

Practical book of the editor

Before you start working with the editor itself:

General recommendations on creating missions

- If something is not clear in the description of this or that step or action of this manual, study the theoretical section of the manual in what concerns the corresponding matter.
- Try to test your map (and then the mission) on each stage as often as you can. In the game itself, things look somewhat different than they are in the editor.
- Don't forget to backup your map. This will help you to avoid doing large amounts of work anew.
- While this manual is incapable of describing all the possibilities and features of the editor, as well as all the possible problems and errors, do not despond but rather turn to the community forums and the game's fan sites: perhaps someone has already encountered the same matters and shared his or her experience. The list of fan sites and communities is available in the end of this manual.

So now, the main stages of working over a mission in the HoMM5 map editor are the following:

Mission description

Mission sketch (optional)

Work in the editor

Preparing to make a mission

Before moving on, you need to decide the mission type that you want to make and prepare a description for it.

The mission type defines certain important issues concerning the map design and gameplay which can make the designer's work easier – or more difficult.

You need the mission description to imagine in advance the queue of events on the map, as well as the mission's objectives and the key areas of its landscape.

Besides, in the description you specify the initial conditions under which the player starts the game, and describe the situation in the area in which the events will be taking place.

Preparing such a description, you'll do good if you at least elucidate the following elements:

1. *Map features*
2. *Initial powers' specification*
3. *Objectives*
4. *Conditions under which the objectives appear*

Besides, when you create a map, you can also specify the following: the initial *Briefing*, the *Biographies* of heroes and towns, the scenario *Rumors*, etc.

In some cases, you may also need to make a sketch of the map.

Deciding the mission type

Before the designer starts working over a mission for the Heroes, he or she decides what kind of a mission it will be: a single-player or a multiplayer one.

There are two distinctions in kind between these two types: at a **multiplayer** map, you want to make the initial conditions equal for all the players, for otherwise it can simply be not interesting to play on such a map with unequal conditions. On the other hand, when creating a **single-player** map, the designer usually creates unequal conditions for different players, according to the scenario he or she has in mind.

The second difference pertains to selecting the mission type for the new map. If you want your mission to be multiplayer, you need to specify a corresponding mission type when you create a map.

And you also need to decide one more thing – who can play for this or that player: a human (user) only, the AI, or any of them?

Of course, if you decide for every player that only a human or only the AI can play for him, creating such kind of map and mission will be much easier. If, however, you make a certain player available both for the user and for the AI, more work is to be applied.

(I) You can set up this or that player's availability for the human user or the AI in the players' properties when you create the map.

In our particular case, let us choose the simplest variant, with a mission of two players, one of which is always human and the other always AI-controlled.

Mission description

The mission description can be started from its general scenario. Since we do not use design elements that are too complicated for educational purpose, this mission's description will not be rich in intricate scenario turns and tasks.

This is an example of a general mission scenario:

Lord Victor is the King's only confidant who survived the last battle between the King's loyal troops and the rebels. While Victor was busy fighting for the King, his castle was robbed and all his servants driven away. Victor's closest neighbor went over to the rebels. The remains of Victor's armies, that have once been boarding in his castle, are dispersed in its whereabouts, but some of them retain their loyalty.

All that is left is a handful of surviving loyal Knights and a portion of family riches that he has managed to salvage.

Victor's task is to have all who are still loyal under his banners again and restore his sacked castle. Meanwhile Alejandro, Victor's neighbor, wastes no time and tries to seize all the mines and production centers in Victor's once rich lands.

The mission's objective is to defeat the treacherous neighbor and seize his castle.

As you can see from this scenario, we have specified the basic initial parameters and approximate rules of behavior in the game, which we shall try to be following from now on.

Map features are key aspects that should always be kept in mind, for otherwise the scenario's integrity can be impaired.

In our case, they are:

1. There are two Haven towns on the map, in its opposite ends.
2. Most of the monsters on the map (except special cases) belong to the Haven faction.
3. All mines are in approximately equal distances from both the towns.
4. The whole map is divided approximately in half by a river that flows from a lake.
5. The nature of the ground is mainly Grass, with impregnations of Dirt.

Powers specification – description of initial conditions for the user and the AI-controlled opponent. Most conveniently, they can be presented as a small table, in which the columns will represent the parties (Players), and the rows will contain parameters. This example mentions the principal parameters that affect the initial alignment of forces.

In our case, they are:

	Player 1 (player)	Player 2 (AI)
Town	None (seize neutral)	Chinchton
Initial buildings	Minimal	Average
Hero level	1	3
Initial resources	Average	Average
Initial troops	Few	Average

As you can see from the table, initially we put the player and the AI into unequal conditions. The main thing is not to forget providing the player with a way to equalize them by quickly building up an army and experience. ☺

Objectives

1. Seize a town within 7 days
2. Eliminate the enemy
3. Survive

(I) If you pay attention to the section that describes the creation and structure of objectives, you will see that all the objectives in our mission are standard ones and exist by default. See the objectives section for more details.

Conditions under which the objectives appear

1. The first objective will appear in the very beginning, since the Player for whom the user plays has no towns, and it must be fulfilled within 7 days, otherwise the mission will be a failure.
2. The second task also appears in the very beginning. This is the mission's key objective.
3. And the third task means that throughout the mission, the player must not lose his or her main hero.

If none of the tasks is failed and the second task is fulfilled, the mission is won.

Sketch of the map

In certain cases (for example, when you co-work with your friends to design a mission), it can be useful to prepare a sketch of the map to visually mark the main landscape features, place the towns and key objects, reckon the position of roads and main bounding elements, such as rivers, forests, Precipices and mountains.

The ‘toolbox’ used for creating the sketch may vary greatly: you can use pencil and paper, Adobe Photoshop, Microsoft Visio, or the HoMM5 map editor itself.

Some people may even find it convenient to use the HoMM3 map editor.

The sketch must at least show the location of towns, main roads (communications), water bodies, mountains and impassible forests. Additionally, you can show the location of scenario-pertaining objects or areas, level of monsters in individual regions of the map, etc.

If you use the editor to make a sketch, it’s enough to apply tiles of different colors to designate forests and mountains.

In our particular case, we shall create the map sketch in the editor once we start working with it, and we’ll be following it further on.

Working with the map editor

The main stages pertaining to working with the editor itself are the following:

1. Creating the map, forming the landscape and filling up the map with objects that modify and supplement the landscape.
2. Placing objects that form the gameplay (towns, heroes, important buildings, resources and artifacts, and, finally, the monsters).
3. Adjusting the properties of the map as a whole (winning and losing conditions, heroes, artifacts and spells admissible to the map, commands, etc.) and of individual objects, such as heroes, monsters, etc.

In addition to that, the users may create and set up the in-game script that provides for detailed adjustment of the mission’s properties and events.

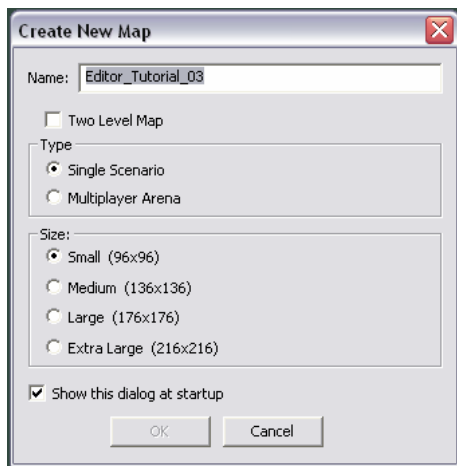
The first stage: the landscape and the key relief elements

Step 1. Create the map

Let us use the editor menu to create a new map.

A dialogue box will appear.

Let us type the map’s name and define its key parameters (which will not be changeable easily in the process of our work). These are **Map size** and **Mission type**. Depending on the decided mission type (Single Scenario or Multiplayer Arena), this mission will be available via the section of the game’s main menu that pertains either to the single player game or to the multiplayer game.

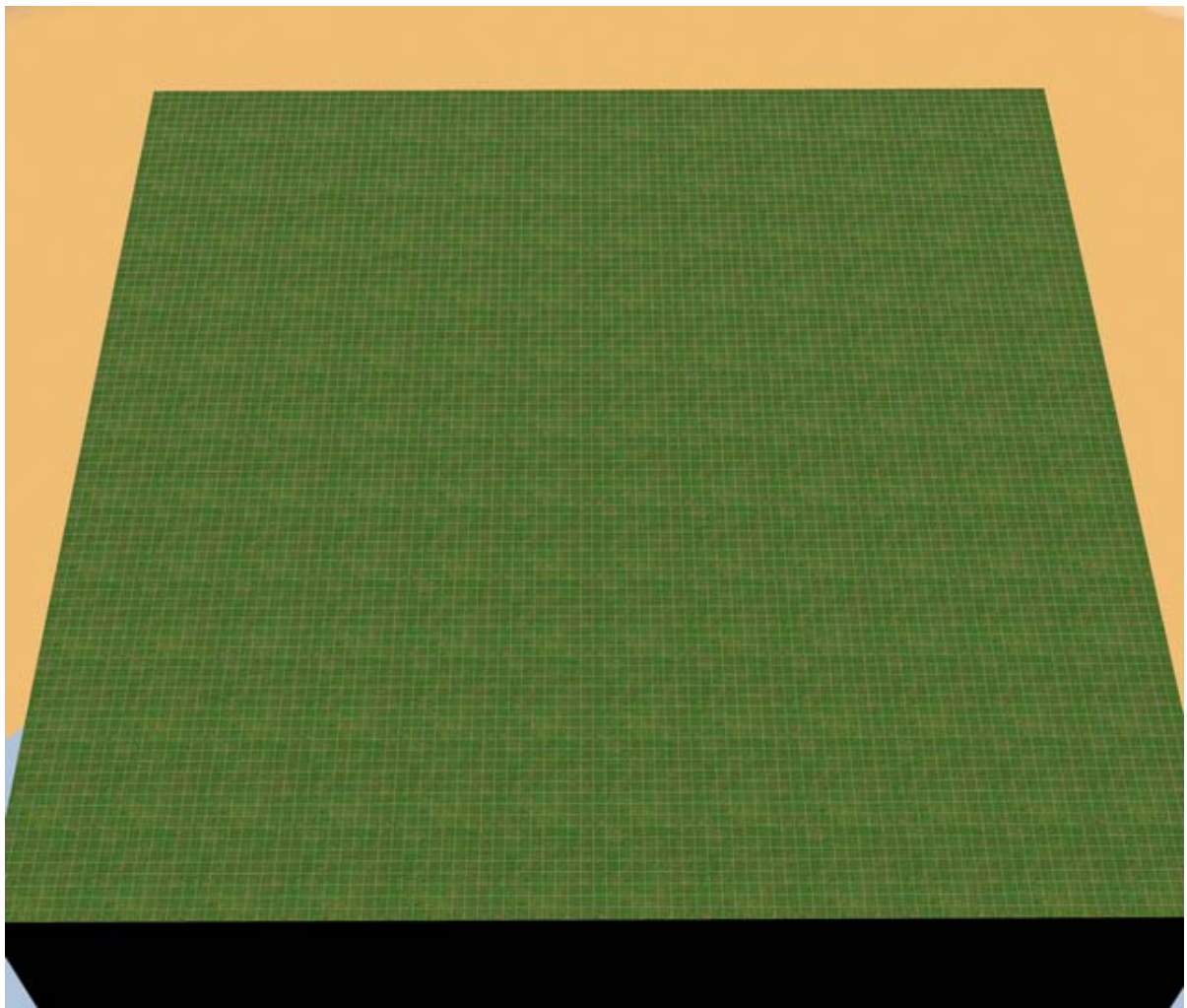


In our particular case, we make a small map without the underground level to play a single-player game.

OK, so now that we have specified these parameters, let us wait for the map to generate with all its properties and settings.

Since we have told it to be a small map, everything must go off quickly.

A flat green surface appears before us, which can be seen on the screen completely if we zoom out. This author thinks that working over the sketch is easier when you can see the whole of the map on the screen.



Step 2. Sketch of the map

Now let us mark the main zones and areas of the map.

Lake and river

This author usually starts with marking the water bodies. Since we do not intend to have seas or oceans on our map, let us choose **Tiles->Terrain Skin=Rivers->Water** and apply the future rivers and lake (this author has selected tile size of 3 right away).



If there was a tremble of hand, or a wink of eye, and the river appeared in a wrong place and not beautifully enough, take the Eraser tool from the same tab and bravely change everything back. To avoid letting slip a piece of the old and ‘bad’ river, orientate yourself with the minimap that shows the location of different types of tiles in different colors.

(I) By the way, once you have managed to make a more or less decent picture, you should save the map and make a backup. The easiest way to backup is by using the Save As command in the File menu of the map editor.

Usually, this author backups after the first saving and generally goes on doing so after having done this or that considerable amount of work. Sometimes you understand in the middle of work that all the changes made during the last couple of days should be undone and everything made in quite another manner. This is where the backup made in good time proves to be helpful!

Towns

Now let us place both of our towns.

Let's go Objects->Towns->Haven Town

Put them in the map's opposite corners, to make the distance between them as great as possible at our small map.



You can see that this author has placed the “remote” town in the very corner while the “nearby” one has enough space before it. And this is why: this town needs to be communicated by a road with corresponding scenery elements, which will be taking some space. Besides, a road paved in the very edge of the map looks ugly.

(I) Like all other types of objects, towns can be easily moved or removed. If you want, you can rotate this or that town, but this is usually done only on multiplayer maps.

Roads

Let us build a road.

Go Tiles->Terra skin=Grass->StoneRoad

In our particular case, we have only one main road that leads from one town to the other.



In addition to the main road, let us draw an auxiliary one that will play the role of an alternate route in our mission. Go Tiles->Terra skin=Dirt->DirtRoad.



And again, if the road happens to appear in a location different from the desired, use the Eraser tool.



Heights (mountains)

Now let us mark the mountainous areas on the map.

Go ***Tiles->Terra skin=Conquest->Stone***

In our particular case, the mountains are encircling the lake, which, by the way, is in an elevated region itself.



Forests

Now let us discuss forests and other kinds of impassible vegetation.

Before you place the objects that will later be identified as 'forest', it's advisable to mark the areas in which these forests will grow.

To do so, select a tile different from the basic one and mark the future forest areas.

Go Tiles->Terra skin=Dirt->Weed.

Our future forests.



And some dark ground (Tiles->Terra skin=Dirt->Dark Ground) in the area of the future swamp.



The minimap shows all our new terrain types clearly.



The sketch is ready.

Step 3. Map relief (landscape)

Now let us specify the key features of the map's relief.

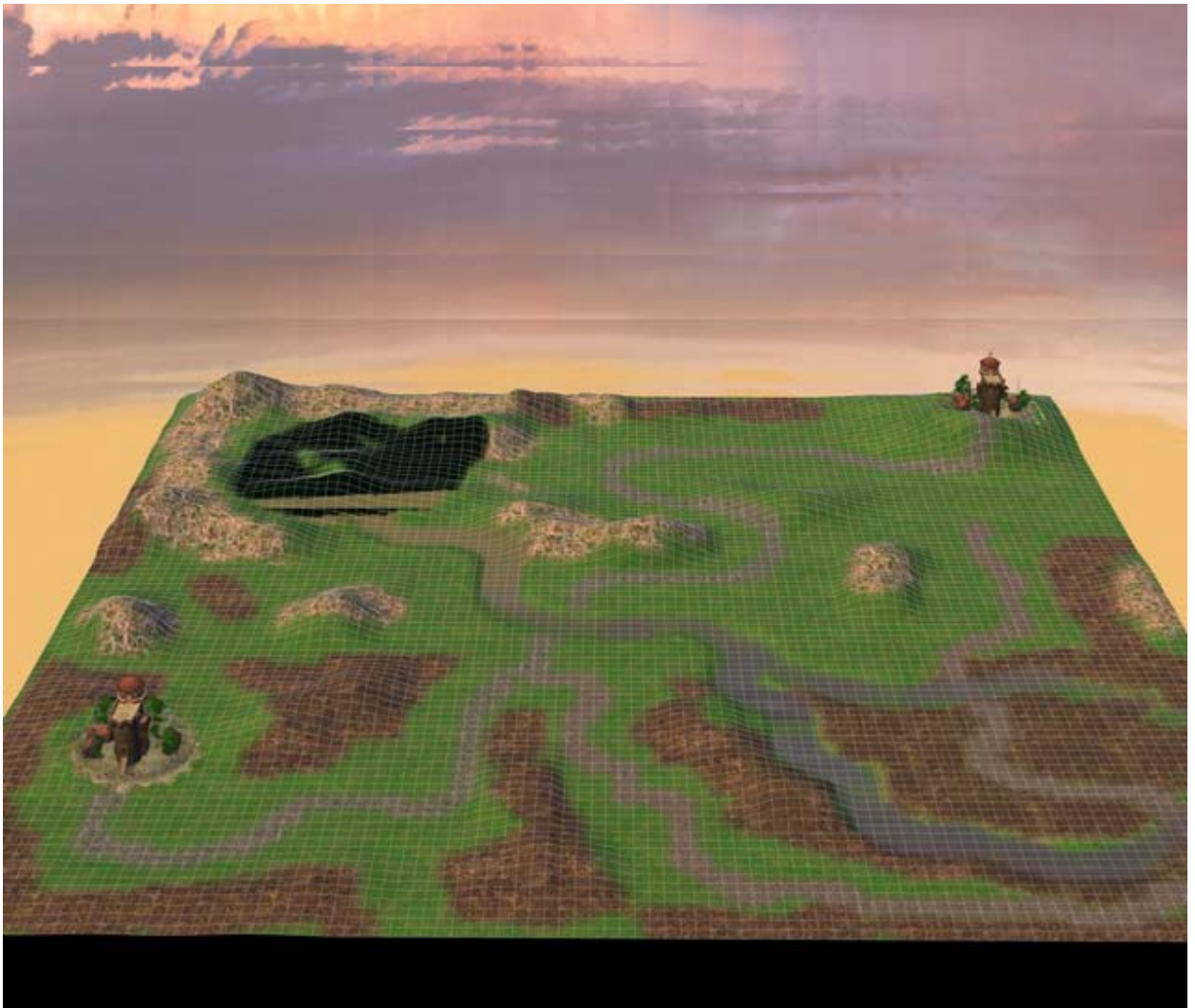
Heights

In our particular case, these are the dominating eminence in the lake region and the flat height surrounded by Precipices in the area of the second (AI's) castle.

(I) Why do we need to modify the relief on this stage? It's simple: if we modify the relief later, with objects already placed, we'll have to correct a great deal of problems with the surface under these objects.

Go Tiles->Draw Tools Size=3->Terraforming=Bulk.

We have marked the main eminences.



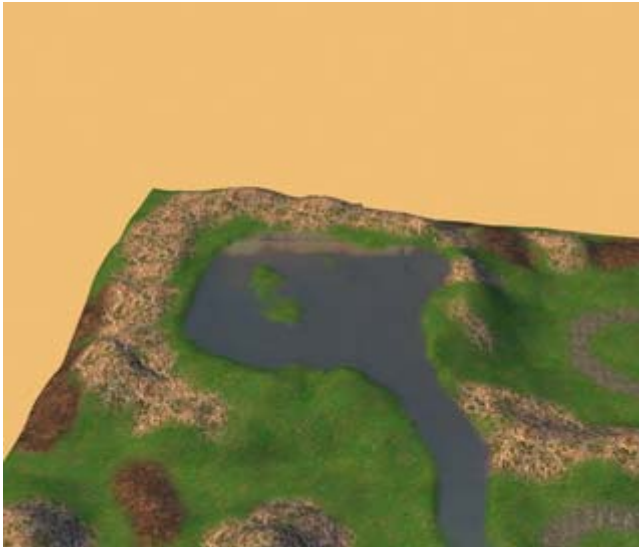
By the way, here you can see that this author has turned the Grid representation mode on. It makes all the relief features visible clearly.

Now that the main heights are assigned, let us straighten them and make them more neat.

To do so, let us use the **Plateau** and **Smth** tools.

We use Plateau to lift a large map area to the same level at once, while Smth is used to smooth all the surface imperfections and places where the eminence meets the flat land. To add some variety, we can use Rnd terrain.

This is what this author made.



By the way, when you work, don't hesitate to make corrections. This picture shows that the author has corrected the channel of the river that flows from the lake.

Precipice

This author wants to place the castle of the player controlled by the AI at a plateau encircled by an impassible precipice. To do so, he has first lifted the terrain in this castle's area and then shall use the Raise tool.

Go Tiles->Draw Tools Size=3->Terraforming=Raise.

The result is a neat flat hill with our opponent's castle on it.



(I) If you first Raise an area and then draw the relief in the same area up or down, there can emerge various artifacts and ugly things which will be difficult to correct then.

To have this 'hill' being in communication with the mainland, you need to connect them with a special terrain element, the Ramp.

Go Tiles->Draw Tools Size=3->Terraforming=Ramp

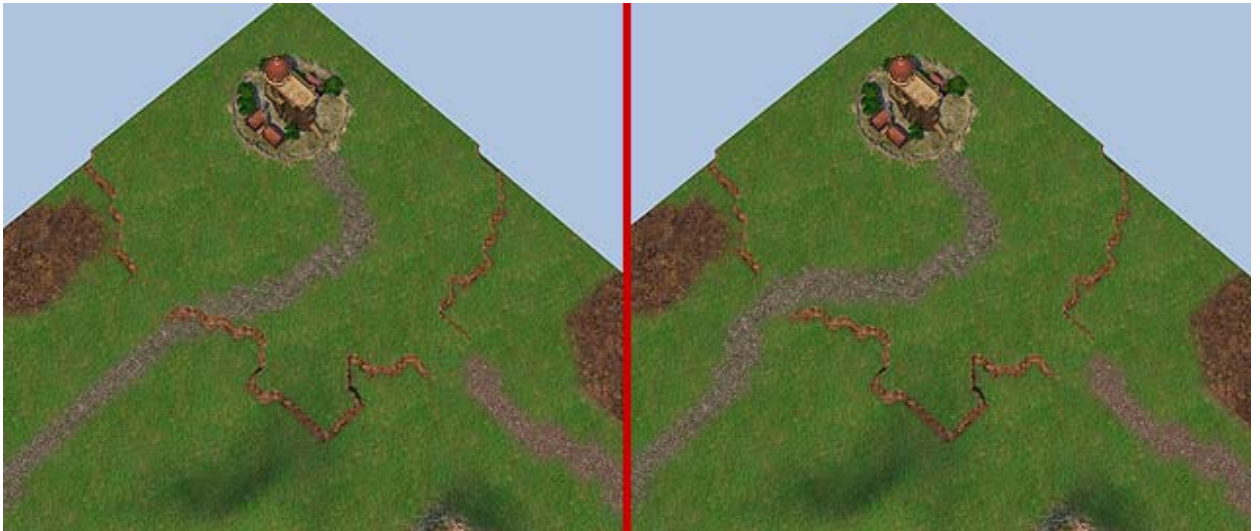
...and make two ramps – probable places through which the top of our hill can be reached.



By the way, here you can see that we have paved the road in an inappropriate location. Let us correct it.

Go Tiles->Terraskin=Grass->StoneRoad + Eraser.

Now it looks much better.



Since Ramp is not the simplest of the tools, it is advisable that you study the corresponding section in the theoretical part of the manual once again.

Step 4. Placing forests and mountains.

Now we are ready to adorn our landscape with mountains and forests.

In HoMM5, forests and mountains are created with the use of objects: trees and proper mountains.

Let us first place the mountains.

Go Objects->Grass objects.

For our season, there are three sizes of mountains. Naturally, we put larger mountains in the heart of the mountainous areas or there where free space is abundant.

Let us use the three different mountains sizes to make something like this:

Now to make the relief somewhat diverse, let us lift and sink the relief under these mountains: the mountains themselves will deform, which makes it possible to get something like this:



As you will see, the impression is as if many more typical elements have been used to create the mountains.



(normal mountain)

(modified mountain)

Let us now turn to creating forests.

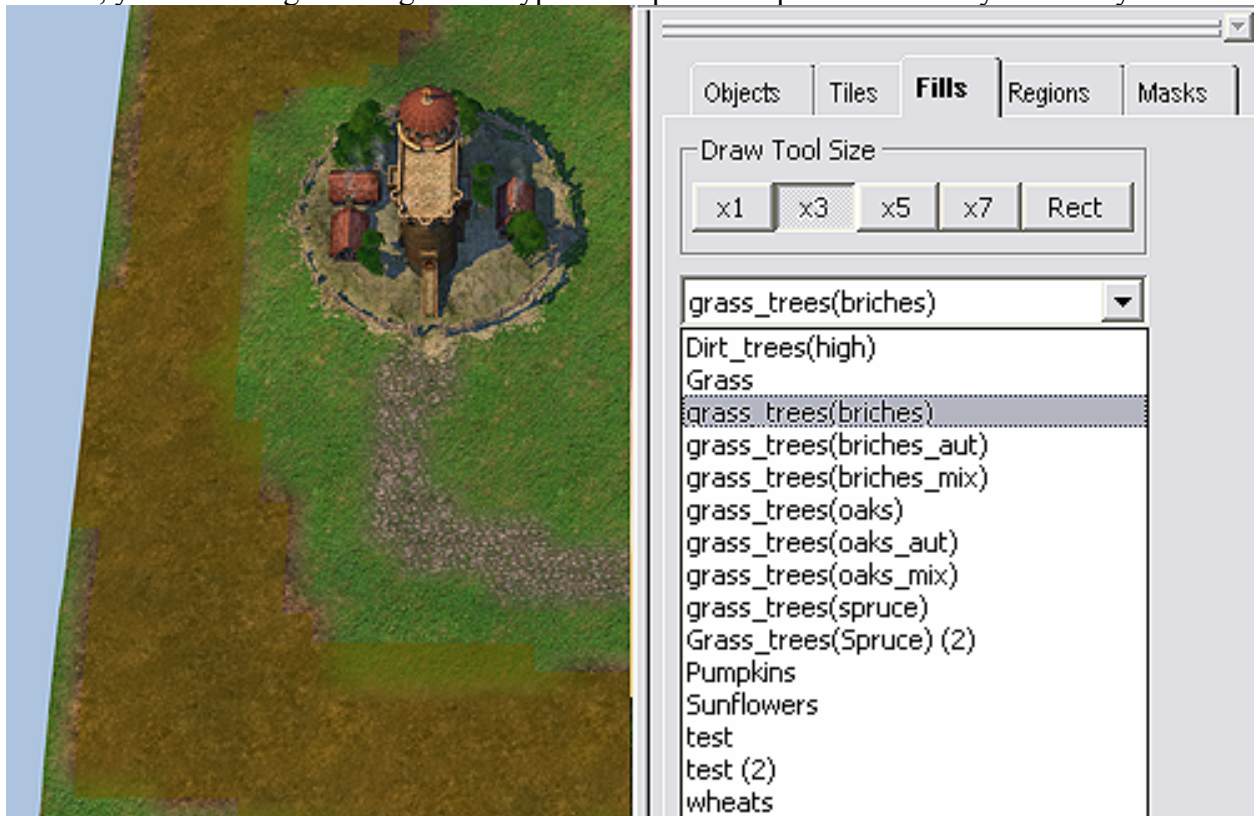
There are two methods to do that: by placing the Tree-type objects manually or by filling the desired areas with the Fills tool. Since this map is nothing more but a test one, let us waste no time and use the tool that has been designed for filling the designated areas with the selected type of vegetation automatically.

Go Fills->Draw Tolls Size=3.

Here select the preferred vegetation type from the pulldown menu.

This author thinks that a summer map with prevailing Grass terrain should have Oaks, Birches, Spruces and Grass.

Then mark the area to be filled with the selected type of vegetation and press the FILL button. If desired, you can change the vegetation type and repeat the operation as many times as you want.



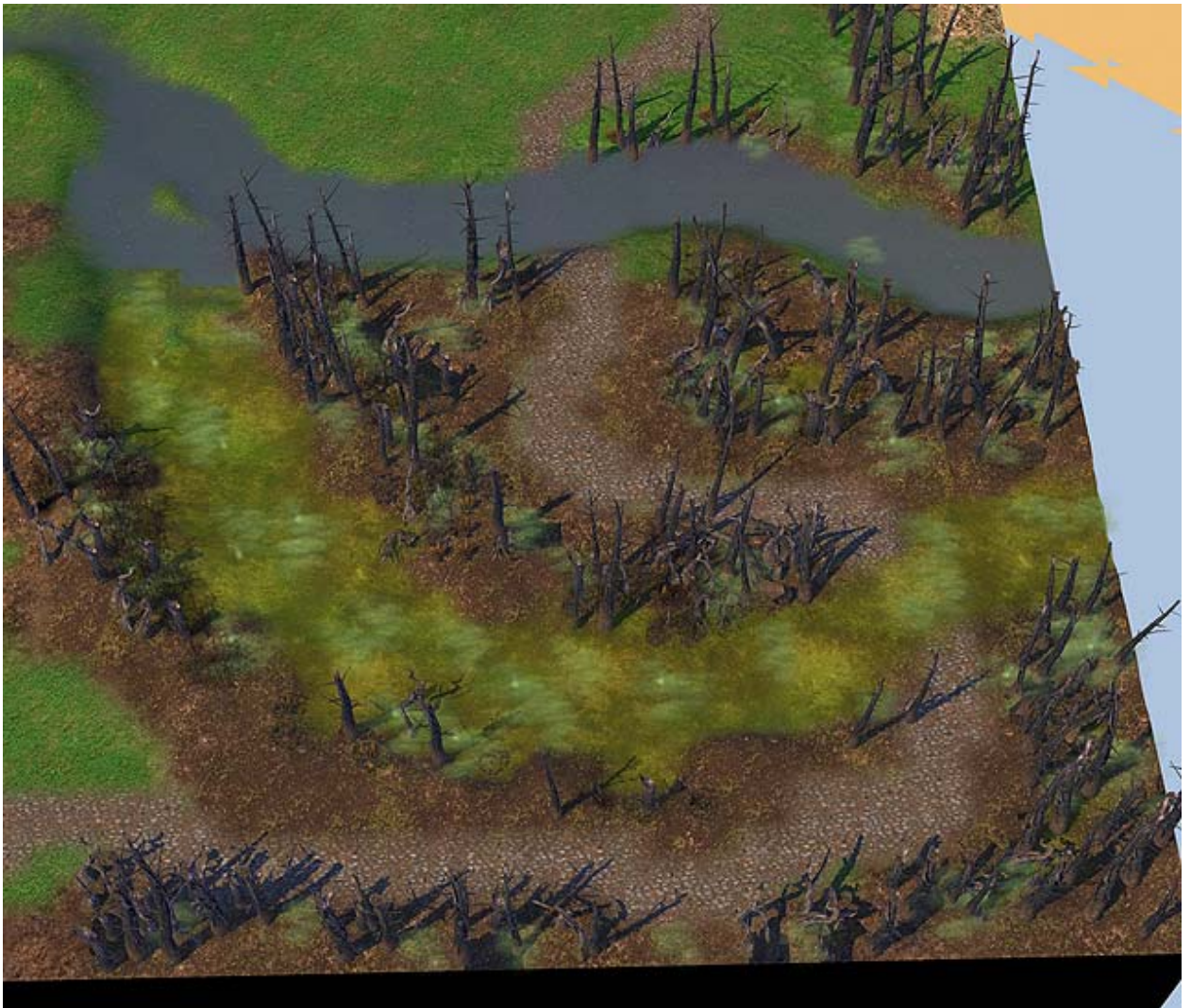
As you know, forests do not consist of trees only, that is why grass and brushes have been placed under them in addition.



But do not create forests that are too dense, since this affects your PC's performance both in the editor and while playing this map.

According to this author's idea, there must be a waterlogged area in the mouth of the river, symbolizing dead wood region; accordingly, some gloomy spruces have been placed there together with some dead trees.

To make the swampy effect, we need to change the water. To do so, go **Tiles->Terra skin=Rivers**. Remove normal water from one of the river's armlets and fill it with stagnant water (**Tiles->Terra skin=Rivers->Bog**).



At this particular stage, we shall not pay too much attention to beatifying, so let's postpone bringing the forests to perfection for the time being.

(I) Small vegetated areas should still be placed manually to reach the desired visual effect.

So now, after all these manipulations, our map looks more or less like this.



Let us now proceed to placing the gameplay objects.

Second stage – Placing the objects

See the theoretical part of this manual for the general information about placing, moving, and rotating the objects. The practical part will not take up placing such objects as bridges, since, once again, it has been described in the theoretical part.

Step 1. Heroes

First of all, let us place the heroes who are to be present on-map from the very beginning.

Go Objects->Heroes->Knight.

(We have Knights playing for both parties).

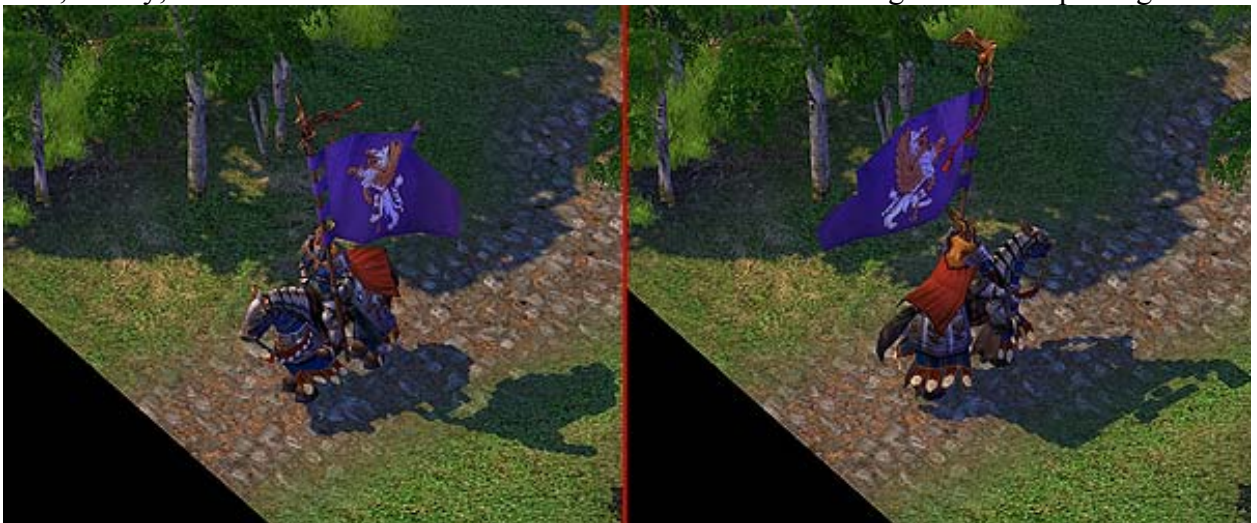
One of the heroes is the player's future hero, and the other will be controlled by the AI. Accordingly, let us place them in the opposite ends of the map.

Since, according to the scenario, the user's hero has no castle, let us put him on the road in the forest as if he has just found himself in this area. The AI's hero will be placed near his castle.



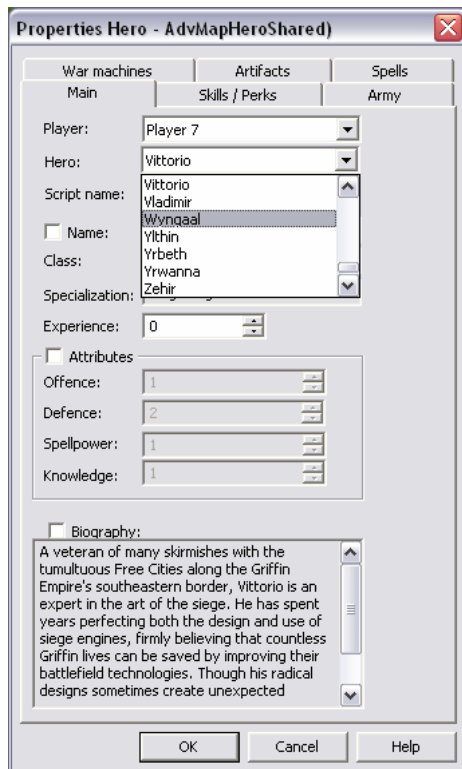
For convenience, we shall from now on identify the hero controlled by the player as 'our' hero.

And, finally, let us rotate our hero so that he does not have his nose against the map's edge.



Now let's set up some of the heroes' properties.

Select the hero whose parameters are to be modified and hit **Space** or the corresponding button in the editor's menu.



Here we can define exactly which hero in the list will be playing for us. Let us then define the Player who will control this hero, type in his name and give him an initial army. Or, you can set up the army a bit later.

This, in fact, is enough for the time being. If you want, you can rewrite this hero's biography and spend some time working over his skills, artifacts and spells. This author, however, chose to do that somewhat later.

Step 2. Mines

Now let us place such important type of objects as mines: the main resource type and the subject of controversy between the Players (we'll consider the Sawmill and the Alchemist's Lab among the mines, too).

On this map, according to the scenario, we shall place one mine of each kind, to provoke the Players' struggle for control over them. Besides, our map is a small one and does not provide room for many mines in any case.

Go Objects->All terrain objects and find the mines (they all are near the end of the list).

This author offers concentrating mines like **Gold**, **Ore**, **Crystals** and **Gems** in the area where we have the mountain lake. Then we'll favor the AI with the **Sawmill**, and place the **Abandoned Mine** in the opposite corner. The **Alchemist's Lab** and the **Sulfur Mine** will be in the whereabouts of our future swamp.



*(I) Remember to rotate the mines in the way that they have convenient access, and the players' heroes could seize them: to do so, turn on the passability presentation mode when you place a mine, and use the **R** key to orientate the green point.*

Let us not touch the settings of the Desolate Mine to randomize the player's life a bit.

Step 3. Monsters: the interlude

This author feels like you want to add some monsters onto the map at this stage.

However, it's too early to do so yet.

Believe this author's advice, since he's tried all the ways of placing monsters at this stage. None worked. That is all because the monsters' main function is to protect various nice objects and block the hero's path to limit his or her travels.

It now follows that in order to have monsters in places other than near the mines you should first fill the map with other objects.

So this is what we're going to do now and return to the monsters a bit later.

Step 4. Objects

Before placing various objects on the map, let us make sure what exactly do we want to have. While listing the objects, put them onto the map in some arbitrary location. The best idea is to put all the objects we shall need in one place. This is done in order to avoid searching for each individual object in the lists later. Though everyone is free to do as he or she finds it most convenient.

Let us remember that we are not going to use all the kinds of objects on our the map, so we shall only list and place only those we're going to need.

(I) When selecting objects from the lists, use the tooltip that will always explain what exactly is this object doing and how.

Luck and Morale

Objects that modify the hero's army's Luck and Morale (in the brackets you see where you can find this object):

- Fountain of Fortune (Objects->All-terrain Objects)
- Fountain of Youth (Objects->All-terrain Objects или Objects->Grass)
- Idol of Fortune (Objects->All-terrain Objects)
- Tattered Flag (Objects->All-terrain Objects)
- Faerie Ring (Objects->Grass)
- Temple (Objects->All-terrain Objects)

Now let us spread these objects on the map. Objects can be placed either in the heroes' way (i. e. close to roads and the main communication routes); or some distance away, somewhere in the wilderness.

Here you can see the intended locations to place objects.



Weekly resources

Objects that provide a visiting hero with some resources every week:

- Garden of the Wee Folk (Objects->Grass)
- Water Wheel (Objects->All-terrain Objects)
- Windmill (Objects->All-terrain Objects)

Place the objects like this: put the Garden of the Wee Folk not far from each of the castles (remember to use Copy+Paste), and the Water Wheel and the Windmill somewhere in the middle of the map. It would be reasonable, of course, to place the Water Wheel on the water.



There is a couple of features pertaining to placing the Water Wheel and similar objects.

First, even though the shores of your rivers and other water bodies can be far from straight, the Water Wheel (as well as certain other objects) should never be rotated with Ctrl (i. e. arbitrarily): use the R key only – remember the rule of rotation for multitle objects.



Second, to place the Water Wheel on the *water*, and not anywhere else, the best method is the following: in the location where you are going to place the Water Wheel on the water body's shore, make a small promontory or embankment which allows placing the Water Wheel further from the shore.



You can use Tiles->River->Water+Up or Tiles->River->Water+Eraser.



After that, place the Water Wheel and cautiously restore the part of the water body under it with Tiles->River->Water+Down.



The result is that you get what you need.

Hiring troops (monsters)

Objects that provide the player with the possibility of hiring additional troops:

Archers Tower (Objects->All-terrain Objects)

Barracks (Objects->All-terrain Objects)

Peasant Hut (Objects->All-terrain Objects)

This author thinks that these objects can be placed in the following locations: Peasants will be living about each of the castles, and the Archers Tower and the Footmen's Barracks will be

somewhere in the middle, which will provoke the Players to compete for these two objects and perform with activity.

Placing the Peasant Huts

The Peasant Huts, of course, need to be supplied with agricultural fields (with crops of wheat or pumpkins, which does not matter).

Let us place the hut in the area near our castle.



Supply it with fields. Take the field from Tiles->Terra Skin=Grass->Field and remember to draw it as much as it needs! Drawing means here that to have the applied tile appear clearly it has to be applied several times onto the same place.



There are four types of crops that can be 'planted' in the fields: Wheat, Corns, Pumpkins and Sunflowers; then you can enclose the field with Fences. All these objects can be found in Objects->Grass Objects.

Note that all the above-mentioned objects belong to the class of those existing in more than one variant. But the editor has only one reference to such an object (a single icon in the corresponding section of objects).

When you place such an object onto the map, you get one of the several variants, selected randomly. If you place such an object onto the map several times, you can see a number of its variants at once.

(I) In fact, it is not necessary to plant anything on the fields: you can leave them lie fallow or place some Haystacks (as if the fields have just been harvested). By and large, do not limit your imagination.

Try to place several Fences in a row onto the map without using the Copy-Paste technique. You will see four variants of the fence randomly.



So now, all the issues pertaining to the Peasant Huts have finally been solved. And this is what we've got.



Use the same methods to place the Peasant Hut near the AI's castle, too.



Fortunately, we don't have many objects at our service that demand placing the scenery like the Peasant Huts.

Let us place the Archers Tower and the Footmen's Barracks

We shall place the Tower and the Barracks at the same meadow so that they could be visited in one turn or so.



Besides appointed objects that produce creatures of the specified race, there also exists a special type of objects called Random Dwelling, which also has the main function of producing troops every week, but this type has certain peculiar features that are mostly used in multiplayer maps. See the corresponding section for more details.

(I) After having placed the objects that can be seized make sure that each of the objects has correct belonging set. The right setting is None (Neutral).

Objects that modify the hero's parameters:

- Arena (Objects->All-terrain Objects)
- Crystal of Revelation (Objects->All-terrain Objects)
- Mercenary Camp (Objects->All-terrain Objects)
- Marletto Tower (Objects->All-terrain Objects)
- Planetarium (Objects->All-terrain Objects)

All these objects increase the hero's basic characteristics.

Dolmen of Knowledge (Objects->All-terrain Objects)

This object can also be considered among the parameters modifying ones.

This author sees no special rules for placing such objects: it's simply desirable not to place them in the most easily accessible locations or in the roads, and that's all.

It's also good to avoid having too many of them. In our particular case, it's advisable to place these objects in locations that are available (and close enough) for our hero, to provide him with a chance to advance quickly in the beginning of the game.

This author has marked such locations.



Remember to check the orientation of each object's green point ('entrance') when you place it.

See below about the Dolmen of Knowledge.

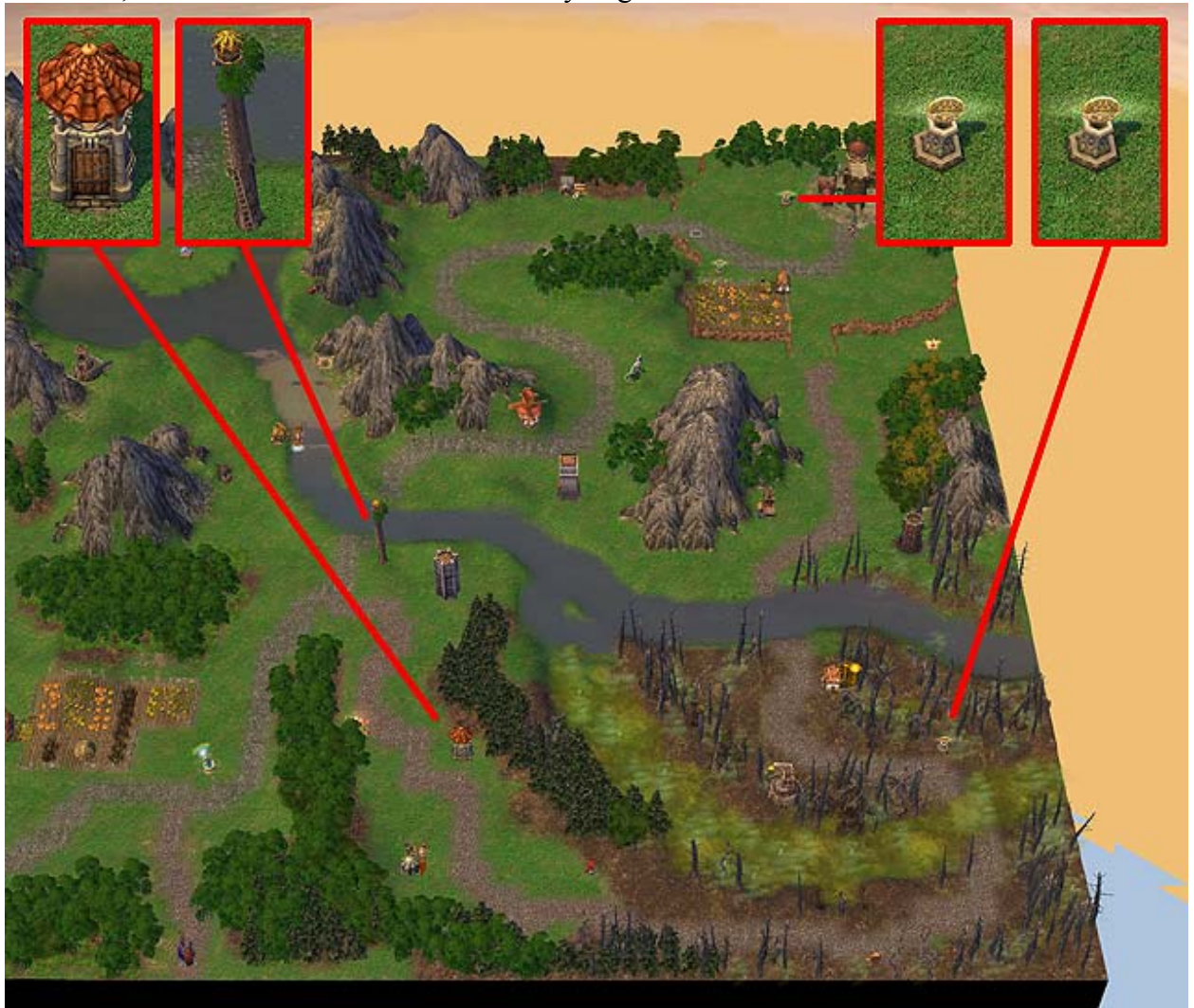
Objects that reveal terrain (map)

There are some objects at our service that can, when used appropriately, direct the player to where we want him or her to go, or motivate him or her to perform such and such actions. These are objects that reveal this or that part of the map.

Hut of the Magi /does not work without Eye of the Magi / (Objects->All-terrain Objects)
Eye of the Magi /does not work without Hut of the Magi/ (Objects->All-terrain Objects)
Redwood Observatory (Objects->All-terrain Objects)

Take notice that Hut of the Magi and Eye of the Magi do not work without each other. Besides, they belong to the so-called “group” objects that need additional adjustment to work correctly.

This author proposes to place the Observatory in the middle of the map, somewhere near the river ford, so that our hero could see his enemy in good time.



The Hut of the Magi should also be placed on our hero’s half of the map, but a couple of the Eyes of the Magi on the AI’s half.

We’ll put the first Eye not far from the AI’s castle, and the other one near the swamp.

By the way, this author has a nice idea of hiding some kind of a powerful artifact in the swamps, to make it possible for the future player to find it, thus making his fight against the AI much easier. So this is what we’ll do. See below about the artifact.

Placing other objects

Of all the remaining object types, we shall only take the following:

Shrine of Magic Thoughts (Objects->All-terrain Objects)

Eldritch Well (Objects->All-terrain Objects)

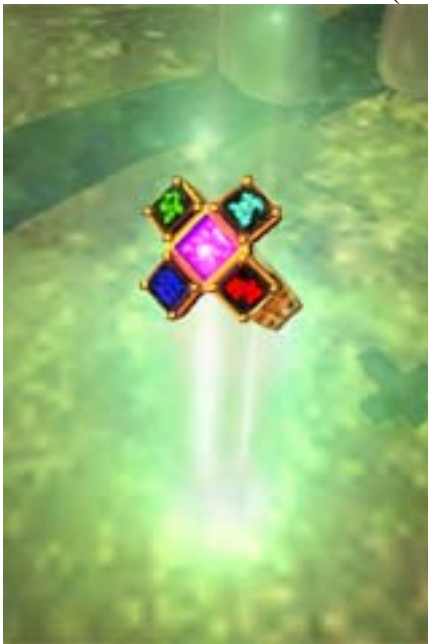
Signpost (Objects->All-terrain Objects)

Monolith Two Way (Objects->All-terrain Objects)

Crypt (Objects->All-terrain Objects)

And, since we need an artifact:

Elemental Waistband (Objects->Artifacts)



For our mission, this author proposes to use some ultimate artifact that provides the hero using it with considerable advantage. Since the heroes are not supposed to have very high levels (taking into account that they start at 1st and 3rd levels), the Elemental Waistband can suffice.

Let us hide the artifact somewhere not too far from the Eye of the Magi in the whereabouts of our swamp.



The main idea is to protect this artifact in such an intricate manner that our hero could gain it and the AI's one couldn't. This author thinks that this can be done with the use of loyalty and desire to join settings of the groups of monsters who guard the artifact.

Oh yes, and what about other objects?

Let us see what they are for and where they can be placed.



Shrine of Magic Gesture –

teaches the visiting hero a random 2nd level spell. In our case, using this object can help to enrich our hero's store of knowledge with a useful spell quickly. What is to be done for that?

First, let us place the Shrine on our hero's half of the map (to be more or less sure that our hero reaches it first).

Second, let us define exactly what spell will the Shrine be teaching. This author offers to choose Mass Endurance: a rather ultimate spell that grants an appreciable Defense bonus when used.

Eldritch Well –

completely restores the hero's mana stock.

This author thinks that it can be placed somewhere in the middle of the map, not far from the main road.

Signpost –

is, generally speaking, just a signpost which can be assigned with a textual message that the hero will see when visiting.

Let us put it in the crossroads of the main and minor roads!

Monolith Two Way –

is a two-way teleport that helps to solve such tasks as access to a secret area or sending the player to a very remote section of the map. If you have played the Warlock campaign in Heroes 5, you may remember the mission where there was a labyrinth with numerous teleports with ‘random’ communication between them. That is, when you entered a teleport, you didn’t know which of them you will appear from. This was very simple: all the teleports belonged to the same group of objects.

This author proposes to place the teleports like this:

- one at an island in the middle of the lake, with various riches displayed there; including the Dolmen of Knowledge that has been mentioned above.
- The other in a place difficult of access and well-guarded.

Crypt –

is another method of providing the player with resources on this or that stage of the mission, just as other ‘***treasury***’ type objects; the Crypt is a burial-vault of ancient heroes that stores their modest donation to the future generations.

As with all the other objects of this type, you can not adjust the number of troops that will guard it, or what exactly treasures it will contain.

Let us choose a place and put the Crypt in the midst of a dense forest.

As you can see, we have placed the Crypt near the ***Tattered Flag*** that symbolizes the site of an ancient battle.

Something about problems with objects placing

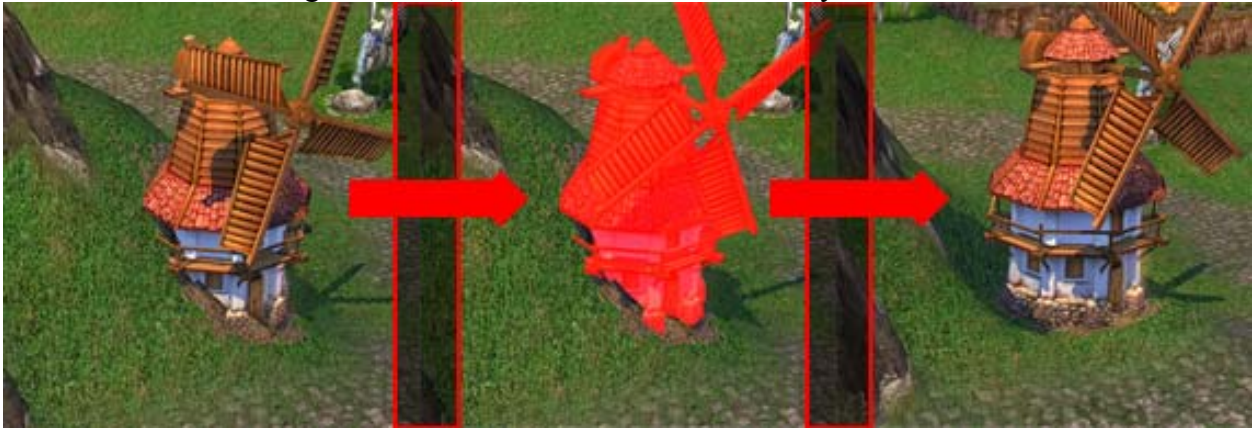
We often happen to place objects on uneven ground. Meanwhile, the objects themselves always have even base (which is natural).

As a result, the objects are often loose-fitting to the ground, orientating by the highest point of the relief which touches the object’s base.

To have it look normally, you should level the ground beneath the object.

To do so, the best idea is to use the ***Level Terrain Under Object*** command (you can find a corresponding icon in the Edit menu).

1. Just select an object under which you need to level the ground and perform the **Level Terrain Under Object** command. The terrain under the object's base, according to its size, will be leveled automatically.



2. Then you can elaborate the object's neighborhood with the Smth tool.
3. This method works with all objects, including Bridges and even Towns.

Resources

Now let us finally place the resources.

As you know, there are 7 kinds of resources in HoMM5. But there are some other objects that also belong to the class of Treasures. Such are the Soulstones, used only on multiplayer maps. See the next chapter for more on them.

Besides, there are some other objects pertaining to resources: Campfire, Treasure Chest and Random Treasure.

(I) All the types of resources are 'one-shot' objects: after their interaction with the hero (for example, when the hero picks the resource up) they disappear from the map.

Well now, we are ready to proceed to placing the resources onto the map.

Go Objects->Treasures...

Choose and place the resources that we need. If there is no need to define what exactly type of resource must be in such and such a location, place Random Treasure there.

The amount of resources on the map must be so great exactly that it would stimulate the player to collect them, though, of course, everything depends on the individual map and mission and the designer's intent.

Each resource placed on the map can be adjusted (even though no one usually does that). The adjustment comes to defining the exact amount of this type of resource in the 'pile'.

By the way, Treasure Chests are adjustable, too, and it has the same settings, but they work somewhat differently: the number specified in the Chest's properties is a multiplier that increases the amount of experience and corresponding gold in the Chest.

Placing the resources

Since usually there are dozens of resources placed onto the map, let us only review one map area in detail.

Let us mark the main locations of resources accumulation and place them there.



Usually, you want to put resources at a distance from the hero's main route (i. e. in the roads' offshoots and some lonely places), and in such a manner that it would be easy to defend them with monsters.

In case if you are not sure what exactly type of resource you want to place in this or that location, use the Random Treasure placeholder.

Monsters: finally placed

Now that we have placed so many objects onto the map to make the heroes be busy visiting and seizing them, it's time to have hostile forces guarding them.

Before we start placing the monsters, let us remember some conditions:

- The map mostly contains monsters from the Haven faction (there are no other factions on the map). Undead monsters are possible in the region of our future swamp. Objects that are especially important (such as the artifact) can be guarded by something exotic.

- In the areas where the heroes initially appear, the monsters must not be of too high level, and their numbers must allow defeating them.
- Let us also not forget that, according to the scenario's idea, some of Victor's troops have been scattered in the vicinity of his former castle. And they are possibly willing to join him again.
- Most of the important objects (such as mines) must be guarded with monsters.

OK now, reasoning from this introduction, let us do the following:

In the area of the first castle (where our hero appears), place several groups of monsters that are willing to join Victor. But have some groups, too, that have aggressive or neutral attitude towards Victor (in order for not making the player too happy and 'grabbing' all the monsters without distinction, thinking that they all will join).

Go Objects->Monsters

Find the Haven monsters (since all the monsters are sorted alphabetically, it's easier to search them out by their names' first letters, or the icons' color since monsters of different factions have different icon colors).

(I) When searching for the objects or monsters on the tab Objects, use searching by the first letters.

This author offers to place all the Haven monsters onto the map right away, then make copies of the type you need.

Now if you need to place this or that group of monsters, just copy and paste it.

Use this technique to place some monsters of 1st to 4th tier in the road and about the castle.

Place the monsters in places where they:

1. Block way to the resources;
2. Limit the hero's movement about the map;
3. Prevent him from attacking the castle immediately.

And this is what we've got. Now follow the same order and principles and fill the whole map with monsters.



So now, we have placed monsters at the map. That means that we have defined which types of monsters (or monsters of which levels or tires) will be standing in such and such places.

By the way, you may have noticed that this author has made some work to fill up the empty spaces. Besides the map's just becoming more beautiful, we can now see our hero's route better and understand it more clearly what and where are our monsters guarding.

Now we have to adjust individual groups of monsters.

Adjusting the monsters

As far as setting up the number of monsters, you have (especially in the beginning of the mission) consider what number of troops will our hero have when he encounters these monsters, and how many monsters he'll be able to defeat. The balance can, of course, be found by testing the map.

But let us make some preliminary work. Calculate approximately, which troops and how many of them shall our hero (or the enemy hero, if the monsters are placed and adjusted

in the areas near his castle) have in the beginning; also take into account the troops that can be hired in the castle (if there is such).

Depending on this, define the size of the army of monsters that our hero will be encountering in the beginning of the game.

In our particular case, we define and set up the initial army in the hero's properties (see below), reasoning from the starting conditions.

Being guided by the size of our hero's army (that created by yourself can, by the way, have a different size of army), we'll set up the troops he will be encountering in his way.

To begin with, let us remember something:

1. Some groups of monsters will be loyal (and join our hero unconditionally according to the scenario).
2. Other groups of monsters, on the other hand, will put up unexpected (as this author wants to believe) resistance.
3. Also there will be a couple of groups that are disposed towards joining but not too eagerly. If they will join, they'll only do so for certain remuneration.

The rest of the map, basically, does not demand following such strict rules. This author has just made the river crossings defended somewhat better than everything else.

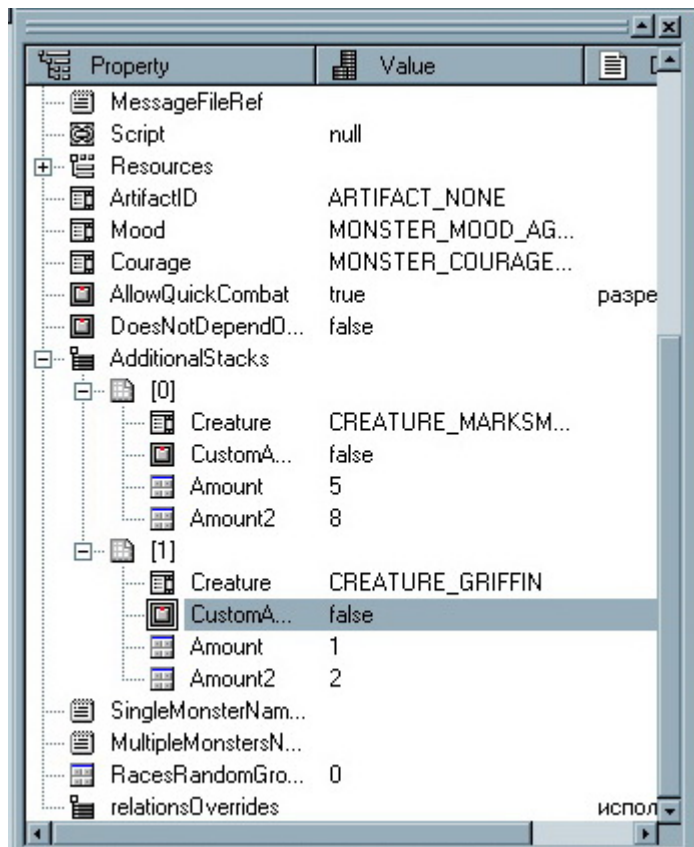
Testing the placed groups of monsters is one of the longest processes in map/mission creation. You will have to readjust them more than once and walk your own map many a time.

As an advice, try to imagine yourself in the player's boots and perform varied, preferably illogical, sequences of actions.

So now, some time later the main work of placing the groups of monsters and testing/adjusting them has been finished. And now we have all the possible objects placed and even more or less adjusted. Let us now proceed to the rest of the settings.

Create an army

In the Tribes Of The East Editor you have ability to create a groups of different monsters which called an Army. To create an Army you need to add stacks to existing monster's group on the map. To do so you should use "AdditionalStacks" properties in the object properties window for previously created single creature monster's group. Just add a required number of stacks, select monster's types and choose a number of monster for each stack (you can randomize a number!).



For adding stacks right click “AdditionalStacks” property and choose “add”.

Adjusting the heroes

Let us first adjust our hero. Then we’ll use similar techniques to adjust the AI’s main hero.

Though we have put the heroes’ figures onto the map already, many things can change in the process of their adjustment.

First, we need to choose a hero ‘template’ from the existing list. This list includes most of the heroes that we have encountered in single-player and multiplayer modes of Heroes 5.

As we can see, each hero has something special and distinguishing. Of course, these distinctions are not very substantial, but still...



For our map, this author chose Vittorio: he's replaced the hero's biography with a different text (see below), and changed his name in the very beginning.

Victor's biography

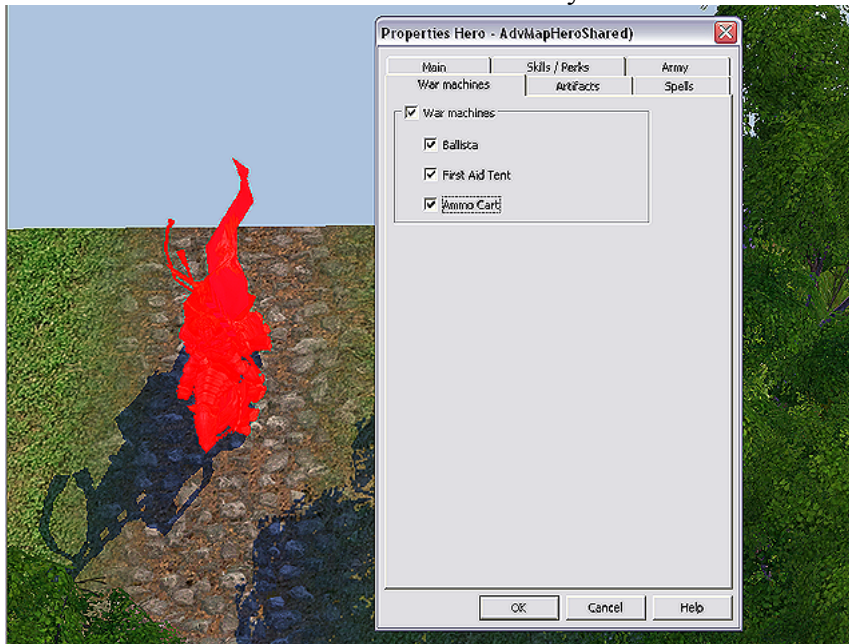
Lord Victor is King George's childhood friend.

Loyal fellow-champion, Victor follows the King in all his expeditions and battles. When several large counties of the Kingdom, incited by the messengers of the Inferno, declared independence, Victor took the field with the King, ordering one of his vassals to look after his lands.

Then, you can modify the following characteristics in our hero's settings:

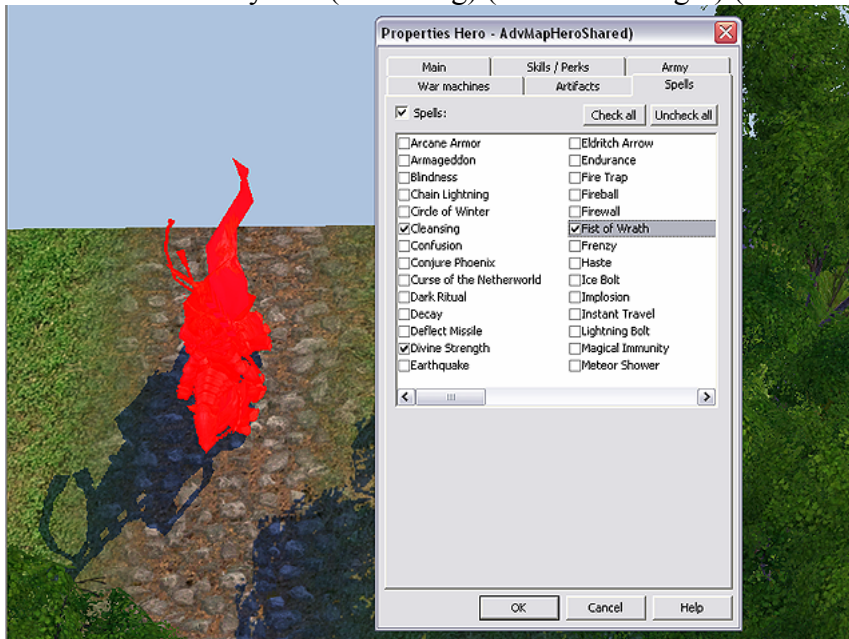
- *Presence of War Machines*
- *Presence of artifacts in his equipment*
- *Known spells*
- *Attributes and perks*
- *Qualitative and quantitative composition of the army*
- *Basic stats*

Here we add War Machines to Victor's army.

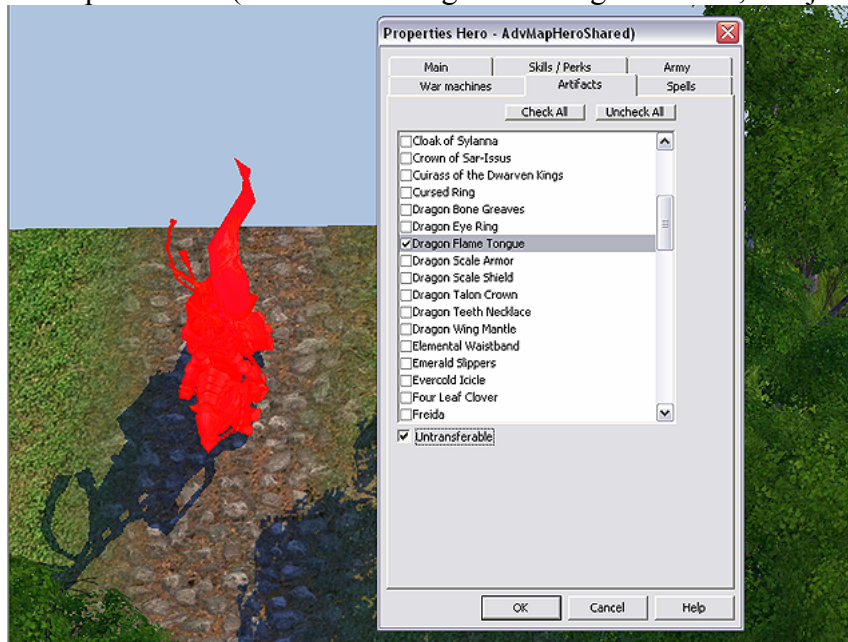


With spells, it's a bit more difficult. But this author has estimated what spells can our hero know, and...

...chose them. They are: (Cleansing) (Divine Strength) (Fist of Wrath)



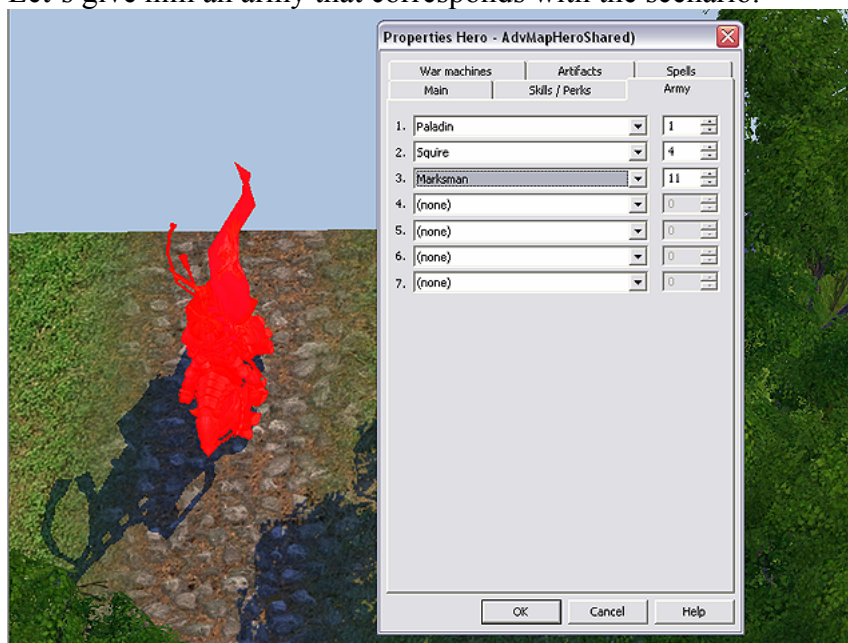
A simple artifact (for he's a distinguished knight after all, and just back from the battlefield).



Our hero's troops:

- 1 Paladin
- 4 Squires
- 11 Marksmans

Let's give him an army that corresponds with the scenario.



Now let's adjust his level and corresponding abilities/stats.

Let us adjust the level of the AI-controlled hero. According to the mission's conditions, our hero is a 1st-level one, so he needs no adjustment.

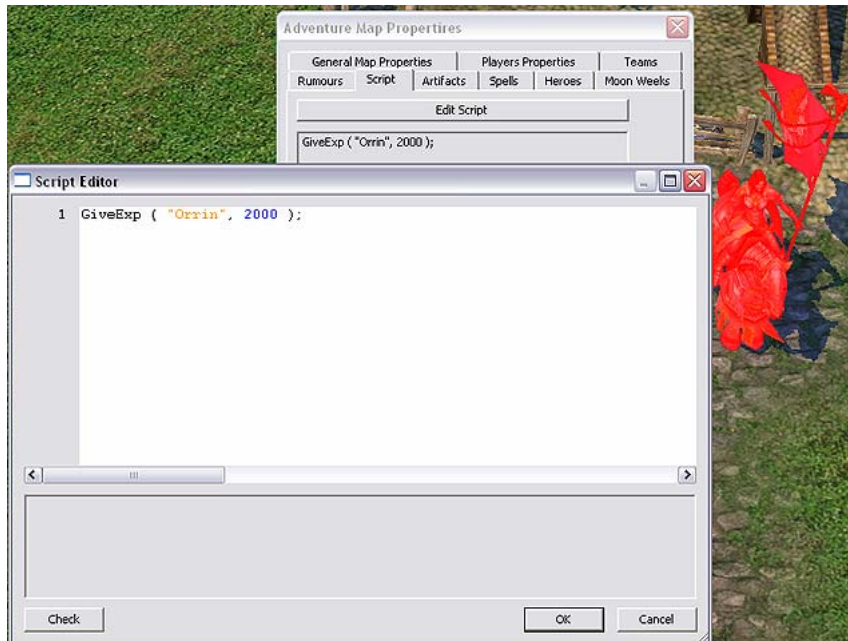
Open the Main tab in the hero's properties.

Due to the look-and-feel of the editor, it is impractical to add experience and modify the heroes' levels from the heroes' properties. That is why we shall be using the script to add experience and modify the hero's level accordingly.

To change the level of heroes that have been placed onto the map, we have to add them experience equivalent to their desired level. You can learn the correspondence between levels and experience from the appendix.

– open script with the code

```
GiveExp( "Orrin", 2000 );
```



These are the AI hero's scripting name and the amount of experience. See that the heroes' scripting names can not be modified; and this and nothing else is the name used in the script.

Insert the necessary code fragment into the mission's script.



a screenshot from the game itself, with the hero being given experience.

Thus the specified amount of experience will be given to the hero in the beginning of the mission, and, accordingly his stats will be modified automatically, while the Player (the user or the AI) will be allowed to select skills and perks for him.

So it follows that we do not really have to busy ourselves with distributing characteristics, skills and perks.

Adjusting the map

We only need the following tabs in the map properties:

1. General map properties – here we are only interested in the fields Map Name and Map Description. Level limitation does not interest us in this particular time (though usually, especially on maps with a long storyline, it can be useful to limit the heroes' maximal level; it's easier to plan and test the map when you know the highest level of heroes on it).
2. Players properties – here we shall make the following settings: choose Player 1 (the Player for which the user will be playing) and check in "Human Playable", while checking out "Computer Playable". For Player 2 (our AI-controlled opponent) we'll do everything the other way round. Such settings correspond to the original idea we have been following from the very beginning.
3. Script (see below).

Adjusting the towns

In the towns, the following tabs demand our attention:

1. Players – the town's belonging to this or that party (Player). For 'our' castle, we'll set it as None (which means it's neutral). The opponent's castle must belong to the opponent.
2. Buildings is an important tab. It allows us to define the initial 'advance' of the town, which means, in the first place, what troops can the hero hire there immediately after the mission begins. In our particular case, where we have a small map, this is especially significant.

On our map, we adjust the neutral town that has been Victor's castle previously and then was sacked by the enemies, as a town of very moderate advance. This means that almost all the buildings are absent.

This author offers to make the following choice:

Village Hall – Tier 1
Tavern
Peasant Huts – Tier 1
Training Grounds

Our opponent's castle, on the other hand, has not been looted, thus its development must be much better. But it must not be maximal: this castle has never belonged to a grand knight (like our Victor :)).

For the opponent's town, the following choice has been made:

Village Hall – Tier 2 (Town Hall)
Fort – Tier 1
Marketplace – Tier 1
Tavern
Magic Guild level one
Peasant Huts – Tier 2 (Peasant Cabins)
Archer Tower – Tier 1
Barracks – Tier 1
Griffin Tower – Tier 1
Farms

3. Garrison – here we choose the town's defenders (regardless of the town's being neutral or already controlled by this or that party).

For the neutral town, it's a good idea to have such an army that our hero would be able to defeat it after having gathered some troops that are willing to join him in the neighborhood.



The opponent's town does not need any special garrison since it initially belongs to the AI-controlled player. So let the AI take care of its defense.

4. Spells – here we define the pool of spells that will be available in this town's Magic Guild.

In our particular case (as well as in most of the other cases), it's not important to make detailed adjustments here. Only sometimes you may want to have a certain super-useful spell on the 5th tier of the magic tower in a certain town. ☺

5. Town Specialization: here we can select the town's specialization.

Town specialization defines what bonuses (in fact, insignificant) the town has. Usually this is made clear in the town's description.

In general, to avoid thinking too much, this author sets Talonguard-TOWN_NO_BONUS for both of the castles.

Adjusting the objectives

If you have read the manual's general section, you already know that all the three objective types that we have marked in the beginning are present in each map by default. That is why we shall not perform any special actions to adjust the objectives.

Adjusting the passability

As you might have already understood, HoMM5 maps are in fact branchy labyrinths, and the heroes' possible maneuvers in them are limited directly or indirectly.

Directly means that the hero can't reach this or that location because of impassible obstacles, such as forests, mountains, water bodies, or precipices.

Indirectly means temporary obstacles, like groups of monsters, garrisons, or Key Masters.

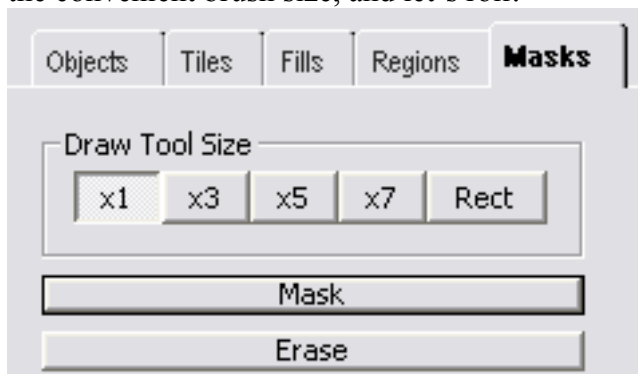
But if we review the passability of various regions of the map, we can clearly see that there are numerous 'bad' locations which may cause unplanned heroes travel routes to appear.

This screenshot shows a number of places where any hero can nick in.

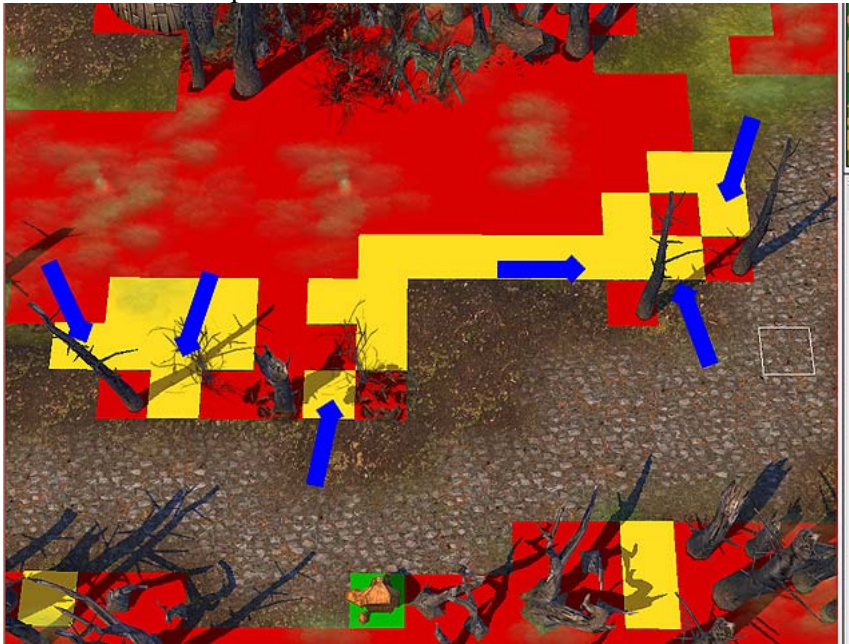


To correct such problems and to make the map's passability more 'beautiful' in general, there is a very useful tool called **Masks**.

Let us switch passability representation on and open the tab with the **Masks** tool. Now choose the convenient brush size, and let's roll!



Correct the 'bad' places.



By the way, the tiles marked with this tool are only colored in yellow when the tab with this tool is open. If you switch to any other tool while retaining the passability representation, you will see the marked tiles change their color.



Script

In general, becoming proficient in writing or editing scripts is not an easy task, and it's not possible to describe all its intricacies in a word. That is why most of simple single and multiplayer maps are created without writing a script, considering that the main functions (such as standard objectives and winning/losing conditions) are specified automatically in each map.

For our map, we shall create a very small and simple trigger, which will show how one can work with scripts in HoMM5.

The scenario for the script is the following: “If our hero appears in a specified region (which we shall mark on the map), remains of the troops once loyal to him will join him and hand over the family valuables salvaged from the sacked castle”. This, by the way, coordinates quite well with the general storyline of our mission.

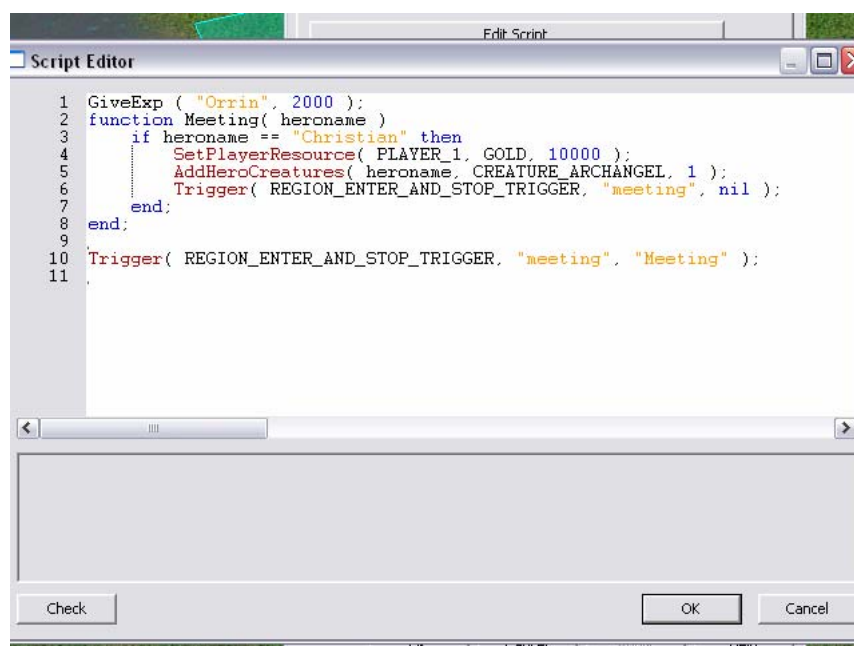
Let’s first make the region for the script: go to the Regions tab and create a small region in this approximate area. Give it some name, example, “meeting”. You can also tint it with any color you like.



(I) If you create more than one region on the map, make sure that they do not intersect.

Now go to the map’s properties. Open the **Script** tab.

And type in the following fragment:



(here the Script text, if u need it)

```
function Meeting( heroname )
    if heroname == "Christian" then
        SetPlayerResource( PLAYER_1, GOLD, 10000 );
        AddHeroCreatures( heroname, CREATURE_ARCHANGEL, 1 );
        Trigger( REGION_ENTER_AND_STOP_TRIGGER, "meeting", nil );
    end;
end;

Trigger( REGION_ENTER_AND_STOP_TRIGGER, "meeting", "Meeting" );
```

This example can be conventionally divided into two blocks: *The description of the function* (1) *and the use of this function* (2) *in the trigger* (3).

In the function, we describe what the player is given when the trigger works, and tell the trigger to work only once.

And then we use this function in the trigger that works when our hero finds himself in the zone 'meeting'.

It's all simple. You can find help in getting more to know about the scripts, first, on the www.lua.org website and then in the examples and counsels that are available at the official HoMM5 website and forum. Good luck!

Oh yes, and see the appendix for more on scripting commands. See the main part of the manual for more on the script editor.

Testing the map

Now that almost everything is ready, it's time for final testing.

And here is a word about testing in general. This is not a one-time-only process. Map can and should be tested virtually after each considerable operation since the very beginning of your work with it.

Accordingly, you should check up for all the problems described in this chapter throughout the whole work, not only in the end of it.

Testing the map has the following as its object:

- Making sure that the map has no visual artifacts or evident imperfections;
- Making sure that all the objects on the map have their belonging set up correctly;
- Verifying the map's passability;
- Testing the workability of the script to make sure that triggers work correctly, tasks are given and processed rightly, etc.;
- Adjusting the map's balance of powers, i. e. countervailing the heroes' possibilities to hire troops as well as the monsters that confront them;
- Check the map for its compliance with the workability criteria (see below).

Criteria to test the finished map, or “What could go wrong?”

Creating a map and a mission is often a long and difficult process. That is why some errors and problems can emerge that would cause unstable or incorrect performance of the mission or the game itself.

Let us check up our map according to the following list of possible mistakes.

(The list is divided into blocks by subject).

Players, towns, and heroes

- Each active Player has no more than 8 heroes;
- There are no more than 8 towns on the map;
- There must be at least one hero on the map, i. e. there must be at least one active Player;
- Heroes can't be neutral;
- Each hero has an army (at least one stack, and no stacks with zero number of creatures).

Tents & Guards

- There can only be one Tent and one Guard of each color on the map;
- If there is a Tent of this or that color, there must also be a Guard of the same color; and vice versa.

Objects and groups

- There must be at least 2 Subterranean Gates in each group;
- There must be at least 2 Whirlpools in each group;
- There must be at least 2 Two-way Monoliths in each group;
- There must be one entrance portal and at least one exit portal in each group of One-Way Portals.

Grail

- If there is an Obelisk on the map, there must also be the Grail; and vice versa.

Monsters

- There must be no monsters with zero number of creatures on the map.

Terrain

- There must be no tiles with surface type TT_NONE on the map;
- Verify the objects' bases where they intersect with precipices.

Map's general criteria

- For each floor of the map, there must be a correct type of AmbientLight specified;
- The map must have a correct number of Players and teams set correctly;
- If the map has a Sphinx, check its settings (the bonus artifact and the riddles);
- No intersecting regions (i. e. those sharing the same tiles).

(You can find other problems and recommendations on their solution described at the community websites and forums).

And remember to perform stock-taking of objects on the map when you test it: find the objects that are lost or unaccounted, etc. It's handy to use the Object List Panel or the standard search function for that.

Creating the map description and the PWL picture

Now that the map is virtually finished and our mission works correctly, there are some more things to take care about.

First, these are the author's identification and the map description. If you still haven't done so, don't forget to register yourself in the map properties.

Second, this is the picture shown on the screen while the map is loading. This picture is called PWL (abbreviated from Please Wait, Loading).

There are two things to be done here:

- Preparing the picture itself;
- Integrating it with the map.

The picture that you insert must have the size of 1024 x 1024 pixels. The PWL screen is shown in the resolution of 1024 x 768 pixels. You can base yourself on a screenshot from the game (like a close-up of a hero's figure) or any other picture that is not copyright protected.

If the original image you want to integrate with the mission has different size, the texture which you insert in the game must always have the size of 1024 x 1024 pixels.

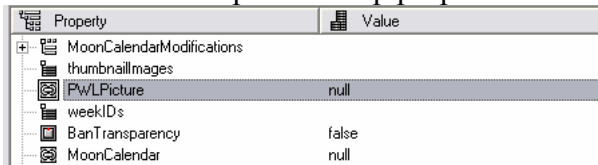
That is why, rescale or crop the image if you need to. If the image takes less space, fill everything that remains with some color (black, for example).

Here is an example of the original image and its appearance against the texture.

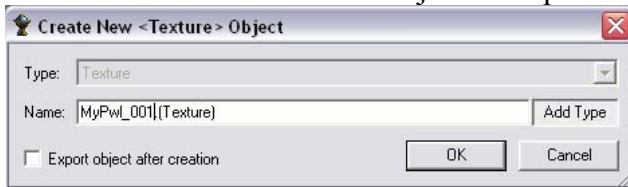


When the picture (texture) is ready for integration with the map, do the following:

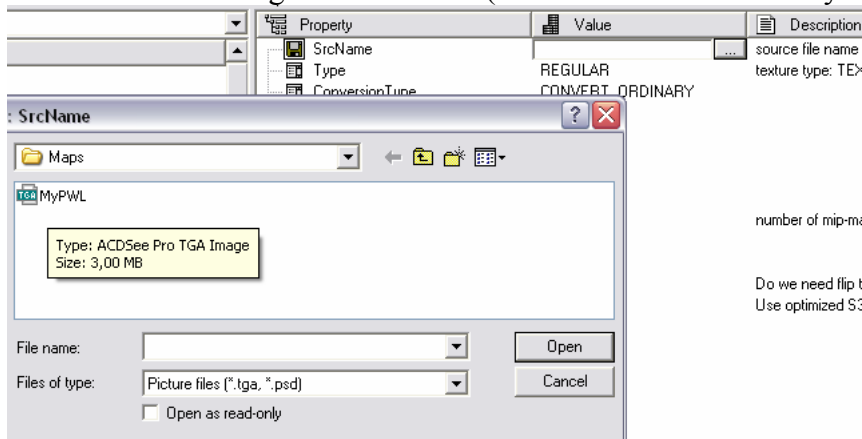
– open the map properties and find the line PWL Picture;



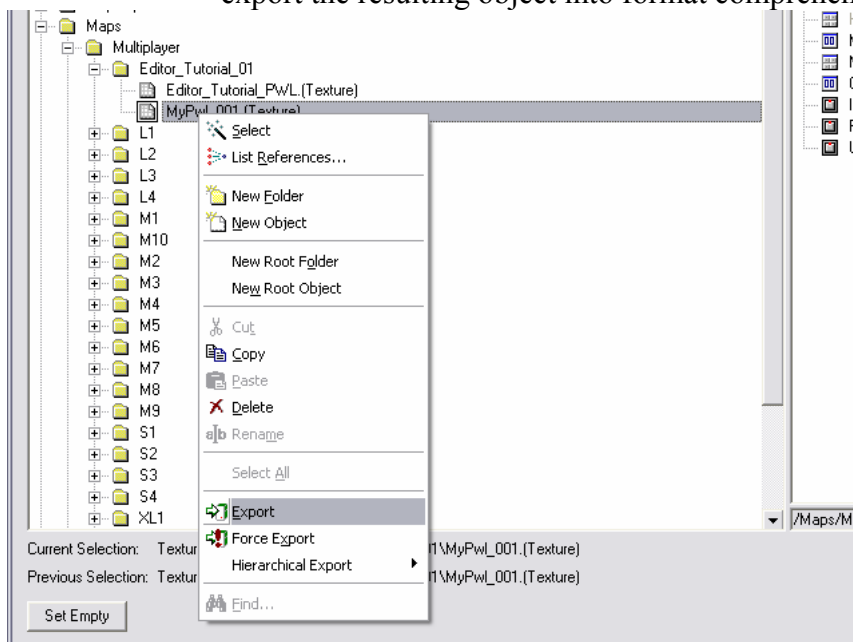
– create a new object and open it for editing;



– integrate the texture (which must be located in your map's folder).



– export the resulting object into format comprehensible for HoMM5..



And the last thing. In order for the mission's title and description to be shown at the map loading screen, you have to fill in the **NameFileRef** and **DescriptionFileRef** fields in the map properties.

Everything's ready! Now the specified picture will be shown while your map loads, accompanied by the mission's description.

Publishing your map

Here comes the most important thing. Once your map is brought to such perfection that you are willing to publish it at the HoMM5 community websites, you need to do so. The manual's general section tells how to save your map, open it, and edit other people's maps.

This author hopes to see numerous maps made by you in the vast expanses of the HoMM5 community!

The rest in brief

Music

As you have noticed, during the campaign or single player missions, the background, or ambient, music is being played.

Perhaps some of you could have an influence upon the choice of music played during the missions. But nothing can be done here: in HoMM5, all the music played during the missions is integrated with the game and selected automatically, depending on the race of the hero the player is controlling, the town which he visits, etc.

The Minimap

The minimap in the HoMM5 mission editor is very simple and handy. It shows in real time mode how the map's terrain changes, and represents clearly the border between individual types of tiles.



You can show or hide the minimap from the View menu (the MiniMap Panel item). You do not need to perform any special actions to refresh the minimap.

Cutscenes

If you have played the HoMM5 campaign, you might have seen many script-based movies. In the Objects tab, you can also see some objects that are designed for designing the script-based movies' arenas.

But creating script-based movies is not a process intended for an unprepared user. That is why this manual does not include any description of such work.

Underground level

As you have noticed, when you create a map, you can choose the Two Level Map option. Of course, many have seen these underground levels in the HoMM5 campaign.

What is these levels' difference from usual ones?

There are three substantial differences.

1. Different Ambient Light;
2. The principle of the level's composition;
3. The settings of various objects' luminescence.

Ambient Light is the background illumination and the texture of sky in the level. It affects the level's sunniness, the presence/absence of shadows, and the reflection of sky (its texture) in the water surfaces.

The principle of building an underground level is the following: while a usual level allows modifying the landscape with tools and objects right away, a special approach is demanded by the underground level, for first you need to create free room (caves and tunnels, by convention) in the Underworld. And only then you can use the tools and objects to modify the terrain (i. e., the second part is completely identical to your work with a usual surface level).

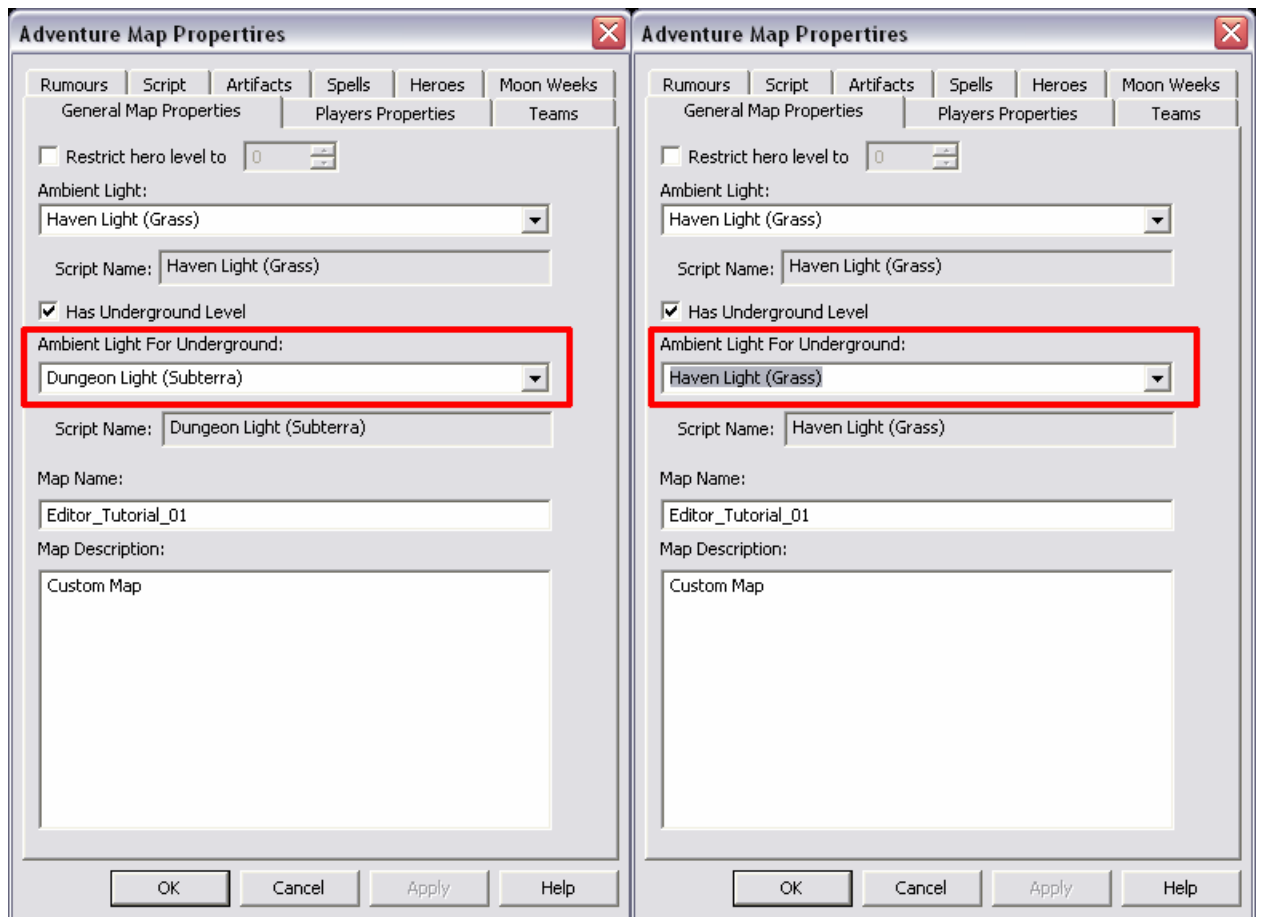
Here are some details concerning the peculiarities of the underground levels.

Changing the Ambient Light

The small description above makes it clear what for is Ambient Light needed. But if you create a map with an underground level, once you have switched to it, you will be just staggered by 'the dark side of the force': the Underworld is so dark and gloomy. This is done, of course, for the purpose of creating an atmosphere in the game. But it's not too easy to edit the underground level and fill it with objects in darkness like that.

To make your work in the Underworld more comfortable, the editor provides for a way to change the Ambient Light from standard to any other of your liking.

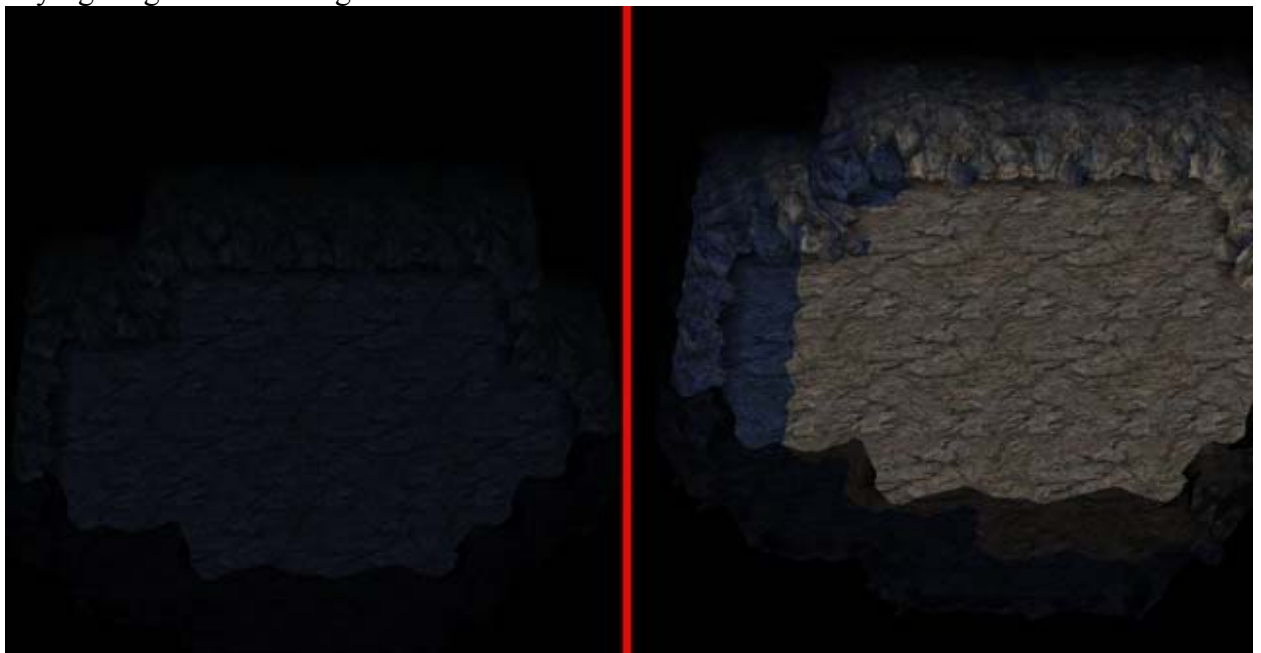
In our particular case, let us change the standard underground Ambient Light to that of the surface level. Such lighting allows seeing more clearly what is going on and how when you edit your Underworld.



(I) If you look into the map properties, you can see that each Ambient Light type has recommendation in the brackets after its name, which tells which type of surface (or underground) this illumination should better be used.

We only change the lighting for editing purposes. It's important not to forget to bring the authentic lighting back after we have finished editing.

To see the changes, you need to save and reload the map. The difference between standard and day lighting on this underground level is self-evident.



Creating caves, passages, tunnels, etc.

As it has been mentioned above, before we fill the level with objects and work with the terrain, we need to create some free room in this underground level, the room in which the heroes will spend their time, towns will expand, and the monsters make ambushes.

To do so, we shall use the **Lower** tool.

Go Tiles->Draw Tools Size=3->Terraforming=Lower, and use the brush to create some free room.

By the way, we are using the brush size of 3, because other sizes don't work when you need to create free room in the underground levels. This concerns, of course, only the Lower tool.

So now, we have prepared some free room for the heroes to explore and conquer.



Now you can do the same things that you have already done at the usual (surface) level. Of course it's advisable to use tiles and objects from the Subterranean category, for otherwise you may have some ugly results.

Let us now proceed to the what will illuminate our (and our heroes') way when we bring back the original Ambient Light: the various objects' luminescence.

Adjusting objects luminescence in the underground level

When you create an underground level, one of the most important tasks is adjusting the additional lighting that comes from the environment of the on-map objects.

Such objects have, among their characteristics, some parameters responsible for their luminescence.

They are:

- Coordinates (height) of luminescence;
- Radius of luminescence;

- Color of luminescence.

All these are adjusted through the object's properties. This list looks somewhat like that of the map properties but it's much shorter.

To see the object properties, you first need to turn the representation of the corresponding panel on. Go View and turn on the Selection Property Window.

Now select any object on the level and see its properties.

We are interested in the ***PointLights*** parameter.

Property	Value
Pos	
Rot	0
Floor	1
Name	
CombatScript	null
pointLights	
Shared	AdvMapStaticShared:MapObjects\Subte...
IsRemovable	false

Let us add the variable that we shall be working with.

Here you see approximately this kind of a tree.

Property	Value
Pos	
Rot	0
Floor	1
Name	
CombatScript	null
pointLights	
[0]	
Pos	
x	0
y	0
z	0
Color	
x	0
y	0
z	0
Radius	0
Shared	AdvMapStaticShared:MapObjects\Subte...
IsRemovable	false

Setting up the luminescence point coordinates.

The coordinates of the luminescence point are the position of the starting point of the luminescence (from which the light emanates) as regards to the parent object.

Usually, we are only interested in the Z coordinate, i. e. the point's vertical displacement.

1. Let us set the height as we need, but it is desirable to have it somewhere that the light would emanate from inside the object. Perhaps it's a good idea for it to come out from the middle of the object, in fact.
2. Now set up the luminescence radius according to your own preferences.
3. And the same for the color. Use any color you like.

Property	Value
Pos	
Rot	0
Floor	1
Name	
CombatScript	null
pointLights	
[0]	
Pos	
x	0
y	0
z	50
Color	
x	0
y	1
z	0.501961
Radius	50
Shared	AdvMapStaticShared:MapObjects\Subte...
IsRemovable	false

Of course it should be remembered that if the luminescence radius is zero, you won't see anything when you adjust its color. It is also advisable to choose the color that will be well-seen. "Bright Yellow", for example, would be somewhat better than "Dark Blue".



(I) A tip for those with creative mind: objects in the surface level can have their own luminescence set, too.

Multiplayer

The Multiplayer mode in HoMM5 has the following key distinguishing features:

- The presence of Ghost Mode;
- Each player's being able to choose the hero's race and the town's type;
- Equal conditions for all players.

Let us gain an understanding of what this all means for a map designer.

Allowing for the Ghost Mode in map design

The presence of Ghost Mode demands placing a special type of resource on the map: the Soul Stones.

When picked up, the Soul Stones bring an additional Ghost under the player's control.

Since the number of the players' Ghosts undergo the same limitations as that of the heroes, placing exceedingly many Soul Stones on the map is senseless.

(I) All Souls Stones placed on the map are converted to gold automatically when the map is played in single player mode.

Placing the Soul Stones is similar to placing usual resources.

Town and hero race depending on the player's choice

Every player's being able to choose the race and town type every time when a multiplayer map starts adds a couple of peculiarities to the process of placing race-dependable objects.

This, first of all, concerns towns and dwellings (objects that allow hiring troops every week). In particular, let us look at the object types Random Town and Random Dwelling Level #.

For the maps providing for arbitrary hero/town race selection, you should place not this or that specific object (town or dwelling) but rather Random Town / Random Dwelling objects.

The Random objects can be linked to each other so that the race (type) of the object was defined not just randomly but in dependence of the player's choice of race in the start of the map.

This is expressed in the following types of relations:

Town of the same race as that chosen by the Player;
Object (dwelling) of the same race (type) as that which Town N. now assumed.

Let us try an example.

Put a Random Town on the map.



Open the Random Type tab in its properties. (See that only Random objects have this tab).



Here we have three options:

- Full Random
- Like Player ...
- Like Town ...

The first option provides for randomizing independently.

The second option means that the choice depends on the race chosen by the player. I. e., if the player has chosen Haven as his or her town type, this Random Town will become a Haven town when the mission starts.

The third option provides for dependence on this or that town's race. If there are Random Dwellings in the vicinity of, say, a Haven town, it would be strange if they would produce creatures of a different race.

Equal conditions for all

Multiplayer maps and the multiplayer mode in general are first of all designed for finding out: which of the players is stronger under the same initial conditions.

That is, the map designer's task is to provide for more or less equal conditions for all players, to give advantage to none of them.

This is manifested in the following features:

- The map's symmetric design;
- The same winning / losing conditions for all.

So now, that's about all. Good luck in creating interesting multiplayer maps.

Searching and filtering on-map objects

When there are dozens of objects and groups of monsters on the map, you often find it difficult to calendar them all or find this or that particular one.

To make this task easier, the HoMM5 map editor has the Object List Panel, as well as a filtering tool.

Object List Panel

This panel allows you to see all objects on the map, including the environment objects like trees, grass, etc.

Each object has a name, type, scripting name (if the object is used in the script), on-map coordinates and size (calculated in tiles).

Besides, there is a handy search/filter tool available. Just type the name of the object in part or in full, and the list will show all the objects that meet the search condition.

This is a very handy tool both for searching for something specific and for selecting this or that object. Sometimes it happens that a placed object is then difficult to be selected for moving, removing, etc. So try to find it via the Object List Panel.

On-map objects filer

When you press Ctrl+H, the object filter selection window is invoked, where you can choose what object types should be shown or hidden. This can help you orientate on the map at the final stage when you adjust only some certain object types.

Hiding some object types (like, for example, vegetation) can also make the program's performance much better.