File: divingoverlaysV3.2.2.lua 19Oct2023

**VS Code Outline view sorted by position in the code,** (as of V3.3.0 the following is not correct)

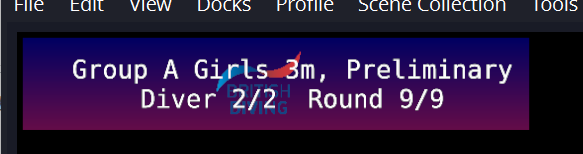
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| local | variable | obs |  | obs = obslua |  |
| local | variable | socket |  | socket = require(“ljsocket”) |  |
| local | variable | our\_server1 |  |  |  |
| local | variable | our\_server2 |  |  |  |
| local | variable | our\_server3 |  |  |  |
| local | variable | our\_server4 |  |  |  |
| local | variable | portClient |  | = 58091 |  |
| local | variable | portServer |  | = 58092 |  |
| local | variable | portWebUp |  | = 58093 |  |
| local | variable | portAwards |  | = 58094 |  |
|  | variable | Address1 |  | = socket.find\_first\_address(“\*”, portClient) |  |
|  | variable | Address2 |  | = socket.find\_first\_address(“\*”, portServer) |  |
|  | variable | Address3 |  | = socket.find\_first\_address(“\*”, portWebUp) |  |
|  | variable | Address4 |  | = socket.find\_first\_address(“\*”, portAwards) |  |
| local | variable | currentDataClient |  | to check if latest received data has changed |  |
| local | variable | interval |  | (ms), time between update file checks -- Again now not needed for UDP communications |  |
| local | variable | dinterval |  | the time to display the TV overlay after update |  |
| local | variable | debug |  | turn on or off debug information display in the Log |  |
| local | variable | synchro |  | default is Individual event |  |
| local | variable | simultaneousEvents |  | default is Single event |  |
| local | variable | event |  | default location for Event source overlay |  |
| local | variable | eventB |  | switch for using Event B data |  |
| local | variable | activeld |  | active file check id's, incremented for each programme parameter change or script initiated re-start |  |
| local | variable | current |  | current data file values to compare with next file update |  |
| local | variable | resultK |  | empty array where we will store data from the first UDP port data stream |  |
| local | variable | resultL |  | empty array where we will store data from the second UDP port data stream |  |
| local | variable | resultM |  | empty array where we will store data from the third UDP port data stream |  |
| local | variable | resultN |  | empty array where we will store data from the fourth UDP port data stream |  |
| local | variable | togglevar1, togglevar2, togglevar3 |  | variables to aid Hotkey Toggle functions |  |
| local | variable | disableUpdate |  | as it says! |  |
| local | variable | eventComplete |  | as it says! |  |
| local | variable | tvBanner\_removed |  | is or is not the banner being displayed? |  |
| local | variable | fileContentsChanged |  | has the data file changed since the last update flag? -- And again not needed for UDP comms |  |
| local | variable | hideDisable |  | default is to hide overlays after timeout |  |
|  | variable | htk\_1 |  |  |  |
|  | variable | htk\_2 |  |  |  |
|  | variable | htk\_3 |  |  |  |
|  | variable | htk\_4 |  |  |  |
|  | variable | htk\_5 |  |  |  |
|  | variable | htk\_6 |  |  |  |
|  | variable | htk\_7 |  |  |  |
|  | variable | htk\_8 |  |  |  |
|  | variable | htk\_9 |  |  |  |
| local | table name | plugin\_info |  | { |  |
|  | table variables | plugin\_info.name  plugin\_info.version  plugin\_info.url  plugin\_info.description  plugin\_info.author |  | } |  |
| local | table name | plugin\_def |  | { |  |
|  | table variables | plugin\_def.id  plugin\_def.type  plugin\_def.output\_flags |  | } |  |
|  | function | **getPath(str)** |  | sorts out the Windows file path separator character |  |
| local | function | **single\_update(v)** |  |  |  |
|  | variable | split\_string2 |  | = v variable passed into the function |  |
|  | variable | lineOne |  | = (“ … ”) set overlay display text line 1 to 50 spaces |  |
|  | variable | lineTwo |  | = (“ … ”) set overlay display text line 2 to 50 spaces |  |
|  | variable | penalty |  | generate the Penalty text |  |
|  |  | ***if synchro event*** |  |  |  |
|  | variable | display1a |  | = (“ “) |  |
|  | variable | displayName |  | divers name and club from split\_string2() |  |
|  | variable | display1b |  | = (“ “) |  |
|  | variable | sourcelineTwo |  | = (“ “) empty string so debug works correctly! |  |
|  | variable | position |  | generate the position text |  |
|  | variable | board |  | generate board text from split\_string2() |  |
|  | variable | display2b |  | Not used |  |
|  | variable | display2a |  | Not used |  |
|  |  | ***if individual event*** |  |  |  |
|  | variable | scores1 |  | from split\_string2(29) = Points |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| local | function | **simultaneous\_update(v)** |  |  |  |
|  |  |  |  |  |  |
|  |  | event A |  |  |  |
|  |  |  |  |  |  |
|  |  | event B |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  | function | **string\_insert(str1, str2, pos)** |  |  |  |
|  |  |  |  |  |  |
|  | function | **toggle\_event\_position(pressed)** |  | Hotkey F12 |  |
|  | function | **toggle\_simultaneous\_events(pressed)** |  | Hotkey S |  |
|  | function | **remove\_overlays(pressed)** |  | Hotkey F1 |  |
|  | function | **display\_overlays(pressed)** |  | Hotkey F2 |  |
|  | function | **redisplay\_overlays(pressed)** |  | Hotkey F3 |  |
|  | function | **toggle\_event\_type(pressed)** |  | Hotkey F9 |  |
|  | function | **toggle\_display\_disable(pressed)** |  | Hotkey F5 |  |
|  | function | **toggle\_disable\_of\_autohide(pressed)** |  | Hotkey F8 |  |
|  | function | **toggle\_event\_a\_or\_b(pressed)** |  | Hotkey F10 |  |
|  | function | **remove\_TVbanner()** |  | removes the overlay under timer control |  |
|  | function | **tvBanner\_remove()** |  | this calls the timer, timer\_remove, to remove the banner using function remove\_TVbanner() |  |
| local | function | **processMessage(k, v, x, y)** |  |  |  |
|  | function | **UDPtimer\_callback()** |  | get UDP data until there is no more, or an error occurs  if the Lua script has reloaded then stop any old timers and return |  |
|  |  | ***First port (58091)*** |  |  |  |
|  | variable | currentDataClient |  | Referee data UDP port |  |
|  |  | ***Second port (58092)*** |  |  |  |
|  | variable | currentDataServer |  | Find servers UPD port |  |
|  |  | ***Third port (58093)*** |  |  |  |
|  | variable | currentDataWebUp |  | Web update UDP port data |  |
|  |  | ***Fourth port (58094)*** |  |  |  |
|  |  | currentDataAwards |  | Awards update data UDP port |  |
|  | function | **Init()** |  |  |  |
|  | variable | id |  | = active ID |  |
|  |  |  |  |  |  |
|  |  | OBS Set-up functions for user configurable items |  |  |  |
|  | table name | hk |  | {} |  |
|  | variable | key\_1 |  |  |  |
|  | variable | key\_2 |  |  |  |
|  | variable | key\_3 |  |  |  |
|  | variable | key\_4 |  |  |  |
|  | variable | key\_5 |  |  |  |
|  | variable | key\_6 |  |  |  |
|  | variable | key\_7 |  |  |  |
|  | variable | key\_8 |  |  |  |
|  | variable | key\_9 |  |  |  |
|  | variable | json\_s |  |  |  |
|  | variable | default\_hotkeys |  | {} |  |
|  | function | **script\_load(settings)** |  | The function named "script\_load" will be called on startup |  |
|  | variable | s |  | loop variable for Hotkey registration |  |
|  | variable | a |  | loop variable for Hotkey registration |  |
|  | variable | h |  | loop variable for Hotkey registration |  |
|  | function | **script\_unload** |  | The function named "script\_unload" will be called on removal of script |  |
|  | function | **script\_update(settings)** |  | The function named "script\_update" will be called when settings are changed by the user |  |
|  | variable | flagLoc |  |  |  |
|  | function | **script\_description** |  | The function named "script\_description" returns the description shown to the user |  |
|  | function | **script\_properties** |  | The function named script\_properties defines the properties that the user can change for the entire script module itself |  |
|  | function | **script\_defaults(settings)** |  | The function named "script\_defaults" will be called to set the default settings and file locations |  |
|  | function | **script\_save(settings)** |  | The function named "script\_save" will be called when the script is saved |  |

**Overlay Sources and Text files**

***Single***

Single = Event 1 or Event 2

Event Banner



Single = EventData

(BD Logo)

Rank = split\_string2[32]

displayName & club = split\_string2[9]

Lineone = lineOne

Dive Overlay

Total = scores1 = split\_string2[29]



Linetwo = lineTwo

Dive description = sourcelineTwo = Dive Description + Position + Board

= split\_string2[61] + lookup{ split\_string2[14] + split\_string2[16] }

Flag = flag\_file

Rank = split\_string2[32]

Lineone = lineOne

displayName & club = split\_string2[9]

Total = scores1 = split\_string2[29]

Scores Overlay



Points

JudgeAwards

SynchroJLabels5 … 11

Penalty = penalty (if there is one)

Flag = flag\_file

***Simultaneous***

Source group = Event A

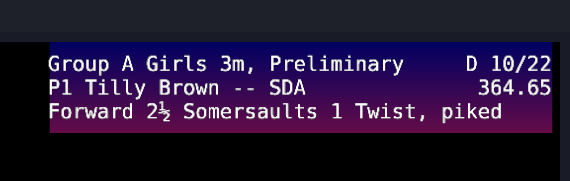
Source group = Event B



Text string = EventData\_B

Text string = EventData\_A

Simultaneous = Event A (left) or Event B (right)

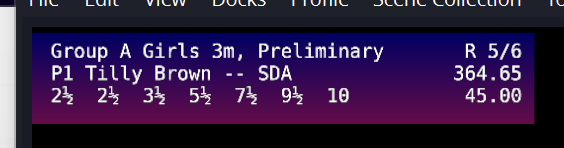


Text string = EventData\_A (left) or EventData\_B (right)

Three lines of text each line 40 characters long, (display1a, display1b and display1c)

Note change of information at end of first line. For dive description display shows Diver number of Total number Divers

Simultaneous = Event A (left) or Event B (right)



Text string = EventData\_A (left) or EventData\_B (right)

Three lines of text each line 40 characters long, ( lineTwo, sourcelineTwo)

Note change of information at end of first line. For dive awards display shows Round number of Total number of Rounds

Can penalty be displayed? Perhaps alternating between awards and the penalty text? Future update!

The Event A or Event B displays are constructed from several strings and then inserted into the text source. The text inserted should be three lines of text each 40 chrs long. However the display seems to randomly space the characters event though monotype font is used. The following is intended to be how the information is inserted into the text string using string\_insert(), (No synchro!):

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| \* | \* | \* | \* | \* | \* | \* | \* | \* | 1 | \* | \* | \* | \* | \* | \* | \* | \* | \* | 2 | \* | \* | \* | \* | \* | \* | \* | \* | \* | 3 | \* | \* | \* | \* | \* | \* | \* | \* | \* | 4 |
| G | r | o | u | p |  | A |  | G | i | r | l | s |  | 3 | m | , |  | P | r | e | l | i | m | i | n | a | r | y |  |  |  |  | D |  | 1 | 0 | / | 1 | 8 |
| P | 1 | 0 |  | T | i | l | l | y |  | B | r | o | w | n |  | - | - |  | S | D | A |  |  |  |  |  |  |  |  |  |  |  | 3 | 6 | 4 |  | . | 6 | 5 |
| F | o | r | w | a | r | d |  | 2 | ½ |  | S | o | m | e | r | s | a | u | l | t | s |  | 1 |  | T | w | i | s | t | , |  | p | i | k | e | d |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| \* | \* | \* | \* | \* | \* | \* | \* | \* | 1 | \* | \* | \* | \* | \* | \* | \* | \* | \* | 2 | \* | \* | \* | \* | \* | \* | \* | \* | \* | 3 | \* | \* | \* | \* | \* | \* | \* | \* | \* | 4 |
| G | r | o | u | p |  | A |  | G | i | r | l | s |  | 3 | m | , |  | P | r | e | l | i | m | i | n | a | r | y |  |  |  |  | R |  | 1 | 0 | / | 1 | 0 |
| P | 1 | 0 |  | T | i | l | l | y |  | B | r | o | w | n |  | - | - |  | S | D | A |  |  |  |  |  |  |  |  |  |  |  | 3 | 6 | 4 |  | . | 6 | 5 |
| 0 | ½ |  | 1 | ½ |  | 2 | ½ |  | 3 | ½ |  | 4 | ½ |  | 5 | ½ |  | 6 | ½ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 4 | 5 | . | 0 | 0 |

Events more than 32 chars long will be truncated by removing the chars after char 32.

Names more than 32 chars long will be truncated by removing the chars after char 32.

Dive description more than 40 chars long will be truncated by removing the chars after char 40.

**Status Dock – Simultaneous Events**

(not like this anymore!)



**Status Dock – Single Event**

(not like this anymore!)

