# Package ‘CamelUp’

October 3, 2019

**Title**  
CamelUp Board Game as a Teaching Aid for Introductory Statistics

**Version** 0.1.1

**Description**  
Implements the board game CamelUp for use in introductory statistics classes using a Shiny app.

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.1.1

**Imports** data.table, parallel, R6, shiny, shinyalert, tidyverse

**NeedsCompilation** no

**Author**  
Michael Czekanski [aut, cre],  
Alex Lyford [aut],  
Tom Rahr [aut],  
Tina Chen [aut]

**Maintainer** Michael Czekanski <middleburystatspackages@gmail.com>

**Repository** CRAN

**Date/Publication** 2019-10-02 23:00:02 UTC

## R topics documented:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>bet</td>
<td>2</td>
</tr>
<tr>
<td>board</td>
<td>2</td>
</tr>
<tr>
<td>camel</td>
<td>3</td>
</tr>
<tr>
<td>cleanColors</td>
<td>3</td>
</tr>
<tr>
<td>die</td>
<td>4</td>
</tr>
<tr>
<td>generateUI</td>
<td>4</td>
</tr>
<tr>
<td>overall.bet</td>
<td>5</td>
</tr>
<tr>
<td>playCamelUp</td>
<td>5</td>
</tr>
<tr>
<td>player</td>
<td>6</td>
</tr>
<tr>
<td>server</td>
<td>6</td>
</tr>
<tr>
<td>space</td>
<td>7</td>
</tr>
<tr>
<td>stack</td>
<td>7</td>
</tr>
<tr>
<td>system</td>
<td>8</td>
</tr>
</tbody>
</table>
bet

A bet object that is placed for a leg on a given camel

Description
A bet object that is placed for a leg on a given camel

Usage
bet

Format
An object of class R6ClassGenerator of length 24.

Examples
x <- bet$new("blue", 5)

board

A board object on which the game is played

Description
A board object on which the game is played

Usage
board

Format
An object of class R6ClassGenerator of length 24.

Examples
y <- system$new(nPlayers = 2, players = c("Michael", "Alex"))
x <- board$new(y)
camel

Implements camel class based off of the board game pieces

Description

Implements camel class based off of the board game pieces

Usage

camel

Format

An object of class R6ClassGenerator of length 24.

Examples

x <- camel$new("blue", 1, 1)

cleanColors

Correctly orders colors for graphing

Description

Correctly orders colors for graphing

Usage

cleanColors(subColors)

Arguments

subColors a subset of the full set of colors of camels

Value

colors in correct order for the game
die  

A three sided die, assigned a color corresponding to a camel

Description
A three sided die, assigned a color corresponding to a camel

Usage
die

Format
An object of class R6ClassGenerator of length 24.

Examples
x <- die$new("blue")

generateUI  

Generate CamelUp UI

Description
Generate CamelUp UI

Usage
generateUI()

Value
a shiny ui object
overall.bet  

A bet object that is placed overall on a given camel

Description

A bet object that is placed overall on a given camel

Usage

overall.bet

Format

An object of class R6ClassGenerator of length 24.

Examples

s <- system$new(nPlayers = 2, players = c("Michael", "Alex"))
p <- s$players[[1]]
x <- overall.bet$new("blue", p)

playCamelUp  

Play the game CamelUp

Description

Run CamelUp in a local web browser. Running locally allows for using the app without an internet connection and running in parallel on the local computer

Usage

playCamelUp()

Value

an object representing the CamelUp app as generated by shiny::shinyApp
player

*Player object to represent each player using bets and a purse*

### Description

Player object to represent each player using bets and a purse

### Usage

```
player
```

### Format

An object of class `R6ClassGenerator` of length 24.

### Examples

```
s <- system$new(nPlayers = 2, players = c("Michael", "Alex"))
x <- player$new("Michael", s$board)
```

---

server

*Define server logic for CamelUp game*

### Description

Define server logic for CamelUp game

### Usage

```
server(input, output)
```

### Arguments

- **input**: input for shiny web app
- **output**: output for shiny web app
**space**

*Implements spaces on the board*

**Description**

Implements spaces on the board

**Usage**

```r
space
```

**Format**

An object of class `R6ClassGenerator` of length 24.

**Examples**

```r
x <- space$new()
```

---

**stack**

*Implements a classic stack with push, pop and a few other methods*

**Description**

Implements a classic stack with push, pop and a few other methods

**Usage**

```r
stack
```

**Format**

An object of class `R6ClassGenerator` of length 24.

**Examples**

```r
x <- stack$new()
x$push(5)
y <- x$pop()
```
system

System class that manages overall game play

Usage

system

Format

An object of class R6ClassGenerator of length 24.

Examples

x <- system$new(nPlayers = 2, players = c("Michael", "Alex"))
Index

*Topic datasets
  bet, 2
  board, 2
  camel, 3
  die, 4
  overall.bet, 5
  player, 6
  space, 7
  stack, 7
  system, 8

bet, 2
board, 2

camel, 3
cleanColors, 3
die, 4
generateUI, 4
overall.bet, 5
playCamelUp, 5
player, 6

server, 6
space, 7
stack, 7
system, 8