```
2c2
< // Swift source code of main.swift Version 8.2.4 at greenchess.com //
---
> // Swift source code of main.swift Version 8.2.3 at greenchess.com //
512,514d511
<    // the annotation glyph index //
<    var GlyphIndex:Int = 0
<
897c894
<    let ChessLightSquareNames:[String] = ["light_bamboo", "light_white", "light_white",
---
>    var ChessLightSquareNames:[String] = ["light_bamboo", "light_white", "light_white",
905c902
<    let ChessDarkSquareNames:[String] = ["dark_bamboo", "dark_blue", "dark_brown", "dark_chalk",
---
>    var ChessDarkSquareNames:[String] = ["dark_bamboo", "dark_blue", "dark_brown", "dark_chalk",
912c909
<    let ChessPieceNames:[String] = ["whitepawn1", "whiteknight1", "whitebishop1", "whiterook1",
---
>    var ChessPieceNames:[String] = ["whitepawn1", "whiteknight1", "whitebishop1", "whiterook1",
926c923
<    let ChessPieceSetNames:[String] = ["Standard Pieces", "3D Pieces", "3D Textured Pieces",
---
>    var ChessPieceSetNames:[String] = ["Standard Pieces", "3D Pieces", "3D Textured Pieces",
930c927
<    let ChessBoardNames:[String] = ["Bamboo Board", "Blue Board", "Brown Board", "Chalk Board",
---
>    var ChessBoardNames:[String] = ["Bamboo Board", "Blue Board", "Brown Board", "Chalk Board",
937c934
<    let GameSoundNames:[String] = ["sound_thump", "sound_wrongsquare", "sound_pop", "sound_click",
---
>    var GameSoundNames:[String] = ["sound_thump", "sound_wrongsquare", "sound_pop", "sound_click",
943c940
<    let RankSymbols:[String] = ["1", "2", "3", "4", "5", "6", "7", "8"]
---
>    var RankSymbols:[String] = ["1", "2", "3", "4", "5", "6", "7", "8"]
946c943
<    let FileSymbols:[String] = ["a", "b", "c", "d", "e", "f", "g", "h"]
---
>    var FileSymbols:[String] = ["a", "b", "c", "d", "e", "f", "g", "h"]
1888c1885
<    let FENLayoutCharacters:[String] = ["k", "q", "r", "b", "n", "p", "", "P", "N", "B", "R", "Q",
---
>    var FENLayoutCharacters:[String] = ["k", "q", "r", "b", "n", "p", "", "P", "N", "B", "R", "Q",
1892c1889
<    let PieceDictionary:[String:Int] = ["p":-1, "n":-2, "b":-3, "r":-4, "q":-5, "k":-6, "P":1,
---
>    var PieceDictionary:[String:Int] = ["p":-1, "n":-2, "b":-3, "r":-4, "q":-5, "k":-6, "P":1,
1896c1893
<    let RankFileDictionary:[Character:Int] = ["a":0, "b":1, "c":2, "d":3, "e":4, "f":5, "g":6,
---
>    var RankFileDictionary:[Character:Int] = ["a":0, "b":1, "c":2, "d":3, "e":4, "f":5, "g":6,
1901c1898
<    let SoundDictionary:[String:Int] = ["thump":0, "wrongsquare":1, "pop":2, "click":3, "clear":4,
---
>    var SoundDictionary:[String:Int] = ["thump":0, "wrongsquare":1, "pop":2, "click":3, "clear":4,
1906c1903
<    let SoundVolumes:[Float] = [1, 0.25, 0.4, 0.1, 0.25, 0.3, 0.4, 0.3, 0.7, 0.7, 0.3]
---
>    var SoundVolumes:[Float] = [1, 0.25, 0.4, 0.1, 0.25, 0.3, 0.4, 0.3, 0.7, 0.7, 0.3]
1909c1906
<    let KCastles:[String] = ["O-O", "O-O+", "O-O#"]
---
>    var KCastles:[String] = ["O-O", "O-O+", "O-O#"]
1912c1909
<    let QCastles:[String] = ["O-O-O", "O-O-O+", "O-O-O#"]
---
>    var QCastles:[String] = ["O-O-O", "O-O-O+", "O-O-O#"]
1915,1930c1912
<    let PromotionDictionary:[Character:Int] = ["n":2, "b":3, "r":4, "q":5]
<
<    // the white piece symbol dictionary //
<    let WhitePieceDictionary:[String:String] = ["K":"â™”", "Q":"â™•", "R":"â™–", "B":"â™—", "N":"â™˜"]
<
<    // the black piece symbol dictionary //
<    let BlackPieceDictionary:[String:String] = ["K":"â™š", "Q":"â™›", "R":"â™œ", "B":"â™", "N":"â™ž"]
<
<    // the annotation glyph codes //
<    let GlyphCode:[String] = ["$133", "$132", "$41", "$40", "$37", "$36", "$33", "$32", "$23",
<                              "$22", "$19", "$18", "$17", "$16", "$15", "$14", "$13", "$10", "$7",
<                              "$6", "$5", "$4", "$3", "$2", "$1"]
<
<    // the annotation glyph symbols //
<    let GlyphSymbol:[String] = ["â‡†", "â‡†", "â†‘", "â†‘", "â†’", "â†’", "âŸ³", "âŸ³", "â¨€", "â¨€", "âˆ’ +", "+ âˆ’",
<                                "âˆ“", "Â±", "â©±", "â©²", "âˆž", "=", "â–¡", "âˆ", "â‰", "â‡", "â€¼", "?", "!"]
---
>    var PromotionDictionary:[Character:Int] = ["n":2, "b":3, "r":4, "q":5]
5938c5920
<       AppNameView.stringValue = "GreenChess 8.2.4"
---
>       AppNameView.stringValue = "GreenChess 8.2.3"
7691,7693c7673,7675
<                                  AVVideoCodecType.h264, AVVideoWidthKey:NSNumber(value:
<                                  Int(VidWidth * Scale)), AVVideoHeightKey:NSNumber(value:
<                                  Int(VidHeight * Scale))])
---
>                                  AVVideoCodecH264, AVVideoWidthKey:NSNumber(value:Int(VidWidth *
>                                  Scale)), AVVideoHeightKey:NSNumber(value:Int(VidHeight *
>                                  Scale))])
8065c8047
<       EngineTask.executableURL = URL(fileURLWithPath:Filename)
---
>       EngineTask.launchPath = Filename
8083,8084c8065,8066
<       // run the uci chess engine //
<       try? EngineTask.run()
---
>       // launch the uci chess engine //
>       EngineTask.launch()
11565,11571c11547,11620
<       // for each element in the annotation glyph code list //
<       for GlyphIndex in 0 ..< GlyphCode.count
<       {
<          // replace the code occurrences with the corresponding symbols //
<          AnnotationString = AnnotationString.replacingOccurrences(of:GlyphCode[GlyphIndex], with:
<                                                                   GlyphSymbol[GlyphIndex])
<       }
---
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$133", with:"â‡†")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$132", with:"â‡†")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$41", with:"â†‘")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$40", with:"â†‘")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$37", with:"â†’")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$36", with:"â†’")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$33", with:"âŸ³")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$32", with:"âŸ³")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$23", with:"â¨€")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$22", with:"â¨€")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$19", with:"âˆ’ +")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$18", with:"+ âˆ’")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$17", with:"âˆ“")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$16", with:"Â±")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$15", with:"â©±")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$14", with:"â©²")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$13", with:"âˆž")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$10", with:"=")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$7", with:"â–¡")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$6", with:"âˆ")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$5", with:"â‰")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$4", with:"â‡")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$3", with:"â€¼")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$2", with:"?")
>
>       // convert the numeric annotation glyph //
>       AnnotationString = AnnotationString.replacingOccurrences(of:"$1", with:"!")
12471,12476c12520,12533
<          // for all the elements in the white piece symbols dictionary //
<          for (Code, Symbol) in WhitePieceDictionary
<          {
<             // replace each code occurrence with the corresponding symbol //
<             PGNLabel = PGNLabel.replacingOccurrences(of:Code, with:Symbol)
<          }
---
>          // replace white king with symbol representation for white king //
>          PGNLabel = PGNLabel.replacingOccurrences(of:"K", with:"â™”")
>
>          // replace white queen with symbol representation for white queen //
>          PGNLabel = PGNLabel.replacingOccurrences(of:"Q", with:"â™•")
>
>          // replace white rook with symbol representation for white rook //
>          PGNLabel = PGNLabel.replacingOccurrences(of:"R", with:"â™–")
>
>          // replace white bishop with symbol representation for white bishop //
>          PGNLabel = PGNLabel.replacingOccurrences(of:"B", with:"â™—")
>
>          // replace white knight with symbol representation for white knight //
>          PGNLabel = PGNLabel.replacingOccurrences(of:"N", with:"â™˜")
12479c12536
<          if FileSymbols.contains(String(PGNLabel.first!))
---
>          if (PGNLabel.first! >= "a") && (PGNLabel.first! <= "h")
12488,12493c12545,12558
<          // for all the elements in the black piece symbols dictionary //
<          for (Code, Symbol) in BlackPieceDictionary
<          {
<             // replace each code occurrence with the corresponding symbol //
<             PGNLabel = PGNLabel.replacingOccurrences(of:Code, with:Symbol)
<          }
---
>          // replace black king with symbol representation for black king //
>          PGNLabel = PGNLabel.replacingOccurrences(of:"K", with:"â™š")
>
>          // replace black queen with symbol representation for black queen //
>          PGNLabel = PGNLabel.replacingOccurrences(of:"Q", with:"â™›")
>
>          // replace black rook with symbol representation for black rook //
>          PGNLabel = PGNLabel.replacingOccurrences(of:"R", with:"â™œ")
>
>          // replace black bishop with symbol representation for black bishop //
>          PGNLabel = PGNLabel.replacingOccurrences(of:"B", with:"â™")
>
>          // replace black knight with symbol representation for black knight //
>          PGNLabel = PGNLabel.replacingOccurrences(of:"N", with:"â™ž")
12496c12561
<          if FileSymbols.contains(String(PGNLabel.first!))
---
>          if (PGNLabel.first! >= "a") && (PGNLabel.first! <= "h")
```