

# HOW TO MAKE A SKIRMISH MAP

## Landscape generation

Enter the map editor, start up the landscape generator by pressing [G].

After drawing something simple, set the bug-check mark into the position:

*DontGenerateRelief* - off

*ReCreateMinimap* - on

*OptimizeSurface* - off

*DoNothing* - off

*DontGenerateTrees* - off

and set the map size to 17000 (pixels) in the field **Size** (or less, if you want to), generate the action arena by pressing the button "**Accept**".

After admiring the generator's work, notice any flaws and correct them:

[S] calls up the smoothing tool (left click - smoothing, mouse wheel adjusts radius);

[D] - deformation tool (left click - raise, right - lower, mouse wheel adjusts radius);

To rotate the landscape you can use the combination

[Alt]+LeftClick (hold)+move the mouse left or right.

The edited landscape might have the wrong "picture" - to correct it, use the "racist" - landscape colorer by nations' "rules". You call it up with the button [R]. The mouse wheel adjusts radius, left click switches "rules", 1-6 chooses the nation.

After finishing with the landscape, use the combination [Ctrl+X] - calculation of blocks and heights.

## Blocks

The block "map" can be called up with the combination [Ctrl+Shift+F5].

Carefully study all the places where we can and cannot pass. If you find a place we dislike, try to change it.

The block map can be changed in two ways: by changing the landscape and by setting or removing blocks. The first way is difficult and sometimes unsightly (unless you are a modeler!).

The second is also not without flaws (it might open vertical inclines, and such), but it can be used for blocking paths.

Press [Ctrl+S] and see a blue circle - this is "unblocking".

Press [Ctrl+S] once more - now the circle is white - this is "full block". If you press the combination the third time, you will see a red circle - this is the "destroyer" of set (but not default!) blocks.

The fourth press turns the blocking tool off.

After setting or removing blocks it is recommended that you press [Ctrl+X].

## Heroes

After choosing the proper place for the hero to start, press the button [P], choose a color (all heroes must have different colors, from red to white, (brown color is reserved for the neutrals!)), a nation (druids, ice, mechanics, or undead), the hero (any one of three), and place it onto the landscape. (left click on the character - choice, left click on the landscape - placement).

## Resources

After placing heroes, we get to resource placement. For this press [F4].

Choose the section **#3d\_stones\_large** and, choose the stone deposit, and place it on the map. ("ring" radius is chosen by values 1-3 and 6-9, but it should be in the radius 3. To choose all elements in the resource "list", either [Shift]+LeftClick each, or [Ctrl+Shift]+LeftClick on any one).

Trees are made the same way, but with the list called up by the button **[F8]**. All types of trees are in the section **#ALL**.

## Objects

All neutral objects (shops, settlements, creeps) in the game are of the last, brown color. Shops should be selected from the "unit list" (button **[P]**) in the section **Missions**, where all objects are grouped.

**BIBLIOTEKA** - card library,  
**ALCHEMISTLABS** - alchemy labs/potions,  
**FOUNTAIN** - fountains of health and mana restoration,  
**ARTIFACT SELLER** - weapon and armour shops.

Creeps and settlements are selected the same way from their groups.

## Groups

Now we have a pretty, but non-functional map. To start, group units by purpose. Group buildings separately from units, but put all buildings for each settlements into one group.

There are at least two ways of grouping selected units:

- 1) with the LEGO tool (call up with the "LEGO" cheat command, click the "GROUP" button, set the name in the upper portion, choose the units, click the button "New Group");
- 2) with the cheat command "**g group\_name**" (when the units are selected, otherwise the group created will be empty; there should also be a single space between g and the group name)

To add units to an existing group, use the LEGO tool, GROUP: choose the group from a list (the "bug" ADV should be on to show the full list), select the units, press "**Add Selected**".

## Scripting

### Beginning

To start, add a few "variables". From the LEGO tool call **SCRIPT**.

In the window that appears find a press the button **VARIABLES**. Open the **Values** section (just click on the green line), then **BOOL\_LIST**.

Add a new element (simply click on "Add..."). Open it and change the Name field to something simpler (b\_Init, for example). Set the initial value to "true" - turn the "bug" on.

Now in the section **ARRAY\_LIST** add an array (or arrays) of guards. Do it to simplify group management. (For example: a\_Aggressive and a\_Passive).

In the variable defining the group array add "parameters" - array values. The parameter names do not matter, but each should contain at least one symbol. The parameter type from the group is: **==MAP OBJECT==:GROUP**. Double-click the pink line (Use Reference to GLOBAL OBJECT...) and choose the group you need.

### Script

Return to the script window. Here we see the line "begin ... end". Open it (before the line "add..."). Create a simple initialization operation: operation that will be executed when the map starts working, and only once!

Add the element "**BASE COMPLEX: if... while... do...**"

In the field choose the modifier "**if (...) then...**".

Open the line where it says "0".

Add the element "**F\_GLOBAL\_VALUE:Get(VAL)**".

Open it and set the value "**BOOL:b\_Init**"

Click the nearest "." twice.

Now open the line "**be\_CbaseArrayFunction**".

Add the command "**GLOBAL\_VALUE:SetValue(...=...)**"

Open what has been just added and fill in the fields: Variable = "BOOL:b\_Init" and collapse "SetValue(...=...)".

In the next line we add the element "PARAM\_FUNC:be\_CRefParamFunction".

Open the line "Select Function Name". In the first line choose "LUAFunction", in the second "ACF\_GuardZone00\_arr".

Now click the pink line-button "fCreateParamList". Fill in the field **ObjectArr** - the group array, **bGuardZone** - group aggression (after opening it can be switched into using a local value, by pressing the pink line),

**R** - aggressive attack radius (in pixels)

**iAddR** - additional radius ( $R+2*iAddR$  = radius of pursuit and "response" to shots,  $R+iAddR$  = "point of return": if the group of "defenders" returns into the original place, but the attackers cross this boundary, "defenders" will attack again).

If you create two group arrays (active and passive), then you can add another similar command.

Also, you can use the function "ACF\_GuardZone00", "ACF\_GuardZone01", and

"ACF\_GuardZone02" to set for separate groups defense conditions different from the other groups (also different radii, different activity, different settling down time (parameter **MarkDelay**, 100 by default), group to defend (SecuredGroup)).

## Settlements

Now we will tell you how to create settlements and what kinds of difficulties you might run into. First of all, make sure that the building and guard groups are separate and are not included in the **GuardZone** arrays.

Enter the variables window VARIABLES.

You should add variables in the upper part of the VARIABLES section:

in the **INT\_LIST** section, those responsible for the value of settlements and the time between building attack groups (for each race, for each "size"),

in **ARRAY\_LIST**, those responsible for setting what exactly to build (there is always enough array "parameters": first (**MAP OBJECT=:UnitType**) is the unit type, the second (**VLAUE=:INT**) - the number of units).

Now open the lowest section (SubFunctionsEditor). Add one line. At the end of the line we see buttons L, S, and X (load, save, and delete).

Load the file Data\Missions\SUBFUNCTIONS\VillageGnoll\_newSUBF.xml (Orc or Goblin, depending on race of the settlement). Now, open and fill the parameters.

ACTIVE - must be on

**l\_bInit\*** - must be true (1): is responsible for initialization and functioning of the settlement

**l\_gGuard** - guard group

**l\_gR\*** - guard radius

**l\_iAddR\*** - additional guard radius

**l\_ablInitialAbility** - hire dialog

**l\_gBaraks** - building group

**l\_gFree\*** - must be true (1)

**l\_iMoneyCoast** - settlement cost (variable with a negative number!)

**l\_nRailyPoint** - node - rally point

**l\_aUnitType** - array of what is to be built

**l\_strYES\_MONEY\*** - must be an empty line

**l\_strNO\_MONEY\*** - must be an empty line

**l\_iWaiteTime** - time between group building (can be a local value - set in milliseconds : 1000 = 1 sec)

**l\_bActive\*** - must be false (0)

**l\_iOwnerNation\*** - must be 7 (brown, neutral)

**l\_iDelay\*** - settling down time (100)

Difficulty #1: all parameters marked \* should be recreated. For this open each one in turn and click their pink lines twice, then give them the proper values.

Difficulty #2: each settlement needs its own "description"... ☺

In order for settlements (and maps) to function properly (that means that you are able to talk to leader of the tribe even after buying etc. ) you should load [L] file

**..*Data\Missions\PARAM\_FUNCTIONS\SettlementMODULE\_FN.xml***

into section **MODULE\_FN\_STORE** of parametrized functions (from SCRIPT window press FUNCTIONS).

## Map Options

This seems to be all. Only one important detail remains: map options. Call up the window Map Options: [Ctrl+E] then [P]. In it set:

**RandomizePlayersPositions** - on

**MaxPlayers** -

**DontAllowCityLife** - on

**HideEnemyUnitsUnderFog** - on

**AutoChangeRoadsNation** - on

Add **VC\_BaseFantasy** to **VictoryRules**

Do not change anything else.

If you want, you can disable unwanted "colors": to do this in Players, turn off the

**DisableInMultiplayer** "bug" for the unwanted players, where 0 is red, 1 is blue, 2 is cyan, 4 is magenta, 4 is orange, 5 is black, 6 is white.

Now, save the final version of the map (it is recommended to follow the naming scheme: **short name(number of players).m3d**, for example, arena(4).m3d).

## Simplification

Exit from the game to Windows. Write this in the command line:

***game path\Data\engine.exe /surf***

Click **INPUT**, choose the map. Click **OUTPUT**, choose the output file. In the combo box choose **Simplify** and click the **START** button.

Wait until **CREATED** is highlighted.

-----

After this you can consider the map done... Copy the final file into the folder Data\Designed and test! If the idea succeeds, send the map to the developers for all to see!

Good luck!

Buggy\_Dan