Rune

0.0.1

Generated by Doxygen 1.8.17

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

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

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Namespace Documentation

5.1 helpers Namespace Reference

Functions

- def distance (x1, y1, x2, y2)
- def midpoint (x1, y1, x2, y2)

5.1.1 Detailed Description

```
@package docstring
helpers.py
@date Jan 8, 2022
@author Harper Weigle
@brief miscellaneous functions that are heavily repeated to clean up main code base. Mostly common equations
```

5.1.2 Function Documentation

5.1.2.1 distance()

```
def helpers.distance (
              x1,
              y1,
              x2,
              y2 )
```

```
@brief      calculates the distance between two points. d=sqrt((x2-x1)^2+(y2-y1)^2)
@param      x1 - x coordinate of point 1; Type: float
            y1 - y coordinate of point 1; Type: float
            x2 - x coordinate of point 2; Type: float
            y2 - y coordinate of point 2; Type: float
@returns    result of the distance equation
```

5.1.2.2 midpoint()

```
def helpers.midpoint (
              x1,
              y1,
              x2,
              y2 )
```

```
@brief      calculates the midpoint between two points. (xm,ym)=((x1+x2)/2,(y1+y2)/2)
@param      x1 - x coordinate of point 1; Type: float
            y1 - y coordinate of point 1; Type: float
            x2 - x coordinate of point 2; Type: float
            y2 - y coordinate of point 2; Type: float
@returns    result of the midpoint equation
```

5.2 runes Namespace Reference

Classes

- class Fire

5.2.1 Detailed Description

```
@package docstring
runes.py
@date Jan 8, 2022
@author Harper Weigle
@brief file to contain rune specific equations
```

5.3 runetracetest Namespace Reference

Classes

- class App

Functions

- def main ()

Variables

- int CANVASWIDTH = 800
- int CANVASHEIGHT = 400
- int RESOLUTION = 100
- int WIGGLE = 20

5.3.1 Detailed Description

```
@package docstring
runetracetest.py
@date Jan 8, 2022
@author Harper Weigle
@brief main app for rune trace test. Handles main loop and core app functions
```

5.3.2 Function Documentation

5.3.2.1 main()

```
def runetracetest.main ( )
```

```
@brief      initializes app, starts app.update recursive cycle, executes mainloop
@param     None
@returns    None
```

5.3.3 Variable Documentation

5.3.3.1 CANVASHEIGHT

```
int runetracetest.CANVASHEIGHT = 400
```

5.3.3.2 CANVASWIDTH

```
int runetracetest.CANVASWIDTH = 800
```

5.3.3.3 RESOLUTION

```
int runetracetest.RESOLUTION = 100
```

5.3.3.4 WIGGLE

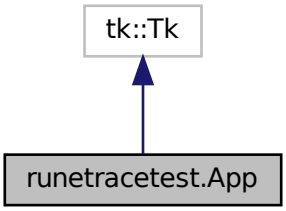
```
int runetracetest.WIGGLE = 20
```

Chapter 6

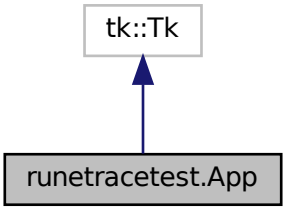
Class Documentation

6.1 runetracetest.App Class Reference

Inheritance diagram for runetracetest.App:



Collaboration diagram for runetracetest.App:



Public Member Functions

- def __init__ (self)
- def update (self)
- def drawCoordPlane (self)
- def checkCursor (self, e)
- def drawFireRune (self)
- def zeroToTrueCoord (self, x, y)
- def trueToZeroCoord (self, x, y)

Public Attributes

- canvas
- scale
- scale_mod
- status
- statusLabel

6.1.1 Detailed Description

```
@brief  core app class handles tkinter root functions as well as core app functions. Must be initialized befor
        Child of tk.Tk
```

6.1.2 Constructor & Destructor Documentation

6.1.2.1 __init__()

```
def runetracetest.App.__init__ (
              self )
```

```
@brief      Initializes:
    super()
    title - tk app title; Type: tk.title
    canvas - app window; Type: tk.canvas
    scale - starting value of relative fib sequence; Type: int
    scale_mod - 0 = Null, >0 = Grow, <0 = Shrink; Type: int
    status - stage of rune user is in; Type: tk.StringVar
    statusLabel - status displayed on canvas; Type: tk.Label

    binds mouse motion to checkCursor()
@param      None
@returns    None
```

6.1.3 Member Function Documentation

6.1.3.1 checkCursor()

```
def runetracetest.App.checkCursor (
              self,
              e )
```

```
@brief      Checks cursor location relative to rune. Updates status, scale_mod, and scale
@param     e - trigger event; Type: tk.Event
@returns    None
```

6.1.3.2 drawCoordPlane()

```
def runetracetest.App.drawCoordPlane (
              self )
```

```
@brief      Draws cartesian plane
@param     None
@returns    None
```

6.1.3.3 drawFireRune()

```
def runetracetest.App.drawFireRune (
              self )
```

```
@brief      draws fire rune based on stages defined in rune.py
@param     None
@returns    None
```

6.1.3.4 trueToZeroCoord()

```
def runetracetest.App.trueToZeroCoord (
              self,
              x,
              y )
```

```
@brief      Converts coordinates relative to the canvas to coordinates relative to the cartesian plane
@param     x - x coordinate; Type: float
    y - y coordinate; Type: float
@returns    Converted coordinates; Type: tuple(float,float)
```

6.1.3.5 update()

```
def runetracetest.App.update (
              self )


@brief      clears canvas, redraws canvas items, and reschedules self
@param     None
@returns    None
```

6.1.3.6 zeroToTrueCoord()

```
def runetracetest.App.zeroToTrueCoord (
              self,
              x,
              y )


@brief      Converts coordinates relative to the cartesian plane to coordinates relative to the canvas
@param     x - x coordinate; Type: float
    y - y coordinate; Type: float
@returns    Converted coordinates; Type: tuple(float,float)
```

6.1.4 Member Data Documentation

6.1.4.1 canvas

```
runetracetest.App.canvas
```

6.1.4.2 scale

```
runetracetest.App.scale
```

6.1.4.3 scale_mod

```
runetracetest.App.scale_mod
```

6.1.4.4 status

```
runetracetest.App.status
```

6.1.4.5 statusLabel

```
runetracetest.App.statusLabel
```

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

- python/runetracetest.py

6.2 runes.Fire Class Reference

Public Member Functions

- def stageOneY (x, s1)
- def stageOneX (y, s1)
- def stageTwoY (x, s1, s2, s3)
- def stageTwoX (y, s1, s2)
- def stageThreeY (x, s1, s3)
- def stageThreeX (y, s1, s3, s4)
- def stageFourY (x, s4)
- def stageFourX (y, s4)
- def stageFiveY (x, s4)
- def stageFiveX (y, s4)
- def stageSixY (x, s1)
- def stageSixX (y, s1, s4)

6.2.1 Detailed Description

```
@brief class has no initialization because it does not store member variables. It simply organizes stages of t
```

6.2.2 Member Function Documentation

6.2.2.1 stageFiveX()

```
def runes.Fire.stageFiveX (
              y,
              s4 )

@brief      calculates stage five given y: x=-sqrt(s4^2-(y-1)^2)
@param      y - y coordinate of rune stage; Type: float
    s4 - fourth value in relative fib sequence; Type: float
@returns    None if y > s4+1 or if y < 0. Result of equation otherwise
```

6.2.2.2 stageFiveY()

```
def runes.Fire.stageFiveY (
              x,
              s4 )


@brief      calculates stage five given x: y=sqrt(s4^2-x^2)+1
@param      x - x coordinate of rune stage; Type: float
    s4 - fourth value in relative fib sequence; Type: float
@returns    None if x > 0 or if x < -s4. Result of equation otherwise
```

6.2.2.3 stageFourX()

```
def runes.Fire.stageFourX (
              y,
              s4 )


@brief      calculates stage four given y: x=-sqrt(s4^2-(y-1)^2)
@param      y - y coordinate of rune stage; Type: float
    s4 - fourth value in relative fib sequence; Type: float
@returns    None if y > 0 or if y < -(s4-1). Result of equation otherwise
```

6.2.2.4 stageFourY()

```
def runes.Fire.stageFourY (
              x,
              s4 )


@brief      calculates stage four given x: y=-sqrt(s4^2-x^2)+1
@param      x - x coordinate of rune stage; Type: float
    s4 - fourth value in relative fib sequence; Type: float
@returns    None if x > 0 or if x < -s4. Result of equation otherwise
```

6.2.2.5 stageOneX()

```
def runes.Fire.stageOneX (
              y,
              s1 )


@brief      calculates stage one given y: x=-sqrt(s1^2-(y+(s1-1)^2)-s1)
@param      y - y coordinate of rune stage; Type: float
    s1 - first value in relative fib sequence; Type: float
@returns    None if y > 1 or if y < -(s1-1). Result of equation otherwise
```

6.2.2.6 stageOneY()

```
def runes.Fire.stageOneY (
              x,
              s1 )

@brief      calculates stage one given x: y=sqrt(s1^2-(x-s1)^2)-(s1-1)
@param      x - x coordinate of rune stage; Type: float
    s1 - first value in relative fib sequence; Type: float
@returns    None if x > s1 or if x < 0. Result of equation otherwise
```

6.2.2.7 stageSixX()

```
def runes.Fire.stageSixX (
              y,
              s1,
              s4 )

@brief      calculates stage five given y: \f\x=sqrt{(s_1^2-(y-(7+4(s_1)))^2)}\fmath.sqrt(s1^2-y-(7+4(s1-1))^2
@param      y - y coordinate of rune stage; Type: float
    s1 - first value in relative fib sequence; Type: float
    s4 - fourth value in relative fib sequence; Type: float
@returns    None if y > 7+4(s_1-1) or if y < s4. Result of equation otherwise
```

6.2.2.8 stageSixY()

```
def runes.Fire.stageSixY (
              x,
              s1 )

@brief      calculates stage six given x: y=-sqrt(s1^2-x^2)+7+4(s1-1)
@param      x - x coordinate of rune stage; Type: float
    s1 - first value in relative fib sequence; Type: float
@returns    None if x > s1 or if x < 0. Result of equation otherwise
```

6.2.2.9 stageThreeX()

```
def runes.Fire.stageThreeX (
              y,
              s1,
              s3,
              s4 )

@brief      calculates stage three given y: x=sqrt(s3^2-(y+s1)^2)
@param      y - y coordinate of rune stage; Type: float
    s1 - first value in relative fib sequence; Type: float
    s3 - third value in relative fib sequence; Type: float
    s4 - fourth value in relative fib sequence; Type: float
@returns    None if y > -s1 or if y < -(s4-1). Result of equation otherwise
```

6.2.2.10 stageThreeY()

```
def runes.Fire.stageThreeY (
              x,
              s1,
              s3 )


@brief      calculates stage three given x: y=-sqrt(s3^2-x^2)-s1
@param      x - x coordinate of rune stage; Type: float
    s1 - first value in relative fib sequence; Type: float
    s3 - third value in relative fib sequence; Type: float
@returns    None if x > s3 or if x < 0. Result of equation otherwise
```

6.2.2.11 stageTwoX()

```
def runes.Fire.stageTwoX (
              y,
              s1,
              s2 )


@brief      calculates stage two given y: x=sqrt(s2^2-(y+s1)^2)+s1
@param      y - y coordinate of rune stage; Type: float
    s1 - first value in relative fib sequence; Type: float
    s2 - second value in relative fib sequence; Type: float
@returns    None if y > 1 or if y < -(s1-1). Result of equation otherwise
```

6.2.2.12 stageTwoY()

```
def runes.Fire.stageTwoY (
              x,
              s1,
              s2,
              s3 )


@brief      calculates stage two given x: y=sqrt(s2^2-(x-s1)^2)-s1
@param      x - x coordinate of rune stage; Type: float
    s1 - first value in relative fib sequence; Type: float
    s2 - second value in relative fib sequence; Type: float
    s3 - third value in relative fib sequence; Type: float
@returns    None if x > s3 or if x < s1. Result of equation otherwise
```

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

- python/runes.py

Chapter 7

File Documentation

7.1 python/helpers.py File Reference

Namespaces

- helpers

Functions

- def helpers.distance (x1, y1, x2, y2)
- def helpers.midpoint (x1, y1, x2, y2)

7.2 python/runes.py File Reference

Classes

- class runes.Fire

Namespaces

- runes

7.3 python/runetracetest.py File Reference

Classes

- class runetracetest.App

Namespaces

- runetracetest

Functions

- def runetracetest.main ()

Variables

- int runetracetest.CANVASWIDTH = 800
- int runetracetest.CANVASHEIGHT = 400
- int runetracetest.RESOLUTION = 100
- int runetracetest.WIGGLE = 20