Log

Functions for writing into Error.log or Benchmark.log

Log.LogError(Message [, Context])

Log.LogMsg(Message, Level [, Context])

Log.BenchmarkStart(Number)

Example:

see Log.LogBenchmark for example

Log.BenchmarkEnd(Number)

Example:

see Log.LogBenchmark for example

Log.LogBenchmark(Text, Number)

Writes the time used from Log.BenchmarkStart(Number) to Log.BenchmarkEnd(Number) into the Benchmark.log

Example:

```
Log.BenchmarkStart(10)
-- Do something
Log.BenchmarkEnd(10)
Log.LogBenchmark("Loading Textures", 10)
--[[ in the Benchmark.log there is now a line like
Loading Textures: 30 miliseconds
--]]
```

Log.LogDebug(Msg, Context)

Log.LogInfo(Msg, Context)

Log.LogStatus(Msg, Context)

Log.LogWarn(Msg, Context)

Log.LogCritical(Msg, Context)

Log.CriticalError(Text)

Log.SetLogLevel(Level)

Log.GetLogLevel()   : integer

---

gl

Functions for drawing with OpenGL

Example:

```
gl.Enable("GL_BLEND")
gl.Color(1, 1, 1, 0.1)
for i = 1,10 do
  gl.Begin("GL_LINE_LOOP")
  gl.Vertex(0+2*i,   0+2*i)
  gl.Vertex(0+2*i,   600-2*i)
  gl.Vertex(800-2*i, 600-2*i)
  gl.Vertex(800-2*i, 0+2*i)
  gl.End()
end
gl.Disable("GL_BLEND")
```

gl.Begin(EnumString)

Example:

```
gl.Begin("GL_LINE_LOOP")
```

gl.BindTexture(EnumString, Texturenumber)

Example:

```
gl.BindTexture("GL_GL_TEXTURE_2D", Tex.Num)
```

gl.BlendFunc(EnumString, EnumString)

Example:

```
gl.BlendFunc("GL_SRC_ALPHA", "GL_ONE_MINUS_SRC_ALPHA")
```

gl.Clear(EnumStringField)

Example:

```
gl.Clear("GL_COLOR_BUFFER_BIT,GL_DEPTH_BUFFER_BIT")
```

gl.ClearAccum(R, G, B, A)

Example:

```
gl.ClearAccum(1.0, 1.0, 1.0, 1.0)
```

gl.ClearColor(R, G, B, A)

Example:

```
gl.ClearColor(1.0, 1.0, 1.0, 1.0)
```

gl.Color(ColorArray)

Example:

```
col = {1.0,0.5,0.0}
gl.Color(col)
```

gl.Color(R, G, B [, A])

Example:

```
gl.Color(1.0,0.5,0.0)
```

gl.CullFace(EnumString)

Example:

```
gl.CullFace("GL_FRONT")
```

gl.DepthFunc(EnumString)

Example:

```
gl.DepthFunc("GL_LEQUAL")
```

gl.DepthRange(Near, Far)

Example:

```
gl.DepthRange(0, 10)
```

gl.Disable(EnumString)

Example:

```
gl.Disable("GL_BLEND")
```

gl.DisableClientState(EnumString)

Example:

```
gl.DisableClientState("GL_COLOR_ARRAY")
```

gl.DrawBuffer(EnumString)

Example:

```
gl.
```

gl.Enable(EnumString)

Example:

```
gl.Enable("GL_BLEND")
```

gl.EnableClientState(EnumString)

Example:

```
gl.EnableClientState("GL_COLOR_ARRAY")
```

gl.End()

Example:

```
gl.End()
```

gl.EndList()

Example:

```
gl.End()
```

gl.Finish()

Example:

```
gl.Finish()
```

gl.Flush()

Example:

```
gl.Flush()
```

gl.FrontFace(EnumString)

Example:

```
gl.
```

gl.InitNames()

Example:

```
gl.InitNames()
```

gl.LoadIdentity()

Example:

```
gl.LoadIdentity()
```

gl.LogicOp(EnumString)

Example:

```
gl.
```

gl.MatrixMode(EnumString)

Example:

```
gl.
```

gl.Ortho(Left, Right, Bottom, Top, Near, Far)

Example:

```
gl.Ortho(0.0, 800.0, 600.0, 0.0, -1.0, 100.0)
```

gl.PopAttrib()

Example:

```
gl.
```

gl.PopClientAttrib()

Example:

```
gl.PopClientAttrib()
```

gl.PopMatrix()

Example:

```
gl.PopMatrix()
```

gl.PopName()

Example:

```
gl.PopName()
```

gl.PushMatrix()

Example:

```
gl.PushMatrix()
```

gl.RasterPos(X, Y[, Z[, W]])

Sets the rasterposition for pixel operations.

Example:

```
gl.
```

gl.ReadBuffer(EnumString)

Example:

```
gl.Begin("GL_FRONT")
```

gl.Rect(PosArray,PosArray)

Draws a Rectangle.

Example:

```
gl.Rect({0+2*i, 0+2*i}, {800-2*i, 600-2*i})
```

gl.Rect(X1, Y1, X2, Y2)

Draws a Rectangle.

Example:

```
gl.Rect(0, 0, 800, 600)
-- does the same as
gl.Begin("GL_POLYGON")
gl.Vertex(0, 0)
gl.Vertex(0, 600)
gl.Vertex(800, 600)
gl.Vertex(800, 0)
gl.End()
```

gl.Rotate(Angle, X, Y, Z)

Example:

```
gl.
```

gl.Scale(X, Y, Z)

Example:

```
gl.Scale(1, -1, 1)
```

gl.ShadeModel(EnumString)

Example:

```
gl.
```

gl.TexCoord(PosArray)

Example:

```
gl.
```

gl.TexCoord(S [, T [, R [, Q]]])

Example:

```
gl.
```

gl.Translate(X, Y, Z)

Example:

```
gl.Translate(1, -1, 1)
```

gl.Vertex(PosArray)

Example:

```
vert = {10.0, 100.0, 2.0}
gl.Vertex(vert)
```

gl.Vertex(X, Y [, Z [, W]])

Example:

```
gl.Vertex(10.0, 20.0)
gl.Vertex(100, 20, 10, 1)
```

gl.Viewport(PosArray,SizeArray)

Defines the Viewport.

Example:

```
gl.Viewport({0, 0}, {800, 600})
```

gl.Viewport(X, Y, width, height)

Defines the Viewport.

Example:

```
gl.Viewport(0, 0, 800, 600)
```

TextGL

Functions for printing text

Texture

Functions for loading Textures

Party

Functions and Variables in Party are only available when the Partymode in the USDX Main menu is selected.
Scripts in Partymode run in their own Lua Environment.

InitPartyMode()

After the Playernames and Teamnames have been set, the Partymode reads all *.lua-files under Scripts\Party, and executes InitPartyMode().
The function InitPartyMode() must return `true` to be added to the list of PartyModi.

Example:

```
-- only use this PartyMode when it is christmas eve
local date = os.date("*t")
if (date.month == 12) and (date.day == 24) then
```

```
    return true
  else
    return false
  end
```

InitPartyMode() must set some variables:
Party.Mode.Name is the name of the Partymode. If this is a string, it will be translated if possible. The string should only be 32 chars long. It can also be a table with different Strings for different Languages. See Until5000.lua for an example. The name is shown in the Roundlist
Party.Mode.Description is the description of the Partymode. This will be translated, see Party.Mode.Name. The description should only use 64 chars.
If these are not set, the PartyMode will be not added to the list, even if InitPartyMode() returns `true`.

There are variables which can be optionally (they have default values) set. If they don't match the settings, the PartyMode will not be added to the List. The variables are:
Party.Mode.MinTeams How many teams there must be minimal (default `2`)
Party.Mode.MaxTeams How many teams there must be maximum (default `3`)
Party.Mode.MinTeamPlayers How many players there must be in each team min. (default `1`).
Party.Mode.MaxTeamPlayers How many players there must be in each team max. (default `4`).
Party.Mode.ForceDuet Set to `true`, if the song must be a duet (default `false`)
Party.Mode.NoDuet Set to `true`, if the song must not be a duet (default `false`)
If ForceDuet **and** NoDuet are set to `true`, all songs are loaded.

More variables which can optionally be set. If these are not set, it doesn't matter. It doesn't affect gameplay nor adding of the script to the list.
Party.Mode.Author is the name of the author of the Partymode.
Party.Mode.AuthorEmail is the eMail-Adress of the author of the Partymode.
Party.Mode.Creator is the name of the author of the Partymode.
Party.Mode.Homepage of the Partymode or the author.
Party.Mode.Version usefull for updates. String

For each round there is one PartyMode randomly selected from the list of available PartyModi. When the round starts, InitPartyMode() is called again (because it is in a new lua environment).

Draw()

The SingScreen draws lines, notes etc (if enabled in InitPartyMode), then Draw() is called.

In Draw() the script can for example draw additional things onto the screen (with the gl-Functions described in this document), and calculate the end of the Round. If the Draw() returns `true`, the Round continues, otherwise it ends.

Finish()

If the Round ends, Finish() is called.
Finish() can calculate from the Teamscores who wins the Round. It must return the winners in a array.

Example:

```
function Finish()
```

```
  local i
  local winners={}
  for i=0,Party.Teams-1 do
    if Party.Team[i].Score >= 5000 then
      table.insert(winners,i)
    end
  end
  return winners
end
```