

How to make an autopainter

By système

Step 1 : The shortcut

Before we begin, make sure to create a shortcut of your battlezone game leading to your map, that way you will be able to open your map without much trouble. To this extent, you need to add in the properties menu, at the end of the target, your map name and /startedit or /edit if your map does not have time-related events (Figure 1). Once done, simply launch your game through the shortcut.

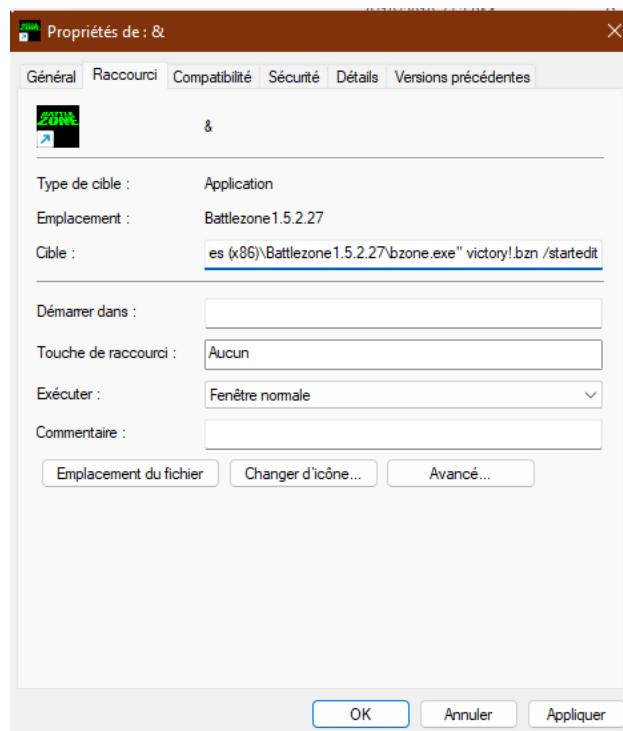


Figure 1 : inserting the right properties in a shortcut

Step 2 : The terrain editor

Once in editor mode, you will have to press ctrl e to switch to terrain editing. This step will be very valuable to acquire heights and verify which textures are used. During this step, you will need to figure the four (4) following points :

- What textures are used (Figure 2, Figure 3)
- Where (if you're editing a map that someone else has made, keep in mind that they could have edited some textures manually (Figure 3))
- Slopes
- Elevations

By pressing ctrl e, the texture terrain editing shows first, that way we can see exactly what texture was used and where. Unfortunately, they're not displayed by name or type, so we need to guess through based on their order in the map.TRN folder. In general they appear in order, the first (and probably the lowest) texture used is texture 0, the second is 1 etc.



Figure 2 : Observing textures in terrain editing



Figure 3 : As we can see in this case, the previous editor has manually painting at least one area (river texture)

After finding exactly the right spot for each texture, we can shift to height editing, by pressing z. The screen would turn into grid mode, in order to allow us to better see the height differences. Moreover, this mode does also precisely indicate the different heights encountered on your map on the bottom right section of the screen (Figure 4). Unfortunately, slopes cannot be clearly identified and need to be estimated either with the naked eye or through precise calculus.

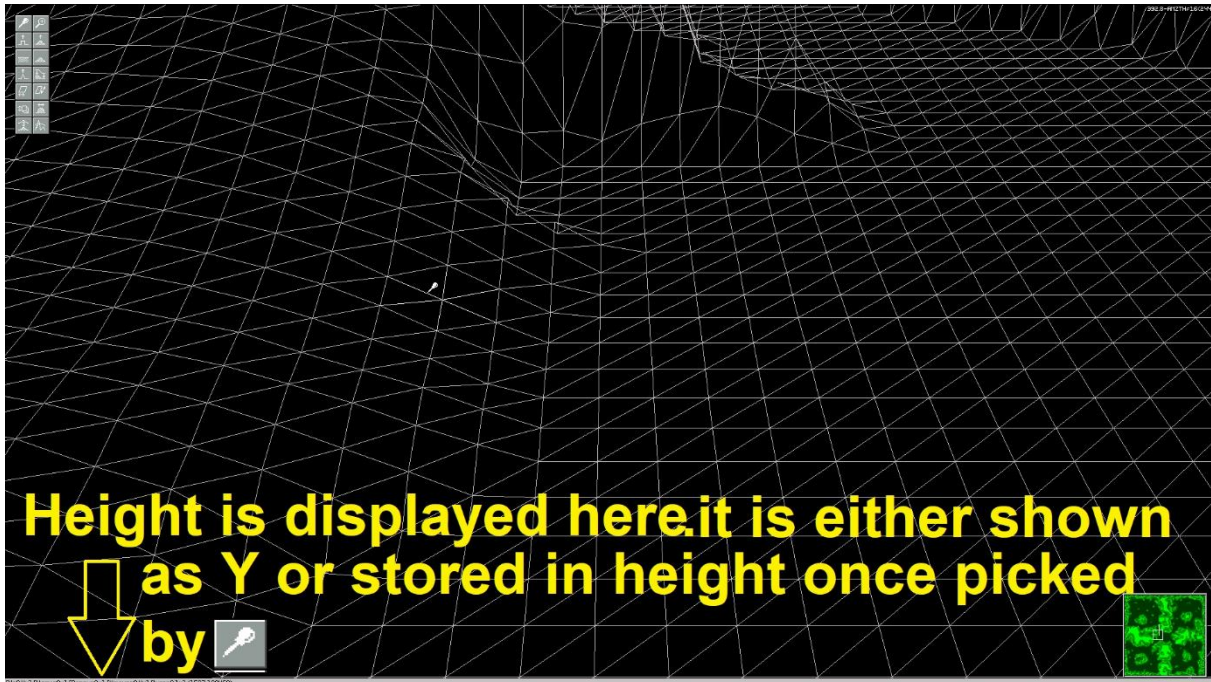


Figure 4 : The terrain editor indicating the elevation of the point shown by the cursor

Step 3 : Preparing the autopainter

Once we have acquired all the heights (either by noting them somewhere or by remembering them), it is time to type the ini file. It is better if you give a unique name for your ini and put it in the same folder as your map, so maketrn.exe can find it and we don't get confused. Usually, the name could be your map name or, if you are planning to use it for several maps, something more generic like terrain1, a custom planet of your choice or even a world already used but make sure you can separate it from already existing ini (io2.ini to recognize it from the already existing io.ini for example). This particular text file consists of the minimum amount of elements required for the painter to work, as shown on Figure 5.

```

[Layer0]
ElevationStart = 0
ElevationEnd   = 147
SlopeStart    = 0
SlopeEnd      = 15
Material      = 0

[Layer1]
ElevationStart = 0
ElevationEnd   = 1000
SlopeStart    = 15
SlopeEnd      = 90
Material      = 3

[Layer2]
ElevationStart = 147
ElevationEnd   = 1000
SlopeStart    = 0
SlopeEnd      = 15
Material      = 4

```

Header

Slope

Elevations

Material

Important: all elevations encountered must be covered by a layer. The same applies to slopes. maketrn does not have the ability to "guess" nor try to fill the gaps based on the closest elevation/slope

Figure 5 : Contents of a standard ini file

The first part of your ini is the header. This is used both as a separation between your texture and to inform the painter which layer to use. Unlike in other files, the header is actually used by maketrn, so make sure all information is correct. The first header is the first layer and therefore layer 0, it is distinct from the material used. This latter is the actual texture used, based on your TRN. Its number must correspond to the order in which textures are put inside the TRN :

- Example 1 : the first texture type, which is the highest in your trn folder, marked as [TextureTypeX], even if it is not necessarily identified as [TextureType0], is the one corresponding to material 0.
- Example 2 : the second texture in the TRN could be indicated as [TextureType4], but it would still match material 2)

Then, we have to put the elevations we recorded in game, elevationstart is the lowest height you want to be painted using that texture and elevationend is the highest. Last but not least, we have to indicate the slopes. They correspond to the angle (in degree), at which your texture can be found (example : you want your material 1 to be a road that can be up to 40°, slopestart must be 0 and slope end must be 40, if you want a wall representing unpracticable terrain you can start it from 40 and end it at 90, the maximum angle encountered in game). The fact elevation and slope are independent allows us to have the same texture on several heights, which we can illustrate by these examples, though many can exist :

- Example 1 : in the previous example, if you want your wall to be present on most places in the game, you can set its minimum elevation to 0 and the maximum to 500, the highest height in the game. By keeping the same slopes, the wall will be applied everytime that slope is encountered.

- Example 2 : on the contrary, you want your road to be present on different elevations, as soon as your wall stops. You can keep minimal slope and set your elevation start and end to both extremums.

- Example 3 : you want to separate your textures, like a sandwich. You are allowed to use the same material for different layers. Thus, your first layer (say [Layer0]) can be made out of the first material (say material 0), you set it to your first heights (say slope start = 0, slope end = 150), you can set another layer (say [Layer1]) on your second limits (between 150 and 200 for instance) and thus your third layer (for example [Layer3]) can be set back to material 0 for your third elevation. The same can apply for slopes if you want to obtain a rough mountain for example ([LayerX] can be from 10 to 20, [LayerX+1] from 20 to 50 and [LayerX] again from 50 to 90).

Step 4 : Using maketrn

If you are working on your own map, then your maketrn should already be inside your map folder. If it is not the case, proceed to put it inside your map folder. Once done, it is time to open command invite (cmd). After reaching your folder through cmd, you must type accordingly to Figure 6 :

```
D:\Program Files (x86)\Battlezone1.5.2.27>maketrn yourmap.trn /p="yourini.ini"
```

Figure 6 : what to type in cmd

If everything is correct, maketrn will type out a message knowing that the process was done correctly (Figure 7), otherwise it would inform you about errors it may find (Figure 8).

```
D:\Program Files (x86)\Battlezone1.5.2.27\addon\SniperSessions>maketrn sniperma.trn /p="sniperma.ini"
...Loading elevations from TRN and HGT files
...Loading layers form sniperma.ini
...Computing materials*
...Writing out material file sniperma.MAT* 1x1 zones
```

Figure 7 : successfully painting sniperma

```
D:\Program Files (x86)\Battlezone1.5.2.27\addon\SniperSessions>maketrn sniperma.trn /p="sniperma.ini"
...Loading elevations from TRN and HGT files
...Loading layers form sniperma.ini
...Computing materials*
Error...missing elevation and slope layer:
elevation = 0m and slope = 88deg
```

Figure 8 : Failing to paint sniperma due to not having a layer going as far as 88°

Step 5 : Enjoying the game

Now that we have successfully painted the map, all that is left to do is to appreciate the map if it is ready, or enjoy editing it further, knowing our terrain is carefully painted.

Important notes :

Using the redux maketrn may lead to incoherent results, such as corrupt .mat files (the file on which the game relies to precisely know the location of the textures on the map) and/or corrupt hg2 folder if you start from a 1.5 map.

Maketrn may sometimes have troubles guessing which texture to apply on a spot where three different textures encounter each other. This is due to the game relying on caps and diagonal textures and this limitation does not cover the case where 3 textures meet.

Make sure consecutive textures are linked together by caps and diag textures (mixed results may ensue)

Sources used : Battlezone1.5.2.27u1, maketrn, victory!.bzn and subsequent map victory hill, Sniperma.ini and subsequent map SniperSessions, your map (hopefully)