Powered by GenMyModel

Snake

My Company

Author: Tim Barber

Date: 2019-02-02

Table of Contents

1. Class Diagrams

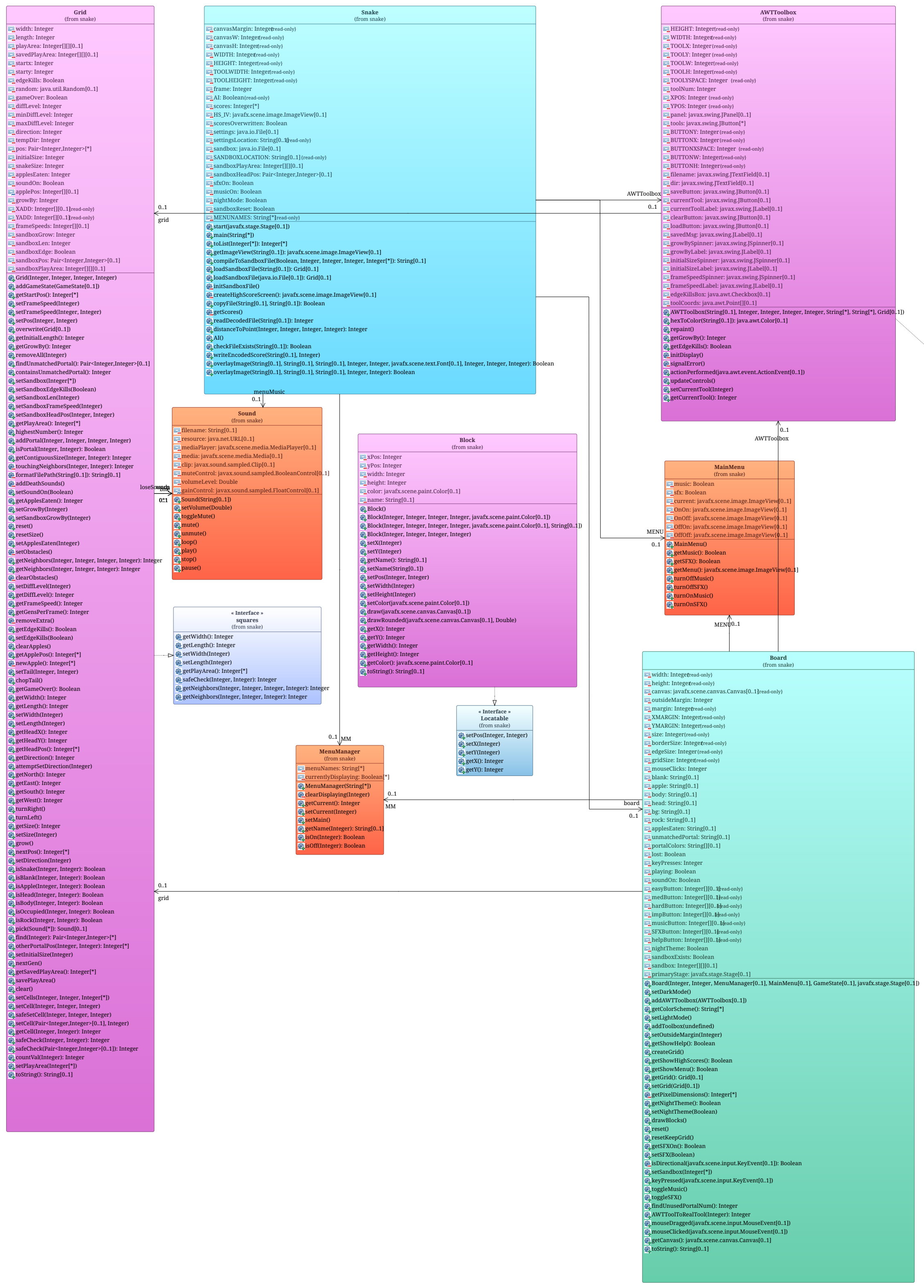
Grid
(from snake)
width: Integer
length: Integer
playArea: Integer[][][0..1]
savedPlayArea: Integer[][][0..1]
startx: Integer
starty: Integer
edgeKills: Boolean
random: java.util.Random[0..1]
gameOver: Boolean
diffLevel: Integer
minDiffLevel: Integer
maxDiffLevel: Integer
direction: Integer
tempDir: Integer
pos: Pair<Integer,Integer>[*]
initialSize: Integer
snakeSize: Integer
applesEaten: Integer
soundOn: Boolean
applePos: Integer[][0..1]
growBy: Integer
XADD: Integer[][0..1](read-only)
YADD: Integer[][0..1](read-only)
frameSpeeds: Integer[][0..1]
sandboxGrow: Integer
sandboxLen: Integer
sandboxEdge: Boolean
sandboxPos: Pair<Integer,Integer>[0..1]
sandboxPlayArea: Integer[][][0..1]
Grid(Integer, Integer, Integer, Integer)
addGameState(GameState[0..1])
getStartPos(): Integer[*]
setFrameSpeed(Integer)
setFrameSpeed(Integer, Integer)
setPos(Integer, Integer)
overwrite(Grid[0..1])
getInitialLength(): Integer
getGrowBy(): Integer
removeAll(Integer)
findUnmatchedPortal(): Pair<Integer,Integer>[0..1]
containsUnmatchedPortal(): Integer
setSandbox(Integer[*])
setSandboxEdgeKills(Boolean)
setSandboxLen(Integer)
setSandboxFrameSpeed(Integer)
setSandboxHeadPos(Integer, Integer)
getPlayArea(): Integer[*]
highestNumber(): Integer
addPortal(Integer, Integer, Integer, Integer)
isPortal(Integer, Integer): Boolean
getContiguousSize(Integer, Integer): Integer
touchingNeighbors(Integer, Integer): Integer
formatFilePath(String[0..1]): String[0..1]
addDeathSounds()
setSoundOn(Boolean)
getApplesEaten(): Integer
setGrowBy(Integer)
setSandboxGrowBy(Integer)
reset()
resetSize()
setApplesEaten(Integer)
setObstacles()
getNeighbors(Integer, Integer, Integer, Integer): Integer
getNeighbors(Integer, Integer, Integer): Integer
clearObstacles()
setDiffLevel(Integer)
getDiffLevel(): Integer
getFrameSpeed(): Integer
getGensPerFrame(): Integer
removeExtra()
getEdgeKills(): Boolean
setEdgeKills(Boolean)
clearApples()
getApplePos(): Integer[*]
newApple(): Integer[*]
setTail(Integer, Integer)
chopTail()
getGameOver(): Boolean
getWidth(): Integer
getLength(): Integer
setWidth(Integer)
setLength(Integer)
getHeadX(): Integer
getHeadY(): Integer
getHeadPos(): Integer[*]
getDirection(): Integer
attemptSetDirection(Integer)
getNorth(): Integer
getEast(): Integer
getSouth(): Integer
getWest(): Integer
turnRight()
turnLeft()
getSize(): Integer
setSize(Integer)
grow()
nextPos(): Integer[*]
setDirection(Integer)
isSnake(Integer, Integer): Boolean
isBlank(Integer, Integer): Boolean
isApple(Integer, Integer): Boolean
isHead(Integer, Integer): Boolean
isBody(Integer, Integer): Boolean
isOccupied(Integer, Integer): Boolean
isRock(Integer, Integer): Boolean
pick(Sound[*]): Sound[0..1]
find(Integer): Pair<Integer,Integer>[*]
otherPortalPos(Integer, Integer): Integer[*]
setInitialSize(Integer)
nextGen()
getSavedPlayArea(): Integer[*]
savePlayArea()
clear()
setCells(Integer, Integer, Integer[*])
setCell(Integer, Integer, Integer)
safeSetCell(Integer, Integer, Integer)
setCell(Pair<Integer,Integer>[0..1], Integer)
getCell(Integer, Integer): Integer
safeCheck(Integer, Integer): Integer
safeCheck(Pair<Integer,Integer>[0..1]): Integer
countVal(Integer): Integer
setPlayArea(Integer[*])
toString(): String[0..1]
Snake
(from snake)
canvasMargin: Integer(read-only)
canvasW: Integer(read-only)
canvasH: Integer(read-only)
WIDTH: Integer(read-only)
HEIGHT: Integer(read-only)
TOOLWIDTH: Integer(read-only)
TOOLHEIGHT: Integer(read-only)
frame: Integer
AI: Boolean(read-only)
scores: Integer[*]
HS_IV: javafx.scene.image.ImageView[0..1]
scoresOverwritten: Boolean
settings: java.io.File[0..1]
settingsLocation: String[0..1](read-only)
sandbox: java.io.File[0..1]
SANDBOXLOCATION: String[0..1](read-only)
sandboxPlayArea: Integer[][][0..1]
sandboxHeadPos: Pair<Integer,Integer>[0..1]
sfxOn: Boolean
musicOn: Boolean
nightMode: Boolean
sandboxReset: Boolean
MENUNAMES: String[*](read-only)
start(javafx.stage.Stage[0..1])
main(String[*])
toList(Integer[*]): Integer[*]
getImageView(String[0..1]): javafx.scene.image.ImageView[0..1]
compileToSandboxFile(Boolean, Integer, Integer, Integer, Integer[*]): String[0..1]
loadSandboxFile(String[0..1]): Grid[0..1]
loadSandboxFile(java.io.File[0..1]): Grid[0..1]
initSandboxFile()
createHighScoreScreen(): javafx.scene.image.ImageView[0..1]
copyFile(String[0..1], String[0..1]): Boolean
getScores()
readDecodedFile(String[0..1]): Integer
distanceToPoint(Integer, Integer, Integer, Integer): Integer
AI()
checkFileExists(String[0..1]): Boolean
writeEncodedScore(String[0..1], Integer)
overlayImage(String[0..1], String[0..1], String[0..1], Integer, Integer, javafx.scene.text.Font[0..1], Integer, Integer, Integer): Boolean
overlayImage(String[0..1], String[0..1], String[0..1], Integer, Integer): Boolean
AWTToolbox
(from snake)
HEIGHT: Integer(read-only)
WIDTH: Integer(read-only)
TOOLX: Integer(read-only)
TOOLY: Integer(read-only)
TOOLW: Integer(read-only)
TOOLH: Integer(read-only)
TOOLSPACE: Integer (read-only)
toolNum: Integer
XPOS: Integer (read-only)
YPOS: Integer (read-only)
panel: javax.swing.JPanel[0..1]
tools: javax.swing.JButton[*]
BUTTONY: Integer(read-only)
BUTTONX: Integer(read-only)
BUTTONXSPACE: Integer (read-only)
BUTTONW: Integer(read-only)
BUTTONH: Integer(read-only)
filename: javax.swing.JTextField[0..1]
dir: javax.swing.JTextField[0..1]
saveButton: javax.swing.JButton[0..1]
currentTool: javax.swing.JButton[0..1]
currentToolLabel: javax.swing.JLabel[0..1]
clearButton: javax.swing.JButton[0..1]
loadButton: javax.swing.JButton[0..1]
savedMsg: javax.swing.JLabel[0..1]
growBySpinner: javax.swing.JSpinner[0..1]
growByLabel: javax.swing.JLabel[0..1]
initialSizeSpinner: javax.swing.JSpinner[0..1]
initialSizeLabel: javax.swing.JLabel[0..1]
frameSpeedSpinner: javax.swing.JSpinner[0..1]
frameSpeedLabel: javax.swing.JLabel[0..1]
edgeKillsBox: java.awt.Checkbox[0..1]
toolCoords: java.awt.Point[][0..1]
AWTToolbox(String[0..1], Integer, Integer, Integer, Integer, String[*], String[*], Grid[0..1])
hexToColor(String[0..1]): java.awt.Color[0..1]
repaint()
getGrowBy(): Integer
getEdgeKills(): Boolean
initDisplay()
signalError()
actionPerformed(java.awt.event.ActionEvent[0..1])
updateControls()
setCurrentTool(Integer)
getCurrentTool(): Integer
« dependency »
FilePicker
(from snake)
filename: javax.swing.JTextField[0..1]
dir: javax.swing.JTextField[0..1]
open: javax.swing.JButton[0..1]
save: javax.swing.JButton[0..1]
FilePicker()
main(String[*])
run(javax.swing.JFrame[0..1], Integer, Integer)
Sound
(from snake)
filename: String[0..1]
resource: java.net.URL[0..1]
mediaPlayer: javafx.scene.media.MediaPlayer[0..1]
media: javafx.scene.media.Media[0..1]
clip: javax.sound.sampled.Clip[0..1]
muteControl: javax.sound.sampled.BooleanControl[0..1]
volumeLevel: Double
gainControl: javax.sound.sampled.FloatControl[0..1]
Sound(String[0..1])
setVolume(Double)
toggleMute()
mute()
unmute()
loop()
play()
stop()
pause()
Block
(from snake)
xPos: Integer
yPos: Integer
width: Integer
height: Integer
color: javafx.scene.paint.Color[0..1]
name: String[0..1]
Block()
Block(Integer, Integer, Integer, Integer, javafx.scene.paint.Color[0..1])
Block(Integer, Integer, Integer, Integer, javafx.scene.paint.Color[0..1], String[0..1])
Block(Integer, Integer, Integer, Integer)
setX(Integer)
setY(Integer)
getName(): String[0..1]
setName(String[0..1])
setPos(Integer, Integer)
setWidth(Integer)
setHeight(Integer)
setColor(javafx.scene.paint.Color[0..1])
draw(javafx.scene.canvas.Canvas[0..1])
drawRounded(javafx.scene.canvas.Canvas[0..1], Double)
getX(): Integer
getY(): Integer
getWidth(): Integer
getHeight(): Integer
getColor(): javafx.scene.paint.Color[0..1]
toString(): String[0..1]
« Interface »
squares
(from snake)
getWidth(): Integer
getLength(): Integer
setWidth(Integer)
setLength(Integer)
getPlayArea(): Integer[*]
safeCheck(Integer, Integer): Integer
getNeighbors(Integer, Integer, Integer, Integer): Integer
getNeighbors(Integer, Integer, Integer): Integer
« Interface »
Locatable
(from snake)
setPos(Integer, Integer)
setX(Integer)
setY(Integer)
getX(): Integer
getY(): Integer
MainMenu
(from snake)
music: Boolean
sfx: Boolean
current: javafx.scene.image.ImageView[0..1]
OnOn: javafx.scene.image.ImageView[0..1]
OnOff: javafx.scene.image.ImageView[0..1]
OffOn: javafx.scene.image.ImageView[0..1]
OffOff: javafx.scene.image.ImageView[0..1]
MainMenu()
getMusic(): Boolean
getSFX(): Boolean
getMenu(): javafx.scene.image.ImageView[0..1]
turnOffMusic()
turnOffSFX()
turnOnMusic()
turnOnSFX()
MenuManager
(from snake)
menuNames: String[*]
currentlyDisplaying: Boolean[*]
MenuManager(String[*])
clearDisplaying(Integer)
getCurrent(): Integer
setCurrent(Integer)
setMain()
getName(Integer): String[0..1]
isOn(Integer): Boolean
isOff(Integer): Boolean
Board
(from snake)
width: Integer(read-only)
height: Integer(read-only)
canvas: javafx.scene.canvas.Canvas[0..1](read-only)
outsideMargin: Integer
margin: Integer(read-only)
XMARGIN: Integer(read-only)
YMARGIN: Integer(read-only)
size: Integer(read-only)
borderSize: Integer(read-only)
edgeSize: Integer(read-only)
gridSize: Integer(read-only)
mouseClicks: Integer
blank: String[0..1]
apple: String[0..1]
body: String[0..1]
head: String[0..1]
bg: String[0..1]
rock: String[0..1]
applesEaten: String[0..1]
unmatchedPortal: String[0..1]
portalColors: String[][0..1]
lost: Boolean
keyPresses: Integer
playing: Boolean
soundOn: Boolean
easyButton: Integer[][0..1](read-only)
medButton: Integer[][0..1](read-only)
hardButton: Integer[][0..1](read-only)
impButton: Integer[][0..1](read-only)
musicButton: Integer[][0..1](read-only)
SFXButton: Integer[][0..1](read-only)
helpButton: Integer[][0..1](read-only)
nightTheme: Boolean
sandboxExists: Boolean
sandbox: Integer[][][0..1]
primaryStage: javafx.stage.Stage[0..1]
Board(Integer, Integer, MenuManager[0..1], MainMenu[0..1], GameState[0..1], javafx.stage.Stage[0..1])
setDarkMode()
addAWTToolbox(AWTToolbox[0..1])
getColorScheme(): String[*]
setLightMode()
addToolbox(undefined)
setOutsideMargin(Integer)
getShowHelp(): Boolean
createGrid()
getShowHighScores(): Boolean
getShowMenu(): Boolean
getGrid(): Grid[0..1]
setGrid(Grid[0..1])
getPixelDimensions(): Integer[*]
getNightTheme(): Boolean
setNightTheme(Boolean)
drawBlocks()
reset()
resetKeepGrid()
getSFXOn(): Boolean
setSFX(Boolean)
isDirectional(javafx.scene.input.KeyEvent[0..1]): Boolean
setSandbox(Integer[*])
keyPressed(javafx.scene.input.KeyEvent[0..1])
toggleMusic()
toggleSFX()
findUnusedPortalNum(): Integer
AWTToolToRealTool(Integer): Integer
mouseDragged(javafx.scene.input.MouseEvent[0..1])
mouseClicked(javafx.scene.input.MouseEvent[0..1])
getCanvas(): javafx.scene.canvas.Canvas[0..1]
toString(): String[0..1]
grid
0..1
AWTToolbox
0..1
menuMusic
0..1
loseSounds
0..1
MENU
0..1
MM
0..1
board
0..1
MENU
0..1
AWTToolbox
0..1
grid
0..1

Figure 1. Snake-diag Diagram

Referenced Elements

- Class snake::Grid — see "Grid" definition

- Class snake::Snake — see "Snake" definition
- Class snake::Sound — see "Sound" definition
- Class snake::AWTToolbox — see "AWTToolbox" definition
- Class snake::Block — see "Block" definition
- Class snake::Board — see "Board" definition
- Class snake::FilePicker — see "FilePicker" definition
- Interface snake::Locatable — see "Locatable" definition
- Class snake::MainMenu — see "MainMenu" definition
- Class snake::MenuManager — see "MenuManager" definition
- Interface snake::squares — see "squares" definition

1.1. Model Snake

No description.

1.2. Package snake

No description.

1.2.1. Class snake::AWTToolbox

@author Tim Barber

Attributes

- HEIGHT : Integer[1] **Read only**

  Description

  Height of the game frame.

- WIDTH : Integer[1] **Read only**

  Description

  Width of the game frame.

- TOOLX : Integer[1] **Read only**
- TOOLY : Integer[1] **Read only**
- TOOLW : Integer[1] **Read only**
- TOOLH : Integer[1] **Read only**
- TOOLYSPACE : Integer[1] **Read only**

- toolNum : Integer[1]
- XPOS : Integer[1] **Read only**
- YPOS : Integer[1] **Read only**
- panel : javax.swing.JPanel[0..1]

  Description

  The main panel

- tools : javax.swing.JButton[*]
- BUTTONY : Integer[1] **Read only**
- BUTTONX : Integer[1] **Read only**
- BUTTONXSPACE : Integer[1] **Read only**
- BUTTONW : Integer[1] **Read only**
- BUTTONH : Integer[1] **Read only**
- filename : javax.swing.JTextField[0..1]
- dir : javax.swing.JTextField[0..1]
- saveButton : javax.swing.JButton[0..1]
- currentTool : javax.swing.JButton[0..1]
- currentToolLabel : javax.swing.JLabel[0..1]
- clearButton : javax.swing.JButton[0..1]
- loadButton : javax.swing.JButton[0..1]
- savedMsg : javax.swing.JLabel[0..1]
- growBySpinner : javax.swing.JSpinner[0..1]
- growByLabel : javax.swing.JLabel[0..1]
- initialSizeSpinner : javax.swing.JSpinner[0..1]
- initialSizeLabel : javax.swing.JLabel[0..1]
- frameSpeedSpinner : javax.swing.JSpinner[0..1]
- frameSpeedLabel : javax.swing.JLabel[0..1]
- edgeKillsBox : java.awt.Checkbox[0..1]
- toolCoords : java.awt.Point[][0..1]

  Description

  The coordinates of the tools

Implemented interfaces

- java.awt.event.ActionListener — [_U_m1UvCCc9EemVJ-LgVHoDQQ]

Super classes

- javax.swing.JFrame — [_U_m1UvDCc9EemVJ-LgVHoDQQ]

Operations

- AWTToolbox(title : String [0..1], width : Integer [1] , height : Integer [1] , xPos : Integer [1] , yPos : Integer [1] , toolColors : String **[] , toolNames : String [**] , grid : Grid [0..1])

  Description

  @paramtitle The name of the window @paramwidth The width of the window @paramheight The height of the window @paramxPos The x-coordinate of the window position @paramyPos The y-coordinate of the window position @paramtoolColors The list of background colors for the tool buttons @paramtoolNames The list of names for the tool buttons @paramgrid The grid object that is being controlled

- hexToColor(hex : String [0..1]) : java.awt.Color

  Description

  @paramhex A six character string containing hex digits representing acolor in the rgb color space @return A Color object with the rgb value of the hex string

- repaint( ) : void

  Description

  Draw the display (cards and messages).

- getGrowBy( ) : Integer

  Description

  @return The value of the grow by spinner

- getEdgeKills( ) : Boolean

  Description

  @return Whether the edge kills checkbox is checked

- initDisplay( ) : void

  Description

  Initialize the display.

- signalError( ) : void

  Description

  Beeps

- actionPerformed(e : java.awt.event.ActionEvent [0..1]) : void
- updateControls( ) : void

  Description
- setCurrentTool(index : Integer [1]) : void

  Description

  @paramindex
- getCurrentTool( ) : Integer

  Description

  @return the index in toolNames of the last clicked button

Associations

- grid : Grid [0..1] — see "Grid" definition

Nested classifiers

- Class OpenListener — see "OpenListener" definition

1.2.2. Class snake::AWTToolbox::OpenListener

No description.

Implemented interfaces

- java.awt.event.ActionListener — [_U_m1UvCCc9EemVJ-LgVHoDQQ]

Operations

- actionPerformed(e : java.awt.event.ActionEvent [0..1]) : void

1.2.3. Class snake::Block

@author Timothy

Attributes

- xPos : Integer[1]
- yPos : Integer[1]
- width : Integer[1]
- height : Integer[1]
- color : javafx.scene.paint.Color[0..1]
- name : String[0..1]

Implemented interfaces

- Locatable — see "Locatable" definition

Operations

- Block( )

  Description

- Block(xPos : Integer [1] , yPos : Integer [1] , width : Integer [1] , height : Integer [1] , color : javafx.scene.paint.Color [0..1])

  Description

  @paramxPos @paramyPos @paramwidth @paramheight @paramcolor

- Block(xPos : Integer [1] , yPos : Integer [1] , width : Integer [1] , height : Integer [1] , color : javafx.scene.paint.Color [0..1], name : String [0..1])

  Description

  @paramxPos @paramyPos @paramwidth @paramheight @paramcolor @paramname

- Block(xPos : Integer [1] , yPos : Integer [1] , width : Integer [1] , height : Integer [1])

  Description

  @paramxPos @paramyPos @paramwidth @paramheight

- setX(xPos : Integer [1]) : void

  Description

  @paramxPos

- setY(yPos : Integer [1]) : void

  Description

  @paramyPos

- getName( ) : String

  Description

  @return

- setName(name : String [0..1]) : void

  Description

  @paramname

- setPos(x : Integer [1] , y : Integer [1]) : void

Description

@paramx @paramy

- setWidth(width : Integer [1]) : void

  Description

  @paramwidth

- setHeight(height : Integer [1]) : void

  Description

  @paramheight

- setColor(color : javafx.scene.paint.Color [0..1]) : void

  Description

  @paramcolor

- draw(canvas : javafx.scene.canvas.Canvas [0..1]) : void

  Description

  @paramcanvas

- drawRounded(canvas : javafx.scene.canvas.Canvas [0..1], radius : Double [1]) : void

  Description

  @paramcanvas @paramradius

- getX( ) : Integer

  Description

  @return

- getY( ) : Integer

  Description

  @return

- getWidth( ) : Integer

  Description

  @return

- getHeight( ) : Integer

  Description

  @return

- getColor( ) : javafx.scene.paint.Color

  Description

  @return

- toString( ) : String

1.2.4. Class snake::Board

@author Timothy

Attributes

- width : Integer[1] **Read only**
- height : Integer[1] **Read only**
- canvas : javafx.scene.canvas.Canvas[0..1] **Read only**
- outsideMargin : Integer[1]
- margin : Integer[1] **Read only**
- XMARGIN : Integer[1] **Read only**
- YMARGIN : Integer[1] **Read only**
- size : Integer[1] **Read only**
- borderSize : Integer[1] **Read only**
- edgeSize : Integer[1] **Read only**
- gridSize : Integer[1] **Read only**
- mouseClicks : Integer[1]
- blank : String[0..1]
- apple : String[0..1]
- body : String[0..1]
- head : String[0..1]
- bg : String[0..1]
- rock : String[0..1]
- applesEaten : String[0..1]
- unmatchedPortal : String[0..1]
- portalColors : String[][0..1]
- lost : Boolean[1]
- keyPresses : Integer[1]
- playing : Boolean[1]

- soundOn : Boolean[1]
- easyButton : Integer[][0..1] **Read only**
- medButton : Integer[][0..1] **Read only**
- hardButton : Integer[][0..1] **Read only**
- impButton : Integer[][0..1] **Read only**
- musicButton : Integer[][0..1] **Read only**
- SFXButton : Integer[][0..1] **Read only**
- helpButton : Integer[][0..1] **Read only**
- nightTheme : Boolean[1]
- sandboxExists : Boolean[1]
- sandbox : Integer[][][0..1]
- primaryStage : javafx.stage.Stage[0..1]

Operations

- Board(w : Integer [1] , h : Integer [1] , mm : MenuManager [0..1], menu : MainMenu [0..1], gs : GameState [0..1], primary : javafx.stage.Stage [0..1])

  Description

  @paramw @paramh @parammm @parammenu @paramprimary

- setDarkMode( ) : void

  Description

- addAWTToolbox(tb : AWTToolbox [0..1]) : void

  Description

  @paramtb

- getColorScheme( ) : String

  Description

  @return

- setLightMode( ) : void

  Description

- addToolbox(tb : Undefined [0..1]) : void

  Description

  @paramtb

- setOutsideMargin(amt : Integer [1]) : void

Description

@paramamt

- getShowHelp( ) : Boolean

  Description

  @return

- createGrid( ) : void

  Description

- getShowHighScores( ) : Boolean

  Description

  @return

- getShowMenu( ) : Boolean

  Description

  @return

- getGrid( ) : Grid

  Description

  @return

- setGrid(newGrid : Grid [0..1]) : void

  Description

  @paramnewGrid

- getPixelDimensions( ) : Integer
- getNightTheme( ) : Boolean

  Description

  @return

- setNightTheme(val : Boolean [1]) : void

  Description

  @paramval

- drawBlocks( ) : void

  Description

  draws the blox

- reset( ) : void

  Description
- resetKeepGrid( ) : void

  Description
- getSFXOn( ) : Boolean

  Description

    @return

- setSFX(val : Boolean [1]) : void

  Description

    @paramval

- isDirectional(i : javafx.scene.input.KeyEvent [0..1]) : Boolean
- setSandbox(playArea : Integer [*]) : void

  Description

    @paramplayArea

- keyPressed(e : javafx.scene.input.KeyEvent [0..1]) : void

  Description

    @parame

- toggleMusic( ) : void

  Description
- toggleSFX( ) : void

  Description
- findUnusedPortalNum( ) : Integer

  Description

    @return

- AWTToolToRealTool(AWTTool : Integer [1]) : Integer

  Description

    @paramAWTTool @return

- mouseDragged(e : javafx.scene.input.MouseEvent [0..1]) : void

  Description

@parame

- mouseClicked(e : javafx.scene.input.MouseEvent [0..1]) : void

  Description

  @parame

- getCanvas( ) : javafx.scene.canvas.Canvas

  Description

  @return

- toString( ) : String

Associations

- GS : GameState [0..1] — see "GameState" definition
- AWTToolbox : AWTToolbox [0..1] — see "AWTToolbox" definition
- MM : MenuManager [0..1] — see "MenuManager" definition
- grid : Grid [0..1] — see "Grid" definition
- MENU : MainMenu [0..1] — see "MainMenu" definition

1.2.5. Class snake::Enigma

@author Tim Barber

Attributes

- maxAmt : Integer[1]
- additive : Integer[1]

Operations

- encode(num : Integer [1]) : String

  Description

  @paramnum @return

- decode(encoded : String [0..1]) : Integer

  Description

  @paramencoded @return @throwsInvalidObjectException

- encodeOld(num : Integer [1]) : String

  Description

  @paramnum @return

- decodeOld(encoded : String [0..1]) : Integer

  Description

  @paramencoded @return @throwsInvalidObjectException

1.2.6. Class snake::FilePicker

@author TimothyMajority of code in this class taken fromhttp://www.java2s.com/Code/Java/Swing-JFC/DemonstrationofFiledialogboxes.htm

Attributes

- filename : javax.swing.JTextField[0..1]
- dir : javax.swing.JTextField[0..1]
- open : javax.swing.JButton[0..1]
- save : javax.swing.JButton[0..1]

Super classes

- javax.swing.JFrame — [_U_m1UvDCc9EemVJ-LgVHoDQQ]

Operations

- FilePicker( )

  Description

- main(args : String [*]) : void

  Description

  @paramargs

- run(frame : javax.swing.JFrame [0..1], width : Integer [1] , height : Integer [1]) : void

  Description

  @paramframe @paramwidth @paramheight

Nested classifiers

- Class OpenListener — see "OpenListener" definition
- Class SaveListener — see "SaveListener" definition

1.2.7. Class snake::FilePicker::OpenListener

No description.

Implemented interfaces

- java.awt.event.ActionListener — [_U_m1UvCCc9EemVJ-LgVHoDQQ]

Operations

- actionPerformed(e : java.awt.event.ActionEvent [0..1]) : void

1.2.8. Class snake::FilePicker::SaveListener

No description.

Implemented interfaces

- java.awt.event.ActionListener — [_U_m1UvCCc9EemVJ-LgVHoDQQ]

Operations

- actionPerformed(e : java.awt.event.ActionEvent [0..1]) : void

1.2.9. Class snake::GameState

@author Tim Barber

Attributes

- preGame : Boolean[1]
- game : Boolean[1]
- postGame : Boolean[1]

Operations

- GameState(state : Integer [1])

  Description

  @paramstate The value of which state the game is currently in: pre,during, or post

- setToPreGame( ) : void

  Description

  Sets the preGame value to true and the others to false

- setToGame( ) : void

  Description

  Sets the game value to true and the others to false

- setToPostGame( ) : void

  Description

  Sets the postGame value to true and the others to false

- getState( ) : Integer

Description

@return Which state the game is in

- isPreGame( ) : Boolean

  Description

  @return Whether or not it’s preGame

- isGame( ) : Boolean

  Description

  @return Whether or not it’s game time

- isPostGame( ) : Boolean

  Description

  @return Whether or not it’s postGame

1.2.10. Class snake::Grid

No description.

Attributes

- width : Integer[1]
- length : Integer[1]
- playArea : Integer[][][0..1]
- savedPlayArea : Integer[][][0..1]
- startx : Integer[1]
- starty : Integer[1]
- edgeKills : Boolean[1]
- random : java.util.Random[0..1]
- gameOver : Boolean[1]
- diffLevel : Integer[1]
- minDiffLevel : Integer[1]
- maxDiffLevel : Integer[1]
- direction : Integer[1]
- tempDir : Integer[1]
- pos : Pair<Integer,Integer>[*]
- initialSize : Integer[1]

- snakeSize : Integer[1]
- applesEaten : Integer[1]
- soundOn : Boolean[1]
- applePos : Integer[][0..1]
- growBy : Integer[1]
- XADD : Integer[][0..1] **Read only**
- YADD : Integer[][0..1] **Read only**
- frameSpeeds : Integer[][0..1]
- sandboxGrow : Integer[1]
- sandboxLen : Integer[1]
- sandboxEdge : Boolean[1]
- sandboxPos : Pair<Integer,Integer>[0..1]
- sandboxPlayArea : Integer[][][0..1]

Implemented interfaces

- squares — see "squares" definition

Operations

- Grid(width : Integer [1] , length : Integer [1] , startX : Integer [1] , startY : Integer [1])

  Description

  @paramwidth The horizontal number of squares @paramlength The vertical number of squares @paramstartX The x-coordinate of the snake's starting position @paramstartY The y-coordinate of the snake's starting position

- addGameState(gs : GameState [0..1]) : void
- getStartPos( ) : Integer

  Description

  @return The initial position of the snake

- setFrameSpeed(amt : Integer [1]) : void

  Description

  @paramamt The number of frames that should be shown per update cycle

- setFrameSpeed(amt : Integer [1] , level : Integer [1]) : void

  Description

  @paramamt The number of frames that should be shown per update cycle @paramlevel The difficulty level to change

- setPos(x : Integer [1] , y : Integer [1]) : void

  Description

  @paramx The x-coordinate of the snake’s new position @paramy The y-coordinate of the snake’s new position

- overwrite(grid : Grid [0..1]) : void

  Description

  @paramgrid The grid object to copy (most of) the values from

- getInitialLength( ) : Integer

  Description

  @return The initial length of the snake

- getGrowBy( ) : Integer

  Description

  @return The increment value for the snake’s size

- removeAll(type : Integer [1]) : void

  Description

  @paramtype The kind of int to remove and set to zero in the playArea

- findUnmatchedPortal( ) : Pair<Integer,Integer>

  Description

  @return The coordinates of the first portal without a pair reading leftto right top down on the grid

- containsUnmatchedPortal( ) : Integer

  Description

  @return -1 if there are no unmatched portals, otherwise returns thelowest unmatched portal number

- setSandbox(playArea : Integer [*]) : void

  Description

  @paramplayArea The two-dimensional list of ints describing various snakeobjects

- setSandboxEdgeKills(val : Boolean [1]) : void

Description

@paramval Whether the walls kill the snake or not

- setSandboxLen(amt : Integer [1]) : void

  Description

  @paramamt The initial length of the snake in sandbox mode

- setSandboxFrameSpeed(val : Integer [1]) : void

  Description

  @paramval The number frames that pass before another update cycle

- setSandboxHeadPos(x : Integer [1] , y : Integer [1]) : void

  Description

  @paramx The x-coordinate of the head in sandbox mode @paramy The x-coordinate of the head in sandbox mode

- getPlayArea( ) : Integer
- highestNumber( ) : Integer

  Description

  @return

- addPortal(x1 : Integer [1] , y1 : Integer [1] , x2 : Integer [1] , y2 : Integer [1]) : void

  Description

  @paramx1 @paramy1 @paramx2 @paramy2

- isPortal(xPos : Integer [1] , yPos : Integer [1]) : Boolean

  Description

  @paramxPos @paramyPos @return

- getContiguousSize(xPos : Integer [1] , yPos : Integer [1]) : Integer

  Description

  @paramxPos @paramyPos @return

- touchingNeighbors(xPos : Integer [1] , yPos : Integer [1]) : Integer
- formatFilePath(badlyFormattedPath : String [0..1]) : String
- addDeathSounds( ) : void
- setSoundOn(sound : Boolean [1]) : void

Description

@paramsound

- getApplesEaten( ) : Integer

  Description

  @return

- setGrowBy(amt : Integer [1]) : void

  Description

  @paramamt

- setSandboxGrowBy(amt : Integer [1]) : void

  Description

  @paramamt

- reset( ) : void

  Description
- resetSize( ) : void

  Description
- setApplesEaten(amt : Integer [1]) : void

  Description

  @paramamt

- setObstacles( ) : void
- getNeighbors(x : Integer [1] , y : Integer [1] , type : Integer [1] , radius : Integer [1]) : Integer
- getNeighbors(x : Integer [1] , y : Integer [1] , type : Integer [1]) : Integer
- clearObstacles( ) : void
- setDiffLevel(level : Integer [1]) : void

  Description

  @paramlevel

- getDiffLevel( ) : Integer

  Description

  @return

- getFrameSpeed( ) : Integer

Description

@return

- getGensPerFrame( ) : Integer

  Description

  @return

- removeExtra( ) : void
- getEdgeKills( ) : Boolean

  Description

  @return

- setEdgeKills(choice : Boolean [1]) : void

  Description

  @paramchoice

- clearApples( ) : void

  Description
- getApplePos( ) : Integer

  Description

  @return

- newApple( ) : Integer
- setTail(x : Integer [1] , y : Integer [1]) : void

  Description

  @paramx @paramy

- chopTail( ) : void

  Description
- getGameOver( ) : Boolean

  Description

  @return

- getWidth( ) : Integer

  Description

  @return

- getLength( ) : Integer

  Description

  @return

- setWidth(width : Integer [1]) : void
- setLength(length : Integer [1]) : void
- getHeadX( ) : Integer

  Description

  @return

- getHeadY( ) : Integer

  Description

  @return

- getHeadPos( ) : Integer

  Description

  @return

- getDirection( ) : Integer

  Description

  @return

- attemptSetDirection(dir : Integer [1]) : void

  Description

  @paramdir

- getNorth( ) : Integer

  Description

  @return

- getEast( ) : Integer

  Description

  @return

- getSouth( ) : Integer

  Description

  @return

- getWest( ) : Integer

  Description

      @return

- turnRight( ) : void

  Description
- turnLeft( ) : void

  Description
- getSize( ) : Integer

  Description

      @return

- setSize(amt : Integer [1]) : void

  Description

      @paramamt

- grow( ) : void

  Description
- nextPos( ) : Integer

  Description

      @return

- setDirection(dir : Integer [1]) : void

  Description

      @paramdir

- isSnake(xPos : Integer [1] , yPos : Integer [1]) : Boolean

  Description

      @paramxPos @paramyPos @return

- isBlank(xPos : Integer [1] , yPos : Integer [1]) : Boolean

  Description

      @paramxPos @paramyPos @return

- isApple(xPos : Integer [1] , yPos : Integer [1]) : Boolean

Description

@paramxPos @paramyPos @return

- isHead(xPos : Integer [1] , yPos : Integer [1]) : Boolean

  Description

  @paramxPos @paramyPos @return

- isBody(xPos : Integer [1] , yPos : Integer [1]) : Boolean

  Description

  @paramxPos @paramyPos @return

- isOccupied(xPos : Integer [1] , yPos : Integer [1]) : Boolean

  Description

  @paramxPos @paramyPos @return

- isRock(xPos : Integer [1] , yPos : Integer [1]) : Boolean

  Description

  @paramxPos @paramyPos @return

- pick(list : Sound [*]) : Sound

  Description

  @paramlist @return

- find(type : Integer [1]) : Pair<Integer,Integer>

  Description

  @paramtype @return

- otherPortalPos(originalPortalX : Integer [1] , originalPortalY : Integer [1]) : Integer

  Description

  @paramoriginalPortalX @paramoriginalPortalY @return

- setInitialSize(amt : Integer [1]) : void

  Description

  @paramamt

- nextGen( ) : void

  Description
- getSavedPlayArea( ) : Integer

Description

@return

- savePlayArea( ) : void

Description

- clear( ) : void

Description

- setCells(xPosition : Integer [1] , yPosition : Integer [1] , cells : Integer [*]) : void

Description

@paramxPosition @paramyPosition @paramcells

- setCell(x : Integer [1] , y : Integer [1] , value : Integer [1]) : void

Description

@paramx @paramy @paramvalue

- safeSetCell(x : Integer [1] , y : Integer [1] , value : Integer [1]) : void

Description

@paramx @paramy @paramvalue

- setCell(pos : Pair<Integer,Integer> [0..1], value : Integer [1]) : void

Description

@parampos @paramvalue

- getCell(x : Integer [1] , y : Integer [1]) : Integer

Description

@paramx @paramy @return

- safeCheck(xPos : Integer [1] , yPos : Integer [1]) : Integer

Description

@paramxPos @paramyPos @return

- safeCheck(square : Pair<Integer,Integer> [0..1]) : Integer

Description

@paramsquare @return

- countVal(value : Integer [1]) : Integer

Description

@paramvalue @return

- setPlayArea(newPlayArea : Integer [*]) : void

  Description

  @paramnewPlayArea

- toString( ) : String

Associations

- loseSounds : Sound [*] — see "Sound" definition
- warp : Sound [0..1] — see "Sound" definition
- GS : GameState [0..1] — see "GameState" definition
- bite : Sound [0..1] — see "Sound" definition

1.2.11. Interface snake::Locatable

@author Timothy

Operations

- setPos(x : Integer [1] , y : Integer [1]) : void

  Description

  @paramx @paramy

- setX(x : Integer [1]) : void

  Description

  @paramx

- setY(y : Integer [1]) : void

  Description

  @paramy

- getX( ) : Integer

  Description

  @return

- getY( ) : Integer

  Description

  @return

1.2.12. Class snake::MainMenu

@author Tim Barber

Attributes

- music : Boolean[1]
- sfx : Boolean[1]
- current : javafx.scene.image.ImageView[0..1]
- OnOn : javafx.scene.image.ImageView[0..1]
- OnOff : javafx.scene.image.ImageView[0..1]
- OffOn : javafx.scene.image.ImageView[0..1]
- OffOff : javafx.scene.image.ImageView[0..1]

Operations

- MainMenu( )

  Description

  Initializes the ImageView objects for the 4 menu screens

- getMusic( ) : Boolean

  Description

  @return Whether the music icon is on (true) or off (false)

- getSFX( ) : Boolean

  Description

  @return Whether the SFX icon is on (true) or off (false)

- getMenu( ) : javafx.scene.image.ImageView

  Description

  @return The ImageView object currently in use

- turnOffMusic( ) : void

  Description

  Sets the music var to false and chooses the right menu image based onwhether or not the SFX is on or off

- turnOffSFX( ) : void

  Description

  Sets the SFX var to false and chooses the right menu image based onwhether or not the music is on or off

- turnOnMusic( ) : void

  Description

  Sets the music var to true and chooses the right menu image based onwhether or not the SFX is on or off

- turnOnSFX( ) : void

  Description

  Sets the SFX var to true and chooses the right menu image based onwhether or not the music is on or off

1.2.13. Class snake::MenuManager

@author Timothy

Attributes

- menuNames : String[*]
- currentlyDisplaying : Boolean[*]

Operations

- MenuManager(menuNames : String [*])

  Description

  @parammenuNames

- clearDisplaying(size : Integer [1]) : void
- getCurrent( ) : Integer

  Description

  @return

- setCurrent(index : Integer [1]) : void

  Description

  @paramindex

- setMain( ) : void

  Description

- getName(index : Integer [1]) : String

  Description

@paramindex @return

- isOn(index : Integer [1]) : Boolean

  Description

  @paramindex @return

- isOff(index : Integer [1]) : Boolean

  Description

  @paramindex @return

1.2.14. Class snake::Snake

@author Tim Barber

Attributes

- canvasMargin : Integer[1] **Read only**
- canvasW : Integer[1] **Read only**
- canvasH : Integer[1] **Read only**
- WIDTH : Integer[1] **Read only**
- HEIGHT : Integer[1] **Read only**
- TOOLWIDTH : Integer[1] **Read only**
- TOOLHEIGHT : Integer[1] **Read only**
- frame : Integer[1]
- AI : Boolean[1] **Read only**
- scores : Integer[*]
- HS_IV : javafx.scene.image.ImageView[0..1]
- scoresOverwritten : Boolean[1]
- settings : java.io.File[0..1]
- settingsLocation : String[0..1] **Read only**
- sandbox : java.io.File[0..1]
- SANDBOXLOCATION : String[0..1] **Read only**
- sandboxPlayArea : Integer[][][0..1]
- sandboxHeadPos : Pair<Integer,Integer>[0..1]
- sfxOn : Boolean[1]
- musicOn : Boolean[1]

- nightMode : Boolean[1]
- sandboxReset : Boolean[1]
- MENUNAMES : String[**] *Read only**

Super classes

- javafx.application.Application — [_U_m1UvNyc9EemVJ-LgVHoDQQ]

Operations

- start(primaryStage : javafx.stage.Stage [0..1]) : void
- main(args : String [*]) : void

  Description

  @paramargs the command line arguments

- toList(obj : Integer [*]) : Integer

  Description

  @paramobj @return

- getImageView(filename : String [0..1]) : javafx.scene.image.ImageView

  Description

  @paramfilename @return

- compileToSandboxFile(edgeKills : Boolean [1] , frmSpd : Integer [1] , initialLength : Integer [1] , growBy : Integer [1] , playArea : Integer [*]) : String

  Description

  @paramedgeKills @paramfrmSpd @paraminitialLength @paramgrowBy @paramplayArea @return

- loadSandboxFile(content : String [0..1]) : Grid

  Description

  @paramcontent @return

- loadSandboxFile(sandboxFile : java.io.File [0..1]) : Grid

  Description

  @paramsandboxFile @return

- initSandboxFile( ) : void
- createHighScoreScreen( ) : javafx.scene.image.ImageView
- copyFile(srcName : String [0..1], destName : String [0..1]) : Boolean

Description

@paramsrcName @paramdestName @return

- getScores( ) : void
- readDecodedFile(fileName : String [0..1]) : Integer

Description

@paramfileName @return

- distanceToPoint(targetXPos : Integer [1] , targetYPos : Integer [1] , selfX : Integer [1] , selfY : Integer [1]) : Integer

Description

@paramtargetXPos @paramtargetYPos @paramselfX @paramselfY @return

- AI( ) : void

Description

- checkFileExists(filename : String [0..1]) : Boolean

Description

@paramfilename @return

- writeEncodedScore(filename : String [0..1], score : Integer [1]) : void

Description

@paramfilename @paramscore

- overlayImage(filename : String [0..1], newFilename : String [0..1], text : String [0..1], xPos : Integer [1] , yPos : Integer [1] , font : javafx.scene.text.Font [0..1], red : Integer [1] , green : Integer [1] , blue : Integer [1]) : Boolean

Description

@paramfilename @paramnewFilename @paramtext @paramxPos @paramyPos @paramfont @paramred @paramgreen @paramblue @return

- overlayImage(filename : String [0..1], newFilename : String [0..1], addFilename : String [0..1], xPos : Integer [1] , yPos : Integer [1]) : Boolean

Description

@paramfilename @paramnewFilename @paramaddFilename @paramxPos @paramyPos @return

Associations

- AWTToolbox : AWTToolbox [0..1] — see "AWTToolbox" definition

- MM : MenuManager [0..1] — see "MenuManager" definition
- menuMusic : Sound [0..1] — see "Sound" definition
- board : Board [0..1] — see "Board" definition
- GS : GameState [0..1] — see "GameState" definition
- MENU : MainMenu [0..1] — see "MainMenu" definition

1.2.15. Class snake::Sound

@author Tim Barber

Attributes

- filename : String[0..1]
- resource : java.net.URL[0..1]
- mediaPlayer : javafx.scene.media.MediaPlayer[0..1]
- media : javafx.scene.media.Media[0..1]
- clip : javax.sound.sampled.Clip[0..1]
- muteControl : javax.sound.sampled.BooleanControl[0..1]
- volumeLevel : Double[1]
- gainControl : javax.sound.sampled.FloatControl[0..1]

Operations

- Sound(filename : String [0..1])

  Description

  @paramfilename

- setVolume(amt : Double [1]) : void

  Description

  @paramamt

- toggleMute( ) : void

  Description

- mute( ) : void

  Description

- unmute( ) : void

  Description

- loop( ) : void

Description

- play( ) : void

  Description

- stop( ) : void

  Description

- pause( ) : void

  Description

  ==== Abstract Class snake::Window

@author Timothy

Attributes

- TITLE : String[0..1] **Read only**
- WIDTH : Integer[1] **Read only**
- HEIGHT : Integer[1] **Read only**
- SCENE : javafx.scene.Scene[0..1] **Read only**
- stage : javafx.stage.Stage[0..1]

Operations

- Window(title : String [0..1], width : Integer [1] , height : Integer [1] , xPos : Integer [1] , yPos : Integer [1] , scene : javafx.scene.Scene [0..1])

  Description

  @paramtitle @paramwidth @paramheight @paramxPos @paramyPos @paramscene

- show( ) : void

  Description

- hide( ) : void

  Description

- getWidth( ) : Integer

  Description

  @return

- getHeight( ) : Integer

  Description

  @return

- getVisible( ) : Boolean

  Description

    @return

- close( ) : void

  Description
- getStage( ) : javafx.stage.Stage

  Description

    @return

- getScene( ) : javafx.scene.Scene

  Description

    @return

- handleMouseClicked(event : javafx.scene.input.MouseEvent [0..1]) : void

  Description

    @paramevent

- setMousePressedHandler( ) : void

  Description

  ==== Interface snake::squares

@author Tim Barber

Operations
- getWidth( ) : Integer

  Description

    @return the horizontal size of the grid

- getLength( ) : Integer

  Description

    @return the vertical size of the grid

- setWidth(width : Integer [1]) : void

  Description

    @paramwidth the new horizontal size of the grid

- setLength(length : Integer [1]) : void

  Description

  @paramlength the new vertical size of the grid

- getPlayArea( ) : Integer

  Description

  @return the grid as it’s native int[][] type

- safeCheck(xPos : Integer [1] , yPos : Integer [1]) : Integer

  Description

  @return the int stored in the grid at (xPos, yPos). If xPos or yPos isout of bounds, returns -1

- getNeighbors(x : Integer [1] , y : Integer [1] , type : Integer [1] , radius : Integer [1]) : Integer

  Description

  @return the number of occupied spaces matching param type near (x, y) @paramx the x component of the chosen space @paramy the y component of the chosen space @paramtype the spaces which contain an int matching type will be counted @paramradius the number of spaces out from the initial, e.g. a radius ofone will count squares in a 3x3 box excluding the middle square

- getNeighbors(x : Integer [1] , y : Integer [1] , type : Integer [1]) : Integer

  Description

  @see getNeighbors(int x, int y, int type, int radius); @return same as getNeighbors(int x, int y, int type, int radius), butwith implied radius 1