Subject: Quasar's fable 3 function listing
Posted by Keshire on Tue, 22 Oct 2013 03:53:44 GMT
View Forum Message <> Reply to Message

This was originally compiled and posted by quasar

Quote:

Functions named "NoName" are mostly subfunctions and from what I've seen, will probably be for/while loops so the majority (if not all of them) can safely be ignored.

Subfunctions are noted by an increased indent, and appear -BEFORE- the defining function. This is due to the way the binary Lua files are (you can't get to some of the current function's data without first going through it's subfunctions). So, for example:

DrawApproxCircle(pos,radius,red,green,blue,in_front_of_scene)
IsSpawnedBy(entity)
GetAllCreaturesSpawnedBy(spawning_entity)

The subfunction "IsSpawnedBy" is _not_ being defined by DrawApproxCircle(. Instead it is being defined in GetAllCreaturesSpawnedBy(.

File Attachments
1) functions.txt, downloaded 6014 times

Subject: Re: Quasar's fable 3 function listing
Posted by Artofeel on Tue, 22 Oct 2013 06:29:42 GMT
View Forum Message <> Reply to Message

I was wondering, how he got them?
also, this is only for main game, we need DLC scripts too...

Subject: Re: Quasar's fable 3 function listing
Posted by Keshire on Fri, 22 Nov 2013 15:08:58 GMT
View Forum Message <> Reply to Message

Artofeel wrote on Mon, 21 October 2013 23:29I was wondering, how he got them?
also, this is only for main game, we need DLC scripts too...

Found this
Quote:
For those wanting to write your own scripts and wishing you had a way to show some basic debug

info, this is what I've been using to grab the functions from the globals table:

```
local gstr
function MyMsgBox(text)
   if (text) then gstr=gstr and gstr.."<>"..text or text end
   if (string.len(gstr) >= 1400) then
      GUI.DisplayMessageBox(gstr)
      while (GUI.IsDisplayBoxActive()) do
         coroutine.yield()
      end
      gstr=nil
   end
end

function MyMsgBox2(text)
   if (text) then
      GUI.DisplayMessageBox(text)
      while (GUI.IsDisplayBoxActive()) do
         coroutine.yield()
      end
   end
end
```

These will display a message (text) on your screen. The first one I call in my loops that are getting the function names over in the "Function Definitions" thread, the second one I use when I'm not using a loop but needing a single tidbit of information. Until someone figures out how to get to the console (or, where cprint is spitting info to) this might be useful for someone wanting to debug something or look into something a bit deeper.

Here's an example use for the first one:

```
local p
local function outbox(inp)
  if (not p) then
    for i in pairs(inp) do
      MyMsgBox(i)
    end
    p="asdf"
  end
end

outbox(_G)
```

If this combined with the function above is put into a Lua file it will spit out all of the indexes of the global table, 1400 characters at a time, separated by "<>". The local variable of "p" is simply there

as a safety feature to ensure the function is only called once (sometimes it would be called more than once and I'd get message box after message box until I finally gave up and alt-f4'd).

Note that if you want to use these to dump table contents you will need a reliable way to continually call the script so it wont just stop on the first display (due to the coroutine.yield(. Setting up your own quest will give you the option to create an "Update" function that will suffice for this, but you'll have to figure that system out for yourself (for now).Global table (_G):

A(B)(C)

A - Name
B - If the type (C) is a table, then this is the # of values in the table
C - Type (String, Boolean, Table, Userdata - CFunction, etc)

Same deal as before, OCR and all that.
