

```

< // Swift source code of Classes.swift Version 9.1.9 at greenchess.com //
---
< // Swift source code of Classes.swift Version 9.1.8 at greenchess.com //
23,25c23,29
< // the core graphics functions //
< var Graphics:CoreGraphics = CoreGraphics()
<
---
> // the marker cursor names //
> var Markers:[Int:String] = [304:"marker_red_icon", 305:"marker_green_icon",
> 306:"marker_blue_icon", 307:"marker_cyan_icon",
> 308:"marker_magenta_icon", 309:
> "marker_yellow_icon", 310:"marker_black_icon",
> 311:"marker_white_icon"]
64,67c68
<
< // the board dimensions //
< var Dims:BoardDimensions = BoardDimensions()
<
---
99,104c100,102
< Cursor = NSCursor.init(image:Graphics
< .CGImageToNSImage(ImageRenderer(content:
< IconImage("paintbrush.pointed.fill",
< Pens[Color!]).cgImage!, Size:Dims.Square.Width
< * 0.5), hotSpot:CGPoint(x:0, y:Dims.Square.Width
< * 0.4))
---
Cursor = NSCursor.init(image:ImageRenderer(content:
MarkerPen(Pens[Color!]).nsImage!, hotSpot:
CGPoint(x:40, y:60))
07a106,114
> // update the cursor //
> @MainActor func UpdateCursor(_ Color:String)
> {
> // update the cursor icon //
> Cursor = NSCursor.init(image:NSImage(named:Markers[304...311])
> .contains(Int(Color)) ? Int(Color)! : 304!)!,
> hotSpot:CGPoint(x:2, y:30))
> }
116,118d122
<
< // refresh the cursor //
< RefreshCursor()
215,216c219,221
< .GetData(Index), InPositionEditor:false, RankAndFile:APoint(),
UpdateBoard:false, Info:Info)
---
.GetData(Index), InPositionEditor:false, FENEnpassantSquare:"",
EnpassantSquare:APoint(), RankAndFile:APoint(), UpdateBoard:false,
Info:Info)
1556c1561,1564
<
---
> // check if have no piece currently moving //
> func NoMovingPiece() -> Bool {return MovingPiece == 0}
1562c1570,1573
<
---
> // check if currently placing an enpassant square in the position editor //
> func PlacingEnpassantSquare() -> Bool {return MovingPiece == 10}
1727a1739,1741
>
> // the default color of red //
> var Color:Int = 304
1731,1734c1745,1752
< // the core graphics functions //
< var Graphics:CoreGraphics = CoreGraphics()
<
---
> // the draw colors for the markers //
> var Colors:[Int:CGColor] = [304:NSColor.red.cgColor, 305:NSColor.green
> .CGColor, 306:NSColor.blue.cgColor, 307:NSColor
> .cyan.cgColor, 308:NSColor.magenta.cgColor, 309:
> NSColor.yellow.cgColor, 310:NSColor.black.cgColor,
> 311:NSColor.white.cgColor]
1771a1790,1792
> // the core graphics functions //
> var Graphics:CoreGraphics = CoreGraphics()
>
1786c1807
< CurrentColor = Pens[Color]!
---
(self.Color, CurrentColor) = (Color, Pens[Color]!)
1788a1810,1822
> Tracker.UpdateCursor(Color)
> }
>
> // update the color and diagram cursor to the selected value //
> @MainActor func UpdateColor(_ Color:String)
> {
> // set color with default of red if selected color doesn't exist //
> self.Color = (304...311).contains(Int(Color)) ? Int(Color)! : 304
>
> // set the current path color //
> CurrentColor = Colors[self.Color]!
>
> // update the cursor to the current icon for specified color //
1797c1831
<
---
1798a1833,1835
> UpdateColor("304")
>
> // reset the path color to red //
1817a1855,1866
> // update the x offset value //
> self.XOffset = XOffset
>
> // draw diagram layer into the full context at the specified x offset //
> Context.draw(Diagram, at:CGPoint(x:XOffset, y:0))
>
> // reset the sketch images to empty //
> SketchImages.ResetImages()
>
> // update the sketch image //
> SketchImages.Update(Context.makeImage(!))
1821,1832d1869
<
< // update the x offset value //
< self.XOffset = XOffset
<
< // draw diagram layer into the full context at specified x offset //
< Context.draw(Diagram, at:CGPoint(x:XOffset, y:0))
<
< // reset the sketch images to empty //
< SketchImages.ResetImages()
<
< // update the sketch image //
< SketchImages.Update(Context.makeImage(!))
1905,1909c1942,1943
< // run on the main thread //
< Task {await MainActor.run
< {
< // reset displaying, tracking, and drawing to false //
---
(Displaying, Tracker.Tracking, Drawing) = (false, false, false)
> // reset displaying, tracking, and drawing to false //
> (Displaying, Tracker.Tracking, Drawing) = (false, false, false)
1911,1915c1945,1949
< // if there are at least two points in the path //
< if Tracker.Points.count > 1
< {
< // add the mouse tracking points to the path points //
---
Path.Points.append(contentsOf:Tracker.Points)
> // if there are at least two points in the path //
> if Tracker.Points.count > 1
> {
> // add the mouse tracking points to the path points //
> Path.Points.append(contentsOf:Tracker.Points)
1917,1920c1951,1953
< // append the path to the paths //
< Paths.append(self.Path)
<
---
})
> // append the path to the paths //
> Paths.append(self.Path)
2152,2153c2185,2186
Bool, RankAndFile:APoint, UpdateBoard:Bool, Info:Bool) ->
CGImage
---
Bool, FENEnpassantSquare:String, EnpassantSquare:APoint,
RankAndFile:APoint, UpdateBoard:Bool, Info:Bool) -> CGImage
2228c2261,2269
<
---
> // if in pos editor and fen enpassant square exists and main size board //
> if InPositionEditor && (FENEnpassantSquare != "-") && (Size == 1320)
> {
> // draw the en passant square in the rectangle //
> Context.draw(Highlights.EnpassantSquareImage, in:Dims.GetRect(Inf
> .Vals("RotateBoard").VB, Point:EnpassantSquare), byTiling:
> false)
> }
2230c2271
---
if !InPositionEditor
2549a2591,2593
>
> // update the position editor toolbar buttons //
> GCUI.Board.UpdatePositionEditorToolbarButtons()
2705,2709c2749,2753
"Helvetica", "Helvetica Neue", "Lucida Grande", "Menlo",
"Noteworthy", "Optima", "Palatino", "PT Mono",
"PT Serif", "SF Compact", "SF Pro", "Skia",
"Snell Roundhand", "Times New Roman", "Trebuchet MS",
"Verdana"]
---
"Helvetica", "Helvetica Neue", "Iowan Old Style",
"Lucida Grande", "Menlo", "Noteworthy", "Optima",
"Palatino", "PT Mono", "PT Serif", "Skia",
"Snell Roundhand", "Times", "Times New Roman",
"Trebuchet MS", "Verdana"]
2980a3025,3063
> // the position view layers class //
> class PositionViewLayers
> {
> // the position layers //
> var Layers:[CALayer] = [CALayer(), CALayer()]
>
> // the saved position animation //
> var PositionAnimation:CABasicAnimation = CABasicAnimation(keyPath:"opacity")
>
> // add the position layers to the view //
> @MainActor func AddLayers(_ View:NSView)
> {
> // loop over all the layers to add to the view //
> for TheLayer in Layers {View.layer?.addSublayer(TheLayer)}
> }
>
> // update the position layer //
> @MainActor func UpdateLayer(_ Image:CGImage)
> {
> // remove the animation effects //
> CATransaction.setDisableActions(true)
>
> // copy saved pos into pos layer one and set layer two opacity to 0 //
> (Layers[0].contents, Layers[1].opacity) = (Image, 0)
> }
>
> // remove all the layer animations //
> @MainActor func RemoveAnimations()
> {
> // loop over all the layers to remove //
> for TheLayer in Layers {TheLayer.removeAllAnimations()}
> }
>
> // animate the position layers //
> @MainActor func Animate(_ View:NSView, Image:CGImage)
> {
> // remove the animations //
> RemoveAnimations()
2981a3065,3112
> // remove position layer two from the super layer //
> Layers[1].removeFromSuperlayer()
>
> // create position layer two //
> Layers[1] = CALayer()
>
> // initialize position layer two //
> Initialize(1)
>
> // set the position layer two contents to position layer one contents //
> Layers[1].contents = Layers[0].contents
>
> // add the position layer two to the load position view //
> View.layer?.addSublayer(Layers[1])
>
> // remove the animation effects //
> CATransaction.setDisableActions(true)
>
> // set the position layer values //
> (Layers[0].contents, PositionAnimation.fromValue, PositionAnimation
> .toValue, Layers[1].opacity) = (Image, 1, 0, 0)
>
> // add an animation to position layer two to animate opacity //
> Layers[1].add(PositionAnimation, forKey:"opacity")
> }
>
> // initialize a position layer //
> @MainActor func Initialize(_ Index:Int)
> {
> // set the bounds of the position layer and the position //
> (Layers[Index].bounds, Layers[Index].position) = (CGRect(x:0, y:0, width:
450, height:450), CGPoint(x:225, y:225))
> }
>
> // the position layers class initializer //
> @MainActor init()
> {
> // initialize position layer one //
> Initialize(0)
>
> // initialize position layer two //
> Initialize(1)
>
> // set the animation duration //
> PositionAnimation.duration = 0.5
> }
3096c3227,3233
<
---
> // the enpassant square cgImage //
> var EnpassantSquareImage:CGImage = CoreGraphics().InitImage()
>
> // the enpassant square nsImage //
> var EnpassantSquareNSImage = NSImage()
3105,3106c3242,3245
SelectedSquareImage) = (Graphics.CreateShape(Dims.Square.WidthRet,
Fraction:0.9, Color:NSColor(red:1, green:1, blue:0.65, alpha:1), Square:
---
SelectedSquareImage, EnpassantSquareImage) = (Graphics.CreateShape(Dims
.Square.WidthRet, Fraction:0.9, Color:NSColor(red:1, green:1, blue:0.65,
alpha:1), Square:false), Graphics.CreateShape(Dims.Square.WidthRet,
Fraction:0.95, Color:NSColor(red:1, green:0.3, blue:0.3, alpha:1), Square:
3108,3112c324,3255
NSColor(red:1, green:0.3, blue:0.3, alpha:1), Square:false), Graphics
.CreateShape(Dims.Square.WidthRet, Fraction:0.95, Color:NSColor(red:1,
green:0.6, blue:0, alpha:1), Square:false), Graphics.CreateShape(Dims
.Square.WidthRet, Fraction:0.97, Color:NSColor(red:0.45, green:0.7, blue:1,
alpha:1), Square:true))
---
NSColor(red:1, green:0.6, blue:0, alpha:1), Square:false), Graphics
.CreateShape(Dims.Square.WidthRet, Fraction:0.97, Color:NSColor(red:0.2,
green:0.7, blue:1, alpha:1), Square:true), Graphics.CreateShape(Dims
.Square.WidthRet, Fraction:0.95, Color:NSColor(red:0.45, green:0.45,
blue:1, alpha:1), Square:false))
>
> // set the enpassant square nsImage //
> EnpassantSquareNSImage = Graphics.CGImageToNSImage(EnpassantSquareImage,
Size:Dims.Square.Width)
3307,3313c3450,3453
var Boards:[String] = ["Bamboo Squares", "Blue Squares", "Brown Squares",
"Chalk Squares", "Cloth Squares", "Felt Squares",
"Glass Squares", "Gold Squares", "Leather Squares",
"Gray Squares", "Green Squares", "Leather Squares",
"Marble Squares", "Plastic Squares", "Rust Squares",
"Sandstone Squares", "Silver Squares", "Teal Squares",
"Wood Squares"]
---
var Boards:[String] = ["Bamboo", "Blue", "Brown", "Chalk", "Cloth", "Felt",
"Glass", "Gold", "Granite", "Gray", "Green", "Leather",
"Marble", "Plastic", "Rust", "Sandstone", "Silver",
"Teal", "Wood"]
3368,3370d3507
<
< // the position editor icons //
< var PosEditorIcons:[Image] = []
3403,3410d3539
<
<
< // loop over all the cgImages //
< for Index in 0..

```