```
2c2
< // Swift source code of main.swift Version 8.2.5 at greenchess.com //
---
> // Swift source code of main.swift Version 8.2.4 at greenchess.com //
5c5
< // import needed system files //
---
> // incorporate needed system files //
135c135
<    var CGBoardSize:[CGFloat] = [220, 400, 660]
---
>    var CGBoardSize:[CGFloat] = [250, 400, 660]
2258,2261d2257
<          // create a table cell //
<          TableCell = View.makeView(withIdentifier:NSUserInterfaceItemIdentifier(rawValue:"PGN"),
<                                    owner:self) as! NSTableCellView
<
2265c2261
<             // if not hiding the illustrations and not displaying the stockifsh analysis string //
---
>             // if not hiding the illustrations or not displaying the stockifsh analysis string //
2271,2280d2266
<
<                // set the table cell image view image //
<                TableCell.imageView!.image = PGNMovesData.GetViewImage(Row / 2, ImageSize:
<                                                                       CGFloat(IllustrationSize))
<             }
<             // no data to extract an image from //
<             else
<             {
<                // set the table cell image to empty //
<                TableCell.imageView!.image = NSImage()
2282,2284c2268,2270
<
<             // set the table text cell to be empty //
<             TableCell.textField!.attributedStringValue = NSAttributedString()
---
>
>             // return the view cell for the illustration from the pgn moves data //
>             return PGNMovesData.GetViewCell(Row / 2, ImageSize:CGFloat(IllustrationSize))
2289,2293c2275,2276
<             // set the table cell text field //
<             TableCell.textField!.attributedStringValue = PGNMovesData.GetViewText(Row / 2)
<
<             // set the table cell image view image to be empty //
<             TableCell.imageView!.image = NSImage()
---
>             // return the view cell for the move text from the pgn moves data //
>             return PGNMovesData.GetTextCell(Row / 2)
2295,2297d2277
<
<          // return the table cell //
<          return TableCell
3236,3241d3215
<       // if the escape key was pressed in fullscreen mode //
<       else if Event.keyCode == 53 && FullScreen
<       {
<          // toggle the main window out of fullscreen mode //
<          MainWindow.toggleFullScreen(nil)
<       }
3748c3722
<          Alert.icon = Graphics.ContextToNSImage(CGContexts[0], Size:CGBoardSize[0])
---
>          Alert.icon = Graphics.ContextToNSImage(CGContexts[2], Size:CGBoardSize[2])
4951c4925
<       SavedPositionSlot = PreferenceInt("SavedPositionSlot", Default:1, Min:1, Max:5)
---
>       SavedPositionSlot = PreferenceInt("SavedPositionSlot", Default:1, Min:1, Max:20)
4954c4928
<       for PositionListIndex in 1 ... 5
---
>       for PositionListIndex in 1 ... 20
5926c5900
<       PGNMovesData.Initialize(CGBoardSize[0], TheScale:Scale)
---
>       PGNMovesData.Initialize(PGNMovesTableView, Size:CGBoardSize[0], Scale:Scale)
5964c5938
<       AppNameView.stringValue = "GreenChess 8.2.5"
---
>       AppNameView.stringValue = "GreenChess 8.2.4"
6076c6050
<       PositionLayers[0].bounds = CGRect(x:0, y:0, width:350, height:350)
---
>       PositionLayers[0].bounds = CGRect(x:0, y:0, width:600, height:600)
6079c6053
<       PositionLayers[0].borderWidth = 0.5
---
>       PositionLayers[0].borderWidth = 1
6085c6059
<       PositionAnimation.duration = 0.5
---
>       PositionAnimation.duration = 1
8112,8114d8085
<       // notify when the chess engine has data to send to standard output //
<       EngineFileHandleRead.readInBackgroundAndNotify()
<
10874c10845
<       PositionLayers[0].position = CGPoint(x:175, y:175)
---
>       PositionLayers[0].position = CGPoint(x:300, y:300)
10901,10902c10872,10873
<       // set the position layer two border width //
<       PositionLayers[1].borderWidth = 0.5
---
>       // set the position layer two border width to 1 //
>       PositionLayers[1].borderWidth = 1
10908c10879
<       PositionLayers[1].bounds = CGRect(x:0, y:0, width:350, height:350)
---
>       PositionLayers[1].bounds = CGRect(x:0, y:0, width:600, height:600)
10914c10885
<       PositionLayers[1].position = CGPoint(x:175, y:175)
---
>       PositionLayers[1].position = CGPoint(x:300, y:300)
10959,10961d10929
<       // update the small sized board //
<       RefreshBoardAndPieces(0, UpdateBoard:false, Display:false)
<
10963c10931
<       Attributes.PositionImage = Graphics.ContextToNSImage(CGContexts[0], Size:CGBoardSize[0])
---
>       Attributes.PositionImage = Graphics.ContextToNSImage(CGContexts[2], Size:CGBoardSize[2])
10975c10943
<       SavedPositionImages.append(Graphics.ContextToNSImage(CGContexts[0], Size:CGBoardSize[0]))
---
>       SavedPositionImages.append(Graphics.ContextToNSImage(CGContexts[2], Size:CGBoardSize[2]))
10986,10987c10954,10955
<          // if the saved position slot is greater than 5 //
<          if SavedPositionSlot > 5
---
>          // if the saved position slot is greater than 20 //
>          if SavedPositionSlot > 20
12905a12874,12876
>
>       // notify when chess engine has data to send to standard output //
>       EngineFileHandleRead.readInBackgroundAndNotify()
14422,14424c14393,14395
<       PGNMovesData.Update(Index, Layout:Layout, CaptureSquares:ListOfCaptureSquares,
<                           ThreatenSquares:ListOfThreatenSquares, StartSquare:StartSquare,
<                           EndSquare:EndSquare)
---
>       PGNMovesData.Data[Index].Update(Layout, TheCaptureSquares:ListOfCaptureSquares,
>                                       TheThreatenSquares:ListOfThreatenSquares, TheStartSquare:
>                                       StartSquare, TheEndSquare:EndSquare)
```