

Package ‘tree3d’

June 23, 2023

Type Package

Title 3D Tree Models

Version 0.1.2

Maintainer Tyler Morgan-Wall <tylermw@gmail.com>

Description Provides customizable 3D tree models (as 'OBJ' files) for use in data visualization. Includes both planar and solid tree models, various crown types (columnar, oval, palm, pyramidal, rounded, spreading, vase, weeping), and options to change the diameter, height, and color of the tree's crown and trunk.

License MIT + file LICENSE

Imports rayvertex (>= 0.7.8)

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 2.10)

URL <https://tylermorganwall.github.io/tree3d/>

BugReports <https://github.com/tylermorganwall/tree3d/issues>

NeedsCompilation no

Author Tyler Morgan-Wall [aut, cph, cre]

(<<https://orcid.org/0000-0002-3131-3814>>),

Samantha Miller [aut, cph] (<<https://orcid.org/0009-0001-1859-7146>>),

Agatha Czekajlo [aut, cph] (<<https://orcid.org/0000-0001-7266-8286>>)

Repository CRAN

Date/Publication 2023-06-23 08:40:02 UTC

R topics documented:

get_crown_file	2
get_flat_crown_file	3
get_solid_crown_file	3
get_tree_data	4
get_trunk_file	5

run_documentation	6
tree_mesh	6
write_tree_to_obj	16

Index	18
--------------	-----------

get_crown_file	<i>Get a 3D Crown File</i>
----------------	----------------------------

Description

Generate a specific 3D representation of a tree crown, given the parameters for the type of tree crown, the desired level of detail or resolution, and whether or not to offset the crown to align it with the trunk at the origin.

Usage

```
get_crown_file(
    tree = "oval",
    resolution = "medium",
    offset_origin = TRUE,
    solid = FALSE
)
```

Arguments

tree	Default "rounded". Crown type. Full list of options: "columnar" "pyramidal1" "pyramidal2" "oval" "palm" "rounded" "spreading1" "spreading2" "vase" "weeping"
resolution	Default "medium". Level of detail of the tree mesh. All options: "low" "medium" "high"
offset_origin	Default TRUE. Whether to offset the crown so that the bottom of the tree will be exactly at the origin, aligning it with the trunk.
solid	Default FALSE. Whether the crown should be a solid mesh (TRUE), or a collection of flat 2D planes (FALSE).

Value

Filename of OBJ file (as .txt file)

Examples

```
#Get a crown OBJ
get_crown_file(tree = "rounded", resolution = "high")
```

get_flat_crown_file *Get a Flat 3D Crown File*

Description

Generate a specific 3D representation of a planar tree crown, given the parameters for the type of tree crown, the desired level of detail or resolution, and whether or not to offset the crown to align it with the trunk at the origin.

Usage

```
get_flat_crown_file(  
    tree = "rounded",  
    resolution = "medium",  
    offset_origin = TRUE  
)
```

Arguments

tree	Default "rounded". Crown type. Full list of options: "columnar" "pyramidal1" "pyramidal2" "oval" "palm" "rounded" "spreading1" "spreading2" "vase" "weeping"
resolution	Default "medium". Level of detail of the tree mesh. All options: "low" "medium" "high"
offset_origin	Default TRUE. Whether to offset the crown so that the bottom of the tree will be exactly at the origin, aligning it with the trunk.

Value

Filename of OBJ file (as .txt file)

Examples

```
#Load an arrow OBJ  
get_flat_crown_file(tree = "rounded", resolution = "high")
```

get_solid_crown_file *Get a Solid 3D Crown File*

Description

Generate a specific 3D representation of a tree crown, given the parameters for the type of tree crown, the desired level of detail or resolution, and whether or not to offset the crown to align it with the trunk at the origin.

Usage

```
get_solid_crown_file(
  tree = "oval",
  resolution = "medium",
  offset_origin = TRUE
)
```

Arguments

tree	Default "oval". Crown type. Full list of options: "columnar" "pyramidal1" "pyramidal2" "oval" "spreading1" "weeping"
resolution	Default "medium". Level of detail of the tree mesh. All options: "low" "medium" "high"
offset_origin	Default TRUE. Whether to offset the crown so that the bottom of the tree will be exactly at the origin, aligning it with the trunk.

Value

Filename of OBJ file (as .txt file)

Examples

```
#Load an arrow OBJ
get_solid_crown_file(tree = "oval", resolution = "high")
```

get_tree_data

Get Tree Data

Description

This function retrieves a data frame of tree default data.

Usage

```
get_tree_data(tree_name = NULL)
```

Arguments

tree_name	Default NULL. The tree name, if only a single tree's info is needed. Otherwise, all trees are returned.
-----------	---

Value

Returns a data frame.

Examples

```
# Fetch the tree data frame.
get_tree_data()

# Get single tree
get_tree_data("palm")
```

get_trunk_file	<i>Get Trunk File</i>
----------------	-----------------------

Description

This function retrieves a trunk OBJ file. The trunk OBJ file can be used to create a tree trunk in a 3D scene. The function has an option to scale the trunk model based on the radius.

Usage

```
get_trunk_file(scaled = TRUE, offset = TRUE)
```

Arguments

scaled	Default TRUE. This parameter determines whether the trunk model should be scaled from a radius of 0.1 to 1. If TRUE, the trunk model will be scaled; otherwise, it will use the original size.
offset	Default TRUE. This parameter determines whether the trunk model should be offset to start at the origin, or centered at the origin (FALSE).

Value

Returns the filename of the selected OBJ file (as a .txt file) for the tree trunk. This file can be used to create tree trunk models in a 3D scene.

Examples

```
# Fetch an trunk OBJ path
get_trunk_file()
```

run_documentation *Run Documentation*

Description

This function determines if the examples are being run in pkgdown. It is not meant to be called by the user.

Usage

```
run_documentation()
```

Value

Boolean value.

Examples

```
# See if the documentation should be run.  
run_documentation()
```

tree_mesh *Generate a Tree Mesh*

Description

This function generates a tree model and transforms it into a ray_mesh object or an OBJ file. The tree model consists of two main parts: the crown (leafy part) and the trunk. Both parts can be customized according to a range of preset types, as well as full control over the tree's colors and dimensions. The tree model can be positioned, scaled, and rotated in 3D space.

Usage

```
tree_mesh(  
  crown_type = "oval",  
  position = c(0, 0, 0),  
  angle = 0,  
  solid = FALSE,  
  resolution = "medium",  
  filename = NULL,  
  tree_height = 1,  
  trunk_height_ratio = NULL,  
  crown_width_ratio = 1,  
  crown_height = NULL,  
  crown_width = NULL,  
  trunk_height = NULL,  
)
```

tree_mesh


7

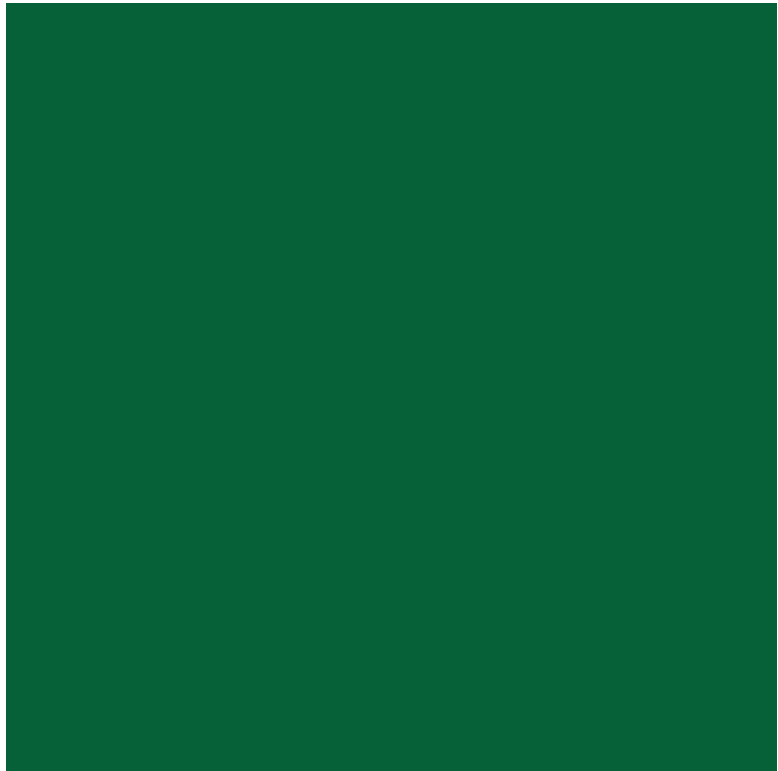
```
trunk_width = NULL,  
crown_color = NA,  
trunk_color = "#8C6F5B",  
diffuse_intensity = 1,  
ambient_intensity = 0.2  
)
```

Arguments

crown_type Default "oval". Crown type (the leafy part of the tree). Full list of options:

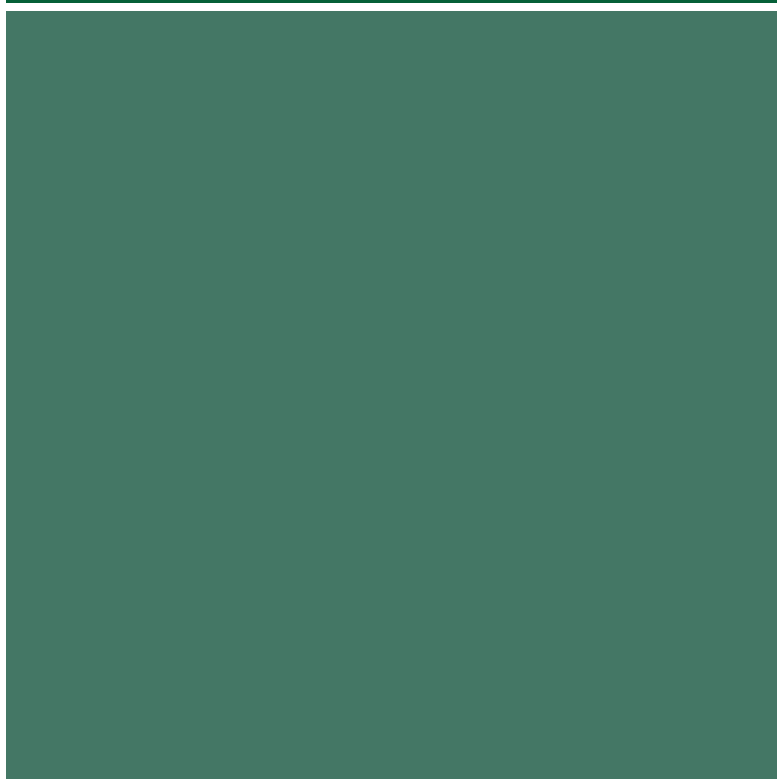
3D/Flat	Name	Crown Color	Trunk R
Both	"columnar"	"#A2C683"	





Both "pyramidal1" "#066038"

1/6



Both "pyramidal2" "#447765"

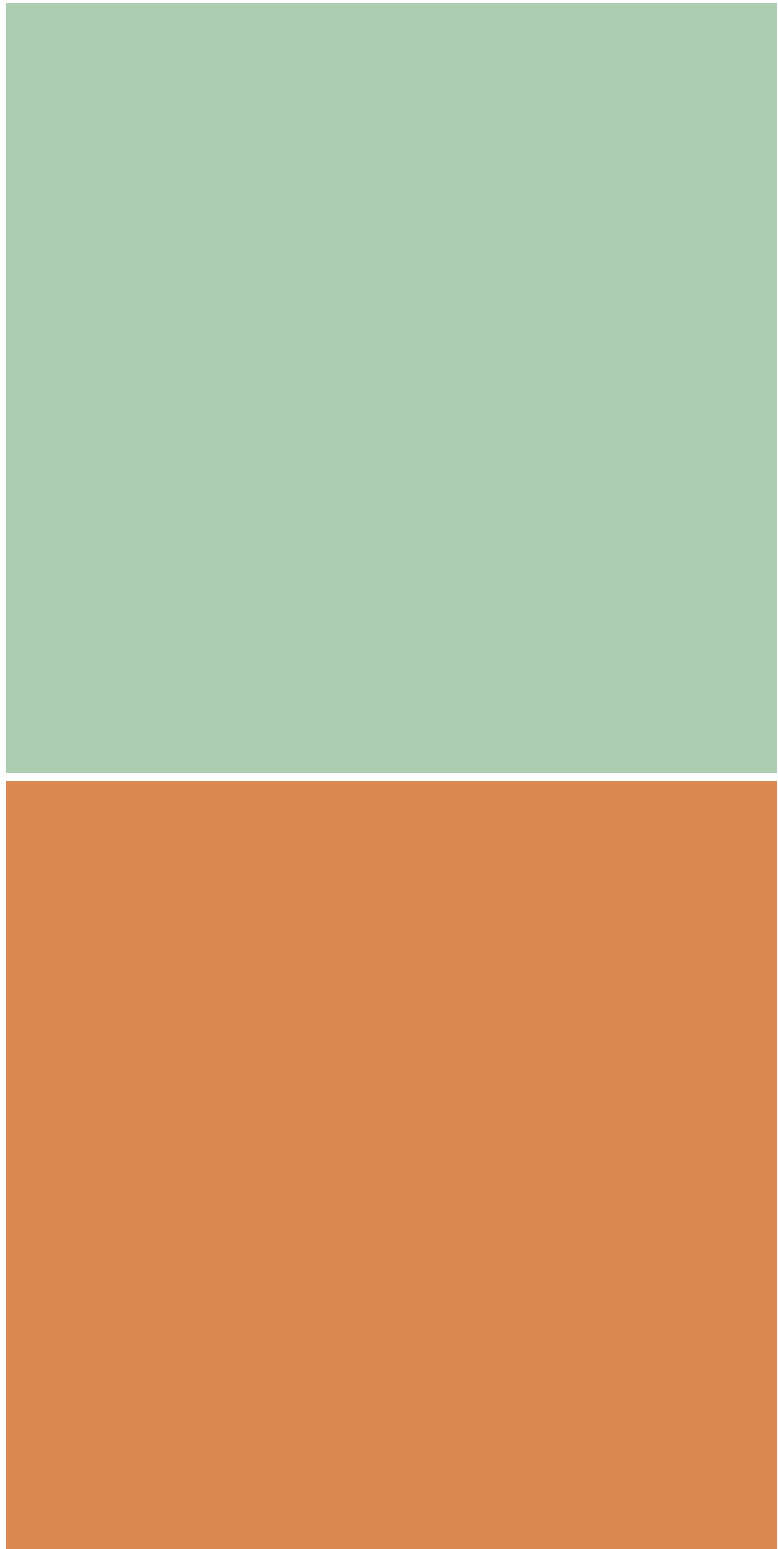
1/6

Both "weeping" "#CBD362"

1/3

Both "spreading1" "#CCB471"

1/3



10

tree_mesh

Both

"oval"

"#7CB262"

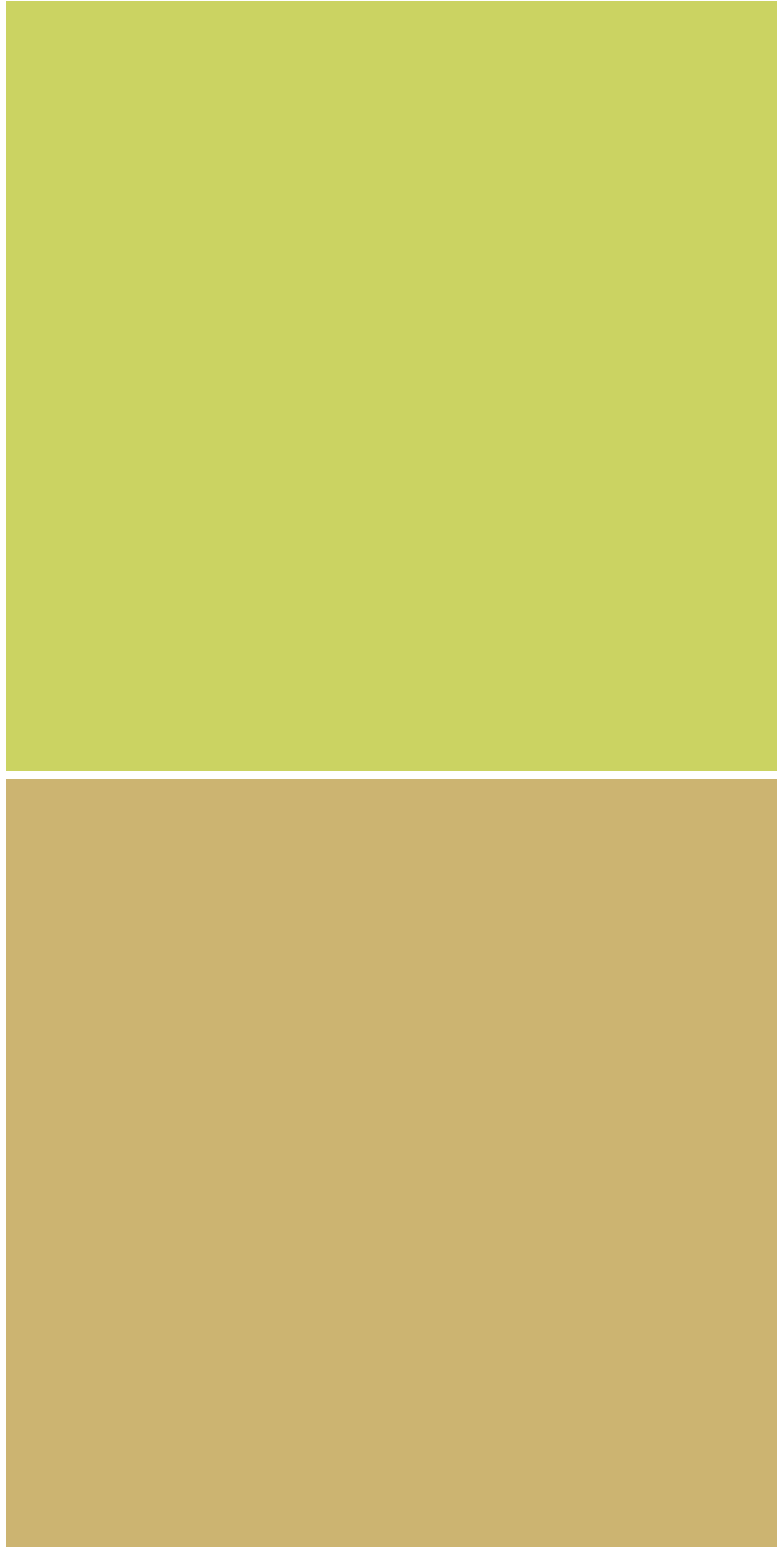
1/3

Flat-only

"palm"

"#DB8952"

1/2

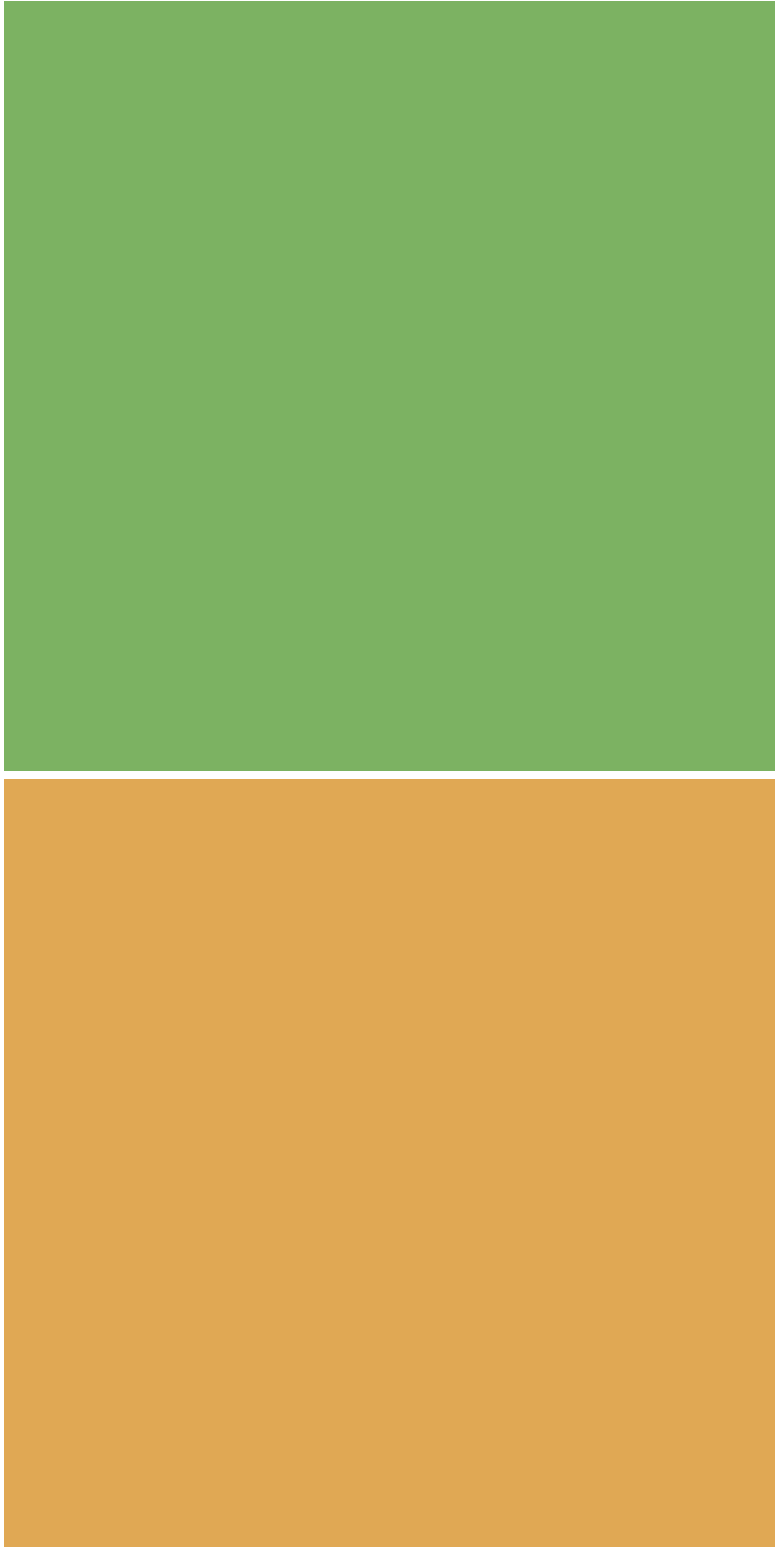


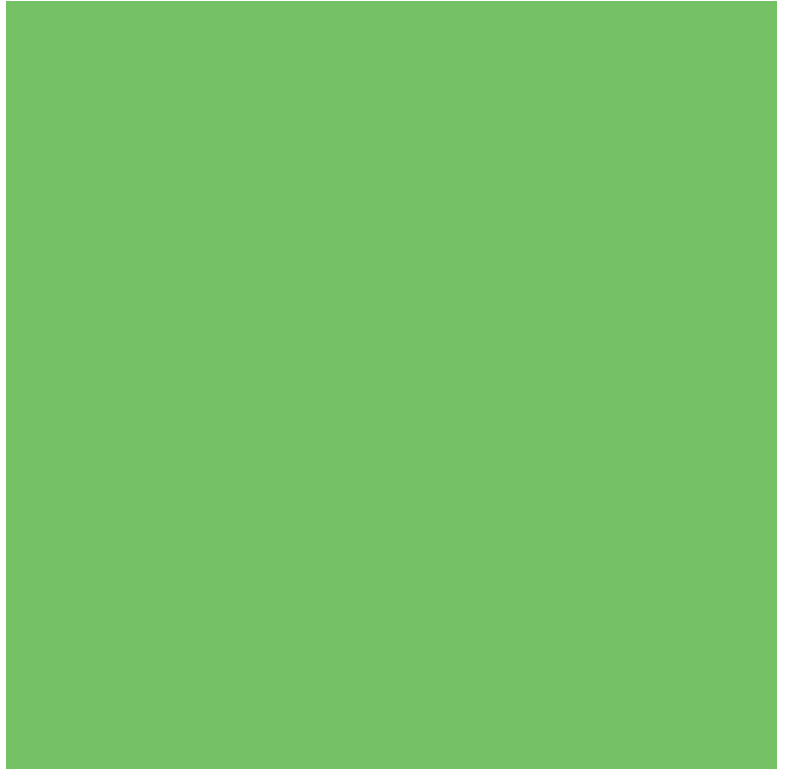
Flat-only "rounded" "#E0A854"

1/3

Flat-only "spreading2" "#75C165"

1/3





Flat-only "vase" "#AECCB1"

1/3

position	Default $c(0, 0, 0)$. A length-3 numeric vector specifying the X, Y, and Z coordinates of the tree mesh in 3D space.
angle	Default 0. Amount of rotation around the y-axis for the tree.
solid	Default FALSE. Whether the crown should be a solid mesh (TRUE), or a collection of flat 2D planes (FALSE).
resolution	Default "medium". String indicating the level of detail of the tree mesh. All options: "low" "medium" "high"
filename	Default NULL. File name of the OBJ file, if saving the mesh to a local file.
tree_height	Default 1. A numeric value setting the total height of the tree.
trunk_height_ratio	Default NULL. A numeric value specifying the ratio of the trunk height to the total height of the tree. If not provided, default values for each tree type will be used.
crown_width_ratio	Default 1. A numeric value specifying the ratio of the crown width to the crown height.
crown_height	Default NULL. A numeric value setting the height of the crown. If not provided, it is calculated based on the tree height and trunk height ratio.
crown_width	Default NULL. A numeric value setting the diameter of the crown. If not provided, it is calculated based on the crown height and crown width ratio.

trunk_height	Default NULL. A numeric value setting the height of the trunk. If not provided, it is calculated based on the tree height and trunk height ratio.
trunk_width	Default NULL. A numeric value setting the diameter of the trunk. If not provided, this is set to 1/10th the crown width.
crown_color	Default NA, use default for crown type. A string specifying the hex code of the crown color.
trunk_color	Default "#8C6F5B". A string specifying the hex code of the trunk color.
diffuse_intensity	Default 1.0. A numeric value controlling the amount of diffuse (shaded) color included in the model.
ambient_intensity	Default 0.2. A numeric value controlling the amount of ambient (constant) color included in the model.

Value

ray_mesh list object

Examples

```
#Load a tree and render it
library(rayvertex)
render_tree_example = function(example_tree_mesh) {
  example_tree_mesh |>
    add_shape(xz_rect_mesh(c(0,0,-9),
                          material = material_list(diffuse="tan",
                                                    ambient = "grey", diffuse_intensity = 0.7,
                                                    ambient_intensity = 0.6),
                          scale=25)) |>
    rasterize_scene(lookat=c(0,0.5,0),
                   light_info = directional_light(c(0.3,1,1), intensity = 0.7),
                   lookfrom=c(0,3,10),
                   fov=8,
                   shadow_map_dims = 2, shadow_map_bias = 0.0005,
                   width = 800, height = 800,
                   #fsaa = 4, ssao = TRUE, ssao_intensity = 1,
                   background = "lightblue")
}
if(run_documentation()) {
  #Render a basic 3D crown
  tree_mesh("columnar",
            solid = TRUE,
            ambient_intensity = 0.3) |>
  render_tree_example()
}
if(run_documentation()) {
  #Render the 2D planar version
  tree_mesh("columnar",
            solid = FALSE,
            ambient_intensity = 0.3) |>
```

```
    render_tree_example()
  }
  if(run_documentation()) {
#Adjust the trunk/crown proportions of the overall tree using ratios
tree_mesh("columnar",
          tree_height = 1,
          trunk_height_ratio = 1/5,
          crown_width_ratio = 1.25,
          trunk_width = 0.125,
          solid = TRUE,
          ambient_intensity = 0.3) |>
    render_tree_example()

  }

  if(run_documentation()) {
#The crown width is set proportional to the crown height (not the overall tree)
tree_mesh("columnar",
          tree_height = 1,
          trunk_height_ratio = 2/3,
          crown_width_ratio = 1,
          trunk_width = 0.075,
          solid = TRUE,
          ambient_intensity = 0.3) |>
    render_tree_example()
  }

  if(run_documentation()) {
#Set the trunk and crown dimensions directly
tree_mesh("columnar",
          trunk_height = 0.25,
          trunk_width = 0.05,
          crown_height = 0.75,
          crown_width = 0.33,
          solid = TRUE,
          ambient_intensity = 0.3) |>
    render_tree_example()
  }

  if(run_documentation()) {
#Change the crown and trunk color
tree_mesh("columnar",
          solid = TRUE,
          crown_color = "orange",
          trunk_color = "tan",
          ambient_intensity = 0.3) |>
    render_tree_example()
  }

  if(run_documentation()) {
#Render different tree types, both 2D and 3D versions
tree_mesh("columnar",
          solid = TRUE,
          ambient_intensity = 0.3) |>
    render_tree_example()
  }
}
```

```
}
if(run_documentation()) {
tree_mesh("columnar",
          solid = FALSE,
          ambient_intensity = 0.3) |>
  render_tree_example()
}
if(run_documentation()) {
tree_mesh("pyramidal1",
          solid = TRUE,
          ambient_intensity = 0.3) |>
  render_tree_example()
}
if(run_documentation()) {
tree_mesh("pyramidal1",
          solid = FALSE,
          ambient_intensity = 0.3) |>
  render_tree_example()
}
if(run_documentation()) {
tree_mesh("pyramidal2",
          solid = TRUE,
          ambient_intensity = 0.3) |>
  render_tree_example()
}
if(run_documentation()) {
tree_mesh("pyramidal2",
          solid = FALSE,
          ambient_intensity = 0.3) |>
  render_tree_example()
}
if(run_documentation()) {
tree_mesh("weeping",
          solid = TRUE,
          ambient_intensity = 0.3) |>
  render_tree_example()
}
if(run_documentation()) {
tree_mesh("weeping",
          solid = FALSE,
          ambient_intensity = 0.3) |>
  render_tree_example()
}
if(run_documentation()) {
tree_mesh("spreading1",
          solid = TRUE,
          ambient_intensity = 0.3) |>
  render_tree_example()
}
if(run_documentation()) {
tree_mesh("spreading1",
          solid = FALSE,
          ambient_intensity = 0.3) |>
```

```

    render_tree_example()
}
if(run_documentation()) {
tree_mesh("oval",
          solid = TRUE,
          ambient_intensity = 0.3) |>
  render_tree_example()
}
if(run_documentation()) {
tree_mesh("oval",
          solid = FALSE,
          ambient_intensity = 0.3) |>
  render_tree_example()
}
if(run_documentation()) {
tree_mesh("palm",
          solid = FALSE,
          ambient_intensity = 0.3) |>
  render_tree_example()
}
if(run_documentation()) {
tree_mesh("rounded",
          solid = FALSE,
          ambient_intensity = 0.3) |>
  render_tree_example()
}
if(run_documentation()) {
tree_mesh("spreading2",
          solid = FALSE,
          ambient_intensity = 0.3) |>
  render_tree_example()
}
if(run_documentation()) {
tree_mesh("vase",
          solid = FALSE,
          ambient_intensity = 0.3) |>
  render_tree_example()
}

```

write_tree_to_obj

Write Tree to OBJ

Description

Export the tree mesh to an OBJ file for use in other 3D programs

Usage

```
write_tree_to_obj(tree_mesh, filename, materials = TRUE, fileext = ".obj")
```


Arguments

<code>tree_mesh</code>	Tree mesh.
<code>filename</code>	Filename for the OBJ file. If the file extension is not included, it will automatically be added.
<code>materials</code>	Default TRUE. Whether to also write the material (MTL) file.
<code>fileext</code>	Default ".obj". File extension to be added if not present.

Value

None

Examples

```
#Save tree mesh to file
tempfileloc = tempfile()
write_tree_to_obj(tree_mesh(), filename = tempfileloc)
```

Index

[get_crown_file](#), 2
[get_flat_crown_file](#), 3
[get_solid_crown_file](#), 3
[get_tree_data](#), 4
[get_trunk_file](#), 5

[run_documentation](#), 6

[tree_mesh](#), 6

[write_tree_to_obj](#), 16