OoD solver

1.0

Generated by Doxygen 1.8.12

Contents

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 pkg Namespace Reference

Namespaces

- isdh
- read_xml
- structure_core
- tree_core
- write_xml

4.2 pkg.isdh Namespace Reference

Namespaces

- component
- deformation_step
- example_main
- isdh_helper

4.3 pkg.isdh.component Namespace Reference

Classes

- class Component

4.4 pkg.isdh.deformation_step Namespace Reference

Classes

- class DeformationStep

Functions

- def __repr__ (self)

  LEGEND:

  'd': the element deforms

  'm': the element moves

  'b': the element breaks
- def __eq__ (self, other)
- def __ne__ (self, other)
- def print (self)

Variables

- transformation

4.4.1 Function Documentation

4.4.1.1 __eq__()

```
def pkg.isdh.deformation_step.__eq__ (
              self,
              other )

Return self == other.

Args:
    other:
        DeformationStep object.
Returns:
    True or False.
Raises:
    NotImplementedError is raised, if other is not a DeformationStep
    object.
```

Definition at line 24 of file deformation_step.py.

4.4.1.2 __ne__()

```
def pkg.isdh.deformation_step.__ne__ (
              self,
              other )

Return self != other.

Args:
    other:
        DeformationStep object.
Returns:
    True or False.
Raises:
    NotImplementedError is raised, if other is not a DeformationStep
    object.
```

Definition at line 40 of file deformation_step.py.

4.4.1.3 __repr__()

```
def pkg.isdh.deformation_step.__repr__ (
              self )
```

LEGEND:

'd': the element deforms

'm': the element moves

'b': the element breaks

```
Return the string representation of the object.

Args:
    nothing is taken
Returns:
    string
Raises:
    nothing is raised
```

Definition at line 12 of file deformation_step.py.

4.4.1.4 print()

```
def pkg.isdh.deformation_step.print (
              self )
```

```
Print a detailed description of the object.

Args:
    nothing is taken.
Returns:
    nothing is returned.
Raises:
    nothing is raised.
```

Definition at line 57 of file deformation_step.py.

4.4.2 Variable Documentation

4.4.2.1 transformation

```
pkg.isdh.deformation_step.transformation
```

Definition at line 67 of file deformation_step.py.

4.5 pkg.isdh.example_main Namespace Reference

Variables

- structure = s.read_xml('some_folder/my_xml_file.xml')
- int n = 0

  solve the structure
- frames_per_mm

4.5.1 Variable Documentation

4.5.1.1 frames_per_mm

```
pkg.isdh.example_main.frames_per_mm
```

Definition at line 49 of file example_main.py.

4.5.1.2 n

```
int pkg.isdh.example_main.n = 0
```

solve the structure

- * * * * * . . . . . * * * * * * . . . . . * * * * * * . . . . . * * * * * REMARK the isdh module works as an interface between the solver_module and the animation_module. No matter how structure.solve() works: the output is defined so that it can be used as an input for a.create_video()
- * * * * * . . . . . * * * * * * . . . . . * * * * * * . . . . . * * * * * create the video

Definition at line 48 of file example_main.py.

4.5.1.3 structure

```
pkg.isdh.example_main.structure = s.read_xml('some_folder/my_xml_file.xml')
```

- * * * * * . . . . . * * * * * * . . . . . * * * * * * . . . . . * * * * * REMARK the solver_module has to implement:
- a function called read_xml() that takes an .xml file as input and outputs an object of a class
- a method of that class called solve() that doesn't take anything as input and outputs the 2 lists of isdh objects defined below
- * * * * * . . . . . * * * * * * . . . . . * * * * * * . . . . . * * * * *

Definition at line 17 of file example_main.py.

4.6 pkg.isdh.isdh_helper Namespace Reference

Classes

- class IsdhHelper

4.7 pkg.read_xml Namespace Reference

Functions

- def read_xml (path)
- def add_nodes (root, struct)
- def add_components (root, struct)
- def gaps_insertor (structure)
- def gap_name (level)
- def init_firewall_and_barrier (structure)

Variables

- node

  add gaps in front of the loadpath
- frontNode
- gap = Component. \
- name = gap_name(loadpath.level)

  add gaps between components create a gap_name iterator (e.g.
- listNodes = list(loadpath.setNodes)
- key
- nodes
- next_nodes
- ignore_me = next(next_nodes)
- list lp_levels
- rightLimit
- backNode

4.7.1 Function Documentation

4.7.1.1 add_components()

```
def pkg.read_xml.add_components (
              root,
              struct )

Add components to the structure, given the root of the tree to parse.

Every compoenent is added to the correct loadpath defined in
struct.listLoadpaths. The loadpath must already contain the left and the
right node of each component.

Args:
  root:
    the root of the tree to parse given by the ElementTree.getroot()
    function.
  struct:
    the structure_core.structure.Structure obj
Returns:
  nothing is returned.
Raises:
  nothing is raised.
```

Definition at line 85 of file read_xml.py.

4.7.1.2 add_nodes()

```
def pkg.read_xml.add_nodes (
              root,
              struct )
```

```
Add nodes to the structure, given the root of the tree to parse.

Every node is added to the correct loadpath defined in
struct.listLoadpaths.

Args:
  root:
    the root of the tree to parse given by the ElementTree.getroot()
    function.
  struct:
    the structure_core.structure.Structure obj
Returns:
  nothing is returned.
Raises:
  nothing is raised.
```

Definition at line 36 of file read_xml.py.

4.7.1.3 gap_name()

```
def pkg.read_xml.gap_name (
              level )
```

```
Yields a string, to be used as a name for the next gap.

Args:
  level:
    an integer, the level of the loadpath to which the gap belongs.
Returns:
  a string
Raises:
  nothing is raised.
```

Definition at line 261 of file read_xml.py.

4.7.1.4 gaps_insertor()

```
def pkg.read_xml.gaps_insertor (
              structure )
```

```
Add gaps to the structure.

Every gap is added (as a component) to the correct loadpath defined in
struct.listLoadpaths. The loadpath must already contain all the nodes and
all the components.
Gaps are added where needed:
- in front of loadpaths that are not directly connected to the barrier
- between non-adjacent components
- behind loadpaths that are not directly connected to the firewall

        gap                 gap                 gap
|xx|            o-----o            o-----o            |x|
```

```
barrier                                                 firewall

Args:
  struct:
    the structure_core.structure.Structure obj
Returns:
  nothing is returned.
Raises:
  nothing is raised.
```

Definition at line 158 of file read_xml.py.

4.7.1.5 init_firewall_and_barrier()

```
def pkg.read_xml.init_firewall_and_barrier (
              structure )
```

```
Initialise the connections of components, cross components and nodes.

The attributes .connectedToBarrier and .connectedToFirewall of every
structure_core.Component and structure_core.CrossComponent object are
initialised.
The attributes .onBarrier and .onFirewall of every structure_core.Node
object are initialised.

Args:
    structure:
        structure_core.structure.Structure object that groups components,
        cross components and nodes to initialise.
Returns:
    nothing is returned.
Raises:
    nothing is raised.
```

Definition at line 277 of file read_xml.py.

4.7.1.6 read_xml()

```
def pkg.read_xml.read_xml (
              path )
```

```
Return a Structure object, based on the .xml at the given path.

Args:
  path:
    a string, that contains the path to the .xml file
Returns:
  an object of the class structure_core.structure.Structure.
Raises:
  nothing is raised.
```

Definition at line 9 of file read_xml.py.

4.7.2 Variable Documentation

4.7.2.1 backNode

```
pkg.read_xml.backNode
```

Initial value:

```
1 =  Node.Node(rightLimit,
2                                node.loadpathLevel)
```

Definition at line 248 of file read_xml.py.

4.7.2.2 frontNode

```
pkg.read_xml.frontNode
```

Initial value:

```
1 =  Node.Node(leftLimit,             # position
2                                node.loadpathLevel)
```

Definition at line 193 of file read_xml.py.

4.7.2.3 gap

```
pkg.read_xml.gap = Component.  \
```

Definition at line 196 of file read_xml.py.

4.7.2.4 ignore_me

```
pkg.read_xml.ignore_me = next(next_nodes)
```

Definition at line 214 of file read_xml.py.

4.7.2.5 key

```
pkg.read_xml.key
```

Definition at line 210 of file read_xml.py.

4.7.2.6 listNodes

```
pkg.read_xml.listNodes = list(loadpath.setNodes)
```

Definition at line 209 of file read_xml.py.

4.7.2.7 lp_levels

```
list pkg.read_xml.lp_levels
```

Initial value:

```
1 =  [crossComp.rightNode.loadpathLevel
2                         for crossComp in node.towardsFirewall]
```

Definition at line 241 of file read_xml.py.

4.7.2.8 name

```
pkg.read_xml.name = gap_name(loadpath.level)
```

add gaps between components create a gap_name iterator (e.g.

'gap-0-1', 'gap-0-2', ...)

Definition at line 207 of file read_xml.py.

4.7.2.9 next_nodes

```
pkg.read_xml.next_nodes
```

Definition at line 212 of file read_xml.py.

4.7.2.10 node

```
pkg.read_xml.node
```

Initial value:

```
1 =  min((node
2                     for node in loadpath.setNodes),
3                    key = lambda node: node.position)
```

add gaps in front of the loadpath

add gaps behind the loadpath

Definition at line 189 of file read_xml.py.

4.7.2.11 nodes

```
pkg.read_xml.nodes
```

Definition at line 212 of file read_xml.py.

4.7.2.12 rightLimit

```
pkg.read_xml.rightLimit
```

Initial value:

```
1 =  max(comp.rightNode.position
2                              for lp in structure.listLoadpaths
3                              if lp.level in lp_levels
4                              for comp in lp.listComponents)
```

Definition at line 244 of file read_xml.py.

4.8 pkg.structure_core Namespace Reference

Namespaces

- Component
- CrossComponent
- Loadpath
- Node
- Structure

4.9 pkg.structure_core.Component Namespace Reference

Classes

- class Component

Variables

- logger = logging.getLogger('component')
- level
- tuple BLACK = (0, 0, 0)
- tuple WHITE = (255, 255, 255)
- tuple RED = (255, 0, 0)
- tuple GREEN = (0, 255, 0)
- tuple BLUE = (0, 0, 255)
- tuple LIGHT_BLUE = (102, 255, 255)
- tuple DARK_GREEN = (0, 100, 0)

4.9.1 Variable Documentation

4.9.1.1 BLACK

```
tuple pkg.structure_core.Component.BLACK = ( 0, 0, 0)
```

Definition at line 12 of file Component.py.

4.9.1.2 BLUE

```
tuple pkg.structure_core.Component.BLUE = ( 0, 0, 255)
```

Definition at line 16 of file Component.py.

4.9.1.3 DARK_GREEN

```
tuple pkg.structure_core.Component.DARK_GREEN = ( 0, 100, 0)
```

Definition at line 18 of file Component.py.

4.9.1.4 GREEN

```
tuple pkg.structure_core.Component.GREEN = ( 0, 255, 0)
```

Definition at line 15 of file Component.py.

4.9.1.5 level

```
pkg.structure_core.Component.level
```

Definition at line 4 of file Component.py.

4.9.1.6 LIGHT_BLUE

```
tuple pkg.structure_core.Component.LIGHT_BLUE = (102, 255, 255)
```

Definition at line 17 of file Component.py.

4.9.1.7 logger

```
pkg.structure_core.Component.logger = logging.getLogger('component')
```

Definition at line 3 of file Component.py.

4.9.1.8 RED

```
tuple pkg.structure_core.Component.RED = (255, 0, 0)
```

Definition at line 14 of file Component.py.

4.9.1.9 WHITE

```
tuple pkg.structure_core.Component.WHITE = (255, 255, 255)
```

Definition at line 13 of file Component.py.

4.10 pkg.structure_core.CrossComponent Namespace Reference

Classes

- class CrossComponent

Variables

- logger = logging.getLogger('CrossComponentLogger')
- tuple BLACK = (0, 0, 0)
- tuple WHITE = (255, 255, 255)
- tuple RED = (255, 0, 0)
- tuple GREEN = (0, 255, 0)
- tuple BLUE = (0, 0, 255)
- tuple DARK_GREEN = (0, 100, 0)

4.10.1 Variable Documentation

4.10.1.1 BLACK

```
tuple pkg.structure_core.CrossComponent.BLACK = ( 0, 0, 0)
```

Definition at line 10 of file CrossComponent.py.

4.10.1.2 BLUE

```
tuple pkg.structure_core.CrossComponent.BLUE = ( 0, 0, 255)
```

Definition at line 14 of file CrossComponent.py.

4.10.1.3 DARK_GREEN

```
tuple pkg.structure_core.CrossComponent.DARK_GREEN = ( 0, 100, 0)
```

Definition at line 15 of file CrossComponent.py.

4.10.1.4 GREEN

```
tuple pkg.structure_core.CrossComponent.GREEN = ( 0, 255, 0)
```

Definition at line 13 of file CrossComponent.py.

4.10.1.5 logger

```
pkg.structure_core.CrossComponent.logger = logging.getLogger('CrossComponentLogger')
```

Definition at line 2 of file CrossComponent.py.

4.10.1.6 RED

```
tuple pkg.structure_core.CrossComponent.RED = (255, 0, 0)
```

Definition at line 12 of file CrossComponent.py.

4.10.1.7 WHITE

```
tuple pkg.structure_core.CrossComponent.WHITE = (255, 255, 255)
```

Definition at line 11 of file CrossComponent.py.

4.11 pkg.structure_core.Loadpath Namespace Reference

Classes

- class Loadpath

Functions

- def valid_components (self)

  def add_member(self, component): """Function addes components to the list of components """ self.list↩ Components.append(component)

4.11.1 Function Documentation

4.11.1.1 valid_components()

```
def pkg.structure_core.Loadpath.valid_components (
              self )
```

def add_member(self, component): """Function addes components to the list of components """ self.list↩ Components.append(component)

```
Returns a list with the current components allowed to deform.

Args:
  nothing is taken
Returns:
  nothing is returned
Raises:
  nothing is raised.
```

Definition at line 26 of file Loadpath.py.

4.12 pkg.structure_core.Node Namespace Reference

Classes

- class Node

Variables

- tuple BLACK = (0, 0, 0)
- tuple RED = (255, 0, 0)
- tuple DARK_GREEN = (0, 100, 0)
- logger = logging.getLogger('node')
- level

4.12.1 Variable Documentation

4.12.1.1 BLACK

```
tuple pkg.structure_core.Node.BLACK = ( 0, 0, 0)
```

Definition at line 7 of file Node.py.

4.12.1.2 DARK_GREEN

```
tuple pkg.structure_core.Node.DARK_GREEN = ( 0, 100, 0)
```

Definition at line 9 of file Node.py.

4.12.1.3 level

```
pkg.structure_core.Node.level
```

Definition at line 12 of file Node.py.

4.12.1.4 logger

```
pkg.structure_core.Node.logger = logging.getLogger('node')
```

Definition at line 11 of file Node.py.

4.12.1.5 RED

```
tuple pkg.structure_core.Node.RED = (255, 0, 0)
```

Definition at line 8 of file Node.py.

4.13 pkg.structure_core.Structure Namespace Reference

Classes

- class Structure

Variables

- bool DEBUG = False

  import itertools from ..tree_core.tree import Tree from .
- bool STEPWISE = False
- list size = [1500, 500]
- screen = pygame.display.set_mode(size)
- tuple BLACK = (0, 0, 0)
- tuple WHITE = (255, 255, 255)
- tuple RED = (255, 0, 0)
- tuple GREEN = (0, 255, 0)
- tuple BLUE = (0, 0, 255)

4.13.1 Variable Documentation

4.13.1.1 BLACK

```
tuple pkg.structure_core.Structure.BLACK = ( 0, 0, 0)
```

Definition at line 14 of file Structure.py.

4.13.1.2 BLUE

```
tuple pkg.structure_core.Structure.BLUE = ( 0, 0, 255)
```

Definition at line 18 of file Structure.py.

4.13.1.3 DEBUG

```
bool pkg.structure_core.Structure.DEBUG = False
```

import itertools from ..tree_core.tree import Tree from .

. import GapsHandeling debugging purpose

Definition at line 6 of file Structure.py.

4.13.1.4 GREEN

```
tuple pkg.structure_core.Structure.GREEN = ( 0, 255, 0)
```

Definition at line 17 of file Structure.py.

4.13.1.5 RED

```
tuple pkg.structure_core.Structure.RED = (255, 0, 0)
```

Definition at line 16 of file Structure.py.

4.13.1.6 screen

```
pkg.structure_core.Structure.screen = pygame.display.set_mode(size)
```

Definition at line 12 of file Structure.py.

4.13.1.7 size

```
list pkg.structure_core.Structure.size = [1500, 500]
```

Definition at line 11 of file Structure.py.

4.13.1.8 STEPWISE

```
bool pkg.structure_core.Structure.STEPWISE = False
```

Definition at line 7 of file Structure.py.

4.13.1.9 WHITE

```
tuple pkg.structure_core.Structure.WHITE = (255, 255, 255)
```

Definition at line 15 of file Structure.py.

4.14 pkg.tree_core Namespace Reference

Namespaces

- NodeTree
- tree

4.15 pkg.tree_core.NodeTree Namespace Reference

Classes

- class NodeTree

4.16 pkg.tree_core.tree Namespace Reference

Classes

- class Tree

Variables

- bool PRINT = False

4.16.1 Variable Documentation

4.16.1.1 PRINT

```
bool pkg.tree_core.tree.PRINT = False
```

Definition at line 3 of file tree.py.

4.17 pkg.write_xml Namespace Reference

Functions

- def create_level (root, index)
- def create_component (level, input_name, input_x1, input_x2, input_defoLength, input_end_lp=None)
- def ask_for_new_level (root)
- def ask_for_new_member (level, lp_i)
- def ask_for_new_connection (level, lp_i)
- def prettify (path)
- def new_xml ()

4.17.1 Function Documentation

4.17.1.1 ask_for_new_connection()

```
def pkg.write_xml.ask_for_new_connection (
              level,
              lp_i )
```

Definition at line 69 of file write_xml.py.

4.17.1.2 ask_for_new_level()

```
def pkg.write_xml.ask_for_new_level (
              root )
```

Definition at line 36 of file write_xml.py.

4.17.1.3 ask_for_new_member()

```
def pkg.write_xml.ask_for_new_member (
              level,
              lp_i )
```

Definition at line 48 of file write_xml.py.

4.17.1.4 create_component()

```
def pkg.write_xml.create_component (
              level,
              input_name,
              input_x1,
              input_x2,
              input_defoLength,
              input_end_lp = None )
```

Definition at line 15 of file write_xml.py.

4.17.1.5 create_level()

```
def pkg.write_xml.create_level (
              root,
              index )
```

Definition at line 5 of file write_xml.py.

4.17.1.6 new_xml()

```
def pkg.write_xml.new_xml ( )
```

Definition at line 98 of file write_xml.py.

4.17.1.7 prettify()

```
def pkg.write_xml.prettify (
              path )
```

Definition at line 91 of file write_xml.py.

4.18 Visualization Namespace Reference

Namespaces

- BlenderObject
- CreateVideo
- initialization
- Member
- setCamera
- setColor
- setFunction
- setLamp
- setRender

4.19 Visualization.BlenderObject Namespace Reference

Classes

- class BlenderObject

Variables

- meshElement = bpy.ops.mesh.primitive_cube_add
- meshMass = bpy.ops.mesh.primitive_uv_sphere_add
- meshText = bpy.ops.object.text_add

4.19.1 Variable Documentation

4.19.1.1 meshElement

```
Visualization.BlenderObject.meshElement = bpy.ops.mesh.primitive_cube_add
```

Definition at line 15 of file BlenderObject.py.

4.19.1.2 meshMass

```
Visualization.BlenderObject.meshMass = bpy.ops.mesh.primitive_uv_sphere_add
```

Definition at line 17 of file BlenderObject.py.

4.19.1.3 meshText

```
Visualization.BlenderObject.meshText = bpy.ops.object.text_add
```

Definition at line 19 of file BlenderObject.py.

4.20 Visualization.CreateVideo Namespace Reference

Classes

- class CreateVideo

4.21 Visualization.initialization Namespace Reference

Functions

- def initialize ()
- def static_numberOfElement ()

4.21.1 Function Documentation

4.21.1.1 initialize()

```
def Visualization.initialization.initialize ( )
```

```
Prepare Blender for anew animation
Args:
    nothing is taken

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 4 of file initialization.py.

4.21.1.2 static_numberOfElement()

```
def Visualization.initialization.static_numberOfElement ( )
```

Definition at line 40 of file initialization.py.

4.22 Visualization.Member Namespace Reference

Classes

- class generalMember

4.23 Visualization.setCamera Namespace Reference

Functions

- def setCamera (scene, x, y, z, lents)

4.23.1 Function Documentation

4.23.1.1 setCamera()

```
def Visualization.setCamera.setCamera (
              scene,
              x,
              y,
              z,
              lents )
```

```
Creates and configure the camera

Args:
    scene:
        bpy.context.scene, contains the scene of the video
    x:
        float, x coordinate of the position of the camera
    y:
        float, y coordinate of the position of the camera
    z:
        float, z coordinate of the position of the camera

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 3 of file setCamera.py.

4.24 Visualization.setColor Namespace Reference

Functions

- def setColor (ob, color)
- def makeColor (name, diffuse)

Variables

- def red = makeColor('Red', (1,0,0))
- def blue = makeColor('Blue', (0,0,1))
- def black = makeColor('Black',(0,0,0))
- def white = makeColor('White',(1,1,1))
- def green = makeColor('Green',(0,1,0))
- def gray = makeColor('Gray', (0.6,0.6,0.6))
- def dark_gray = makeColor('Gray', (0.2,0.2,0.2))

4.24.1 Function Documentation

4.24.1.1 makeColor()

```
def Visualization.setColor.makeColor (
              name,
              diffuse )


Creates color

Args:
    name:
        string, name of the color
    diffuse:
        value of difussion

Returns:
    new color

Raises:
    nothing is raised
```

Definition at line 21 of file setColor.py.

4.24.1.2 setColor()

```
def Visualization.setColor.setColor (
              ob,
              color )


Set the color of an object

Args:
    ob:
        blender object, the object to color
    color:
        color object, color

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 3 of file setColor.py.

4.24.2 Variable Documentation

4.24.2.1 black

```
def Visualization.setColor.black = makeColor('Black',(0,0,0))
```

Definition at line 47 of file setColor.py.

4.24.2.2 blue

```
def Visualization.setColor.blue = makeColor('Blue', (0,0,1))
```

Definition at line 46 of file setColor.py.

4.24.2.3 dark_gray

```
def Visualization.setColor.dark_gray = makeColor('Gray', (0.2,0.2,0.2))
```

Definition at line 51 of file setColor.py.

4.24.2.4 gray

```
def Visualization.setColor.gray = makeColor('Gray', (0.6,0.6,0.6))
```

Definition at line 50 of file setColor.py.

4.24.2.5 green

```
def Visualization.setColor.green = makeColor('Green',(0,1,0))
```

Definition at line 49 of file setColor.py.

4.24.2.6 red

```
def Visualization.setColor.red = makeColor('Red', (1,0,0))
```

Definition at line 45 of file setColor.py.

4.24.2.7 white

```
def Visualization.setColor.white = makeColor('White',(1,1,1))
```

Definition at line 48 of file setColor.py.

4.25 Visualization.setFunction Namespace Reference

Functions

- def movement (object, initialFrame, finalFrame, amount, offset=0)
- def rotation (object, initialFrame, finalFrame, amount)
- def deformation (object, initialFrame, finalFrame, amount)
- def color (object, initialFrame, color)
- def elimination (object, initialFrame, finalFrame)
- def interpolation (object)

4.25.1 Function Documentation

4.25.1.1 color()

```
def Visualization.setFunction.color (
              object,
              initialFrame,
              color )
```

```
Set the frames where the change of color takes place

Args:
    object:
        blender mesh, current object which will color
    initialFrame:
        integer, the starting frame of the color

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 95 of file setFunction.py.

4.25.1.2 deformation()

```
def Visualization.setFunction.deformation (
              object,
              initialFrame,
              finalFrame,
              amount )
```

```
Set the frames where the deformation takes place

Args:
    object:
        blender mesh, current object which will deform
    initialFrame:
        integer, the starting frame of the deformation
    finalFrame:
        integer, the final frame of the deformation
    amount:
        float, amount of deformation

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 66 of file setFunction.py.

4.25.1.3 elimination()

```
def Visualization.setFunction.elimination (
              object,
              initialFrame,
              finalFrame )


Set the frames where the removal of the tag takes place

Args:
    object:
        blender mesh, current object which will vanish
    initialFrame:
        integer, the starting frame of the vanishing
    finalFrame:
        integer, the final frame of the vanishing

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 120 of file setFunction.py.

4.25.1.4 interpolation()

```
def Visualization.setFunction.interpolation (
              object )


Linear interpolation
This interpolation makes the deformation and the movement smooth
without any kind of acceleration

Args:
    object:
        blender mesh, current object

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 147 of file setFunction.py.

4.25.1.5 movement()

```
def Visualization.setFunction.movement (
              object,
              initialFrame,
              finalFrame,
              amount,
              offset = 0 )
```

```
Set the frames where the displacement takes place

Args:
    object:
        blender mesh, current object which will move
    initialFrame:
        integer, the starting frame of the movement
    finalFrame:
        integer, the final frame of the movement
    amount:
        float, amount of distance to move horizontally
    offset:
        float, amount of distance to move vertically

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 4 of file setFunction.py.

4.25.1.6 rotation()

```
def Visualization.setFunction.rotation (
              object,
              initialFrame,
              finalFrame,
              amount )
```

```
Set the frames where the rotation takes place

Args:
    object:
        blender mesh, current object which will rotate
    initialFrame:
        integer, the starting frame of the rotation
    finalFrame:
        integer, the final frame of the rotation
    amount:
        float, amount of rotation

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 37 of file setFunction.py.

4.26 Visualization.setLamp Namespace Reference

Functions

- def setLamp (scene, x, y, z)

4.26.1 Function Documentation

4.26.1.1 setLamp()

```
def Visualization.setLamp.setLamp (
              scene,
              x,
              y,
              z )


Set lamp function
This function set the parameters to create the lights for
the video

Args:
    scene:
        bpy.context.scene, contains the scene of the video
    x:
        float, x coordinate of the position of the lamp
    y:
        float, y coordinate of the position of the lamp
    z:
        float, z coordinate of the position of the lamp

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 3 of file setLamp.py.

4.27 Visualization.setRender Namespace Reference

Functions

- def Parameters (numberOfFrames, resolution, locationWall, locationBackground, width, height, path↩ Directory)

4.27.1 Function Documentation

4.27.1.1 Parameters()

```
def Visualization.setRender.Parameters (
              numberOfFrames,
              resolution,
              locationWall,
              locationBackground,
              width,
              height,
              pathDirectory )
```

```
Define the parameters of the render

Args:
    numberOfFrames:
        integer, the number of frames of the animation.
    resolution:
        float, the resolution
    locationWall:
        coordinate of the position of the element which
        represents the blue wall.
    locationBackground:
        coordinate of the position of the element which
        represents the background.
    width:
        float, the width of the structure.
    height:
        float, the height of the structure.
    pathdirectory:
        string, path where the video will be store

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 19 of file setRender.py.

Chapter 5

Class Documentation

5.1 Visualization.BlenderObject.BlenderObject Class Reference

Public Member Functions

- def __init__ (self, name, location, rotation, dimension, color, type)
- def get_geometricalObject (self)

Public Attributes

- name
- location
- rotation
- dimension
- color
- type
- object
- obj

5.1.1 Detailed Description

Definition at line 21 of file BlenderObject.py.

5.1.2 Constructor & Destructor Documentation

5.1.2.1 __init__()

```
def Visualization.BlenderObject.BlenderObject.__init__ (
              self,
              name,
              location,
              rotation,
              dimension,
              color,
              type )
```

```
Set the color of an object

Args:
    name:
string, object name
    location:
coordinate of the object
    rotation:
define rotation
    dimension:
float, size of the object
    color
    type:
type of blender object according to the definition above

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 28 of file BlenderObject.py.

5.1.3 Member Function Documentation

5.1.3.1 get_geometricalObject()

```
def Visualization.BlenderObject.BlenderObject.get_geometricalObject (
              self )

Define the object in Blender

Args:
    nothing is taken

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 63 of file BlenderObject.py.

5.1.4 Member Data Documentation

5.1.4.1 color

```
Visualization.BlenderObject.BlenderObject.color
```

Definition at line 59 of file BlenderObject.py.

5.1.4.2 dimension

```
Visualization.BlenderObject.BlenderObject.dimension
```

Definition at line 57 of file BlenderObject.py.

5.1.4.3 location

`Visualization.BlenderObject.BlenderObject.location`

Definition at line 53 of file BlenderObject.py.

5.1.4.4 name

`Visualization.BlenderObject.BlenderObject.name`

Definition at line 51 of file BlenderObject.py.

5.1.4.5 obj

`Visualization.BlenderObject.BlenderObject.obj`

Definition at line 83 of file BlenderObject.py.

5.1.4.6 object

`Visualization.BlenderObject.BlenderObject.object`

Definition at line 78 of file BlenderObject.py.

5.1.4.7 rotation

`Visualization.BlenderObject.BlenderObject.rotation`

Definition at line 55 of file BlenderObject.py.

5.1.4.8 type

`Visualization.BlenderObject.BlenderObject.type`

Definition at line 61 of file BlenderObject.py.

The documentation for this class was generated from the following file:

- Visualization/BlenderObject.py

5.2 pkg.isdh.component.Component Class Reference

Public Member Functions

- def __init__ (self, name, x1, x2, defo_length, lp_level1, lp_level2, p=0)
- def __repr__ (self)

Public Attributes

- name
- x1
- x2
- defo_length
- lp_level1
- lp_level2
- mass
- mass_position

5.2.1 Detailed Description

Definition at line 1 of file component.py.

5.2.2 Constructor & Destructor Documentation

5.2.2.1 __init__()

```
def pkg.isdh.component.Component.__init__ (
              self,
              name,
              x1,
              x2,
              defo_length,
              lp_level1,
              lp_level2,
              p = 0 )
```

Definition at line 2 of file component.py.

5.2.3 Member Function Documentation

5.2.3.1 __repr__()

```
def pkg.isdh.component.Component.__repr__ (
              self )


Return the string representation of the object.

Args:
    nothing is taken
Returns:
    string
Raises:
    nothing is raised
```

Definition at line 16 of file component.py.

5.2.4 Member Data Documentation

5.2.4.1 defo_length

`pkg.isdh.component.Component.defo_length`

Definition at line 7 of file component.py.

5.2.4.2 lp_level1

`pkg.isdh.component.Component.lp_level1`

Definition at line 8 of file component.py.

5.2.4.3 lp_level2

`pkg.isdh.component.Component.lp_level2`

Definition at line 9 of file component.py.

5.2.4.4 mass

`pkg.isdh.component.Component.mass`

Definition at line 12 of file component.py.

5.2.4.5 mass_position

`pkg.isdh.component.Component.mass_position`

Definition at line 13 of file component.py.

5.2.4.6 name

`pkg.isdh.component.Component.name`

Definition at line 4 of file component.py.

5.2.4.7 x1

`pkg.isdh.component.Component.x1`

Definition at line 5 of file component.py.

5.2.4.8 x2

`pkg.isdh.component.Component.x2`

Definition at line 6 of file component.py.

The documentation for this class was generated from the following file:

- pkg/isdh/component.py

5.3 pkg.structure_core.Component.Component Class Reference

Public Member Functions

- def __init__ (self, leftNode, rightNode, rigidLength, componentsName, isGap=False)
- def __repr__ (self)
- def draw (self, screen, offset, y_scaling)
- def length (self)
- def deformable_length (self)
- def moves (self, list_of_nodes)
- def link_to_barrier (self)
- def link_to_firewall (self)
- def next_gap (self)

Public Attributes

- name
- leftNode
- rightNode
- rigidLength
- isGap
- connectedToBarrier
- connectedToFirewall

5.3.1 Detailed Description

`Defines the components in the topological model`

Definition at line 20 of file Component.py.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 __init__()

```
def pkg.structure_core.Component.Component.__init__ (
              self,
              leftNode,
              rightNode,
              rigidLength,
              componentsName,
              isGap = False )
```

```
Constructs the class structure_core.Component.Component.

Args:
  leftNode:
the node object that defines the left node of the component
  rightNode:
the node object that defines the right node of the component
  rigidLength:
scalar value of the rigid length of the component
  componentsName:
string that represents the name of the component
Returns:
  an object of the class.
Raises:
  nothing is raised.
```

Definition at line 27 of file Component.py.

5.3.3 Member Function Documentation

5.3.3.1 __repr__()

```
def pkg.structure_core.Component.Component.__repr__ (
              self )
```

Definition at line 57 of file Component.py.

5.3.3.2 deformable_length()

```
def pkg.structure_core.Component.Component.deformable_length (
              self )
```

```
Calculates the current deformale length of the component.

Args:
  nothing is taken.
Returns:
  scalar value of the current deformable length of the component.
Raises:
  nothins is raised.
```

Definition at line 107 of file Component.py.

5.3.3.3 draw()

```
def pkg.structure_core.Component.Component.draw (
              self,
              screen,
              offset,
              y_scaling )
```

```
Draws onto the screan the current statw of the structure (DEBUG purpose).

Args:
  screan:
defines the screan to output the debugging data and its properties
  offset:
...
  y_scaling:
...
Returns:
  nothing is returned
Raises:
  nothing is raised
```

Definition at line 60 of file Component.py.

5.3.3.4 length()

```
def pkg.structure_core.Component.Component.length (
              self )
```

Definition at line 103 of file Component.py.

5.3.3.5 link_to_barrier()

```
def pkg.structure_core.Component.Component.link_to_barrier (
              self )
```

```
Determine whether the component is connected to the barrier (no gap is
standing in its way).

Args:
  nothing is taken.
Returns:
  nothing is returned.
Raises:
  nothing is raised.
```

Definition at line 141 of file Component.py.

5.3.3.6 link_to_firewall()

```
def pkg.structure_core.Component.Component.link_to_firewall (
              self )
```

```
Determine whether the component is connected to the firewall (no gap is
standing in its way).

Args:
  nothing is taken.
Returns:
  nothing is returned.
Raises:
  nothing is raised.
```

Definition at line 161 of file Component.py.

5.3.3.7 moves()

```
def pkg.structure_core.Component.Component.moves (
              self,
              list_of_nodes )
```

```
Determines the nodes that are going to be moved as a result to the
defomation step.

Args:
  list_of_nodes:
list that contains the nodes that are deforming their components
Returns:
  list of the nodes that are going to be moved by the deformation step.
Raises:
  nothing is raised.
```

Definition at line 124 of file Component.py.

5.3.3.8 next_gap()

```
def pkg.structure_core.Component.Component.next_gap (
              self )
```

```
Determines whether there is another gap along the loadpath to be deformed.

Args:
  nothing is taken.
Returns:
  if there is no other gap left, it returns None, otherwise the gap object.
Raises:
  nothing is raised.
```

Definition at line 181 of file Component.py.

5.3.4 Member Data Documentation

5.3.4.1 connectedToBarrier

`pkg.structure_core.Component.Component.connectedToBarrier`

Definition at line 51 of file Component.py.

5.3.4.2 connectedToFirewall

`pkg.structure_core.Component.Component.connectedToFirewall`

Definition at line 52 of file Component.py.

5.3.4.3 isGap

`pkg.structure_core.Component.Component.isGap`

Definition at line 50 of file Component.py.

5.3.4.4 leftNode

`pkg.structure_core.Component.Component.leftNode`

Definition at line 47 of file Component.py.

5.3.4.5 name

`pkg.structure_core.Component.Component.name`

Definition at line 46 of file Component.py.

5.3.4.6 rightNode

`pkg.structure_core.Component.Component.rightNode`

Definition at line 48 of file Component.py.

5.3.4.7 rigidLength

`pkg.structure_core.Component.Component.rigidLength`

Definition at line 49 of file Component.py.

The documentation for this class was generated from the following file:

- pkg/structure_core/Component.py

5.4 Visualization.CreateVideo.CreateVideo Class Reference

Public Member Functions

- def __init__ (self, i_s, d_h, v_o, fps, resolution, pathDirectory)

5.4.1 Detailed Description

Definition at line 11 of file CreateVideo.py.

5.4.2 Constructor & Destructor Documentation

5.4.2.1 __init__()

```
def Visualization.CreateVideo.CreateVideo.__init__ (
              self,
              i_s,
              d_h,
              v_o,
              fps,
              resolution,
              pathDirectory )


Creates video in blender

Args:
    i_s:
initial state
    d_h:
deformation history
    v_o:
vertical offset-distance between horizontal paths
    fps:
frames per second
    resolution:
Number of pixels in the render image
    path directory:
path where the video will be stored

Returns:
    nothing is returned

Raises:
    exceptions raised by the initial state
```

Definition at line 18 of file CreateVideo.py.

The documentation for this class was generated from the following file:

- Visualization/CreateVideo.py

5.5 pkg.structure_core.CrossComponent.CrossComponent Class Reference

Public Member Functions

- def __init__ (self, name, leftNode, rightNode, rigidLength)
- def __repr__ (self)
- def draw (self, screen, offset, y_scaling)
- def left_deforms (self, list_of_nodes)
- def right_deforms (self, list_of_nodes)
- def length (self)
- def deformable_length (self)
- def is_valid (self)
- def link_to_barrier (self)
- def link_to_firewall (self)

Public Attributes

- name
- leftNode
- rightNode
- rigidLength
- breakable
- broken
- connectedToBarrier
- connectedToFirewall

5.5.1 Detailed Description

```
Defiens the cross-components in the topological model
```

Definition at line 17 of file CrossComponent.py.

5.5.2 Constructor & Destructor Documentation

5.5.2.1 __init__()

```
def pkg.structure_core.CrossComponent.CrossComponent.__init__ (
              self,
              name,
              leftNode,
              rightNode,
              rigidLength )

Constructs the clas structure_core.CrossComponent.crossComponent.

Args:
  name:
string of the name of the cross_component
  leftNode:
node object of the left node of the cross_component
  rightNode:
node object of the right node of the cross_component
  rigidLength:
scalar value of the rigid length of the cross_component
Returns:
  an object of the class.
Raises:
  nothing is raised.
```

Definition at line 23 of file CrossComponent.py.

5.5.3 Member Function Documentation

5.5.3.1 __repr__()

```
def pkg.structure_core.CrossComponent.CrossComponent.__repr__ (
              self )
```

Definition at line 56 of file CrossComponent.py.

5.5.3.2 deformable_length()

```
def pkg.structure_core.CrossComponent.CrossComponent.deformable_length (
              self )
```

```
Calculates the current deformale length of the cross_component.

Args:
  nothing is taken.
Returns:
  scalar value of the current deformable length of the cross_component.
Raises:
  nothins is raised.
```

Definition at line 150 of file CrossComponent.py.

5.5.3.3 draw()

```
def pkg.structure_core.CrossComponent.CrossComponent.draw (
              self,
              screen,
              offset,
              y_scaling )
```

```
Draws onto the screan the current statw of the structure (DEBUG purpose).

Args:
  screan:
defines the screan to output the debugging data and its properties
  offset:
...
  y_scaling:
...
Returns:
  nothing is returned
Raises:
  nothing is raised
```

Definition at line 59 of file CrossComponent.py.

5.5.3.4 is_valid()

```
def pkg.structure_core.CrossComponent.CrossComponent.is_valid (
              self )
```

```
Determines whether the cross_component is broken.

Args:
  nothing is taken.
Returns:
  True if the cross_component is not broken, False otherwise.
Raises:
  nothing is raised.
```

Definition at line 166 of file CrossComponent.py.

5.5.3.5 left_deforms()

```
def pkg.structure_core.CrossComponent.CrossComponent.left_deforms (
              self,
              list_of_nodes )
```

```
Determines whether the deformation step will move the left node of the
cross_component.

Args:
  list_of_nodes:
list that contains the nodes that are deforming their components
Returns:
  True if the defomation step will lead to motion to the left node, False otherwise.
Raises:
  nothing is raised.
```

Definition at line 104 of file CrossComponent.py.

5.5.3.6 length()

```
def pkg.structure_core.CrossComponent.CrossComponent.length (
              self )
```

```
Calculates the current length of the cross_component.

Args:
  nothing is taken.
Returns:
  scalar value of the current length of the cross_component.
Raises:
  nothing is raised.
```

Definition at line 138 of file CrossComponent.py.

5.5.3.7 link_to_barrier()

```
def pkg.structure_core.CrossComponent.CrossComponent.link_to_barrier (
              self )
```

```
Determine whether the cross_component is connected to the barrier
(no gap is standing in its way).

Args:
  nothing is taken.
Returns:
  nothing is returned.
Raises:
  nothing is raised.
```

Definition at line 178 of file CrossComponent.py.

5.5.3.8 link_to_firewall()

```
def pkg.structure_core.CrossComponent.CrossComponent.link_to_firewall (
              self )
```

```
Determine whether the cross_component is connected to the firewall
(no gap is standing in its way).

Args:
  nothing is taken.
Returns:
  nothing is returned.
Raises:
  nothing is raised.
```

Definition at line 195 of file CrossComponent.py.

5.5.3.9 right_deforms()

```
def pkg.structure_core.CrossComponent.CrossComponent.right_deforms (
              self,
              list_of_nodes )
```

```
Determines whether the deformation step will move the right node of the
cross_component.

Args:
  list_of_nodes:
list that contains the nodes that are deforming their components
Returns:
  True if the defomation step will lead to motion to the right node, False otherwise.
Raises:
  nothing is raised.
```

Definition at line 121 of file CrossComponent.py.

5.5.4 Member Data Documentation

5.5.4.1 breakable

`pkg.structure_core.CrossComponent.CrossComponent.breakable`

Definition at line 47 of file CrossComponent.py.

5.5.4.2 broken

`pkg.structure_core.CrossComponent.CrossComponent.broken`

Definition at line 48 of file CrossComponent.py.

5.5.4.3 connectedToBarrier

`pkg.structure_core.CrossComponent.CrossComponent.connectedToBarrier`

Definition at line 50 of file CrossComponent.py.

5.5.4.4 connectedToFirewall

`pkg.structure_core.CrossComponent.CrossComponent.connectedToFirewall`

Definition at line 51 of file CrossComponent.py.

5.5.4.5 leftNode

`pkg.structure_core.CrossComponent.CrossComponent.leftNode`

Definition at line 43 of file CrossComponent.py.

5.5.4.6 name

`pkg.structure_core.CrossComponent.CrossComponent.name`

Definition at line 42 of file CrossComponent.py.

5.5.4.7 rightNode

`pkg.structure_core.CrossComponent.CrossComponent.rightNode`

Definition at line 44 of file CrossComponent.py.

5.5.4.8 rigidLength

```
pkg.structure_core.CrossComponent.CrossComponent.rigidLength
```

Definition at line 45 of file CrossComponent.py.

The documentation for this class was generated from the following file:

- pkg/structure_core/CrossComponent.py

5.6 pkg.isdh.deformation_step.DeformationStep Class Reference

Public Member Functions

- def __init__ (self, amount, initial_deformation_amount, transformation)

Public Attributes

- amount
- frame_begin
- frame_end
- transformation

5.6.1 Detailed Description

Definition at line 1 of file deformation_step.py.

5.6.2 Constructor & Destructor Documentation

5.6.2.1 __init__()

```
def pkg.isdh.deformation_step.DeformationStep.__init__ (
              self,
              amount,
              initial_deformation_amount,
              transformation )
```

Definition at line 2 of file deformation_step.py.

5.6.3 Member Data Documentation

5.6.3.1 amount

```
pkg.isdh.deformation_step.DeformationStep.amount
```

Definition at line 3 of file deformation_step.py.

5.6.3.2 frame_begin

`pkg.isdh.deformation_step.DeformationStep.frame_begin`

Definition at line 4 of file deformation_step.py.

5.6.3.3 frame_end

`pkg.isdh.deformation_step.DeformationStep.frame_end`

Definition at line 5 of file deformation_step.py.

5.6.3.4 transformation

`pkg.isdh.deformation_step.DeformationStep.transformation`

Definition at line 6 of file deformation_step.py.

The documentation for this class was generated from the following file:

- pkg/isdh/deformation_step.py

5.7 Visualization.Member.generalMember Class Reference

Public Member Functions

- def __init__ (self, nameOfMember, x1, x2, deformableLength, level1, level2, separation, mass_position=0)
- def __repr__ (self)
- def move (self, initialFrame, finalFrame, distance)
- def deform (self, initialFrame, finalFrame, amount, newAmount, newAngle, oldAngle, newDefoLength, old↩ DefoLength, currentStep, numberOfSteps)

Public Attributes

- dL
- l1
- l2
- sep
- mass_position
- dY
- dX
- rL
- angle
- totalLength
- rDL
- rRL
- elementSize
- rPM
- deformPart
- nonDeformPart
- tag
- lNode
- rNode
- mass

5.7.1 Detailed Description

Definition at line 14 of file Member.py.

5.7.2 Constructor & Destructor Documentation

5.7.2.1 __init__()

```
def Visualization.Member.generalMember.__init__ (
              self,
              nameOfMember,
              x1,
              x2,
              deformableLength,
              level1,
              level2,
              separation,
              mass_position = 0 )
```

```
Creates the object member

Args:
    numberOfFrames:
float, number of frames that the objects needs for the action
    x1:
float, location in x of the point one
    x2:
float, location in x of the point two
    deformableLength:
float, deformable length
    level1:
float, level of point one
    level2:
float, level of point two
    separation:
float, vertical position of the element
    mass_position:
float, the absolute position of mass

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 23 of file Member.py.

5.7.3 Member Function Documentation

5.7.3.1 __repr__()

```
def Visualization.Member.generalMember.__repr__ (
              self )
```

Definition at line 158 of file Member.py.

5.7.3.2 deform()

```
def Visualization.Member.generalMember.deform (
              self,
              initialFrame,
              finalFrame,
              amount,
              newAmount,
              newAngle,
              oldAngle,
              newDefoLength,
              oldDefoLength,
              currentStep,
              numberOfSteps )


Define the movement of the element

Args:
    initialFrame:
integer, the initial frame of the movement
    finalFrame:
integer, the final frame of the movement
    amount:
float, the previous value of the amount of deformation
    newAmount:
float, the current value of the amount of deformation
    newAngle:
float, the current angle to rotate
    oldAngle:
float, the old angle to rotate
    newDefoLength:
float, the current deformable length
    oldDefoLength:
float, the old deformable length
    currentStep:
integer, steps in which the action is
    numberOfSteps:
integer, number of steps that covers the action

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 224 of file Member.py.

5.7.3.3 move()

```
def Visualization.Member.generalMember.move (
              self,
              initialFrame,
              finalFrame,
              distance )


Define the movement of the element

Args:
    initialFrame:
integer, the initial frame of the movement
    finalFrame:
```

```
integer, the final frame of the movement
    distance:
float, the distance of movement

Returns:
    nothing is returned

Raises:
    nothing is raised
```

Definition at line 164 of file Member.py.

5.7.4 Member Data Documentation

5.7.4.1 angle

```
Visualization.Member.generalMember.angle
```

Definition at line 68 of file Member.py.

5.7.4.2 deformPart

```
Visualization.Member.generalMember.deformPart
```

Definition at line 93 of file Member.py.

5.7.4.3 dL

```
Visualization.Member.generalMember.dL
```

Definition at line 52 of file Member.py.

5.7.4.4 dX

```
Visualization.Member.generalMember.dX
```

Definition at line 64 of file Member.py.

5.7.4.5 dY

```
Visualization.Member.generalMember.dY
```

Definition at line 62 of file Member.py.

5.7.4.6 elementSize

```
Visualization.Member.generalMember.elementSize
```

Definition at line 76 of file Member.py.

5.7.4.7 l1

`Visualization.Member.generalMember.l1`

Definition at line 54 of file Member.py.

5.7.4.8 l2

`Visualization.Member.generalMember.l2`

Definition at line 56 of file Member.py.

5.7.4.9 lNode

`Visualization.Member.generalMember.lNode`

Definition at line 131 of file Member.py.

5.7.4.10 mass

`Visualization.Member.generalMember.mass`

Definition at line 156 of file Member.py.

5.7.4.11 mass_position

`Visualization.Member.generalMember.mass_position`

Definition at line 60 of file Member.py.

5.7.4.12 nonDeformPart

`Visualization.Member.generalMember.nonDeformPart`

Definition at line 107 of file Member.py.

5.7.4.13 rDL

`Visualization.Member.generalMember.rDL`

Definition at line 72 of file Member.py.

5.7.4.14 rL

`Visualization.Member.generalMember.rL`

Definition at line 66 of file Member.py.

5.7.4.15 rNode

`Visualization.Member.generalMember.rNode`

Definition at line 143 of file Member.py.

5.7.4.16 rPM

`Visualization.Member.generalMember.rPM`

Definition at line 78 of file Member.py.

5.7.4.17 rRL

`Visualization.Member.generalMember.rRL`

Definition at line 74 of file Member.py.

5.7.4.18 sep

`Visualization.Member.generalMember.sep`

Definition at line 58 of file Member.py.

5.7.4.19 tag

`Visualization.Member.generalMember.tag`

Definition at line 119 of file Member.py.

5.7.4.20 totalLength

`Visualization.Member.generalMember.totalLength`

Definition at line 70 of file Member.py.

The documentation for this class was generated from the following file:

- Visualization/Member.py

5.8 pkg.isdh.isdh_helper.IsdhHelper Class Reference

Public Member Functions

- def __init__ (self)
- def register (self, tree)
- def save (self, activeNode)
- def unsave (self, activeNode)
- def save_defo_step (self, comp, stepType, stepAmount)
- def update_amount (self, amount)
- def init_ood (self)
- def copy_ood (self)
- def save_ood (self)

Public Attributes

- i_s
- d_h
- isdh_dict
- ood
- amount

5.8.1 Detailed Description

Definition at line 5 of file isdh_helper.py.

5.8.2 Constructor & Destructor Documentation

5.8.2.1 __init__()

```
def pkg.isdh.isdh_helper.IsdhHelper.__init__ (
              self )
```

Definition at line 6 of file isdh_helper.py.

5.8.3 Member Function Documentation

5.8.3.1 copy_ood()

```
def pkg.isdh.isdh_helper.IsdhHelper.copy_ood (
              self )
```

```
Return a copy of self.ood, with original key and a copy of value.

The copy returned is a dictionary, in which the keys are references to
the original keys, and the values are copies.
This way the items in self.i_s can be used as keys for the copy and,
at the same time, modifications to self.ood do not modify the copy.

Args:
    nothing is taken.
Returns:
    a dictionary is returned.
Raises:
    nothing is raised.
```

Definition at line 207 of file isdh_helper.py.

5.8.3.2 init_ood()

```
def pkg.isdh.isdh_helper.IsdhHelper.init_ood (
              self )
```

```
Initialise self.ood.

Args:
    nothing is taken
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 192 of file isdh_helper.py.

5.8.3.3 register()

```
def pkg.isdh.isdh_helper.IsdhHelper.register (
              self,
              tree )
```

```
Register the saver to tree.

Append self to tree.savers, initialise self.i_s, initialise self.ood.

Args:
    tree:
a tree_core.tree.Tree object
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 13 of file isdh_helper.py.

5.8.3.4 save()

```
def pkg.isdh.isdh_helper.IsdhHelper.save (
              self,
              activeNode )
```

```
Save a defromation step of the whole structure.

Args:
    activeNode:
a tree_core.node_tree.NodeTree object
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 71 of file isdh_helper.py.

5.8.3.5 save_defo_step()

```
def pkg.isdh.isdh_helper.IsdhHelper.save_defo_step (
              self,
              comp,
              stepType,
              stepAmount )
```

```
Save or un-save a deformation step of one component.

Args:
    comp:
a structure_core.component.Component object
or
a structure_core.cross_component.CrossComponent object
    stepType:
'd' or 'm' (deformation or movement)
    stepAmount:
a double, the amount of deformation of the step
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 134 of file isdh_helper.py.

5.8.3.6 save_ood()

```
def pkg.isdh.isdh_helper.IsdhHelper.save_ood (
              self )
```

```
Save the Order of Deformation in self.ood as a solution.

Args:
    nothing is taken.
Returns:
    nothing is returned.
Raises:
    nothing is raised.
```

Definition at line 227 of file isdh_helper.py.

5.8.3.7 unsave()

```
def pkg.isdh.isdh_helper.IsdhHelper.unsave (
              self,
              activeNode )
```

```
Un-save a defromation step of the whole structure.

Args:
    activeNode:
a tree_core.node_tree.NodeTree object
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 103 of file isdh_helper.py.

5.8.3.8 update_amount()

```
def pkg.isdh.isdh_helper.IsdhHelper.update_amount (
              self,
              amount )
```

```
Update the amount of deformation occured so far.

Args:
    amount:
a double, the amount of deformation of the new deformation
step.
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 178 of file isdh_helper.py.

5.8.4 Member Data Documentation

5.8.4.1 amount

```
pkg.isdh.isdh_helper.IsdhHelper.amount
```

Definition at line 11 of file isdh_helper.py.

5.8.4.2 d_h

```
pkg.isdh.isdh_helper.IsdhHelper.d_h
```

Definition at line 8 of file isdh_helper.py.

5.8.4.3 i_s

```
pkg.isdh.isdh_helper.IsdhHelper.i_s
```

Definition at line 7 of file isdh_helper.py.

5.8.4.4 isdh_dict

```
pkg.isdh.isdh_helper.IsdhHelper.isdh_dict
```

Definition at line 9 of file isdh_helper.py.

5.8.4.5 ood

```
pkg.isdh.isdh_helper.IsdhHelper.ood
```

Definition at line 10 of file isdh_helper.py.

The documentation for this class was generated from the following file:

- pkg/isdh/isdh_helper.py

5.9 pkg.structure_core.Loadpath.Loadpath Class Reference

Public Member Functions

- def __init__ (self, level)

Public Attributes

- listComponents
- setNodes
- level

5.9.1 Detailed Description

```
Groups all components and nodes at the same loadpath level.
```

Definition at line 3 of file Loadpath.py.

5.9.2 Constructor & Destructor Documentation

5.9.2.1 __init__()

```
def pkg.structure_core.Loadpath.Loadpath.__init__ (
              self,
              level )
```

```
Constructor of the class structure_core.loadpath.Loadpath.

Args:
  level:
an integer, that identifies the loadpath level
Returns:
  an object of the class.
Raises:
  nothing is raised.
```

Definition at line 6 of file Loadpath.py.

5.9.3 Member Data Documentation

5.9.3.1 level

`pkg.structure_core.Loadpath.Loadpath.level`

Definition at line 19 of file Loadpath.py.

5.9.3.2 listComponents

`pkg.structure_core.Loadpath.Loadpath.listComponents`

Definition at line 17 of file Loadpath.py.

5.9.3.3 setNodes

`pkg.structure_core.Loadpath.Loadpath.setNodes`

Definition at line 18 of file Loadpath.py.

The documentation for this class was generated from the following file:

- pkg/structure_core/Loadpath.py

5.10 pkg.structure_core.Node.Node Class Reference

Public Member Functions

- def __init__ (self, point, loadpathLevel)
- def __repr__ (self)
- def __eq__ (self, other)
- def __hash__ (self)
- def draw (self, screen, offset, y_scaling)
- def change_position (self, deformationStep)

Static Public Attributes

- position
- loadpathLevel
- towardsFirewall
- towardsBarrier
- onFirewall
- onBarrier

5.10.1 Detailed Description

```
Constructs the nodes that define the boundary of the components
```

Definition at line 14 of file Node.py.

5.10.2 Constructor & Destructor Documentation

5.10.2.1 __init__()

```
def pkg.structure_core.Node.Node.__init__ (
              self,
              point,
              loadpathLevel )
```

Definition at line 17 of file Node.py.

5.10.3 Member Function Documentation

5.10.3.1 __eq__()

```
def pkg.structure_core.Node.Node.__eq__ (
              self,
              other )
```

Definition at line 41 of file Node.py.

5.10.3.2 __hash__()

```
def pkg.structure_core.Node.Node.__hash__ (
              self )
```

Definition at line 49 of file Node.py.

5.10.3.3 __repr__()

```
def pkg.structure_core.Node.Node.__repr__ (
              self )
```

Definition at line 37 of file Node.py.

5.10.3.4 change_position()

```
def pkg.structure_core.Node.Node.change_position (
              self,
              deformationStep )
```

```
Chnages the position of the node.

Args:
  deformationStep:
    the amount of the motion the node should move with
Returns:
  nothing is returned
Raises:
  notjhing is raised
```

Definition at line 79 of file Node.py.

5.10.3.5 draw()

```
def pkg.structure_core.Node.Node.draw (
              self,
              screen,
              offset,
              y_scaling )
```

```
Draws onto the screan the current statw of the structure (DEBUG purpose).

Args:
  screan:
    defines the screan to output the debugging data and its properties
  offset:
    ...
  y_scaling:
    ...
Returns:
   nothing is returned
Raises:
   nothing is raised
```

Definition at line 52 of file Node.py.

5.10.4 Member Data Documentation

5.10.4.1 loadpathLevel

```
pkg.structure_core.Node.Node.loadpathLevel  [static]
```

Definition at line 31 of file Node.py.

5.10.4.2 onBarrier

```
pkg.structure_core.Node.Node.onBarrier  [static]
```

Definition at line 35 of file Node.py.

5.10.4.3 onFirewall

```
pkg.structure_core.Node.Node.onFirewall  [static]
```

Definition at line 34 of file Node.py.

5.10.4.4 position

```
pkg.structure_core.Node.Node.position  [static]
```

```
Constructor of the class structure_core.Node.Node.

Args:
  point:
a scalar value defines the position of the node
  loadpathLevel:
a scalar value defines  the level of the containing loadpath
Returns:
  an object of the class
Raises:
  nothing is raised
```

Definition at line 30 of file Node.py.

5.10.4.5 towardsBarrier

```
pkg.structure_core.Node.Node.towardsBarrier  [static]
```

Definition at line 33 of file Node.py.

5.10.4.6 towardsFirewall

```
pkg.structure_core.Node.Node.towardsFirewall  [static]
```

Definition at line 32 of file Node.py.

The documentation for this class was generated from the following file:

- pkg/structure_core/Node.py

5.11 pkg.tree_core.NodeTree.NodeTree Class Reference

Public Member Functions

- def __init__ (self, deformingComps, structure=None, parent=None)
- def __repr__ (self)
- def d_print (self)
- def add_child (self, deformingComps, structure)
- def check_amount (self)
- def next_children (self, previous_children=None)
- def substitute_children (self, next_children)
- def determine_amount (self)
- def cross_components_amount (self, deformationLeadingNodes)
- def check_keep_deforming (self)
- def deform (self)
- def undeform (self)

Public Attributes

- amount
- structure
- parent
- children
- isValid
- keep
- substitute
- deformingComps
- movingComps
- deformingCrossComps
- movingCrossComps
- stretchingCrossComps

5.11.1 Detailed Description

Definition at line 2 of file NodeTree.py.

5.11.2 Constructor & Destructor Documentation

5.11.2.1 __init__()

```
def pkg.tree_core.NodeTree.NodeTree.__init__ (
              self,
              deformingComps,
              structure = None,
              parent = None )
```

Definition at line 3 of file NodeTree.py.

5.11.3 Member Function Documentation

5.11.3.1 __repr__()

```
def pkg.tree_core.NodeTree.NodeTree.__repr__ (
              self )
```

```
Return the string representation of the object.

If the isValid attribute is defined and True, the string representation
of the deformingComps attribute is returned.
If the isValid attribute is defined and False, the string INVALID is
returned
f the isValid attribute is not defined, "NodeTree obj" is returned.

Args:
    nothing is taken
Returns:
    string
Raises:
    nothing is raised
```

Definition at line 18 of file NodeTree.py.

5.11.3.2 add_child()

```
def pkg.tree_core.NodeTree.NodeTree.add_child (
              self,
              deformingComps,
              structure )
```

```
Append a child to the list self.children.

If the child is not valid, because the deformingComps contains
undeformable gaps, other children are created (varying deformingComps).
This process continues until a valid list of children is found.
Then the list is appended to the list self.children.

Args:
    deformingComps:
tuple of structure_core.Component objects to deform
    structure:
the unique structure_core.Structure object, to which
structure_core.Component objects belong
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 59 of file NodeTree.py.

5.11.3.3 check_amount()

```
def pkg.tree_core.NodeTree.NodeTree.check_amount (
              self )
```

```
Saves the amount of deformation in self.amount.

The amount of deformation that the structure can overcome, for the
deforming components, is determined.
If the amount is 0, the node is marked as invalid.
If there is any gap in self.deformingComps, the attribute
self.substitute is set to True, so that the node can be replaced
by other nodes which contain different gaps.

Args:
    nothing is taken
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 95 of file NodeTree.py.

5.11.3.4 check_keep_deforming()

```
def pkg.tree_core.NodeTree.NodeTree.check_keep_deforming (
              self )
```

```
Check if the components are still deforming after the previous step.

When the attribute keep of the parent nodeTree is True, all the
components, that were deforming in the previous deformation step,
should keep on deforming. Thus, if this is not the case, the NodeTree
is marked as invalid.
Args:
    nothing is taken
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 323 of file NodeTree.py.

5.11.3.5 cross_components_amount()

```
def pkg.tree_core.NodeTree.NodeTree.cross_components_amount (
              self,
              deformationLeadingNodes )
```

```
Returns the amount of deformation allowed by the crossComponents.

Args:
    deformationLeadingNodes:
list of Node objects.
Returns:
    a double, if there are deforming crossComponents.
    None, otherwise.
Raises:
    nothing is raised.
```

Definition at line 247 of file NodeTree.py.

5.11.3.6 d_print()

```
def pkg.tree_core.NodeTree.NodeTree.d_print (
              self )
```

```
Prints a detailed description of the object state.

Args:
    nothing is taken
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 42 of file NodeTree.py.

5.11.3.7 deform()

```
def pkg.tree_core.NodeTree.NodeTree.deform (
              self )
```

```
Deforms the structure, as defined by self.deformingComps.

Args:
    nothing is taken
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 350 of file NodeTree.py.

5.11.3.8 determine_amount()

```
def pkg.tree_core.NodeTree.NodeTree.determine_amount (
              self )
```

```
Computes the correct value of the deformation amount.

The deformation amount of the structure is first determined as the
minimum deformable length of the deformingComps.
Then connections are taken into account.
Finally, the minimum amount is saved in self.amount.

Args:
    nothing is taken
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 200 of file NodeTree.py.

5.11.3.9 next_children()

```
def pkg.tree_core.NodeTree.NodeTree.next_children (
              self,
              previous_children = None )
```

```
Returns a list of NodeTree objects to replace the previous list.

For each NodeTree object in previous_children, a list of NodeTree
objects is generated.
The NodeTree objects in this list are a clone of the original NodeTree
object, where a gap in deformingComps has been replaced by the next
right gap.
The union of all these lists gives the next_children list.

Args:
    previous_children:
list of NodeTree objects.
Returns:
    list of NodeTree objects.
Raises:
    nothing is raised.
```

Definition at line 118 of file NodeTree.py.

5.11.3.10 substitute_children()

```
def pkg.tree_core.NodeTree.NodeTree.substitute_children (
              self,
              next_children )
```

```
Sets the objects in next_children as proper children of self.parent.

Args:
    self:
the NodeTree object 'child' to substitute, created in
.add_child()
REMARK: at this point self.parent exists, but
self.parent.children doesn't contain self: i.e. the link
between child and parent only goes from the child to the parent
    next_children:
list of NodeTree objects to substitute self in the tree.
Returns:
    Nothing is returned.
Raises:
    Nothing is raised.
```

Definition at line 178 of file NodeTree.py.

5.11.3.11 undeform()

```
def pkg.tree_core.NodeTree.NodeTree.undeform (
              self )
```

```
Undeforms the structure, as defined by self.deformingComps.

Args:
    nothing is taken
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 369 of file NodeTree.py.

5.11.4 Member Data Documentation

5.11.4.1 amount

```
pkg.tree_core.NodeTree.NodeTree.amount
```

Definition at line 4 of file NodeTree.py.

5.11.4.2 children

```
pkg.tree_core.NodeTree.NodeTree.children
```

Definition at line 7 of file NodeTree.py.

5.11.4.3 deformingComps

`pkg.tree_core.NodeTree.NodeTree.deformingComps`

Definition at line 12 of file NodeTree.py.

5.11.4.4 deformingCrossComps

`pkg.tree_core.NodeTree.NodeTree.deformingCrossComps`

Definition at line 14 of file NodeTree.py.

5.11.4.5 isValid

`pkg.tree_core.NodeTree.NodeTree.isValid`

Definition at line 8 of file NodeTree.py.

5.11.4.6 keep

`pkg.tree_core.NodeTree.NodeTree.keep`

Definition at line 9 of file NodeTree.py.

5.11.4.7 movingComps

`pkg.tree_core.NodeTree.NodeTree.movingComps`

Definition at line 13 of file NodeTree.py.

5.11.4.8 movingCrossComps

`pkg.tree_core.NodeTree.NodeTree.movingCrossComps`

Definition at line 15 of file NodeTree.py.

5.11.4.9 parent

`pkg.tree_core.NodeTree.NodeTree.parent`

Definition at line 6 of file NodeTree.py.

5.11.4.10 stretchingCrossComps

`pkg.tree_core.NodeTree.NodeTree.stretchingCrossComps`

Definition at line 16 of file NodeTree.py.

5.11.4.11 structure

`pkg.tree_core.NodeTree.NodeTree.structure`

Definition at line 5 of file NodeTree.py.

5.11.4.12 substitute

`pkg.tree_core.NodeTree.NodeTree.substitute`

Definition at line 10 of file NodeTree.py.

The documentation for this class was generated from the following file:

- pkg/tree_core/NodeTree.py

5.12 pkg.structure_core.Structure.Structure Class Reference

Public Member Functions

- def __init__ (self, listLoadpaths, listCrossComponents=None)
- def draw (self)
- def task_one (self)
- def task_two (self, blackbox)
- def reset_connections_to_barrier_and_firewall (self)
- def get_deforming_components (self)

Public Attributes

- listLoadpaths
- listCrossComponents
- listGaps

5.12.1 Detailed Description

`Groups all the entities of the topological model.`

Definition at line 21 of file Structure.py.

5.12.2 Constructor & Destructor Documentation

5.12.2.1 __init__()

```
def pkg.structure_core.Structure.Structure.__init__ (
              self,
              listLoadpaths,
              listCrossComponents = None )

Constructor of the class structure_core.structure.Structure.

Args:
  listLoadpaths:
a list of structure_core.loadpath.Loadpath objects
  listCrossComponents:
structure_core.cross_component.CrossComponent
Returns:
  an object of the class.
Raises:
  nothing is raised.
```

Definition at line 24 of file Structure.py.

5.12.3 Member Function Documentation

5.12.3.1 draw()

```
def pkg.structure_core.Structure.Structure.draw (
              self )

Draws onto the screen the current state of the strucure (DEBUG purpose).

Args:
  nothing is taken
Returns:
  nothing is returned
Raises:
  nothing is raised.
```

Definition at line 44 of file Structure.py.

5.12.3.2 get_deforming_components()

```
def pkg.structure_core.Structure.Structure.get_deforming_components (
              self )

Return a list of deforming components tuples.

Returns a list of tuples. Each tuple contains the components to deform, one
from every loadpath. There exists a tuple for every possible combination of
components.

Args:
nothing is taken
Returns:
a list of tuples, e.g.:
  [(comp1-lp1, comp1-lp2),
   (comp1-lp1, comp2-lp2),
   (comp2-lp1, comp1-lp2),
   (comp2-lp1, comp2-lp2),
   (comp3-lp1, comp1-lp2),
   (comp3-lp1, comp2-lp2)]
Raises:
nothing is raised
```

Definition at line 220 of file Structure.py.

5.12.3.3 reset_connections_to_barrier_and_firewall()

```
def pkg.structure_core.Structure.Structure.reset_connections_to_barrier_and_firewall (
              self )
```

```
Updates the connections of components, cross components and nodes.

The attributes .connectedToBarrier and .connectedToFirewall of every
structure_core.Component and structure_core.CrossComponent object are
updated.
The attributes .onBarrier and .onFirewall of every structure_core.Node
object are updated.

Args:
nothing is taken
Returns:
nothing is returned
Raises:
nothing is raised
```

Definition at line 166 of file Structure.py.

5.12.3.4 task_one()

```
def pkg.structure_core.Structure.Structure.task_one (
              self )
```

```
Finds all the Order of Deformation of the structure.

self.task_one() -> [i_s, d_h].

Args:
  nothing is taken
Returns:
  i_s, a list of isdh.component.Component objects
  d_h, a list of dictionaries such as:
  {  isdh-comp1 : [DeformationStep1,
                   DeformationStep2,
                   ...],
     isdh-comp2 : [DeformationStep1,
                   DeformationStep2,
                   ...],
     isdh-comp3 : [DeformationStep1,
                   DeformationStep2,
                   ...],
  },
  where the keys are the elements of i_s and the values are lists of
  isdh.deformation_step.DeformationStep objects.
Raises:
  nothing is raised.
```

Definition at line 70 of file Structure.py.

5.12.3.5 task_two()

```
def pkg.structure_core.Structure.Structure.task_two (
              self,
              blackbox )
```

```
Finds the physical Order of Deformation of the structure.

self.task_one() -> [i_s, d_h].

Args:
  blackbox:
a function that decides whether the activeNode of the tree of the
structure is the valid next deformationStep or not.
Returns:
  i_s, a list of isdh.component.Component objects
  d_h, a list with one dictionary such as:
  {  isdh-comp1 : [DeformationStep1,
                   DeformationStep2,
                   ...],
     isdh-comp2 : [DeformationStep1,
                   DeformationStep2,
                   ...],
     isdh-comp3 : [DeformationStep1,
                   DeformationStep2,
                   ...],
  },
  where the keys are the elements of i_s and the values are lists of
  isdh.deformation_step.DeformationStep objects.
Raises:
  exceptions raised by the blackbox, remain unhandled.
```

Definition at line 121 of file Structure.py.

5.12.4 Member Data Documentation

5.12.4.1 listCrossComponents

```
pkg.structure_core.Structure.Structure.listCrossComponents
```

Definition at line 38 of file Structure.py.

5.12.4.2 listGaps

```
pkg.structure_core.Structure.Structure.listGaps
```

Definition at line 40 of file Structure.py.

5.12.4.3 listLoadpaths

```
pkg.structure_core.Structure.Structure.listLoadpaths
```

Definition at line 37 of file Structure.py.

The documentation for this class was generated from the following file:

- pkg/structure_core/Structure.py

5.13 pkg.tree_core.tree.Tree Class Reference

Public Member Functions

- def __init__ (self, structure)
- def __repr__ (self)
- def print (self)
- def add_children (self)
- def end (self)
- def go_down (self)
- def go_up (self)
- def go_right (self)
- def deform (self)
- def undeform (self)
- def surf (self, blackbox)

Public Attributes

- root
- activeNode
- structure
- savers

5.13.1 Detailed Description

Definition at line 4 of file tree.py.

5.13.2 Constructor & Destructor Documentation

5.13.2.1 __init__()

```
def pkg.tree_core.tree.Tree.__init__ (
              self,
              structure )
```

```
Constructor of the class tree_core.Tree.

Args:
    structure:
the structure_core.Structure object to which the tree is linked
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 5 of file tree.py.

5.13.3 Member Function Documentation

5.13.3.1 __repr__()

```
def pkg.tree_core.tree.Tree.__repr__ (
              self )
```

```
Return the string representation of the object.

If the activeNode attribute is defined, its string representation is
returned. Otherwise "Tree obj" is returned.

Args:
    nothing is taken
Returns:
    string
Raises:
    nothing is raised
```

Definition at line 27 of file tree.py.

5.13.3.2 add_children()

```
def pkg.tree_core.tree.Tree.add_children (
              self )
```

```
Add children to self.activeNode.

The children of self.activeNode are generated and appended to the list
self.activeNode.children.
If the end of the tree as been reached (no valid child has been found),
the current Order of Deformation is saved.

Args:
    nothing is taken
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 76 of file tree.py.

5.13.3.3 deform()

```
def pkg.tree_core.tree.Tree.deform (
              self )
```

```
Deforms the structure according to the active node.

The deformationSteps that occur as a consequence are saved by the
savers.

Args:
    nothing is taken
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 203 of file tree.py.

5.13.3.4 end()

```
def pkg.tree_core.tree.Tree.end (
              self )
```

```
end() -> True or False.

Args:
    nothing is taken
Returns:
    True if self.activeNode has any valid children,
    False otherwise
Raises:
    Nothing is raised
```

Definition at line 111 of file tree.py.

5.13.3.5 go_down()

```
def pkg.tree_core.tree.Tree.go_down (
              self )
```

```
Changes the activeNode to its first valid child.

If there isn't any valid child, the activeNode is the rightest child
and the function raises a StopIteration error.

Args:
    nothing is taken
Returns:
    nothing is returned
Raises:
    StopIteration
```

Definition at line 129 of file tree.py.

5.13.3.6 go_right()

```
def pkg.tree_core.tree.Tree.go_right (
              self )
```

```
Changes the activeNode to its right neighbour.

If the neighbour doesn't exist, a StopIteration exception is raised.

Args:
    nothing is taken
Returns:
    nothing is returned
Raises:
    StopIteration
```

Definition at line 181 of file tree.py.

5.13.3.7 go_up()

```
def pkg.tree_core.tree.Tree.go_up (
              self )
```

```
Changes the activeNode to its parent.

The parent becomes the activeNode and the structure is consistently
undeformed restoring the state before the deformation of self.

Args:
    nothing is taken
Returns:
    nothing is returned
Raises:
    Nothing is raised
```

Definition at line 159 of file tree.py.

5.13.3.8 print()

```
def pkg.tree_core.tree.Tree.print (
              self )
```

```
Print in detail the current Order of Deformation, as saved.

The dictionary self.savers[0].ood contains, for each component, the
list of isdh.DeformationStep objects from the tree root to
self.activeNode.
self.savers[0].ood = {  isdh-comp1 : [DeformationStep1,
                                      DeformationStep2,
                                      ...],
                isdh-comp2 : [DeformationStep1,
                                      DeformationStep2,
                                      ...],
                isdh-comp3 : [DeformationStep1,
                                      DeformationStep2,
                                      ...],
             }
The content of this dictionary is printed in detail.

Args:
    nothing is taken
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 45 of file tree.py.

5.13.3.9 surf()

```
def pkg.tree_core.tree.Tree.surf (
              self,
              blackbox )
```

```
Changes the activeNode.

It surfs the tree going down or right according to the blackbox
response.

Args:
    blackbox:
a function that decides whether self.activeNode is the valid
next deformationStep or not.
Returns:
    True, if self.activeNode or one of its neighbours was the correct
    one.
    False, if neither self.activeNode nor one of its neighbours was the
    correct one.
Raises:
    exceptions raised by the blackbox, remain unhandled.
```

Definition at line 249 of file tree.py.

5.13.3.10 undeform()

```
def pkg.tree_core.tree.Tree.undeform (
              self )
```

```
Undeforms the structure according to the active node.

The deformationSteps that are undone as a consequence are unsaved by
the savers.

Args:
    nothing is taken
Returns:
    nothing is returned
Raises:
    nothing is raised
```

Definition at line 227 of file tree.py.

5.13.4 Member Data Documentation

5.13.4.1 activeNode

```
pkg.tree_core.tree.Tree.activeNode
```

Definition at line 17 of file tree.py.

5.13.4.2 root

```
pkg.tree_core.tree.Tree.root
```

Definition at line 16 of file tree.py.

5.13.4.3 savers

`pkg.tree_core.tree.Tree.savers`

Definition at line 19 of file tree.py.

5.13.4.4 structure

`pkg.tree_core.tree.Tree.structure`

Definition at line 18 of file tree.py.

The documentation for this class was generated from the following file:

- pkg/tree_core/tree.py

Chapter 6

File Documentation

6.1 pkg/__init__.py File Reference

Namespaces

- pkg

6.2 pkg/isdh/__init__.py File Reference

Namespaces

- pkg.isdh

6.3 pkg/structure_core/__init__.py File Reference

Namespaces

- pkg.structure_core

6.4 pkg/tree_core/__init__.py File Reference

Namespaces

- pkg.tree_core

6.5 Visualization/__init__.py File Reference

Namespaces

- Visualization

6.6 pkg/isdh/component.py File Reference

Classes

- class pkg.isdh.component.Component

Namespaces

- pkg.isdh.component

6.7 pkg/structure_core/Component.py File Reference

Classes

- class pkg.structure_core.Component.Component

Namespaces

- pkg.structure_core.Component

Variables

- pkg.structure_core.Component.logger = logging.getLogger('component')
- pkg.structure_core.Component.level
- tuple pkg.structure_core.Component.BLACK = (0, 0, 0)
- tuple pkg.structure_core.Component.WHITE = (255, 255, 255)
- tuple pkg.structure_core.Component.RED = (255, 0, 0)
- tuple pkg.structure_core.Component.GREEN = (0, 255, 0)
- tuple pkg.structure_core.Component.BLUE = (0, 0, 255)
- tuple pkg.structure_core.Component.LIGHT_BLUE = (102, 255, 255)
- tuple pkg.structure_core.Component.DARK_GREEN = (0, 100, 0)

6.8 pkg/isdh/deformation_step.py File Reference

Classes

- class pkg.isdh.deformation_step.DeformationStep

Namespaces

- pkg.isdh.deformation_step

Functions

- def pkg.isdh.deformation_step.__repr__ (self)

LEGEND:

'd': the element deforms

'm': the element moves

'b': the element breaks

- def pkg.isdh.deformation_step.__eq__ (self, other)
- def pkg.isdh.deformation_step.__ne__ (self, other)
- def pkg.isdh.deformation_step.print (self)

Variables

- pkg.isdh.deformation_step.transformation

6.9 pkg/isdh/example_main.py File Reference

Namespaces

- pkg.isdh.example_main

Variables

- pkg.isdh.example_main.structure = s.read_xml('some_folder/my_xml_file.xml')
- int pkg.isdh.example_main.n = 0

  solve the structure
- pkg.isdh.example_main.frames_per_mm

6.10 pkg/isdh/isdh_helper.py File Reference

Classes

- class pkg.isdh.isdh_helper.IsdhHelper

Namespaces

- pkg.isdh.isdh_helper

6.11 pkg/read_xml.py File Reference

Namespaces

- pkg.read_xml

Functions

- def pkg.read_xml.read_xml (path)
- def pkg.read_xml.add_nodes (root, struct)
- def pkg.read_xml.add_components (root, struct)
- def pkg.read_xml.gaps_insertor (structure)
- def pkg.read_xml.gap_name (level)
- def pkg.read_xml.init_firewall_and_barrier (structure)

Variables

- pkg.read_xml.node

  add gaps in front of the loadpath
- pkg.read_xml.frontNode
- pkg.read_xml.gap = Component. \
- pkg.read_xml.name = gap_name(loadpath.level)

  add gaps between components create a gap_name iterator (e.g.
- pkg.read_xml.listNodes = list(loadpath.setNodes)
- pkg.read_xml.key
- pkg.read_xml.nodes
- pkg.read_xml.next_nodes
- pkg.read_xml.ignore_me = next(next_nodes)
- list pkg.read_xml.lp_levels
- pkg.read_xml.rightLimit
- pkg.read_xml.backNode

6.12 pkg/structure_core/CrossComponent.py File Reference

Classes

- class pkg.structure_core.CrossComponent.CrossComponent

Namespaces

- pkg.structure_core.CrossComponent

Variables

- pkg.structure_core.CrossComponent.logger = logging.getLogger('CrossComponentLogger')
- tuple pkg.structure_core.CrossComponent.BLACK = (0, 0, 0)
- tuple pkg.structure_core.CrossComponent.WHITE = (255, 255, 255)
- tuple pkg.structure_core.CrossComponent.RED = (255, 0, 0)
- tuple pkg.structure_core.CrossComponent.GREEN = (0, 255, 0)
- tuple pkg.structure_core.CrossComponent.BLUE = (0, 0, 255)
- tuple pkg.structure_core.CrossComponent.DARK_GREEN = (0, 100, 0)

6.13 pkg/structure_core/Loadpath.py File Reference

Classes

- class pkg.structure_core.Loadpath.Loadpath

Namespaces

- pkg.structure_core.Loadpath

Functions

- def pkg.structure_core.Loadpath.valid_components (self)

  def add_member(self, component): """Function addes components to the list of components """ self.list↩ Components.append(component)

6.14 pkg/structure_core/Node.py File Reference

Classes

- class pkg.structure_core.Node.Node

Namespaces

- pkg.structure_core.Node

Variables

- tuple pkg.structure_core.Node.BLACK = (0, 0, 0)
- tuple pkg.structure_core.Node.RED = (255, 0, 0)
- tuple pkg.structure_core.Node.DARK_GREEN = (0, 100, 0)
- pkg.structure_core.Node.logger = logging.getLogger('node')
- pkg.structure_core.Node.level

6.15 pkg/structure_core/Structure.py File Reference

Classes

- class pkg.structure_core.Structure.Structure

Namespaces

- pkg.structure_core.Structure

Variables

- bool pkg.structure_core.Structure.DEBUG = False

  import itertools from ..tree_core.tree import Tree from .
- bool pkg.structure_core.Structure.STEPWISE = False
- list pkg.structure_core.Structure.size = [1500, 500]
- pkg.structure_core.Structure.screen = pygame.display.set_mode(size)
- tuple pkg.structure_core.Structure.BLACK = (0, 0, 0)
- tuple pkg.structure_core.Structure.WHITE = (255, 255, 255)
- tuple pkg.structure_core.Structure.RED = (255, 0, 0)
- tuple pkg.structure_core.Structure.GREEN = (0, 255, 0)
- tuple pkg.structure_core.Structure.BLUE = (0, 0, 255)

6.16 pkg/tree_core/NodeTree.py File Reference

Classes

- class pkg.tree_core.NodeTree.NodeTree

Namespaces

- pkg.tree_core.NodeTree

6.17 pkg/tree_core/tree.py File Reference

Classes

- class pkg.tree_core.tree.Tree

Namespaces

- pkg.tree_core.tree

Variables

- bool pkg.tree_core.tree.PRINT = False

6.18 pkg/write_xml.py File Reference

Namespaces

- pkg.write_xml

Functions

- def pkg.write_xml.create_level (root, index)
- def pkg.write_xml.create_component (level, input_name, input_x1, input_x2, input_defoLength, input_end↩ _lp=None)
- def pkg.write_xml.ask_for_new_level (root)
- def pkg.write_xml.ask_for_new_member (level, lp_i)
- def pkg.write_xml.ask_for_new_connection (level, lp_i)
- def pkg.write_xml.prettify (path)
- def pkg.write_xml.new_xml ()

6.19 Visualization/BlenderObject.py File Reference

Classes

- class Visualization.BlenderObject.BlenderObject

Namespaces

- Visualization.BlenderObject

Variables

- Visualization.BlenderObject.meshElement = bpy.ops.mesh.primitive_cube_add
- Visualization.BlenderObject.meshMass = bpy.ops.mesh.primitive_uv_sphere_add
- Visualization.BlenderObject.meshText = bpy.ops.object.text_add

6.20 Visualization/CreateVideo.py File Reference

Classes

- class Visualization.CreateVideo.CreateVideo

Namespaces

- Visualization.CreateVideo

6.21 Visualization/initialization.py File Reference

Namespaces

- Visualization.initialization

Functions

- def Visualization.initialization.initialize ()
- def Visualization.initialization.static_numberOfElement ()

6.22 Visualization/Member.py File Reference

Classes

- class Visualization.Member.generalMember

Namespaces

- Visualization.Member

6.23 Visualization/setCamera.py File Reference

Namespaces

- Visualization.setCamera

Functions

- def Visualization.setCamera.setCamera (scene, x, y, z, lents)

6.24 Visualization/setColor.py File Reference

Namespaces

- Visualization.setColor

Functions

- def Visualization.setColor.setColor (ob, color)
- def Visualization.setColor.makeColor (name, diffuse)

Variables

- def Visualization.setColor.red = makeColor('Red', (1,0,0))
- def Visualization.setColor.blue = makeColor('Blue', (0,0,1))
- def Visualization.setColor.black = makeColor('Black',(0,0,0))
- def Visualization.setColor.white = makeColor('White',(1,1,1))
- def Visualization.setColor.green = makeColor('Green',(0,1,0))
- def Visualization.setColor.gray = makeColor('Gray', (0.6,0.6,0.6))
- def Visualization.setColor.dark_gray = makeColor('Gray', (0.2,0.2,0.2))

6.25 Visualization/setFunction.py File Reference

Namespaces

- Visualization.setFunction

Functions

- def Visualization.setFunction.movement (object, initialFrame, finalFrame, amount, offset=0)
- def Visualization.setFunction.rotation (object, initialFrame, finalFrame, amount)
- def Visualization.setFunction.deformation (object, initialFrame, finalFrame, amount)
- def Visualization.setFunction.color (object, initialFrame, color)
- def Visualization.setFunction.elimination (object, initialFrame, finalFrame)
- def Visualization.setFunction.interpolation (object)

6.26 Visualization/setLamp.py File Reference

Namespaces

- Visualization.setLamp

Functions

- def Visualization.setLamp.setLamp (scene, x, y, z)

6.27 Visualization/setRender.py File Reference

Namespaces

- Visualization.setRender

Functions

- def Visualization.setRender.Parameters (numberOfFrames, resolution, locationWall, locationBackground, width, height, pathDirectory)

Index