



Hochschule Konstanz
Technik, Wirtschaft und Gestaltung

Dame by Jurie Pato

Konstanz, 28. Juni 2018

Inhaltsverzeichnis

Abbildungsverzeichnis	III
Tabellenverzeichnis	IV
Listingverzeichnis	V
1 Folie 1: Überschrift	1
2 Folie 2: VCS	2
3 Folie 3: MVC	3
4 Folie 4: Code Coverage	4
5 Folie 5: Continuous Integration	5
6 Folie 6: Design Pattern: Observer	6
7 Folie 7: Components Interface	7
8 Folie 8: TUI	8
9 Folie 9: GUI	9
Anhang	10
A.1 Quellcode	10
A.1.1 Quellcode Versuch 1	10
A.1.2 Quellcode Versuch 2	10
A.1.3 Quellcode Versuch 3	10
A.1.4 Quellcode Versuch 4	10

A.2 Messergebnisse	10
Literaturverzeichnis	11

Abbildungsverzeichnis

1.1	Versuchsaufbau	1
2.1	Versuchsaufbau	2
3.1	Versuchsaufbau	3
4.1	Versuchsaufbau	4
5.1	Versuchsaufbau	5
6.1	Versuchsaufbau	6
7.1	Versuchsaufbau	7
8.1	Versuchsaufbau	8
9.1	Versuchsaufbau	9

Tabellenverzeichnis

Listingverzeichnis

1

Folie 1: Überschrift

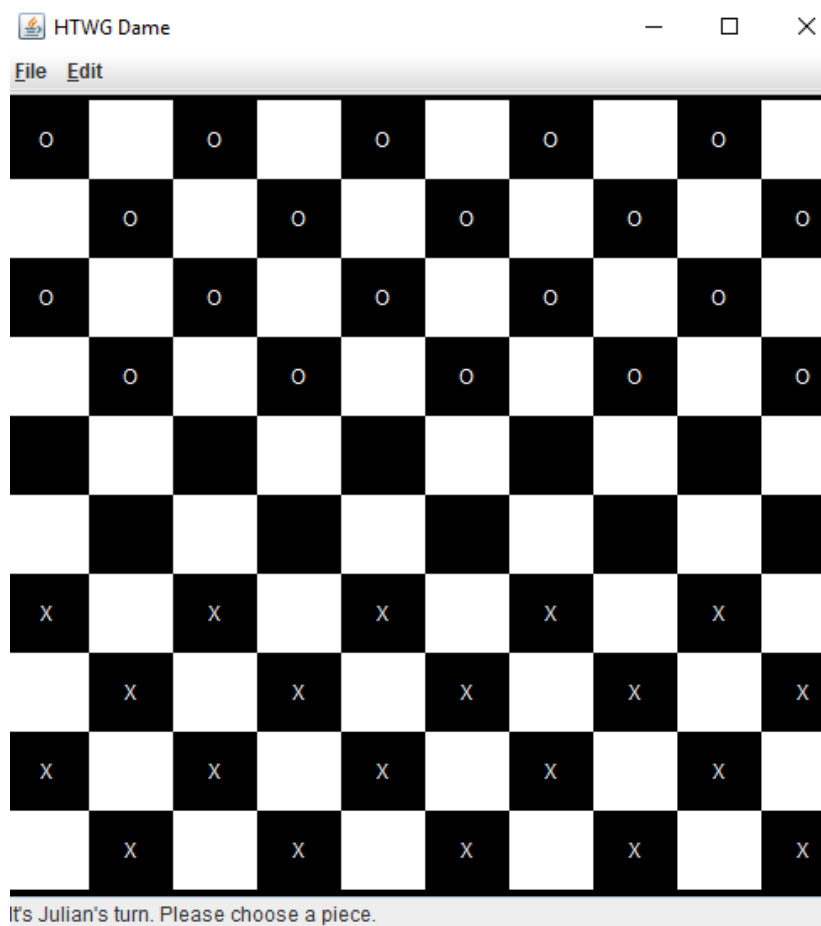


Abbildung 1.1: Versuchsaufbau

2

Folie 2: VCS

Git

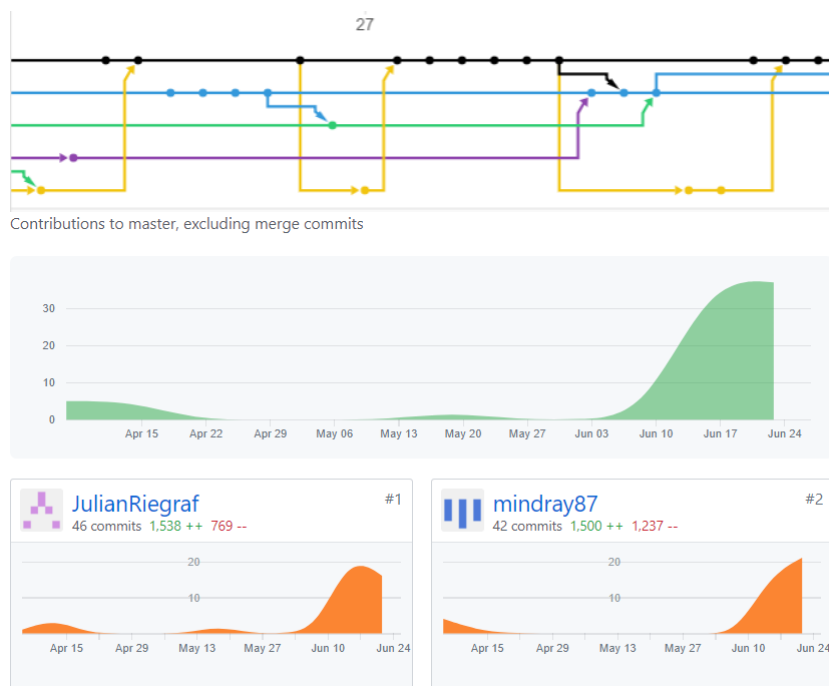


Abbildung 2.1: Versuchsaufbau

3

Folie 3: MVC

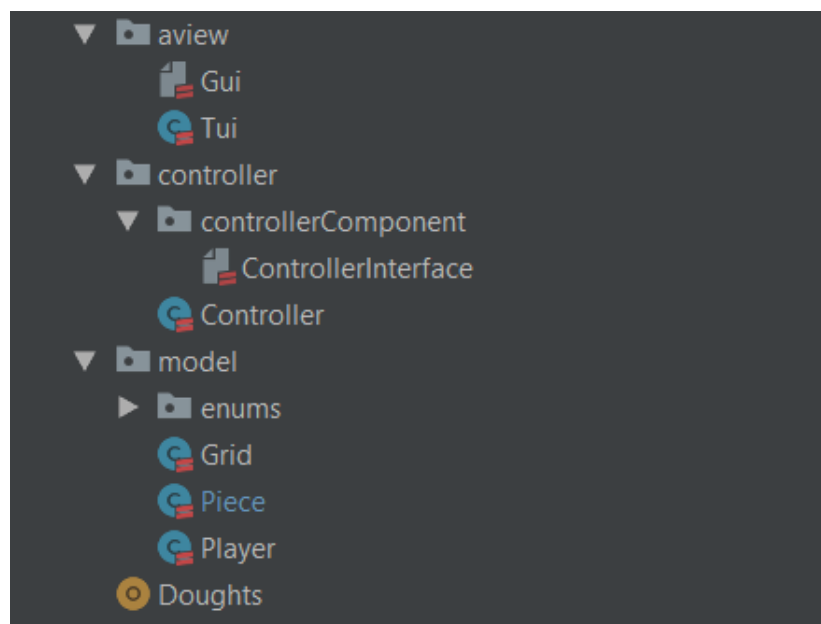


Abbildung 3.1: Versuchsaufbau

4

Folie 4: Code Coverage

50% classes, 100% lines covered in package 'enums'

Element	Class, %	Method, %	Line, %
Color	50% (1/2)	100% (1/1)	100% (2/2)
GameState	50% (1/2)	100% (1/1)	100% (2/2)
PieceType	50% (1/2)	100% (1/1)	100% (2/2)
enums	50% (3/6)	100% (3/3)	100% (6/6)
Grid	50% (1/2)	76% (13/17)	53% (39/73)
Piece	50% (1/2)	50% (1/2)	50% (1/2)
Player	50% (1/2)	75% (3/4)	75% (3/4)

Abbildung 4.1: Versuchsaufbau

5

Folie 5: Continuous Integration

We used Travis, as it was the one Tool that we liked most and also used the badge of course.



Abbildung 5.1: Versuchsaufbau

6

Folie 6: Design Pattern: Observer

We use the Observer Pattern with Events Reactor and Publisher

```
def init : Unit = {  
  player1.pieces = createPieces(player1)  
  player2.pieces = createPieces(player2)  
  
  setInitialPiecePosition(player1, player2);  
  publish(new UpdateEvent)  
}  
  
reactions += {  
  case _ : UpdateEvent => printTui  
  case e : ErrorEvent => println(e.message)  
  case e : PrintMovesEvent => println(controller.gridToString(e.position, e.moves))  
}
```

Abbildung 6.1: Versuchsaufbau

7

Folie 7: Components Interface

```
trait ControllerInterface {  
  def getGridSize: Int  
  
  def undo : Unit  
  def redo : Unit  
  def save : Unit  
  def load : Unit  
  def getPlayer(x: Int): Player  
  def move(src: (Int, Int), dest: (Int, Int)) : Boolean  
}  
  
class UpdateEvent extends Event  
case class ErrorEvent(message : String) extends Event  
case class PrintMovesEvent(position : (Int, Int), moves : List[(Int, Int)]) extends Event
```

Abbildung 7.1: Versuchsaufbau

8

Folie 8: TUI

	0	1	2	3	4	5	6	7	8	9
0	o		o		o		o		o	
1		o		o		o		o		o
2	o		o		o		o		o	
3				o		o		o		o
4	o									
5										
6	x		x		x		x		x	
7		x		x		x		x		x
8	x		x		x		x		x	
9		x		x		x		x		x

Abbildung 8.1: Versuchsaufbau

9

Folie 9: GUI

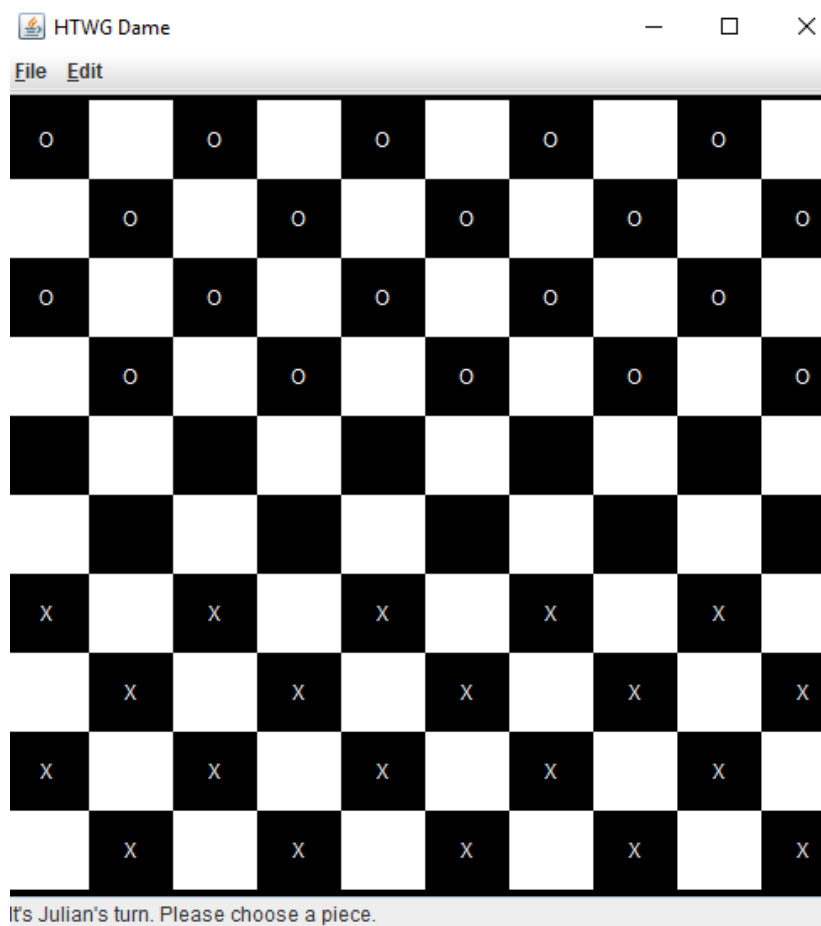


Abbildung 9.1: Versuchsaufbau

Anhang

A.1 Quellcode

A.1.1 Quellcode Versuch 1

A.1.2 Quellcode Versuch 2

A.1.3 Quellcode Versuch 3

A.1.4 Quellcode Versuch 4

A.2 Messergebnisse

Literaturverzeichnis

- [1] Franz, Prof. Dr. Matthias O.: *Vorlesung 10 - Sprache und Spracherkennung: Kurzzeit-Fouriertransformation, Erzeugung und Wahrnehmung von Sprache, Mustererkennung durch Korrelation*. In: *Vorlesung Technische Grundlagen der angewandten Informatik*, 2016.
- [2] Franz, Prof. Dr. Matthias O.: *Vorlesung 1 - Einführung*. In: *Vorlesung Technische Grundlagen der angewandten Informatik*, 2016.