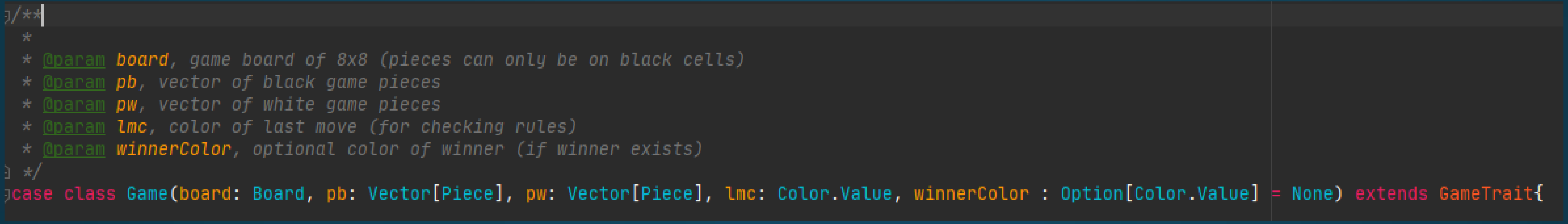
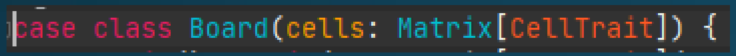
Checkers – Group 7

What is Checkers?

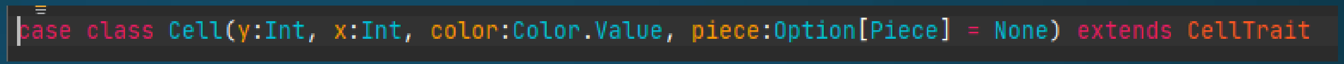
Data Structures



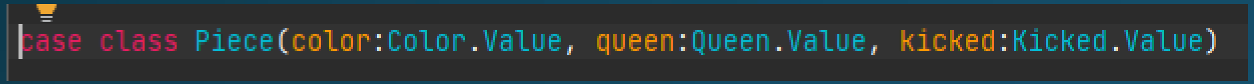
```
/**
 *
 * @param board, game board of 8x8 (pieces can only be on black cells)
 * @param pb, vector of black game pieces
 * @param pw, vector of white game pieces
 * @param lmc, color of last move (for checking rules)
 * @param winnerColor, optional color of winner (if winner exists)
 */
case class Game(board: Board, pb: Vector[Piece], pw: Vector[Piece], lmc: Color.Value, winnerColor : Option[Color.Value] = None) extends GameTrait{
```



```
case class Board(cells: Matrix[CellTrait]) {
```

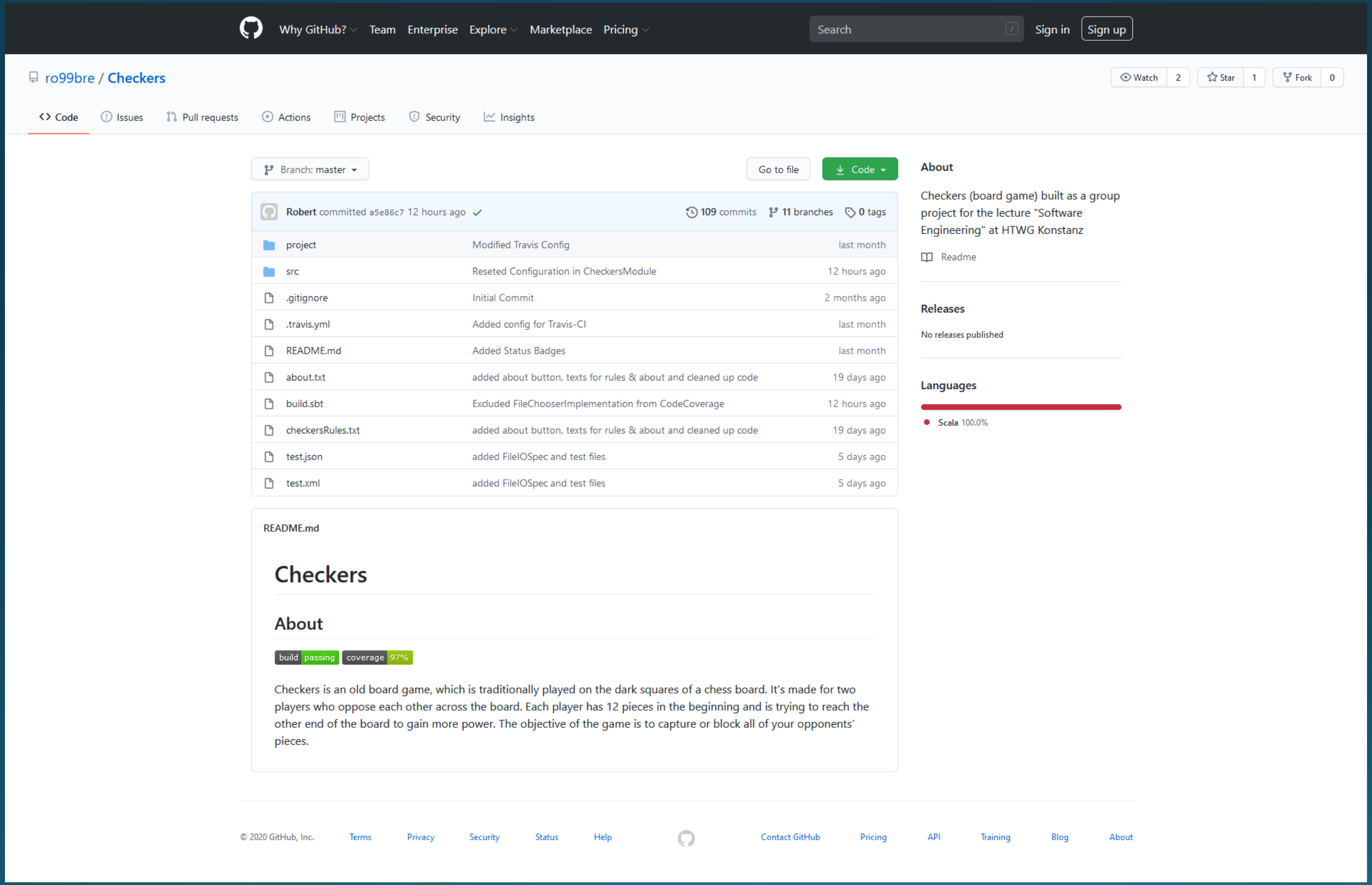


```
case class Cell(y:Int, x:Int, color:Color.Value, piece:Option[Piece] = None) extends CellTrait
```



```
case class Piece(color:Color.Value, queen:Queen.Value, kicked:Kicked.Value)
```

Git and GitHub



ro99bre / Checkers

Robert committed a5e86c7 12 hours ago ✓ 109 commits 11 branches 0 tags

project	Modified Travis Config	last month
src	Reseted Configuration in CheckersModule	12 hours ago
.gitignore	Initial Commit	2 months ago
.travis.yml	Added config for Travis-CI	last month
README.md	Added Status Badges	last month
about.txt	added about button, texts for rules & about and cleaned up code	19 days ago
build.sbt	Excluded FileChooserImplementation from CodeCoverage	12 hours ago
checkersRules.txt	added about button, texts for rules & about and cleaned up code	19 days ago
test.json	added FileIOSpec and test files	5 days ago
test.xml	added FileIOSpec and test files	5 days ago

About

Checkers (board game) built as a group project for the lecture "Software Engineering" at HTWG Konstanz

Releases

No releases published

Languages

● Scala 100.0%

README.md

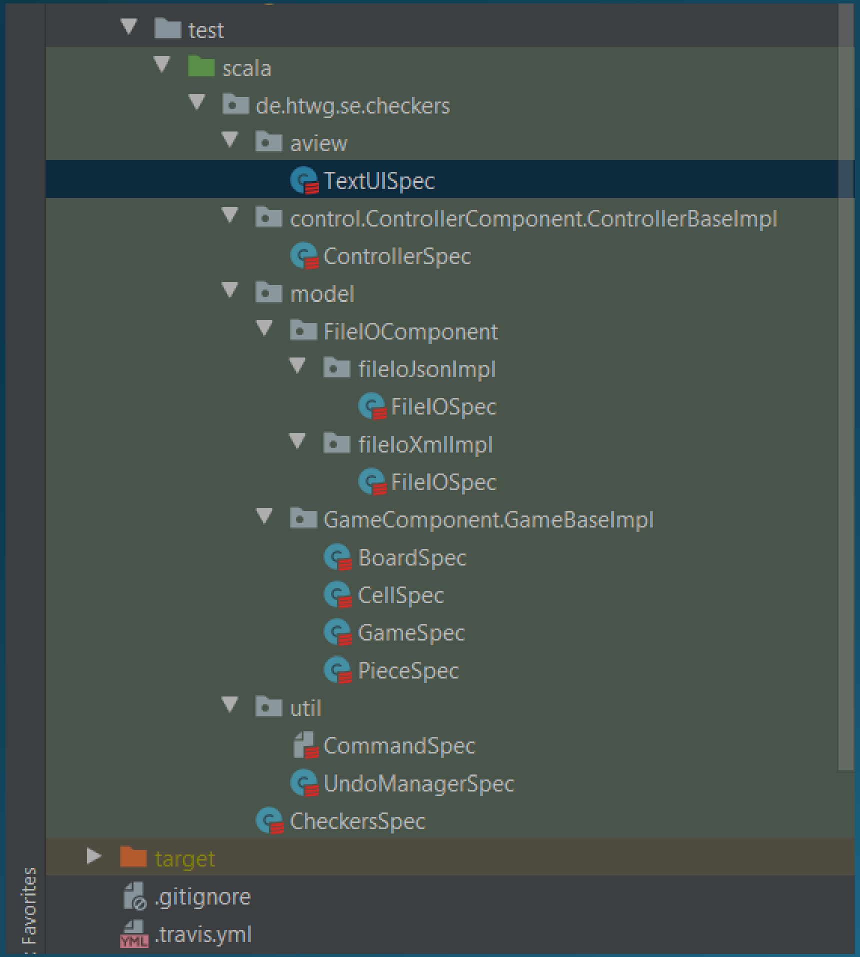
Checkers

About

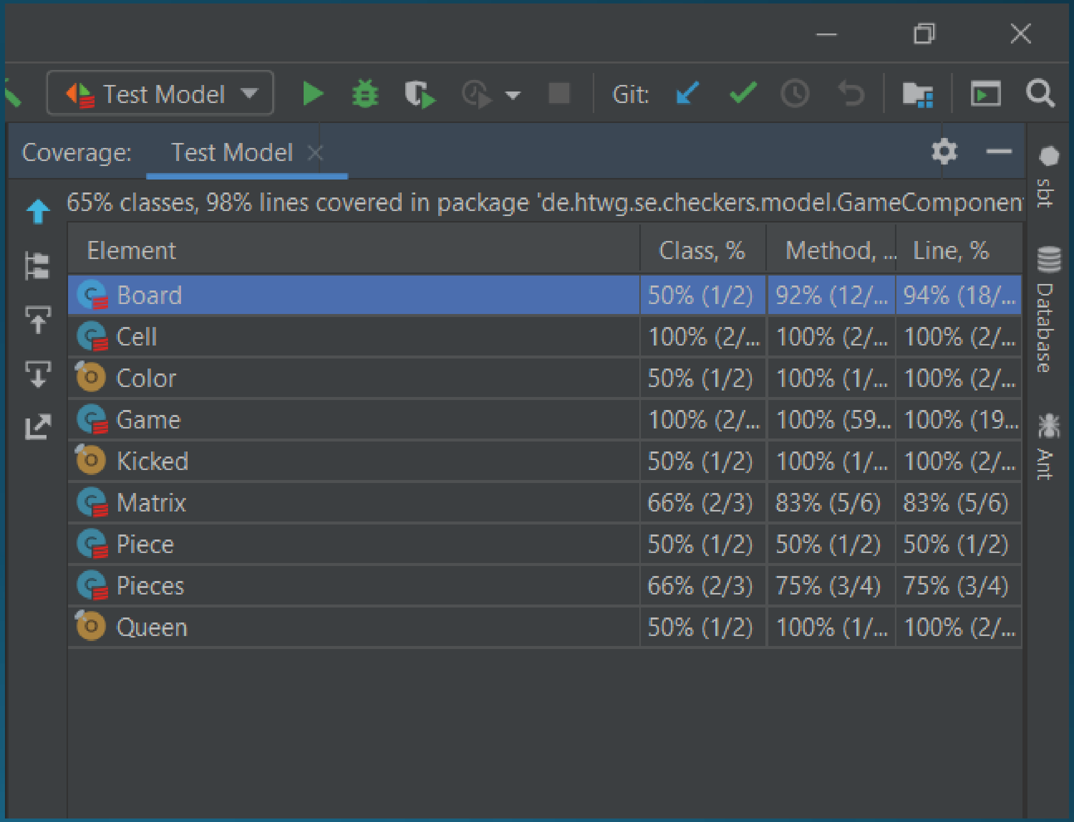
build passing coverage 97%

Checkers is an old board game, which is traditionally played on the dark squares of a chess board. It's made for two players who oppose each other across the board. Each player has 12 pieces in the beginning and is trying to reach the other end of the board to gain more power. The objective of the game is to capture or block all of your opponents' pieces.

Tests



```
test
  scala
    de.htwg.se.checkers
      aview
        TextUISpec
      control.ControllerComponent.ControllerBaseImpl
        ControllerSpec
      model
        FileIOComponent
          fileIoJsonImpl
            FileIOSpec
          fileIoXmlImpl
            FileIOSpec
        GameComponent.GameBaseImpl
          BoardSpec
          CellSpec
          GameSpec
          PieceSpec
      util
        CommandSpec
        UndoManagerSpec
      CheckersSpec
target
.gitignore
.travis.yml
```

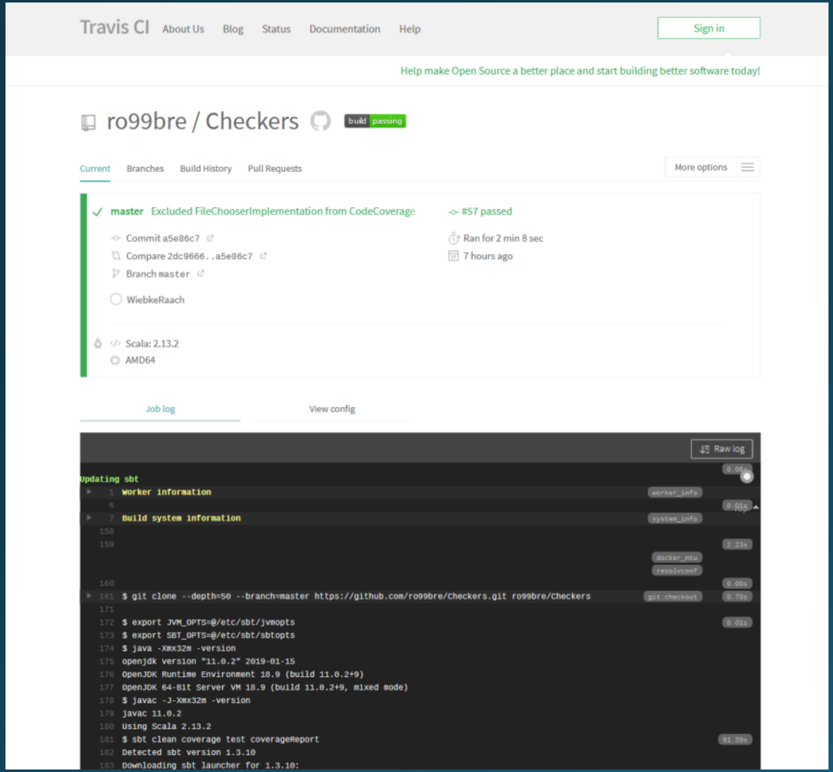


Coverage: Test Model

65% classes, 98% lines covered in package 'de.htwg.se.checkers.model.GameComponen

Element	Class, %	Method, ...	Line, %
Board	50% (1/2)	92% (12/...	94% (18/...
Cell	100% (2/...	100% (2/...	100% (2/...
Color	50% (1/2)	100% (1/...	100% (2/...
Game	100% (2/...	100% (59...	100% (19...
Kicked	50% (1/2)	100% (1/...	100% (2/...
Matrix	66% (2/3)	83% (5/6)	83% (5/6)
Piece	50% (1/2)	50% (1/2)	50% (1/2)
Pieces	66% (2/3)	75% (3/4)	75% (3/4)
Queen	50% (1/2)	100% (1/...	100% (2/...

Continous Integration



Help make Open Source a better place and start building better software today!

ro99bre / Checkers

✓ master Excluded FileChooserImplementation from CodeCoverage

#57 passed

Commit a5e86c7

Compare 2dc9666..a5e86c7

Branch master

WiebkeRaach

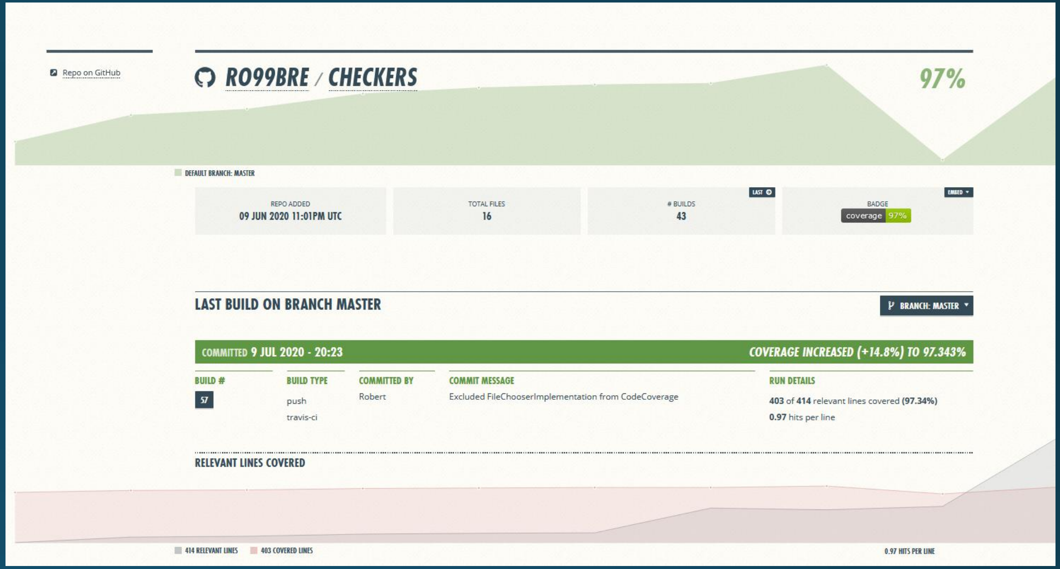
Ran for 2 min 8 sec

7 hours ago

Scala: 2.13.2

AMD64

```
Updating sbt
Worker information

Build system information



$ git clone --depth=50 --branch=master https://github.com/ro99bre/Checkers.git ro99bre/Checkers

$ export JVM_OPTS=@/etc/sbt/jvmopts
$ export SBT_OPTS=@/etc/sbt/sbtopts
$ java -Xmx32m -version
openjdk version "11.0.2" 2019-01-15
OpenJDK Runtime Environment 18.9 (build 11.0.2+9)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.2+9, mixed mode)
$ javac -J-Xmx32m -version
javac 11.0.2
Using Scala 2.13.2
$ sbt clean coverage test coverageReport
Detected sbt version 1.3.10
Downloading sbt launcher for 1.3.10:
```



REPO ADDED 09 JUN 2020 11:01PM UTC

TOTAL FILES 16

BUILDS 43

BADGE coverage 97%

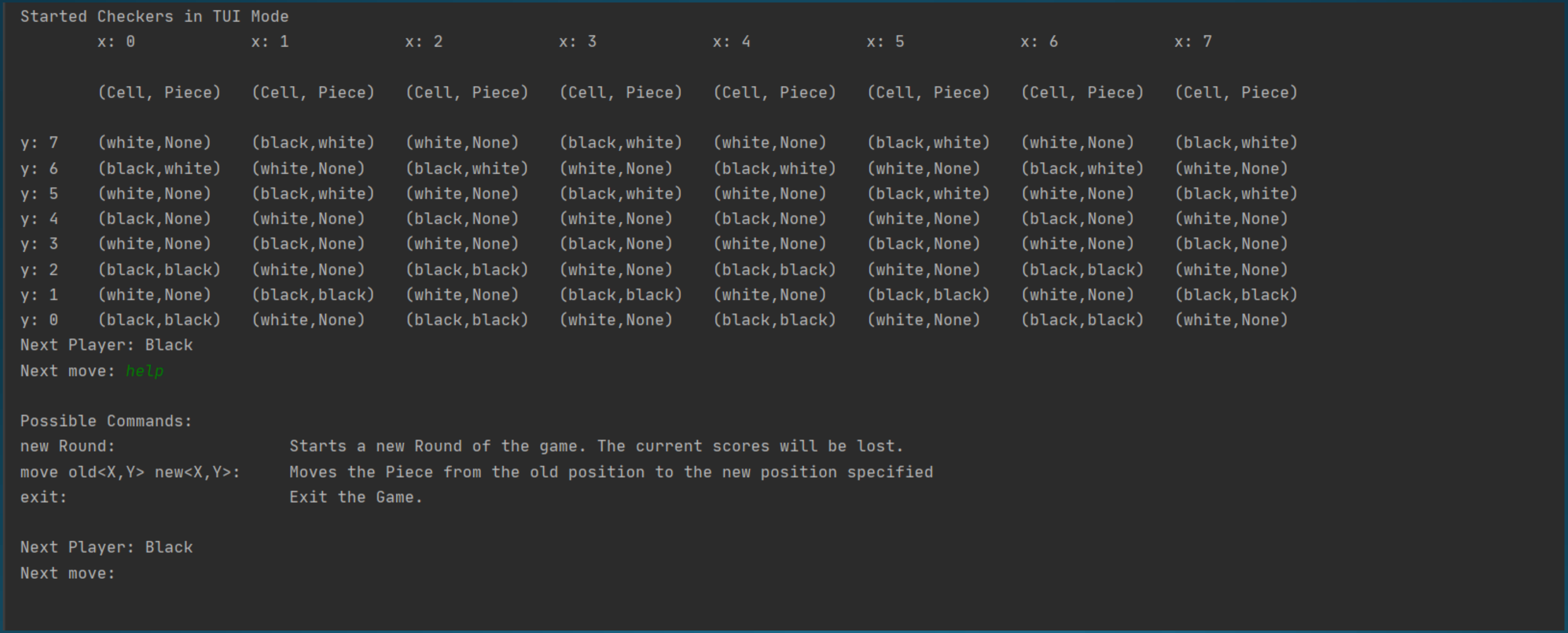
LAST BUILD ON BRANCH MASTER

COMMITTED 9 JUL 2020 - 20:23 COVERAGE INCREASED (+14.8%) TO 97.343%

BUILD #	BUILD TYPE	COMMITTED BY	COMMIT MESSAGE	RUN DETAILS
57	push travis-ci	Robert	Excluded FileChooserImplementation from CodeCoverage	403 of 414 relevant lines covered (97.34%) 0.97 hits per line

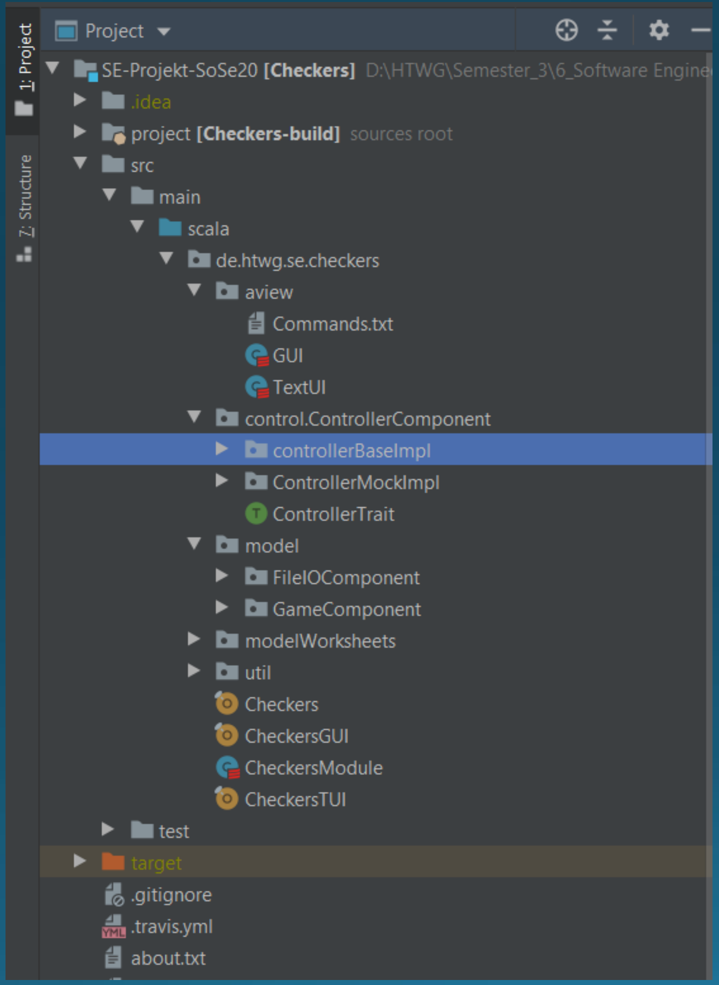
RELEVANT LINES COVERED

TextUI



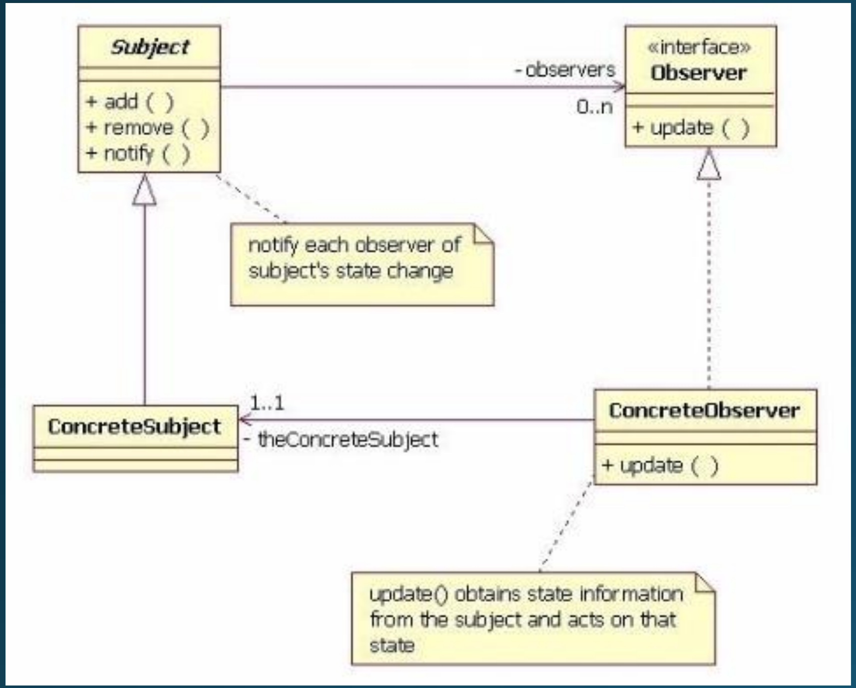
```
Started Checkers in TUI Mode
        x: 0            x: 1            x: 2            x: 3            x: 4            x: 5            x: 6            x: 7

        (Cell, Piece)   (Cell, Piece)   (Cell, Piece)   (Cell, Piece)   (Cell, Piece)   (Cell, Piece)   (Cell, Piece)   (Cell, Piece)

y: 7    (white,None)    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)
y: 6    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)   (white,None)
y: 5    (white,None)    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)
y: 4    (black,None)    (white,None)    (black,None)    (white,None)    (black,None)    (white,None)    (black,None)    (white,None)
y: 3    (white,None)    (black,None)    (white,None)    (black,None)    (white,None)    (black,None)    (white,None)    (black,None)
y: 2    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)   (white,None)
y: 1    (white,None)    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)
y: 0    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)   (white,None)
Next Player: Black
Next move: help

Possible Commands:
new Round:                  Starts a new Round of the game. The current scores will be lost.
move old<X,Y> new<X,Y>:     Moves the Piece from the old position to the new position specified
exit:                       Exit the Game.

Next Player: Black
Next move:
```

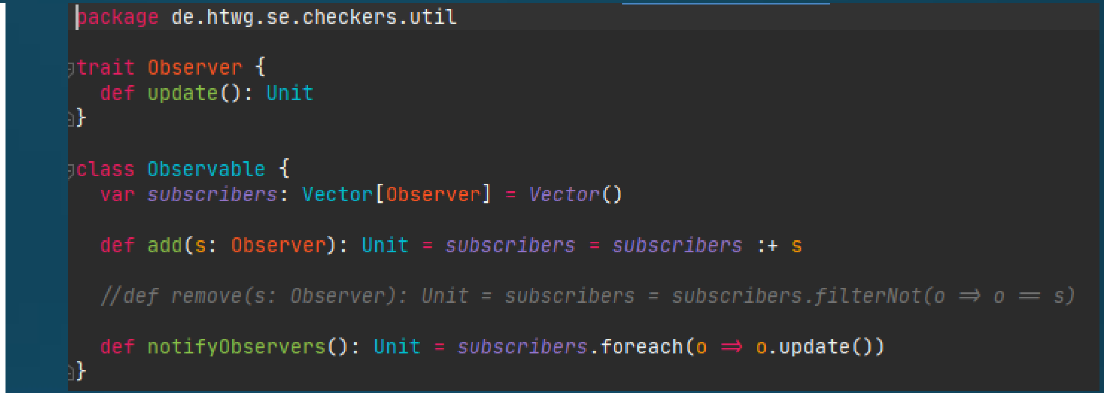
MVC-Architecture



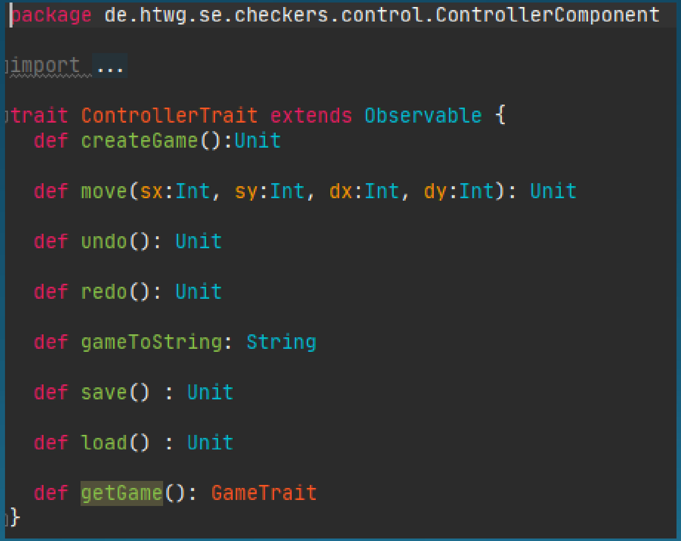
```
SE-Projekt-SoSe20 [Checkers]  D:\HTWG\Semester_3\6_Software Engine
  .idea
  project [Checkers-build]  sources root
  src
    main
      scala
        de.htwg.se.checkers
          aview
            Commands.txt
            GUI
            TextUI
          control.ControllerComponent
            controllerBaseImpl
            ControllerMockImpl
            ControllerTrait
          model
            FileIOComponent
            GameComponent
          modelWorksheets
          util
          Checkers
          CheckersGUI
          CheckersModule
          CheckersTUI
    test
  target
  .gitignore
  .travis.yml
  about.txt
```

Design Pattern (1) – Observer Pattern

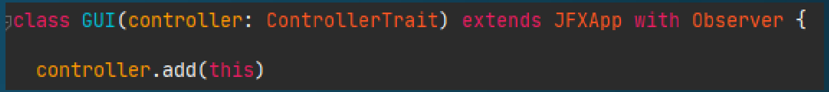




```scala
package de.htwg.se.checkers.util

trait Observer {
  def update(): Unit
}

class Observable {
  var subscribers: Vector[Observer] = Vector()

  def add(s: Observer): Unit = subscribers = subscribers :+ s

  //def remove(s: Observer): Unit = subscribers = subscribers.filterNot(o => o == s)

  def notifyObservers(): Unit = subscribers.foreach(o => o.update())
}
```

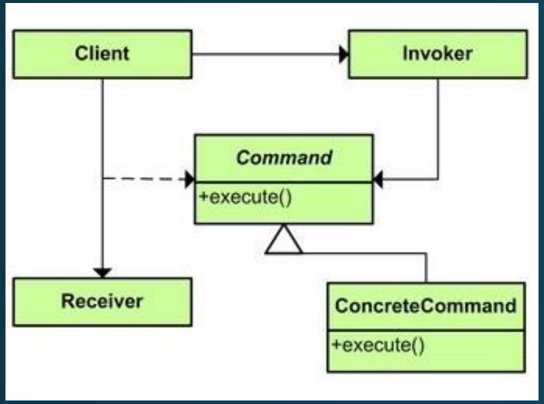


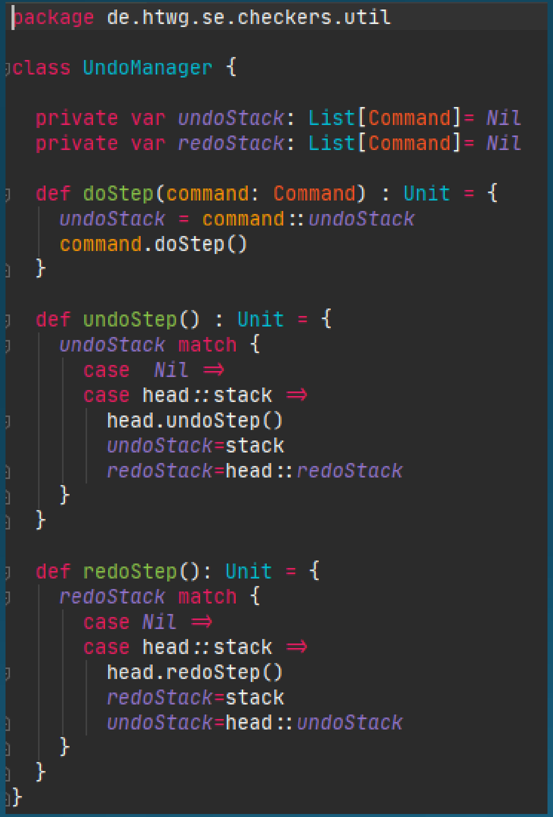
```scala
package de.htwg.se.checkers.control.ControllerComponent

import ...

trait ControllerTrait extends Observable {
  def createGame():Unit

  def move(sx:Int, sy:Int, dx:Int, dy:Int): Unit

  def undo(): Unit

  def redo(): Unit

  def gameToString: String

  def save() : Unit

  def load() : Unit

  def getGame(): GameTrait
}
```



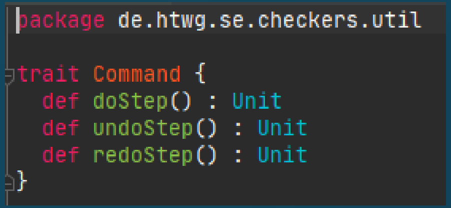
```scala
class GUI(controller: ControllerTrait) extends JFXApp with Observer {

  controller.add(this)
```

Design Pattern (2) – Command Pattern

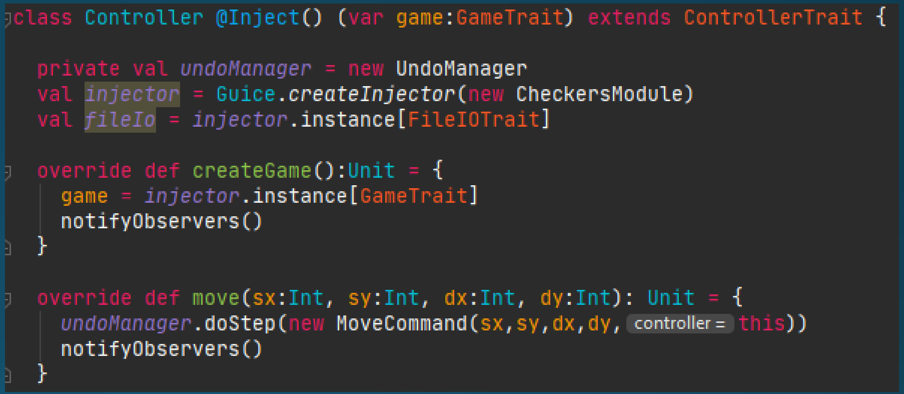




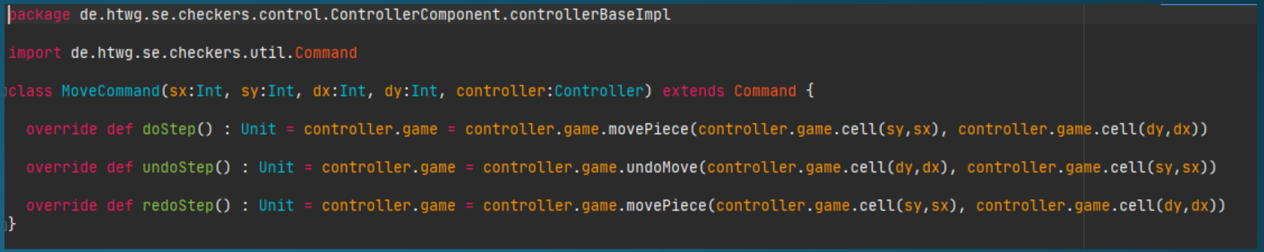
```
package de.htwg.se.checkers.util

class UndoManager {

  private var undoStack: List[Command]= Nil
  private var redoStack: List[Command]= Nil

  def doStep(command: Command) : Unit = {
    undoStack = command::undoStack
    command.doStep()
  }

  def undoStep() : Unit = {
    undoStack match {
      case  Nil ⇒
      case head::stack ⇒
        head.undoStep()
        undoStack=stack
        redoStack=head::redoStack
    }
  }

  def redoStep(): Unit = {
    redoStack match {
      case Nil ⇒
      case head::stack ⇒
        head.redoStep()
        redoStack=stack
        undoStack=head::undoStack
    }
  }
}
```



```
package de.htwg.se.checkers.util

trait Command {
  def doStep() : Unit
  def undoStep() : Unit
  def redoStep() : Unit
}
```

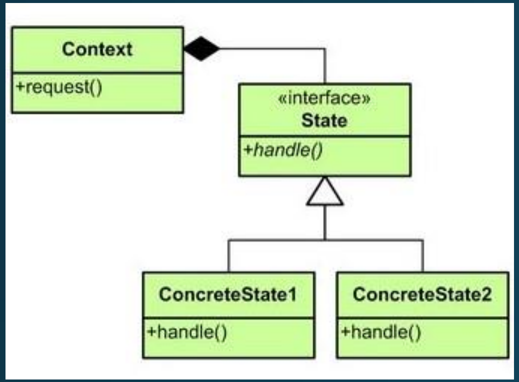


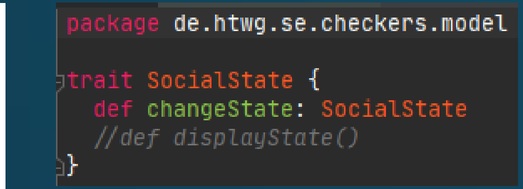
```
class Controller @Inject() (var game:GameTrait) extends ControllerTrait {

  private val undoManager = new UndoManager
  val injector = Guice.createInjector(new CheckersModule)
  val fileIo = injector.instance[FileIOTrait]

  override def createGame():Unit = {
    game = injector.instance[GameTrait]
    notifyObservers()
  }

  override def move(sx:Int, sy:Int, dx:Int, dy:Int): Unit = {
    undoManager.doStep(new MoveCommand(sx,sy,dx,dy, controller = this))
    notifyObservers()
  }
```



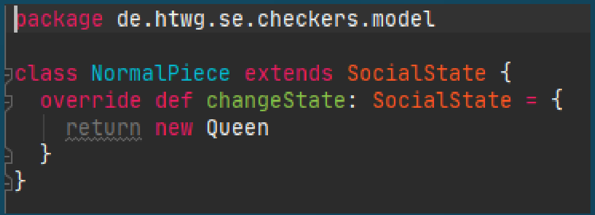
```
package de.htwg.se.checkers.control.ControllerComponent.controllerBaseImpl

import de.htwg.se.checkers.util.Command

class MoveCommand(sx:Int, sy:Int, dx:Int, dy:Int, controller:Controller) extends Command {

  override def doStep() : Unit = controller.game = controller.game.movePiece(controller.game.cell(sy,sx), controller.game.cell(dy,dx))

  override def undoStep() : Unit = controller.game = controller.game.undoMove(controller.game.cell(dy,dx), controller.game.cell(sy,sx))

  override def redoStep() : Unit = controller.game = controller.game.movePiece(controller.game.cell(sy,sx), controller.game.cell(dy,dx))
}
```

Design Pattern (x) – State Pattern

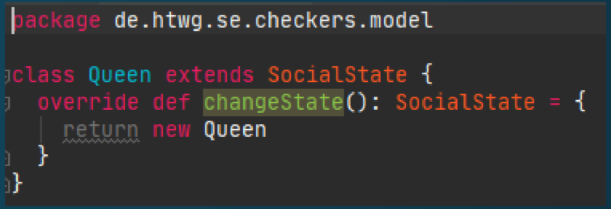




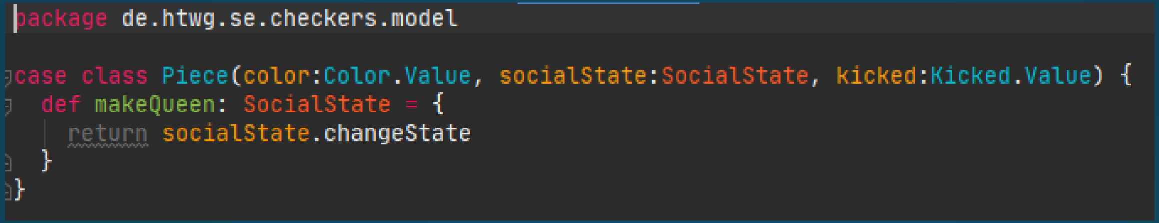
```
package de.htwg.se.checkers.model

trait SocialState {
  def changeState: SocialState
  //def displayState()
}
```



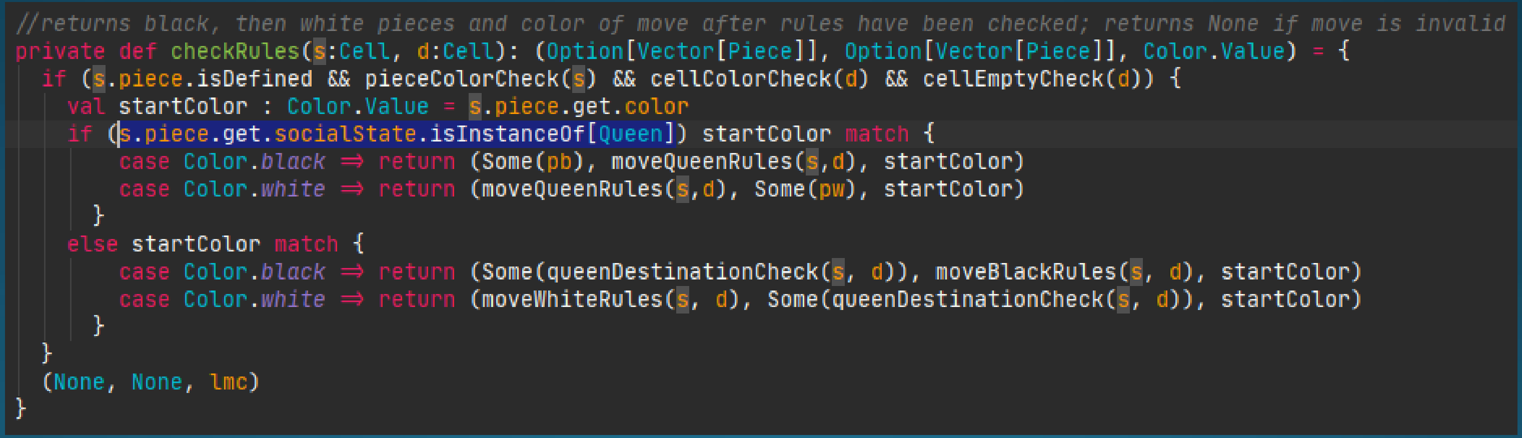
```
package de.htwg.se.checkers.model

class NormalPiece extends SocialState {
  override def changeState: SocialState = {
    return new Queen
  }
}
```



```
package de.htwg.se.checkers.model

class Queen extends SocialState {
  override def changeState(): SocialState = {
    return new Queen
  }
}
```

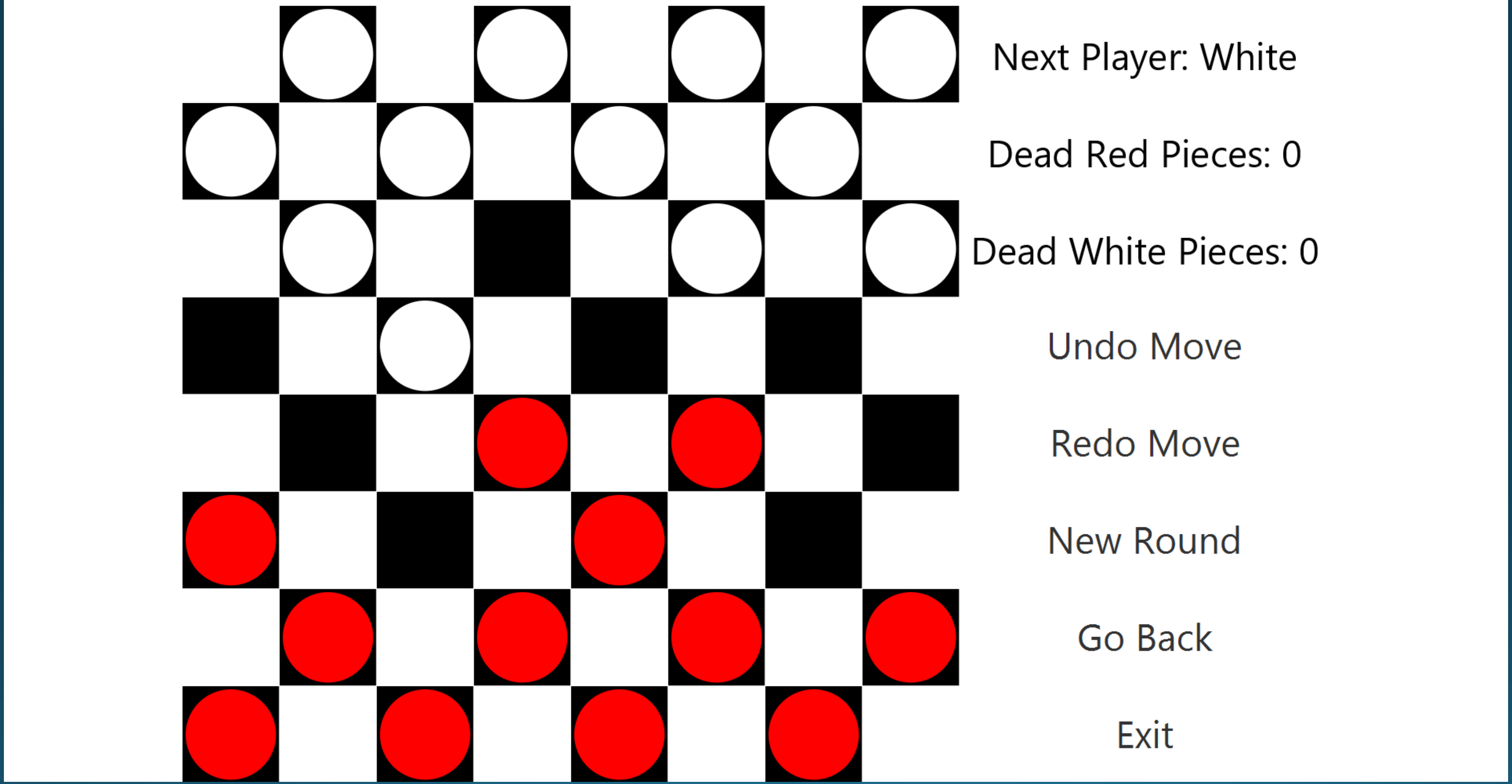


```
package de.htwg.se.checkers.model

case class Piece(color:Color.Value, socialState:SocialState, kicked:Kicked.Value) {
  def makeQueen: SocialState = {
    return socialState.changeState
  }
}
```



```
//returns black, then white pieces and color of move after rules have been checked; returns None if move is invalid
private def checkRules(s:Cell, d:Cell): (Option[Vector[Piece]], Option[Vector[Piece]], Color.Value) = {
  if (s.piece.isDefined && pieceColorCheck(s) && cellColorCheck(d) && cellEmptyCheck(d)) {
    val startColor : Color.Value = s.piece.get.color
    if (s.piece.get.socialState.isInstanceOf[Queen]) startColor match {
        case Color.black ⇒ return (Some(pb), moveQueenRules(s,d), startColor)
        case Color.white ⇒ return (moveQueenRules(s,d), Some(pw), startColor)
      }
    else startColor match {
        case Color.black ⇒ return (Some(queenDestinationCheck(s, d)), moveBlackRules(s, d), startColor)
        case Color.white ⇒ return (moveWhiteRules(s, d), Some(queenDestinationCheck(s, d)), startColor)
      }
  }
  (None, None, lmc)
}
```

GUI

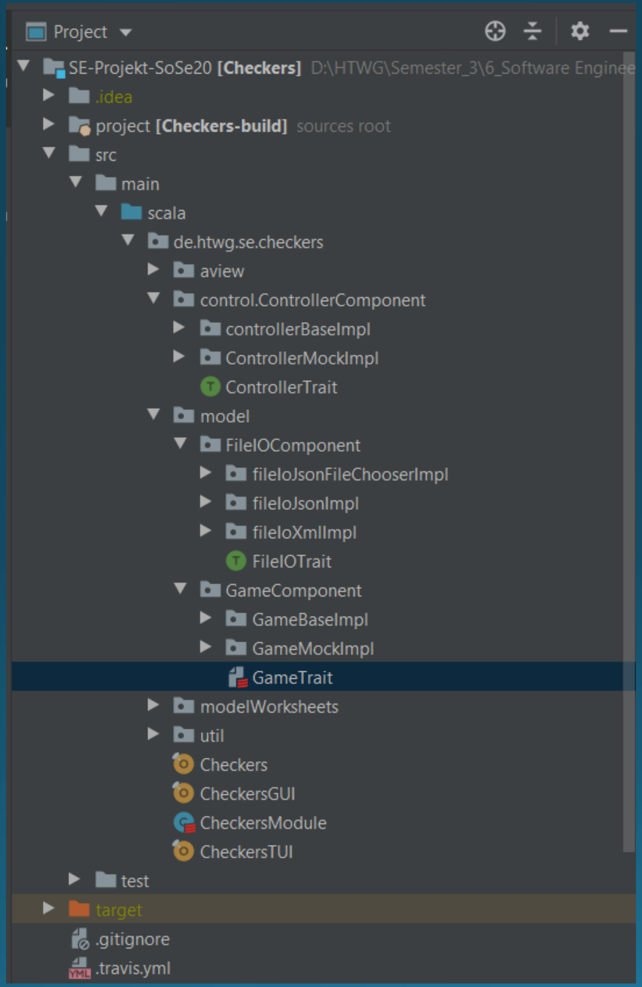


Next Player: White

Dead Red Pieces: 0

Dead White Pieces: 0

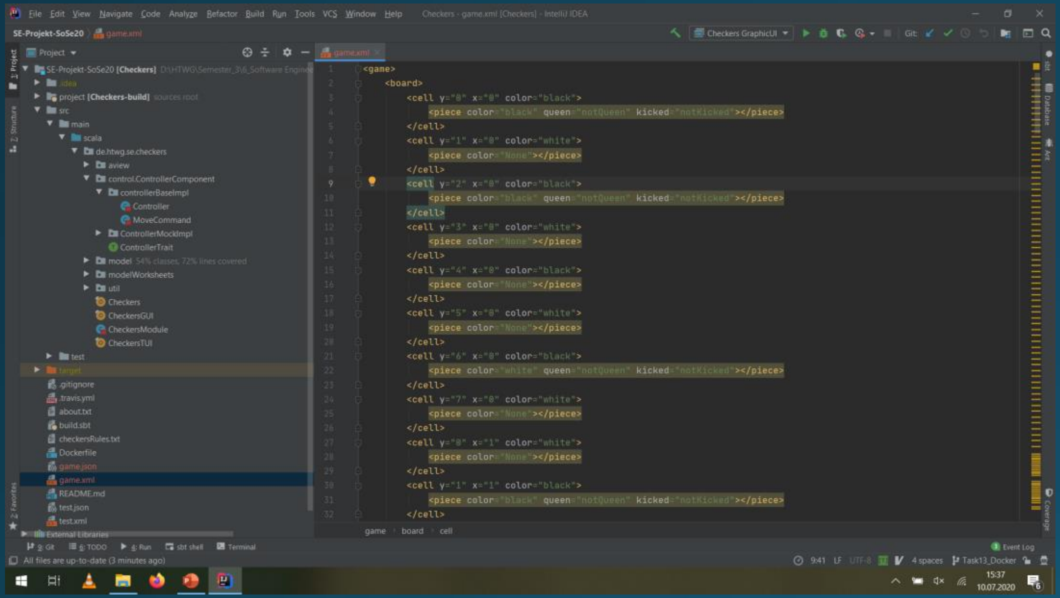
Components



```
Project
SE-Projekt-SoSe20 [Checkers] D:\HTWG\Semester_3\6_Software Enginee
  .idea
  project [Checkers-build] sources root
  src
    main
      scala
        de.htwg.se.checkers
          aview
          control.ControllerComponent
            controllerBaseImpl
            ControllerMockImpl
            ControllerTrait
          model
            FileIOComponent
              fileIoJsonFileChooserImpl
              fileIoJsonImpl
              fileIoXmlImpl
              FileIOTrait
            GameComponent
              GameBaseImpl
              GameMockImpl
              GameTrait
          modelWorksheets
          util
          Checkers
          CheckersGUI
          CheckersModule
          CheckersTUI
    test
  target
  .gitignore
  .travis.yml
```

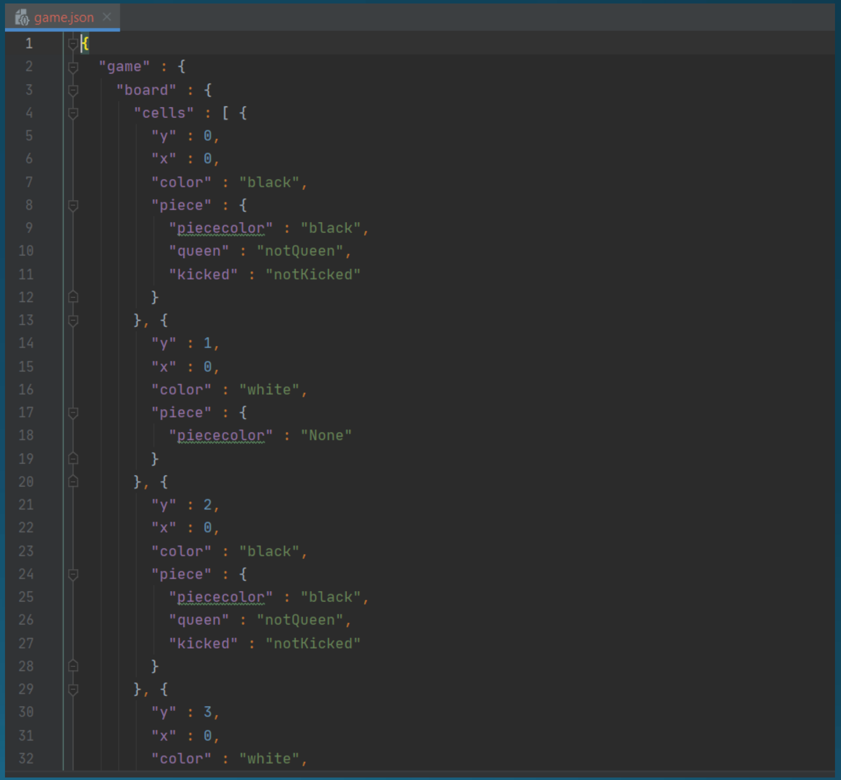
FileIO

XML



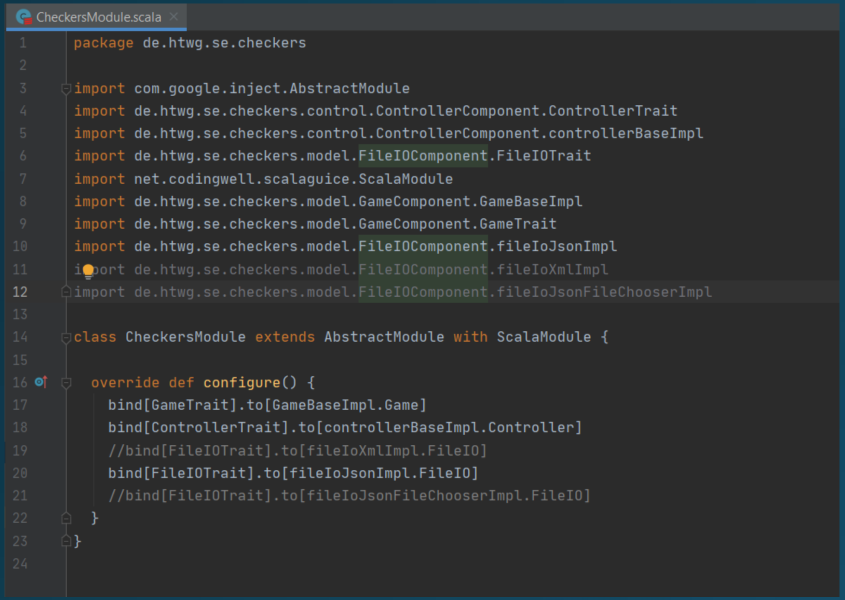
```xml
<game>
    <board>
        <cell y="0" x="0" color="black">
            <piece color="black" queen="notQueen" kicked="notKicked"></piece>
        </cell>
        <cell y="1" x="0" color="white">
            <piece color="None"></piece>
        </cell>
        <cell y="2" x="0" color="black">
            <piece color="black" queen="notQueen" kicked="notKicked"></piece>
        </cell>
        <cell y="3" x="0" color="white">
            <piece color="None"></piece>
        </cell>
        <cell y="4" x="0" color="black">
            <piece color="None"></piece>
        </cell>
        <cell y="5" x="0" color="white">
            <piece color="None"></piece>
        </cell>
        <cell y="6" x="0" color="black">
            <piece color="white" queen="notQueen" kicked="notKicked"></piece>
        </cell>
        <cell y="7" x="0" color="white">
            <piece color="None"></piece>
        </cell>
        <cell y="0" x="1" color="white">
            <piece color="None"></piece>
        </cell>
        <cell y="1" x="1" color="black">
            <piece color="black" queen="notQueen" kicked="notKicked"></piece>
        </cell>
```

JSON

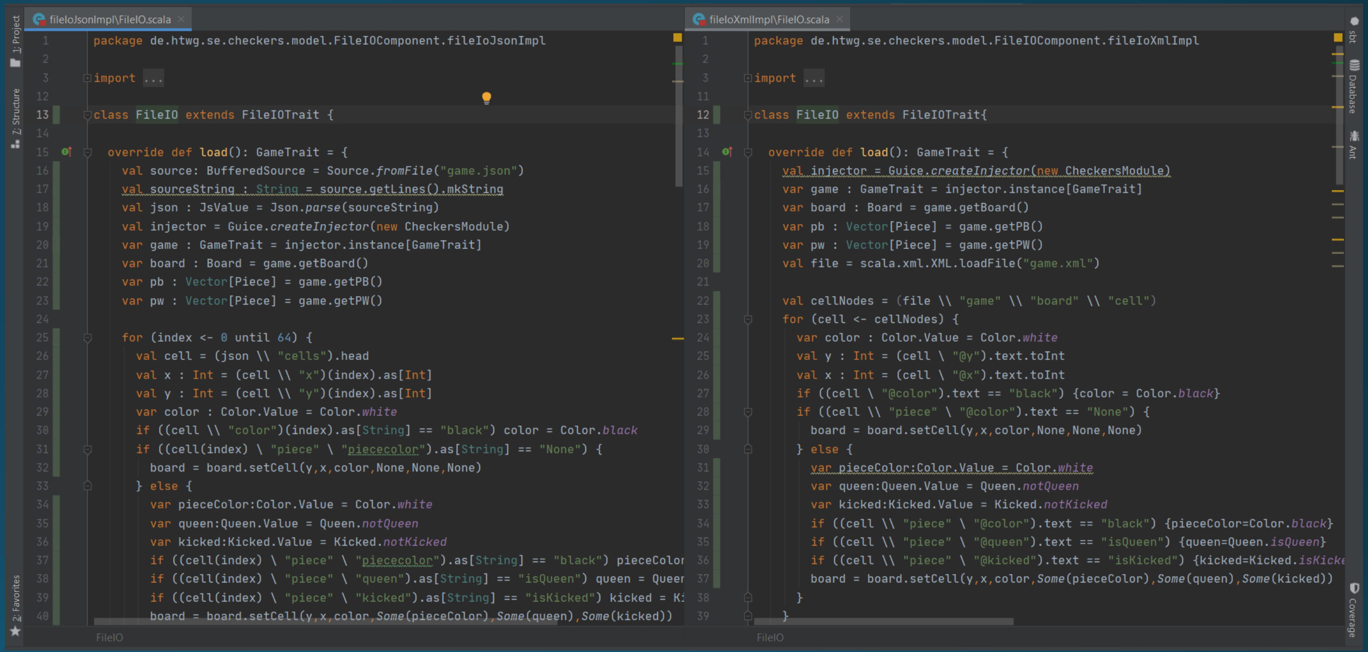


```json
{
  "game" : {
    "board" : {
      "cells" : [ {
        "y" : 0,
        "x" : 0,
        "color" : "black",
        "piece" : {
          "piececolor" : "black",
          "queen" : "notQueen",
          "kicked" : "notKicked"
        }
      }, {
        "y" : 1,
        "x" : 0,
        "color" : "white",
        "piece" : {
          "piececolor" : "None"
        }
      }, {
        "y" : 2,
        "x" : 0,
        "color" : "black",
        "piece" : {
          "piececolor" : "black",
          "queen" : "notQueen",
          "kicked" : "notKicked"
        }
      }, {
        "y" : 3,
        "x" : 0,
        "color" : "white",
```

Dependency Injection



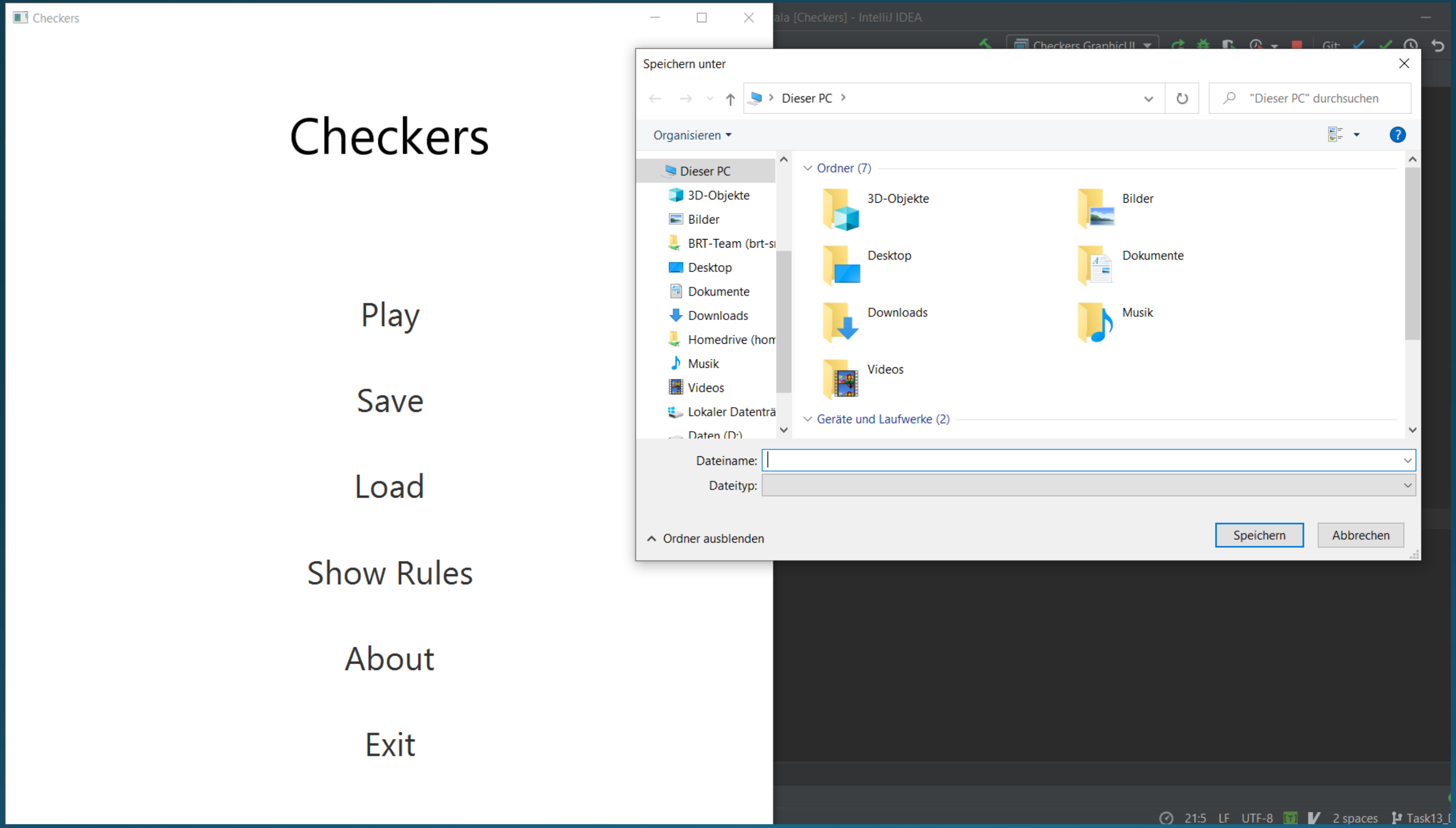
```scala
package de.htwg.se.checkers

import com.google.inject.AbstractModule
import de.htwg.se.checkers.control.ControllerComponent.ControllerTrait
import de.htwg.se.checkers.control.ControllerComponent.controllerBaseImpl
import de.htwg.se.checkers.model.FileIOComponent.FileIOTrait
import net.codingwell.scalaguice.ScalaModule
import de.htwg.se.checkers.model.GameComponent.GameBaseImpl
import de.htwg.se.checkers.model.GameComponent.GameTrait
import de.htwg.se.checkers.model.FileIOComponent.fileIoJsonImpl
import de.htwg.se.checkers.model.FileIOComponent.fileIoXmlImpl
import de.htwg.se.checkers.model.FileIOComponent.fileIoJsonFileChooserImpl

class CheckersModule extends AbstractModule with ScalaModule {

  override def configure() {
    bind[GameTrait].to[GameBaseImpl.Game]
    bind[ControllerTrait].to[controllerBaseImpl.Controller]
    //bind[FileIOTrait].to[fileIoXmlImpl.FileIO]
    bind[FileIOTrait].to[fileIoJsonImpl.FileIO]
    //bind[FileIOTrait].to[fileIoJsonFileChooserImpl.FileIO]
  }
}
```



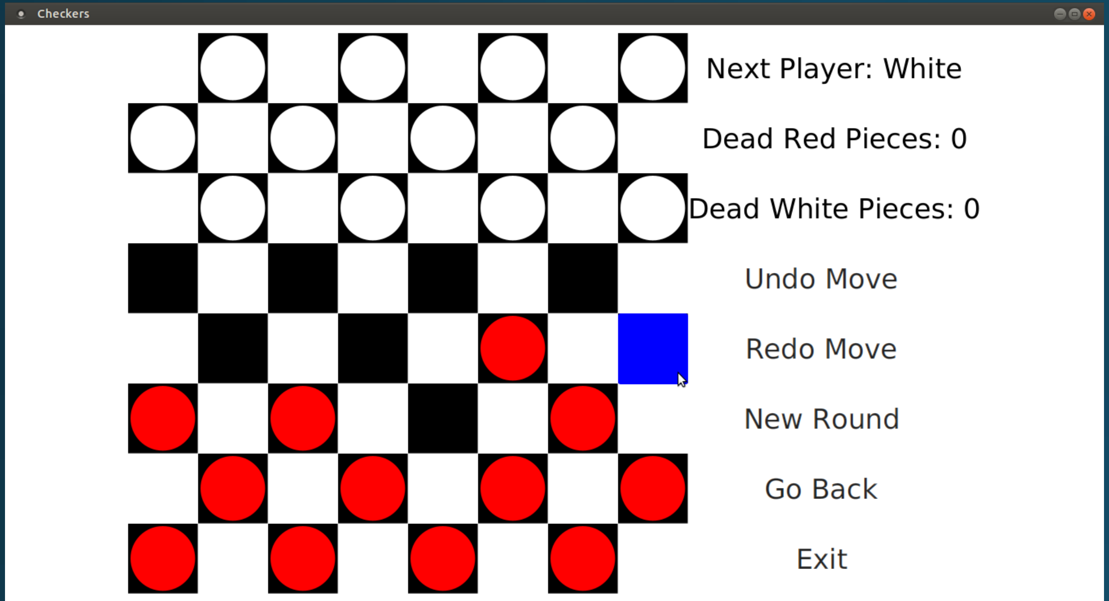
```scala
package de.htwg.se.checkers.model.FileIOComponent.fileIoJsonImpl

import ...

class FileIO extends FileIOTrait {

  override def load(): GameTrait = {
    val source: BufferedSource = Source.fromFile("game.json")
    val sourceString : String = source.getLines().mkString
    val json : JsValue = Json.parse(sourceString)
    val injector = Guice.createInjector(new CheckersModule)
    var game : GameTrait = injector.instance[GameTrait]
    var board : Board = game.getBoard()
    var pb : Vector[Piece] = game.getPB()
    var pw : Vector[Piece] = game.getPW()

    for (index <- 0 until 64) {
      val cell = (json \\ "cells").head
      val x : Int = (cell \\ "x")(index).as[Int]
      val y : Int = (cell \\ "y")(index).as[Int]
      var color : Color.Value = Color.white
      if ((cell \\ "color")(index).as[String] == "black") color = Color.black
      if ((cell(index) \ "piece" \ "piececolor").as[String] == "None") {
        board = board.setCell(y,x,color,None,None,None)
      } else {
        var pieceColor:Color.Value = Color.white
        var queen:Queen.Value = Queen.notQueen
        var kicked:Kicked.Value = Kicked.notKicked
        if ((cell(index) \ "piece" \ "piececolor").as[String] == "black") pieceColor
        if ((cell(index) \ "piece" \ "queen").as[String] == "isQueen") queen = Queer
        if ((cell(index) \ "piece" \ "kicked").as[String] == "isKicked") kicked = Ki
        board = board.setCell(y,x,color,Some(pieceColor),Some(queen),Some(kicked))
```

```scala
package de.htwg.se.checkers.model.FileIOComponent.fileIoXmlImpl

import ...

class FileIO extends FileIOTrait{

  override def load(): GameTrait = {
    val injector = Guice.createInjector(new CheckersModule)
    var game : GameTrait = injector.instance[GameTrait]
    var board : Board = game.getBoard()
    var pb : Vector[Piece] = game.getPB()
    var pw : Vector[Piece] = game.getPW()
    val file = scala.xml.XML.loadFile("game.xml")

    val cellNodes = (file \\ "game" \\ "board" \\ "cell")
    for (cell <- cellNodes) {
      var color : Color.Value = Color.white
      val y : Int = (cell \ "@y").text.toInt
      val x : Int = (cell \ "@x").text.toInt
      if ((cell \ "@color").text == "black") {color = Color.black}
      if ((cell \\ "piece" \ "@color").text == "None") {
        board = board.setCell(y,x,color,None,None,None)
      } else {
        var pieceColor:Color.Value = Color.white
        var queen:Queen.Value = Queen.notQueen
        var kicked:Kicked.Value = Kicked.notKicked
        if ((cell \\ "piece" \ "@color").text == "black") {pieceColor=Color.black}
        if ((cell \\ "piece" \ "@queen").text == "isQueen") {queen=Queen.isQueen}
        if ((cell \\ "piece" \ "@kicked").text == "isKicked") {kicked=Kicked.isKicke
        board = board.setCell(y,x,color,Some(pieceColor),Some(queen),Some(kicked))
      }
    }
```

FileIO with FileChooser



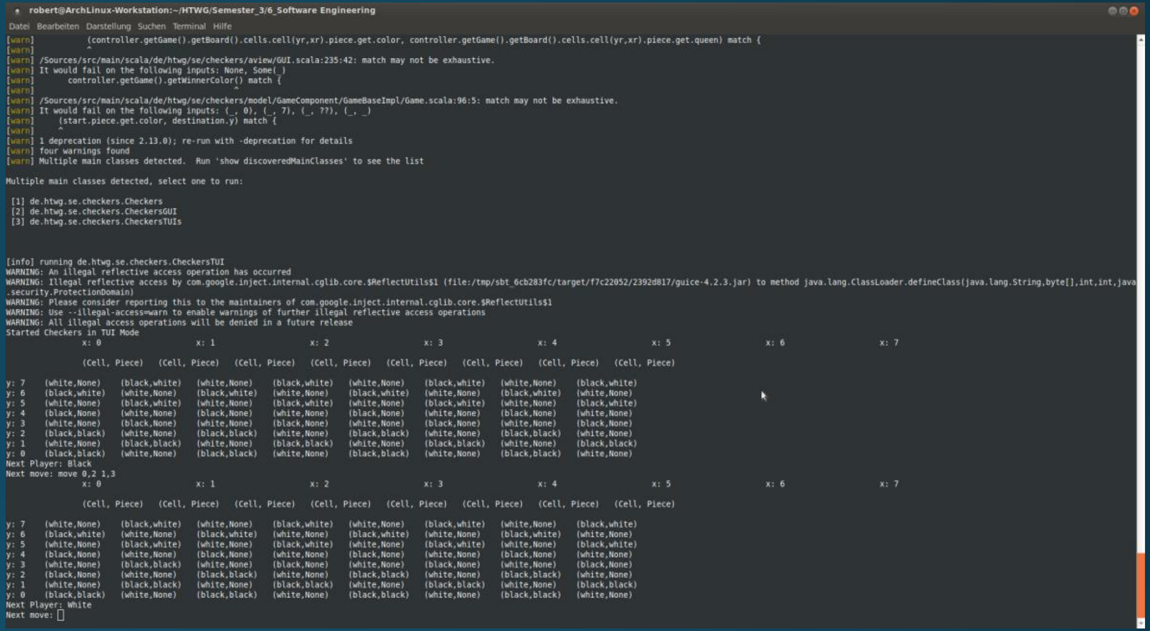
Docker



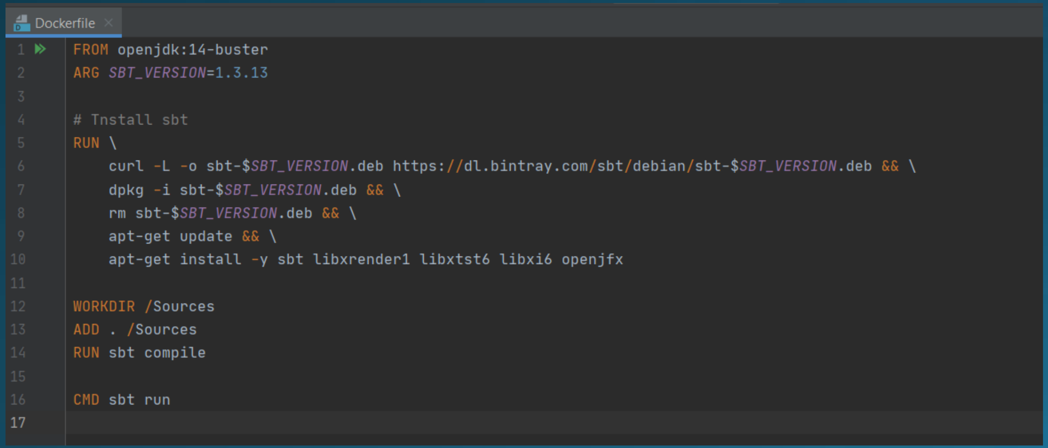
Next Player: White

Dead Red Pieces: 0

Dead White Pieces: 0


robert@ArchLinux-Workstation:~/HTWG/Semester_3/6_Software Engineering
Datei
Bearbeiten
Darstellung
Suchen
Terminal
Hilfe

```
[warn]          (controller.getGame().getBoard().cells.cell(yr,xr).piece.get.color, controller.getGame().getBoard().cells.cell(yr,xr).piece.get.queen) match {
[warn]          ^
[warn] /Sources/src/main/scala/de/htwg/se/checkers/aview/GUI.scala:235:42: match may not be exhaustive.
[warn] It would fail on the following inputs: None, Some(_)
[warn]      controller.getGame().getWinnerColor() match {
[warn]                                        ^
[warn] /Sources/src/main/scala/de/htwg/se/checkers/model/GameComponent/GameBaseImpl/Game.scala:96:5: match may not be exhaustive.
[warn] It would fail on the following inputs: (_, 0), (_, 7), (_, ??), (_, _)
[warn]    (start.piece.get.color, destination.y) match {
[warn]    ^
[warn] 1 deprecation (since 2.13.0); re-run with -deprecation for details
[warn] four warnings found
[warn] Multiple main classes detected.  Run 'show discoveredMainClasses' to see the list

Multiple main classes detected, select one to run:

 [1] de.htwg.se.checkers.Checkers
 [2] de.htwg.se.checkers.CheckersGUI
 [3] de.htwg.se.checkers.CheckersTUIs



[info] running de.htwg.se.checkers.CheckersTUI
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by com.google.inject.internal.cglib.core.$ReflectUtils$1 (file:/tmp/sbt_6cb283fc/target/f7c22052/2392d817/guice-4.2.3.jar) to method java.lang.ClassLoader.defineClass(java.lang.String,byte[],int,int,java
.security.ProtectionDomain)
WARNING: Please consider reporting this to the maintainers of com.google.inject.internal.cglib.core.$ReflectUtils$1
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Started Checkers in TUI Mode
                x: 0                    x: 1                     x: 2                    x: 3                     x: 4                    x: 5                     x: 6                    x: 7

                (Cell, Piece)   (Cell, Piece)   (Cell, Piece)   (Cell, Piece)   (Cell, Piece)   (Cell, Piece)   (Cell, Piece)   (Cell, Piece)

y: 7    (white,None)    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)
y: 6    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)   (white,None)
y: 5    (white,None)    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)
y: 4    (black,None)    (white,None)    (black,None)    (white,None)    (black,None)    (white,None)    (black,None)    (white,None)
y: 3    (white,None)    (black,None)    (white,None)    (black,None)    (white,None)    (black,None)    (white,None)    (black,None)
y: 2    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)   (white,None)
y: 1    (white,None)    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)
y: 0    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)   (white,None)
Next Player: Black
Next move: move 0,2 1,3
                x: 0                    x: 1                     x: 2                    x: 3                     x: 4                    x: 5                     x: 6                    x: 7

                (Cell, Piece)   (Cell, Piece)   (Cell, Piece)   (Cell, Piece)   (Cell, Piece)   (Cell, Piece)   (Cell, Piece)   (Cell, Piece)

y: 7    (white,None)    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)
y: 6    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)   (white,None)
y: 5    (white,None)    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)   (white,None)    (black,white)
y: 4    (black,None)    (white,None)    (black,None)    (white,None)    (black,None)    (white,None)    (black,None)    (white,None)
y: 3    (white,None)    (black,black)   (white,None)    (black,None)    (white,None)    (black,None)    (white,None)    (black,None)
y: 2    (black,None)    (white,None)    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)   (white,None)
y: 1    (white,None)    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)
y: 0    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)   (white,None)    (black,black)   (white,None)
Next Player: White
Next move:
```



```
FROM openjdk:14-buster
ARG SBT_VERSION=1.3.13

# Tnstall sbt
RUN \
    curl -L -o sbt-$SBT_VERSION.deb https://dl.bintray.com/sbt/debian/sbt-$SBT_VERSION.deb && \
    dpkg -i sbt-$SBT_VERSION.deb && \
    rm sbt-$SBT_VERSION.deb && \
    apt-get update && \
    apt-get install -y sbt libxrender1 libxtst6 libxi6 openjfx

WORKDIR /Sources
ADD . /Sources
RUN sbt compile

CMD sbt run

```

Danke für‘s Zuhören