the track at approx. 2km intervals. The first one is normally about 2km in, the last should be towards the end of the track but before the starting grid and pit stalls. That way the AI start with properly cold tires.

Start Speed Checkpoint / End Speed Checkpoint:

These are crucial for passing, and in particular it's important to have at least one pair. Each start point should have a corresponding end point. Original GPL only supports 24 pairs, so you should normally limit yourself to that many. The long track mod increases that number to 1024, so you can go wild on long tracks :)

I'll describe the end point first. The end point is just a speed trap - GPL keeps track of the speed that each individual car passes each end point. It actually stores a weighted average per car / per end point, with the most recent laps being given more weight.

When a car reaches a start point, it checks whether there is a car close ahead of it (within 64.5m). If there is, it compares its own speed at the corresponding end point with that of the car in front. If it is quicker than the car in front, it makes a mental note that it would like to pass that car. (That's a bit simplified as the driver aggression rating also has an effect, but that's the general idea). Then GPL calls this car in front the 'designated passee'. Note that this designated passee status does not stop when the car reaches the end point, but it will get reassigned at the next start point.

Having a designated passee makes a large difference to how the AI behaves. One thing that's easy to spot is that they'll follow that car much more closely than normal. If you experiment with these flags you'll see that you can get cars to pass each other even when the pass1/pass2 lines are the same as the race lines.

Papyrus seem to use these checkpoint zones in two ways. Often on a straight you'll see a start point followed immediately by an end point. That's basically saying 'if I'm faster here I'll try to pass'. But in corners and braking zones, you'll more likely see a big gap between the start and end points, so that's more like saying 'I'm quicker through the upcoming corner, so I should get into position to pass now'.

Outbrake Decision Point:

When a car hits an outbrake decision point and it has a designated passee, it will try to pass, following either the pass1 or pass2 line. Which of pass1/pass2 it uses is determined by the race.lp yaw speed at that point - if it's curving left it will try to pass on the left, otherwise the right, i.e. it's always attempting to outbrake on the inside. Since it's based on having a designated passee, I think it's good practice to place outbrake decision points within a start/end speed checkpoint pair.

AiEdit

Unfortunately GPL had a huge bug which meant that outbrake decision points generally didn't work as they were intended - the car would normally switch back to the race line immediately. I've fixed that in the 66 mod, and attempted outbraking manoeuvres are now very clear.

Start Long Straight / End Long Straight:

These are quite simple, and demarcate a straight. I've called them 'long' straight, but if you look at the Papy lps they use them on quite short straights, and also sections that curve but are still taken flat out. Designating something as a straight makes the AI much more likely to attempt a pass. In particular it allows them to make abrupt changes of direction which would get them into trouble on corners.

The code I added to make lapped traffic move out of the way does so only during long straights.

Start No Pass Zone / End No Pass Zone:

Reasonably self explanatory. The AI won't initiate a pass within a no pass zone, though they may complete one. I think no pass zones should only be used as a last resort as it means they will happily sit behind extremely slow cars. However Papy still used an extremely long no pass zone around most of Monaco, and for narrow tracks with walls either side there may be no alternative.

Abandon Pass:

This just removes any designated passee. It's normally used immediately prior to a no pass zone.

Start Pass1 / Pass2 Zone, and Abrupt equivalents:

Papy never use these flags. They simply force the AI to use the pass1/pass2 lines instead of the race line. The effect last for a certain amount of time - I forget exactly how long. If you make the race line the line you want, in general you won't want to use these flags. I did find a sneaky use for them however, though again it's rather a last resort. If you really want the AI to follow the race line somewhere but instead they're too busy passing each other, what you can do is make either the pass1 or pass2 line the same as the race line, and then force them onto the appropriate pass line. The forcing seems to be rather brutal, which can cause cars to run into each other. Also once you've done this you need the pass and race line to be the same for long enough that the effect will expire.

Initiate Cornering:

Papy never use this. Normally the AI think they're in a corner if the current trk section is a corner with a radius that's not too large. Knowing they're in a corner affects passing attempts etc. You can use this flag to initiate cornering behaviour earlier. The effect lasts until the end of the current trk section. I haven't come across a use for this flag.

Set Speed Coëff to 1:

Papy never use this. It sets the speed modifier to 0, but you can also do that by... setting the speed modifier to 0 :) Useless.

Pass Lps:

The pass lps aren't used quite how I expected, so I thought I'd mention that here. I imagined that pass lps would affect where passes would be attempted, but that doesn't appear to be the case (at least not normally). Whether a pass is attempted is controlled much more by the flags, in combination with some general principles in the code. Between the flags and the code, it makes a decision as to whether or not to pass, and also to what side and at what latitude. Only then do the pass lps come into play. The code checks whether the appropriate pass lp line is closer to the desired latitude than the race lp line - if it is it uses the pass lp as the guide line instead of race.lp. Either way, it doesn't simply use the line, rather it runs parallel to the line, based on the latitude that it originally decided to pass it. Once a pass is initiated it continually modifies the latitude offset, but it will still be in some sense following either the race line or one of the pass lines.

A postscript to that last comment. I think it's true in respect of an actual pass attempt. However a different situation is when a car considers moving to the left or right just to avoid hitting a car in front. In those cases I think the amount of gap between a passing lp and the car to be avoided is important - if there is sufficient gap the car is more likely to go around on that side.

AiEdit

And if there isn't sufficient gap on either side, it's more likely to simply slow down to stay behind. I like to make my passing laps similar to what Zandvoort has, where the pass laps form a kind of corridor around most of the track, with the race lap snaking from side to side within that corridor.

2. Preparations before using AiEdit

2.1 Replays

Before starting with AiEdit we still need some reference laps in replay format (.rpy) and the program Rpy2lp to create the necessary base .lp files.

We need to have a raceline replay which gives us the lateral position of the ai car along the track. So first of all drive a lap around the circuit and try to follow a realistic driving line. The speed will be generated by AiEdit so it doesn't have to be an alien lap (drive as slow as you need in order to get a proper line) In case of a repair of faulty ai use the already made groove if it has a realistic line. It doesn't have to be a super smooth line (you can iron out most faults with AiEdit) but abrupt changes give you more work than necessary =)

Second we need a pit raceline replay which gives us the lateral position of the ai car along the track. I always drive a lap through the pitlane. Make a full lap and drive through the pitlane again. When approaching the pit make sure you position the car on the side of the track. This is to make sure other cars are able to pass when the ai car goes into or out of pitmode (slowing down or accelerating out of pits) If you are not certain at which point you want to have the cars move out of the racing line it's best to extend the sections at which you want to start pitmode. E.g. move out of racing line earlier before pits and stay longer of racing line out of pits (the trigger for this is based in the track.ini file and will be explained further on)

Note: make the replay a bit longer than the actual lap. So start saving well before start/finish (or pits) and end saving well after T1 (or pits)

In case of repairing faulty .lp files first delete the present .lp files (or back them up if you need) They are not always the correct length and can cause unnecessary errors.

Before firing up Rpy2lp it is necessary to investigate the track.ini for possible errors. I will explain the parameters within the Track.ini first so it is easier to spot errors.

2.2 The Track.ini explained

The Track.ini file interacts with the .lp files to tell GPL were to start race behaviour, pit in/out mode, etc.. I will only explain the parameters that have some meaning to the ai as well as parameters that have been misunderstood often.

Warning: If you start with a new Track.ini from scratch the best you can do is copy an existing Track.ini and alter the values. Why? Forgetting to place a comment (";" followed by text) somewhere in the file can cause weird things to happen: Track won't load, cars getting out of fuel after just one lap etc..

[track]

track_length = track length in mile. Failing to add the 'm' after the length will result in a faulty distance being displayed. This is also used to calculate the race distance. So 7 laps @ 2miles = 14 miles race distance. I probably shouldn't need to mention that putting in weird values causes problems ;)

[ai_track]

end_start_behavior_section = 2.000000 ; section at which race start behaviour should end

This determines at which **section** the shuffling for position stops and normal race behaviour should start.

Don't use a too high value and don't use 0. These will cause the ai to shuffle for positions for a very long time or the entire race. Although this can be fun to watch if I'm not mistaken accidents occur more often this way.

AiEdit

`merge_from_pitlane_dlong = 1034.00` ; dlong at which cars leaving pits will merge to race line
What it says at this point the cars will drive from pitline to raceline. So if you have done it right a car driving out of pits will not plunge for a hotlapping car but stay out of its way until it's (more or less) safe to merge with the racingline. Choose this point several 100 meters out of pits.

`pit_at_end_of_track = 0` ; 0 if all stalls beyond s/f, 1 if not
If the pit positions(stalls) overlap the s/f or are all before s/f set it to 1.0 otherwise use a zero.

`pit_lane_end_dlong = 12336.00` ; dlong where pit lane divider ends
Position were the car can exit the pitlane.

`pit_lane_entrance_width = 48.0` ; dlong width of pit lane entrance in meters
Pit entry width in meters. So how much room has the car for entering the pitlane.

`pit_lane_has_wall = 1.000000` ; 0 if no pit wall, 1 if pit wall exists
Set it to 1.0 if there is a wall between the pit positions and the track itself.

`pit_lane_on_right = 1.000000` ; 0 if lane on left, 1 if on right
Set it to 1.0 if the pits is on the right side of the raceline.

`pit_lane_start_dlong = 3250.000000` ; base dlong for calculating transition to pit line
Opposite of `merge_from_pitlane_dlong` at this point the cars will drive from raceline to the pitline. So if you have done it right a car driving into the pits will not decelerate in front of a hotlapping car but pull to the right/left and decelerates. It will stay out of its way until it's arrived at the pit entrance. So choose this value well before the pitlane itself.

`pit_lane_width_left = 2.00000` ; width of pit lane in meters to LEFT of PIT.LP
This will tell GPL how much room there is left of the pit driving line. This might require some fiddling with numbers to get it right.

`pit_lane_width_right = 8.000000` ; width of pit lane in meters to RIGHT of PIT.LP
Same as before but now it tells GPL how much room there is to the right of the pit driving line.
Again this might need some shuffling with numbers to get it right. Wrong values can prevent cars from getting out of the pits.

`start_of_pit_stalls = 2724.00` ; dlong where pit stalls begin
I haven't figured out the use of this value yet sorry!

`brabham_fuel_load_per_lap = 4.83` ; Brabham AI fuel loading in litres per lap
There's a value for all 7 base chassis in the track.ini. This amount tells GPL how much fuel the car consumes driving 1 lap. I normally would take the length of the track divided by the Monza track length to determine the factor of litres required for the track. I.e. lets say this track = 7.7miles. 7.7miles divided by 3.573miles (Monza length) gives you a factor of 2.15505177.....etc. So to determine the fuel load of the Brabham on your track you take the amount for the monza track and multiply it with the above factor: 2.24 litres * 2.155 = 4.8272 = 4.83. Do this for all chassis to get proper values. If you use wrong values the cars might be to heavy or to light in relation to the planned race length.

`track_dlong_sep_coeff = 0.9` ; slightly closer racing here
This value is forcing the ai to drive closer to each other. Might be useful on the smaller tracks.
The value is a % so 0.9 makes them drive 10% closer to each other. A value above 1.0 will do the opposite.

`Track_global_hype_scaling = 0.8` ; Speed modifier
This is also a scalar factor it will slow down the ai (<1.0) or speed them up (>1.0)
The way AiEdit is designed you shouldn't need this value.

`glass_wall_offset = -0.75` ; bump setting for armco
This one is only used on Monaco and Montjuich as far as I know. It is used to force ai back on the track if they come within the distance (offset value set in the parameter) of a wall. Use this only as a last resort because it doesn't look very realistic. It is normally only necessary on narrow tracks with armco on either side of the road (so there's no escape for the ai)

AiEdit

`track_start_speed_limit = 1.231` ; limiter to prevent dragrace starts
I've only seen this one at the Zandvoort track. Here the s/f line is around the spot were the cars have a topspeed.
So I guess this parameter prevents the cars getting to quick from the start position.

[GP]

`dlong_speed_adj_coeff = 1.024000` ; value modifies dlong velocities (i.e. RELS) in .LP
Use this value to increase speed if you used a grip parameter below 1 in AiEdit.

`dlong_speed_maximum = 2.260000` ; max. dlong speed (meters/tick) for .LP after all modifiers (e.g.,
`dlong_speed_adj_coeff`)
This value is used to limit top speed at a track. The way AiEdit is designed this value should be set as high as possible. So set it always to 2.5. You can lower the value if you feel the top speeds are too high.

[pit_lane_0]

`lane_on_right = 1` ; 0 if lane on left, 1 if on right
`lane_has_wall = 1` ; 0 if no pit wall, 1 if pit wall exists
Pretty self explanatory I guess. But keep in mind that these should have the same values as in the [ai_track] section!

Warning! Read this carefully!!!

Beneath you'll find the values that have been interpreted false very often and they always result in cars not leaving the pits!

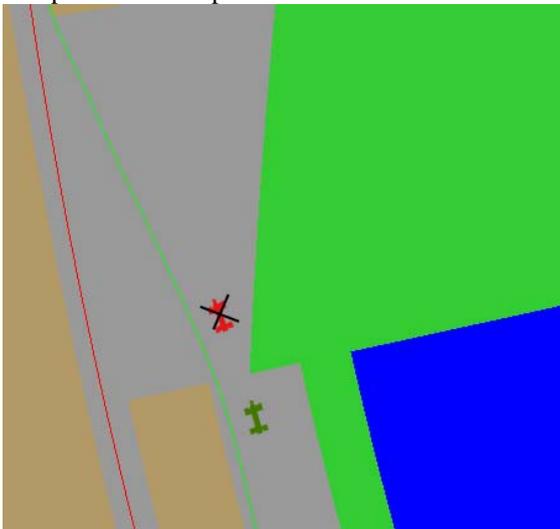
`lane_bounds_dlong = 11667.0 12336.0` ; start/end dlong of the pit lane
The left value should always be the dlong arriving at the pits. The right value should always be the dlong exiting the pits. So the second value should always be higher than the first. (unless the pitlane spans beyond s/f line!)

`lane_bounds_dlat = -35.0 -25.00` ; start/end dlat of the pit lane
The 1st value should always be the value to outermost value on the track. A positive value means on the left side of the track and a negative value means on the right side of the track. If not all cars get out of pits in intermediate or pro modes these values could be switched.

`stall_0 = 5662.00 -16.00`
`stall_1 = 5654.26 -16.00`
.
.
`stall_19 = 5515.00 -16.00`

The stalls are also determined from the centre of track and should always be within the pit lane_bound values!

Note: Do not place cars to close to the line exiting the pits (see red car in the picture below)
The pitline and the pitstall should have the same heading so it is physically possible for a car to use the pitline.



AiEdit

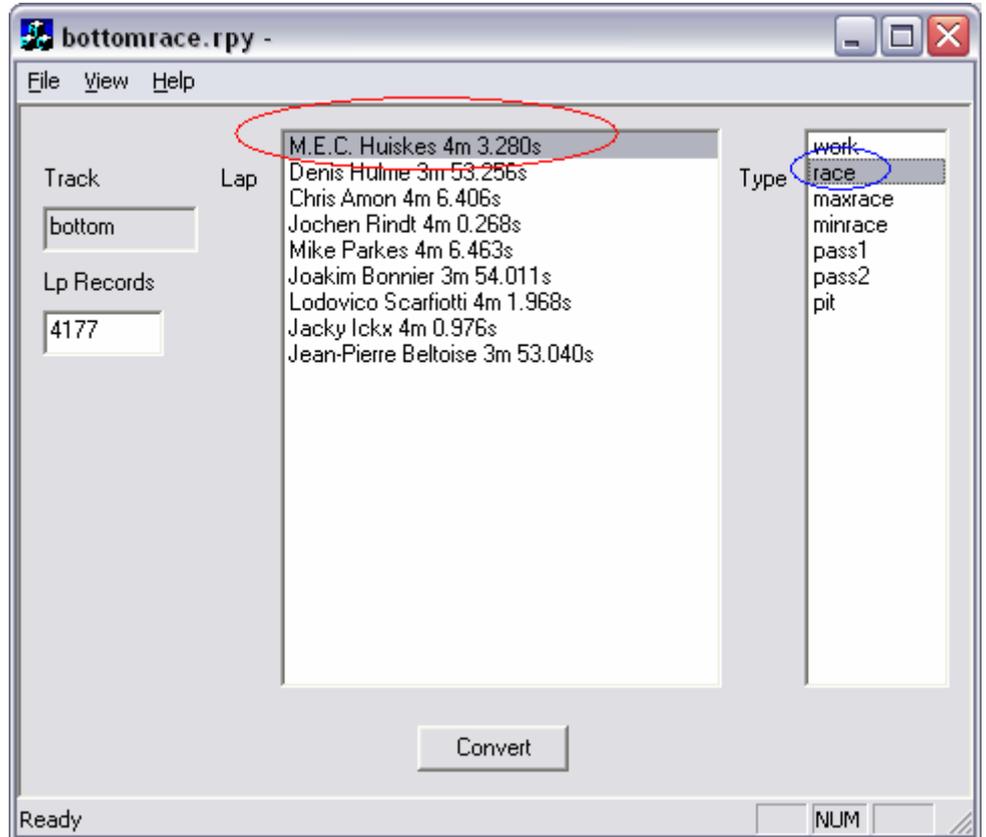
2.3 creating the lp files from the replays

Now that most of the facts have been handled it's time to get to work!

Start the Rpy2lp.exe program.

Select File – Open and navigate to the folder were you've stored the race and pit replays.
Your display should look something like this:

Now choose the right driver in the lap section (red circle) If you've driven the lap in training mode you should be the only driver selectable. If you've driven more than one lap you need to choose the right lap. Next choose the type of lp file you want to create (blue circle) and press Convert.



AiEdit

If there aren't any errors in your Track.ini you should get the message displayed on the right. If not you will most likely get some assertion error. Go back to your track.ini and examine the pit lane or starting grid parameters carefully. But first make sure you've removed all old .lp files before converting because these can cause these errors also. Check your track folder, there should be a new race.lp file present. Do the same for the pit replay but this time select for the type Pit and press convert again. Since we are done with Rpy2lp you can close it now and move on to the more exciting stuff.



2.4 Starting with AiEdit

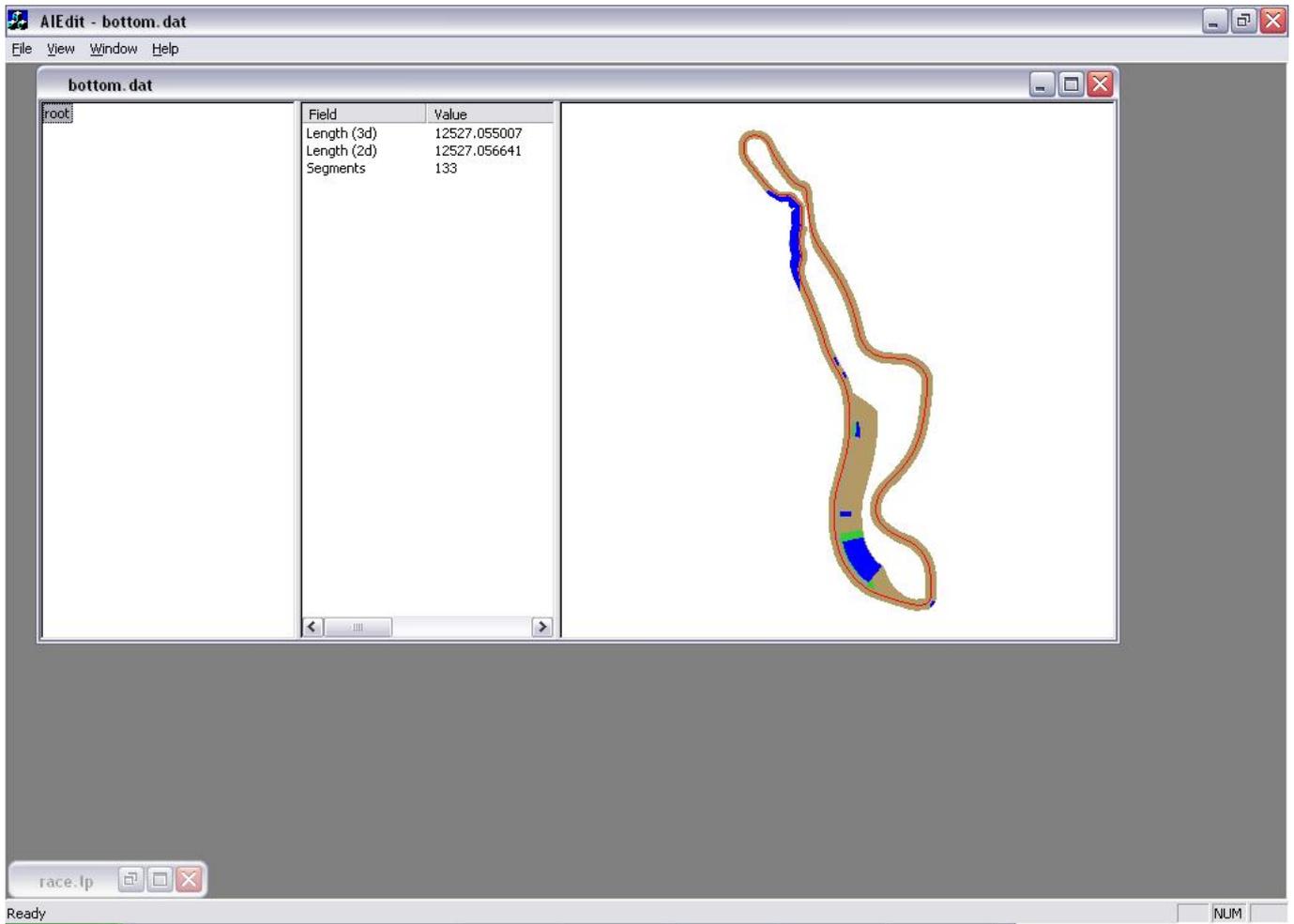
Start AiEdit, select File – Open and navigate to the folder where the track is of which you want to create ai for.

Select the proper .dat file and click on Open.

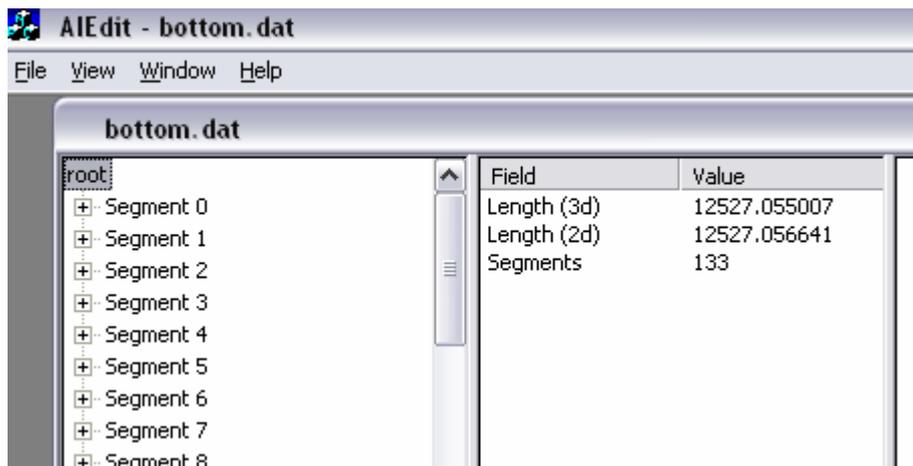


AiEdit

Once AiEdit has loaded you should see something like this:



AiEdit has loaded the track data and displays the layout of the track on the screen. Left below you should see the file race.lp which you created with Rpy2lp.exe. The track is divided in sections, to display these sections you need to double-click on root.



AiEdit

If you want you can see what data is behind the segment by clicking on the +:

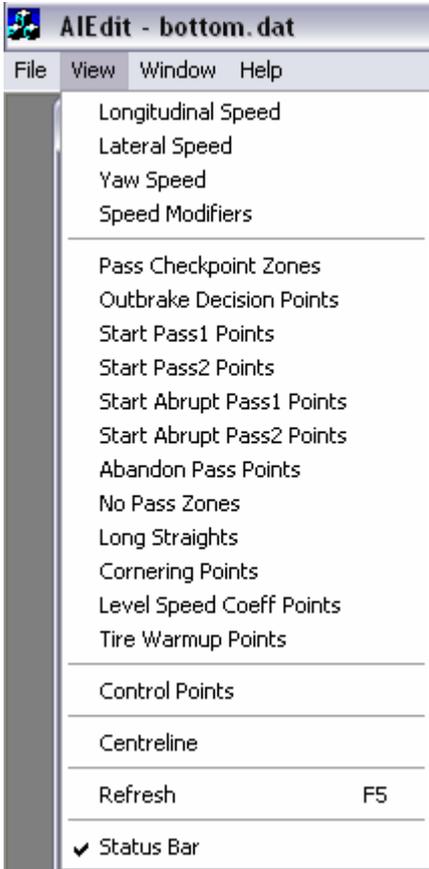
Field	Value
Start Latitude	-300.000010
End Latitude	-300.000010
Start Altitude	29.999991
End Altitude	30.999989
Linear Coefficient	0.617271
Square Coefficient	1.213002
Cube Coefficient	-0.830275

More important is the info you can find in the column between the segments and the track layout. Here you can find the start and length of a segment (section) of the track:

Field	Value
Type	Corner
Length (3d)	229.264921
Length (2d)	229.264908
Start (3d)	59.795258
Start (2d)	59.795258
Start Orientation	90.000000°

You will also see that the overall track layout has changed to the segment you've pointed to. The selected section is surrounded with a red line. Furthermore the race.lp file is placed on the track as a red line to indicate the position of the ai around the track in racemode.

Now press on the view option in the menu bar.



A list with several options is displayed.

By selecting one of these options a display feature is turned on.

Selecting it again turns it off again.

The first 4 options display coloured bars next to the racing line along the track. A high value means a darker colour. A low value..well you get the point.

The next set of options (12) are displaying the flags you can set in the race.lp

The control points option we will explain later.

The centreline option will display the track centreline in a white colour.

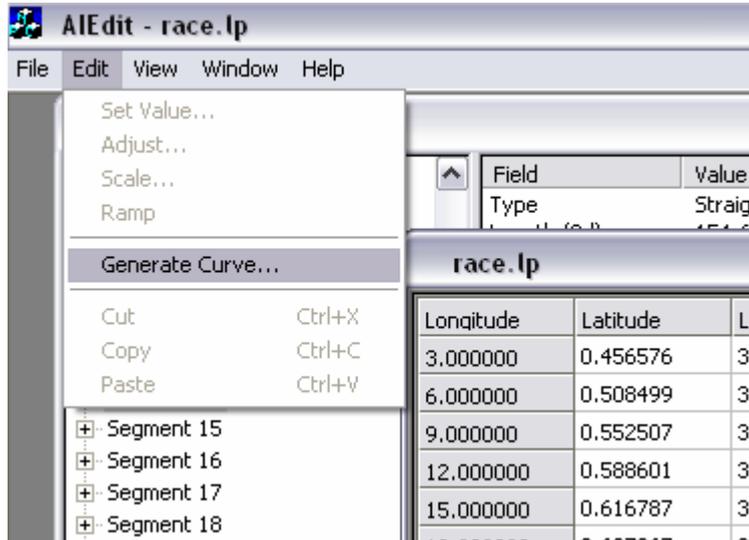
The Refresh is for displaying any changes you have made in the files.

Note: Right now you can't see any of these values cross the race.lp you have created doesn't have any flags or speed modifiers. Second there's been added a scale option which isn't displayed here. Here you can zoom in or out on the track section. It's pretty self explanatory so i won't explain more here.

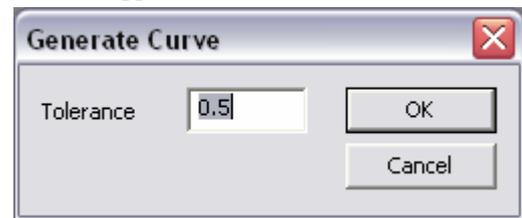
AiEdit

2.5 .lcv files

Lcv stands for 'Lap Curve', and a lcv file stores a line (racing, passing etc.) around the track as a series of cubic curves. There's no speed information like you would have in a lp file, just the line. With help of these .lcv lines it is possible to shift lines along the track. Now it is possible to alter the raceline and create passing lines and min/max lines. So first we need to create a race.lcv file from the race.lp file.

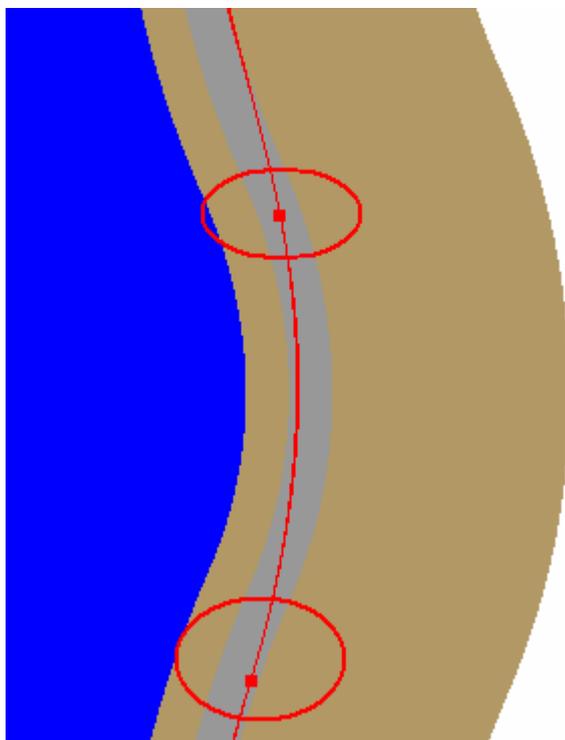


To do this with the lp window selected, choose 'Generate Curve' from the Edit menu. This generates an lcv file, with a series of cubic curves that approximate the line from the lp file. You can choose how close you want the approximation to be - the default value is



generally good for race lps, you might want to drop to around 0.1 for minrace/maxrace. Save the file as race.lcv (to keep it simple) Notice the new file race.lcv which is added. Now select the track.dat window and press view.

Select the option Control Points to display them for the race.lcv file.



If you've done it right you should see that the raceline now has points added to it (see the circles in the image to the left) Now select the race.lcv file.

Longitude	Control	Latitude	Longitudinal...	Lateral Speed	Yaw Speed
3.000000		0.528992	3.600000	0.048171	-0.039594
6.000000		0.566024	3.600000	0.040709	-0.039559
9.000000		0.596842	3.600000	0.033254	-0.039525
12.000000		0.621450	3.600000	0.025807	-0.039490
15.000000		0.639854	3.600000	0.018366	-0.039456
18.000000		0.652061	3.600000	0.010931	-0.039422
21.000000		0.658075	3.600000	0.003504	-0.039389
24.000000		0.657902	3.600000	-0.003918	-0.039355
27.000000		0.651547	3.600000	-0.011333	-0.039322
30.000000	1	0.639015	3.600000	-0.018743	-0.053407
33.000000		0.619204	3.600000	-0.028799	-0.053289
36.000000		0.591022	3.600000	-0.038834	-0.053171
39.000000		0.554487	3.600000	-0.048848	-0.053053
42.000000		0.509616	3.600000	-0.058840	-0.052934
45.000000		0.456426	3.600000	-0.068811	-0.052815

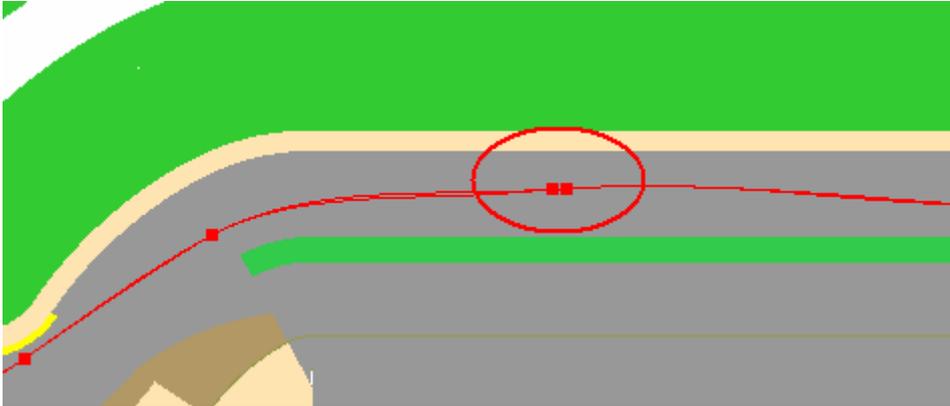
The column Control has several fields with a "1" in them. The 1's indicate a control point on the track layout. By altering the values in the latitude column (only in the line which has a 1 in the control

column!) you are now able to shift the racing line in the position you want. But before you do you need to know some basic rules for altering these lines.

AiEdit

2.5.1 Control Points

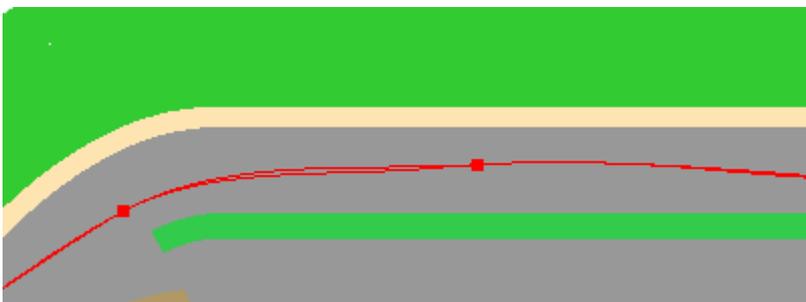
Once a race.lp (or Pit.lp) is converted it is possible that there are too many control points at the start of the lcv file and at the end. To smoothen things out you need to delete some of these points. So go to the start of the track.dat (select the 0 segment) and look for control points close to each other. Below is a sample of Jhall where AiEdit has created 2 control points close to each other:



Before deleting check which section you've seen these points. Select the race.lcv again and go to the longitudinal position you expect the points are.

Longitude	Control	Latitude	Longitudinal ...	Lateral Speed	Yaw Speed
30.000000		-2.069811	3.600000	0.071588	0.078931
33.000000		-2.002761	3.600000	0.090692	0.123583
36.000000		-1.916330	3.600000	0.118223	0.168148
39.000000		-1.803390	3.600000	0.154202	0.212815
42.000000		-1.656923	3.600000	0.198706	0.257733
45.000000	1	-1.469827	3.600000	0.251859	0.302996
48.000000	1	-1.276306	3.600000	0.213769	-0.181639
51.000000		-1.115589	3.600000	0.172334	-0.214296
54.000000		-0.988596	3.600000	0.132703	-0.205039
57.000000		-0.893877	3.600000	0.094877	-0.195683

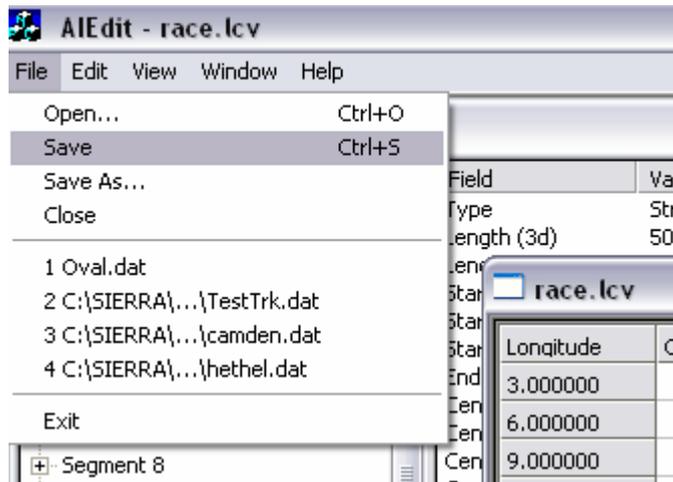
Now set the pointer in one of the control points and delete the one. You will see that a new line is being calculated. Switch back to the track.dat and press F5 (refresh)



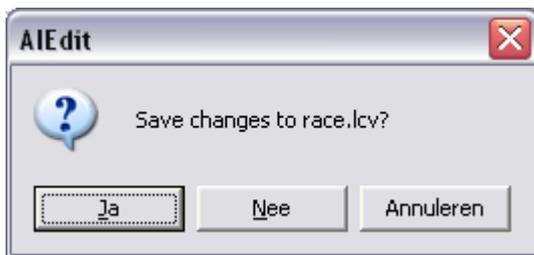
Notice that the race.lp line which stayed the same is visible under the new race.lcv line. Now check all segments to be sure there aren't more of these control points too close to each other.

AiEdit

Let's assume you've adjusted the racing line to your liking and created a pretty realistic driving line for the ai. It's time to save your work. With the Race.lcv window active press on File and Save.

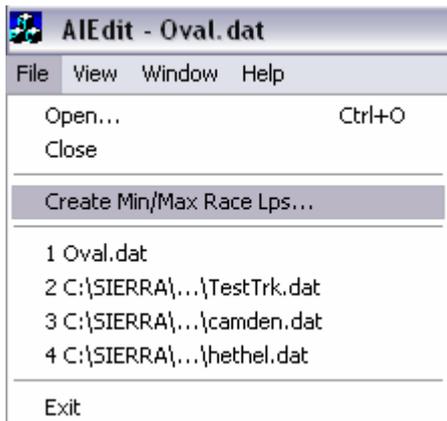


Confirm the next question



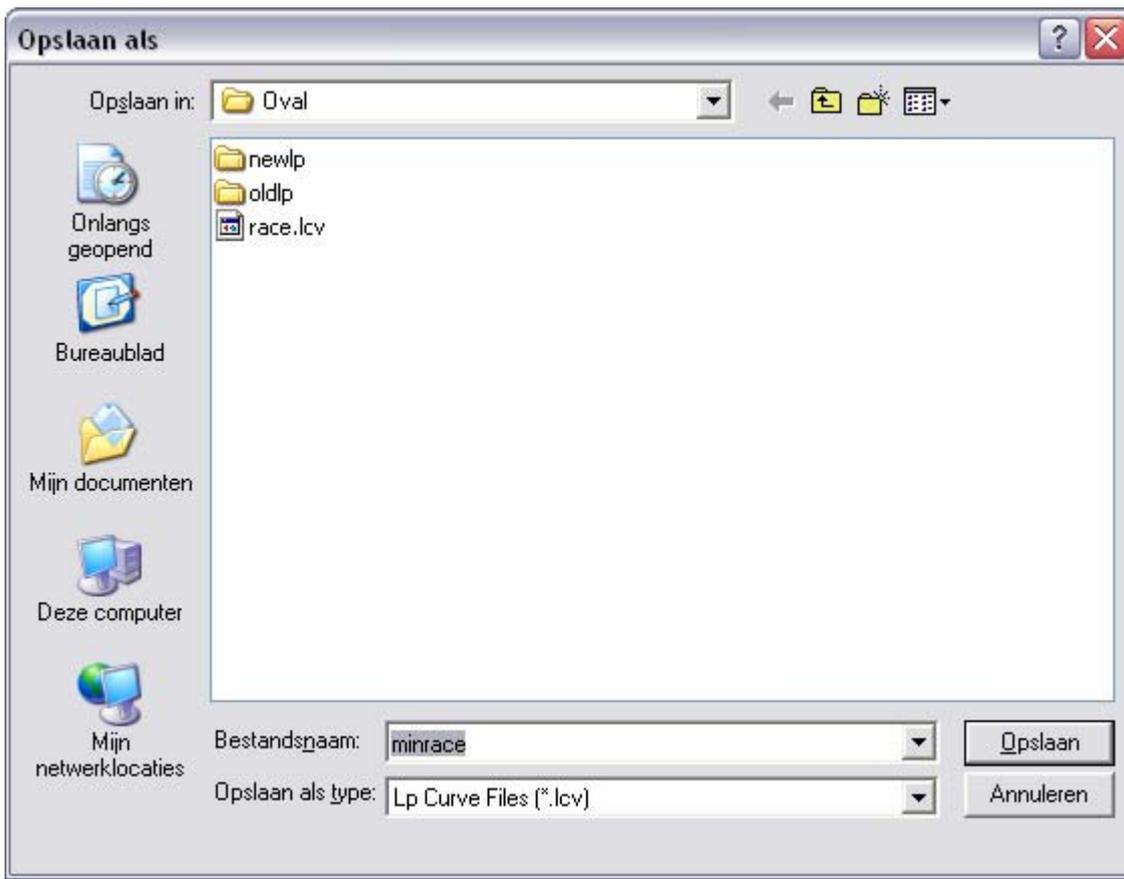
2.5.2 Min/Maxrace.lcv

It's time to create some Min/Maxrace files. Press on File and Create Min/Max Race Lps...



AiEdit

Click on the Save button to save the minrace.lcv file that is being created



If you don't get any errors (the program is trying to determine where there is room for the min and max race line. Again if the starting grid values or pit values in the ini file are way-off you could get some errors) You'll be asked again to save the file this time the Maxrace.lcv.

Notice that you now have 2 extra files on the bottom of your window.



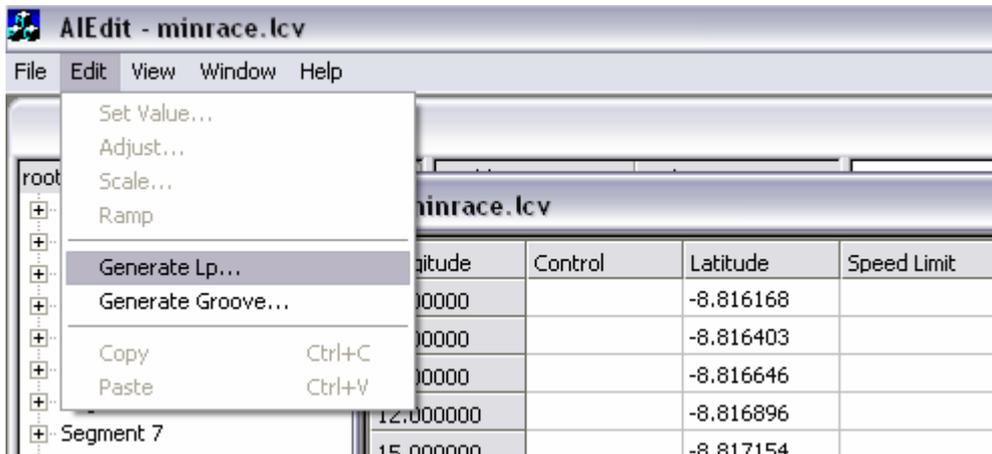
And that there have been 2 yellow lines added to the track :



You can examine them by selecting the segments on the track display one by one. If there are any strange lines present you can alter them the same way you did with the race.lcv file.

AiEdit

Click on the minrace window to enlarge the minrace window.
Select from the top menu Edit and click on Generate Lp



A new window is displayed.



You can accept the default settings and let AiEdit create the minrace.lp file.
Save the new created lp file by clicking on the Save button on the new displayed window.
Now do the same for the Maxrace.lcv.

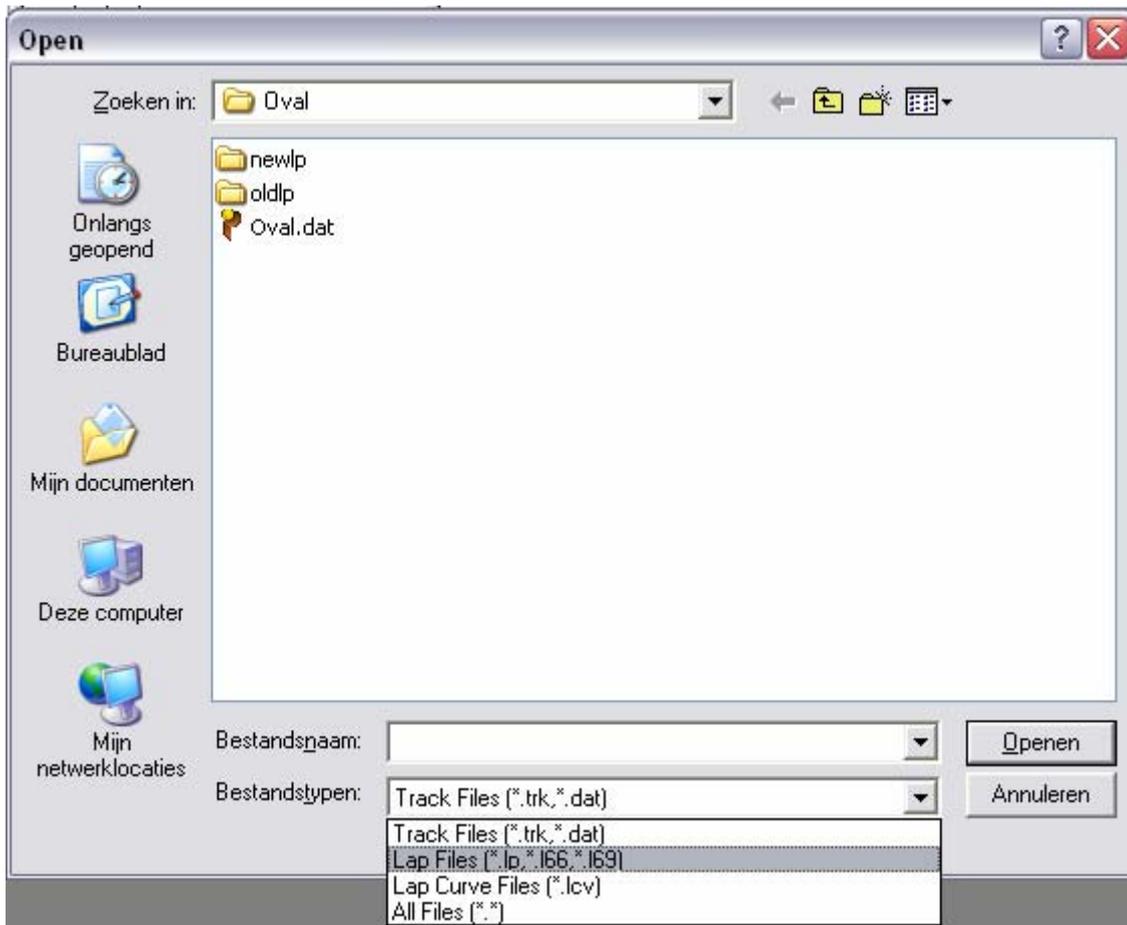
That's it! You've created a Min and Maxrace.lp file without actually having to drive the lines for them.

2.5.3 Pit.lcv

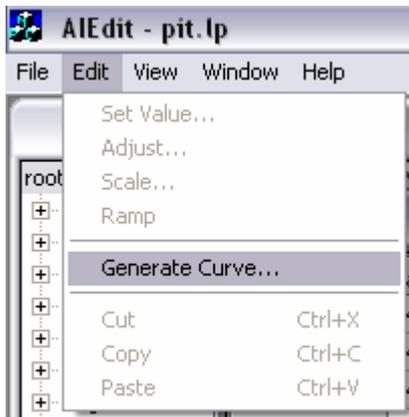
Before we start working on the passing files it's best to create a pitting line first.
This is so you can easily spot any problems with passing and racing ai running into a pitting ai driver.
For the basic line you use the Pit.lp you have created by rpy2lp. So click on File Open.

In the dialog window Open select in the file type: the .lp files.

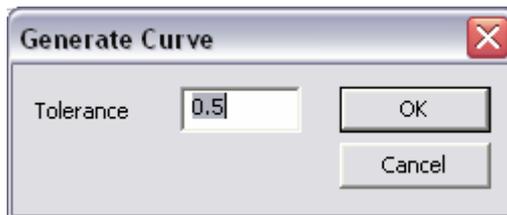
AiEdit



Once you have selected the .lp files the Pit.lp should be visible. Select it and click on Open. Click on the Pit.lp file in your window and maximise (or enlarge it) Click on Edit and Generate Curve to create a Pit.lcv file.



Accept the default setting for the tolerance:

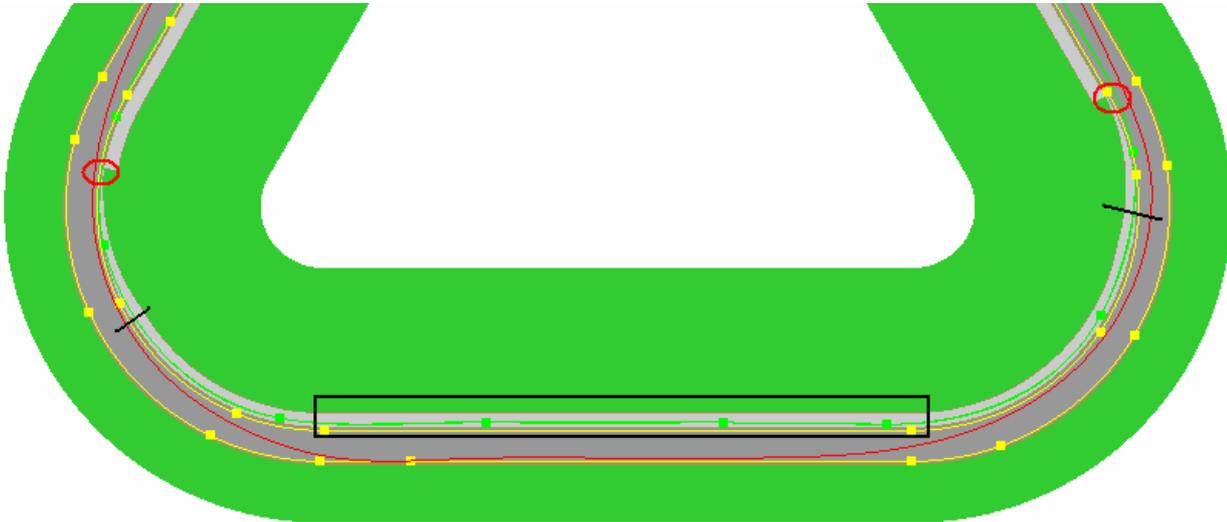


And press on Ok. You should now see a green pitline along the track. Before starting with passing lines it is necessary to check/alter the Track.ini file for the correct pit boundaries and inserting a pit speed limit on the Pit.lcv file.

AiEdit

2.5.4 Pit Boundaries and Speed limits

In the picture below I've placed the pit parameters on the track to show what position belongs were:



[ai_track]

merge_from_pit_line_dlong = 900.00 - way after pit exit (right red circle)
pit_at_end_of_track = 0.0 - Pitline crosses s/f line
pit_lane_end_dlong = 450.00 - See right **black** line here the armco (pit divider) ends
pit_lane_entrance_width = 8.0 - difference between 11.0 and 19.0
pit_lane_has_wall = 1.000000 - Yes, there is armco between pit and raceline
pit_lane_on_right = 0.000000 - Pitline is left of racingline
pit_lane_start_dlong = 1150.00 - way before pit entrance (left red circle)
pit_lane_width_left = 5.000000 - The difference between avg. Pitline lateral and pitwidth max. 19.0
pit_lane_width_right = 1.000000 - very low coss the pitline is at 14.2 and pitwidth min. is at 11.0
start_of_pit_stalls = 3.00 - [See pitlane_0] section

[pit_lane_0]

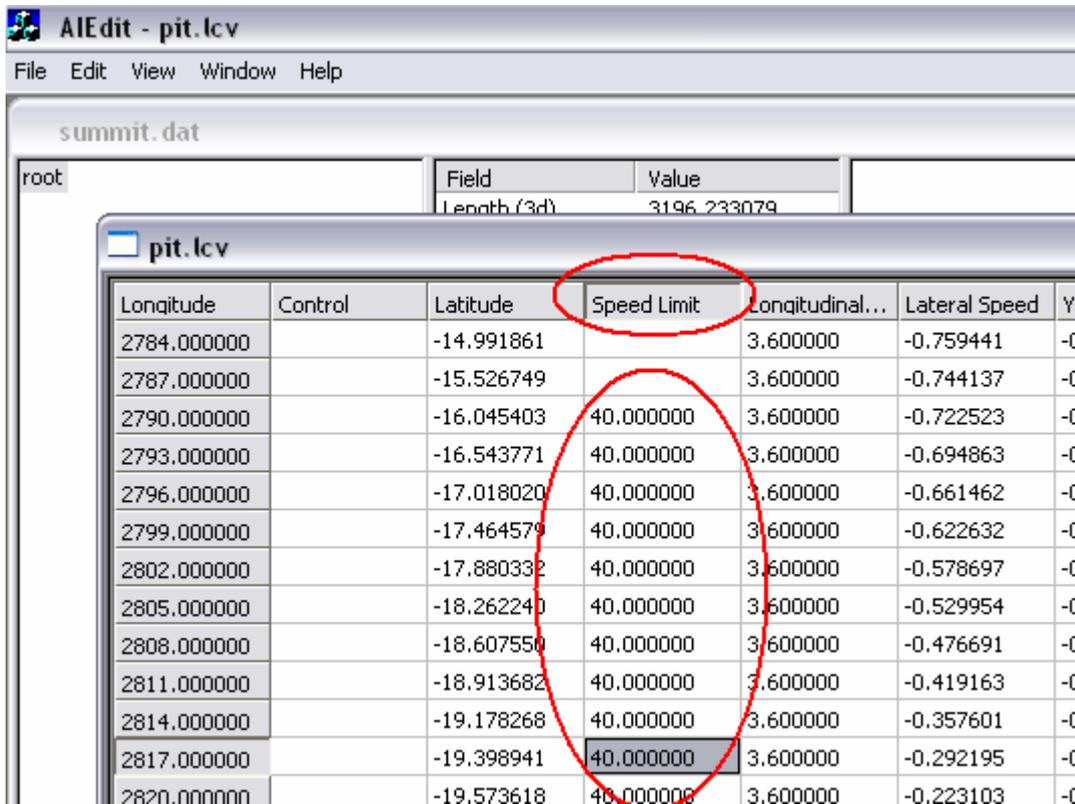
lane_on_right = 0 - See above
lane_has_wall = 1 - See above
lane_bounds_dlong = 1400.00 550.00 - See the left and right black lines
lane_bounds_dlat = 11.0 19.0 - Values found by clicking in segment (somewhere in the pitlane) on surface type

[pit_lane_0]

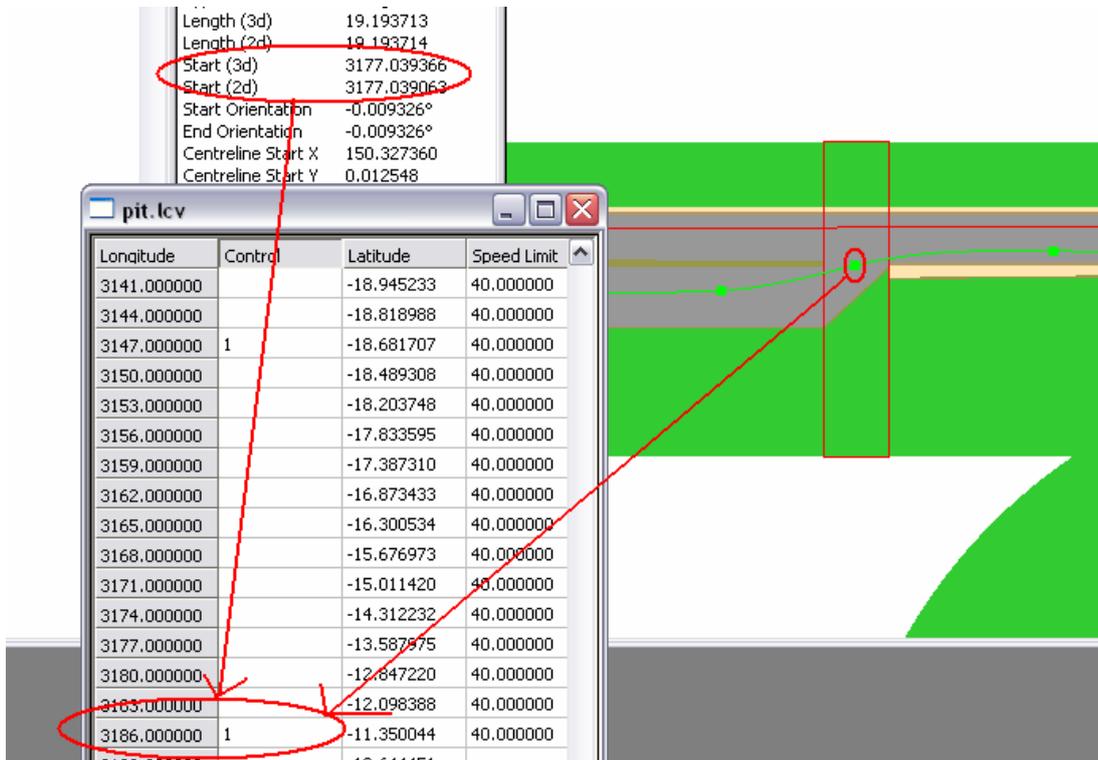
stall_0 = 255.00 18.20 - See **black** box drawn on pitlane
stall_1 = 241.74 18.20
.
.
stall_18 = 16.26 18.20
stall_19 = 3.00 18.20

AiEdit

To create a pitlane Speed limit you can use the Pit.lcv file. Check in the track section about were your pitlane starts and ends. Now type in the Speed Limit Column the value in kph. It is possible to select several lines(fields) in the column with your left mouse button pressed. Then hit Ctrl+C (or menu edit – copy) and point to the next empty field. Hit Ctrl+V (or menu edit – paste). This will save you time typing all the values in the designated pitlane fields.



Be moderate with the speed. Besides being unrealistic a too high pitlane speed will result in cars overshooting their pit stalls. End the speed limit halfway the pit entrance and exit. Cars will speed up and decelerate in a realistic way (see picture below)



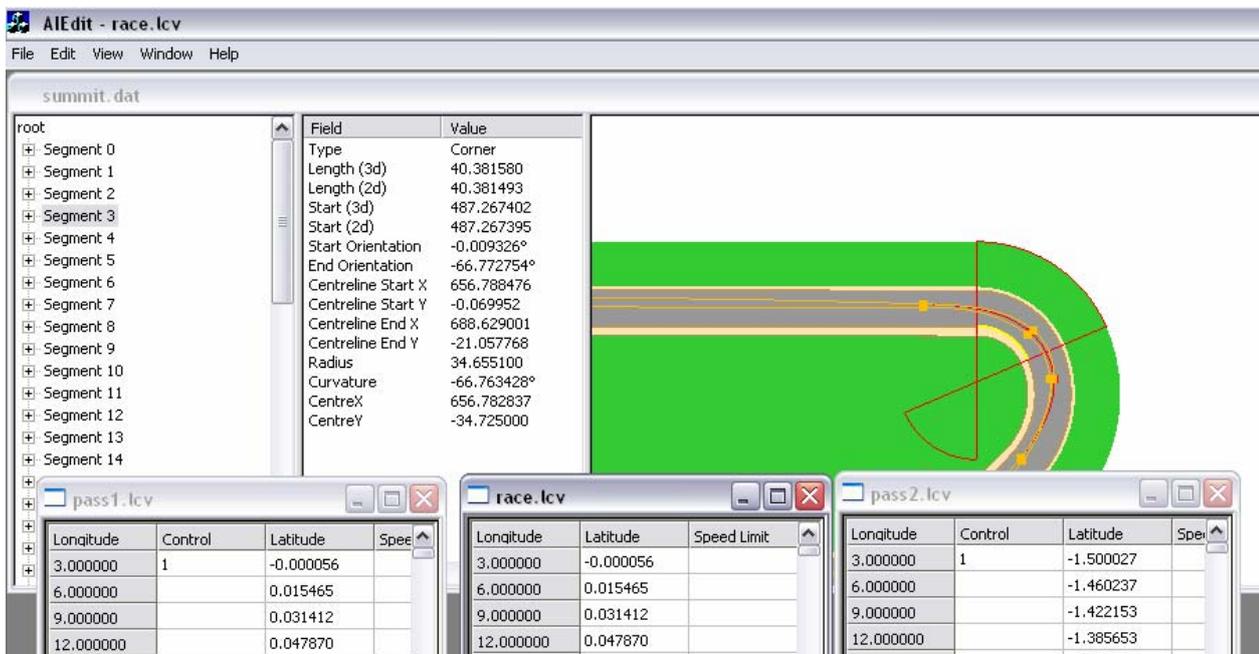
AiEdit

2.5.5 Passing lcv's

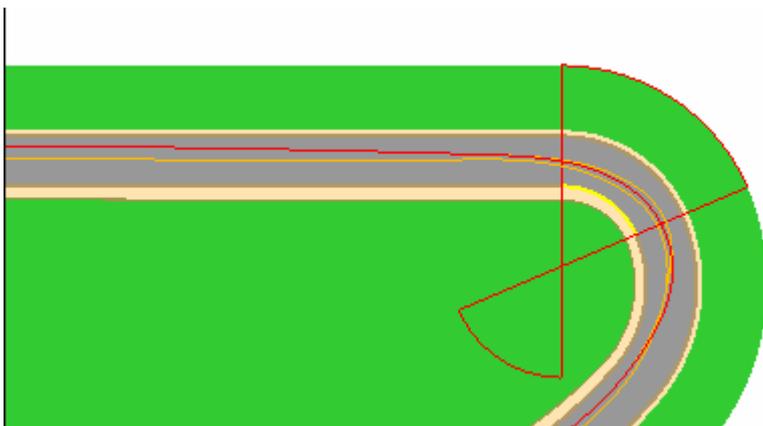
Right now comes the tricky part. If you examine ai files from some track makers you'll see that they just made a copy of the race.lp and renamed it to pass1.lp and pass2.lp. The race.lp would often also be missing the necessary flags. Now this might seem to work at first but it creates a number of problems. Why does this work? The way GPL works is that if the pass1.lp and pass2.lp follow the same lines it will calculate the difference between the race.lp lateral position and the min/Maxrace.lp lateral position. This value is to determine if a car can make an evasive manoeuvre in the case of havoc or if a flag is telling it that it can pass. This value is normally a lot higher than the passing line lateral position. So it's possible that an ai car will make such a movement that it loses control and flies off the track. Another problem is that on the s/f line there are no overtaking lines and flags so by the start of the race it's possible that a lot of cars stick to their position instead of using full throttle and trying to overtake the leading car. This often results in a traffic jam at the start of a race.

So before your contemplating on skipping these files and skip the flagging business please don't! Creating a set of simple passing lines and putting in some minor flags would be the least you can do now you've gotten this far ;-) Besides you would notice the difference in behaviour trust me.

To make life some what easier were going to use the race.lcv as a starting point for the passing lines. Go to your track map and make 2 copies of the race.lcv. Rename the files to pass1.lcv and pass2.lcv. Now open up AiEdit again and open the fresh created pass1.lcv and pass2.lcv. Set the control points to visible. Now place the 3 lcv files (race, pass1 & pass2) in this order on your screen:



Pass1.lcv is the passing line on the left, Race.lcv in the middle and pass2.lcv on the right. Now it's easier to determine which file to amend. (Note: need to buy a bigger screen sometime)



Now start altering the control points till you get a display similar like the one on the left. E.g. the race line closely followed by the passing lines. At some points it may be necessary to alter the race line a bit to leave room for a passing line. At some points you may have to decide if it's necessary to have a passing line or to just leave the line the same as the race line. In the twisty sections it's best to have only a race line.

AiEdit

After you've made the necessary changes on the passing and race lcv's you can generate the .lp files.

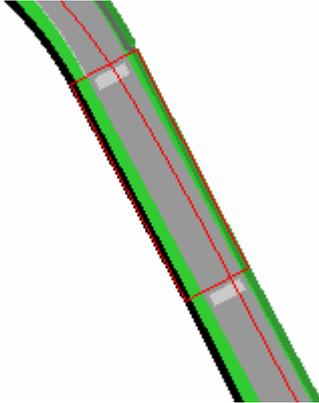
2.6 Implementing the flags

The last hurdle we need to take is to add the flags for overtaking, tire warm-up etc..

The functions of the flags are already mentioned in section 1.3 but how you should place them around the track is another thing.

Below you'll find some samples on how to place those flags on the track sections:

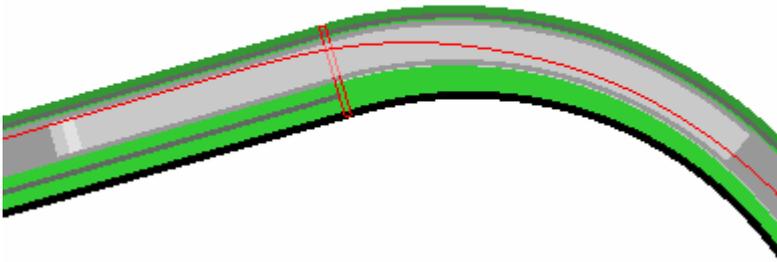
Start/End points:



race.lp								
Longitude	Latitude	Longitudinal...	Lateral Speed	Yaw Speed	Speed Modifier	Start Speed ...	End Speed ...	Outbrake D...
2607.000000	0.359500	138.736802	13.854240	-16.294919	0.020000			
2610.000000	0.686900	139.294082	14.761440	-10.313240	0.020000			
2613.000000	1.013300	140.317922	12.752640	-1.031324	0.020000	1		
2616.000000	1.276300	141.393595	10.238400	2.475178	0.020000		1	
2619.000000	1.423500	142.158238	8.916480	4.950356	0.020000			
2622.000000	1.606300	143.480161	6.998400	6.806739	0.020000			

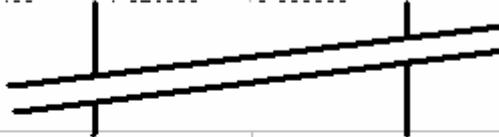
AiEdit

Outbrake Decision Point:



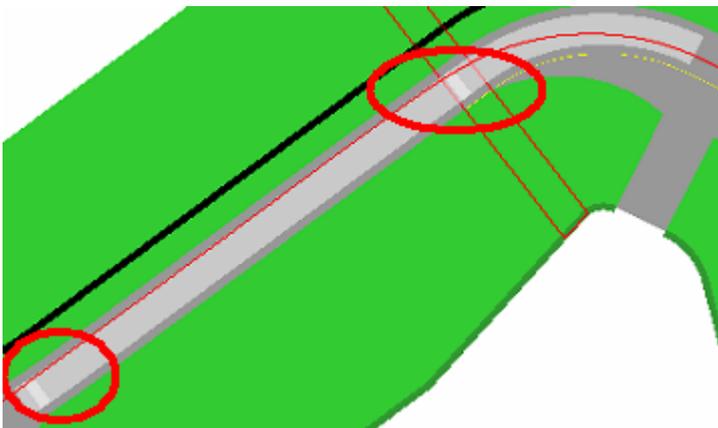
race.lp

Longitude	Latitude	Longitudinal...	Lateral Speed	Yaw Speed	Speed Modifier	Start Speed ...	End Speed ...	Outbrake D...
2403.000000	3.513400	176.670713	1.892160	-1.650118	0.020000			
2406.000000	3.540800	177.059516	1.684800	-1.443854	0.020000	1		
2409.000000	3.575000	176.839206	1.321920	-0.618794	0.020000			1
2412.000000	3.593300	176.152320	1.114560	-0.412530	0.020000			
2415.000000	3.608900	174.998874	0.959040	-0.206265	0.020000			
2418.000000	3.622300	173.132633	0.816480	-0.412530	0.020000			
2421.000000	3.633400	170.683204	0.686880	-0.618794	0.020000			



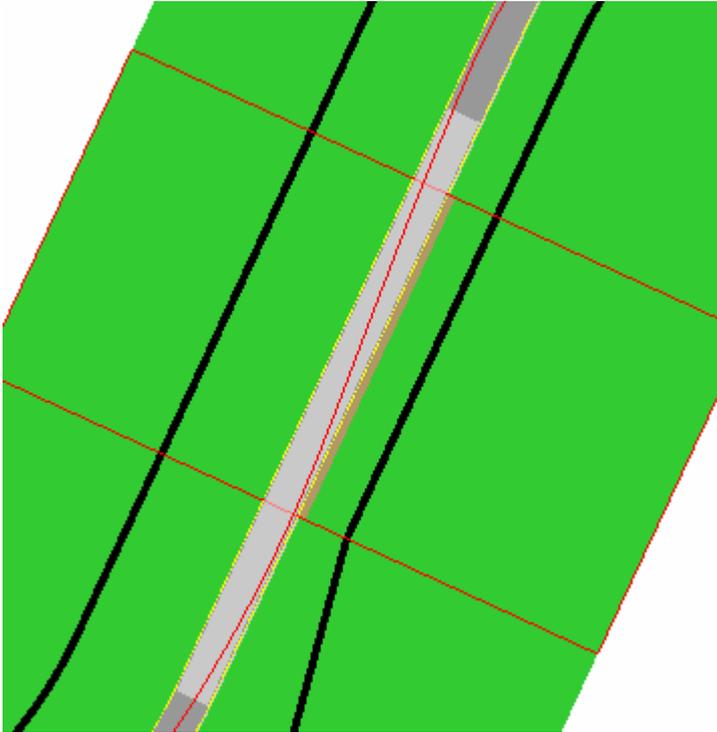
2568.000000	-2.195500	137.311197	4.199040	-16.294919	0.020000			
2571.000000	-2.087400	137.648167	5.339520	-18.563832	0.020000			
2574.000000	-1.953600	137.842553	6.350400	-21.657804	0.020000		1	
2577.000000	-1.852400	137.933272	6.907680	-21.657804	0.020000			
2580.000000	-1.680100	138.011045	7.698240	-21.657804	0.020000			

Use of outbrake flags with a different approach. This time there is a second outbrake flag just before the corner.

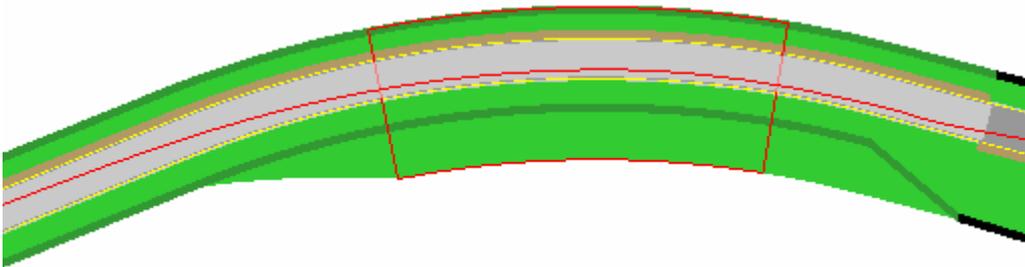


Long Straights:

Can also be used for short straights:



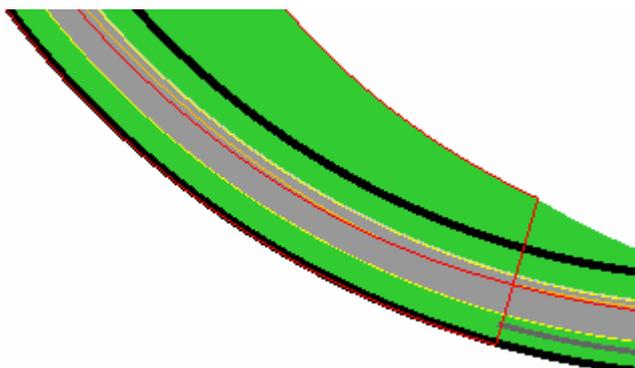
Can be used for long straights in combination with high speed corners:



Right that's it for the flag positioning the rest off them are pretty obvious.

2.7 Tips and hints

You'll discover that the majority of the open tracks won't give you much trouble in creating an ai set without too much crashes/accidents. Armco surrounded tracks are a whole new ballgame though. Most crashes happen when an overtaking car is following a passing line that approaches the racingline again (see image)



The overtaking car forces the race line car to the outside of the track. If there is armco it will run into it. Sometimes it might be possible to move both lines more to the center of the corner so both cars can drive through it. But that might not be a very realistic driving line. Another option could to shift the overtake(flags) zone a bit so passing will be executed before the lines cross. Another possibility is using speed modifiers to speed up overtaking (realistic at monaco but not at open tracks)

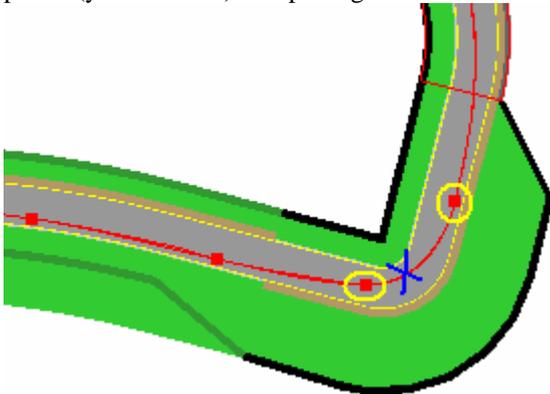
AiEdit

Sometimes it is better to reduce the number of control points to smooth out the racing line a bit.

You'll notice that cars driving into corners might be hesitating a bit or shifting from left to right and vice versa.

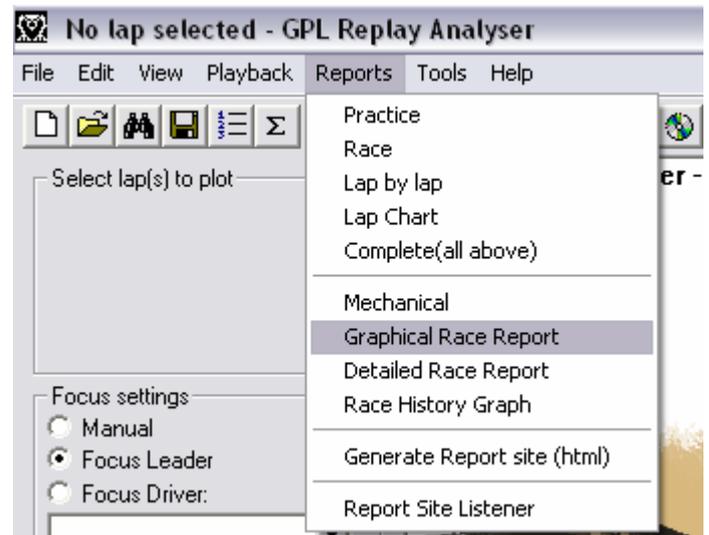
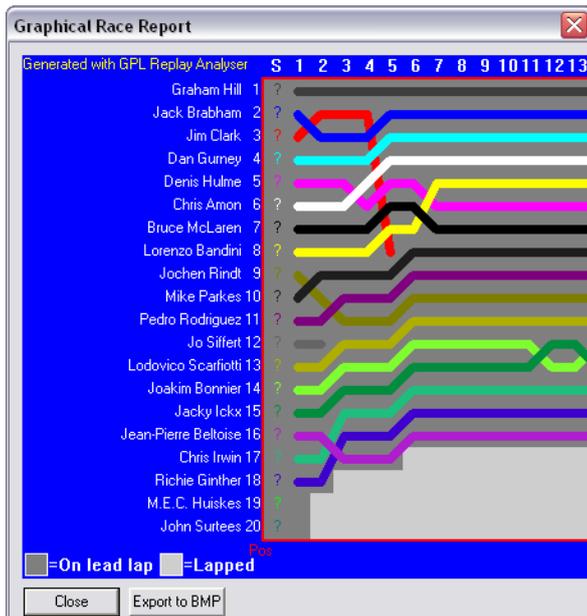
If this happens the control points could be too close to each other. Another possibility is that the points you created are not on the ideal racing line. If this is the case you can try creating a new control point (blue cross) in between and delete the 2 enclosing points (yellow circle) Just putting in the new control point is not enough you need to alter the lateral position as well.

This is a bit of trial and error. Sometimes it's even better to delete a few points to see what the calculated line looks like and then fiddle a bit with new created points to restrain the racing line from leaving the track.



Testing and modifying your set is the next step.

Just run some races in Novice first to see how they behave. After the race is done use the race report to write down which drivers had crashes. Now save the replay and open it in GPLReplayanalyzer. Start a graphical race report and write down on which lap the accident took place for the drivers you wrote down earlier



You can export the bitmap to view it as a whole for the longer races.

Now you can start on amending the .lev files or flags to reduce crashes or improve driving lines.

Important: If your making a track and consider doing it the easy way by making just the race, pit and min/max lp's....please don't. I've explained before why you shouldn't (overtaking, racestart etc..) but more important...if you spend a huge time on making a track what's a couple of days work more fore creating some good ai?

Well that's it basically =)