Subject: Learn me good!
Posted by reborn on Wed, 18 Nov 2009 09:26:51 GMT
View Forum Message <> Reply to Message

I am struggling with something and would like some help.
Spoony requested a fix for the Nod rocket soldier so that the preset has some damage points associated with it, a clear bug that it currently doesn't...

I said I would do this, and I will... It isn't really difficult, but I want to take the opportunity to learn something and make a plugin that's a little more useful while I'm at it.

My plan is to allow server owners to configure the damage points for any preset, without having to directly modify the objects.ddb file on the server.

I could write a huge list in an ini file of all the presets, and assign them some floating point value for the damage point to be set on object created event. However, this is not very dynamic, it's incredibly static and boring...

What I would ideally like to do is allow the server owner to create a list of presets themselves, and then that list of presets would form what the rest of the keys in the .ini file would be. Meaning that the ini file would only consist of presets and keys that they wanted to change. Making it dynamic and more efficient.

My first attempt at this was to use the iniclasswrapper in the SSGM plugin example. I first used loadlist to read all the preset names, and then based on this list I wanted to do a little loop, but that doesn't really work...

I suppose what I would really need is soemthing like loadlist, but something that is Two dimensional, so it can take the preset name as a string, then the floating value for the damage points after it.

I want to open my mind a little bit on this. I am not averse to moving away from the iniclasswrapper and just digging into the iniclass itself. Perhaps get even scarier and have a look at xml files?

I'll get Spoony's fix done one way or another, I might even make a really quick static fox for him now actually. However, I would like to expand on this idea. I'm just a little stuck really.

If anyone has any suggestions, please post. However, what would be really kind is an example.

Subject: Re: Learn me good!
Posted by CarrierII on Wed, 18 Nov 2009 12:26:17 GMT
View Forum Message <> Reply to Message

Assuming you've got a large (preferably alphabetical) list of every preset:
Psuedocode:
CODE

```
Function BinarySearch(AString : String; Data: array of string) : integer; // -1 if not found, else the index in "Data" for which AString = Data[Index].
```

(I'll assume you can either code a binary search, or use another option if a binary search isn't possible for whatever reason)

```
For I := 0 to (InputFile.LineCount - 1) do
begin
  Index := BinarySearch(LeftStr(InputFile.Lines[I], Pos(',', InputFile.Lines[I], My_List_Of_Every_Preset)));
  If Index != -1 then begin
  SetPresetDamagePointsForPrefixByName(My_List_Of_Every_Preset[Index], StrToFloat(RightStr(InputFile.Lines[Index],  Length(InputFiles.Lines) - Pos(',', InputFile.Lines[Index]))); // I think I got the brackets right
  end
end else Continue;
```

Where the input file must look like this:

```
*PRESET_NAME*[U],[/U] DAMAGE_VALUE
```

In short, a simple text file and then check every line for whether the first part (before the comma) is a valid preset name (by binary searching an alphabetical list of every preset name) then setting the damage points for that preset (I don't know how that's achieved, so I made a up a routine for it.)

Subject: Re: Learn me good!
Posted by reborn on Wed, 18 Nov 2009 12:30:35 GMT
View Forum Message <> Reply to Message

I suppose I could read a disk file every time, but really I would prefer for it to be loaded in memory and save all the constant I/O.
I'm really looking for some sort of 2D ini key or something.

Thank you for the help though.

Subject: Re: Learn me good!
Posted by Hex on Wed, 18 Nov 2009 14:14:54 GMT
View Forum Message <> Reply to Message

If its just for the Nod rocket soldier just update the score on death?

```
 if(strstr(Commands->Get_Preset_Name(o),"CnC_Nod_RocketSoldier_1Off"))
 {
  for (GenericSLNode* PlayerIter = PlayerList->HeadNode; (PlayerIter != NULL); PlayerIter =
PlayerIter->NodeNext)
  {
   cPlayer *p = (cPlayer *)PlayerIter->NodeData;

   if (!p->IsActive)
   {
    continue;
   }
   if (p->PlayerId == Get_Player_ID(killer))
   {
    int NewScore = p->Score.Get() + xxx /*score you want to add or character*/;
    p->Score.Set(NewScore);
   }
  }
 }
```

This would work for a single character or for any you wanted to do if you want to do it for all characters you could do something like

```
int GetKillPoints(const char *Preset)
{
 int Points = 0;

 if (stricmp(Preset,"CnC_Nod_RocketSoldier_1Off") == 0) Points = xxx;
 else if (stricmp(Preset,"CnC_GDI_Grenadier_2SF") == 0) Points = xxx;
 return Points;
}
```

and

```
GetKillPoints(Commands->Get_Preset_Name(o))
```

Hex

Subject: Re: Learn me good!
Posted by reborn on Wed, 18 Nov 2009 14:26:54 GMT
View Forum Message <> Reply to Message

Oh yeah, sure. I'm not too worried about how to implement the change for the preset itself. There's a function for that anyway, Set_Damage_Points(GameObject *obj) or something like that. I would just call that on the object created hook, no big deal there...

What I am struggling with really is creating an ini file that has a vector list, something like:

```
[Presets]
01=CnC_Nod_RocketSoldier_1Off
02=CnC_something_else
03=another_cnc_preset
```

But then each element creates it's own key...

So then in the same ini file I could have:

```
CnC_Nod_RocketSoldier_10ff = 0.05
CnC_something_else = 0.01
another_cnc_preset = 0.2
```

So when the ini file loads the list, it then loads each element in that list as another key in the ini file.

I'm trying to allow the server owners to define the presets they want to edit the damage points for themselves in the ini file, without me having to create a long ass list. It would just be a little bit more dynamic...

However, I don't think the method I just posted is possible. Because that would mean declaring and initialising the variables inside the settings loader itself.

I was thinking possibly it might work if there was such thing as Load2dList, where the ini file would look like this:

```
[Presets]
01 = CnC_Nod_RocketSoldier_10ff, 0.05
02 =
```

However, I am not sure you can create such a thing. I am just looking for any possiblity or anyway to explore making it a reality. I do not like the idea of one long static list in the ini file, much of which will not be needed to change the default values, and therefore casuing un-neccessary over-head.

Subject: Re: Learn me good!
Posted by Hex on Wed, 18 Nov 2009 14:43:05 GMT
View Forum Message <> Reply to Message

just load it all in to an array?

```
[Presets]
01 = CnC_Nod_RocketSoldier_1Off 0.05
```

less to load

Subject: Re: Learn me good!
Posted by reborn on Wed, 18 Nov 2009 14:49:49 GMT
View Forum Message <> Reply to Message

Indeed, yeah, something like that. But how would I create that function to do it? The IniClassWrapper that SSGM/Plugin example uses does not have that sort of functionality...
Ideally I would be able to load a vector list, but one that takes first the std::string (or a proper string type, I seem to get abused when I use anything std::) and then take a floating point type...

Thank you for your replies, Hex.

Subject: Re: Learn me good!
Posted by Hex on Wed, 18 Nov 2009 15:06:24 GMT
View Forum Message <> Reply to Message

Hand written so some parts prob wrong

```
class PresetList
{
public:
 static char Preset[256];
 static float Points;
};

void PresetList()
{
 char Dir[64];
 char Preset_[256];
 float Points_;
 GetCurrentDirectory(64, Dir);
 strcat(Dir, "\\Presets.ini");
 INIClass *ini = Get_INI(Dir);
 if(!ini)
 {
  return 0;
 }


 ini->Get_String("Presets", "Preset", "ERR", Preset_, 256);
 ini->Get_Float("Presets", "Points", "ERR", Points_, 256);

 strcpy(Preset,Preset_);
 Points = Points_;
 Release_INI(ini);
}
```

Subject: Re: Learn me good!
Posted by reborn on Wed, 18 Nov 2009 15:09:12 GMT
View Forum Message <> Reply to Message

Hex wrote on Wed, 18 November 2009 10:06Hand written so some parts prob wrong

```
class PresetList
{
public:
 static char Preset[256];
 static float Points;
};

void PresetList()
{
 char Dir[64];
 char Preset_[256];
 float Points_;
 GetCurrentDirectory(64, Dir);
 strcat(Dir, "\\Presets.ini");
 INIClass *ini = Get_INI(Dir);
 if(!ini)
 {
  return 0;
 }


 ini->Get_String("Presets", "Preset", "ERR", Preset_, 256);
 ini->Get_Float("Presets", "Points", "ERR", Points_, 256);

 strcpy(Preset,Preset_);
 Points = Points_;
 Release_INI(ini);
}
```

That's the sort of thing I was looking for, thanks man. I've only ever used the wrapper, so it was nice of you to post that. Thanks!

Subject: Re: Learn me good!
Posted by jnz on Wed, 18 Nov 2009 17:11:24 GMT
View Forum Message <> Reply to Message

If you're looking for something other than INIs, check out my "config" project in the list of releases. It is a small class that will read a file setup like so:

```
//comment
/*comment
comment*/


foo
{
   bar = hello world from foo
   meh
   {
      whatever
      {
         bar = hello world from foo / meh / whatever
      }
   }
}
```