```
2c2
< // Swift source code of main.swift Version 9.1.1 at greenchess.com //
---
> // Swift source code of main.swift Version 9.1.0 at greenchess.com //
18,23d17
<    // the move rows list //
<    var MoveRowsList:[AListRow] = []
<
<    // the info rows list //
<    var InfoRowsList:[AListRow] = []
<
342a337,339
>    // the game info data //
>    var GameInfoData:GameInfoDataSource = GameInfoDataSource()
>
548a546,551
>
>    // the table view of the pgn moves list table view //
>    @IBOutlet var PGNMovesTableView:NSTableView!
>
>    // the game info table view //
>    @IBOutlet var GameInfoTableView:NSTableView!
564a568,573
>    // the white time clock display //
>    @IBOutlet var WhiteTimeClock:TimeClockDisplay!
>
>    // the black time clock display //
>    @IBOutlet var BlackTimeClock:TimeClockDisplay!
>
603,611c612,613
<
<    // update the app after finishes launching //
<    func applicationDidFinishLaunching(_ Notif:Notification)
<    {
<       // refresh the table views //
<       GCUI.RefreshViews(["GameInfoTableView", "MoveListTableView"])
<    }
<
<    // initialize the app before finishes launching //
---
>
>    // initialize the app during launch //
628c630
<       StartNewGame("All")
---
>       StartNewGame()
632c634
<
---
>
701,702c703,704
<       GCUI.GetClockView("WhiteClockView").UpdateClockWindowColor(GCUI
<            .GetColorView("WindowColorView").GetCurrentHue())
---
>       WhiteTimeClock.UpdateClockWindowColor(GCUI.GetColorView("WindowColorView")
>                                              .GetCurrentHue())
705,706c707,708
<       GCUI.GetClockView("BlackClockView").UpdateClockWindowColor(GCUI
<            .GetColorView("WindowColorView").GetCurrentHue())
---
>       BlackTimeClock.UpdateClockWindowColor(GCUI.GetColorView("WindowColorView")
>                                              .GetCurrentHue())
801a804,847
>
>    // return the view for the specified row in the table view //
>    func tableView(_ View:NSTableView, viewFor Column:NSTableColumn?, row Row:
>                  Int) -> NSView?
>    {
>       // switch over the table view to return the table cell //
>       switch View
>       {
>          // if the current view is the pgn moves table view //
>          case PGNMovesTableView :
>
>             // create a table cell //
>             let Cell = View.makeView(withIdentifier:
>                        NSUserInterfaceItemIdentifier(rawValue:"PGN"), owner:
>                        self) as! NSTableCellView
>
>             // update the table cell image and text field //
>             (Cell.imageView!.image, Cell.textField!.attributedStringValue) =
>             (Inf.Vals("HideIllustrations").VB ? NSImage() : FrameNSImage(Row +
>             1, Size:CGFloat(Inf.Vals("IllustrationSize").VI), UpdateBoard:false,
>             Table:true), GCUI.PGNMovesData.GetViewText(Row))
>
>             // return the table cell //
>             return Cell
>
>          // if is the game info table view //
>          case GameInfoTableView :
>
>             // create a table cell //
>             let Cell = View.makeView(withIdentifier:
>                        NSUserInterfaceItemIdentifier(rawValue:"Info"), owner:
>                        self) as! NSTableCellView
>
>             // set the table cell attributed string value //
>             Cell.textField!.attributedStringValue = GameInfoData
>                                                     .GameInfoList[Row]
>
>             // return the table cell //
>             return Cell
>
>          // the default switch case //
>          default : return nil
>       }
>    }
802a849,857
>    // get the dynamic table view row height //
>    func tableView(_ TableView:NSTableView, heightOfRow Row:Int) -> CGFloat
>    {
>       // return the table view dynamic height //
>       return (TableView == PGNMovesTableView) ? min(GCUI.PGNMovesData
>              .RowHeight(Row), 2000) : ((TableView == GameInfoTableView) ?
>              GameInfoData.RowHeight(Row) : 30)
>    }
>
1026c1081
<       // update if the game is over //
---
>       // if the game is over //
1046c1101
<       // resize the window to small //
---
>       // resize the window to small /
1064,1065c1119
<       (GCUI.GetClockView("WhiteClockView").isHidden, GCUI
<       .GetClockView("BlackClockView").isHidden) = (Inf
---
>       (WhiteTimeClock.isHidden, BlackTimeClock.isHidden) = (Inf
1140,1146c1194,1199
<          (GCUI.GetClockView("WhiteClockView").frame.origin, GCUI
<          .GetClockView("BlackClockView").frame.origin, GCUI
<          .Objs["GameInfoTableView"]!.Place, GCUI.Objs["MoveListTableView"]!
<          .Place) = (CGPoint(x:695, y:280), CGPoint(x:845, y:280), Placement(XP:
<          820, YP:-520 + HideClock * 27.5, Width:300, Height:260 + HideClock *
<          55), Placement(XP:820, YP:-140 - HideClock * 27.5, Width:300, Height:
<          260 + HideClock * 55))
---
>          (WhiteTimeClock.frame, BlackTimeClock.frame, PGNMovesTableView
>          .enclosingScrollView!.frame, GameInfoTableView.enclosingScrollView!
>          .frame) = (CGRect(x:705, y:280, width:100, height:100), CGRect(x:830,
>          y:280, width:100, height:100), CGRect(x:675, y:15, width:290, height:
>          250 + HideClock * 60), CGRect(x:675, y:395 - HideClock * 60, width:290,
>          height:250 + HideClock * 60))
1152,1159c1205,1211
<          (GCUI.GetClockView("WhiteClockView").frame.origin, GCUI
<          .GetClockView("BlackClockView").frame.origin, GCUI
<          .Objs["GameInfoTableView"]!.Place, GCUI.Objs["MoveListTableView"]!
<          .Place) = (CGPoint(x:1000, y:550 + InFullScreen * 70), CGPoint(x:1145,
<          y:550 + InFullScreen * 70), Placement(XP:1122.5, YP:-275 - HideClock *
<          55 - InFullScreen * 35, Width:295, Height:530 + HideClock * 110 +
<          InFullScreen * 70), Placement(XP:817.5, YP:-330 - InFullScreen * 35,
<          Width:295, Height:640 + InFullScreen * 70))
---
>          (WhiteTimeClock.frame, BlackTimeClock.frame, PGNMovesTableView
>          .enclosingScrollView!.frame, GameInfoTableView.enclosingScrollView!
>          .frame) = (CGRect(x:1005, y:545 + InFullScreen * 70, width:100, height:
>          100), CGRect(x:1130, y:545 + InFullScreen * 70, width:100, height:100),
>          CGRect(x:675, y:15, width:290, height:630 + InFullScreen * 70),
>          CGRect(x:975, y:15, width:290, height:515 + HideClock * 115 +
>          InFullScreen * 70))
1161,1163c1213,1218
<
<       // refresh the table views //
<       GCUI.RefreshViews(["GameInfoTableView", "MoveListTableView"])
---
>
>       // scroll to the last row in the movelist //
>       PGNMovesTableView.scrollRowToVisible(GCUI.PGNMovesData.LastRow())
>
>       // scroll to the first row in the game info list //
>       GameInfoTableView.scrollRowToVisible(0)
1330c1385
<       // set that move exists and was able to be found from possible moves //
---
>       // set that the move exists and was able to be found from possible moves //
1610,1611c1665,1670
<       (GCUI.GetClockView("WhiteClockView").RefreshAll, GCUI
<       .GetClockView("BlackClockView").RefreshAll) = (true, true)
---
>       for Clock in [WhiteTimeClock, BlackTimeClock,
>                     GCUI.GetClockView("ClockView")]
>       {
>          // set that need to refresh the entire clock //
>          Clock!.RefreshAll = true
>       }
1635,1636c1694
<          for TheClock in [GCUI.GetClockView("WhiteClockView"), GCUI
<              .GetClockView("BlackClockView")]
---
>          for TheClock in [WhiteTimeClock, BlackTimeClock]
1639c1697
<             TheClock.alphaValue -= (TheClock.alphaValue > 0.2) ? 0.02 : 0
---
>             TheClock!.alphaValue -= (TheClock!.alphaValue > 0.2) ? 0.02 : 0
1647,1648c1705
<       for TheClock in [GCUI.GetClockView("WhiteClockView"), GCUI
<           .GetClockView("BlackClockView")]
---
>       for TheClock in [WhiteTimeClock, BlackTimeClock]
1651c1708
<          TheClock.alphaValue += (TheClock.alphaValue < 1) ? 0.02 : 0
---
>          TheClock!.alphaValue += (TheClock!.alphaValue < 1) ? 0.02 : 0
1715,1716c1772,1773
<       if (ElapsedWhiteTime != GCUI.GetClockView("WhiteClockView").ElapsedTime)
<          || GCUI.GetClockView("WhiteClockView").RefreshAll
---
>       if (ElapsedWhiteTime != WhiteTimeClock.ElapsedTime) || WhiteTimeClock
>          .RefreshAll
1719c1776
<          GCUI.GetClockView("WhiteClockView").UpdateClock(ElapsedWhiteTime)
---
>          WhiteTimeClock.UpdateClock(ElapsedWhiteTime)
1723,1724c1780,1781
<       if (ElapsedBlackTime != GCUI.GetClockView("BlackClockView").ElapsedTime)
<          || GCUI.GetClockView("BlackClockView").RefreshAll
---
>       if (ElapsedBlackTime != BlackTimeClock.ElapsedTime) || BlackTimeClock
>          .RefreshAll
1727c1784
<          GCUI.GetClockView("BlackClockView").UpdateClock(ElapsedBlackTime)
---
>          BlackTimeClock.UpdateClock(ElapsedBlackTime)
1732c1789
<    func StartNewGame(_ Tasks:String)
---
>    func StartNewGame()
1764c1821
<       Refresh(Tasks)
---
>       Refresh("All")
2021,2024c2078,2080
<       (GCUI.GetClockView("WhiteClockView").isHidden, GCUI
<       .GetClockView("BlackClockView").isHidden, GCUI.GameSounds.SoundOff) =
<       (Inf.Vals("HideGameClockInPanel").VB, Inf.Vals("HideGameClockInPanel").VB,
<       Inf.Vals("SoundOff").VB)
---
>       (WhiteTimeClock.isHidden, BlackTimeClock.isHidden, GCUI.GameSounds
>       .SoundOff) = (Inf.Vals("HideGameClockInPanel").VB, Inf
>       .Vals("HideGameClockInPanel").VB, Inf.Vals("SoundOff").VB)
2027c2083,2085
<       MoveListFont.Read(27, TheScale:1, Tag:"MoveList")
---
>       MoveListFont.Read(27, TheScale:1, VertOffset:Inf.Vals("IllustrationSize")
>                        .VI, UseVertOffset:!Inf.Vals("HideIllustrations").VB,
>                        Tag:"MoveList")
2030c2088,2089
<       GameInfoFont.Read(27, TheScale:1, Tag:"GameInfo")
---
>       GameInfoFont.Read(27, TheScale:1, VertOffset:0, UseVertOffset:false, Tag:
>                        "GameInfo")
2033c2092,2093
<       GameListFont.Read(27, TheScale:1, Tag:"GameList")
---
>       GameListFont.Read(27, TheScale:1, VertOffset:0, UseVertOffset:false, Tag:
>                        "GameList")
2284,2285c2344,2345
<       // add the colors and clocks views and tables to the screen //
<       GCUI.AddNSViewsTables(self)
---
>       // add the colors and clocks views to the screen //
>       GCUI.AddNSViews(self)
2336c2396
<       // set button, titles, responders, concurrent drawing //
---
>       // set button, titles, responders, concurrent drawing, and data sources //
2338c2398,2400
<       .allowsConcurrentViewDrawing, MainWindow.initialFirstResponder,
---
>       .allowsConcurrentViewDrawing, PGNMovesTableView.dataSource,
>       GameInfoTableView.dataSource, PGNMovesTableView.backgroundColor,
>       GameInfoTableView.backgroundColor, MainWindow.initialFirstResponder,
2342,2345c2404,2409
<       AppNameView.stringValue) = (true, true, self, true, "Engine", "Hide",
<       "Timed Game", GCUI.BPComp.Highlights.EnpassantSquareNSImage, CGPoint(x:
<       Dims.Square.Width * 0.5, y:Dims.Square.Width * 0.5), NSFont
<       .systemFont(ofSize:20), .justified, "GreenChess 9.1.1")
---
>       AppNameView.stringValue) = (true, true, GCUI.PGNMovesData, GameInfoData,
>       NSColor.white.withAlphaComponent(0.35), NSColor.white
>       .withAlphaComponent(0.35), self, true, "Engine", "Hide", "Timed Game",
>       GCUI.BPComp.Highlights.EnpassantSquareNSImage, CGPoint(x:Dims.Square.Width
>       * 0.5, y:Dims.Square.Width * 0.5), NSFont.systemFont(ofSize:20),
>       .justified, "GreenChess 9.1.0")
2370c2434
<       GCUI.GetClockView("WhiteClockView").Initialize()
---
>       WhiteTimeClock.Initialize()
2373c2437
<       GCUI.GetClockView("BlackClockView").Initialize()
---
>       BlackTimeClock.Initialize()
2782c2846
<          DispatchQueue.main.async {self.UpdateCursor()}
---
>          DispatchQueue.main.async{self.UpdateCursor()}
3245c3309
<       StartNewGame("")
---
>       StartNewGame()
3369,3371c3433,3435
<       (MainToolbar.selectedItemIdentifier, GCUI.Objs["GameInfoTableView"]!
<       .ObjView.alphaValue, GCUI.Objs["MoveListTableView"]!.ObjView.alphaValue) =
<       (NSToolbarItem.Identifier(rawValue:"204"), 0.25, 0.25)
---
>       (MainToolbar.selectedItemIdentifier, PGNMovesTableView
>       .enclosingScrollView!.alphaValue, GameInfoTableView.enclosingScrollView!
>       .alphaValue) = (NSToolbarItem.Identifier(rawValue:"204"), 0.25, 0.25)
3451c3515
<          StartNewGame("")
---
>          StartNewGame()
3477,3479c3541,3543
<       // set table view alpha values to one //
<       (GCUI.Objs["GameInfoTableView"]!.ObjView.alphaValue, GCUI
<       .Objs["MoveListTableView"]!.ObjView.alphaValue) = (1, 1)
---
>       // unhide the table views //
>       (PGNMovesTableView.enclosingScrollView!.alphaValue, GameInfoTableView
>       .enclosingScrollView!.alphaValue) = (1, 1)
3608c3672
<       // set that game explorer window currently closed and analyzing game //
---
>       // set that game explorer window is currently closed and analyzing game //
3745,3746c3809
<       (GCUI.GetClockView("WhiteClockView").RefreshAll, GCUI
<       .GetClockView("BlackClockView").RefreshAll) = (true, true)
---
>       (WhiteTimeClock.RefreshAll, BlackTimeClock.RefreshAll) = (true, true)
3749c3812
<       GCUI.GetClockView("WhiteClockView").UpdateClock(ElapsedWhiteTime)
---
>       WhiteTimeClock.UpdateClock(ElapsedWhiteTime)
3752c3815
<       GCUI.GetClockView("BlackClockView").UpdateClock(ElapsedBlackTime)
---
>       BlackTimeClock.UpdateClock(ElapsedBlackTime)
3847c3910,3911
<       GameListFont.UpdateFont(Key, Selection:Selection)
---
>       GameListFont.UpdateFont(0, UseVertOffset:false, Key:Key, Selection:
>                               Selection)
3854c3918,3920
<       MoveListFont.UpdateFont(Key, Selection:Selection)
---
>       MoveListFont.UpdateFont(Inf.Vals("IllustrationSize").VI, UseVertOffset:
>                               !Inf.Vals("HideIllustrations").VB, Key:Key,
>                               Selection:Selection)
3864c3930,3931
<       GameInfoFont.UpdateFont(Key, Selection:Selection)
---
>       GameInfoFont.UpdateFont(0, UseVertOffset:false, Key:Key, Selection:
>                               Selection)
3874c3941,3942
<       GCUI.BPComp.BoardLabelsFont.UpdateFont(Key, Selection:Selection)
---
>       GCUI.BPComp.BoardLabelsFont.UpdateFont(0, UseVertOffset:false, Key:Key,
>                                              Selection:Selection)
3887,3894c3955,3965
<       // loop over all the clocks //
<       for TheClock in [GCUI.GetClockView("WhiteClockView"), GCUI
<           .GetClockView("BlackClockView"), GCUI.GetClockView("ClockView")]
<       {
<          // update the clock background color //
<          TheClock.UpdateClockBackgroundColor(GCUI
<                  .GetColorView("ClockSurfaceColorView").GetCurrentHue())
<       }
---
>       // update the white time clock background color //
>       WhiteTimeClock.UpdateClockBackgroundColor(GCUI
>                     .GetColorView("ClockSurfaceColorView").GetCurrentHue())
>
>       // update the black time clock background color //
>       BlackTimeClock.UpdateClockBackgroundColor(GCUI
>                     .GetColorView("ClockSurfaceColorView").GetCurrentHue())
>
>       // update the time clock preview background color //
>       GCUI.GetClockView("ClockView").UpdateClockBackgroundColor(GCUI
>            .GetColorView("ClockSurfaceColorView").GetCurrentHue())
3904,3911c3975,3985
<       // loop over all the clocks //
<       for TheClock in [GCUI.GetClockView("WhiteClockView"), GCUI
<           .GetClockView("BlackClockView"), GCUI.GetClockView("ClockView")]
<       {
<          // update the clock border color //
<          TheClock.UpdateClockBorderColor(GCUI.GetColorView("ClockColorView")
<                  .GetCurrentHue())
<       }
---
>       // update the white time clock border color //
>       WhiteTimeClock.UpdateClockBorderColor(GCUI.GetColorView("ClockColorView")
>                                             .GetCurrentHue())
>
>       // update the black time clock border color //
>       BlackTimeClock.UpdateClockBorderColor(GCUI.GetColorView("ClockColorView")
>                                             .GetCurrentHue())
>
>       // update the time clock border preview color //
>       GCUI.GetClockView("ClockView").UpdateClockBorderColor(GCUI
>            .GetColorView("ClockColorView").GetCurrentHue())
3960d4033
<       // update the game list view //
4136c4209
<       StartNewGame("")
---
>       StartNewGame()
4181c4254
<       StartNewGame("All")
---
>       StartNewGame()
4248,4250c4321,4327
<       // set clocks to refresh all //
<       (GCUI.GetClockView("WhiteClockView").RefreshAll, GCUI
<       .GetClockView("BlackClockView").RefreshAll) = (true, true)
---
>       // loop over all the clocks //
>       for TheClock in [WhiteTimeClock, BlackTimeClock, GCUI
>           .GetClockView("ClockView")]
>       {
>          // set the clock to refresh //
>          TheClock!.RefreshAll = true
>       }
4748,4749c4825,4826
<       for TheClock in [GCUI.GetClockView("WhiteClockView"), GCUI
<           .GetClockView("BlackClockView"), GCUI.GetClockView("ClockView")]
---
>       for TheClock in [WhiteTimeClock, BlackTimeClock, GCUI
>           .GetClockView("ClockView")]
4752,4753c4829,4830
<          TheClock.UpdateClockBackgroundOpacity(Inf
<                  .Vals("ClockBackgroundVisibility").VI)
---
>          TheClock!.UpdateClockBackgroundOpacity(Inf
>                   .Vals("ClockBackgroundVisibility").VI)
4961c5038
<
---
>
4965,4969c5042,5044
<       // create the frames diagram list //
<       GCUI.FramesMovie.CreateImageFrames(Inf.VF("IllustrationSize") * 2)
<
<       // update the move list font //
<       MoveListFont.UpdateFont("", Selection:0)
---
>       // update the move list font //
>       MoveListFont.UpdateFont(Inf.Vals("IllustrationSize").VI, UseVertOffset:
>                   !Inf.Vals("HideIllustrations").VB, Key:"", Selection:0)
4972c5047
<       GameInfoFont.UpdateFont("", Selection:0)
---
>       GameInfoFont.UpdateFont(0, UseVertOffset:false, Key:"", Selection:0)
4974c5049
<       // clear the pgn moves list //
---
>       // intialize the pgn moves list array //
5067a5143,5146
>
>       // refresh the pgn moves list //
>       GCUI.PGNMovesData.RefreshPGNMovesList(PGNMoveList, MoveListFont:
>                                             MoveListFont)
5069,5070c5148,5149
<       // clear the move rows list //
<       MoveRowsList.removeAll()
---
>       // refresh the game info list //
>       GameInfoData.RefreshGameInfoList(GameInfoList, GameInfoFont:GameInfoFont)
5072,5081c5151,5152
<       // clear the info rows list //
<       InfoRowsList.removeAll()
<
<       // loop over all the frame images to display //
<       for Index in 0..<GCUI.FramesMovie.ImageFrames.count
<       {
<          // add the image and text to the move rows list //
<          MoveRowsList.append(AListRow(id:Index, RowImage:GCUI.FramesMovie
<                             .ImageFrames[Index], RowText:PGNMoveList[Index]))
<       }
---
>       // reload the game data to the pgn moves table view //
>       PGNMovesTableView.reloadData()
5083,5103c5154,5161
<       // loop over the game info list //
<       for Index in 0..<GameInfoList.count
<       {
<          // add the text to the game info rows list //
<          InfoRowsList.append(AListRow(id:Index, RowImage:Image(""), RowText:
<                             GameInfoList[Index]))
<       }
<
<       // initialize the move list and game info table view values //
<       (GCUI.Objs["MoveListTableView"]!.ListRows, GCUI.Objs["MoveListTableView"]!
<       .Inf.DisplayImage, GCUI.Objs["MoveListTableView"]!.FontType, GCUI
<       .Objs["MoveListTableView"]!.TextColor, GCUI.Objs["GameInfoTableView"]!
<       .ListRows, GCUI.Objs["GameInfoTableView"]!.FontType, GCUI
<       .Objs["GameInfoTableView"]!.TextColor) = (MoveRowsList, !Inf
<       .Vals("HideIllustrations").VB, MoveListFont.FontType, Color(GCUI
<       .GetColorView("MoveListColorView").GetCurrentHue()), InfoRowsList,
<       GameInfoFont.FontType, Color(GCUI.GetColorView("GameInfoColorView")
<       .GetCurrentHue()))
<
<       // refresh the table views //
<       GCUI.RefreshViews(["MoveListTableView", "GameInfoTableView"])
---
>       // reload the game info list //
>       GameInfoTableView.reloadData()
>
>       // scroll to the last row of the pgn moves table view //
>       PGNMovesTableView.scrollRowToVisible(GCUI.PGNMovesData.LastRow())
>
>       // scroll to the first row in the game info table view //
>       GameInfoTableView.scrollRowToVisible(0)
5261c5319
<
---
>
5573c5631
<             StartNewGame("All")
---
>             StartNewGame()
```