Package 'thorn'

November 12, 2020

Type Package

Title 'HTMLwidgets' Displaying Some 'WebGL' Shaders

Version 0.2.0

Description Creates some 'WebGL' shaders. They can be used as the background of a 'Shiny' app. They also can be visualized in the 'RStudio' viewer pane or included in 'Rmd' documents, but this is pretty useless, besides contemplating them.

License GPL-3

Encoding UTF-8

LazyData true

Imports htmlwidgets

Suggests shiny, htmltools

URL https://github.com/stla/thorn

BugReports https://github.com/stla/thorn/issues

RoxygenNote 7.1.1

NeedsCompilation no

Author Stéphane Laurent [aut, cre],
Scott Boyle [ctb, cph] ('Hamster.js' library),
Mathew Groves [ctb, cph] ('PixiJS' library),
Chad Engler [ctb, cph] ('PixiJS' library)

Maintainer Stéphane Laurent <laurent_step@outlook.fr>

Repository CRAN

Date/Publication 2020-11-12 19:30:02 UTC

R topics documented:

thorn *HTML widget displaying a shader*

Description

Creates a HTML widget displaying a shader.

Usage

```
thorn(shader, width = NULL, height = NULL, elementId = NULL)
```

Arguments

shader	the name of the shader, one of "thorn", "thorn-color", "ikeda", "sweet", "biomorph1", "biomorph2", "biomorph3", "apollony", "smoke", "plasma"
width, height	a valid CSS measurement (like "100%", "400px", "auto") or a number, which will be coerced to a string and have "px" appended
elementId	a HTML id for the widget

Examples

```
library(thorn)
thorn("ikeda") # click on the shader to animate it
thorn("thorn") # you can also use the mouse wheel on this one

# four shaders ####
library(htmltools)

hw1 <- thorn("thorn-color", width = "50vw", height = "50vh")
hw2 <- thorn("ikeda", width = "50vw", height = "50vh")
hw3 <- thorn("sweet", width = "50vw", height = "50vh")
hw4 <- thorn("biomorph3", width = "50vw", height = "50vh")

if(interactive()){
  browsable(
    withTags(
      div(
        div(
          style = "position:absolute; top:0;",
          div(hw1, style="position:fixed; left:0;"),
          div(hw2, style="position:fixed; left:50vw;")
        ),
        div(
          style = "position:absolute; top:50vh;",
          div(hw3, style="position:fixed; left:0;"),
          div(hw4, style="position:fixed; left:50vw;")
        )
      )
    )
```

```
  )
}
```

---

thorn-shiny *Shiny bindings for* thorn

Description

Output and render functions for using thorn within Shiny applications and interactive Rmd documents.

Usage

```
thornOutput(outputId, width = "100%", height = "100%")

renderThorn(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

outputId	output variable to read from
width, height	a valid CSS measurement (like "100%", "400px", "auto") or a number, which will be coerced to a string and have "px" appended
expr	an expression that generates a shader created with thorn
env	the environment in which to evaluate expr
quoted	logical, whether expr is a quoted expression

Examples

```
# use a shader as the background of a Shiny app ####
library(thorn)
library(shiny)

ui <- fluidPage(
  thornOutput("thorn", width = "100%", height = "100%"),
  br(),
  sidebarLayout(
    sidebarPanel(
      sliderInput(
        "slider", "Slide me",
        value = 10, min = 0, max = 20
      ),
      selectInput(
        "select", "Select me", choices = c("Choice 1", "Choice 2")
      )
    ),
    mainPanel()
  )
)
```

```
server <- function(input, output){

  output[["thorn"]] <- renderThorn({
    thorn("biomorph2")
  })

}

if(interactive()){
  shinyApp(ui, server)
}

# all available shaders ####
library(thorn)
library(shiny)

ui <- fluidPage(
  br(),
  sidebarLayout(
    sidebarPanel(
      wellPanel(
        radioButtons(
          "shader", "Shader",
          choices = c(
            "thorn",
            "thorn-color",
            "ikeda",
            "biomorph1",
            "biomorph2",
            "biomorph3",
            "sweet",
            "apollony",
            "smoke"
          )
        )
      )
    ),
    mainPanel(
      thornOutput("shader", width = "calc(100% - 15px)", height = "400px")
    )
  )
)

server <- function(input, output){

  output[["shader"]] <- renderThorn({
    thorn(input[["shader"]])
  })

}

if(interactive()){
```

```
  shinyApp(ui, server)
}
```

Index