Package: ovva (via r-universe)

September 6, 2024

Title A Shiny App for Volleyball Video Analysis

Version 0.12.6

Description A Shiny app and supporting functions for analysis of
volleyball video and scouted match files.

URL https://ovva.openvolley.org, https://github.com/openvolley/ovva

BugReports https://github.com/openvolley/ovva/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Depends datavolley (>= 0.13.0)

Imports assertthat, base64enc, dplyr (>= 1.0.0), digest, DT, fs,
future, future.apply, htmltools, methods, ovideo (>= 0.20.2),
peranavolley, rappdirs, rlang, servr (>= 0.15.1), shiny,
shinyWidgets, sys, utils

Suggests covr, rintrojs, testthat

Remotes openvolley/datavolley, openvolley/ovideo,
openvolley/peranavolley

RoxygenNote 7.2.1

Repository https://openvolley.r-universe.dev

RemoteUrl https://github.com/openvolley/ovva

RemoteRef HEAD

RemoteSha 65e8da67dcd388b2948aeb42562726aad6839365

Contents

___

ovva   **ovva**

___

Description

A Shiny app and supporting functions for volleyball video and scouted match files.

___

ovva_highlight_handler
                              The default handler for highlights within the shiny app

___

Description

`ovva_highlight_handler` provides functions that create highlight playlists. Highlights are judged according to the weighting scheme defined by `ovva_highlight_weighting`.

Usage

```
ovva_highlight_handler(clip_duration = 180)

ovva_highlight_weighting(weights)
```

Arguments

clip_duration	numeric: the maximum highlight clip length in seconds
weights	data.frame: a data.frame or tibble with columns `skill`, `evaluation_code`, and `highlight_weighting`. Any values provided here will override the corresponding defaults
	You can cook up your own version of this highlight handler function to suit your liking, and pass it as the `highlight_handler` parameter to [ovva_shiny](#).

Value

A tibble, with columns 'skill', 'specific', and 'fun'. Each row defines a playlist entry that corresponds to specific match conditions. This playlist entry is referred to by the name given in `specific`, and applies to the skill in column `skill`. The corresponding entry in `fun` should be a function that takes three arguments (`x`, the play-by-play data; `team`, the team names to look for, and `player`, the player names to look for) and returns the subset of rows of `x` that correspond to the playlist conditions.

See Also

ovva_shiny

---

ovva_install_lighttpd   *Install lighttpd*

Description

This is a helper function to install lighttpd. Currently it only works on Windows platforms. The lighttpd bundle will be downloaded from <http://lighttpd.dtech.hu/> and saved to your user appdata directory.

Usage

```
ovva_install_lighttpd(force = FALSE)
```

Arguments

force   logical: force reinstallation if lighttpd already exists

Value

the path to the installed executable

References

<http://lighttpd.dtech.hu/>

Examples

```
## Not run:
  ovva_install_lighttpd()

## End(Not run)
```

---

ovva_playlist_handler   *The default handler for playlists within the shiny app*

Description

You can cook up your own version of this function to suit your liking, and pass it as the playlist_handler parameter to ovva_shiny.

Usage

```
ovva_playlist_handler()
```

Value

A tibble, with columns 'skill', 'specific', and 'fun'. Each row defines a playlist entry that corresponds to specific match conditions. This playlist entry is referred to by the name given in specific, and applies to the skill in column skill. The corresponding entry in fun should be a function that takes three arguments (x, the play-by-play data; team, the team names to look for, and player, the player names to look for) and returns the subset of rows of x that correspond to the playlist conditions.

See Also

ovva_shiny

---

ovva_shiny   *Launch the Shiny app*

Description

Launch the Shiny app

Usage

```
ovva_shiny(
  data_path,
  playlist_handler = ovva_playlist_handler(),
  highlight_handler = ovva_highlight_handler(),
  video_server = "lighttpd",
  launch_browser = TRUE,
  video_timing_df = ov_video_timing_df(),
  ...
)
```

Arguments

data_path	character or function: a named character vector of paths to data files. The names will be used as the competition names. So e.g. c("Competition 1" = "/path/to/dvw/files", "Competition 2" = "/path/to/other/dvw/files"). If data_path is a function, it should return such a character vector
playlist_handler	
	tibble: a tibble that provides playlist handler capabilities (see ovva_playlist_handler for details)
highlight_handler	
	tibble: a tibble that provides playlist handler capabilities (see ovva_highlight_handler for details)
video_server	string or function: if string, either "lighttpd", "servr", or "none". If a function, it will be used to modify the video file path present in each dvw file. Details TBD
launch_browser	logical: if TRUE, launch the app in the system's default web browser (passed to runApp's launch.browser parameter). If NULL, don't launch the app, just return the shinyApp object
video_timing_df	
	data.frame: data.frame of default clip timings
...	: additional parameters passed to the UI and server functions

See Also

ovva_shiny_demo

Examples

```
## Not run:
  ## to use your own data, assuming that your data files are in the data/volley/ folder
  ovva_shiny(data_path = c(MyData = "data/volley"))

  ## or to use the bundled demonstration file
  ovva_shiny_demo()

## End(Not run)
```

---

ovva_shiny_demo	*Start the ovva shiny app with a demonstration data set*

Description

The demonstration data set is the 2018 women's final (first of 3) from the German Bundesliga: Allianz MTV Stuttgart vs SSC Palmberg Schwerin.

Usage

```
ovva_shiny_demo()
```

See Also

ovva_shiny

Examples

```
## Not run:
  ovva_shiny_demo()

## End(Not run)
```

---

ovva_video_server   *Start a separate server process for serving video*

Description

Note that performance with "servr" may not be great, particularly with each new video, because servr has to read the entire video file before it can serve clips from it.

Usage

```
ovva_video_server(method, port)
```

Arguments

method	string: either "lighttpd" or "servr". If "lighttpd" but the lighttpd executable cannot be found, "servr" will be used as a fallback
port	integer: the port to open the server on. If missing or NULL, a random port between 8001 and 12000

Value

A list with components `method`, `url`, `port`, `dir` (the source dir from which files will be served), and `cleanup_fun` (a function to run when finished with the server, which will stop the server process and clean up its `dir`)

See Also

ovva_shiny

Index