

Random Map Scripting Guide

Age of Empires® II: The Conquerors Expansion

Important: Age of Empires II: The Conquerors Expansion allows you to create your own random map scripts. You may share these custom random map scripts for the purposes of gameplay, but you may not sell or make other commercial uses of the custom random map scripts. Microsoft Corporation reserves all other rights to the editors and files.

COPYRIGHT NOTICE

This information contained in this publication/document (the “Information”) may be redistributed only as permitted by the text below and provided that such Information and this copyright notice remain intact. No part of the Information may in any form or by any electronic, mechanical, photocopied, recorded or any other means be reproduced, stored in a retrieval system, broadcast or transmitted without the prior written permission of the publisher of the Information, Ensemble Studios Corporation. Neither the Information nor any portion thereof may be stored in a computer or other electronic storage device except for personal and non-commercial use.

Users of the Information must seek permission from Ensemble Studios Corporation for all uses that are not allowed by fair use and other provisions of the U. S. Copyright Act, as amended from time to time (the “Act”). This Information may not, under any circumstances, be resold or redistributed for compensation of any kind without the prior written permission of Ensemble Studios Corporation, 10440 N. Central Expressway, Suite 1600, Dallas, TX 75231.

The Information may only be used as a source of data and may only be used for non-commercial purposes. Any copy of this Information or portion thereof must include this copyright notice.

Any published data contained in the Information, including URL and other Internet Web Site references, is subject to change from time to time without notice.

The Information does not carry a warranty of any kind, whether express or implied. Although every effort is made to provide the most accurate and up-to-date information possible, Ensemble Studios Corporation acknowledges that this Information could include certain errors, including typographical errors and technical inaccuracies. Additions and changes will be made to the Information on an on-going basis; thus the Information contained herein is subject to change without prior notice. Complying with the Act and all other applicable federal or state copyright laws is the sole responsibility of the user. The user hereby is deemed to assume all such copyright law compliance responsibility by his or her use of the Information.

Ensemble Studios Corporation may have U.S. federal and/or state patents, patent applications, trademarks, service marks, trade names, copyrights or other intellectual property rights covering subject matter in the Information. Except as

expressly provided in any written license agreement from Ensemble Studios Corporation, the furnishing of the Information does not give the user any license to such patents, patent applications, trademarks, service marks, trade names, copyrights or other intellectual property rights, which are expressly retained by Ensemble Studios Corporation.

Table of Contents

Introduction.....	3
Syntax.....	3
Map sizes.....	4
Global instructions.....	4
Local instructions.....	5
Generation Instructions.....	5
Script writing tips.....	15
Resources.....	15
Setting zones.....	15
Testing player start areas.....	15
Scattering objects.....	15
Making it all fit.....	15
Standard resources.....	16
Object names.....	16
Annotated Random Map Script.....	23

Introduction

Random Map scripts are text files that can be composed in any text editor. They must end with *.RMS and must be placed in the RANDOM directory. Random Map scripts are automatically transferred in multiplayer games to players who do not have them.

Random Map scripts place the same starting terrain and resources for every player. It is not possible to start different players with different resources. Thus, all scripts in theory will be fair to all players. However, knowledge of a map script can still help a player when playing on that map. Perhaps the player knows that there is a lot of gold generated in the center of the map (as in Gold Rush). When you play on someone else's Random Map script, you are placing some trust in that person that the map is going to be interesting and fair.

Computer Players should be able to play on any Random Map script, but unless you compose a new .AI file for that map, the Computer Player may not play with optimal efficiency. For example, the Computer Player may not build boats even if the map has ample water.

Making Random Maps that can consistently give each player the resources they need can be difficult and may involve a lot of trial and error. Don't get frustrated. Internet fan sites are a great resource for hints, tricks and other support for making maps that work.

A Random Map starts with Land first, then Elevation, then Terrain, and finally Cliffs and Connections. The order is important.

Syntax

Random map scripts are case sensitive.

Braces {} enclose commands under specific instruction. Opening and closing braces must be on their own line.

/* Slashes and asterisks enclose comments that the scripting system ignores, such as remarks, sub-headings, etc. Use /* to open a comment and */ to close it. Be sure to leave a space between the * and other characters.

Brackets <> enclose variables. These are not used in the random map scripts themselves, only in the context of this document. If an instruction says “<number>”, just type “35” not “<35>”. Instructions that begin with “set” do not require a variable.

Map sizes

Map label	Tiles on a side	Total tiles	Scaling factor
tiny_map	72x72	5,184	0.5
small_map	96x96	9,216	0.9
medium_map	120x120	14,400	1.4
large_map	144x144	20,736	2.1
huge_map	200x200	40,000	4.0
gigantic_map	255x255	62,025	6.2

Scaling factor can be used to place more objects on a larger map relative to a smaller map. A Gigantic Map is 12.4 times larger than a Tiny Map, even though one side of each map is only 3.4 times larger. Map labels can be used in if-then statements.

Example:

```
If medium_map
  create_object STONE
{
  number_of_objects      7
  number_of_groups       2
  group_placement_radius 3
}
endif
```

Global instructions

Global instructions are used to control script flow, make the script change from run to run, and make script control easier.

#define <label>

Defines a variable that you can use later on.

Example:

```
Start_random
  percent_chance 33
  #define DESERT_MAP
end_random
```

This instruction gives a 33% chance of the map being a “DESERT_MAP”. You could then say if DESERT_MAP later on to put down sand instead of grass or palm trees instead of oaks.

<const_name> **#number**

Constants are used to tie a name, such as an object or terrain, with its identifier in the database. Ensemble Studios has already pre-defined all necessary constants in the random_map.def file so they do not need to be changed.

```
if <label>
elseif <label>
else
endif
```

Conditional statements are used when there is more than one option. For example: If the map is gigantic, then add extra resources. If the map is a desert map, then use sand instead of grass. "If" initiates the conditional statement and "endif" completes it. Use "else" to specify all other conditions, or "elseif" to specify a single other condition.

Example:

```
if ALPINE_MAP
    base_terrain WATER
elseif DESERT_MAP
    base_terrain DESERT
else
    base_terrain GRASS2
endif
```

```
start_random
percent_chance <percent>
end_random
```

Random statements are used to specify an instruction running only part of the time. The random chance is specified as a percent (1-100). If the percents do not add up to 100, then the remaining percentage will apply towards a default setting. Everything between start_random and end_random is considered part of the same random instruction. Random statements can be nested inside each other.

Example:

```
start_random
    percent_chance 20
    #define DESERT_MAP
    percent_chance 20
    #define ALPINE_MAP
end_random
```

The map will be a desert 20% of the time, an alpine map 20% of the time, and a normal (probably grass) map the remaining 60% of the time. Alternatively, the instruction could have had the line to define grass map at 60%, but this is not necessary.

Local instructions

The following instructions are used to create a random map. Some instructions require a variable, which is generally a number #, a number of tiles #tiles, or a terrain constant <terrain constant>. Instructions that begin with "set" do not require a variable.

Generation Instructions

Before any specific instructions can be given, you must indicate what part of the random map the instructions refer to. For example, <TERRAIN_GENERATION> tells the random map generator that the instructions that follow refer to terrain. A Random Map starts with Land first, then Elevation, then Terrain and finally Cliffs and Connections. The order is important. The seven generation instructions are explained below along with their specific instructions.

<PLAYER_SETUP>	Places players
<LAND_GENERATION>	Creates types of land, such as islands and player lands
<TERRAIN_GENERATION>	Creates terrain, such as forests and desert
<OBJECTS_GENERATION>	Creates objects, such as trees, gold mines and villagers
<CONNECTION_GENERATION>	Connects different land zones such as shallows between islands
<ELEVATION_GENERATION>	Creates elevation to place hills
<CLIFF_GENERATION>	Creates cliffs

<PLAYER_SETUP>

random_placement

Places players randomly. This is the only valid entry under <PLAYER_SETUP>. All maps must have this line in them.

<LAND_GENERATION>

Land is used to create something aside from the default terrain type, most typically islands of grass in water. There are two types of land, normal lands (just called Lands) and Player Lands. In most random maps, Player Lands include a Town Center, Scout Cavalry, villagers, and the starting resources placed near the Town Center. Player Lands is also useful for telling terrain, such as forests, to avoid the player start areas. Land is all generated at the same time, so the order used in placing land is not important. (Terrain and objects, however, are placed in order.)

base_terrain <terrain constant>

Specifies what default terrain to start with. An island map probably uses water, while a grassland map uses grass.

Example:

```
base_terrain          WATER
```

create_player_lands {instructions}

Starts creating Player Lands. The percentage of land allotted to player lands is divided among all the players. Therefore, if player lands were specified to take up 20% of the map, then 2 players would each get 10%, but 4 players would each get 5%. It is important that player lands be large enough to contain a player's town. If player lands on an island map are too small, then the Town Center might be susceptible to fire from boats. On a hill country map, hills might be placed too close to the Town Center. Player lands are always set at a constant elevation of 2.

Example:

```
create_player_lands
{
    terrain_type      grass
    base_size         10
    land_percent      20
}
```

}

Creates an area for each player that is 10 tiles of grass in size.

create_land {instructions}

Creates an area of land that is not for players. What type of lands are there to make? In an island map, you might want to lay down bonus islands that contain gold. You might want to create a mountain range in the middle of a map.

terrain_type <terrain_constant>

Specifies what type of terrain to make the land. For Player Lands, this must be grass or desert to avoid goofy results.

land_percent <percent>

Defines what percent of the map is taken up by the land area. For Player Lands, this area will be divided by the number of players, so if the number is set to 60% and there are 6 players, each will start with about 10% of the map. Best results seem to occur when using 60-80%.

number_of_tiles <# tiles>

An alternate way of specifying land area size, in this case by number of tiles. Unlike Land_percent, these areas will not scale with map size. It is necessary to use only Land_percent or Number_of_tiles.

base_size <# tiles>

Specifies a minimum radius that the land grows from. This instruction is useful for insuring that each player has an area large enough to build a town and keep at least some buildings free from naval bombardment. If Base_size is not specified, Player Lands may be thin and snaky.

left_border <percent>
right_border <percent>
top_border <percent>
bottom_border <percent>

Percent from edge to stop land growth. This instruction recognizes the distance to the map border, so it is useful for placing terrain near the middle of the map. For example, defining a border of 25 will place land near the center of the map, while a border of 5 will place land almost to the edge of the map. In Mediterranean and Baltic maps, this instruction places the inland sea near the center of the map. In Continental maps, this instruction insures there is a border of water. "Top" is random on the maps, so just specifying Top and Bottom but not Left or Right can create a narrow strip of terrain. Note that the map land had a hard-coded feature to round off edges to make land look more natural. As maps get smaller (border > 20%) they may look less like rectangles and more like circles or octagons.

border_fuzziness <percent>

The percent chance per tile of stopping at a border. If this instruction is not used, borders will be straight lines. Specifying a low number, 5-20, will make edges ragged and more like real geography.

zone <# zone>

This is a descriptive command, used to assign a label to a certain land area. Zones with the same zone id can overlap, while land areas with different id's are distinct. If no zone is specified, each player will be on their own island (though it will only look like an island if the base terrain is water).

land_id <# id>

Assigns a label to a certain land area that can be used to assign objects only to that land.

set_zone_by_team

Will keep all players on a certain team in the same zone. For example, on an island map this will allow a team to share an island. To keep everyone on their own island, do not include this command.

set_zone_randomly

Will randomly determine zones, so that some players may be on the same island, while others may not be.

other_zone_avoidance_distance <# tiles>

This instruction is the one that specifies how large a land should be, so if it is not included, the land will not appear. Zone avoidance is used to specify the width of rivers between player lands.

assign_to_player <# player>

Assigns a land area to a certain player. Note, this does not work with player lands.

clumping_factor <# factor>

Clumping affects how much land tends to form squares instead of rectangles. The default value is 8, and the range is 1 to 15. Lower numbers tend to produce snaky islands while higher numbers tend to produce squares. Note that `clumping_factor` for land and for terrain have different ranges and defaults.

<TERRAIN_GENERATION>

Terrain is different from Land. Land is used to create something aside from the default terrain type, most typically islands of grass in water. Terrain includes features on that land, such as desert and forests. An important distinction is that land is placed down before elevation, but terrain is placed down after elevation. The order terrain is placed is also important. If you place palm desert on desert terrain before you place any desert terrain on the map, then you aren't going to see any palm trees. Remember, land is placed all at once. Terrain and objects are placed in order.

create_terrain <terrain constant > {instructions}

Creates a clump of terrain

Example:

```
create_terrain PALM_DESERT
{
  base_terrain                            DESERT
```



```
spacing_to_other_terrain_types 3
land_percent 1
number_of_clumps 3
set_avoid_player_start_areas
}
```

Creates 3 clumps of Palm Desert on Desert terrain. These clumps will take up a total of 1% of all tiles and will be placed away from Player Start Areas.

base_terrain <terrain constant >

Specifies what terrain type the new terrain will be placed on. For example, palm desert could be placed on desert, or grass3 could be placed on grass1.

land_percent <#percent of map to cover>

The percent of total land that this terrain will cover. It is best to use small percentages (1-10) for terrain if many different types of terrain are going to be laid down.

number_of_tiles <#size of terrain clump in tiles>

The size of terrain can be placed in terms of tiles instead of as a percent. Percent is generally more useful as it will automatically scale with map size.

number_of_clumps <#number of clumps to place>

The percent or number of tiles of terrain are evenly distributed into clumps. If number of clumps=3 and number of tile=18, the terrain will be appear as 3 clumps of 6 tiles each.

spacing_to_other_terrain_types <#distance from other terrain types>

Specifies how far terrain should be from other terrain (including terrain of the same type). This command is useful for preventing trees from becoming connected walls, or keeping forests away from water. It is a good idea to specify some distance for terrain that blocks movement, but is not necessary for terrain such as desert and grass.

set_scale_by_groups

Scales number of terrain clumps with map size (base 100x100 map). So, if 2 clumps are specified, then a large map would have 2 (2 x 1.4) clumps, but a Gigantic map would have 13 (2 x 6.5) clumps.

set_scale_by_size

Scales size of terrain patch with map size. So, if 10 tiles are specified, then a large map would have 140 (10 x 1.4) tiles, but a Gigantic map would have 650 (10 x 6.5) tiles.

set_avoid_player_start_areas

Most terrain, like forests and water, will avoid player start areas by default, but this instruction can be used, for example, if you want desert to avoid Town Centers.

clumping_factor <#factor>

Clumping affects how much land tends to form squares instead of rectangles. The default value is 20. Lower numbers tend to produce snaky patches while higher numbers tend to produce squares. Note that `clumping_factor` for land and for terrain have different ranges and defaults.

height_limits **<#minimum tile height>** **<#maximum tile height>**

Specifies on what elevation terrain can be placed (from 0-8). This instruction can be used to place grass on hill tops or place water only in depressions.

set_flat_terrain_only

Instructs terrain to avoid hills. Useful for making sure ponds don't cross more than one elevation.

<OBJECTS_GENERATION>

Once land, elevation and terrain are placed, you are ready to place objects. Note that some objects may already be placed by terrain. Forests, for example, will be filled with tree objects. However, you will need to specify starting units, resources and other objects needed to make a map have more character and detail. Remember, land is placed all at once. Terrain and objects are placed in order.

create_object **<object constant > {instructions}**

Places an object.

Example:

```
create_object GOLD
{
  number_of_groups           2
  number_of_objects         3
  group_placement_radius    2
  set_tight_grouping
  set_gaia_object_only
  min_distance_to_players   40
  min_distance_group_placement 9
}
```

Gives every player 2 piles of gold with 3 gold objects in each pile. The pile is confined to a radius of 2 tiles and is placed from 9 to 40 tiles from the center of a player's lands.

min_distance_to_players **<#number of tiles>**
max_distance_to_players **<#number of tiles>**

Specifies the limits where the object can be placed relative to the center of a piece of land. The object will appear at random anywhere between the minimum and maximum. For example, a Town Center placed at minimum 0 and maximum 0 will appear in the center of each player's lands. A gold mine placed at 10 to 12 will appear about one screen away from the Town Center, but not too far away. The defaults for these instructions are 0 and infinity, respectively, so if no distance is specified, the object will appear somewhere on that land. Be careful not to specify a minimum that is greater than the land radius, or the object may not appear or may appear in the wrong location. Remember that when used with `place_on_specific_land_id` instead of `set_place_for_every_player` that the distances refer to the center of that piece of land and not to the players.

set_scaling_to_map_size

Scales the number of groups to larger or smaller maps. When this instruction is used, `number_of_groups` will apply only to Large Maps. For example, if 6 groups are specified on a Large Map, then a Medium Map will have 4 groups, and a Small Map will have 2. Note: If `number_of_groups` is not used (essentially making all the objects separate), then scaling will apply to the objects themselves. This is a good way to scatter things like fish and trees on a map. Specify a large number of objects but no groups and then use `set_scaling_of_groups_to_map_size`. Alternatively, you can create a large number of groups with one object each. The effect will be the same. Note that scaling can be set to player number or map size, not both. However, it is possible to have different objects of the same type set to different scales. One gold mine could scale to player number, while another scales to map size.

Example:

```
create_object FISH
{
  number_of_objects           50
  set_scaling_to_map_size
  terrain_to_place_on        WATER
  set_gaia_object_only
  min_distance_group_placement 4
}
```

Creates 50 fish (on a Large Map, fewer on a smaller map, more on a larger map) scattered across the water, but never more than 4 tiles from another fish.

set_scaling_to_player_number

Note that scaling can be set to player number or map size, not both. However, it is possible to have different objects of the same type set to different scales. One gold mine could scale to player number, while another scales to map size.

min_distance_group_placement <#tiles>

Distance to separate center of a group—prevents a massive wad of gold, stone and berries all together. Just as in `set_scaling_of_groups_to_map_size`, if no groups are assigned, then this instruction will apply to all objects. Note: `group_placement_radius`, `set_loose_grouping`, and `set_tight_grouping` will all override scattering of objects and keep a group together.

Example:

```
create_object FISH
{
  number_of_objects           50
  set_scaling_of_groups_to_map_size
  terrain_to_place_on        WATER
  set_gaia_object_only
  min_distance_group_placement 4
}
```

Creates 50 fish (on a Large Map, fewer on a smaller map, more on a larger map) scattered across the water, but never more than 4 tiles from another fish.

max_distance_to_other_zones <#tiles>

Specifies how close the objects can be to other zones. This is useful if keeping objects away from the shore (and enemy ships.)

number_of_objects <# objects>

Specifies how many objects are placed, e.g. gold mines in a patch of mines. If no groups are specified, then there will be one group for each object. In other words, the objects will all be scattered.

number_of_groups <# groups>

Specifies the number of groups placed. Each group will have the number of objects specified in `number_of_objects`. If no groups are specified, then there will be one group for each object. In other words, the objects will all be scattered.

group_variance <# number added or subtracted from
number_of_groups>

Sometimes you want some randomness in the number of objects placed. `group_variance` will add or subtract from the number of objects in a group. For example, a group of 3 deer with a variance of 2 will actually place from 1 to 5 deer at random.

group_placement_radius <#tiles>

Specifies how much area is occupied by a group. Groups of large objects can become very long if the objects are placed in a row. Specifying a small `group_placement_radius` will confine a group to a smaller area.

set_loose_grouping
set_tight_grouping

Loose groups can have a tile or two of space among the objects, like sheep or deer. Tight groups have no space among the objects, like gold or stone.

terrain_to_place_on <terrain constant >

Confines the group to a certain terrain. For example, you can place gold mines only on desert terrain. If no terrain is specified, the object will be placed anywhere on the map within reason. Be sensible--some objects can only be placed on water (like fish) and others can only be placed on land (like villagers and trees).

set_gaia_object_only

Used if you want an object to be Gaia (belonging to no player), such as sheep or bonus units hidden on the map. Will not affect objects with no inherent ownership, like gold or trees.

set_place_for_every_player

Places a group for every player who joins the game. If you want there to be one Relic for every player, or one Town Center for every player, use this command. If you want every player to start with a pile of gold, use this command. If you want an extra pile of gold lying out in the wilderness, do not use this command.

place_on_specific_land_id <land id>

If you designated a piece of land as a particular id, you can now use that id to place objects only on this land. On the Crater Lake map, this command is used to place the bonus gold on the spire at the center of the inland lake.

<ELEVATION_GENERATION>

After land is placed, but before terrain is placed, you can specify how hilly a section of land is. Elevation is laid down basically like terrain. Base_terrain, number_of_clumps, number_of_tiles, set_scale_by_groups and set_scale_by_size all work for elevation the way they do for terrain. Elevations always avoid player start areas.

create_elevation <# maximum height> {instructions}

Places hills. The range of elevation is from 1 to 7. Note that elevations are not placed at the specified elevation but up to the specified elevation. Thus if you state elevation 6, you will get some elevations at 1-6. Stating elevation 7 will create 1-7.

Example:

```
create_elevation                      7
{
if DESERT_MAP
  base_terrain                      DIRT
elseif ASIAN_MAP
  base_terrain                      GRASS2
else
  base_terrain                      GRASS
endif
  number_of_clumps                  14
  number_of_tiles                   1000
  set_scale_by_groups
  set_scale_by_size
}
```

Places 1000 tiles of up to level 7 elevation on dirt, grass2 or grass.

<CLIFF_GENERATION>

Cliffs can add extra diversity to maps with a lot of open space, though they may crowd smaller islands.

Example:

```
min_number_of_cliffs 5
max_number_of_cliffs 8
min_length_of_cliff 4
max_length_of_cliff 10
cliff_curliness 10
min_distance_cliffs 3
```

Places 5-8 cliffs of 4-10 length on the map, but keeps them 3 tiles away from each other.

min_number_of_cliffs <#number of cliffs>
max_number_of_cliffs <#number of cliffs>

Specifies a range for how many cliffs are placed on a map. Specifying a minimum of 1 and a maximum of 12 will place 1 to 12 cliffs on a map.

min_length_of_cliff <#number of tiles>
max_length_of_cliff <#number of tiles>

Specifies the length of each cliff.

cliff_curliness <#percent>

Specifies the percent chance a cliff will turn instead of continuing straight. High numbers will produce zigzag cliffs, while low numbers will produce straight lines. As with everything, experiment.

min_distance_cliffs <#tiles>

In order to keep cliffs from forming too close to other cliffs, specify a minimum distance.

<CONNECTION_GENERATION>

Connections are lines drawn among player lands. Connections can be used to place roads between Town Centers, to place shallows across rivers, or to cut clearings through forests.

create_connect_teams_lands {Instructions}
create_connect_all_lands {Instructions}
create_connect_all_players_land {Instructions}

Land can be connected just among team members, among all players, or among all lands placed on a map. More than one connection can be placed. For Black Forest, one set of connections opens up paths among players while another set places roads just among team members.

Example:

```
<CONNECTION_GENERATION>
create_connect_all_players_land
{
  replace_terrain WATER          SHALLOW
  replace_terrain MED_WATER      SHALLOW
  replace_terrain DEEP_WATER     SHALLOW
  terrain_cost    WATER          7
  terrain_cost    MED_WATER      9
  terrain_cost    DEEP_WATER     15
  terrain_size    WATER          3      1
  terrain_size    MED_WATER      3      1
  terrain_size    DEEP_WATER     3      1
}
```

Connects every player by placing shallows across water, with a preference for shallow water over deep water.

replace_terrain <terrain_constant> <terrain_constant>

Usually when you make connections, you want to replace one terrain with another, such as replacing water with shallows or replacing forest with road. You need to use a replace terrain command for each type of terrain placed on a map. So, if you mix grass1, grass2 and grass3, you need a replace terrain command for each.

terrain_cost <terrain_constant> <#cost to pass through a tile>

Assigning costs to terrain can force a connection to take a different route. If you want your shallows to avoid deep water and try and follow shallow water, then assign a high cost to deep water (like 15) and a lower cost to shallow water (like 7).

terrain_size <terrain_constant> <#tile radius> <#variance>

You need to specify how wide a connection is going to be. A three-tile wide stretch of shallows will be easier to defend than a fifteen-tile wide shallows. For each terrain through which a connection may pass, specify a tile radius and a variance on that radius. For example, `terrain_size MED_WATER 3 1`, create shallows that are 1-4 tiles (3 +/- 1) in width.

Script writing tips

Resources

On Ensemble Studios map types, every player starts with a set number of resources (two gold mines, for example). Additionally, extra resources are added for each player using `set_scaling_to_map_size` or by adding if statements using map size as the condition. Finally, the larger map sizes have additional resources.

Setting zones

Many Ensemble Studios random maps use the random statement to specify one of four zone types: set zone by team, set zone to a specific id, set zone randomly, or don't set zone at all. This ensures that some of the time the players on a team will share a zone (set zone by team), some of the time they all players will share a zone (set zone to a specific id), some of the time some players will overlap (set zone randomly), and some of the time players will all be separate (don't set zone at all). Zones have the greatest impact on play when the base terrain type is water, which will make zones equivalent to islands. Note that in the following script, the default (no zone at all) will occur 40% of the time.

```
start_random
  percent_chance      20
    set_zone_by_team
  percent_chance      20
    zone               1
  percent_chance      20
    set_zone_randomly
end_random
```

Testing player start areas

Different lands and player lands can be temporarily set to terrain types to distinguish different land areas. Setting the base terrain to grass and the player lands to desert will reveal where player lands occur. If you are trying to have a large open area of any terrain type, a Mediterranean-style map is a good place to start, as it creates a large "lake" near the middle of the map.

Scattering objects

There are two ways to lay down objects, individually or in groups. If `number_of_groups` is not specified, then each object is treated as its own group and will respond to the `set_scaling_of_groups_to_map_size` and `min_distance_group_placement` instructions, unless instructions to keep the group together override it. Instructions to keep a group together are `group_placement_radius`, `set_loose_grouping`, and `set_tight_grouping`. Standard Ensemble Studios maps scatter fish, but keep gold and stone mines in tight groups.

Making it all fit

Land is placed all at once but terrain is placed in the order you specify, and then objects are placed on top of terrain. If you place down too much forest, for example, there might not be enough open space for all of the gold and stone. The map generator does not try several iterations until it creates a map that works—if there is not enough space for boar for player 2, then player 2 will not have any boar. If your maps seem like they are missing resources, you can try relaxing the constraints on where the resources are placed. Try decreasing the minimum and increasing the maximum distance the resources can be placed from a Town Center. You may have to ultimately cut back on the amount of water or forests on a map.

Standard resources

Many (but not all) Ensemble maps use the same amount of starting resources so that players know what to expect. There are exceptions, however, including Scandinavia's lack of berries and Yucatan's extra food. You can find the standard resources placed according to map size and player number in the map description below. It might be a good idea to start with these resources so that your map will not seem too difficult for other people to play on it.

The standard starting resources include:

- 6 Berries
- 1 group of 7 gold mines
- 2 groups of 4 gold mines
- 1 group of 5 stone mines
- 1 group of 4 stone mines
- 1 group of 4 sheep
- 2 groups of 2 sheep
- 1 group of 4 deer
- 2 boar
- 2 wolves
- 1 group of 3 straggler trees
- 1 group of 2 straggler trees

Object names

These are the names of objects and terrain that the Random Map generator knows. You must refer to these names when placing terrain or objects.

MAP NAMES

ARABIA
ARCHIPELAGO
ARENA
BALTIC
BLACK_FOREST
COASTAL
CONTINENTAL

CRATER_LAKE
FORTRESS
GHOST_LAKE
GOLD_RUSH
HIGHLAND
ISLANDS
MEDITERRANEAN
MIGRATION
MONGOLIA
NOMAD
OASIS
RIVERS
SALT_MARSH
SCANDANAVIA
TEAM_ISLANDS
YUCATAN

GAME NAMES

KING_OF_THE_HILL
REGICIDE

TERRAIN NAMES

BAMBOO
BEACH
DESERT
DIRT
DIRT2
DIRT3
FOREST
GRASS
GRASS2
GRASS3
ICE
SNOW
GRASS_SNOW
DIRT_SNOW
JUNGLE
LEAVES
PALM_DESERT
PINE_FOREST
SNOW_FOREST
SHALLOW
WATER
MED_WATER
DEEP_WATER
ROAD
ROAD2

OBJECT NAMES, GAIA

BAMBOO_FOREST_TREE
BAMBOO_TREE
BOAR
BROKEN_CART

CACTUS
CARAVAN
CRACKS
CRATER
DEER
DIRE_WOLF
DORADO
FLOWER_BED
FLOWERS_1
FLOWERS_2
FLOWERS_3
FLOWERS_4
FORAGE
FORAGE_BUSH
FOREST_TREE
GOLD
GRAVE
HAWK
HAY_STACK
HEAD
IRON_BOAR
JAGUAR
JAVELINA
JUNGLE_TREE
MACAW
MARLIN1
MARLIN2
MOUNTAIN_1
MOUNTAIN_2
MOUNTAIN_3
MOUNTAIN_4
OAK_FOREST_TREE
OAKTREE
OLD_STONE_HEAD
PALM_FOREST_TREE
PALMTREE
PATH_1
PATH_2
PATH_3
PATH_4
PIECE_OF_THE_TRUE_CROSS
PINE_FOREST_TREE
PINETREE
PLANT
RABID_WOLF
RELIC
ROCK
ROMAN_RUINS
RUGS
RUINS
SALMON
SEA_ROCKS_1
SEA_ROCKS_2
SHEEP
SHORE_FISH
SIGN

SKELETON
SNAPPER
SNOW_PINE_TREE
STATUE
STONE
STUMP
TREE_A
TREE_B
TREE_C
TREE_D
TREE_E
TREE_F
TREE_G
TREE_H
TREE_I
TREE_J
TREE_K
TREE_L
TREE_TD
TREE1
TREE2
TREE3
TREE4
TREE5
TUNA
TURKEY
WILD_BOAR
WILD_HORSE
WOLF

OBJECT NAMES, SCENARIO

CATHEDRAL
ES_FLAG
FLAG_A
FLAG_B
FLAG_C
FLAG_D
FLAG_E
GREAT_PYRAMID
HORSE
KING
MAP_REVEALER
MONUMENT
MOSQUE
NINE_BANDS
PAVILION
PAVILION2
PAVILION3
PYRAMID
RUBBLE_1_X_1
RUBBLE_2_X_2
RUBBLE_3_X_3
SHEEP
SHIPWRECK
SHIPWRECK2

TORCH
TORCH_CONVERTING
TRADE_WORKSHOP
YURT
YURT2
YURT3
YURT4
YURT5
YURT6
YURT7
YURT8

OBJECT NAMES, ARCHERY RANGE UNITS

ARCHER
CAVALRY_ARCHER
HAND_CANNONEER
SKIRMISHER
ARBALEST
CROSSBOWMAN
ELITE_SKIRMISHER
HEAVY_CAVALRY_ARCHER

OBJECT NAMES, BARRACKS UNITS

EAGLE_WARRIOR
MILITIA
SPEARMAN
CHAMPION
ELITE_EAGLE_WARRIOR
HALBERDIER
LONG_SWORDSMAN
MAN_AT_ARMS
PIKEMAN
TWO_HANDED_SWORDSMAN

OBJECT NAMES, CASTLE UNITS

BERSERK
CATAPHRACT
CHU_KO_NU
CONQUISTADOR
HUSKARL
JAGUAR_WARRIOR
JANISSARY
LONGBOWMAN
MAMELUKE
MANGUDAI
PETARD
PLUMED_ARCHER
SAMURAI
TARKAN
TEUTONIC_KNIGHT
THROWING_AXEMAN
TREBUCHET
TREBUCHET_PACKED

WAR_ELEPHANT
WAR_WAGON
WOAD_RAIDER
ELITE_BERSERK
ELITE_CATAPHRACT
ELITE_CHU_KO_NU
ELITE_CONQUISTADOR
ELITE_HUSKARL
ELITE_JAGUAR_WARRIOR
ELITE_JANISSARY
ELITE_LONGBOWMAN
ELITE_MAMELUKE
ELITE_MANGUDAI
ELITE_PLUMED_ARCHER
ELITE_SAMURAI
ELITE_TARKAN
ELITE_TEUTONIC_KNIGHT
ELITE_THROWING_AXEMAN
ELITE_WAR_ELEPHANT
ELITE_WAR_WAGON
ELITE_WOAD_RAIDER

OBJECT NAMES, CHURCH UNITS

MISSIONARY
MONK

OBJECT NAMES, DOCK UNITS

CANNON_GALLEON
DEMOLITION_SHIP
FIRE_SHIP
FISHING_SHIP
GALLEY
LONGBOAT
TRADE_COG
TRANSPORT_SHIP
TURTLE_SHIP
ELITE_CANNON_GALLEON
ELITE_LONGBOAT
ELITE_TURTLE_SHIP
FAST_FIRE_SHIP
GALLEON
HEAVY_DEMOLITION_SHIP
WAR_GALLEY

OBJECT NAMES, HERO UNITS

ADMIRAL_YI_SUN_SHIN
AETHELFIRTH
ARCHBISHOP
ARCHERS_OF_THE_EYES
ATTILA_THE_HUN
BAD_NEIGHBOR
BAD_NEIGHBOR_PACKED
BELISARIUS

BLEDA_THE_HUN
CHARLEMAGNE
CHARLES_MARTEL
CONSTABLE_RICHEMONT
DOME_OF_THE_ROCK
DUKE_D_ALENCON
EL_CID
EL_CID_CAMPEADOR
EMPEROR_IN_A_BARREL
ERIK_THE_RED
FRANKISH_PALADIN
FRIAR_TUCK
GAWAIN
GENGHIS_KHAN
GODS_OWN_SLING
GODS_OWN_SLING_PACKED
GUY_JOSSELYNE
HARALD_HARDRAADE
HENRY_V
HROLF_THE_GANGER
HUNTING_WOLF
IMAM
JEAN_BUREAU
JEAN_DE_LORRAIN
JOAN_OF_ARC
JOAN_THE_MAID
KING_ALFONSO
KING_ARTHUR
KING_SANCHO
KITABATAKE
KUSHLUK
LA_HIRE
LANCELOT
LORD_DE_GRAVILLE
MASTER_OF_THE_TEMPLAR
MINAMOTO
MORDRED
NOBUNAGA
ORNLU_THE_WOLF
POPE_LEO_I
REYNALD_DE_CHATILLON
RICHARD_THE_LIONHEARTED
ROBIN_HOOD
ROLAND
SABOTEUR
SCYTHIAN_SCOUT
SCYTHIAN_WILD_WOMAN
SHAH
SHERIFF_OF_NOTTINGHAM
SIEGFRIED
SIEUR_BERTRAND
SIEUR_DE_METZ
SIR_JOHN_FASTOLF
SUBOTAI
TAMERLANE
THE_ACCURSED_TOWER

THE_BLACK_PRINCE
THE_TOWER_OF_FLIES
THEODORIC_THE_GOTH
WILLIAM_WALLACE
WILLIAM_THE_CONQUEROR
WILLIAM_THE_CONQUEROR2

OBJECT NAMES, MARKET UNIT

TRADE_CART

OBJECT NAMES, SIEGE WORKSHOP UNITS

BATTERING_RAM
BOMBARD_CANNON
MANGONEL
SCORPION
CAPPED_RAM
HEAVY_SCORPION
ONAGER
SIEGE_ONAGER
SIEGE_RAM

OBJECT NAMES, STABLE UNITS

CAMEL
KNIGHT
SCOUT
SCOUT_CAVALRY
CAVALIER
HEAVY_CAMEL
HUSSAR
LIGHT_CAVALRY
PALADIN

OBJECT NAMES, TOWN CENTER UNIT

VILLAGER

OBJECT NAMES, BUILDING

ARCHERY_RANGE
BARRACKS
BLACKSMITH
BOMBARD_TOWER
CASTLE
DOCK
FARM
FISH_TRAP
FORTIFIED_WALL
GUARD_TOWER
HOUSE
KEEP
LUMBER_CAMP
MARKET
MILL

```

MINING_CAMP
MONASTERY
OUTPOST
PALISADE_WALL
SIEGE_WORKSHOP
STABLE
STONE_WALL
TOWN_CENTER
UNIVERSITY
WALL
WATCH_TOWER
WONDER

```

Annotated Random Map Script

The following script was created by Ensemble Studios for the Coastal Random Map type. You can use it to figure out how to start maps of your own and see what standard resources go into most maps. Remember that everything between `/* slash marks */` is a comment and is not read by the Map Generator.

```

/* ***** COASTAL ***** */
/* 29 MAR 00 */

/* ***** */
<PLAYER_SETUP>
    random_placement

/* ***** */
<LAND_GENERATION>

base_terrain WATER
/* Coastal is basically a large island placed on water, so water is the base terrain. Using water1 (the
shallow water) makes it easier to add deeper water later */

start_random
    percent_chance 15
    #define DESERT_MAP
    percent_chance 15
    #define ALPINE_MAP
    percent_chance 15
    #define ASIAN_MAP
    percent_chance 15
    #define FROZEN_MAP
    percent_chance 15
    #define TROPICAL_MAP
end_random

/* Coastal maps can have different terrain, such as desert or rain forest. There is a 15% chance of the map
using one of these different terrain sets. These names are just used as variables. You can define your own.
*/

create_player_lands
{
if DESERT_MAP
    terrain_type                DIRT

```



```

elseif ALPINE_MAP
    terrain_type                GRASS2
elseif ASIAN_MAP
    terrain_type                GRASS2
elseif FROZEN_MAP
    terrain_type                SNOW
else
    terrain_type                GRASS
endif
/* Depending on what terrain type is chosen, we use a different base terrain */

start_random
    percent_chance 33
    land_percent    52
    percent_chance 33
    land_percent    60
    percent_chance 33
    land_percent    65
end_random
/* The amount of land used is random. Using 52% means there is often a lake in the middle of the map.
Using 65% means that there tends to be a large, square landmass */
base_size                8
start_random
percent_chance 25
    left_border           6
    top_border            8
    bottom_border         6
percent_chance 25
    left_border           6
    top_border            6
    right_border          8
percent_chance 25
    right_border          6
    left_border           6
    bottom_border         8
percent_chance 25
    right_border          8
    top_border            6
    bottom_border         6
end_random
/* There is a random chance for various border sizes. Coastal always has one edge that touches the border
of the map, but it is random which border this is */
border_fuzziness        15
zone                    1
other_zone_avoidance_distance 7
clumping_factor         15
}

/* ***** */
<TERRAIN_GENERATION>

/* MIXING WATER */

create_terrain MED_WATER
{
base_terrain WATER

```

```
number_of_clumps          10
spacing_to_other_terrain_types 2
land_percent 40
}
```

```
create_terrain MED_WATER
{
base_terrain WATER
number_of_clumps          30
spacing_to_other_terrain_types 1
land_percent 1
}
```

```
create_terrain DEEP_WATER
{
base_terrain MED_WATER
number_of_clumps          8
spacing_to_other_terrain_types 3
land_percent 20
}
```

```
create_terrain DEEP_WATER
{
base_terrain MED_WATER
number_of_clumps          30
spacing_to_other_terrain_types 1
land_percent 1
}
```

```
create_terrain MED_WATER
{
base_terrain DEEP_WATER
number_of_clumps          30
spacing_to_other_terrain_types 1
land_percent 1
}
```

```
create_terrain WATER
{
base_terrain MED_WATER
number_of_clumps          30
spacing_to_other_terrain_types 1
land_percent 1
}
```

/* This adds deeper water to the shallow water. We use small percentages and spacing of 1-3 tiles to make sure the shallow water is closest to land and other waters are placed farther from land. Experiment with different percentages of water so that there are not large expanses of one color. */

```
/* PRIMARY FOREST */
```

```
if DESERT_MAP
  create_terrain PALM_DESERT
  {
    base_terrain          DIRT
    spacing_to_other_terrain_types 5
  }
```

```

    land_percent                9
    number_of_clumps            10
    set_avoid_player_start_areas
    set_scale_by_groups
}
elseif ALPINE_MAP
create_terrain PINE_FOREST
{
    base_terrain                GRASS2
    spacing_to_other_terrain_types 5
    land_percent                9
    number_of_clumps            10
    set_avoid_player_start_areas
    set_scale_by_groups
}
elseif FROZEN_MAP
create_terrain SNOW_FOREST
{
    base_terrain                SNOW
    spacing_to_other_terrain_types 5
    land_percent                9
    number_of_clumps            10
    set_avoid_player_start_areas
    set_scale_by_groups
}
elseif ASIAN_MAP
create_terrain PINE_FOREST
{
    base_terrain                GRASS2
    spacing_to_other_terrain_types 5
    land_percent                9
    number_of_clumps            10
    set_avoid_player_start_areas
    set_scale_by_groups
}
elseif TROPICAL_MAP
create_terrain JUNGLE
{
    base_terrain                GRASS
    spacing_to_other_terrain_types 5
    land_percent                9
    number_of_clumps            10
    set_avoid_player_start_areas
    set_scale_by_groups
}
else
create_terrain FOREST
{
    base_terrain                GRASS
    spacing_to_other_terrain_types 5
    land_percent                9
    number_of_clumps            10
    set_avoid_player_start_areas
    set_scale_by_groups
}
endif

```

```
/* The Primary Forest on Coastal covers 9% of the map and can be oak forest, pine forest or even rain forest, depending on the kind of map we want */
```

```
/* PRIMARY PATCH */
```

```
if DESERT_MAP
    create_terrain DESERT
{
    base_terrain            DIRT
    number_of_clumps        12
    spacing_to_other_terrain_types 0
    land_percent            8
    set_scale_by_size
}
elseif ALPINE_MAP
create_terrain GRASS3
{
    base_terrain            GRASS2
    number_of_clumps        8
    spacing_to_other_terrain_types 0
    land_percent            6
    set_scale_by_size
}
elseif FROZEN_MAP
create_terrain GRASS_SNOW
{
    base_terrain            SNOW
    number_of_clumps        8
    spacing_to_other_terrain_types 0
    land_percent            6
    set_scale_by_size
}
elseif ASIAN_MAP
create_terrain GRASS3
{
    base_terrain            GRASS2
    number_of_clumps        8
    spacing_to_other_terrain_types 0
    land_percent            6
    set_scale_by_size
}
else
create_terrain DIRT
{
    base_terrain            GRASS
    number_of_clumps        8
    spacing_to_other_terrain_types 0
    land_percent            9
    set_scale_by_size
}
endif
```

```
/* These patches and additional forests are used for variation. Adding grass or dirt on top of the base terrain is really only for cosmetic purposes. Forests affect how defensive the map can be and how hard it is to path from one town to another */
```

```

/* SECONDARY FOREST */

if DESERT_MAP
create_terrain FOREST
{
    base_terrain                GRASS
    spacing_to_other_terrain_types 3
    land_percent                1
    number_of_clumps            3
    set_avoid_player_start_areas
    set_scale_by_groups
}
elseif ALPINE_MAP
create_terrain FOREST
{
    base_terrain                GRASS2
    spacing_to_other_terrain_types 3
    land_percent                1
    number_of_clumps            3
    set_avoid_player_start_areas
    set_scale_by_groups
}
elseif FROZEN_MAP
create_terrain SNOW_FOREST
{
    base_terrain                GRASS_SNOW
    spacing_to_other_terrain_types 3
    land_percent                1
    number_of_clumps            3
    set_avoid_player_start_areas
    set_scale_by_groups
}
elseif ASIAN_MAP
create_terrain BAMBOO
{
    base_terrain                GRASS2
    spacing_to_other_terrain_types 3
    land_percent                1
    number_of_clumps            3
    set_avoid_player_start_areas
    set_scale_by_groups
}
else
create_terrain PALM_DESERT
{
    base_terrain                DIRT
    spacing_to_other_terrain_types 3
    land_percent                1
    number_of_clumps            3
    set_avoid_player_start_areas
    set_scale_by_groups
}
endif

/* SECONDARY PATCH */

```

```

if DESERT_MAP
    create_terrain DIRT3
{
    base_terrain                DIRT
    number_of_clumps            24
    spacing_to_other_terrain_types 0
    land_percent                2
    set_scale_by_size
}
elseif ALPINE_MAP
create_terrain DIRT3
{
    base_terrain                GRASS2
    number_of_clumps            24
    spacing_to_other_terrain_types 0
    land_percent                2
    set_scale_by_size
}
elseif FROZEN_MAP
create_terrain DIRT_SNOW
{
    base_terrain                SNOW
    number_of_clumps            24
    spacing_to_other_terrain_types 0
    land_percent                2
    set_scale_by_size
}
elseif ASIAN_MAP
create_terrain DIRT3
{
    base_terrain                GRASS2
    number_of_clumps            24
    spacing_to_other_terrain_types 0
    land_percent                2
    set_scale_by_size
}
else
create_terrain GRASS3
{
    base_terrain                GRASS
    number_of_clumps            24
    spacing_to_other_terrain_types 0
    land_percent                2
    set_scale_by_size
}
endif

/* TERTIARY PATCH */

if DESERT_MAP
    create_terrain GRASS3
{
    base_terrain                DIRT
    number_of_clumps            30
    spacing_to_other_terrain_types 0

```

```

    land_percent                2
    set_scale_by_size
}
elseif ALPINE_MAP
create_terrain GRASS
{
    base_terrain                GRASS2
    number_of_clumps            30
    spacing_to_other_terrain_types 0
    land_percent                2
    set_scale_by_size
}
elseif FROZEN_MAP
create_terrain GRASS2
{
    base_terrain                GRASS_SNOW
    number_of_clumps            30
    spacing_to_other_terrain_types 0
    land_percent                2
    set_scale_by_size
}
elseif ASIAN_MAP
create_terrain GRASS
{
    base_terrain                GRASS2
    number_of_clumps            30
    spacing_to_other_terrain_types 0
    land_percent                2
    set_scale_by_size
}
else
create_terrain DIRT3
{
    base_terrain                GRASS
    number_of_clumps            30
    spacing_to_other_terrain_types 0
    land_percent                2
    set_scale_by_size
}
endif

/* ***** */
<OBJECTS_GENERATION>

/* PLAYER START OBJECTS */

create_object TOWN_CENTER
{
    set_place_for_every_player
    group_placement_radius    18
    min_distance_to_players    0
    max_distance_to_players    0
}

create_object VILLAGER
{

```

```

    set_place_for_every_player
    min_distance_to_players      6
    max_distance_to_players      6
}

create_object SCOUT
{
    number_of_objects            1
    set_place_for_every_player
    min_distance_to_players      7
    max_distance_to_players      9
}

create_object RELIC
{
    number_of_objects            5
    min_distance_to_players      25
    temp_min_distance_group_placement 20
}

/* Each player starts with a Town Center, villagers and a scout. The Map Generator knows that Chinese
need 6 villagers and that Aztecs have an Eagle Warrior. There are also 5 Relics placed on most map
types. */

/* SPECIAL STUFF FOR REGICIDE */

if REGICIDE
create_object VILLAGER
{
    number_of_objects            7
    set_place_for_every_player
    min_distance_to_players      6
    max_distance_to_players      6
}

create_object KING
{
    set_place_for_every_player
    min_distance_to_players      6
    max_distance_to_players      6
}

create_object CASTLE
{
    set_place_for_every_player
    min_distance_to_players      10
    max_distance_to_players      10
}

endif

/* Regicide games require a Castle, King and more villagers. Other game types, such as Defend the
Wonder, are handled automatically. */

/* NEAR FORAGE */

```



```

create_object FORAGE
{
    number_of_objects          6
    group_placement_radius     3
    set_tight_grouping
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    10
    max_distance_to_players    12
    min_distance_group_placement 6
}

/* On most maps, everyone gets berry bushes near their town */

/* NEAR GOLD */

create_object GOLD
{
    number_of_objects          7
    group_placement_radius     3
    set_tight_grouping
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    12
    max_distance_to_players    16
    min_distance_group_placement 7
    max_distance_to_other_zones 7
}

/* MEDIUM GOLD */

create_object GOLD
{
    number_of_objects          4
    group_placement_radius     3
    set_tight_grouping
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    18
    max_distance_to_players    26
    min_distance_group_placement 7
    max_distance_to_other_zones 7
}

/* FAR GOLD */

create_object GOLD
{
    number_of_objects          4
    group_placement_radius     3
    set_tight_grouping
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    25
    max_distance_to_players    35
    min_distance_group_placement 7
}

```

```

    max_distance_to_other_zones 7
}

/* Everyone gets 3 sets of gold mines. Sometimes the far gold mine is really far (35 tiles) from the
player's Town Center. */

/* NEAR STONE */

create_object STONE
{
    number_of_objects          5
    group_placement_radius     2
    set_tight_grouping
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    14
    max_distance_to_players    18
    min_distance_group_placement 7
    max_distance_to_other_zones 7
}

/* FAR STONE */

create_object STONE
{
    number_of_objects          4
    group_placement_radius     2
    set_tight_grouping
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    20
    max_distance_to_players    26
    min_distance_group_placement 7
    max_distance_to_other_zones 7
}

/* We place stone last since it is more important to start with berries and gold. If the map runs out of
room it might not place stone. This is not ideal, but it is better than losing gold. */

/* BAA BAA */

if TROPICAL_MAP
create_object TURKEY
{
    number_of_objects 4
    set_loose_grouping
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    10
    max_distance_to_players    12
}
create_object TURKEY
{
    number_of_objects 2
    number_of_groups 2
    set_loose_grouping

```

```

    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    14
    max_distance_to_players    30
}
else
create_object SHEEP
{
    number_of_objects 4
    set_loose_grouping
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    10
    max_distance_to_players    12
}
create_object SHEEP
{
    number_of_objects 2
    number_of_groups 2
    set_loose_grouping
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    14
    max_distance_to_players    30
}
endif

```

/* Sheep and Turkeys are small, so they are easy to place. If you really want to confuse players, mix sheep and turkeys on a map. */

```

create_object DEER
{
    number_of_objects          4
    group_variance             1
    group_placement_radius     3
    set_loose_grouping
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    14
    max_distance_to_players    30
}

if TROPICAL_MAP
create_object JAVELINA
else
create_object BOAR
endif
{
    number_of_objects          1
    set_loose_grouping
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    16
    max_distance_to_players    22
}

```

```

if TROPICAL_MAP
create_object JAVELINA
else
create_object BOAR
endif
{
    number_of_objects          1
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    16
    max_distance_to_players    22
}

/* Stuff to hunt. */

if TROPICAL_MAP
create_object JAGUAR
{
    number_of_objects          2
    set_gaia_object_only
    set_place_for_every_player
    min_distance_group_placement 12
    min_distance_to_players    30
    max_distance_to_players    40
}
else
create_object WOLF
{
    number_of_objects          2
    set_gaia_object_only
    set_place_for_every_player
    min_distance_group_placement 12
    min_distance_to_players    30
    max_distance_to_players    40
}
endif

/* Bad guys. Leave them off if you want a rushing map. */

/* RESOURCES BY MAP */

if LARGE_MAP
create_object FORAGE
{
    number_of_groups          1
    number_of_objects          5
    group_placement_radius    3
    set_tight_grouping
    set_gaia_object_only
    min_distance_to_players    40
    max_distance_to_players    120
    min_distance_group_placement 7
}
endif

```

/* Because large maps have more open space, they need more resources. These resources do not “belong” to anyone. They are scattered for the taking. */

```
if HUGE_MAP
create_object FORAGE
{
    number_of_groups          2
    number_of_objects         5
    group_placement_radius    3
    set_tight_grouping
    set_gaia_object_only
    min_distance_to_players   40
    max_distance_to_players   120
    min_distance_group_placement 7
}
endif
```

```
if GIGANTIC_MAP
create_object FORAGE
{
    number_of_groups          3
    number_of_objects         5
    group_placement_radius    3
    set_tight_grouping
    set_gaia_object_only
    min_distance_to_players   40
    max_distance_to_players   120
    min_distance_group_placement 7
}
endif
```

/* EXTRA GOLD */

```
if TINY_MAP
create_object GOLD
{
    number_of_groups          2
    number_of_objects         3
    group_placement_radius    2
    set_tight_grouping
    set_gaia_object_only
    min_distance_to_players   40
    min_distance_group_placement 9
}
endif
```

```
if SMALL_MAP
create_object GOLD
{
    number_of_groups          2
    number_of_objects         3
    group_placement_radius    2
    set_tight_grouping
    set_gaia_object_only
    min_distance_to_players   40
    min_distance_group_placement 9
}
```

```

}
endif

if MEDIUM_MAP
create_object GOLD
{
    number_of_groups          3
    number_of_objects         3
    group_variance            1
    group_placement_radius    2
    set_tight_grouping
    set_gaia_object_only
    min_distance_to_players   40
    min_distance_group_placement 9
}
endif

if LARGE_MAP
create_object GOLD
{
    number_of_groups          3
    number_of_objects         3
    group_variance            1
    group_placement_radius    2
    set_tight_grouping
    set_gaia_object_only
    min_distance_to_players   40
    min_distance_group_placement 9
}
endif

if HUGE_MAP
create_object GOLD
{
    number_of_groups          4
    number_of_objects         4
    group_variance            1
    group_placement_radius    2
    set_tight_grouping
    set_gaia_object_only
    min_distance_to_players   40
    min_distance_group_placement 9
}
endif

if GIGANTIC_MAP
create_object GOLD
{
    number_of_groups          5
    number_of_objects         4
    group_variance            1
    group_placement_radius    2
    set_tight_grouping
    set_gaia_object_only
    min_distance_to_players   40
    min_distance_group_placement 9
}

```

```

}
endif

/* EXTRA STONE */

if TINY_MAP
create_object STONE
{
    number_of_groups          1
    number_of_objects        4
    group_variance           1
    group_placement_radius   2
    set_tight_grouping
    set_gaia_object_only
    min_distance_to_players  40
    min_distance_group_placement 9
}
endif

if SMALL_MAP
create_object STONE
{
    number_of_groups          1
    number_of_objects        4
    group_variance           1
    group_placement_radius   2
    set_tight_grouping
    set_gaia_object_only
    min_distance_to_players  40
    min_distance_group_placement 9
}
endif

if MEDIUM_MAP
create_object STONE
{
    number_of_groups          2
    number_of_objects        4
    group_variance           1
    group_placement_radius   2
    set_tight_grouping
    set_gaia_object_only
    min_distance_to_players  40
    min_distance_group_placement 9
}
endif

if LARGE_MAP
create_object STONE
{
    number_of_groups          3
    number_of_objects        4
    group_variance           1
    group_placement_radius   2
    set_tight_grouping
    set_gaia_object_only

```

```

    min_distance_to_players      40
    min_distance_group_placement  9
}
endif

if HUGE_MAP
create_object STONE
{
    number_of_groups            4
    number_of_objects           4
    group_variance              1
    group_placement_radius      2
    set_tight_grouping
    set_gaia_object_only
    min_distance_to_players      40
    min_distance_group_placement  9
}
endif

if GIGANTIC_MAP
create_object STONE
{
    number_of_groups            5
    number_of_objects           4
    group_variance              1
    group_placement_radius      2
    set_tight_grouping
    set_gaia_object_only
    min_distance_to_players      40
    min_distance_group_placement  9
}
endif

if DESERT_MAP
create_object PALMTREE
{
    number_of_objects           2
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players      4
    max_distance_to_players      5
    min_distance_group_placement  2
}
create_object PALMTREE
{
    number_of_objects           3
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players      6
    max_distance_to_players      8
    min_distance_group_placement  2
}
elseif ALPINE_MAP
create_object PINETREE
{
    number_of_objects           2

```



```

    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players      4
    max_distance_to_players      5
    min_distance_group_placement  2
}
create_object PINETREE
{
    number_of_objects            3
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players      6
    max_distance_to_players      8
    min_distance_group_placement  2
}
elseif ASIAN_MAP
create_object BAMBOO_TREE
{
    number_of_objects            2
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players      4
    max_distance_to_players      5
    min_distance_group_placement  2
}
create_object PINETREE
{
    number_of_objects            3
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players      6
    max_distance_to_players      8
    min_distance_group_placement  2
}
elseif TROPICAL_MAP
create_object PALMTREE
{
    number_of_objects            2
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players      4
    max_distance_to_players      5
    min_distance_group_placement  2
}
create_object OAKTREE
{
    number_of_objects            3
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players      6
    max_distance_to_players      8
    min_distance_group_placement  2
}
elseif FROZEN_MAP
create_object SNOWPINETREE
{

```

```

    number_of_objects          2
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    4
    max_distance_to_players    5
    min_distance_group_placement 2
}
create_object SNOWPINETREE
{
    number_of_objects          3
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    6
    max_distance_to_players    8
    min_distance_group_placement 2
}
else
create_object OAKTREE
{
    number_of_objects          2
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    4
    max_distance_to_players    5
    min_distance_group_placement 2
}
create_object OAKTREE
{
    number_of_objects          3
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players    6
    max_distance_to_players    8
    min_distance_group_placement 2
}
endif

/* These are the "straggler" trees near every Town Center. */

if TROPICAL_MAP
create_object MACAW
{
    number_of_objects          6
    set_scaling_to_map_size
}
else
create_object HAWK
{
    number_of_objects          6
    set_scaling_to_map_size
}
endif

if TROPICAL_MAP
create_object JAGUAR
{

```

```

    number_of_groups          3
    set_loose_grouping
    start_random
    percent_chance 75
        number_of_objects 1
    percent_chance 25
        number_of_objects 2
    end_random
    set_scaling_to_map_size
    set_gaia_object_only
    min_distance_group_placement 12
    min_distance_to_players 40
    max_distance_to_players 120
}
else
create_object WOLF
{
    number_of_groups          3
    set_loose_grouping
    start_random
    percent_chance 75
        number_of_objects 1
    percent_chance 25
        number_of_objects 2
    end_random
    set_scaling_to_map_size
    set_gaia_object_only
    min_distance_group_placement 12
    min_distance_to_players 40
    max_distance_to_players 120
}
endif

/* More wildlife. Birds have no affect on gameplay, but make the map more interesting. */

create_object FORAGE
{
    number_of_groups          1
    number_of_objects          5
    group_placement_radius    3
    set_tight_grouping
    set_gaia_object_only
    min_distance_to_players    19
    max_distance_to_players    120
    min_distance_group_placement 9
}

create_object DEER
{
    number_of_objects 4
    group_variance 1
    set_loose_grouping
    set_gaia_object_only
    set_place_for_every_player
    min_distance_to_players 19
}

```

```

if DESERT_MAP
create_object PALMTREE
{
    number_of_objects          30
    set_gaia_object_only
    set_scaling_to_map_size
    min_distance_to_players    8
}
elseif ALPINE_MAP
create_object PINETREE
{
    number_of_objects          30
    set_gaia_object_only
    set_scaling_to_map_size
    min_distance_to_players    8
}
elseif FROZEN_MAP
create_object SNOWPINETREE
{
    number_of_objects          30
    set_gaia_object_only
    set_scaling_to_map_size
    min_distance_to_players    8
}
elseif ASIAN_MAP
create_object BAMBOO_TREE
{
    number_of_objects          30
    set_gaia_object_only
    set_scaling_to_map_size
    min_distance_to_players    8
}
elseif TROPICAL_MAP
create_object PALMTREE
{
    number_of_objects          30
    set_gaia_object_only
    set_scaling_to_map_size
    min_distance_to_players    8
}
else
create_object OAKTREE
{
    number_of_objects          30
    set_gaia_object_only
    set_scaling_to_map_size
    min_distance_to_players    8
}
endif

```

/* These are the lone trees scattered across the map. It makes the map look more believable than if there were just forests and no straggler trees. */

```

if DESERT_MAP
create_object DORADO

```

```

{
    number_of_objects          15
    set_scaling_to_map_size
    set_gaia_object_only
    max_distance_to_other_zones 4
}
create_object SNAPPER
{
    number_of_objects          10
    set_scaling_to_map_size
    set_gaia_object_only
    max_distance_to_other_zones 4
}
elseif ALPINE_MAP
create_object SALMON
{
    number_of_objects          15
    set_scaling_to_map_size
    set_gaia_object_only
    max_distance_to_other_zones 4
}
create_object SNAPPER
{
    number_of_objects          10
    set_scaling_to_map_size
    set_gaia_object_only
    max_distance_to_other_zones 4
}
elseif FROZEN_MAP
create_object SALMON
{
    number_of_objects          15
    set_scaling_to_map_size
    set_gaia_object_only
    max_distance_to_other_zones 4
}
create_object SNAPPER
{
    number_of_objects          10
    set_scaling_to_map_size
    set_gaia_object_only
    max_distance_to_other_zones 4
}
elseif ASIAN_MAP
create_object TUNA
{
    number_of_objects          15
    set_scaling_to_map_size
    set_gaia_object_only
    max_distance_to_other_zones 4
}
create_object SNAPPER
{
    number_of_objects          10
    set_scaling_to_map_size
    set_gaia_object_only

```

```

    max_distance_to_other_zones      4
}
else
create_object DORADO
{
    number_of_objects                5
    set_scaling_to_map_size
    set_gaia_object_only
    max_distance_to_other_zones      4
}
create_object TUNA
{
    number_of_objects                10
    set_scaling_to_map_size
    set_gaia_object_only
    max_distance_to_other_zones      4
}
create_object SNAPPER
{
    number_of_objects                10
    set_scaling_to_map_size
    set_gaia_object_only
    max_distance_to_other_zones      4
}
endif

create_object SHORE_FISH
{
    number_of_objects                25
    set_scaling_to_map_size
    min_distance_group_placement     3
    set_gaia_object_only
}

create_object MARLIN1
{
    number_of_groups                  5
    number_of_objects                 1
    set_scaling_to_map_size
    set_gaia_object_only
    min_distance_group_placement     10
    max_distance_to_other_zones      7
}

create_object MARLIN2
{
    number_of_groups                  5
    number_of_objects                 1
    set_scaling_to_map_size
    set_gaia_object_only
    min_distance_group_placement     10
    max_distance_to_other_zones      7
}

```

/* One of our designers is picky about which fish live in which part of the country. Only the Marlin and the Shore fish have different amounts of food in them. The other fish are just for variety. */

```
/* ***** */
```

```
<ELEVATION_GENERATION>
```

```
create_elevation      7
{
  if DESERT_MAP
    base_terrain      DIRT
  elseif ALPINE_MAP
    base_terrain      GRASS2
  elseif FROZEN_MAP
    base_terrain      SNOW
  elseif ASIAN_MAP
    base_terrain      GRASS2
  else
    base_terrain      GRASS
  endif
  number_of_clumps    14
  number_of_tiles     2000
  set_scale_by_groups
  set_scale_by_size
}
```

/* Notice that the elevation must be placed on a certain terrain, whichever is the base terrain for the map. If you use lots of different terrain, you may need to have multiple elevation statements. */

```
<CLIFF_GENERATION>
```

```
min_number_of_cliffs 5
max_number_of_cliffs 8
min_length_of_cliff  4
max_length_of_cliff  10
cliff_curliness      10
min_distance_cliffs  3
```

/* Cliffs work okay on land maps but are best left off island maps. */

```
/* ***** */
```

```
<CONNECTION_GENERATION>
```

```
create_connect_all_players_land
{
  if FROZEN_MAP
    replace_terrain WATER      ICE
    replace_terrain MED_WATER  ICE
    replace_terrain DEEP_WATER ICE
  else
    replace_terrain WATER      SHALLOW
    replace_terrain MED_WATER  SHALLOW
    replace_terrain DEEP_WATER SHALLOW
  endif
```

/* Every player on a team must be connected in Coastal. This part puts shallows or ice across water to make sure no one is on an island. On an Archipelago map, it is okay if some players are on their own islands. */

```

terrain_cost    WATER          7
terrain_cost    MED_WATER      9
terrain_cost    DEEP_WATER     15
terrain_cost    FOREST         7
terrain_cost    PINE_FOREST    7
terrain_cost    PALM_DESERT    7
terrain_cost    SNOW_FOREST    7
terrain_cost    JUNGLE         7
terrain_cost    SHALLOW       3
terrain_cost    DESERT         2
terrain_cost    DIRT          2
terrain_cost    DIRT2         2
terrain_cost    DIRT3         2
terrain_cost    DIRT_SNOW     2
terrain_cost    GRASS         2
terrain_cost    GRASS2        2
terrain_cost    GRASS3        2
terrain_cost    GRASS_SNOW    2
terrain_cost    SNOW         2
terrain_cost    BEACH         7
terrain_size    WATER          2      1
terrain_size    MED_WATER      2      1
terrain_size    DEEP_WATER     2      1
terrain_size    GRASS         0      0
terrain_size    GRASS2        0      0
terrain_size    GRASS3        0      0
terrain_size    GRASS_SNOW    0      0
terrain_size    SNOW         0      0
terrain_size    DIRT          0      0
terrain_size    DIRT2         0      0
terrain_size    DIRT3         0      0
terrain_size    DIRT_SNOW     0      0
terrain_size    SNOW_FOREST    0      0
terrain_size    FOREST         0      0
terrain_size    PINE_FOREST    0      0
terrain_size    PALM_DESERT    0      0
terrain_size    JUNGLE         0      0
terrain_size    DESERT         0      0
}

```

/* Terrain cost is used to make the connections try and go over land and not through forests which looks fake, unless you pave the connections with road, as in Black Forest. */