
Subject: [script]random map rotation code && !setnextmap

Posted by [reborn](#) on Fri, 01 Aug 2008 21:46:35 GMT

[View Forum Message](#) <> [Reply to Message](#)

I wanted to make the map rotation of my reborn server random, it was also needed for the APB server too. This is how I implemented it.

In gmmain.cpp for ssgm 2.0.2 I declared a global variable, like this:

```
char *mapname;
```

Then I made a function that sends a message in-game telling players what the next map will be. I wrote this first as it is called several times from different functions, and it suited me to place it at the top of my .cpp file.

```
//This just announces what the next map will be. The global variable "mapname" gets set by the  
request random map code.  
void mapnameannounce(){  
Console_Input(StrFormat("msg The next map will be %s",mapname).c_str());  
}
```

Then I needed to write the function itself, I have included some comments to help people read it properly.

```
//This is the function that makes the next map a random one from the current list of maps on your  
server.  
//You can call it at any time with "requestrandommap();".  
void requestrandommap(){  
//I get the name of the current map here  
char *currmapname = The_Game()->MapName;  
//I initialise and delclare the variable "numberofmaps here"  
int numberofmaps = 0;  
//Many thanks to Roshambo for this nice little "for" loop  
//The loop is basically responsible for getting the amount of maps in the rotation  
for(;*The_Game()->MapList[numberofmaps] != 0; numberofmaps++);  
//I get a random number between 0 (maps use 0 based indexing) and the amount of maps in  
rotation (hence the need to know the amount of maps).  
int RandomNum = Commands->Get_Random_Int(0,numberofmaps);  
//This code here makes the server think that the current map is a different one, so it logically will  
play the map next in the list to the one it thinks is currently being played  
The_Game()->MapNumber = RandomNum;  
//Therefore the next map that will get played is the one after the one that the server thinks is  
playing right now (but isn't), so this is how I get the name of the next map  
mapname = The_Game()->MapList[RandomNum +1];  
if (RandomNum + 1 > numberofmaps - 1){  
mapname = The_Game()->MapList[0];  
}  
//Code to make sure the next map will never be the same one as the current map, you need to be
```

running more than one map for this to work. I will always use more than one map so I never bothered to account for this.

```
if ((strcmp(currmapname,mapname))== 0){
  Console_Output("Had to request a respawn, just saved you playing the same map twice...\n");
  requestrandommap();
}
else {
//Just log the next map on the console
Console_Output("The next map will be: %s\n",mapname);
//Console_Output("The next map number is: %i\n",RandomNum + 1);
//Call the function that announces the next map to be played in-game.
mapnameannounce();
}
}
```

To make it actually work, I decided to call the function on the level_load event, so as soon as a map loads, it's deciding immediately what map to play next. I called the function here:

```
void Level_Loaded() {
  requestrandommap();
}
```

There is a problem though.. Most regulators have a !nextmap command. This command will produce errors. For our own regulator at MP-Gaming one of the other devs is changing the bot to remove this command, as I have made a chathook to replace it. You may need to ask Danpaul or the relevant person to disable this command for your server if you wish to run this code.

Also, here is the chat hook for !nextmap

```
class mapChatCommand : public ChatCommandClass {
  void Triggered(int ID,const TokenClass &Text,int ChatType) {
  mapnameannounce();
  }
};
ChatCommandRegistrant<mapChatCommand>
mapChatCommandReg("!nextmap",CHATTYPE_ALL,0,GAMEMODE_ALL);
```

Whilst I am making a thread... Does anyone know how to make several alias for the chat command? I would prefer to have several commands trigger it "!next" || "!N" || "!Next etc etc, rather than make like 8 different chat hooks to all call the same function.

Also, some people showed an interest in an in-game moderator function to set the next map of the servers rotation. So here it is (sorry for no comments, was in a hurry):

```
class setmapChatCommand : public ChatCommandClass {
```

```

void Triggered(int ID,const TokenClass &Text,int ChatType) {
    if(Is_Mod(Get_Player_Name_By_ID(ID))){
if (!Text[1].empty()) {
int number = 0;

std::string mapname2 = Text[1].c_str();

int numberofmaps = 0;
for(*The_Game()->MapList[numberofmaps] != 0; numberofmaps++);

for (int i = 0; i < numberofmaps; i++) {
if ((strcmp(The_Game()->MapList[i],mapname2.c_str()))==0) {
The_Game()->MapNumber = i-1;

mapname = The_Game()->MapList[i];
number++;
Console_Input(StrFormat("msg Moderator %s has changed the next
map!",Get_Player_Name_By_ID(ID)).c_str());
mapnameannounce();
break;
}
}

if(number < 1){
Console_Input(StrFormat("page %d Map does not exist, are you sure you spelt it
correctly?",ID).c_str());
}
}
}
else {
Console_Input(StrFormat("page %d You're not a mod, GTFO.",ID).c_str());
}
}
};
ChatCommandRegistrant<setmapChatCommand>
setmapChatCommandReg("!setnextmap",CHATTYPE_ALL,1,GAMEMODE_ALL);

```

To use it, you have to have your name listed in the mods.txt file, then you would type !setnextmap C&C_Hourglass.mix or some other map name in the servers rotation.

Oh, and the ismod function is here:

```

bool Is_Mod(const char *Name)
{
    ifstream file("mods.txt", ios::in);
    string tmp;
    while(file >> tmp)

```

```
{
  if(strcmp(Name, tmp.c_str()) == 0)
  {
    return 1;
  }
}
return 0;
}
```

You just need a a mods.txt file in your servers directory with player names of people you want to be a mod in it.

Many thanks to roshambo for his pretty neat for loop to count maps, awesome dude! Really helped a million
