Random Map Scripting for AoE II HD

William "HousedHorse" Findlay

November 26, 2018

Contents

1   About the Author

William "HousedHorse" Findlay has been playing Age of Empires II since his father introduced it to him when William was three years old.

He has played Starcraft II at a semi-professional level, reaching rank number 5 on the North American server. He is also 2100 rated on Age of Empires II HD.

William is an admin of the Age of Empires Memes Community group on Facebook, an Age fan page with over 30,000 members.

He works as a teaching assistant in the Computer Science department at Carleton University in Ottawa, Ontario, Canada. William enjoys scripting and coding and sees random map scripting as a great way to blend his passion for programming and his love of Age of Empires II.

2   What is RMS?

Welcome to my **random map scripting** guide! If you are here, you probably have some general idea of what an RMS is. If not, don't worry! I will be explaining all about random map scripting in this section and all the great things you can do with it.

When you think about custom games in Age of Empires II, typically a scenario is the first thing that comes to mind. Scenarios are great. You can do lots of cool things in them. The only issue with a scenario is that it is the same thing every time. And if a mapmaker does not take great care, a scenario can easily become too unbalanced for competitive play.

Age of Empires takes care of this problem with the concept of random maps. While a lot of games use "scenarios" for their competitive matches, Age of Empires competitive matches are played **exclusively** on random maps. By learning how to create random map scripts of your own, you can create custom map types for players to enjoy competitively.

Some examples of random maps include Arabia, Arena, Black Forest, Nomad, and many, many more. These are all examples of maps that come with the base game. However, if we journey to the `Custom` section in the lobby screen, we can see a list of **custom random maps**. In this guide, we are going to learn to create maps like these.

2.1   Why is it Called a Random Map Script?

Age of Empires II uses a random map generator to create its random maps. A random map generator is an **algorithm** which takes a random value called a seed and carries out a series of steps to output a full randomized map.

By default, the generator will spit out a flat grassy map with no resources and a town center, scout, and villagers for every player. But this is super boring. How do we make it do interesting things?

Well, it turns out we can write a text file called a **random map script**. This text file is structured in a special way so the random map generator can read it and use it as a guideline for map generation.

For example, we could set the terrain to be snowy instead of the default grass and add randomly placed lakes and resources. The generator will read these instructions and adjust its random output so the map fits our guidelines. Now we're starting to have something that looks more like a real map!

3 How to Use This Guide

In this section we will detail exactly how to read this guide. We will be going over some simple conventions we use to make reading the guide simpler and following that up with some basic RMS[1] syntax[2] to get you started and thinking about what a proper script looks like.

3.1 Guide Conventions

In this guide, we will be showing script examples throughout. The examples will be shown in the form of code blocks which look like Figure 1.

```
1  /* hello! */
2  /* this is an example code block! */
3
4  /* my custom terrain generation goes here! */
5  <LAND_GENERATION>
6  base_terrain GRASS /* this will create a grassy map! */
```

Figure 1: An example RMS code block.

Each block will be labeled with a figure number. This is the number that will be referenced in text. If you see a figure number in the text, you can take a look nearby for the accompanying sample code.

Code blocks like in Figure 1 will be surrounded by a black box. To the left of the blocks, you will see line numbers for your reference. If you are copying the sample code, **do not copy the line numbers**. They are just for your reference.

3.2 Basic RMS Syntax

In this subsection, we will learn about some basic syntax to get you started thinking about what an RMS should look like. Syntax is the way our script has to be formatted in order to allow the game to read it properly. Syntax is **super important** or your script will **break horribly** and you will be sad.

3.2.1 Comments

Comments are bits of text that you can read and understand but the random map generator ignores completely. It is important to comment your script so that you could come back and understand it later.

I also like to use comments in my script template so that when I am making a new script I have easy access to all the constants and functions I need. I will be going over my template much later in the guide.

So, how do we write a comment? It's pretty easy. Comments start with `/*` and end with `*/` . The `/*` and `*/` do not need to be on the same line and in this way you can write a comment that spans multiple lines. A comment can be as short or as long as you like. See Figure 2 for an example.

As you can see in Figure 2, it's pretty easy to make a comment. **BUT there is one super important thing to watch out for!** When you write a comment it is **required** to leave a space between the `/*` and the first word and the `*/` and the last word. **If you do not use those spaces your script WILL NOT WORK**.

---

[1]Random Map Script.
[2]The rules we have to follow so the generator can read our script.

```
1  /* this is a comment. */
2
3  /* this is
4  a multi−line
5  comment. */
```

Figure 2: The two types of comments in a random map script. Note the /* and */ which denote the comment's beginning and end. Also note the space between the beginning and end of a comment and the comment itself.

3.2.2 Constants

3.2.3 Sections

3.2.4 Blocks

3.2.4.1 Fields

4   Outline of a Script

4.1   The Preamble

4.2   Player Setup

4.3   Land Generation

4.4   Elevation Generation

4.5   Terrain Generation

4.6   Connection Generation

4.7   Object Generation

4.7.1   Player Towns

4.7.1.1   Regicide Support

5   Constants

6   Example Scripts