Hochschule Konstanz
Fakultät Informatik

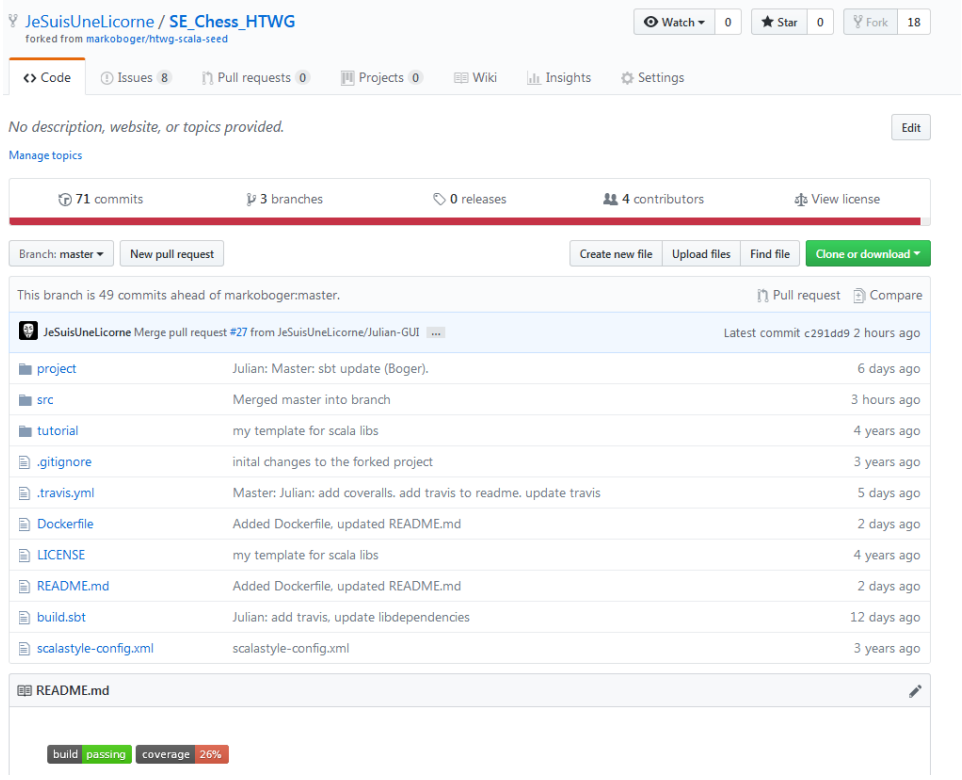
Software Engineering: Chess

Jörg Stenger – Julian Zimmermann

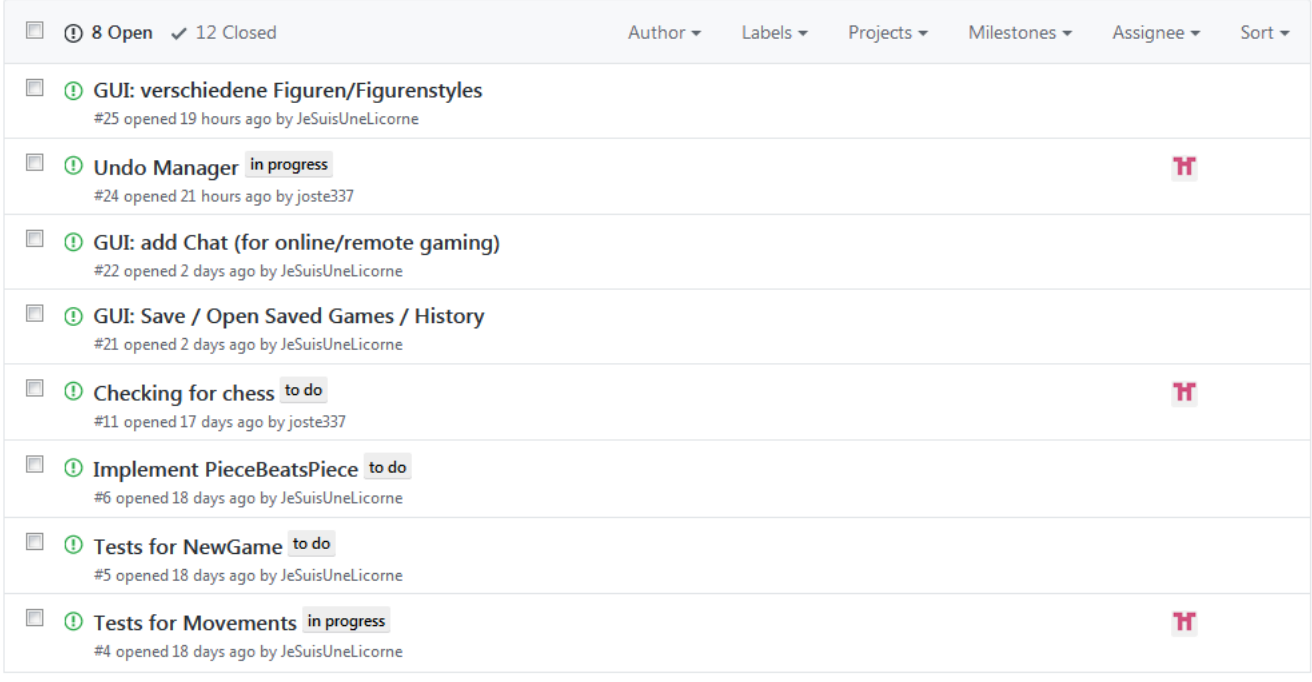
Gliederung

- Version Control Systems - Git
- Agile Development
- Scala
- MVC-Architecture
- Continuous Deployment
- Design Pattern
- TUI
- GUI
- Components
- Dependency Injection
- File IO
- Docker
- Fazit

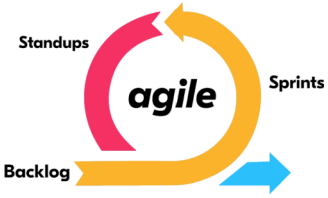
Version Control Systems - Git

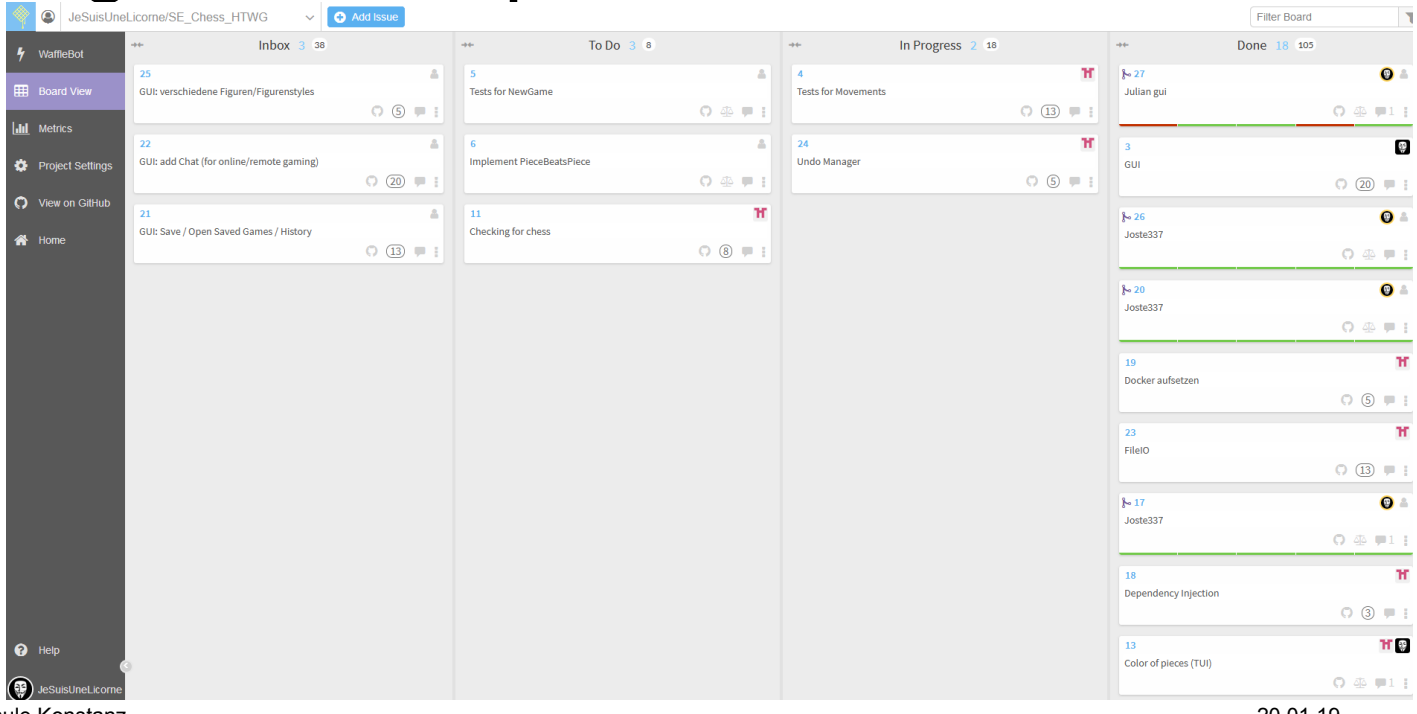


Version Control Systems - Git



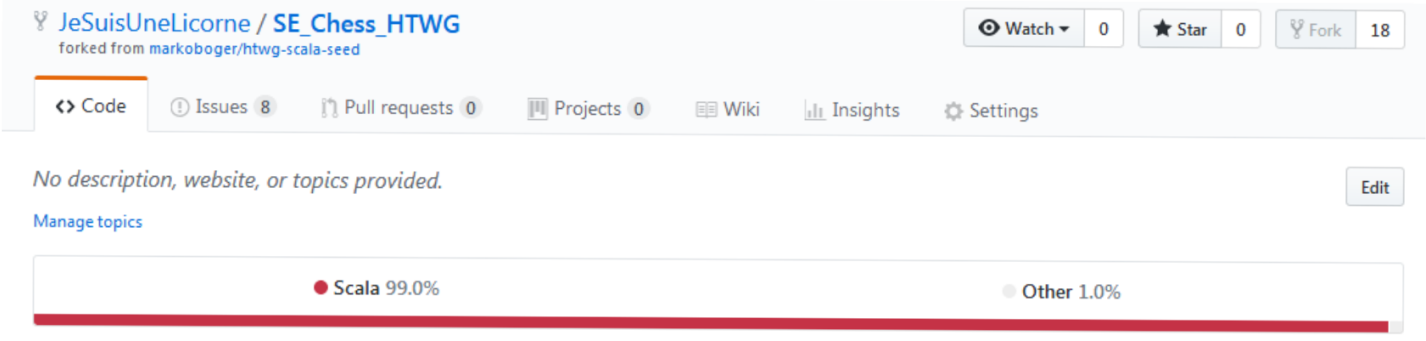
Agile Development





Scala


Scala


JeSuisUneLicorne / SE_Chess_HTWG
forked from markoboger/htwg-scala-seed
Watch 0
Star 0
Fork 18
Code
Issues 8
Pull requests 0
Projects 0
Wiki
Insights
Settings
No description, website, or topics provided.
Edit
Manage topics
Scala 99.0%
Other 1.0%

Scala

```scala
package de.htwg.se.SE_Chess_HTWG.model.pieceComponent

import de.htwg.se.SE_Chess_HTWG.model.gridComponent.{Cell, GridInterface}
import de.htwg.se.SE_Chess_HTWG.model.movement.{Move, MovementHelper}
import de.htwg.se.SE_Chess_HTWG.util.MovementResult
import de.htwg.se.SE_Chess_HTWG.util.MovementResult.MovementResult

private[pieceComponent] case class King(val isWhite: Boolean, var row: Int, var col: Int, var hasMoved: Boolean = false) extends PieceInterface {
  override def toString: String = if (isWhite) "\u2654" else "\u265A"
  override def toSimpleString: String = "K"
  val imageName = if (isWhite) "king_w" else "king_b"

  def executeMove(grid: GridInterface, move: Move): MovementResult = {
    if (getPossibleSquares(grid) contains move.getToCell) move.doMove() else MovementResult.ERROR
  }

  def getPossibleSquares(grid: GridInterface): List[Cell] = {
    val possibleSquares: List[(Int, Int)] = List((row + 1, col + 1), (row + 1, col - 1), (row + 1, col),
      (row - 1, col - 1), (row - 1, col + 1), (row - 1, col), (row, col + 1), (row, col - 1))
    MovementHelper.getSquaresInGrid(grid, possibleSquares, isWhite)
  }
}

```

Continuous Deployment

Travis CI

JeSuisUneLicorne / SE_Chess_HTWG build passing

Current Branches Build History Pull Requests More options

master Merge pull request #27 from JeSuisUneLicorne/Julian-GUI

Julian gui

Commit c291dd9

Compare 85c94da..c291dd9

Branch master

Julian

Scala: 2.12.8

#45 passed

Ran for 2 min 34 sec

about 2 hours ago

Restart build

Job log View config

Remove log Raw log

```
1  Worker information                                                                                    worker_info
6  Build system information                                                                              system_info
413
414 Updating sbt
415
416 $ git clone --depth=50 --branch=master https://github.com/JeSuisUneLicorne/SE_Chess_HTWG.git JeSuisUneLicorne/SE_Chess_HTWG   git.checkout  0.60s
426
427 $ export JVM_OPTS=@/etc/sbt/jvmopts
```

Continuous Deployment

JeSuisUneLicorne / SE_Chess_HTWG build passing

Current Branches Build History Pull Requests More options

Default Branch

✓ master 9 builds	# 45 passed about 2 hours ago	c291dd9 Julian	✓ #45 ✓ ✓ ✗ ✓

Active Branches

✓ Julian-GUI 12 builds	# 43 passed about 3 hours ago	9344f93 joste337	✓ ✓ ✗ ✓ ✗
✓ joste337 15 builds	# 40 passed about 4 hours ago	10830ab joste337	✓ ✗ ✗ ✗ ✓

Inactive Branches

Design Pattern

- Observer Pattern (MVC)
- Command Pattern (UndoManager)
- Factory Pattern (DI, PieceFactory)

Design Pattern

```
5  trait PieceFactory {
6    def getPiece(pieceSimpleString: Piece, isWhite: Boolean, row: Int, col: Int, hasMoved: Boolean = false): PieceInterface
7  }
8
9  class PieceFactoryImpl extends PieceFactory {
10   def getPiece(pieceSimpleString: Piece, isWhite: Boolean, row: Int, col: Int, hasMoved: Boolean = false): PieceInterface = {
11     pieceSimpleString match {
12       case Piece.PAWN => new Pawn(isWhite, row, col, hasMoved)
13       case Piece.ROOK => new Rook(isWhite, row, col, hasMoved)
14       case Piece.KNIGHT => new Knight(isWhite, row, col, hasMoved)
15       case Piece.BISHOP => new Bishop(isWhite, row, col, hasMoved)
16       case Piece.QUEEN => new Queen(isWhite, row, col, hasMoved)
17       case Piece.KING => new King(isWhite, row, col, hasMoved)
18     }
19   }
20 }
```

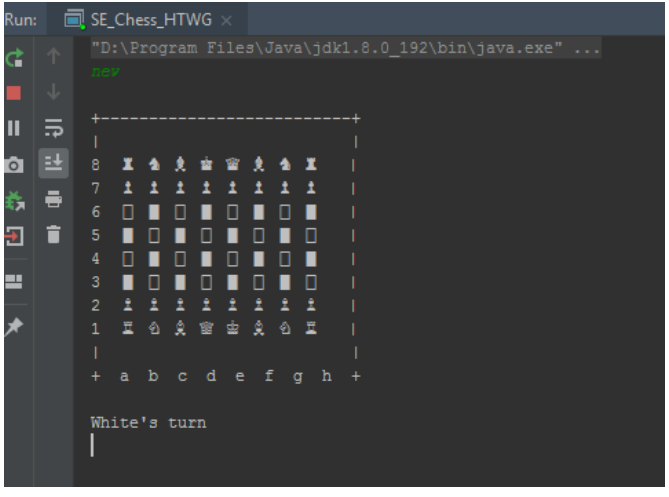
Design Pattern

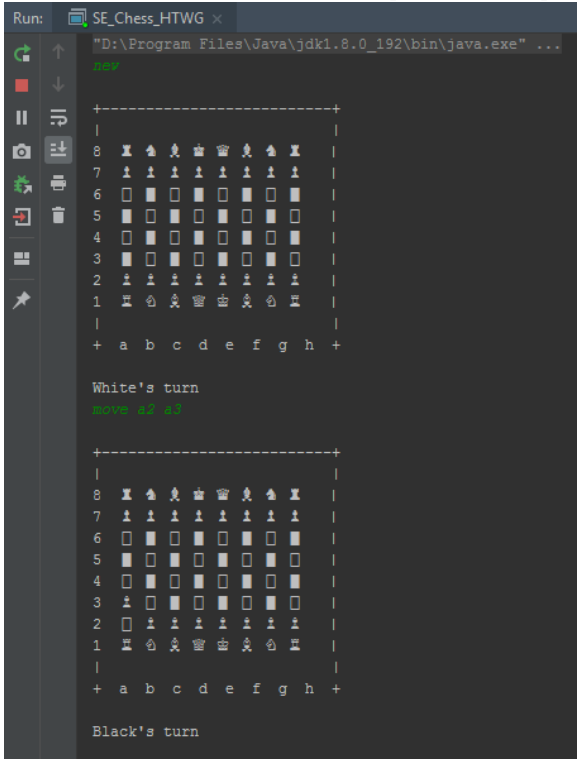
```
42    object Piece extends Enumeration {
43      type Piece = Value
44      val PAWN, ROOK, KNIGHT, BISHOP, QUEEN, KING = Value
45    }
```

```
private[pieceComponent] class Pawn(val isWhite: Boolean, var row: Int, var col: Int, var hasMoved: Boolean = false) extends PieceInterface {
  override def toString: String = if (isWhite) "\u265F" else "\u2659"
  override def toSimpleString: String = "P"
```

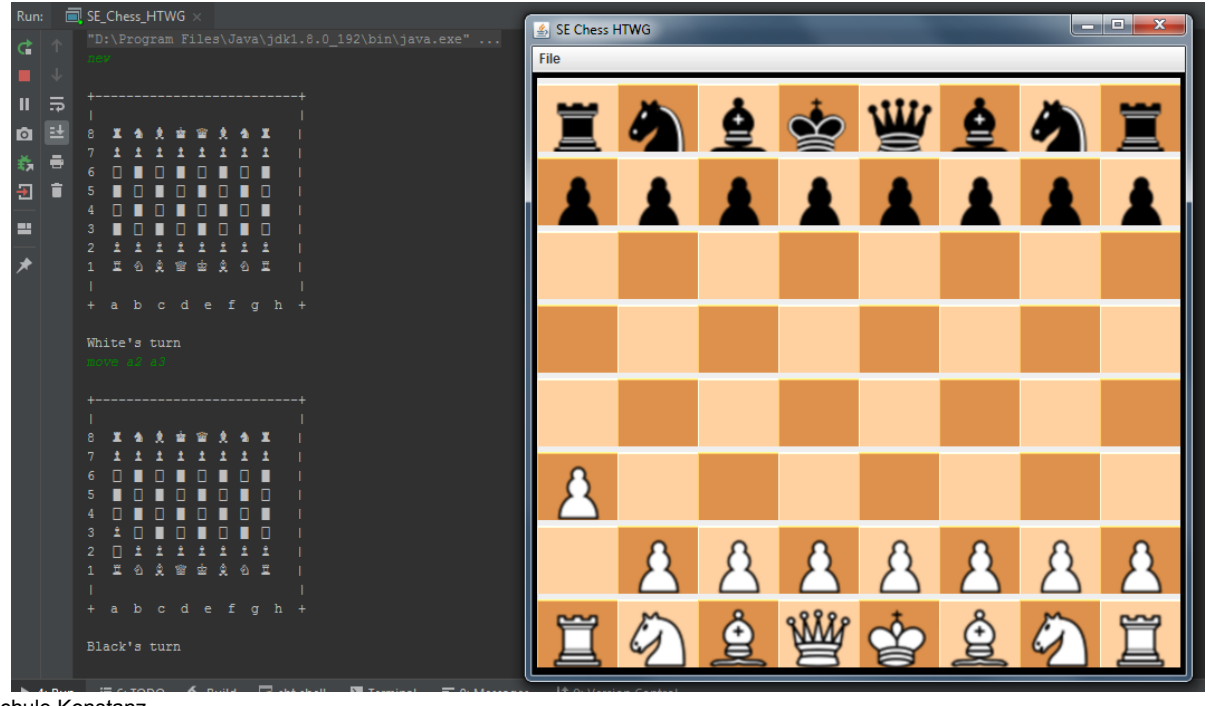
```
class ChessModule extends AbstractModule with ScalaModule{

  def configure() : Unit = {
    bind[GridInterface].to[GridImpl]
    bind[ControllerInterface].to[ControllerImpl]
    bind[FileIOInterface].to[FileIOXmlImpl]
    bind[PieceFactory].to[PieceFactoryImpl]
  }

}
```

Text-based User Interface (TUI)

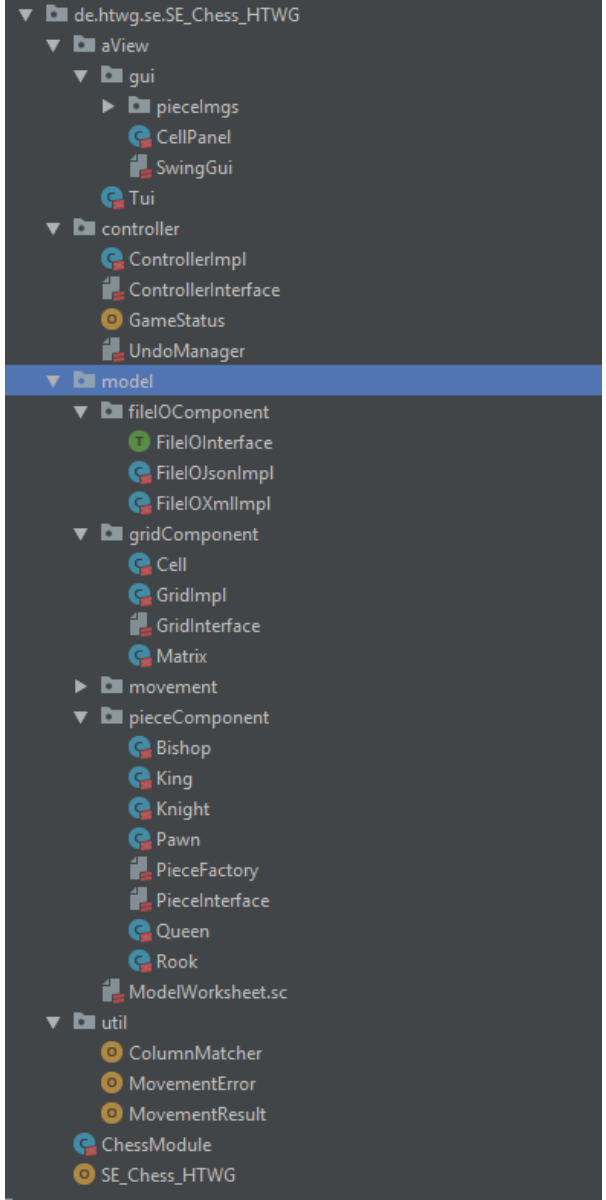




Graphical User Interface - GUI



Components


de.htwg.se.SE_Chess_HTWG
aView
gui
pieceImgs
CellPanel
SwingGui
Tui
controller
ControllerImpl
ControllerInterface
GameStatus
UndoManager
model
fileIOComponent
FileIOInterface
FileIOJsonImpl
FileIOXmlImpl
gridComponent
Cell
GridImpl
GridInterface
Matrix
movement
pieceComponent
Bishop
King
Knight
Pawn
PieceFactory
PieceInterface
Queen
Rook
ModelWorksheet.sc
util
ColumnMatcher
MovementError
MovementResult
ChessModule
SE_Chess_HTWG

Dependency Injection

```
class ChessModule extends AbstractModule with ScalaModule{

  def configure() : Unit = {
    bind[GridInterface].to[GridImpl]
    bind[ControllerInterface].to[ControllerImpl]
    bind[FileIOInterface].to[FileIOXmlImpl]
    bind[PieceFactory].to[PieceFactoryImpl]
  }

}
```

```
class ControllerImpl @Inject() (var grid: GridInterface) extends ControllerInterface with Publisher {

  val injector = Guice.createInjector(new ChessModule)
  val fileIo: FileIOInterface = injector.getInstance(classOf[FileIOInterface])
  val undoManager: UndoManager = new UndoManagerImpl(grid)
  var gameStatus: GameStatus = IDLE
```

File I/O

```
def gridToJson(grid: GridInterface, gameStatus: GameStatus): JsObject = {
  Json.obj(
    fields = "game" -> Json.obj(
      fields = "status" -> GameStatus.toOutputString(gameStatus),
      "pieces" -> grid.getSetCells.map(cell => Json.toJson(cell.value.get))
    )
  )
}
```

```
object PieceInterface extends Enumeration {
  implicit val pieceWrites = new Writes[PieceInterface] {
    def writes(piece: PieceInterface) :JsObject = Json.obj(
      fields = "row" -> piece.row,
      "col" -> piece.col,
      "value" -> piece.toSimpleString,
      "isWhite" -> piece.isWhite,
      "hasMoved" -> piece.hasMoved
    )
  }
```

File I/O

```
<game status="pl">
    <piece row="7" col="7" isWhite="false" hasMoved="false"> R </piece>
    <piece row="7" col="6" isWhite="false" hasMoved="false"> N </piece>
    <piece row="7" col="5" isWhite="false" hasMoved="false"> B </piece>
    <piece row="7" col="4" isWhite="false" hasMoved="false"> Q </piece>
    <piece row="7" col="3" isWhite="false" hasMoved="false"> K </piece>
    <piece row="7" col="2" isWhite="false" hasMoved="false"> B </piece>
    <piece row="7" col="1" isWhite="false" hasMoved="false"> N </piece>
    <piece row="7" col="0" isWhite="false" hasMoved="false"> R </piece>
    <piece row="6" col="7" isWhite="false" hasMoved="false"> P </piece>
    <piece row="6" col="6" isWhite="false" hasMoved="false"> P </piece>
    <piece row="6" col="5" isWhite="false" hasMoved="false"> P </piece>
    <piece row="6" col="4" isWhite="false" hasMoved="false"> P </piece>
    <piece row="6" col="3" isWhite="false" hasMoved="false"> P </piece>
    <piece row="6" col="2" isWhite="false" hasMoved="false"> P </piece>
    <piece row="6" col="1" isWhite="false" hasMoved="false"> P </piece>
    <piece row="6" col="0" isWhite="false" hasMoved="false"> P </piece>
    <piece row="1" col="7" isWhite="true" hasMoved="false"> P </piece>
    <piece row="1" col="6" isWhite="true" hasMoved="false"> P </piece>
    <piece row="1" col="5" isWhite="true" hasMoved="false"> P </piece>
```

```
{
  "game" : {
    "status" : "pl",
    "pieces" : [ {
      "row" : 7,
      "col" : 7,
      "value" : "R",
      "isWhite" : false,
      "hasMoved" : false
    }, {
      "row" : 7,
      "col" : 6,
      "value" : "N",
      "isWhite" : false,
      "hasMoved" : false
    }, {
      "row" : 7,
      "col" : 5,
      "value" : "B",
      "isWhite" : false,
      "hasMoved" : false
    }, {
      "row" : 7,
      "col" : 4,
      "value" : "Q",
      "isWhite" : false,
      "hasMoved" : false
    }, {
```

Docker

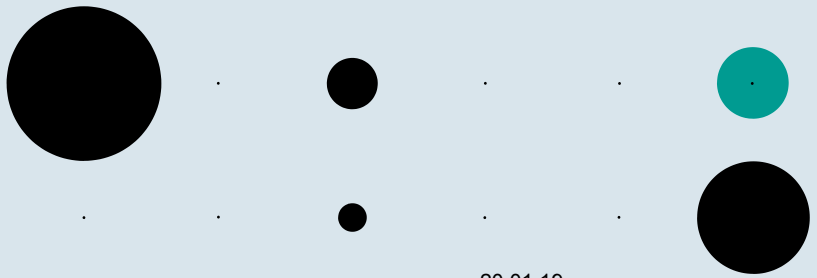
```
FROM hseeberger/scala-sbt
WORKDIR /chess
ADD . /chess
CMD sbt run
```

- Docker build -t chess .
- Docker run -p 4000:80 chess

Docker


```
[info] downloading https://repo1.maven.org/maven2/junit/junit/4.8/junit-4.8.jar ...
[info]  [SUCCESSFUL ] junit#junit;4.8!junit.jar (725ms)
[info] downloading https://repo1.maven.org/maven2/org/scalactic/scalactic_2.12/3.0.4/scalactic_2.12-3.0.4.jar ...
[info]  [SUCCESSFUL ] com.google.guava#guava;19.0!guava.jar(bundle) (7538ms)
[info] downloading https://repo1.maven.org/maven2/org/scala-lang/modules/scala-parser-combinators_2.12/1.0.4/scala-parser-combinators_2.12-1.0.4.jar .
..
[info]  [SUCCESSFUL ] org.scala-lang.modules#scala-parser-combinators_2.12;1.0.4!scala-parser-combinators_2.12.jar(bundle) (1484ms)
[info] downloading https://repo1.maven.org/maven2/org/scala-lang/scala-compiler/2.12.8/scala-compiler-2.12.8.jar ...
[info]  [SUCCESSFUL ] org.scalactic#scalactic_2.12;3.0.4!scalactic_2.12.jar(bundle) (2725ms)
[info]  [SUCCESSFUL ] org.scala-lang#scala-library;2.12.8!scala-library.jar (13099ms)
[info]  [SUCCESSFUL ] org.scalafx#scalafx_2.12;8.0.144-R12!scalafx_2.12.jar (13784ms)
[info]  [SUCCESSFUL ] org.scala-lang#scala-reflect;2.12.8!scala-reflect.jar (11926ms)
[info]  [SUCCESSFUL ] org.scalatest#scalatest_2.12;3.0.4!scalatest_2.12.jar(bundle) (10922ms)
[info]  [SUCCESSFUL ] org.scala-lang#scala-compiler;2.12.8!scala-compiler.jar (11762ms)
[info] Done updating.
[warn] There may be incompatibilities among your library dependencies; run 'evicted' to see detailed eviction warnings.
[info] Compiling 28 Scala sources to /chess/target/scala-2.12/classes ...
[info] Non-compiled module 'compiler-bridge_2.12' for Scala 2.12.8. Compiling...
[info]   Compilation completed in 25.348s.
[warn] there was one deprecation warning (since 2.11.0); re-run with -deprecation for details
[warn] one warning found
[info] Done compiling.
[info] Packaging /chess/target/scala-2.12/se_chess_htwg_2.12-0.0.1.jar ...
[info] Done packaging.
[info] Running de.htwg.se.SE_Chess_HTWG.SE_Chess_HTWG
```

Noch Fragen?



Danke für eure Aufmerksamkeit