Lucerne University of
Applied Sciences and Arts

HOCHSCHULE
LUZERN

Informatik

Modul Programmieren & Datenstrukturen

Battleship - Projektdokumentation

Vorgabe der Dozenten

Inhaltsverzeichnis

Abbildungen

Versionen:

Rev.	Datum	Autor	Bemerkungen
0.1	27.03.2013	tisuter	Initiale Version
0.2	08.04.2013	tisuter	Überarbeitung mit P. Sollberger
0.3	17.04.2013	zasollbe	Einleitung angefügt
0.4	29.03.2017	zaklaper	Logo, Kosmetik, Typos behoben

1. Einleitung

Geschätzte Studierende

in der zweiten Hälfte des Semesters bearbeiten Sie in einer kleinen Gruppe ein kleineres Softwareprojekt. Mit diesem Projekt sind die folgenden Ziele verbunden:

- Sie wenden im Unterricht gelernte Konzepte der Sprache Java in einem grösseren Kontext an.
- Sie wiederholen wesentliche Elemente der Programmiersprache Java.
- Sie implementieren eine Softwarelösung im Team.
- Sie können Kontextdiagramme lesen und interpretieren.
- Sie können Komponentendiagramme lesen und interpretieren.
- Sie können Sequenzdiagramme lesen und interpretieren.
- Sie können Zustandsdiagramme lesen und interpretieren.
- Sie können Sourcecode mit Hilfe von Klassendiagrammen dokumentieren.
- Sie können in einem grösseren Programm die Übersicht wahren.

Das gesamte Projekt ist als Lernprojekt zu verstehen, bei dem Sie Schritt für Schritt Kenntnisse erwerben und anwenden.

Die Dozierenden und Assistierende begleiten Sie während des Projekts und ermöglichen Ihnen, wesentliche Erfahrungen zu reflektieren. Zur Unterstützung dieses Prozesses erstellen Sie jede Woche für die Dozierenden einen kurzen Projekt-Statusrapport mit folgendem Inhalt:

- Welche Arbeiten wurden in der letzten Woche ausgeführt. Was hat gut geklappt, wo hatten oder haben Sie Probleme?
- Welche Tätigkeiten sind für die nächste Woche vorgesehen?
- Welche Knackpunkte (Herausforderungen oder Risiken) bestehen noch? Was gedenken Sie dagegen zu unternehmen?

Eine Vorlage für diesen Projekt-Statusrapport [2] finden Sie im ILIAS.

Viel Erfolg sowie spannende und wertvolle Projekterfahrungen wünschen Ihnen

Ihr Dozierendenteam

1.1. Begriffe, Abkürzungen

Die Anwendung wird in Englisch erstellt. Gewisse Begriffe werden aber in dieser SysSpec auf Deutsch benutzt. In der folgenden Tabelle sind die Übersetzungen.

Engl. Begriff	*Dt. Begriff*
Battleship	Schiffe versenken
Grid	Spielfeld
Engine	Spielsteuerung
Opponent	Gegenspieler

1.2. Referenzen

[1] Projektauftrag Schiffe versenken (Projektauftrag.pdf)

[2] Projekt-Statusrapport (PRG2_Projekt-Statusrapport.docx)

2. Anforderungen

Aus der Aufgabenstellung [1] lassen sich zwei User Storys ableiten:

User Story 1:

Als Spieler möchte ich über das Netzwerk gegen einen Gegenspieler Schiffe versenken spielen um Spass zu haben.

User Story 2:

Als Spieler möchte ich gegen einen Computergegner Schiffe versenken spielen um Spass zu haben.

2.1. Systemübersicht

Aus diesen User Stories lässt sich das System visualisieren.

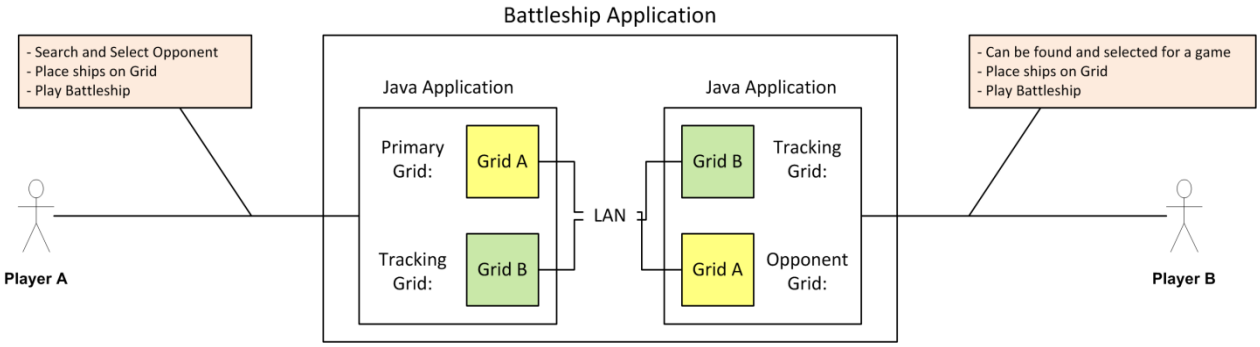

Abbildung 1: Kontextdiagramm der Battleship Application bei einem Spiel gegen einen Gegenspieler über das Netzwerk

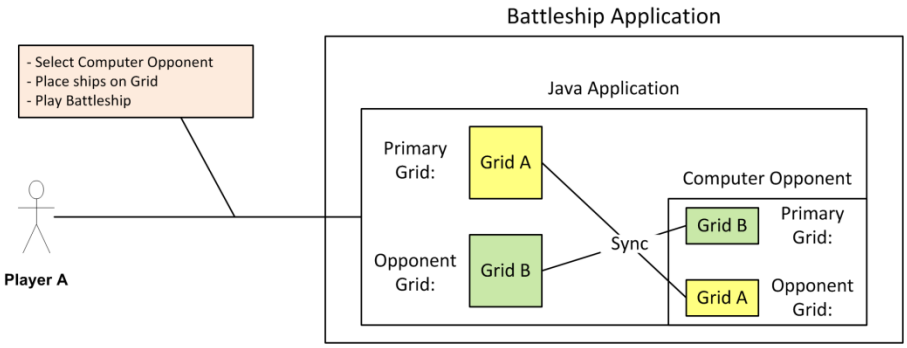

Abbildung 2: Kontextdiagramm der Battleship Application bei einem Spiel gegen einen Computergegner

Bei beiden Anwendungen wird deutlich, dass es vier Grids gibt, wobei jeweils zwei miteinander synchronisiert werden müssen. Diese Synchronisation ist die einzige Netzwerkaktivität während des Spiels.

2.2. Anwendungsfall

Hier ist der Anwendungsfall, welcher aus der User Story 1 abgeleitet wurde.

Name	Spiel gegen Netzwerkgegenspieler
Beschreibung	Netzwerkgegenspieler werden gesucht und danach wird gegen einen dieser Gegenspieler Battleship gespielt.
Beteiligte Akteure	User (Player A) Netzwerkgegenspieler (Player B)
Auslöser	Zwei Spieler wollen gegeneinander Battleship spielen.
Vorbedingungen	Beide Spieler haben die Applikation gestartet Die beiden Spieler befinden sich im selben Netzwerk
Ergebnis	Einer der beiden Spieler gewinnt das Spiel.
Standardablauf	1. Mögliche Gegenspieler werden im Netzwerk gesucht 2. Ein möglicher Gegenspieler wird selektiert, damit Battleship gespielt werden kann. 3. Wenn der Gegenspieler akzeptiert, wird ein neues Spiel erzeugt. 4. Beide Spieler setzen ihre Schiffe auf dem eigenen Spielfeld. Dabei gelten folgende Regeln: - Jedes Schiff besetzt eine bestimmte Anzahl von horizontal oder vertikal aufeinanderfolgenden Feldern. - Die Schiffe können sich nicht überlappen - Beide Spieler haben dieselben Schiffe 5. Sobald alle Schiffe gesetzt sind kann der Spieler sein eigenes Spielfeld als bereit markieren. Ab diesem Zeitpunkt können die Schiffe nicht mehr bewegt werden. 6. Sobald beide Spieler bereit sind beginnt das eigentliche Spiel. 7. Der Spieler der an der Reihe ist, attackiert ein Feld des Gegners. Bei einem Angriff gelten folgende Regeln: - Der Gegner signalisiert, ob sich auf diesem Feld ein Schiff befindet (Treffer) oder ob es Meer ist (Fehlschuss). - Bei einem Treffer darf der attackierende Spieler nochmals schiessen. - Bei einem Fehlschuss ist der andere Spieler dran. - Sobald alle Felder eines Schiffs getroffen worden sind, gilt dieses als versenkt. - Der attackierte Spieler teilt bei einem Treffer mit, ob sein Schiff versenkt ist. 8. Punkt 7 wird solange wiederholt, bis bei einem Spieler alle Schiffe versenkt wurden. Dann ist das Spiel vorbei und der Spieler, welcher noch Schiffe besitzt, hat gewonnen.
Exceptions (Spielabruch)	Beide Spieler selektieren sich gleichzeitig für ein Spiel. Die Verbindung zwischen zwei Spielern wird beendet. Einer der Spieler beendet die Applikation.

3. Systemspezifikation

Für die Anwendung wird das **Model-View-Controller Muster** verwendet:

- View: GUI
- Controller: Engine und Opponent
- Model: Grid

Die Komponente Opponent ist ein Stellvertreter (Proxy) für einen Gegenspieler. Alle Kommandos die an diese Komponente gesendet werden, werden weitergeleitet und via Netzwerk von einem Netzwerkgegenspieler oder von dem Computergegenspieler ausgeführt.

Die einzelnen Komponenten können auf der folgenden Abbildung betrachtet werden:

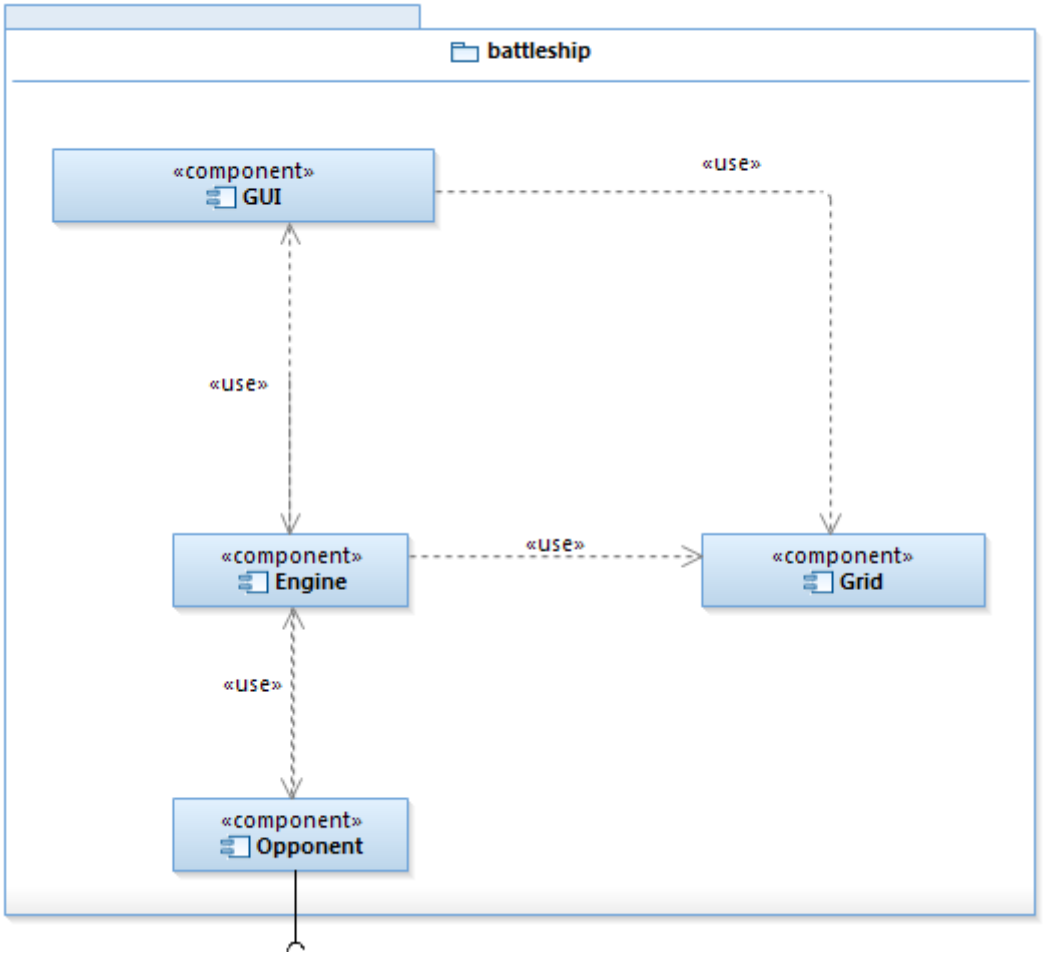

Abbildung 3: Die vier Komponenten der Battleship Application

3.1. Zustände

Währen der Laufzeit der Applikation treten vier Zustände ein:

- Gegenspieler auswählen (SelectingOpponent)
- Spielfeld vorbereiten (PreparingGrid)
- Spielen (Play)
- Spiel beendet (Finished)

Diese 4 Zustände sind im folgenden Diagramm dargestellt. Was in den einzelnen Zuständen passiert, ist im Folgenden beschrieben. Weiter existiert für jeden Zustand auch noch ein Sequenzdiagram.

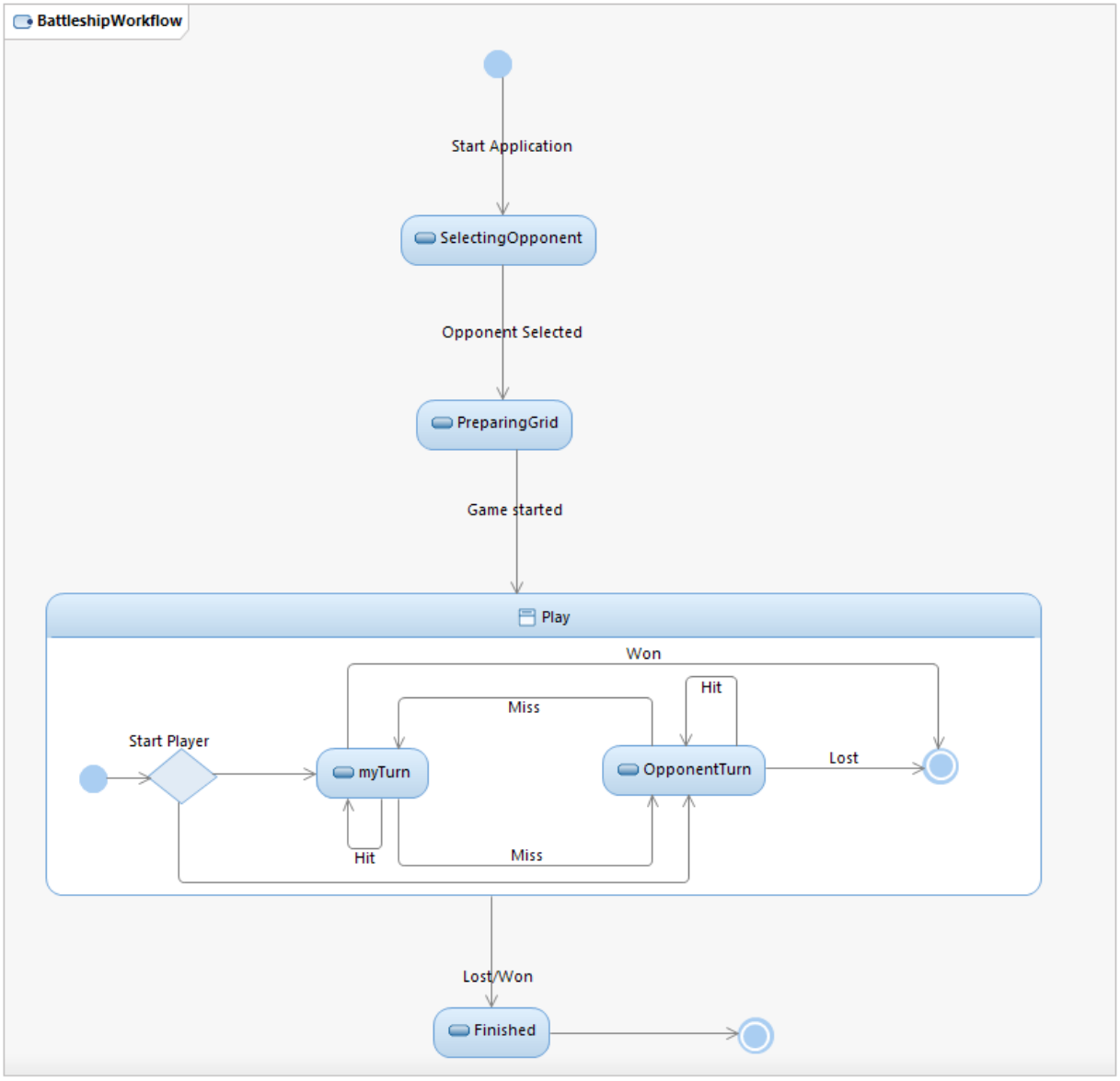

Abbildung 4: Statemaschine der Anwendung

3.1.1. SelectingOpponent

[TODO] Inhalt dieses Kapitels:

- Beschreibung des Zustandes

3.1.2. PreparingGrid

Beide Spieler setzen ihre Schiffe auf ihrem Grid. Sobald ein Spieler alle Schiffe gesetzt hat, wird signalisiert, dass der Spieler bereit ist. Ab diesem Zeitpunkt dürfen die Positionen der Schiffe nicht mehr verändert werden. Sobald beide Spieler bereit sind beginnt das eigentliche Spiel.

3.1.3. Play

[TODO] Inhalt dieses Kapitels:

- Beschreibung des Zustandes

3.1.4. Sequenzdiagramme

Die Abläufe in den Zuständen werden in Sequenzdiagrammen dargestellt. Diese zeigen die Interaktionen zwischen den Komponenten.

SelectingOpponent

[TODO] Inhalt dieses Kapitels:

PreparingGrid

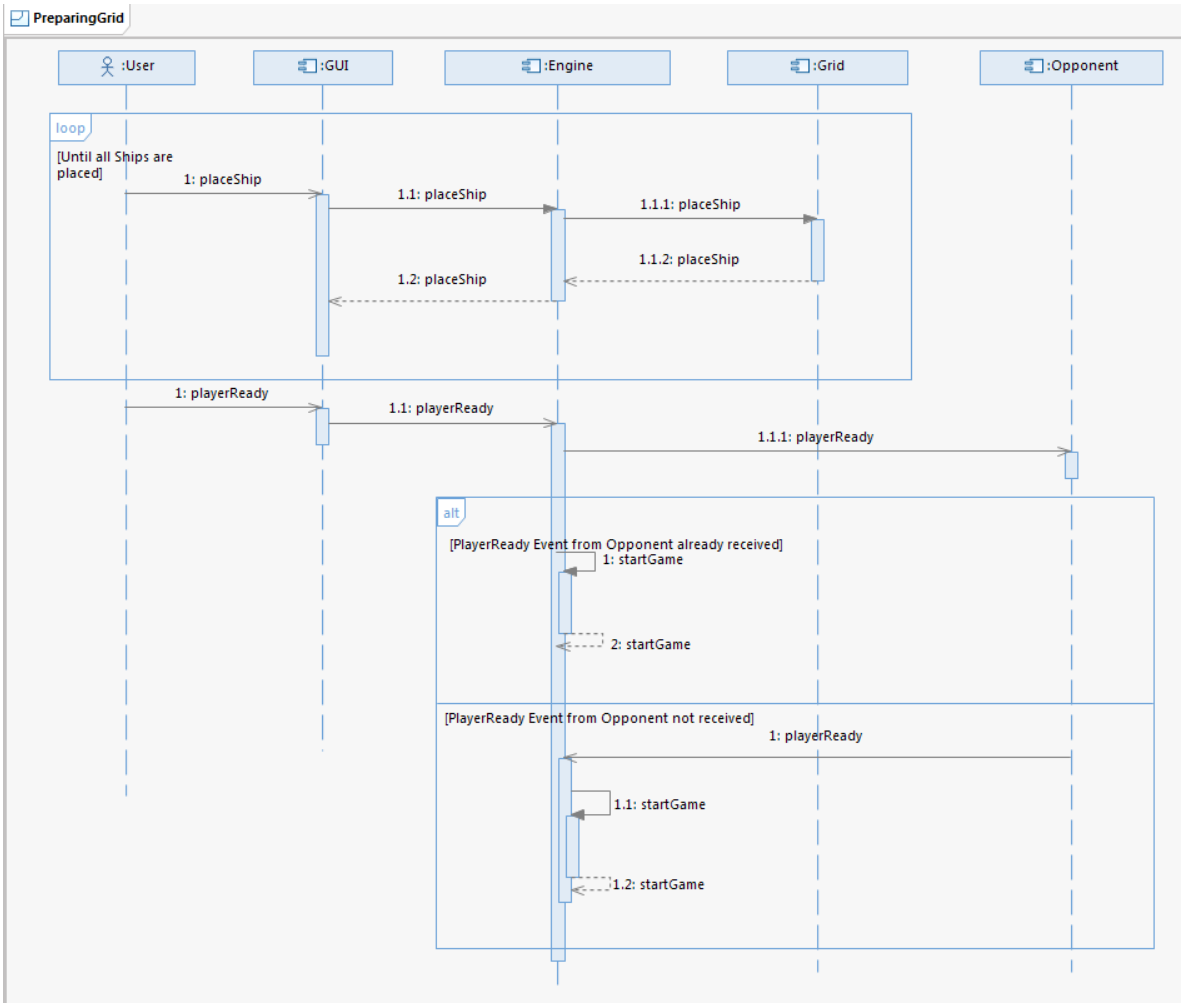

Abbildung 5: Sequenzdiagramm PreparingGrid

SelectingOpponent

[TODO] Inhalt dieses Kapitels:

3.2. Funktionale Sicht

[TODO] Inhalt dieses Kapitels:

- Wie sind die einzelnen Komponenten aufgebaut?
- Was ist die Aufgabe dieser Komponenten?
- Klassendiagramme der Komponenten

3.3. Verteilungs- und Betriebssicht

[TODO] Inhalt dieses Kapitels:

- Wie läuft die Applikation im Betrieb.
- Auf welchen Rechnern läuft die Applikation

3.4. Datensicht

[TODO] Inhalt dieses Kapitels:

- Wie sieht die Applikation aus Datensicht aus?
- Was für Daten werden von der Applikation verwaltet, geteilt, gespeichert?

3.5. Schnittstellen

[TODO] Inhalt dieses Kapitels:

- Was gibt es für Schnittstellen?
- Gibt es applikationsinterne Schnittstellen, was laufen für Daten über diese Schnittstellen?

3.6. Kommunikation über Netzwerk

[TODO] Fragen in diesem Kapitel:

- Wie sieht die Kommunikation über das Netzwerk aus?
- Gibt es ein Protokoll?

3.7. Erweiterungen

[TODO] Fragen in diesem Kapitel

- Gibt es Erweiterungen für das Programm?
- Was wurde nicht implementiert?