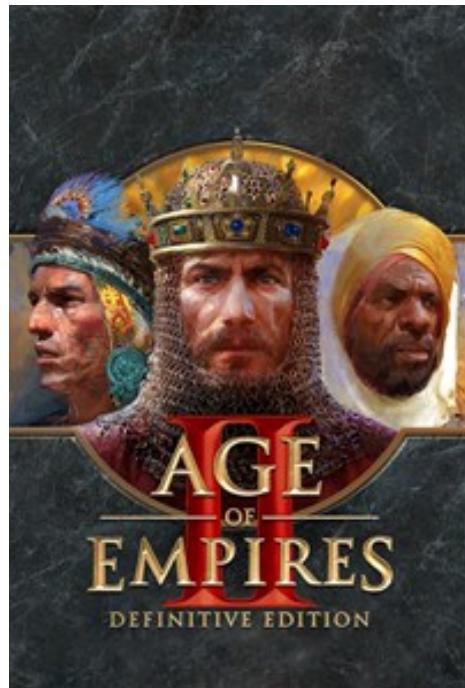


Age of Empires DE, a tutorial for modding

Koldar

Version 1.0.0



Date: December 30, 2020

Contents

List of Figures	ii
List of Tables	iii
Acronyms	iv
About this report	v
1 Introduction	1
2 Background and Related Works	2
3 Basics and Definitions	4
4 Creating a basic mod (MWE)	7
5 Implementing other mods using Advance Genie Editor	10
5.1 Altering corpses decaying	10
6 Creating Random Maps using Random Map Scripting (RMS)	11
6.0.1 Filling PLAYER_SETUP	11
Appendices	12
Semantics of Age of Empires: DE Exe	13
.1 BattleServer	13
.2 certificates	13
.3 Docs	13
.4 Schema	13
.5 Support	13
.6 Tools_Builds	13
.7 webclient	13
.8 wwise	14
.9 resources	14
.9.1 __common	14
.10 widgetui	14

.10.1	atlas	15
.10.2	backgrounds	15
.10.3	campaign	16
.10.4	ingame	16
.10.5	menu	16
Advance Genie Editor		17
.11	Required Knowledge	18
.12	Units parameters	18
.12.1	Language Files	19
.12.2	Graphics	19
.12.3	Statistics	19
.12.4	Projectiles	21
.12.5	Attributes	21
.12.6	Sounds	21
.12.7	Tasks	21
.13	Graphics	21
.13.1	Deltas	22
.13.2	Angle Sounds	22
RMS file format and layout		23
.14	RMS control flow	24
.14.1	Conditional flow	24
.14.2	#include_drs	24
.14.3	#const	24
.15	RMS file sections	25
.15.1	PLAYER_SETUP	25
.15.2	LAND_GENERATION	26
.16	Functions	30
.16.1	Random	30
.16.2	Choose a randomly chosen scenario	30
.17	Popular included files	30
.17.1	thebr_setup.inc	30
.17.2	F_seasons.inc	31
.18	Constants	31
.18.1	Terrain Type	31
.18.2	Map Type	36
Terms		38
Bibliography		39

List of Figures

2.1	Examples of maps available in Age of Empires: DE. Red, Yellow, Green and Blue diamond represents players positions. If a team game is played, Red and blue are allied against team yellow and green. A shoe represents a nomadic map.	3
4.1	Image to put as background	7
4.2	My mods window	9
2	Background menu examples	15
3	Standard Advance Genie Editor window	18
4	Standard Advance Genie Editor window. The red box highlights the category tabs grouping the parameters you can tweak according to the Age of Empires: DE field.	19
5	Standard Advance Genie Editor window. The red box highlights the search boxes you can use to filter	20
6	Example of how the bottom bar can be used to check how many changes we have made so far.	20
7	Example of how players are laid out in a map is automatic placement is used.	29

List of Tables

3.1	Examples of important paths used in Age of Empires: DE	4
1	Semantic of each Garrison Type value.	22
2	Available seasons in Age of Empires: DE.	32
3	List of all the default terrain types available	34
4	List of all the map types available	37

Acronyms

AI Artificial Intelligence. 24

DAT DATA. 15

DDS DirectDraw Surface. 6, 12, 13

DLL Dynamic Link Library. 15

EBNF Extended Backus–Naur Form. 21

PC Personal Computer. v

PNG Portable Network Graphics. 13

RMS Random Map Scripting. 9, 12, 21–29

UI User Interface. 12, 13

URL Uniform Resource Locator. 7

About this report

When writing, this guide, the following convention are adopted.

Generally speaking, definitions of concepts are shown as:

Definition 0.0.1. Square root of a number The square root of a number x (\sqrt{x}) is defined as the number that, multiplied with itself, yield x .

A note is something that add context to a topic and is shown as below:



Default methods in C# interfaces have been heavily inspired by java's default method implementation

A warning is something that should be aware of.



You shouldn't read a variable value before setting it.

An attention contains information that, if not followed, will cause unexpected results;



In C, don't read a variable value before setting it.

Reference to the glossary are printed as below:

Personal Computer are used throughout the world.

Whole reference to the acronym table are shown as:

Personal Computer (PC) are used throughout the world.

Citation are shown as follows: in A* algorithm, we use $f = g + h$ to estimate search states [1].

Chapter 1

Introduction

This guide helps you creating a new mod in Age of Empires: DE: by reading this file, you should be able to create your own mod. I have written this guide when I started my Age of Empires: DE modding experience and contain all the information I have gathered while scouting the internet. The Information presented here may be lacking or missing.

Note that this guide focuses its attention on Age of Empires: DE, while other version of the game (e.g., Age of Conqueror or HD version) are not considered whatsoever: this usually means that some functions or characteristics may not be available in such game versions.

All code (alongside the source code of this document) is available at <https://github.com/Koldar/aoe-mod-tutorial>: please open an issue whenever you find any typos, missing references of false information.

Chapter 2

Background and Related Works

When a map is **nomad** the players will start with no towncenter, and some villagers. Each player is required to create a town center wherever she desires. When a map is **michi** the players will be separated by forest.

Figure 2.1 shows some maps available to play in Age of Empires: DE.

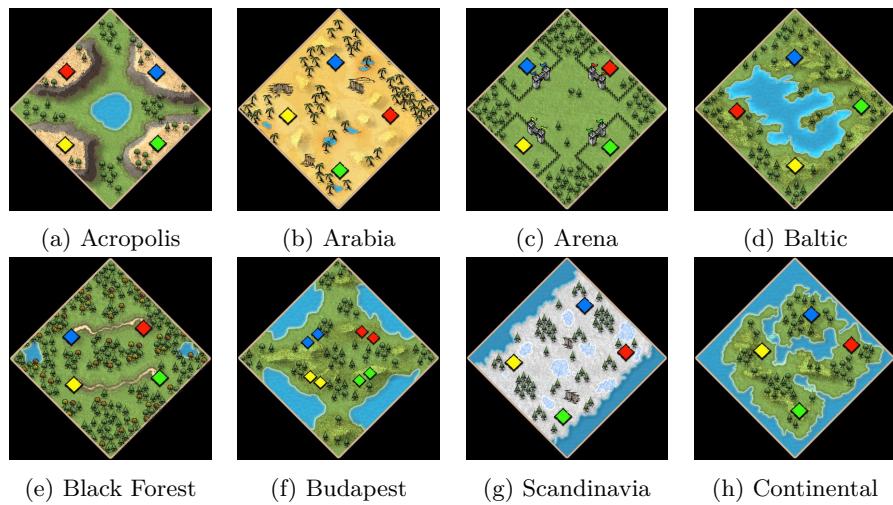


Figure 2.1: Examples of maps available in Age of Empires: DE. Red, Yellow, Green and Blue diamond represents players positions. If a team game is played, Red and blue are allied against team yellow and green. A shoe represents a nomadic map.

Chapter 3

Basics and Definitions

The first thing is to install Age of Empires: DE, for instance, via Steam [2]. After that, you need to identify Advance Genie Editor program: it is usually installed at `C:\Program Files (x86)\Steam\steamapps\common\AoE2DE\Tools_Builds\AdvancedGenieEditor3.exe`. Data mods are discouraged [3], hence you should create mods by replacing individual files and avoiding data modding whenever possible. We will refer to the folder `C:\Program Files (x86)\Steam\steamapps\common\AoE2DE\` as `Age of Empires: DE Exe`.

The mods you install from Age of Empires: DE are available in a directory that usually is `%HOMEPATH%\Games\Age of Empires 2 DE` (from here on dubbed `Age of Empires: DE Home`). Within `Age of Empires: DE Home`, there should be a folder whose name is the user `steam id` (which is a big number) (from here on such a folder is dubbed as `Age of Empires: DE Weird Number`) [4]. Within it, there is a folder called `mods\subscribed` containing all the mods the user has subscribed to. Along side “subscribed” folder, there is a folder named “local” where local mods in development may be put (such a folder will be referred to as `Age of Empires: DE Local Mod`). The structure of `Age of Empires: DE Home` is shown in Listing 3.1.

For example, Table 3.1 shows all the involved paths for the modding.

Path	Example
<code>Age of Empires: DE Exe</code>	<code>C:\Program Files (x86)\Steam\steamapps\common\AoE2DE\</code>
<code>Age of Empires: DE Home</code>	<code>C:\Users\FooBar\Games\Age of Empires 2 DE</code>
<code>Age of Empires: DE Weird Number</code>	<code>C:\Users\FooBar\Games\Age of Empires 2 DE\982357864790234</code>

Table 3.1: Examples of important paths used in Age of Empires: DE

In the “mods” folder, there is a file called `mods-status.json` (an example of the content is shown in Listing 3.2).

```
1  [
2    {
```

```
Age of Empires 2 DE
|--- logs
|   |--- ...
|--- testharness
|--- 0
|--- 893465689237890
|   |--- mods
|   |   |--- local
|   |   |--- subscribed
|   |       |--- one folder per mod installed
|   |--- mod-status.json
```

Listing 3.1: Age of Empires home folder directory structure

```
3         "CheckSum": "3716621926",
4         "Enabled": true,
5         "LastUpdate": "1581833077032",
6         "Path": "subscribed//2695_Improved Small Trees",
7         "Priority": 1,
8         "PublishID": 0,
9         "Title": "2695_Improved Small Trees",
10        "WorkshopID": 2695
11    },
12    {
13        "CheckSum": "1216235570",
14        "Enabled": false,
15        "LastUpdate": "1575122745034",
16        "Path": "subscribed//1592_Better Main Menu Night Time",
17        "Priority": 2,
18        "PublishID": 0,
19        "Title": "Better Main Menu Night Time",
20        "WorkshopID": 1592
21    }
22 ]
```

Listing 3.2: Example of mods-status file

The json stored is a sequence of objects, each representing:

- Checksum: MD5 of the mod?
- **Enable**: if `true`, the mod is enabled, `false` otherwise;
- Last Update: timestamp representing the number of milliseconds when the mod has been lastly updated;
- **Path**: path, relative to `Age of Empires: DE Weird Number` directory, where the specific mod is installed;
- **Priority**: priority used to load the mod;
- Publish Id: always 0?

- **Title:** name of the mod to show to the user;
- WorkShop Id: An id that uniquely identifies this mod on www.ageofempires.com site;

At high level Age of Empires: DE install each mod by following Algorithm

1. After sorting out the enabled mods, the software “patches” the set of files specified by the mod path. From the algorithm, it can be seen that the priority of each mod determine the order each mod is used. Priority may change the behavior is a mod x relies on the installation of the mod y (if $y.priority < x.priority$) [2].

Algorithm 1: Age of Empires: DE mod boot strap algorithm

```

1 mods  $\leftarrow$  Gather mods in Age of Empires: DE Weird Number;
2 sortedmods  $\leftarrow$  Sort mods by prioritizing mods with small priority;
3 foreach m  $\in$  reversed(sortedmods) do mods subscribed
4   | if  $\neg m.Enable$  then
5   |   | continue;
6   | end
7   | Copy the files in Age of Empires: DE Weird
      |   Number\mods\m.Path into Age of Empires: DE Exe;
8 end
9 Start Age of Empires: DE;

```

Chapter 4

Creating a basic mod (MWE)

When developing a mod you should work in the directory `Age of Empires: DE Local Mod`. Start by creating a folder with the same name as the mod you want to create. For this tutorial, we will create a mod that changes the background of the main menu: we will name such a folder “carcassone-menu”.

For this very reason we need to put the background image (in this case the one shown in Figure 4.1)



Figure 4.1: Image to put as background

```

carcassonne-menu
|--- widgetui
    |--- textures
        |--- backgrounds
            |--- mainmenu_bg.dds
|--- info.json
|--- thumbnail.jpg

```

Listing 4.1: Carcassonne mod layout

```

1 carcassonne-menu
2     |-- widgetui
3         |-- textures
4             |-- backgrounds
5                 |-- mainmenu_bg.dds
6     |-- info.json
7     |-- thumbnail.jpg

```

Listing 4.2: Carcassonne mod layout

As shown in the Appendix 6.0.1, the background image is specified (relative to `Age of Empires: DE Exe`) in `widgetui/textures/backgrounds/mainmenu_bg.dds`. Hence, in the “`carcassonne-menu`” folder, you need to create the subfolder `widgetui/textures/backgrounds/mainmenu_bg`, then you need to put the Carcassone image, name it “`mainmenu_bg`” (ensure that the image follows the DirectDraw Surface (DDS) extension).

Aside the folders you have just created, you need two additional files that represents mod metadata. Both files needs to be put in “`carcassonne-menu`” folders. The first is `thumbnail.jpg`, which is a jpg format that is shown when browsing the mods. The other metadata file is call “`info.json`” (an example is shown in Listing 4.2): it is a json containing 3 values:

- Author: name of the author of this mod;
- Description: a description of the mod;
- Title: title to show of the mod;

If the developer puts other fields in this json, they will be automatically removed. Any formatting will be overwritten as well. All the metadata is shown in the right panel of the mod browsing window (in `Age of Empires: DE` program, Mods section). Additional files in `Age of Empires: DE Local Mod` mod will be left in the folder

The thumbnail can have any dimensions, although 400x150-ish dimensions may be preferred. After this operation, open `Age of Empires: DE`, go to the mods, specifically to “My Mods”. If you click “Import My Mods” `Age of Empires: DE` will search into the `Age of Empires: DE Local Mod` for compliant mods. You should see the new mods (as shown in Figure 4.2).



Figure 4.2: My mods window

The image in the background will be clipped. “mod-status.json” will be automatically updated.

⚠ If `thumbnail.jpg` is not present, a default image will be put instead. If `info.json` is absent, default values will be put: at the moment of writing, the author is set to “Unpublished”, description to “No Description” and name as the same folder as the mod root dir. If “`info.json`” content is lacking one of those 3 fields, Age of Empires: DE will automatically fill the missing ones.

In the mod window, if you select the “carcassonne-menu” you click “More Info”. If you click this, Age of Empires: DE will automatically open the browser at the Uniform Resource Locator (URL) <https://www.ageofempires.com/mods/details/XYZ> where *XYZ* is the WorkShop Id of the mod. If Age of Empires: DE fails to load the mods, you can *debugging* the mod by make changes the mod in Age of Empires: DE Local Mod and click “Import Local Mods” button.

Chapter 5

Implementing other mods using Advance Genie Editor

In this chapter we describe how to implement some more complex mods.

5.1 Altering corpses decaying

We now try to create a mod that simply changes the time when the corpse decays. Let us call it **ya-decaying-corpses**.¹ In this mod, we need to change data files. Data files are located in the `Age of Empires: DE Exe\resources_common`. As show in Appendix .10.5, you can use Advance Genie Editor to do so: Copy the file `Age of Empires: DE Exe\resources_common\dat\empires2_x2_p1.dat` in the mod dir; then open it via Advance Genie Editor. Switch to the *Units* tabs and select all the units of all type (except maybe Gaia's ones). For every one of them, go to the “Statisitcs” section and change the value *Resource Decay* to *-1*.

¹“ya” stands for “Yet Another” and is a popular way of naming in computer engineering whenever the naming creativity is low.

Chapter 6

Creating Random Maps using RMS

In this chapter we will explain how to use RMS language to generate random maps. For information about RMS, see Appendix .13.2 or look at [5].

6.0.1 Filling `PLAYER_SETUP`

Appendices

Semantics of Age of Empires: DE Exe

.1 BattleServer

Folder with `BattleServer.exe` file. Usually not useful when modding.

.2 certificates

X509 file format of Age of Empires: DE. Usually not useful when modding.

.3 Docs

Manual of PDF of Age of Empires: DE. Not useful when modding.

.4 Schema

Not useful when modding.

.5 Support

Contains some link to Age of Empires: DE website support. Not useful when modding.

.6 Tools_Builds

Represents Advance Genie Editor directory.

.7 webclient

Not useful when modding.

.8 wwise

Not useful when modding.

.9 resources

Specifies all the data that are not widgets of the User Interface (UI). The folder contains data like the cursors icons, the Advance Genie Editor dat files.

.9.1 __common

This is the main folder of **resources** directory.

cursors

List of all the cursors the cursor directed by the user mouse will display when a given action is performed. For instance, when you need to garrison a villager, a specific cursor image replace the classic arrow icon. All the cursors image have the **cur** extension.

dat

Contains the dat files containing all the information regarding units, civilization, buildings: **empires2_x2_p1.dat** contains all such information.¹

drs

gamedata_x2 This important folder contains RMS files that can be used in your RMS programming (e.g., **F_Season.inc**). See Appendix .13.2.

.10 widgetui

Specifies which set of textures Age of Empires: DE needs to display alongside their configuration. If you need to alter a picture, it will probably be here. Contains a set of json. Furthermore, it contains a folder named “textures”, representing all the textures and images in the game. There are versions, *textures* and *textures-sd*: the former contains the texture to use and the latter contains the low resolution version of the same textures. The first texture are mandatory while the second are optional (modwise). All textures are saved via DDS extensions.²

Textures folder contains other folders:

¹The file can be opened by Advance Genie Editor

²You can use **Irfan View** 32-bit version to view DDS images.

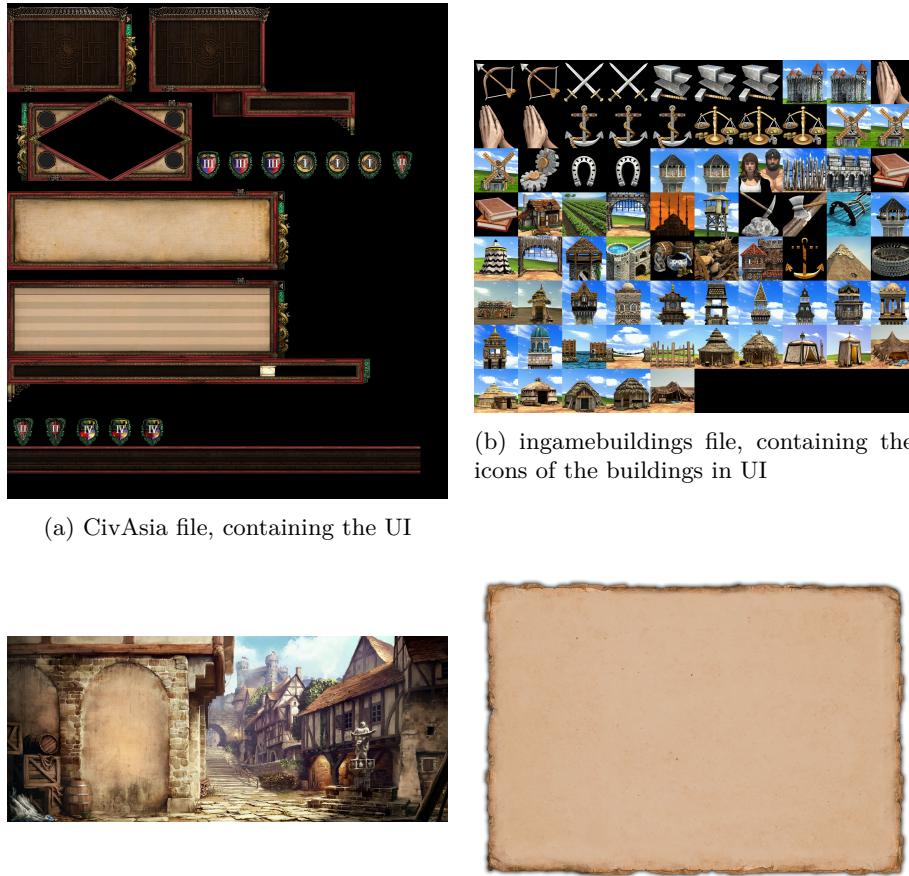


Figure 2: Background menu examples

.10.1 atlas

Contains textures that are present in the Campaigns menu (e.g., the scenes where you can choose which campaign to play), Sun Tzu icons in the “Art of War”, images representing the UI you see while playing (Figure 1a), the window chat, buildings icons (Figure 1b).

.10.2 backgrounds

Contains the menu background (“mainmenu_bg” and “mainmenu_bg_1”) images as well the windows in the Age of Empires: DE menus (e.g., lobby creation, history). Examples of these figures are shown in Figure ???. Note that some of images here are represented via Portable Network Graphics (PNG) rather than DDS.

.10.3 campaign

Texture regarding the campaigns, one sub directory per campaign. For instance, “cam3” is the *Saladin Campaign*. Within it:

- $X.dss$ file, where X is the campaign name, is the campaign image menu file (i.e., the image where you can choose the specific mission);
- $X_background$ is the image where the mission intro and outro are presented;
- a subfolder, one per mission in the given campaign. Each sub folder name has a number (starting from 1) as name and contains the drawings in the intro and outros of the associated mission.

.10.4 ingame

TODO

.10.5 menu

TODO

Advance Genie Editor

This is a small guide to Advance Genie Editor software. The guide is based on version 2020.3.30. You should not consider this guide as something as official. Advance Genie Editor is a program for editing data of genie (DATa (DAT) and Dynamic Link Library (DLL)) files. It can edit properties of units, civilizations, technologies, graphics, terrains, sounds, player colors and some other things [6]. Advance Genie Editor program can be found in `Age of Empires: DE Exe`, under “Tools_Build” directory.

When you execute it three windows automatically open. The *Open files...* allows you to open a Advance Genie Editor file. To manage Age of Empires: DE files, click on the button *Age of Empires II: Definitive Edition* on the right of *Defaults:* label. Set the *Genie version:* to *Age of Empires II: Definitive Edition*. The file that you need to open is the one specified by *Compressed data set (*.dat)*: such a file should be called `empires2_x2_p1.dat`. When you open such a file you can view all the unit, buildings, civilizations parameters and properties. The vanilla file is `C:\Program Files (x86)\Steam\steamapps\common\AoE2DE\resources_common\dat\empires2_x2_p1.dat`.

After opening the file, you can now make changes on the file. Figure 3 shows the state fo the program when it opens the file.

As shown in Figure 4, Advance Genie Editor splits its parameters regarding the Age of Empires: DE field.

 You can switch between tabs by the performing hotkeys `Ctrl + Pag. Up` and `Ctrl + Pag. Down`.

In the units left pane (called “Units”) there is displayed the list of all the units a particular civilization can manage (e.g., in Figure 5 the civ is “Gaia”). Since the unit tabs show several parameters and unit, you can filter the units using the search box in the units pane (e.g., Figure 5 highlighted in the red box).

The search box works as follows: first you select the civilization, then you can add at most 2 criteria involving the unit name (in and with the civilization): namely, if the 2 criteria are γ_1 and γ_2 and the civilization involved is c , the whole search criterion is $civ = c \wedge (\gamma_1 \in unit.name) \wedge (\gamma_2 \in unit.name)$, where `unit.name` is the string representing a generic unit name. For instance, if $\gamma_1 = dead$ then we select all the units whose name contains the substring “dead”.

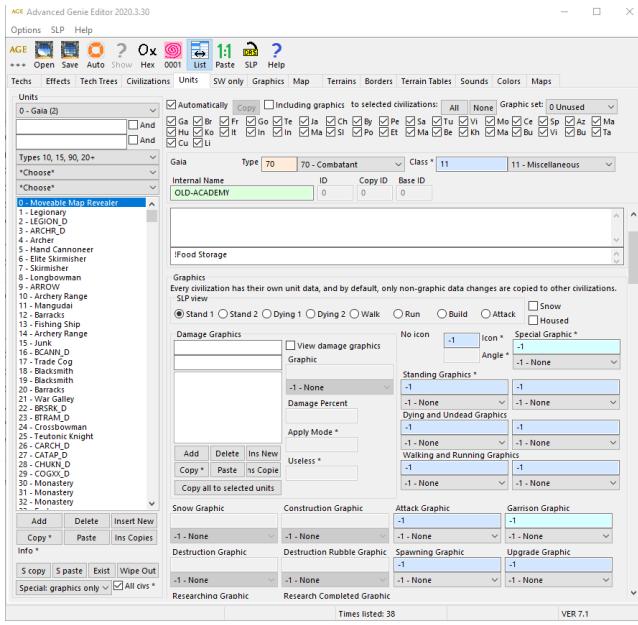


Figure 3: Standard Advance Genie Editor window

You can add “|” to perform a logical or (‘ \vee ’) inside γ_i ($i \in \{1, 2\}$). For instance if $\gamma_1 = \text{_D}|\text{dead}$, the global search criterion is $civ = c \wedge ((\text{_D} \in \text{unit.name}) \vee (\text{dead} \in \text{unit.name}))$: such a search query will generate all the units whose name contains either “_D” or “dead”.

You can select multiple units: when you alter a value from the unit pane, the same parameter will be updated also in every other civilization checked in the top checkboxes (e.g., “Ga”, “Br”, “Fr”, “Go”). At the bottom you can see how many changes performed in this session (as shown in Figure 6).

11 Required Knowledge

Unit names. The units ending with “_D” represent the corpses of the associated unit: for instance “Archer” unit is the actual unit with all its properties while “ARCHR_D” represents the archer corpse.

Buildings. Every time the user access to a new era, each building changes. For instance, in *graphics* you can see that there are, for each stable building civ, a stable for feudal, one for the castle age and another one for the imperial age.

12 Units parameters

Manage a single unit parameters. This section is divided in multiple sub areas:

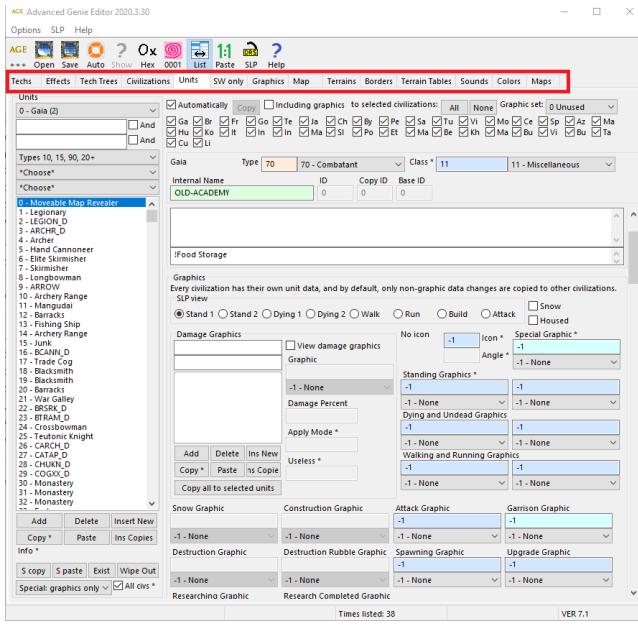


Figure 4: Standard Advance Genie Editor window. The red box highlights the category tabs grouping the parameters you can tweak according to the Age of Empires: DE field.

- Language Files;
- Graphics;
- Statistics;
- Projectiles;
- Attributes;
- Sounds;
- Tasks;

The documentation in this area has been copied from [7].

.12.1 Language Files

.12.2 Graphics

.12.3 Statistics

Hit Points: hit points of the unit. If -1, the unit will immediately die [8].
Speed: the speed of the unit. **Rotation Speed** makes the unit slower. **Line**

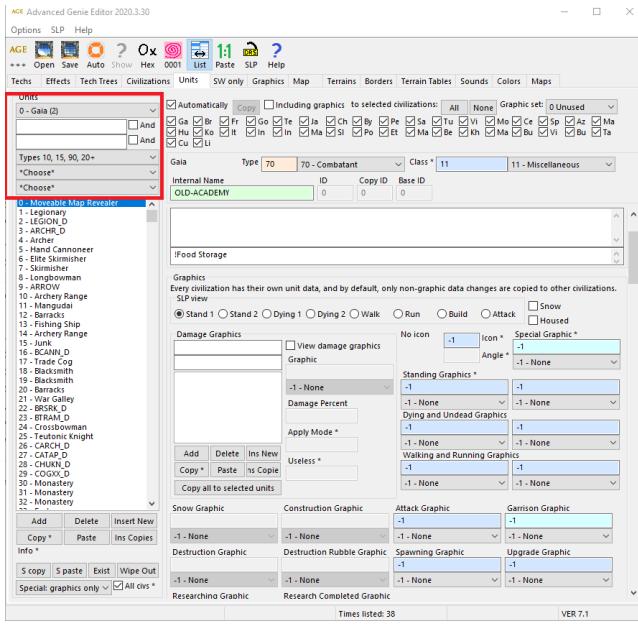


Figure 5: Standard Advance Genie Editor window. The red box highlights the search boxes you can use to filter

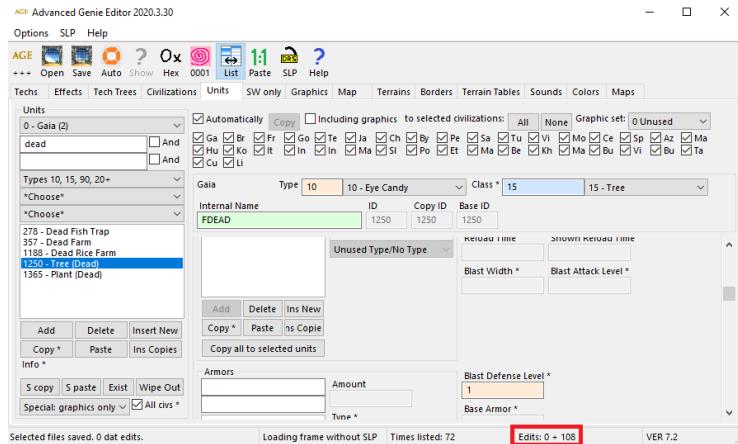


Figure 6: Example of how the bottom bar can be used to check how many changes we have made so far.

of Sight represents the number of cells the unit can see. Generates a circle centered in the unit. **Search Radius** is determines the area within which the unit will recognize and react to other units. If anything enters this area, the unit will notice it; If you enter a value higher than LOS, a unit will chase

enemy units outside of its sight range. **Min Range (max Range)** represents the minimum (maximum) range the ranged unit can shoot its projectile: for example an archer can shoot any target between 0 and 4 cells [7].

Resource capacity determines the quantity of resources a unit can hold (in range $[0, 3392]$): for villagers, this is the max amount of resources they can carry. For other units, it can be other resources, like the time it takes for a dead unit to decompose [7].

Resource Decay determines how fast objects (like corpses) decay. Set -1 for never decaying. A negative value will prevent the corpse from decaying permanently. The larger a positive value is, the longer the corpse will last. (The value is approximately the equivalent number of game seconds). Many non-corpse units have resource decay value, but the purpose of it in other units is unknown [7].

Work Rate determines the base work speed of units. This affects rates like villager work rates (building, gathering, etc.), conversion speed, and trebuchet pack (unpack) speed. the value needs to satisfy the constraint $x \geq 0$ [7].

Garrison Capacity is the number of units a transport or building can hold. allowed value x must satisfy the constraint $x \in [-1, 127]$. Only rams, buildings, and transport ships can normally hold units.³ If you give a rams an original speed of 0, garrisoning units will allow it to move.

Garrison type determines what units can garrison into the building ($x \in [0, 15]$). Such a integer value should be interpreted as an enumeration, which semantic is shown in Table 1. The number are encoded in Advance Genie Editor as a 8-bit value [7].

Garrison Heal Rate determines how fast a building heals units garrisoned within it ($x \geq 0$). This is different from the Garrison Recovery, which determines how fast the unit heals when garrisoned in any building [7].

.12.4 Projectiles

.12.5 Attributes

.12.6 Sounds

.12.7 Tasks

.13 Graphics

Represents the graphics and the animations involved in every units and buildings. This section has been copied from [9]. The left pane represents the sprites of each unit, building and so on. There is a search box that is similar to the one in “Units”.

Internal Name is the name used to identify the sprites in the Advance Genie Editor in the left pane of “Graphics” panel.

Frames per Angle represents the framerate of the sprite.

³Garrisoning infantry into rams will increase both the ram’s speed and attack.

Value	Semantic
0	None
1	Villagers only
2	Infantry only
3	Villagers and infantry
4	Cavalry
5	Cavalry and villagers
6	Cavalry and infantry
7	Cavalry, infantry and villagers
8	Monks only
9	Monks and villagers
10	Monks and infantry
11	Monks, infantry and villagers
12	Monks and cavalry
13	Monks, villagers and cavalry
14	Monks, cavalry and infantry
15	Monks, villagers, infantry and cavalry

Table 1: Semantic of each Garrison Type value.

.13.1 Deltas**.13.2 Angle Sounds**

RMS file format and layout

RMS files are used to generate random maps that follows some specific pattern. Random maps change each time you play since they are randomly generated starting from a seed s . Examples of such maps are *Black Forest* or *Scandinavia*. If you need to create a new type of random map, you need to provide a `rms` file. `rms` files are written by using a language (called *rms*) created by the developers of Age of Empires: DE [10]. This section will describes you a RMS file is structured. This section is based on the guide made by Ryan Andrews [10] and by [5].

At high level, a RMS file has 7 basic sections:

1. PLAYER_SETUP;
2. LAND_GENERATION;
3. ELEVATION_GENERATION;
4. CLIFF_GENERATION;
5. TERRAIN_GENERATION;
6. CONNECTION_GENERATION;
7. OBJECTS_GENERATION;

As an example, consider Listing 1 made by vierklee, which represents a random map which create an area where all the teams are located behind procedurally generated walls.

At high level, in a `rms` file comments follows C conventions (i.e., comments start with “`/*`” and end with “`*/`”). Each section starts with a **section declaration** and is followed by a section body. Each section declaration follow the regular expression `<[a-zA-Z0-9_]>\n`. Listing 2 specifies the Extended Backus–Naur Form (EBNF) of the RMS language. The language is case-sensitive and indentation is ignored [5].

```

1 Age of Empires 2 DE
2     |-- logs
3         |-- ...
4         |-- testharness
5         |-- 0
6         |-- 893465689237890
7             |-- mods
8                 |-- local
9                 |-- subscribed
10                |-- one folder per mod installed
11                |-- mod-status.json

```

Listing 1: The full source code of the rms file which generates the team maps “team arena”.

.14 RMS control flow

.14.1 Conditional flow

In RMS conditional flow (i.e., `if`, `then`, `else`) are implemented via the keywords `if`, `elseif`, `else`, `endif`. Both `elseif` and `else` are optional. Listing .14.1 shows an example of how to use the conditional flow.

```

1 if FOOBAR
2     /* do something when FOOBAR preprocessor directive is
3        ↪ specified */
4 elseif FOOBAZ
5     /* do something when FOOBAZ preprocessor directive is
6        ↪ specified */
7 else
8     /* do something else */
9 endif

```

.14.2 #include_drs

Include a file located in a given folder. If the filepath is relative, it is relative to the folder `Age of Empires: DE` `Exe\resources_common\drs\gamedata_x2` (e.g., `C:\Program Files (x86)\Steam\steamapps\common\AoE2DE\resources_common\drs\gamedata_x2`). Inside such a folder, you can several include files that you can use to ease your rms programming. The content of the included file will be dumped as-is in the current file.

.14.3 #const

C-like preprocessor. Every time the application see the identifier, it will be replaced by the associated value.

```

1 #const ANSWER 42

```

For example, in Listing .14.3 the statement allows the program to replace `ANSWER` with the string `42`.

```

1 identifier <- [a-zA-Z_](a-zA-Z0-9_)*
2 string <- [a-zA-Z_.\0-9]*
3 number <- [0-9]*
4
5 rms-file <-
6     section*
7
8
9 section <-
10    '<' identifier '>' section-statement*
11
12 section-statement <-
13     preprocessor-statement
14     | command-statement
15     | command-complex-value-statement
16     | if-statement
17
18 preprocessor-statement <-
19     "#define" identifier
20     | "#include_drs" string
21     | "#include" string
22
23 command-statement <-
24     identifier value*
25
26 command-complex-value-statement <-
27     identifier '{' section-statement* '}'
28
29 if-statement <- "if" identifier section-statement+ elseif-statement*
30     ↪ else-statement? "endif"
31
32 elseif-statement <- "elseif" identifier section-statement+
33
34 else-statement <- "else" section-statement+
35
36 value <-
37     number
38     | string

```

Listing 2: EBNF-like statements representing the language of the rms file.

.15 RMS file sections

In the following, we will describe each section.

.15.1 PLAYER_SETUP

Player placement In PLAYER_SETUP we need to first choose how the players starting position are put in the map. So, you need to put one of the following specifications:

- **random_placement**: players are positioned in a circle/oval. This is the default value;

- **grouped_by_team**: players are positioned in close proximity to each other. Distance between team members is double the base_size used in create_plater_lands;
- **direct_placement**: Allows to manually set the player positions. via create_land attribute in LAND_GENERATION. For example in Listing ?? the first player is positioned in the middle of the map.

```

1      <PLAYER_SETUP>
2      direct_placement
3      <LAND_GENERATION>
4      create_land {
5          terrain_type DESERT
6          land_percent 3
7          land_position 50 50
8          assign_to_player 1
9      }

```

If you plan to make the map as a nomad map, you should set the command nomad_resources: this will add the cost fo a town center to each player.

ai_info_map_type Next thing you can set is the Artificial Intelligence (AI) that Age of Empires: DE will use in the map. You can see as a hint that Age of Empires: DE will use. Such an hint is declared via command ai_info_map_type. The signature of the command is the one shown in Listing .15.1, where:

```

1  ai_info_map_type MapType IsNomad IsMichi ShowType

```

- MapType is the map type constants (see Section .18.2);
- IsNomad either 1 or 0: if you plan to make the map as “nomad”, you are **required** to set this value to 1;
- IsMichi: either 1 or 0: if true a forest will **completely** separating the players;
- ShowType: either 1 or 0: if true, the map type will be shown in the objective tab;

.15.2 LAND_GENERATION

Place some large areas of terrain in the map. For instance, some maps have some texture representing ground at each player starting location while the rest of the map is filled with grass. This section of the file performs this very feature.

The first to declare in this section is the type of the terrain you want to have, which can be achieved by using base_terrain command. Such a command requires a single input, which should be fetched from the terrain types (see Appendix .18.1).

Now you need to create large pieces of terrains on the maps. Terrain maps can be used to create a walkable texture at the starting position of a player or a lake in the center like the baltic map [11].

create_player_lands

The command allows you to create a specific **land** terrain for **every** player in the map. This command applies to all the players present in the map. If you want to assign different terrains to different players, consider using **create_land** with **assign_to_player** instead. The command is optional in the script. If the command can be used multiple times, but every player will have multiple town centers as well as result: this is probably how map “Budapest” is coded. Note that you give to a player any land, you cannot give to her any starting units or resources as well, so it is **very important** that each player has at least one starting land.



* **create_player_lands** cannot be used, due to a bug, alongside **direct_placement**.

Listing 3 shows an example of usage of the command. It will create, for each player, some mass of initial land for the players, where [?]:

```

1   create_player_lands
2   {
3       terrain_type          GRASS
4       base_size              19
5       land_percent           5
6       clumping_factor        3
7       border_fuzziness       15
8       set_zone_by_team
9       other_zone_avoidance_distance 0
10      circle_placement
11      circle_radius 38
12  }
```

Listing 3: Example of usage of **create_player_lands** in Team Arean by vierklee.

Figure 7 shows how teamless map generates each player starting position base.

The command requires several parameters, which are described next.

terrain_type the terrain you want to generate. See Section .18.1.

land_percent Percentage of the total map that the land to create should cover. the number belongs to the range [0, 100] (default to 100). If used in **create_player_lands**, the percentage is divided equally by all the players.



Figure 7: Example of how players are laid out in a map if automatic placement is used.

number_of_tiles Fixed number of tiles that the land should grow by, in addition to the land specified by base_size. When behavior_version is set to 1, the square origin is included in the total number of tiles, resulting in smaller lands. If used in create_player_lands, the player share the same amount of number of tiles.

land_position Allows you to specify the exact origin point for a land, as a percentage of the total map dimensions. It requires 2 parameters x and y which are, respectively, the x coordinate from the left point to the right and the y coordinate from the bottom to the top of the starting point. You need to satisfy the constraint $x \in [0, 100]$, $y \in [0, 99]$. The command cannot be specified when added into create_player_lands or when assign_to_player or assign_to is specified into create_land (unless direct_placement is specified in PLAYER_SETUP). The command ignores border restrictions and, if the land is placed outside the border, it will not grow beyond its base_size.

left_border, right_border, top_border, bottom_border All needs to satisfy the constraint $x \in [0, 97]$, default to 0. They specify an subset of the map where the land will be put. For instance, this allows the developer to put the players only in the bottom right part of the map (e.g., in Pilgrims); in create_player_lands, the border coordinates effectively shift the circle where the players are put; when using this function, the created land will have octoagonal shape. For further information, see https://docs.google.com/document/d/e/2PACX-1vR_I1I9u-hI2FFm-EYyjoM_-9dNJF0fTaIgr05wXNbdpv9tI18b6r18ARULy3Jqyvlq6-1c2VjX6xP4/pub#h.xrncn5cs75or.

set_zone_by_team

base_size It is the radius (in tiles) that each player starting position land will have, in map cells; The bigger it is, the bigger is each player base.

circle_radius It represents the radius of the circle where each player base will be placed. The bigger it is, the more distant each player will be w.r.t.the other ones. The players are put on the circonference of said circle; You need to specify 2 numbers a and b : if $a \neq b$, it is produced an ellipse instead of a circle; If $a = b$, b may not be specified.

circle_placement

border_fuzziness It applies some noise to each player starting land borders ($x \in [0, 100]$). Values near 0 corresponds to perfectly rectangular bases while bigger values tends to create bases with a square-less shape. A good value may be 15; border_fuzziness Specifies the extent to which land growth respects the assigned borders. Bigger values means that the borders are fully respected while values near 0 means that the generated area may go beyond the specified border.

clumping_factor Alters the shape of the created land ($x \in [0, 100]$). High values represent rounder lands while low values represents more elongated shapes [5, ?].

base_elevation elevates the entire land by the specified height ($x \in [0, 9]$). The value is ignore if the terrain is water. Slopes are automatically added to blend the elevation with the sorrounding terrain.

assign_to_player give ownership of this land to the given player. Allowed values are $x \in [1, 8]$ (which are the player lobby positional number). If missing, the land is neutral. Lands assigned to players which are not playing will not be created. Mutually exclusive with assign_to.

assign_to Given ownership of this land to the given entity. Mutually exclusive with assign_to_player. It requires 4 additional arguments.

1. AssignTarget, which tells which entity will take ownership of the land.
Can be either AT_PLAYER, AT_COLOR or AT_TEAM;
2. Number x : represents the specific entity Id of AssignTarget $t \in 0, 1, 2$.
If $t = \text{AT-PLAYER}$ (0), $x \in [1, 8]$ (refers to the lobby order); if $t = \text{AT-COLOR}$ (1), $x \in [1, 8]$ (refers to the player color); If $t = \text{AT-TEAM}$ (2), $x \in \{-10, -4, -3, -2, -1, 0, 1, 2, 3, 4\}$ (refers to the lobby order of the team, not the number chosen by the team): in this cse is the team 0 refers to unteammed players, while any negative value x represents a player

that is not in the team whose lobby order is $-x$. Team -10 is a special team that consists to any player;

3. Mode m : semanticful only if AssignTarget si set to AT_TEAM. either -1 or 0. 0 means the land is assigned randomly in the team while -1 means that the land is assigned in an orderly matter;
4. Flag $f \in [0, 3]$. It is a 2-bit biset number. First bit determines if we need to reset players who have already been assigned before startign whiel the second bit tells to avoid remembering the fact that we have assign to a given player this land;

When you assign a piece of land to a player, you can also place object owned by such a player. Lands owned by players which are not playing are not generated at all.

zone Allows the land you are creating to be assigned with a particular **zone id**. Lands which share the same zone id are allwoed to touch eacother when growing. If 2 lands that are going to overlap have different zone ids, will be forbidden to do so: even better, they are garantueed to be distant by other_zone_avoidance_distance. The command is mutually exclusive with set_zone_by_team and with set_zone_randomly. Lands created with create_player_lands have their own unique zone id, while all lands created with create_land share the same zone id.



Zone 99 will crash the game.

set_zone_by_team Assign the same zone to all the members of the same team. Usage is recommended only in create_player_lands.

set_zone_randomly TODO

other_zone_avoidance_distance Number of cells that each starting player position needs be distant from the other zones. If such a value is set to 0, each starting position base may overlap with others (like in Team Arena by vierklee).

min_placement_distance

land_id Assign a numeric label to a land, which can later be used to place object specifically on that land via place_on_specific_land_id command. Multiple lands may share the same ID. The commands needs to be used after assign_to_player or assign_to, since they will reset the ID.

create_land Creates a single piece of land which is owned by no players. Hence it can be used to place neutral mass of lands (e.g., shared continent or shared center lakes, like in Baltic). As an example, consider Listing 4: the example create a lake centered in the middle of the map. The lake itself covers 20% of the map

```

1     create_land
2     {
3         terrain_type WATER
4         land_percent 20
5         land_position 50 50
6     }

```

Listing 4: Example of usage of `create_land` in Team Arean by vierklee.

.16 Functions

The section provides the utility functions you can use within RMS file.

.16.1 Random

If you need to generate a random number, use `rnd` function.

1. `min`: the minimum number that the function can generate;
2. `max`: the maximum number that the function can generate;

It generates a number $x \in [min, max]$. The example in Listing .16.1 yields numbers in the set $\{5, 6, 7\}$.

```

1     foobar rnd(5,7)

```

.16.2 Choose a randomly chosen scenario

If you need to perform a different command depending on a randomly generated number, you can use the construct `start_random`, `end_random`, as shown in Listing .16.2: such a construct will set `FOO` to 3 in the 25% of the cases, 4 in the 25% of the cases and 5 in the remainder 50% of cases.

```

1     start_random
2         percent_chance 25 #const FOO 3
3         percent_chance 25 #const FOO 4
4         percent_change 50 #const FOO 5
5     end_random

```

.17 Popular included files

.17.1 thebr_setup.inc

Seems empty

.17.2 F_seasons.inc

You can include `F_seasons.inc` file. Such a file changes the texture of the random map, depending on which macros are defined: Table 2 shows which macros should be define in order to tweak the aesthetic of the map. Usually you need to first define one of the macros and then import `F_seasons.inc` (via `include_drs`).

.18 Constants

In this section there are described all the constants that you may use in a RMS file.

.18.1 Terrain Type

Values used in `base_layer` command. A full detailed list of how to use terrains is available at https://docs.google.com/document/d/e/2PACX-1vR_I1I9u-hI2FFm-EYyjoM_-9dNjFOfTaIgr05wXNbdpv9tI18b6r18ARULy3Jqvvlq6-1c2VjX6xP4/pub#h.3bdjnf7tryyk. Still, some default terrains are shown in Table 3 as well.

Macro	Description
PH_ALPINE	
PH_ALPINE_B	No leaves allowed
PH_SPRING	
PH_SPRING_C	
PH_MEDISOUTH	
PH_SPRING_B	No Leaves allowed
PH_TROPHICALSOUTH	
PH_TROPHICALSOUTH_B	Jungle read instead of grass 2
PH_TROPHICALEAST	
PH_DESERT	
PH_AFRICAN	
PH_ASIAN_B	Dry grass on layer C replaced by grass 1
PH_ASIAN	If no season is chosen, this one is used
PH_AUTUMN	
PH_AUTUMN_B	
PH_FROZEN	
PH_AFRICAN_B	cracked instead of Savannah for Layer A, terrain 6 not used
PH_AFRICAN_C	PALM DESERT instead of ACACIA for main forest
PH_AFRICAN_D	swap dirt 4 and 6 + baobab swap
PH_AFRICAN_E	replacing acacia trees with palm desert or dragon trees entirely, leaving only stragglers

Table 2: Available seasons in Age of Empires: DE.

ID	RMS Name	WololoKingdom Name	DE Scenario Editor	RMS FILE FORMAT AND LAYOUT	HD Scenario
0	GRASS	GRASS	Grass 1		Grass 1
1	WATER	WATER, DLC_WATER5	Water, Shallow		Water, Shallow
2	BEACH	BEACH, DLC_BEACH2, DLC_BEACH3, DLC_BEACH4	Beach		Beach
3	DIRT3	DIRT3, DIRT4, DLC_DIRT4	Dirt 3		Dirt 3
4	SHALLOW	SHALLOW, DLC_NEWSHALLOW	Shallows		Shallows
5	LEAVES	LEAVES, DLC_JUNGLELEAVES	Underbrush		Leaves
6	DIRT	DIRT	Dirt 1		Dirt 1
7	-	-	Farm		Farm
8	-	-	Farm, Dead		Farm, dead
9	GRASS3	GRASS3, MOORLAND	Grass 3		Grass 3
10	FOREST	FOREST, DLC_RAINFOREST	Forest, Oak		Forest
11	DIRT2	DLC_MANGROVEHALLOW	Dirt 2		Dirt 2
12	GRASS2	GRASS2, DLC_JUNGLEGRASS	Grass 2		Grass 2
13	PALM_DESERT	PALM_DESERT	Forest, Palm Desert		Palm Desert
14	DESERT	DESERT, SAVANNAH	Desert, Sand		Desert, Sand
15	-	-	Water 2D, Shoreless		Water 2D
16	ROCK1	BAOBAB, BAOBAB_FOREST, BAOBABS	Grass, Other		Grass, Other
17	JUNGLE	JUNGLE	Forest, Jungle		Jungle
18	BAMBOO	BAMBOO	Forest, Bamboo		Bamboo
19	PINE_FOREST	PINE_FOREST	Forest, Pine		Pine Forest
20	-	DLC_MANGROVEFOREST	Forest, Oak Bush		Oak Forest
21	SNOW_FOREST	SNOW_FOREST, DRAGONFOREST	Forest, Pine Snow		Snow Pine
22	DEEP_WATER	DEEP_WATER, DLC_WATER4	Water, Deep		Water, Deep
23	MED_WATER	MED_WATER	Water, Medium		Water, Medium
24	ROAD	ROAD	Road		Road
25	ROAD2	ROAD2, DLC_DRYROAD	Road, Broken		Road, Broken
26	-	-	Ice, Navigable		Ice
27	-	DIRT2	Grass, Foundation		Grass, Foundation
28	-	-	Water 2D, Bridge		Water 2D, Bridge
29	-	-	Farm, 0%		Farm, 0%
30	-	-	Farm, 33%		Farm, 33%
31	-	-	Farm, 67%		Farm, 67%
32	SNOW	SNOW	Snow		Snow
33	DIRT_SNOW	ROAD_SNOW, ROAD_SNOWY	-		Snow Dirts
34	GRASS_SNOW	GRASS_SNOW	-		Snow Grass
35	ICE	ICE	Ice		Ice2
36	-	DIRT_SNOW	Snow, Foundation		Snow, Foundation
37	ICYSHORE	-	Beach, Ice		Ice, Beach
38	-	CRACKEDIRT	-		Rocky Ground

Table 3: List of all the default terrain types available

.18.2 Map Type

Values used in ai_info_map_type. If the map deviate too much from the standard maps, consider using “CUSTOM” or leaving out the command entirely.

Name	Value	Note
ARABIA	9	
ARCHIPELAGO	10	
ARENA	29	
BALTIC	11	
BLACK_FOREST	12	
COASTAL	13	
CONTINENTAL	14	
CRATER_LAKE	15	
FORTRESS	16	
GHOST_LAKE	32	
GOLD_RUSH	17	
HIGHLAND	18	
ISLANDS	19	
KING_OF_THE_HILL	30	
MEDITERRANEAN	20	
MIGRATION	21	
MONGOLIA	26	
NOMAD	33	
OASIS	31	
RIVERS	22	
SALT_MARSH	28	
SCANDANAVIA	25	
TEAM_ISLANDS	23	
YUCATAN	27	
STEPPE	34	added in HD
BUDAPEST	35	added in HD
VALLEY	36	added in HD
ATLANTIC	37	added in HD
LAND_OF_LAKES	38	added in HD
LAND_NOMAD	39	added in HD
CENOTES	40	added in HD
GOLDEN_HILL	41	added in HD
MEGARANDOM	42	added in HD
MICHI	43	added in HD
AMBUSH	44	added in HD
CUSTOM	45	added in HD
NILE_DELTA	46	added in HD
MOUNTAIN_PASS	47	added in HD
SERENGETI	48	added in HD
SOCOTRA	49	added in HD
KILIMANJARO	50	added in HD

Table 4: List of all the map types available

Terms

Personal Computer A general purpose machine. v

Bibliography

- [1] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [2] Ozhara. Ozhara’s modding guide for aoe 2 hd. <https://steamcommunity.com/sharedfiles/filedetails/?id=190070326>. (accessed: 28.12.2020).
- [3] Yorok0. A guide to modding? <https://forums.ageofempires.com/t/guide-to-modding/61109/4>. (accessed: 28.12.2020).
- [4] [DD] Krampus. Bug: Stable keeps disappearing with video (now with how to fix it!). <https://steamcommunity.com/app/813780/discussions/2/2145343448814946626/?ctp=4#c2260186248407607030>. (accessed: 28.12.2020).
- [5] Zetnus. Definitive random map scripting guide. https://docs.google.com/document/d/e/2PACX-1vR_I1I9u-hI2FFm-EYyjoM_-9dNJFOfTaIgr05wXNbdpv9tI18b6r18ARULy3Jqyvlq6-lc2VjX6xP4/pub. (accessed: 29.12.2020).
- [6] TurnupHyperion4. A guide to modding? <https://forums.ageofempires.com/t/guide-to-modding/61109/2>. (accessed: 28.12.2020).
- [7] A Fandom user. Units. <https://agecommunity.fandom.com/wiki/Units>. (accessed: 28.12.2020).
- [8] Estien Nifo Mikko, Apre. Advanced genie editor. Advanced Genie Editor 2020.3.30.
- [9] Nelmatd. Graphics. <https://agecommunity.fandom.com/wiki/Graphics>. (accessed: 28.12.2020).
- [10] Zetnus Ryan Andrews, Bultrom. Age of empires rms guide. <http://aoe.dragonferocity.com/guide/>. (accessed: 29.12.2020).
- [11] Zetnus. Random map scripting links and faq. <http://aok.heavengames.com/cgi-bin/forums/display.cgi?action=ct&f=28,42485,,30>. (accessed: 30.12.2020).