Scifi Racing Team Modding Guide

Where are mods stored?

Mods are stored in the ‘mods’ folder in the game files. To access it from Steam, go to the game page in your library, click on the cogwheel, go to manage > browse local files and then go to the ‘mods’ folder.

How to create my own mod?

Once you get to the ‘mods’ folder, create a new folder for your own mod. Everything we are going to talk about in this guide will refer to what is contained within that new folder you just created.

Mod structure

Inside your mod folder, you will have to make one folder per mod component (teams, sponsors, etc.), one text file that describes the mod’s basic information, one text file to be used for a complete description (this one is optional) and one image that will act as the mod’s thumbnail.

Here are the different folders you can make (note that you must respect the casing):

`championships`
`circuits`
`default_logos`
`helmets_designers`
`helmets_drivers`
`liveries`
`manufacturers`
`racepod_meshes`
`sponsors`
`teams`

track_meshes

The text file that describes the mod's basics must be named `info.txt` and its contents look like this:

```
name    "Mod name"
version "Some version"
author  "Author name"
visibility public
```

All you have to do is to replace `Mod name` with your own mod name, `Author name` with the mod's author's name and `Some version` with whatever version of the mod you want (it could be `v1`, `1.0`, `2025-05-19`, etc.).

`visibility` lets you decide whether or not your mod should be public when you upload it to the Steam Workshop. You can set it to `public` or `private`.

*Important: once you settle on a mod name, **do not change it**. This could break people's saves.*

The text file that can be used for a full description of the mod must be named `description.txt` and its contents are literally the mod's description.
This will be used as the mod's description if you want to upload your mod to the Steam Workshop.

The thumbnail you want to use for the mod must be named `preview.png`

Championships

Championships live in the `championships` folder. To create a new championship, first create a new folder within the `championships` directory (name it with whatever you want, **but do not change that name once you ship the mod** since that could break people's saves).

There are two files you must place within the new folder you just made:

- `general_rules.txt`
- `logo.png`

`general_rules.txt` is a text file detailing the rules of the championship and `logo.png` is a PNG file that will act as your championship's logo.

The `general_rules.txt` contents look like this:

```
chassis_rules must_be_built_by_the_team
engine_rules  do_whatever_you_want

points_awarded_for_fastest_race_lap 2

max_points_awarded_per_win   20
last_place_to_receive_points 5
points_distribution          square

max_points_awarded_for_qualifying           2
last_place_to_receive_points_for_qualifying 1
qualifying_points_distribution              linear

num_qualifying_laps 5
num_laps_per_race   60

min_seed_capital_for_bot_team 50000000
max_seed_capital_for_bot_team 400000000

max_sponsor_money_per_season 30000000
max_sponsor_money_per_race 1000000

prize_pool         600000000
prize_distribution square
```

```
entry_fee_per_point_scored_in_previous_season 200000

min_number_of_teams 8
max_number_of_teams 15

max_races_per_season 6
```

This is directly taken from the `general_rules.txt` file used for the Class A championship.

Chassis and engine rules

`chassis_rules` and `engine_rules` let you decide whether teams or manufacturers are responsible for supplying the chassis and the engine. The different values you can use are the following:

- `must_be_built_by_the_team`: the part must be built by the team.
- `do_whatever_you_want`: a team may get that part from a manufacturer, a team or may build it themselves.
- `suppliers_only`: teams must get that part from a manufacturer of their choosing.
- `single_supplier`: there is a single manufacturer supplying all teams with that part.

Points awarded after a race

There are two ways teams may score points out of a race:

- By doing the fastest lap of the race.
- Depending on their finish position.

`points_awarded_for_fastest_race_lap` lets you set the number of points the team with the fastest lap of the race would score (this can be `0`).

`max_points_awarded_per_win` lets you set how many points the winner would score (this can be set to `0`).

`last_place_to_receive_points` lets you set the last place to receive points (set this to `0` if you want everybody to receive points at the end of a race).

`points_distribution` lets you control the number of points distributed per team. There are 3 possible values you can use:

- `linear`: the number of points will linearly decrease, which means the number of points you get is directly proportional to your position.
- `square`: the number of points will dramatically decrease, which means high positions (like the winner) will get a lot of points and low positions will only get a few.
- `square_root`: the number of points will slowly decrease, which means everybody will get a similar amount of points except for the teams in very low positions.

Points awarded after qualifying

You can score points based on your position after the qualifying session.
The logic is identical to how you score points based on your position after finishing a race.

`max_points_awarded_for_qualifying` lets you set how many points the top team in qualifying would get (this can be set to `0`).

`last_place_to_receive_points_for_qualifying` works the same way as `last_place_to_receive_points` described previously.

`qualifying_points_distribution` works the same way as `points_distribution` described previously.

Number of laps

`num_laps_per_race` lets you set the number of laps to do during a race.

`num_qualifying_laps` lets you set the number of laps you allow teams to qualify (it should be set at least to `2` and should be less than what you set for `num_laps_per_race`).

Money

You can use `min_seed_capital_for_bot_team` and `max_seed_capital_for_bot_team` to set the amount of money a brand new bot-controlled team will start with when they enter the championship.

Use `max_sponsor_money_per_season` and `max_sponsor_money_per_race` to set the maximum amount of money sponsors will give to their teams at the beginning of the season and at the end of every race.

`prize_pool` is used to set the total prize money available at the end of a season.
`prize_distribution` is used to determine how the prize money is distributed amongst teams. The possible values are the same as described for `points_distribution`.

*Note: **prize money is also used to determine a championship's prestige**. The championship with the biggest prize money will also be the most prestigious and this is where all the designers and drivers will want to transfer to.*

`entry_fee_per_point_scored_in_previous_season` describes the amount of money a team will have to pay per point they scored this season if they want to participate in the next season (e.g. if a team scored 10 points and the entry fee per point scored is $1,000, the team will have to pay $10,000 to participate in the next season).

Miscellaneous

`max_races_per_season` is used to set the number of races on the season's calendar (its minimum value is `1`).

`min_number_of_teams` and `max_number_of_teams` are used to determine the size of the grid.

Circuits

Circuits are stored in the `circuits` folder.
To create a new circuit, create a new folder in the `circuits` directory (name it whatever you want **but do not rename it once you ship your mod** to not break people's saves).

In the newly created folder, you need to things:

- The circuit layout.
- Information about the circuit.

Circuit layout editor

To create a circuit layout, you need to use the circuit editor. To access this editor, you need to choose the 'Circuit editor' launch option when launching the game from Steam.

The editor lets you design a layout using control points. Each control point represents a turn. You can drag them around, add them and delete them.

The game needs to know how big the circuit is and where a lap starts from.
This is done by setting up the grid's boundaries. You need to set two boundaries (you can insert them by first pressing G/E, depending on the boundary, and clicking somewhere on the circuit).

The game will use the distance between the boundaries as a base unit to compare the rest of the circuit against. This is how it knows how big the circuit is.

The green arrow will tell you the circuit's direction. The default circuit layout is anti-clockwise, so if you want to make a clockwise circuit instead you will have to shuffle everything around.

Once you are done with the layout, save the file to the folder you created for your circuit (it must be named `layout.circuit`).

Circuit information

Circuit information should be contained in a file named `info.txt`.
Its contents look like this:

```
name "Circuito de Acatavia"

oxygen_range              [0 .2]
oxygen_variation_strength .3

temperature_range              [-1 -.5]
temperature_variation_strength .5
```

This is directly taken from the core game.

`name` is the name of the circuit.

`oxygen_range` lets you set the lowest and highest possible amounts of oxygen. The first number is the lowest possible amount and the second is the highest. The numbers should be comprised between `0` and `1`.

`oxygen_variation_strength` lets you decide how much oxygen variation there can be, `0` meaning no variation and `1` being maximum variation.

`temperature_range` is similar to `oxygen_range` except the minimum possible value is `-1`. `-1` represents maximum cold, `1` represents maximum heat.

`temperature_variation_strength` works just like `oxygen_variation_strength`.

Default logos

Default logos are logos the player can use when setting up their team. They live in the `default_logos` folder.
To add a new default logo, just copy the PNG file that represents the logo to this folder.

Do not rename the files once you ship your mod since this could otherwise break people's saves.

Driver and designer helmets

A helmet is a PNG file that must respect certain rules regarding the way they use colours. All you need to do is to copy that image within the `helmets_drivers` or `helmets_designers` folder, depending on whether this is a driver helmet or not.
You should use **square images** for best results.

Do not rename the files once you ship your mod since this could otherwise break people's saves.

There are two aspects to a helmet:

- The gloss (light reflection, sort of).
- The different colour patterns.

The game will apply up to 3 different colours to a helmet so that two helmets that use the same image will actually look different. Now, how do you store all that information (meaning the gloss and the colour pattern)?

This is done by using the different colour channels.
Each pixel of the image is made out of 4 channels:

- The red, green and blue channels, which are mixed together to form the final colour (think of it as mixing red, green and blue paint in different quantities).
- The alpha channel, which tells how transparent the pixel is.

Colour pattern

The red and green channels are used to determine the colour pattern of a helmet.

If both the red and green values are set to 0, the result is black.
If the red channel is set to its maximum value while the green channel is set to 0, the result is red.
Similarly, if the green channel is set to its maximum value while the red channel is set to 0, you get green.
And finally, you can set both channels to their maximum value which would yield yellow (mixing red and green).

So this gives you 4 colours to play with: black, red, green and yellow.

Obviously, the game will not literally display red, green and yellow, it will detect that they are separate colours and then replace them with randomly generated colours to get the final look (for instance, red could be replaced with some shade of blue, green could be replaced with white and yellow could be replaced with purple).

Black is never swapped with another colour, so you could use it for the helmet's visor or goggles, for instance.

Gloss

To get glossy reflections on the helmet, you can provide the gloss values using the blue channel. If you do not want to go for a realistic look, you can set the blue value of every pixel to 0.

How to do this with 2D software

If you want to go for a cartoon/2D look, you can easily draw your helmet using black, red, green and yellow. Do not use white, blue or purple.

How you would do this with 3D software

If you want to make a render of a 3D helmet, you can do a first render *without* reflections, where you only get the colours (they should be black, red, green and yellow, do not use white, blue or purple), and then you would do a render *with reflections only (specular)*, which should yield a greyscale image.

Then, you can mix both renders by taking the colour values of the first one and feeding them into the red, green and alpha channels of the final image and then you feed the red channel of the second render (the one that contains the reflections) into the blue channel of the final image.

Liveries

Each livery is stored as a PNG image in the `liveries` folder.

A livery is made out of 2 colours: red and black. Each colour represents the primary and secondary colour of a team.
The game will automatically swap them out for the team's colours.
You should probably use high-resolution liveries (1K, 2K), especially if you have intricate patterns.

Liveries are always directly projected onto a vehicle from above.

Do not rename the livery files once you shipped your mod since this could otherwise break people's saves.

Manufacturers

Manufacturers are stored in the `manufacturers` folder.
To create a new manufacturer, create a new folder within the `manufacturers` directory and name it whatever you want (**but keep that name once you ship your mod** since otherwise this could break people's saves).

Similar to a championship, there are two files you need to put in the folder you just created:

- `info.txt`, which contains information about the manufacturer.
- `logo.png`, a PNG file that is used as the manufacturer's logo.

The logo **must have a transparent background** for best results. You will set the background colour for that manufacturer in `info.txt`

`info.txt` contains the following:

```
name       "Braun Powertrains"
short_name "Braun"

services "engine"
background_colour_srgb 0xfffe00
```

This is directly taken from the manufacturer "Braun Powertrains" in the core game.

`services` lets you specify what parts the manufacturer should be able to supply.
For a chassis-only manufacturer, set the value to `"chassis"`.
For an engine-only manufacturer, set the value to `"engine"`.

For a manufacturer that supplies both, set the value to `"chassis" "engine"` (the order does not matter).

`background_colour_srgb` is used to set the background colour of the manufacturer's logo. You can use a hexadecimal value.
In the example above, the blue value is `00`, the green value is `fe` and the red value is `ff`. Together they form the hexadecimal number `0xfffe00`.

Note: always put the `0x` prefix so that the game knows this is using the hexadecimal format, so for instance if your colour picker gives you `#fffe00`, replace the `#` with `0x`.

Vehicles

Vehicles are stored in the `racepod_meshes` folder.
Each vehicle data is stored in a Wavefront OBJ file (`obj`).

Note: make sure that once you ship the mod you do not change the name of these files to not break people's saves!

The game expects the `obj` file to contain a few things:

- The vehicle mesh with two materials applied, one for the general body and one for the cockpit.
- Quads that represent the different logos that are going to be projected onto the vehicle from above.
- An object that represents the point below which logos should not be drawn on the vehicle.

The game will automatically detect which object is which by using the following nomenclature:

- `pod`: this represents the vehicle. It should have two materials applied: one named `body` for the overall body and one named `cockpit` for the cockpit. These materials will tell the game how to render the different parts of the vehicle (the specifics of the materials you assign do not matter).

- Any object starting with `decal_engine_supplier` represents where on the vehicle the logo of the engine supplier should be drawn.
- Any object starting with `decal_main_sponsor` represents where on the vehicle the logo of the main sponsor should be drawn.
- Any object starting with `decal_sponsor_1`, `decal_sponsor_2`, `decal_sponsor_3` or `decal_sponsor_4` represents where on the vehicle the logo of a secondary sponsor should be drawn. `1` represents the biggest secondary sponsor and `4` the smallest one.
- `decal_clipping_plane`: the position of this object along the up axis specifies the point below which no logo should be drawn on the vehicle.

Important: logos are always projected from the top along the vertical axis, which means all the objects that are used to position logos should only be rotated along the vertical axis to ensure a correct projection (unless you want to have some fun with the way the final projected result looks).

Sponsors

Sponsors are stored in the `sponsors` folder.
To create a new sponsor, create a new folder in the `sponsors` directory (named whatever you want **but keep that name once you ship your mod** so as not to break people's saves).

Within that directory you just created you will have to supply two items:

- `info.txt`, which contains information regarding the sponsor.
- `logo.png` which represents the sponsor's logo.

Similar to manufacturers, **the logo must use a transparent background for best results** (you can set the background colour in `info.txt`).

`info.txt` contains the following:

```
name       "Amarillo Group"
short_name "Amarillo"
```

```
size 4
background_colour_srgb 0xe3ea00
```

This is directly taken from the sponsor “Amarillo” from the core game.

As you can see, this is quite similar to a manufacturer, with the only difference being the `size` value. This value (comprised between `1` and `4`) tells the game how big this sponsor is (`4` being the biggest), and by extension how much money they give out and what their expectations are.

For `background_colour_srgb`, refer to what is explained in the manufacturers section.

Teams

Teams are stored in the `teams` folder.
To create a new team, create a folder within the `teams` directory (name it whatever you want **but keep this name once you ship the mod** to not break people’s saves).

Within that newly created folder, you must supply two files:

- `info.txt`, which contains information about the team.
- `logo.png`, the team’s logo.

*Note that unlike manufacturers and sponsors, team logos should **not** have a transparent background.*

`info.txt` contains the following:

```
name       "Aegis Performance"
short_name "Aegis"

seed_capital   3
has_an_academy 1
```

`academy_driver_ability_to_focus_on cornering_speed`

This is taken from the team “Aegis Performance” from the core game.

`seed_capital` represents the size of the team (and by extension how much money it starts with). Possible values can be `1`, `2`, `3` or `4`.
`4` is the biggest size.

`has_an_academy` lets you decide whether that team should have its own academy or not (a value of `0` means the team will not and a value of `1` means the team will).

`academy_driver_ability_to_focus_on` lets you decide which driver characteristic will be boosted when a driver is generated for the first time in the team’s academy.
Possible values are:

- `straight_speed`
- `cornering_speed`
- `overtaking`
- `defending`

Track backgrounds

Track backgrounds are stored in the `track_meshes` folder.

A track background is the blurry background stuff you see when a vehicle is displayed (for instance in the main career menu, in the championship menu, in the podium menu, etc.).

A track background is made of a 3D object with textures applied on it.
The 3D object should be exported in the Wavefront OBJ format.

To add a new track background, simply copy the `obj` and accompanying `mtl/png` files that contain the 3D data in the `track_meshes` folder.

Important
The game uses a right-handed coordinate system with the Z axis (+Z) as the up axis and the X axis (+X) as the forward axis, so bear that in mind when you export your mesh.

Make sure there is enough space in the middle of the mesh so that the game can display a vehicle in there.

Only colour textures (not roughness and normal maps) will be taken into account by the game when it comes to colours/materials, so no need to tweak the material you are using for the mesh.

Note on images

Apart from liveries, helmets and images used by track backgrounds, **all** images you provide will be resized to 1024x512.
An image will **not** be distorted in any way. If its aspect ratio is different than 2:1 then the image will be extended.

How to check for errors

To test your mod, launch the game. If you want to see if there are errors, check what the terminal displays (errors should appear in red).
The terminal is a separate window associated with the game. You should be able to use the `alt + tab` key combo to access it.

How to pack your mod

Your mod will be packed every time you launch the game as long as there are no errors.
The packed mod is a single file named `.mod` that you will find in your mod's folder.

How to distribute your mod

Generic

If you want to simply upload the mod somewhere and let people download it, you can retrieve the `.mod` file generated by the game in your mod's folder (you should give it a name, like `funky_liveries.mod` for example). People can then install the mod by copying that file into their `mods` folder.

Steam Workshop

To upload your mod to the Steam Workshop, launch the game, go to the "mods" menu, and navigate to the "My mods" tab.
If your mod is ready, it should appear there and you can upload it by clicking on the "Share this mod" button. You may be prompted to accept the Steam Workshop agreement.

Note that you need to do this every time you update your mod.