

Package: foosball3 (via r-universe)

October 1, 2026

Title An App to Manage and Organise Foosball Tournaments

Version 0.0.0.9023

Author Pepijn de Vries [aut, cre, dtc] (0000-0002-7961-6646)

Description A relational SQLite database used to manage foosball tournaments. This package provides a features and a graphical user interface to easily access the database. It also allows to visualise (historical) data and account player history. It can be used to facilitate a live tournament.

Depends R (>= 4.1.0)

Imports blob, dbplyr, dplyr, methods, rlang, RSQLite, shiny, stringr, tidyr

Suggests base64enc, bsicons, bslib, DiagrammeR, DiagrammeRsvg, dm, DT, ggplot2, ggiraph, imager, jsonlite, knitr, lubridate, mirai, promises, shinybusy, shinyjs, shinyvalidate, shinyWidgets

License GPL (>= 3)

Encoding UTF-8

Roxygen list(markdown = TRUE)

URL <https://github.com/pepijn-devries/foosball3>,
<https://pepijn-devries.github.io/foosball3/>

BugReports <https://github.com/pepijn-devries/foosball3/issues>

Collate 'create.R' 'dependencies.R' 'export.R' 'suggests.R'
'foosball3.R' 'generate.R' 'import.R' 'meta.R'

Config/roxygen2/version 8.1.0

Config/pak/sysreqs cmake make libicu-dev libuv1-dev zlib1g-dev

Repository <https://pepijn-devries.r-universe.dev>

Date/Publication 2026-09-30 21:10:55 UTC

RemoteUrl <https://github.com/pepijn-devries/foosball3>

RemoteRef main

RemoteSha 1be9c00a5492b703676e4d408c938d00872fb8aa

Contents

foosball3	2
foosball3_create_db	2
foosball3_export_db	3
foosball3_generate_matches	4
foosball3_import_db	6
foosball3_install_suggests	7
foosball3_list_suggests	7
foosball3_meta_data	8
foosball3_suggests_ok	9

Index	10
-------	----

foosball3	<i>Start the Foosball App</i>
-----------	-------------------------------

Description

Renders the Graphical User Interface and starts the server to interact with a Foosball SQLite database.

Usage

foosball3(...)

Arguments

... Ignored

Value

Returns NULL invisibly.

foosball3_create_db	<i>Create a New Foosball Database</i>
---------------------	---------------------------------------

Description

Creates an SQLite file at the specified location. It will contain all required relationships and constraints and some non-trivial data.

Usage

foosball3_create_db(file, ...)

Arguments

file	File path where the database needs to be stored
...	Ignored

Value

Returns NULL invisibly.

Examples

```
f1 <- tempfile(fileext = ".sqlite")
foosball3_create_db(f1)

## Clean up:
unlink(f1, TRUE, TRUE)
```

foosball3_export_db	<i>Export an Foosball SQLite Database to csv Format</i>
---------------------	---

Description

Export a database to a collection of csv-files. Note that you will lose all views, relational information and constraints from the database.

Usage

```
foosball3_export_db(source, destination, validate = FALSE, ...)
```

Arguments

source	File path, where the source SQLite database can be found
destination	File path, where the zipped collection of csv-files should be stored
validate	A logical value. If TRUE, the source will be imported to check for validity (slower). If FALSE, source will be used directly without checks.
...	Ignored.

Value

NULL invisible

Examples

```
new_db <- tempfile()
foosball3_create_db(new_db)
new_export <- tempfile(fileext = ".zip")
foosball3_export_db(new_db, new_export)

## Clean up the files created in this example
unlink(new_db, TRUE, TRUE)
unlink(new_export, TRUE, TRUE)
```

```
foosball3_generate_matches
```

Generate Matches for a Tournament Phase

Description

Generates matches for a specific phase in a foosball tournament for a given set of participants.

Usage

```
foosball3_generate_matches(
  participants,
  tournament_type = "individual",
  tournament_phase = "qualification",
  options,
  progress,
  ...
)
```

Arguments

participants	A data.frame with two columns: first PARTICIPANT_ID, a unique integer identifier for each participant. Second a column named QUALIFICATION_RATE, it holds numeric values greater than zero. A qualification rate of 0.5 indicates an average player. Values less than 0.5 are for less experienced players. Values greater than 0.5 indicate more experienced players.
tournament_type	Type of tournament for which to render matches. At the moment only "individual" is supported. This is a tournament where player play in variable team compositions, but for individual results.
tournament_phase	The phase of the tournament. At the moment only qualification is supported. These are matches preceding a semi finals, where individuals can collect points individually to qualify for the semi finals.
options	Options used for generating the matches. It can be used to tweak the effort for balancing matches for player experience. For "individual" tournaments the following options are available, for generating qualifying matches:

- `nsim`: Number of simulated tournaments, from which to pick a suitable configuration. Given the constraint that all players play the same number of matches on each of the positions of the table.
- `revolutions`: Number of times players play on each position of the table during the qualification.
- `weights`: A named list of weights used for selecting suitable tournaments from simulated permutations.

The weights list should contain the following numeric weights scaled between 0 (not applied) and 1 (fully applied):

- `teamup`: Avoid that specific duos team up frequently
- `opposing`: Avoid that specific duos oppose each other frequently
- `match_var`: Minimise variation in predicted match results (based on historical performance)
- `match_bal`: Ensure mean matches are balanced
- `match_bal_var`: Minimise variance balance in matches
- `match_bal_extr`: Avoid extreme unbalance in matches
- `part_bal`: Ensure mean balance of participants
- `part_bal_var`: Minimise variance in participants balance
- `part_bal_extr`: Avoid extremes in participants balance

`progress` A callback function for reporting progress. It needs to be a function that accepts 1 numeric argument. It is called by the generator, with a number between 0 and 1 indicating the progress of the process.

`...` Ignored

Value

It returns a `data.frame` with three columns. Firstly, `MATCH_ID` which is a unique integer identifier for each match. Secondly, `PARTICIPANT_ID`, which is from the input `participants`, and indicates the participants enrolled in the indicated match. Thirdly, `POSITION_CODE`, indicating where the participant is positioned for each specific match: S1 = striker side 1, D1 = defender side 1, S2 = striker side 2 and finally D2 = defender side 2.

Examples

```
participants <-
  dplyr::tibble(
    PARTICIPANT_ID = 5L:9L,
    QUALIFICATION_RATE = c(0.3, 0.5, 1, 1.3, 1.5)
  )
opts <- list(
  nsim = 100,
  revolutions = 2,
  weights = list(
    teamup = 1,
    opposing = 1,
    match_var = 1,
    match_bal = 1,
```

```

        match_bal_var = 1,
        match_bal_extr = 1,
        part_bal = 1,
        part_bal_var = 1,
        part_bal_extr = 1
    ),
    seed = 0
)

foosball3_generate_matches(
  participants, options = opts)

```

foosball3_import_db	<i>Import Foosball Tournament Data from a File</i>
---------------------	--

Description

If you have an older file with tournament results, you can try to import those results with this function. It will create a new copy of the database (using [foosball3_create_db\(\)](#)), making sure it complies with the latest database scheme specification used by this package.

Usage

```
foosball3_import_db(file, target, ...)
```

Arguments

file	An SQLite file or a zipped collection of csv files (created with foosball3_export_db()), to be imported into the standardised database structure.
target	Target file path where the clean database will be stored. Existing files at this location may be overwritten.
...	Ignored

Value

Creates a new clean copy of the database. Returns NULL invisibly.

Examples

```

tf <- tempfile()
export <- tempfile(fileext = ".zip")
import <- tempfile()
foosball3_create_db(tf)
foosball3_export_db(tf, export)
foosball3_import_db(export, import)

# Clean up example files
unlink(tf, TRUE, TRUE)
unlink(export, TRUE, TRUE)
unlink(import, TRUE, TRUE)

```

`foosball3_install_suggests`*Install Packages Suggested by foosball3*

Description

Call this function if `foosball3_suggests_ok()` returns FALSE. This will install all suggested packages required to run the graphical user interface (`foosball3()`).

Usage

```
foosball3_install_suggests(ignore = "mirai", update = TRUE, ...)
```

Arguments

<code>ignore</code>	A vector of packages to ignore. By default "mirai" is ignored, as this package is not needed for basic functionality. It is recommended to install for better performance.
<code>update</code>	A logical value. If TRUE, it will try to update suggested packages that are already installed. If FALSE, it will only install suggested packages that are not yet available.
<code>...</code>	Arguments passed to <code>utils::install.packages()</code>

Value

invisible NULL

Examples

```
foosball3_suggests_ok()
```

`foosball3_list_suggests`*List Packages Suggested by the foosball3 Package*

Description

These are all packages required for the graphical user interface (`foosball3()`). Make sure that they are installed

Usage

```
foosball3_list_suggests(...)
```

Arguments

... Ignored

Value

Returns a vector of package names that are suggested by foosball3 and required by its graphical user interface.

Examples

```
foosball3_list_suggests()
```

foosball3_meta_data	<i>Get Meta Data for Tables in the Foosball Database Scheme</i>
---------------------	---

Description

Returns a list of tables and fields in the Foosball database. In addition, it contains a description of the table and each field.

Usage

```
foosball3_meta_data(...)
```

Arguments

... Ignored.

Value

Returns a `data.frame` with description information

Examples

```
foosball3_meta_data()
```

foosball3_suggests_ok Check if All Suggested Packages Are Available

Description

This function returns TRUE if all packages required for the graphical user interface (GUI) ([foosball3\(\)](#)) are available.

Usage

```
foosball3_suggests_ok(ignore = "mirai", ...)
```

Arguments

ignore	A vector of packages to ignore. By default "mirai" is ignored, as this package is not needed for basic functionality. It is recommended to install for better performance.
...	Arguments passed to foosball3_list_suggests()

Value

A logical value, indicating if all required suggests are available.

Examples

```
foosball3_suggests_ok()
```

Index

foosball3, [2](#)
foosball3(), [7](#), [9](#)
foosball3_create_db, [2](#)
foosball3_create_db(), [6](#)
foosball3_export_db, [3](#)
foosball3_export_db(), [6](#)
foosball3_generate_matches, [4](#)
foosball3_import_db, [6](#)
foosball3_install_suggests, [7](#)
foosball3_list_suggests, [7](#)
foosball3_list_suggests(), [9](#)
foosball3_meta_data, [8](#)
foosball3_suggests_ok, [9](#)
foosball3_suggests_ok(), [7](#)

utils::install.packages(), [7](#)