Anforderungsbeschreibung

Auftrag zum Bau eines Anwendungssystems in Form eines Software-Agenten

Stand: V5.10 vom 15. August 2012

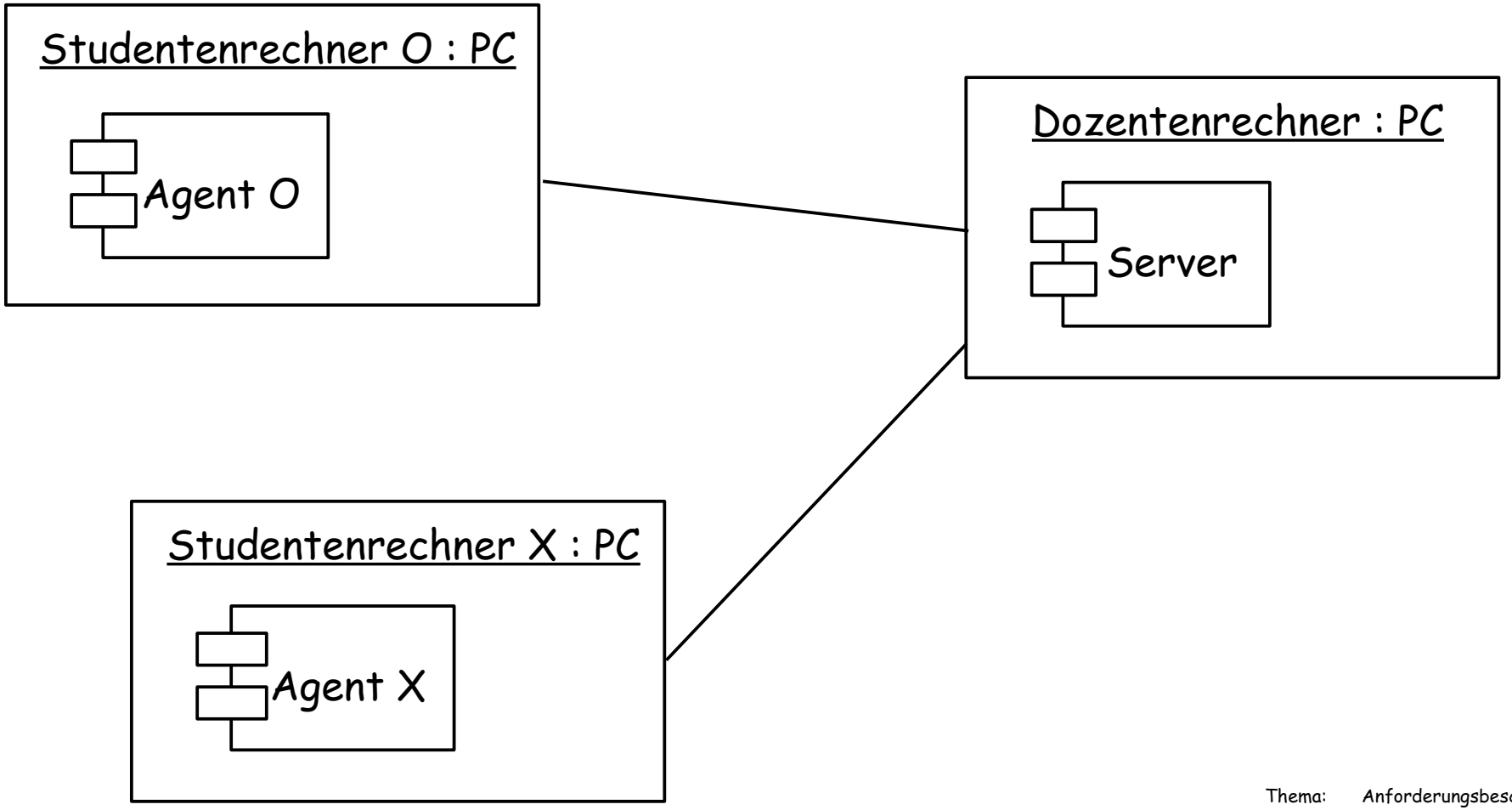
History of changes

Datum	Version	Status	Änderungen
15.08.12	5.10	Freigegeben	Korrekturen und Änderungen von Ai (Di) eingebaut
07.12.09	4.82	Freigegeben	Spielregeln angepasst. Änderungen in 4.82 gelb markiert.
16.11.09	4.81	Freigegeben	Nummernfont korrigiert. Arial sieht unlesbar aus.
16.02.09	4.80	Freigegeben	Vorgaben für Weblink, Formate und Import ergänzt.
29.09.08	4.70	Freigegeben	Anforderungen an Datenhaltung, Lernprojekt und Eigenständigkeit präzisiert.
20.02.08	4.60	Freigegeben	Spezifikation für Prototyp, Release und Datenhaltung präzisiert.
12.02.07	4.50	Freigegeben	Anpassung für Kurs; Prototyp; Hinweis auf Löschen im Kontaktpfad verdeutlicht.
02.05.06	4.40	Freigegeben	marginale Korrekturen
12.09.05	4.30	Freigegeben	Serverdatei ist XML-konform; Kontaktpfad: Zugriffsintervall >=300 ms; Zugzeit 3 s spezifiziert.
01.08.05	4.20	Freigegeben	

Kurzbeschreibung

- Bauen sie einen Software-Agenten, der autonom das Spiel „4 gewinnt“ gegen einen anderen Spieler spielen kann.
- Ihr Agent kommuniziert mit einem Server-Programm, mit dem auch der Gegenspieler, d. h. der Agent des anderen Teams, verbunden ist.
- Den Bau des Agenten führen sie - im Rahmen der vorgegebenen Anforderungen - selbstständig im Team durch. Das bedeutet insbesondere:
  - Sie managen ihr Projekt eigenverantwortlich.
  - Sie gestalten eine Lösung entsprechend ihren Vorstellungen.
  - Sie wenden eine Methodik ihrer Wahl an.
  - Implementierungsdetails liegen weitgehend in ihrem Ermessen.
- Der Rahmen der vorgegebenen Anforderungen wird maßgeblich durch die wichtigsten zu liefernden Ergebnisse bestimmt (genauer siehe nächste Seiten):
  - Agent (Prototyp und Release)
  - Dokumentation
  - Turnierteilnahme

Deployment Diagram



Allgemeine Anforderungen

- Team
  - siehe Glossar
- Anforderungserfüllung
  - In Bezug auf den Ablauf (siehe Anwendungslogik) und die Kommunikation mit dem Server sind alle spezifizierten Anforderungen zwingend einzuhalten.
  - Darüber hinaus darf der Agent wesentlich mehr leisten, sofern dies Ablauf und Kommunikation nicht beeinträchtigt.
  - „Schmalspurlösungen“ - z. B. manuelle Bedienung des Agenten oder Realisierung des Agenten als Konsolenprogramm - sind möglich, entsprechen jedoch nicht dem durchschnittlichen Leistungs- und Anspruchsniveau der Veranstaltung und sollten daher höchsten als Konzeptstudie, Prototyp oder als Vorversion der korrekten Lösung dienen.
- Beachten Sie bitte die Anforderungen bzgl. Lernprojekt auf der folgenden Seite.
- Beachten Sie bitte die Anforderungen bzgl. Eigenständigkeit.
- Beachten Sie bitte die Begriffsklärungen im Glossar.

Anforderungen in einem Lernprojekt

- Dies ist ein **Lernprojekt** in welchem möglichst viele Erfahrungen hinsichtlich bedeutsamer Themen moderner Softwareentwicklung gemacht werden können und sollen. Daher ist das Projekt eigenständig so aufzusetzen, als würde eine kommerziell relevante Software mit einer längeren Lebensdauer entwickelt, woraus sich zwangsläufig eine **Vielzahl möglicher Themen zur vertieften Bearbeitung** ergibt.
- Da nicht alle mit dem Setting Lernprojekt verbundenen Aufgaben aus Zeitgründen vollständig bzw. in praktisch erforderlicher Tiefe gelöst werden können, müssen sich die Teams **auf einzelne Vertiefungsthemen konzentrieren**. Die Auswahl liegt im Ermessen der Teams und richtet sich nach ihren beruflichen Interessen.
- Erfahrungsgemäß wählen Teams als Ganze sowie die Mitglieder der Teams je unterschiedliche Schwerpunkte. Der **Wissenstransfer** ist teamintern durch die Studenten selber sicherzustellen. Teamübergreifend wird der Transfer durch den Dozenten unterstützt und angeregt.
- **Bewertungskriterien** bzgl. der Vertiefungsthemen sind deren **Passgenauigkeit (2)** in Bezug auf das Softwareprojekt sowie die **Qualität (1)** und **Quantität (3)** der Bearbeitung (in Klammern Gewichtung).

Anforderungen bzgl. Eigenständigkeit

- Es dürfen nur die vom Dozenten zugelassenenen Materialien (Tools, Dokumentationen, Bibliotheken, Templates usw.) verwendet werden. Dies sind insbesondere alle genannten Standards bzw. die mit diesen verbundenen Artefakte (z. B. Java-Tutorial, Java-Bücher, Java-Api-Doc, XML-Spezifikation, HSQLDB-User Guide) sowie die vom Dozenten bereitgestellten Bausteine.
- Bestehen Zweifel darüber, ob etwas verwendet werden darf, ist es sicher (!) besser, sich bzgl. des betreffenden Materials beim Dozenten zu erkundigen.
- Die Verwendung spezifischer 4-gewinnt-Algorithmen, spezifischen 4-gewinnt-Codes oder spezifischer 4-gewinnt-Dokumentation aus anderen Quellen als der eigenen Entwicklungsarbeit ist ganz sicher (!) nicht zulässig.

Agent (1)

- Lieferumfang
  - Prototyp
    - Termin: zu Beginn der 4. Veranstaltung
    - Anforderungen:
      - Import muss entsprechend Importanleitung (siehe Glossar) für Dozenten möglich sein.
      - Nach dem Import in Eclipse dürfen keine Compilerfehler angezeigt werden. Laufzeitfehler auf Dozentenrechner sind zulässig.
      - Ein auf Studentenrechner lauffähiger und vorzeigbarer **Oberflächenprototyp** ist gefragt. Die eine oder andere Funktionalität oder ein automatischer Demonstrationsablauf wären schön, sind aber als optionale Anforderungen zu sehen.
  - Beta
    - Termin: zu Beginn der 6. Veranstaltung
    - Anforderungen:
      - Wie Prototyp zzgl.: mindestens 5 Züge über Server spielen können
      - Zufallszug des eigenen Agents genügt; Siegmustererkennung oder Spalte-ist-voll-Erkennung nicht notwendig.
  - Release
    - Termin: am Ende der letzten Veranstaltung (unmittelbar nach Turnier)
    - Anforderungen:
      - auslieferungsfähiges Bundle (zusammen mit der Dokumentation und dem Quellcode als Menge von Dateien auf CD/DVD)
      - mit dessen Hilfe ein Anwender auf der Basis der Dokumentation den Agenten installieren, konfigurieren und bedienen kann
      - und ein Entwickler das Programm warten und erweitern kann (Eclipse-Import muss funktionieren).

Agent (2)

- wichtigste Eigenschaft
  - Funktionsfähigkeit,
  - d. h. fehlerfreier Ablauf des zentralen Geschäftsprozesses, dem Ablauf eines Satzes des Spiels zwischen zwei Agenten über den Server
- Technologie
  - J2SE (nach Maßgabe der Version im jeweiligen PC-Raum)
  - Swing (alternativ JavaFx), HSQLDB
- Weitere Anforderungen zur Architektur des Agenten werden auf den nächsten Folien unter den folgenden Überschriften beschrieben:
  - Benutzerschnittstelle
  - Anwendungslogik
  - Datenhaltung

Agent (3)

- nichtfunktionale Anforderungen
  - während der gesamten Projektlaufzeit (d. h während der Vorlesung) Bereitstellung eines Zugangs zum Teamrepository für den Dozenten
    - Termin: zu Beginn der 2. Vorlesung
    - Form: eine URL per Mail an matthias.lauterbach@dhbw-mannheim.de
    - Über das bzw. im Teamrepository sollen alle entwickelten Artefakte, insbesondere der aktuelle Stand des Quellcodes und der Dokumentation, zugreifbar sein.
    - Arbeitsergebnisse, die für den Dozenten nicht im Repository zugreifbar sind, können leider nicht berücksichtigt werden!
  - Die dem Kunden (dem Dozenten) zur Verfügung gestellten Artefakte sollen mit allgemein(st)en Standardprogrogrammen auf einem normalen MS-Windows-PC angesehen bzw. verwendet werden können. In diesem Sinne geeignete Formate und Werkzeuge sind:
    - *.java, *.txt, *.jpg, *.bmp, *.pdf, *.html, *.css, *.jar, *.zip
    - Open-Office, MS-Office
    - Bitte kein Visio, MS-Projekt, Flash, Silverlight, QuickTime o. ä. und auch keine (anderen) speziellen Plugins oder Clients beim Kunden (Dozenten) voraussetzen.
    - Arbeitsergebnisse, die aus diesen Anforderungen nicht entsprechen, können leider nicht berücksichtigt werden!

Agent (4)

- Evaluation und Rückmeldung
  - Zu Vorlesungsterminen wird der aktuelle Bearbeitungsstand des Projektes gruppenweise bzgl. eines oder mehrere Kriterien durch den Dozenten evaluiert.
  - Zu Vorlesungsterminen werden jeder Gruppe Fragen beantwortet und das aktuelle Evaluationsergebnis mitgeteilt.
- Benotung
  - Die Gruppennote ergibt sich zu je 50 % aus dem Arbeitsprozess (s. laufende Evaluationen) und dem abgelieferten Ergebnis (s. Lieferumfang und Turnier).
  - Achtung: Was der Dozent nicht im Teamrepository findet oder dort nicht lesen kann, siehe hierzu auch Folie Agent (3), das wird als nicht vorhanden gewertet. Daher bitte unbedingt alle relevanten Arbeitsergebnisse sofort, gut auffindbar und entsprechend den Formatforderungen lesbar ins Repository bringen.
    Es wäre sonst wirklich schade drum!

Agent - Benutzerschnittstelle

- Anforderungen an die Benutzerschnittstelle:

- grafische Oberfläche
- Anzeige des Satzstatus
- Anzeige des Spielstandes
- Anzeige Spielfeldes
- Anzeige aller Züge des aktuellen Satzes auf dem Spielfeld
- Realisierung auf der Basis von Swing

Agent - Anwendungslogik (1. Konfigurieren)

- Die Anforderungen an die Anwendungslogik des Agenten werden in Form von Szenarien für das Spielen von Sätzen beschrieben:

Nr	Beschreibung	System	Durchführung
1.1	Namen in Turnierplan eintragen	Powerpoint	Dozent
1.2	server.ini pflegen	Texteditor	Dozent
1.3	AgentO starten oder zurücksetzen	AgentO	Spieler O
1.4	AgentX starten oder zurücksetzen	AgentX	Spieler X
1.5	Server starten oder zurücksetzen	Console / Server	Dozent
1.6	Dateien im Kontaktpfad löschen	Server	Server !!!
1.7	server.ini auslesen und Server konfigurieren	Server	Server
1.8	Konfiguration überprüfen und ggf. anpassen	Server	Dozent
1.9	Konfiguration fixieren	Server	Dozent

Achtung!
'C:' als Kontaktpfad kann zum Verlust von wichtigen Dateien führen!

Agent - Anwendungslogik (2. Satzbeginn festlegen)

Nr	Beschreibung	System	Durchführung
2.1	Satzbeginn selektieren	Server	Dozent
2.2	Satzbeginn fixieren klicken	Server	Dozent
2.3	wenn Zufallsauswahl, dann Startspieler „auslosen“	Server	Server
2.4	beginnenden Spieler anzeigen	Server	Server

Agent - Anwendungslogik (3. Satz spielen)

Nr	Beschreibung	System	Durchführung
3.1	Start klicken	Server	Dozent
3.2	Server in Zustand "Satz spielen" versetzen	Server	Server
	1. Zug		
3.3	einen Agenten mittels Serverfile für einen Zug freigeben	Server	Server
3.4	auf Agentfile warten (s. a. Spezialfall Fehler)	Server	Server
3.5	Serverfile lesen	Agent	Agent
3.6	Zug berechnen und Agentfile schreiben	Agent	Agent
3.7	Agentfile lesen	Server	Server
3.8	Agentfile und Serverfile löschen	Server	Server
3.9	Zug aus Agentfile auswerten und darstellen	Server	Server
3.10	3.3 bis 3.9 für den anderen Agenten	Server	Server
	2. Zug		
3.11	einen Agenten mittels Serverfile für Zug freigeben	Server	Server
3.12	auf Agentfile warten (s. a. Überschreitung der Zugzeit)	Server	Server
3.13	usw.		

Agent - Anwendungslogik (Spezialfall: Satzstart)

- Nach dem Klicken von Start am Server erhält der Agent des Startspielers die 1. Serverdatei mit folgendem Inhalt:

```
<freigabe>true</freigabe>
<satzstatus>Satz spielen</satzstatus>
<gegnerzug>-1</gegnerzug>
<sieger>offen</sieger>
```

- Nachdem der Startspieler seinen Zug gemacht hat, im Beispiel hier einen Stein in Spalte 5, geht die 2. Serverdatei an den anderen Spieler bzw. Agenten:

```
<freigabe>true</freigabe>
<satzstatus>Satz spielen</satzstatus>
<gegnerzug>5</gegnerzug>
<sieger>offen</sieger>
```

Agent - Anwendungslogik (Spezialfall: Fehler)

- Folgende Fehler von Agenten kann der Server feststellen:
  - F1: Keine Agentendatei innerhalb der Zugzeit (Timeout).
  - F2: Agentendatei ungültig, wenn
    - das erste Zeichen in der Datei keine ganze Zahl zwischen 0 und 6 darstellt oder
    - in eine Spalte, die schon voll ist, "geworfen" wird.
- Reaktion auf F1
  - Der Timeout wird vom Server angezeigt.
  - Der Server ermittelt einen zufälligen Zug für den Agenten.
  - Der Agent erhält kein Serverfile mit dem Zufallszug.
  - Dem Agenten des Gegners wird der Zug wie ein ganz normaler Zug mittels Serverfile mitgeteilt.
- Reaktion auf F2
  - Der Server geht in den Zustand "gestoppt" und gibt eine Fehlermeldung an der Konsole aus.
  - Nach Klicken von Weiter wird ein Zufallszug anstelle des falschen Zuges ausgeführt.
  - Die Agenten werden wie bei F1 informiert.

Agent - Anwendungslogik (Spezialfall: Satzende)

- Folgende Ereignisse führen zum Ende eines Satzes:
  - E1: Gewinnsituation erkannt. E2: Spielfeld ist voll. E3: Manuelles Beenden des Servers.
- Reaktion auf E1: Wird eine Gewinnsituation erkannt, z. B. wenn Spieler O mit Wurf in Spalte 3 gewonnen hat, dann werden beide Agenten informiert.
  - server2spielero.xml:
    - <freigabe>false</freigabe>
<satzstatus>beendet</satzstatus>
<gegnerzug>-1</gegnerzug>
<sieger>Spieler O</sieger>
  - server2spielerx.xml
<freigabe>false</freigabe>
<satzstatus>beendet</satzstatus>
<gegnerzug>3</gegnerzug>
<sieger>Spieler O</sieger>
- Reaktion auf E2: Ist das Spielfeld voll, z. B. nachdem Spieler O den letzten möglichen Zug in Spalte 4 gemacht hat, dann wird der Sieger per Zufall ermittelt - hier z. B. Spieler X - und beiden Agenten mitgeteilt.
  - server2spielero.xml:
    - <freigabe>false</freigabe>
<satzstatus>beendet</satzstatus>
<gegnerzug>-1</gegnerzug>
<sieger>Spieler X</sieger>
  - server2spielerx.xml
<freigabe>false</freigabe>
<satzstatus>beendet</satzstatus>
<gegnerzug>4</gegnerzug>
<sieger>Spieler X</sieger>

Agent - Anwendungslogik (Spezialfall: Satzende)

- Reaktion auf E3: Manuelles Beenden des Servers.
  - Beschreibung
    - Ein laufender oder gestoppter Satz wird am Server durch Klicken auf Button Beenden abgebrochen.
    - Abbruch z. B. nicht wegen E1 oder E2, sondern wegen F1 oder F2.
  - Ein abgebrochener Satz muss vom Agenten sinnvoll behandelt werden.
  - minimale funktionale Anforderungen, die letztlich unter der Überschrift "nachträgliches Editieren von Ergebnissen" zusammengefasst werden können
    - Satz verwerfen
      - Es werden keine Daten gespeichert.
    - Satz nicht verwerfen
      - alle eigenen und gelesenen Züge bis zum Abbruch automatisch speichern
      - Sieger von Satz sowie die Punktezahl manuell überschreiben können
    - Sieger des Spiels überschreibar

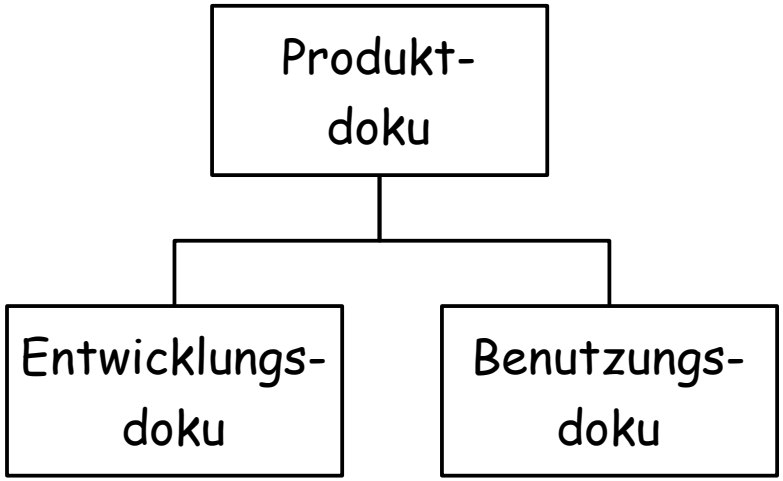
Agent - Datenhaltung

Anforderungen an die Datenhaltung:

- Speichern der Daten in einer Datenbank
- automatische Speicherung der Daten durch den Agenten
- zu speichernde Daten sind:
  - Gegner, Startspieler, Sieger, Punkte, Spiele, Sätze, Züge
  - minimal: alle Daten nach Maßgabe des Standardmodus, keine Punkte speichern
- Realisierung auf der Basis von HSQLDB
- wenigstens drei Abfragen bzw. Abfragevarianten implementieren; Minimalanforderung:
  - alle gespielten Spiele anzeigen
  - alle Züge eines frei wählbaren Spiels der angezeigten Spiele anzeigen
- wenigstens drei Spiele für Beispielabfragen mit Release-Version ausliefern

Dokumentation

- Eine Produktdokumentation ist erforderlich. Zum Lieferumfang gehören hier mindestens:
  - ~ Entwicklungsdokumentation
    - + erforderlich für Wartung und Pflege des Anwendungssystems
    - + zeigt z. B. Produktstruktur und Modelle
    - + Prozessmodellierung nach ARIS, Grögler oder UML
    - + dokumentiert wichtige Entwurfsentscheidungen
  - ~ Benutzungsdokumentation
    - + beschreibt die Bedienung
    - + enthält hier auch alle notwendigen Angaben für Installation und Konfiguration der Anwendung
- Sie können weitere Bestandteile zur Dokumentation hinzufügen oder die Dokumentation anders strukturieren. Die geforderten Elemente sollen jedoch auf jeden Fall enthalten sein und eindeutig identifiziert werden können.



Turnierteilnahme

- siehe Spielregeln
- siehe Turnierplan
  - Vorrunde
  - Endrunde

Spielregeln (Standardmodus)

optional

- Punkte: Der Sieger eines Satzes erhält einen Punkt.
  Kommt es zu keiner Gewinnsituation, ermittelt der Server per Zufall einen Sieger.
- Spiel: Ein Spiel erfordert drei Sätze: einen Hinsatz (Team O beginnt), einen Rücksatz (Team X beginnt) und einen Zufallssatz (zufällige Auswahl des beginnenden Teams). Wer zwei oder mehr Punkte erzielt, hat das Spiel gewonnen.
- Vorrunde (Turnier mit 5 Teams): Eine Vorrunde ist nötig bei 5 oder 6 Teams. Team 5, d. h. das mit der höchsten Augenzahl, ist sofort in der Endrunde. Die 4 Teams mit den niedrigsten Nummern treten in 2 Spielen an. Die Gewinner ziehen in die Endrunde ein. Die Verlierer spielen um den letzten freien Endrundenplatz.
- Vorrunde (Turnier mit 6 Teams): Die Teams 5 und 6, also die mit den höchsten Augenzahlen in der Auslosung, ziehen direkt in die Endrunde ein. Die Teams mit den Nummern 1 bis 4 treten in 2 Spielen an. Die Verlierer spielen um die Plätze 5 und 6. Die Gewinner ziehen in die Endrunde ein.
- Endrunde: In den ersten beiden Spielen des Halbfinales (1. und 2. Spiel der Endrunde) werden die im Finale und im Spiel um Platz 3 antretenden Teams ermittelt. Die Verlierer der Halbfinalspiele spielen um Platz 3 (3. Spiel der Endrunde), die Gewinner stehen im Finale.

Spielregeln (Vorsprungsmodus)

optional

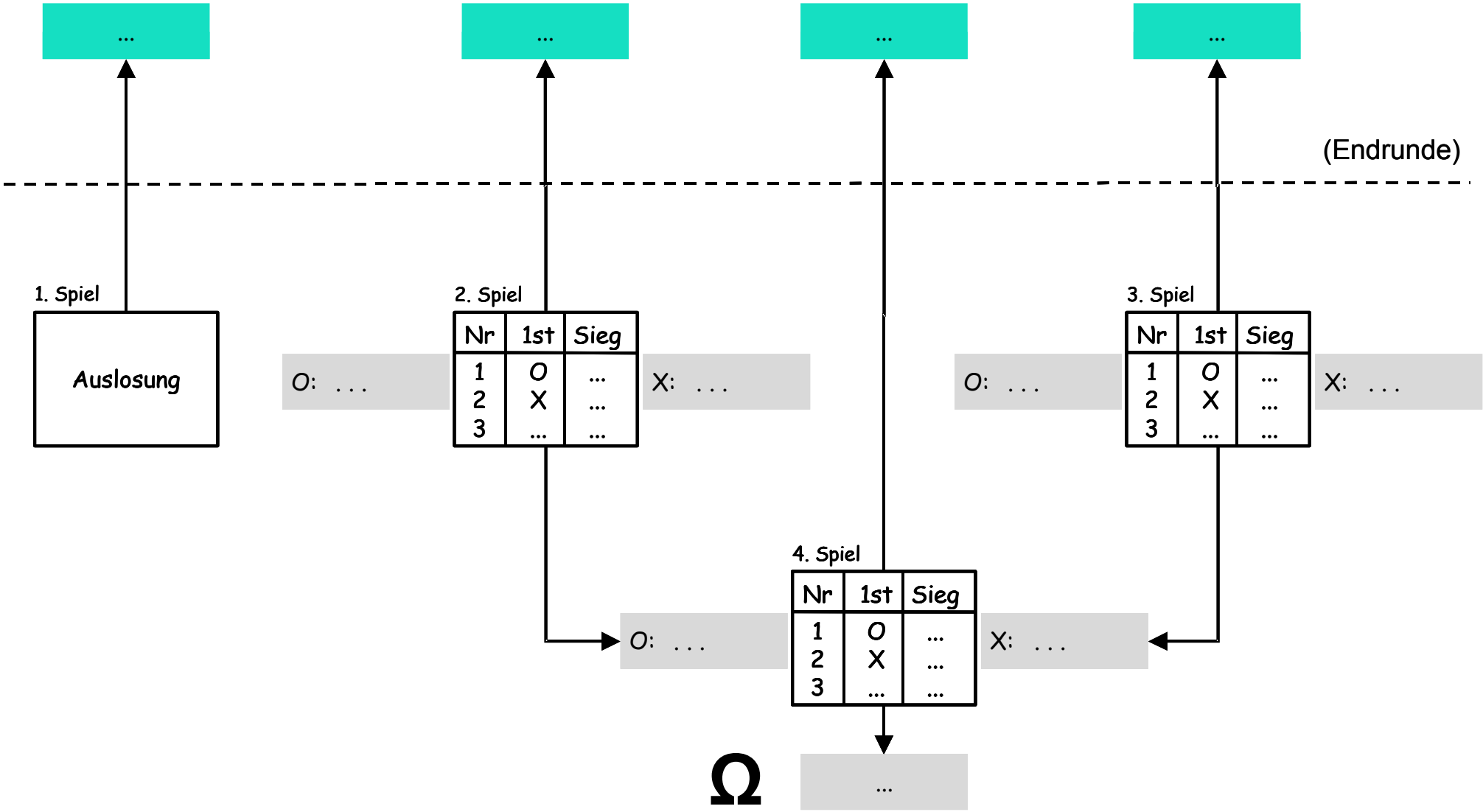
- Punkte: Im 1. bis 4. Satz werden - pro Satz und pro Agent und für das Turnierergebnis relevant - wie folgt Punkte vergeben: Sieg => 1 Punkte, Remis => 0 Punkte und Niederlage => 0 Punkte (wie im Standardmodus). Kommt es ab dem 5. Satz nicht zu einer Gewinnsituation, dann entscheidet der Server, d. h. ein Remis in einem Satz ist ab dem 5. Satz ausgeschlossen.
- Spiel: Ein Spiel gilt als gewonnen, sobald ein Agent 2 Punkte Vorsprung hat. Hat nach den ersten 2 Sätzen kein Spieler 2 Punkte Vorsprung, dann wird die Zugzeit vermindert und das Spiel fortgesetzt. Steht nach dem 4. Satz immer noch kein Sieger fest, wird ab jetzt der Startspieler immer zufällig gewählt und ein eventuell aufgetretenes Remis immer vom Server durch Auslosung des Siegers verhindert. Es werden dann solange Sätze gespielt, bis ein Sieger feststeht. Die Teams dieses Spiels können ab dem 5. Satz - einstimmig und bevor der Startspieler feststeht ! - satzweise über die Zugzeit entscheiden.
- Vorrunde: Die Teams 5 und 6, also die mit den höchsten Augenzahlen in der Auslosung, ziehen direkt in die Endrunde ein. Die Teams mit den Nummern 1 bis 4 treten in 2 Spielen an. Die Verlierer spielen um die Plätze 5 und 6. Die Gewinner ziehen in die Endrunde ein.
- Endrunde: In den ersten beiden Spielen des Halbfinales (1. und 2. Spiel der Endrunde) werden die im Finale und im Spiel um Platz 3 antretenden Teams ermittelt. Die Verlierer des Halbfinales spielen um Platz 3 (3. Spiel der Endrunde), die Gewinner stehen im Finale.

Spielregeln (Punktemodus)

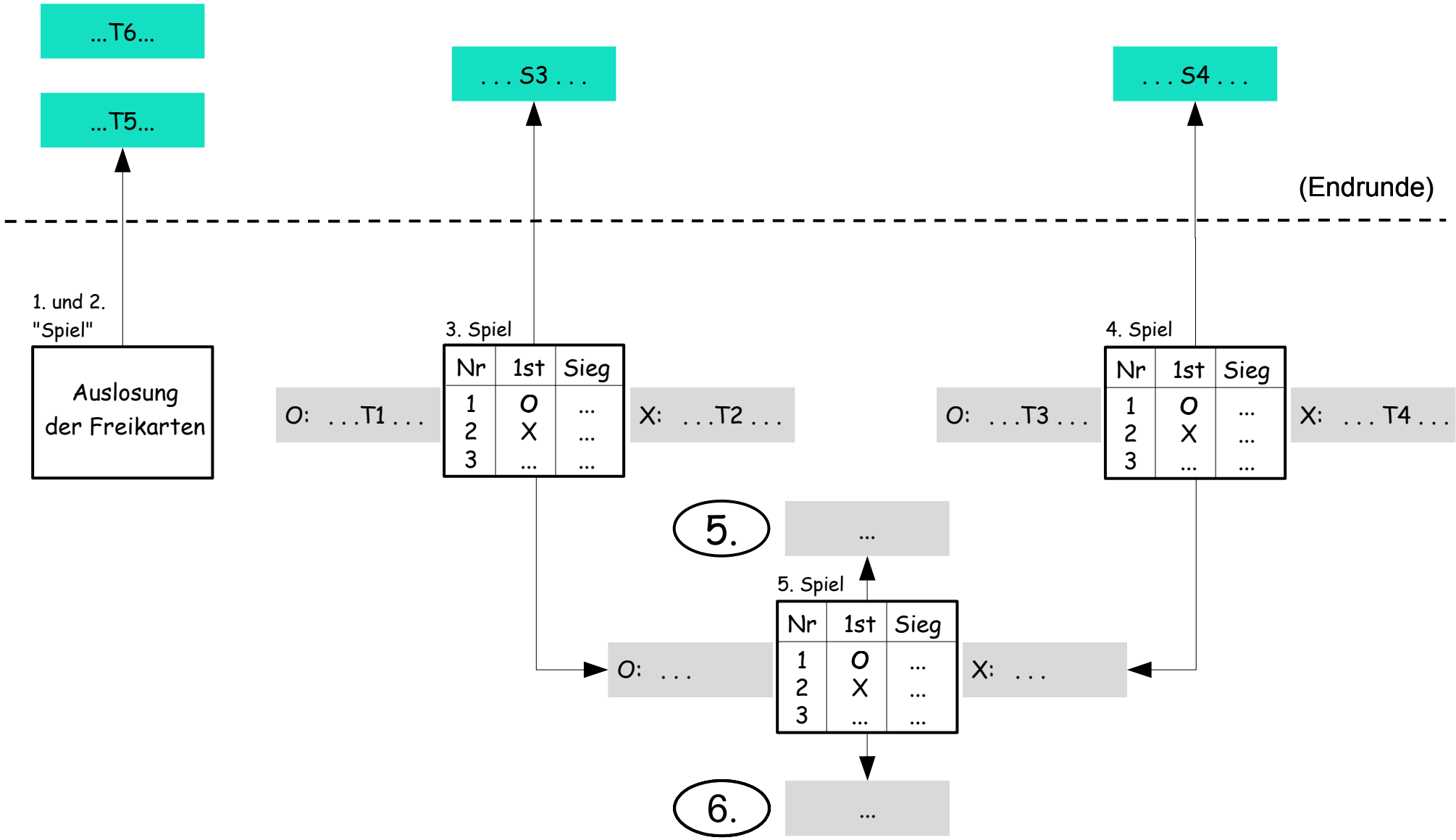
- Punkte: Pro Satz und Agenten werden für das Turnierergebnis relevant wie folgt Punkte vergeben:
  Sieg => 2 Punkte, Remis => 1 Punkt und Niederlage => 0 Punkte.
- Spiel: Ein Spiel besteht in Runde 1 immer aus zwei Sätzen, in Runde 2 immer aus drei Sätzen.
- Runden: Das Turnierergebnis wird in ein oder zwei Runden ermittelt. Runde 2 ist nur nötig, wenn in Runde 1 kein eindeutiges Ergebnis bestimmt werden konnte.
- Runde1: Jeder spielt mit jedem (Bei n = 6 Teams: n!/(2*(n-2)! => 15 Spiele a 2 Sätze).
- Runde 2: Alle Teams für die keine eindeutige Aussage bzgl. Ihres Turnierplatzes getroffen werden kann, gehen in die 2. Runde. Im relativ unwahrscheinlichen worst case sind dies alle Teams. Runde 2 verläuft wie Runde 1, aber mit verminderter Zugzeit und drei Sätzen pro Spiel. Steht es nach dem 2. Satz immer noch unentschieden, entscheidet der 3. Satz bei zufälliger Wahl des Startspielers und Remisauflösung durch den Server. Die Teams dieses Spiels können für den 3. Satz - einstimmig und bevor der Startspieler feststeht ! - beschließen, im 3. Satz die Zugzeit erneut zu senken.

Spielregeln (alle Modi)

- Auslosung: Die Nummern der Teams werden durch Würfeln "ausgelost". Das Team mit der niedrigsten Augenzahl bekommt die Nummer 1. Das Würfeln erfolgt in der Reihenfolge der alphabetisch aufsteigend sortierten Teamnamen.
  Ist eine Augenzahl schon gefallen, wird bis zur nächsten eindeutigen Augenzahl gewürfelt.
- Satz: Ein Agent spielt gegen einen anderen Agenten über den Server. Kommt es zu keiner Gewinnsituation, ermittelt der Server immer einen Zufallssieger, der aber nur abhängig vom Spielmodus für das Turnierergebnis relevant ist.
- Startspieler: Den ersten Satz eines Spiels beginnt immer Spieler O. Nach jedem entschiedenen Satz wechselt der Startspieler zwingend; einzige Ausnahme: Die ggf. zufällige Bestimmung des Startspielers durch den Server. Wer Spieler O und wer Spieler X ist, das ergibt sich aus Auslosung und Turnierplan und kann sich im Turnierablauf zudem ändern.
- Regeländerungen: Mit Zustimmung der jeweils betroffenen Teams können Spielregeln geändert werden, einstimmige Entscheidungen vorausgesetzt. Z. B. ist es manchmal vorteilhaft, die sich aus dem Turnierplan ergebende Reihenfolge von Spielen und Sätzen ändern zu können. Können sich Team oder Kurs nicht einigen, bestimmt der Turnierleiter.
- Erfolg: Aus Sicht der Vorlesung ist die funktional-technisch erfolgreiche Teilnahme am Turnier, nicht die Plazierung entscheidend.
- Fehler: Einmaliges funktionales oder technisches Versagen eines Agenten pro Satz wird als negativ angesehen, der Satz wird jedoch nochmals gestartet.
- Doppelfehler: Ein zweiter Fehler eines Agenten in einem Satz kann vom Spielpartner in der Art toleriert werden, das der betreffende Satz nochmals gestartet wird. Das Gegnerteam muss dann allerdings auch das durch das Spielen des zum dritten Mal gestarteten Satzes zustande gekommene Punkteergebnis akzeptieren. Toleriert der Gegner den Doppelfehler nicht, gilt der Satz für das fehlerverursachende Team als verloren.
- Disqualifikation: Eine Manipulation des Spiels wird als Betrug gewertet und führt zur Disqualifikation des Teams.

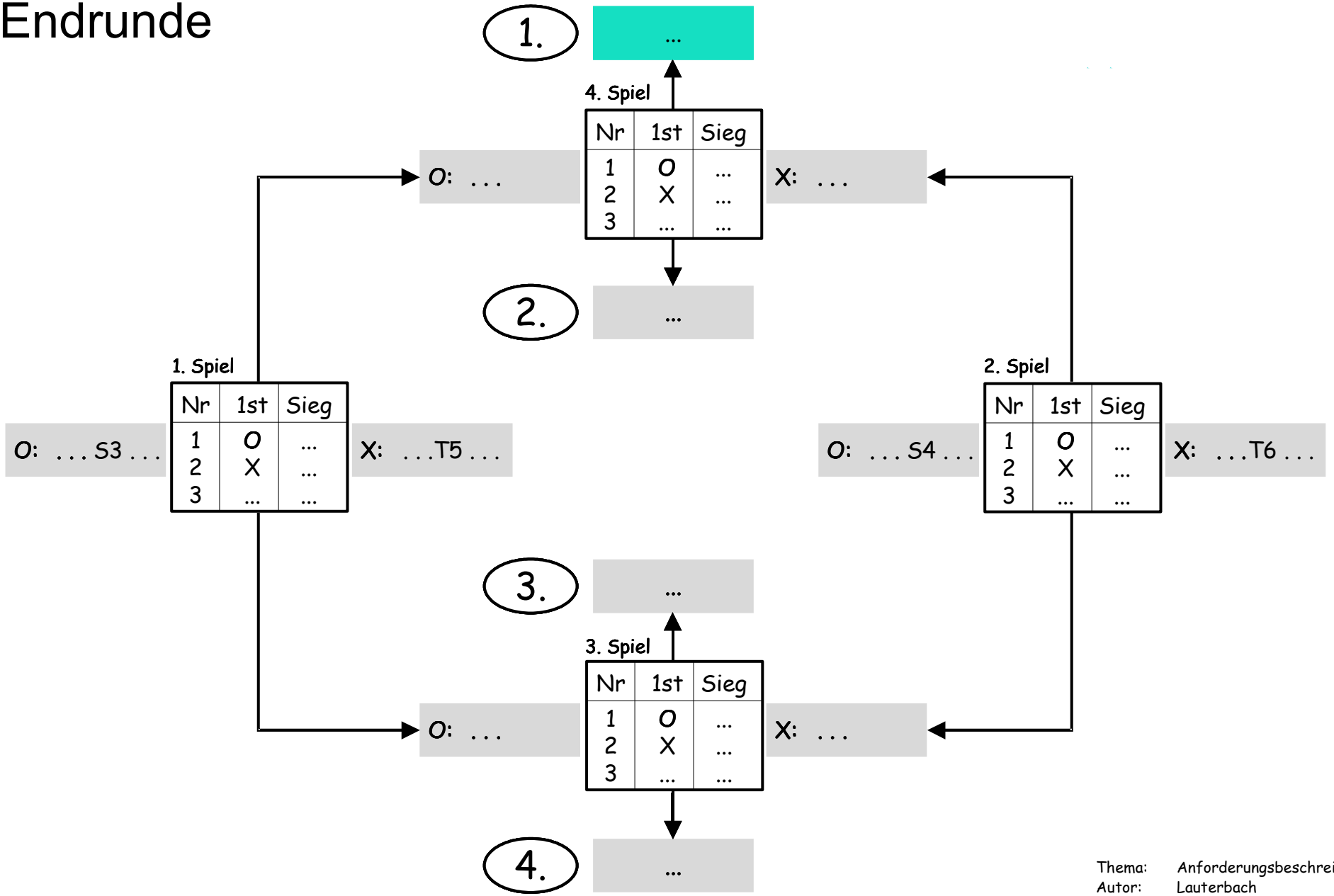


Vorrunde (Standardmous, 5 Teams)



Vorrunde (Standardmodus, 6 Teams)

Endrunde


1.
...
4. Spiel
Nr 1st Sieg
1 O ...
2 X ...
3 ... ...
O: . . .
X: . . .
2.
...
1. Spiel
Nr 1st Sieg
1 O ...
2 X ...
3 ... ...
O: . . . S3 . . .
X: . . .T5 . . .
2. Spiel
Nr 1st Sieg
1 O ...
2 X ...
3 ... ...
O: . . . S4 . . .
X: . . .T6 . . .
3.
...
3. Spiel
Nr 1st Sieg
1 O ...
2 X ...
3 ... ...
O: . . .
X: . . .
4.
...

Turnierplan (Punktemodus, 6 Teams)

Spiel	1		2		3		4		5		6		7		8		9		10		11		12		13		14		15		
Satz	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	
Team																															**Punkte**
s. Tab W&A	0	0									0	0							0	0					0	0			0	0	**0**
s. Tab W&A	0	0	0	0									0	0							0	0					0	0			**0**
s. Tab W&A			0	0	0	0					0	0			0	0							0	0							**0**
s. Tab W&A					0	0	0	0					0	0			0	0	0	0											**0**
s. Tab W&A							0	0	0	0					0	0					0	0			0	0					**0**
s. Tab W&A									0	0							0	0					0	0			0	0	0	0	**0**

Glossar A

- **Agent**
  - Programm, welches autonom am Spiel teilnimmt.
  - Agiert in der Rolle eines Spielers.
  - Wird durch ein Team entwickelt und verwendet.
- **Agentfile**
  - Datei zur Kommunikation von Agent zu Server
  - Agent darf Agentfile überschreiben und löschen
  - Name: spielero2server.txt oder spielerx2server.txt
  - Format: Textdatei, ASCII
  - Inhalt: genau eine Ziffer für die gewählte Spalte, z. B. 3 (Texteditor zeigt auch 3 an)

Glossar B - I

- **Bausteine**
  - Zusammenstellung essentieller Deliverables im Sinne von Schwerpunktsetzung/Minimalanforderungen:
  - Modellierung: UML-Klassendiagramm des Agenten als Analysemodell
  - Agent: lauffähiges und leicht zu installierendes Konsolenprogramm mit Zufallslogik
  - Doku: Pflichtenheft, Entwicklungsdokumentation, Benutzerhandbuch
  - Management: realistischer, aktueller Projektplan
- **Importanleitung**
  - Der Import von Prototyp, Beta und Release (Quellcode incl. aller erforderlichen Ressourcen) muss gemäß des in http://www.nadviser.de/Bausteine/IdeEclipse/ExportImportGuide.txt (Variante 1!) beschriebenen Ablaufs möglich sein.

Glossar K - Sa

- **Kontaktpfad**
  - synonym zu Kommunikationspfad, Kontaktverzeichnis, Kommunikationsverzeichnis
  - das Verzeichnis, in dem Agent- und Serverfiles ausgetauscht werden
  - Zwischen 2 Zugriffen eines Agents auf den Kontaktpfad müssen mindestens 300 ms liegen.
  - Der Kontaktpfad ist von Spiel zu Spiel unterschiedlich.
  - Achtung: Der Server löscht alle Dateien im Kontaktverzeichnis. Verwenden Sie daher stets ein dezidiertes Verzeichnis zur Kommunikation von Server und Agent, z. B. F:\Kontakt oder C:\ComDir, keinesfalls C:\.
- **Satzstatus**
  - Status des aktuellen Satzes; s. a. Serverfile
- **Satz**
  - Ein Agent spielt gegen einen anderen Agenten über den Server. Der Sieger erhält einen Punkt. Kommt es zu keiner 4g-Situation, ermittelt der Server per Zufall einen Sieger.

Glossar Se

Serverfile

- Datei zur Kommunikation von Server zu Agent
- <dateiname> ::= server2spielero.xml | server2spielerx.xml
- <inhalt> ::= <declaration> <content>

<declaration> ::= "<"?xml version='1.0' encoding='utf-8'?">"
<content> ::= "<"content">" <freigabezeile> <statuszeile> <zugzeile> <siegerzeile> "<"/content">"
<freigabezeile> ::= "<"freigabe">" true "<"/freigabe">" | "<"freigabe">" false "<"/freigabe">"
<statuszeile> ::= "<"satzstatus">" Satz spielen "<"/satzstatus">" | "<"satzstatus">" beendet"<"/satzstatus">"
<zugzeile> ::= "<"gegnerzug">" <spalte> "<"/gegnerzug">"
<spalte> ::= 0 | 1 | 2 | ... | 6 | -1
<siegerzeile> ::= "<"sieger">" <sieger> "<"/sieger">"
<sieger> ::= offen | Spieler X | Spieler O

Beispiel für Serverfile von Server an Team X (Dateiname: server2spielerx.xml):

```
<?xml version='1.0' encoding='utf-8'?>
<content>
        <freigabe>true</freigabe>
        <satzstatus>Satz spielen</satzstatus>
        <gegnerzug>-1</gegnerzug>
        <sieger>offen</sieger>
</content>
```

Glossar Sp

- **Spalte**
  - Senkrechte im Spielfeld
  - Nummerierung von links nach rechts: 0, 1, 2, 3, 4, 5, 6
- **Spiel**
  - Das Spiel heißt "4 gewinnt".
  - Ein Spiel erfordert im Standardmodus drei Sätze: einen Hinsatz (Spieler O beginnt), einen Rücksatz (Spieler X beginnt) und einen Zufallssatz (zufällige Auswahl des beginnenden Spielers).
- **Spieler**
  - Teilnehmer am Spiel
  - Rolle in Bezug auf die die Spiellogik
  - s. a. Team und Agent
- **Spielfeld**
  - 7 Spalten x 6 Zeilen
- **Spielstand**
  - Angabe der jeweils gewonnenen Sätze im aktuellen Spiel, z. B. 0:0, 0:1 o. ä.

Glossar T - Ze

- **Team**
  - baut einen Agenten und verwendet diesen in der Rolle eines Spielers
  - Die Begriffe Agent, Spieler und Team können oft synonym verwendet werden.
  - Teamname
    - Festlegung durch Team
    - 3 bis 15 Kleinbuchstaben oder Ziffern
    - muss mit Kleinbuchstaben beginnen
  - Teamgröße
    - minimal 4, maximal 5 Mitglieder
    - Dies erfordert auch Abstimmung zwischen den Teams, denn kleinere oder größere Teams sind nicht zulässig.
- **Zeile**
  - Waagerechte im Spielfeld
  - Es gibt 6 Zeilen.
  - Nummerierung
    - im Agent beliebig
    - im Server von oben nach unten: 2, 3, 4, 5, 6, 7

Glossar Zu

- **Zugzeit**
  - Zeit, die der Server nach dem Lesen eines Agentfiles bis zum Lesen des nächsten Agentfiles (dem des Gegners) wartet
  - beträgt in der Regel 2 s (Standardeinstellung)
  - Verkürzung auf 1 s (im Stechen) oder Verlängerung auf 3 s (bei Timeout-Problemen beider Agenten) kann bedarfsweise vom Turnierleiter festgelegt werden.