

PROJEKTPRÄSENTATION SOFTWARE ENGINEERING

A close-up, slightly blurred photograph of a Go board. The board is made of light-colored wood with a grid of lines. Several black and white Go stones are scattered across the board, mostly in the center and lower-left areas. The stones are smooth and round. The background is out of focus, showing more stones and the board's texture.

Gruppe 10: Florian Djokaj und
Markus Käppeler

GLIEDERUNG

1. Go Regeln
2. Version Control System
3. Scrum
4. Layers an MVC
5. Code Coverage
6. CI
7. Design Patterns
8. Components and Interfaces
9. Dependency Injection
10. FILE IO in JSON and XML
11. Documentation
12. TUI
13. GUI

GO RULES

- 2 Spieler, rundenbasiert, abwechselnd
- Schlagen durch Umzingeln
- Selbstmordverbot
- Passen und Spielende
- Tote Steine
- Abrechnung

VERSION CONTROL SYSTEM

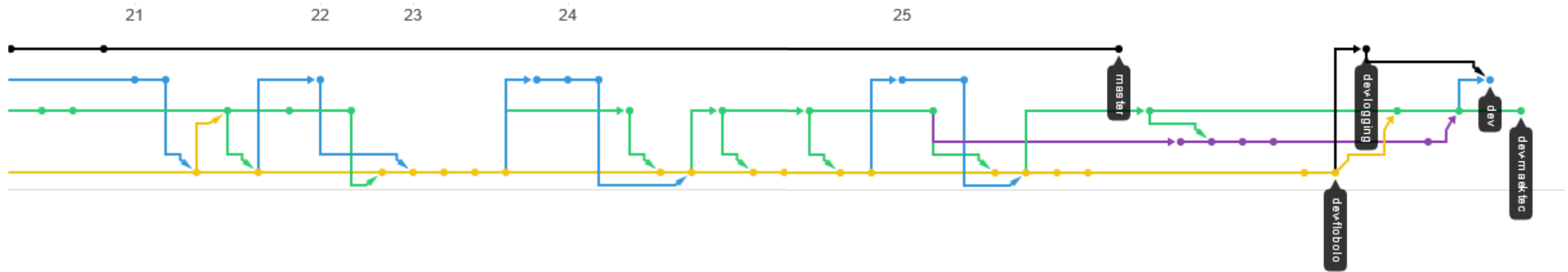
Oct 15, 2017 – Jan 26, 2018

Contributions: Commits ▾

Contributions to master, excluding merge commits



VERSION CONTROL SYSTEM



SCRUM

scrumwise

▼ Markus Nobody else is online Help Settings Log out

Overview Projects People Backlog Releases Burnup Sprints Task board Burndown More

SE2017WS-10-Go

Send feedback

Usefull end of a game1 h Sprint completed

2 Players turn by turn2 h Sprint completed

Sprint 2Resume this sprintCompleted

Team 116 hours completed

Florian5 hMarkus11 h

Implement a design patterns3 h Sprint completed

Scoring8 h Sprint completed

Menu

Play field5 h Sprint completed

Sprint 3Complete this sprintIn progress

Team 11.5 hours behind

Florian3.5 h leftMarkus1 h left

ComponentsDone

Dependency InjectionDone

FILE IO2.5 h left In progress

Logging1 h To do

DocumentationDone

Settings1 h left In progress

Bug fixes1 h left In progress

Backlog items that are not assigned to any sprint

Game rules

Special optional rules8 h left In progress

Game rulesEpic

Auswertung des SpielstandsNew

Add a backlog itemShowing 1 of 2 items

Check for Clean Code3 hours

Implement a design patternsEpic

Implement a design pattern4 hours

Standard rules2 h left In progress

Add a backlog itemShowing 2 of 3 items

GUIEpic

MenuEpic

Add a backlog itemShowing 0 of 2 items

Add a backlog itemShowing 1 of 2 items

GUI

Basic play field4 hours

Menu

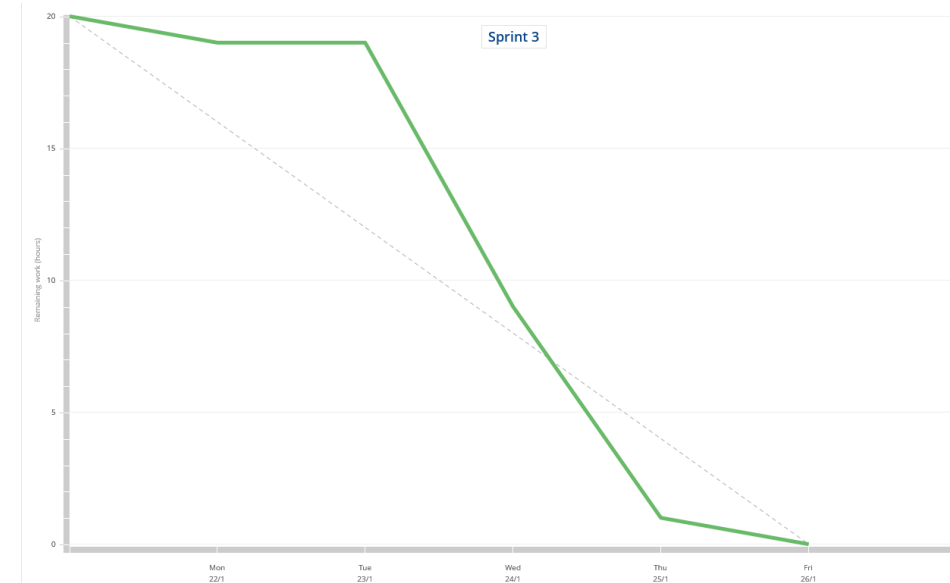
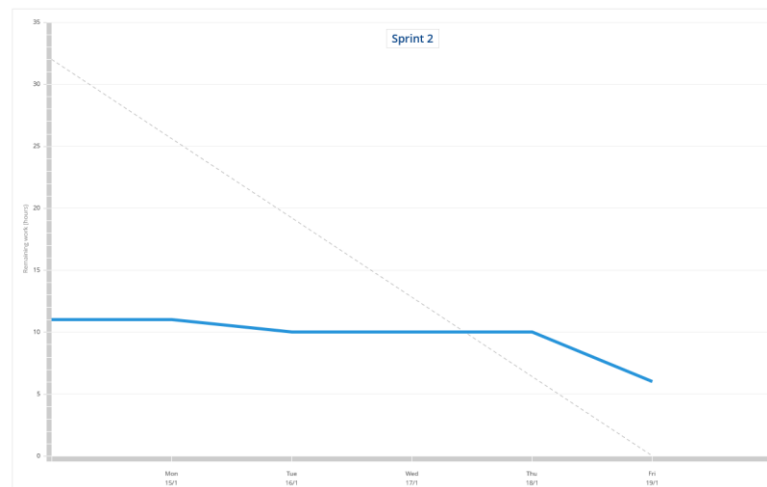
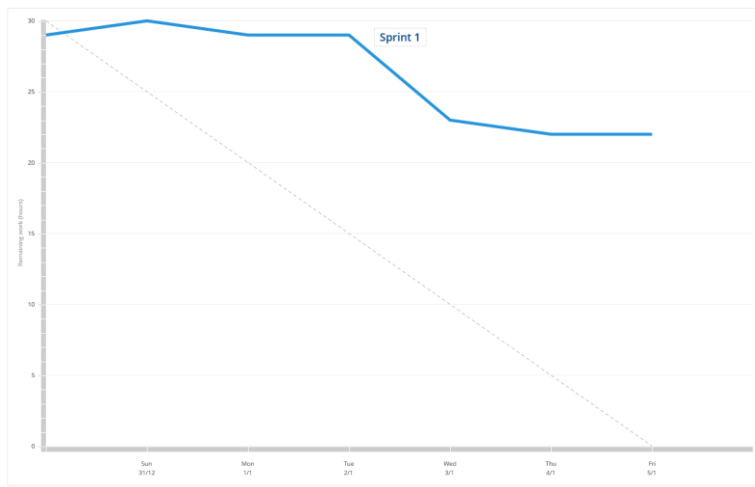
Startup menu1 hour

+ Add a sprintShowAll sprints

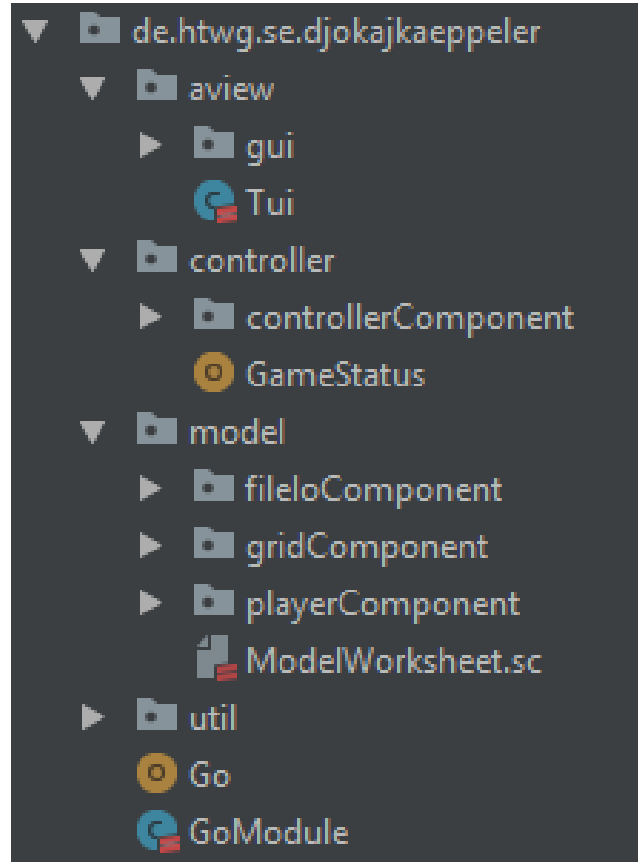
Q

+ Add a backlog item

BURNDOWNCHARTS



LAYERS AND MVC



CODE COVERAGE

COVERAGE	FILE	LINES	RELEVANT	COVERED	MISSED	HITS/LINE
0.0	...a/de/htwg/se/djokajkaepeler/aview/gui/Board.scala	91	46	0	46	0.0
0.0	...e/htwg/se/djokajkaepeler/aview/gui/SwingGui.scala	126	63	0	63	0.0
0.0	src/main/scala/de/htwg/se/djokajkaepeler/Go.scala	41	17	0	17	0.0
0.0	...a/de/htwg/se/djokajkaepeler/util/Observable.scala	15	4	0	4	0.0
0.0	...ollerComponent/controllerMockImpl/Controller.scala	55	21	0	21	0.0
0.0	...ppeler/model/gridComponent/gridMockImpl/Grid.scala	49	20	0	20	0.0
0.0	...r/model/fileIoComponent/fileIoXmlImpl/FileIO.scala	109	49	0	49	0.0
0.0	.../model/playerComponent/playerMockImpl/Player.scala	10	2	0	2	0.0
72.0	...n/scala/de/htwg/se/djokajkaepeler/aview/Tui.scala	67	25	18	7	1.0
85.71	.../de/htwg/se/djokajkaepeler/util/UndoManager.scala	30	14	12	2	1.0
87.93	...ollerComponent/controllerBaseImpl/Controller.scala	139	58	51 +1	7 -1	1.0
90.7	.../model/fileIoComponent/fileIoJsonImpl/FileIO.scala	103	43	39	4	1.0
95.83	...llerComponent/controllerBaseImpl/SkipCommand.scala	54	24	23	1	1.0
96.15	...llerComponent/controllerBaseImpl/TurnCommand.scala	57	26	25	1	1.0
100.0	...ppeler/model/gridComponent/gridBaseImpl/Grid.scala	153	67	67	0	1.0
100.0	.../gridBaseImpl/GridEvaluationStrategyTemplate.scala	56	24	24	0	1.0
100.0	...t/gridBaseImpl/GridEvaluationChineseStrategy.scala	22	10	10	0	1.0
100.0	...eler/model/gridComponent/gridBaseImpl/Matrix.scala	9	5	5	0	1.0
100.0	...ppeler/model/gridComponent/gridBaseImpl/Cell.scala	96	55	55 +7	0 -7	1.0
100.0	.../model/playerComponent/playerBaseImpl/Player.scala	10	1	1	0	1.0
100.0	...ollerComponent/controllerBaseImpl/SetCommand.scala	24	6	6	0	1.0
100.0	...twg/se/djokajkaepeler/controller/GameStatus.scala	39	19	19	0	1.0
100.0	...in/scala/de/htwg/se/djokajkaepeler/GoModule.scala	29	8	8	0	1.0

CONTINUOUS INTEGRATION

MaekTec / GoGame  build passing

Current	Branches	Build History	Pull Requests	Requests	More
✓	🔗 8069664	📅 10 minutes ago	Merge pull request #27 from MaekTec/dev	# 118	Build created successfully
✓	🔗 #27 43e8044	📅 16 minutes ago	Merge pull request #26 from MaekTec/dev-maektec	# 117	Build created successfully
✓	🔗 5a1b8a8	📅 17 minutes ago	Merge pull request #26 from MaekTec/dev-maektec	# 116	Build created successfully
✓	🔗 #26 0db59fd	📅 18 minutes ago	added some tests	# 115	Build created successfully
✓	🔗 53220d5	📅 21 minutes ago	added some tests	# 114	Build created successfully
✓	🔗 61c3702	📅 24 minutes ago	Merge pull request #25 from MaekTec/dev-flobolo	# 113	Build created successfully
✓	🔗 #25 11942cc	📅 30 minutes ago	Test for Controller and GameStatus added	# 112	Build created successfully
✓	🔗 #24 deb1141	📅 37 minutes ago	Test for Controller and GameStatus added	# 111	Build created successfully
✓	🔗 #23 c99e7c3	📅 40 minutes ago	Test for Controller and GameStatus added	# 110	Build created successfully
✓	🔗 59bffab	📅 40 minutes ago	Test for Controller and GameStatus added	# 109	Build created successfully
✓	🔗 #23 79e65e0	📅 about an hour ago	added test files	# 108	Build created successfully
✓	🔗 3995444	📅 about an hour ago	added test files	# 107	Build created successfully
✓	🔗 #23 f47ddb1	📅 about an hour ago	Test for Controller and GameStatus added	# 106	Build created successfully
✓	🔗 add3989	📅 about an hour ago	Test for Controller and GameStatus added	# 105	Build created successfully
✓	🔗 8d30bc2	📅 about 2 hours ago	xml save	# 104	Build created successfully
✓	🔗 ec6c7c6	📅 about 2 hours ago	fixed player 2 wins always bug and added tests	# 103	Build created successfully
✓	🔗 7420b3c	📅 about 3 hours ago	Merge pull request #22 from MaekTec/dev-logging	# 102	Build created successfully
✓	🔗 #22 4b55c26	📅 about 3 hours ago	logging	# 101	Build created successfully

DESIGN PATTERNS

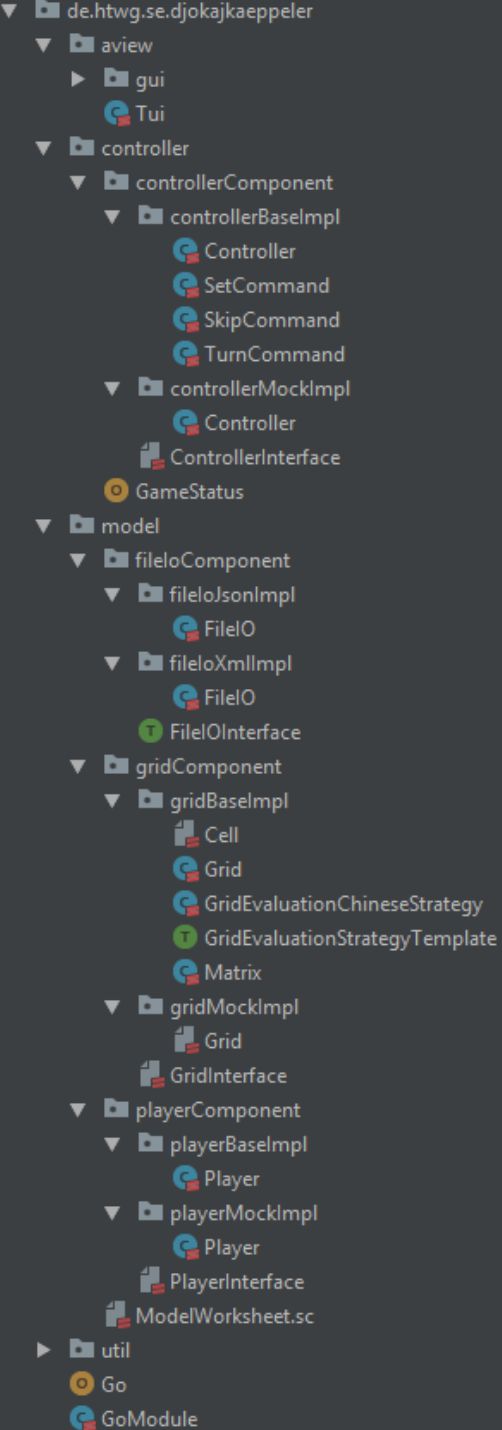
Observer Pattern (for MVC)

Command Pattern (do, undo and redo)

Strategy Pattern (for different rules)

Factory Pattern (in Dependency Injection)

COMPONENTS AND INTERFACES



DEPENDENCY INJECTION

```
class GoModule extends AbstractModule with ScalaModule{

  override def configure() = {
    install(new FactoryModuleBuilder().implement(classOf[PlayerInterface], classOf[Player]).build(classOf[PlayerFactory]))
    install(new FactoryModuleBuilder().implement(classOf[GridInterface], classOf[Grid]).build(classOf[GridFactory]))
    install(new FactoryModuleBuilder().implement(classOf[CellInterface], classOf[Cell]).build(classOf[CellFactory]))
    install(new FactoryModuleBuilder().implement(classOf[ControllerInterface], classOf[controllerBaseImpl.Controller]).build(classOf[ControllerFactory]))
    bind[CellInterface].to[Cell]

    bind[FileIOInterface].to[fileIoXmlImpl.FileIO]

  }
}
```

```
trait GridFactory {
  def create(size: Int): GridInterface
  def create(cells:Matrix[CellInterface]): GridInterface
}
```

```
val injector = Guice.createInjector(new GoModule)
```

```
def createEmptyGrid(size: Int, player: (String, String)):Unit = {
  val grid = injector.instance[GridFactory].create(size)
}
```

FILE IO IN JSON AND XML

```
{
  "state" : "NEXT_PLAYER",
  "playerOne" : "Player 2",
  "playerTwo" : "Player 1",
  "playerOneCellstatus" : "WHITE",
  "playerTwoCellstatus" : "BLACK",
  "grid" : {
    "size" : 11,
    "cells" : [ {
      "cellstatus" : "EMPTY",
      "row" : 0,
      "col" : 0
    }, {
      "cellstatus" : "WHITE",
      "row" : 0,
      "col" : 1
    }, {
      "cellstatus" : "BLACK",
      "row" : 0,
      "col" : 2
    }, {
```

```
<go>
  <information>
    <activePlayer>
      Player 1
    </activePlayer>
    <activePlayerCellstatus>
      BLACK
    </activePlayerCellstatus>
    <otherPlayer>
      Player 2
    </otherPlayer>
    <otherPlayerCellstatus>
      WHITE
    </otherPlayerCellstatus>
    <state>
      NEXT_PLAYER
    </state>
  </information><grid size="11">
    <cell row="0" col="0">
      WHITE
    </cell><cell row="0" col="1">
      WHITE
    </cell><cell row="0" col="2">
      EMPTY
```

DOCUMENTATION

Go in Scala

This is a implementation of Go in Scala for the SE the class Software Engineering at the University of Applied Science HTWG Konstanz, Germany. (WS 17/18)

Goals of Project

- learning Scala
- learning Git
- Tests
- Srum
- TUI and GUI
- MVC Architecture
- Continious Integregation with Travis CI
- Design Patterns
 - Observer Pattern (for MVC)
 - Command Pattern (do, undo and redo)
 - Strategy Pattern (for different rules)
 - Factory Pattern
- Components and Interfaces
- Dependency Injection
- FILE IO, Serialization in XML and JSON

Go Rules

- <https://senseis.xmp.net/?ChineseRules> (the rules we implemented)
- <https://senseis.xmp.net/?Scoring> (All rules)
- <https://senseis.xmp.net/?ComputerGoAlgorithms> (famous algorithms for Go problems)
- <https://www.brettspielnetz.de/spielregeln/go.php> (in German, but not sure which rules)

Game Instructions

See the Chinese rules link above for rules.

- Start a new game
 - 'n 11' for a 11 * 11 Grid
 - Options -> Size 11 * 11
- Make a turn
 - '1 1' for example places a stone at row 1 and column 1
 - Click on a crossing
- Skip a turn
 - 's'
 - Skip
- Undo/Redo
 - 'z' and 'y'
 - Edit -> Undo/Redo
- Quit
 - 'q'

When two players skip the evaluation start. Every player can mark dead stones or confirm that all dead stones are marked or play the game out.

GUI DEVELOPMENT

