



Höhere Fachschule
Südostschweiz

Meine Schule. Meine Zukunft.

Semesterarbeit

Modul Grundlagen Software-Entwicklung

Projektskizze "Tank Wars"

Rade Ilic, Phillip Tuor, Stefan Hutter

16.1102A-2019
NDS HF Applikationsentwicklung
Abgabedatum: 10.01.2020

Fachkorrektor:

Ueli Kunz
ibW Höhere Fachhochschule Südostschweiz
Gürtelstrasse 48, 7000 Chur

1 Projektskizze

Diese Semesterarbeit behandelt die Implementierung einer Java Applikation namens Tank Wars. Die Grundidee und die systematische Vorgehensweise zur Realisierung der Applikation wird in den folgenden Kapiteln der Projektskizze beschrieben.

1.1 Use Case

Der Use Case wird in Abbildung 1 graphisch dargestellt. Dabei wird dieser in drei verschiedene Stufen unterteilt:

1. GUI mit zwei Benutzeroberflächen (Spieleinstellungen und Spielverlauf)
2. Die Applikation, welche den Code beinhaltet
3. Optionale Features, welche möglicherweise implementiert werden

Wird die Applikation gestartet, erscheint die Benutzeroberfläche für die Spieleinstellungen. Der Spieler hat die Möglichkeit die Menge der Panzer und die Spielfelddimension auszuwählen.

- Panzer: Die Grösse eines Panzers beträgt eine Zelle im Spielfeld
- Spielfeld: Die Anzahl der Zellen im Spielfeld wird erstellt, indem der eingegebene Wert mit 2 potenziert wird (z.B. bei Eingabe von 5 hat das Spielfeld 25 Zellen).

Sind die Einstellungen bestätigt, erscheint ein „Play“-Button. Nach Drücken dieses Buttons öffnet sich eine weitere Benutzeroberfläche für den Spielverlauf.

Auf dieser Benutzeroberfläche werden die Spielfelder des Spielers und des Gegners visualisiert. Zuerst muss der Spieler die gewählte Anzahl Panzer auf dem eigenen Spielfeld, auf der linken Seite, platzieren. Die verbleibende Anzahl der zu setzenden Panzer wird in einem Textfeld angezeigt. Sobald alle eigenen Panzer gesetzt sind, erscheint das gegnerische Feld auf der rechten Seite. Der virtuelle Computergegner setzt anschliessend dieselbe Anzahl an Panzern zufällig auf dem gegnerischen Spielfeld. Der Spieler kann jetzt nicht mehr in sein eigenes Feld klicken. Beide Felder haben die Farbe schwarz für die Rahmenlinien und weiss für den Zellinhalt. Die gesetzten Panzer auf dem Spielfeld des Spielers werden grün angezeigt, die des virtuellen Gegners werden nicht angezeigt.

Die Spielvorbereitungen sind abgeschlossen und somit kann das eigentliche Spiel beginnen. Dabei kann der Spieler eine Zelle des gegnerischen Spielfelds anklicken. Geht der Angriff daneben wird die Zelle schwarz eingefärbt, bei einem Treffer rot. Nach jedem ausgeführten Angriff startet der Gegner einen zufälligen Angriff auf das Feld des Spielers. Ebenfalls wird bei einem Fehlschuss die Zelle schwarz und bei einem Treffer rot gefärbt. Das Spiel wird solange weiter gespielt bis einer die komplette Anzahl Panzer des Gegners getroffen hat und somit das

Spiel gewinnt. Die Anzahl der getroffenen und zu treffenden Panzer wird in einem Textfeld angezeigt. Das Spiel kann jederzeit durch Drücken des „Cancel“-Buttons beendet werden.

Optionale Features:

- Multiplayer Modus, um gegen eine reale Person spielen zu können. GUI Erweiterung mit Eingabe der IP-Adresse des Gegners.
- Festlegung der Panzergröße (z.B 2, 3... Zellen)
- Registrierung des Spielers und Spielerdaten Aufzeichnung mithilfe einer Datenbank.

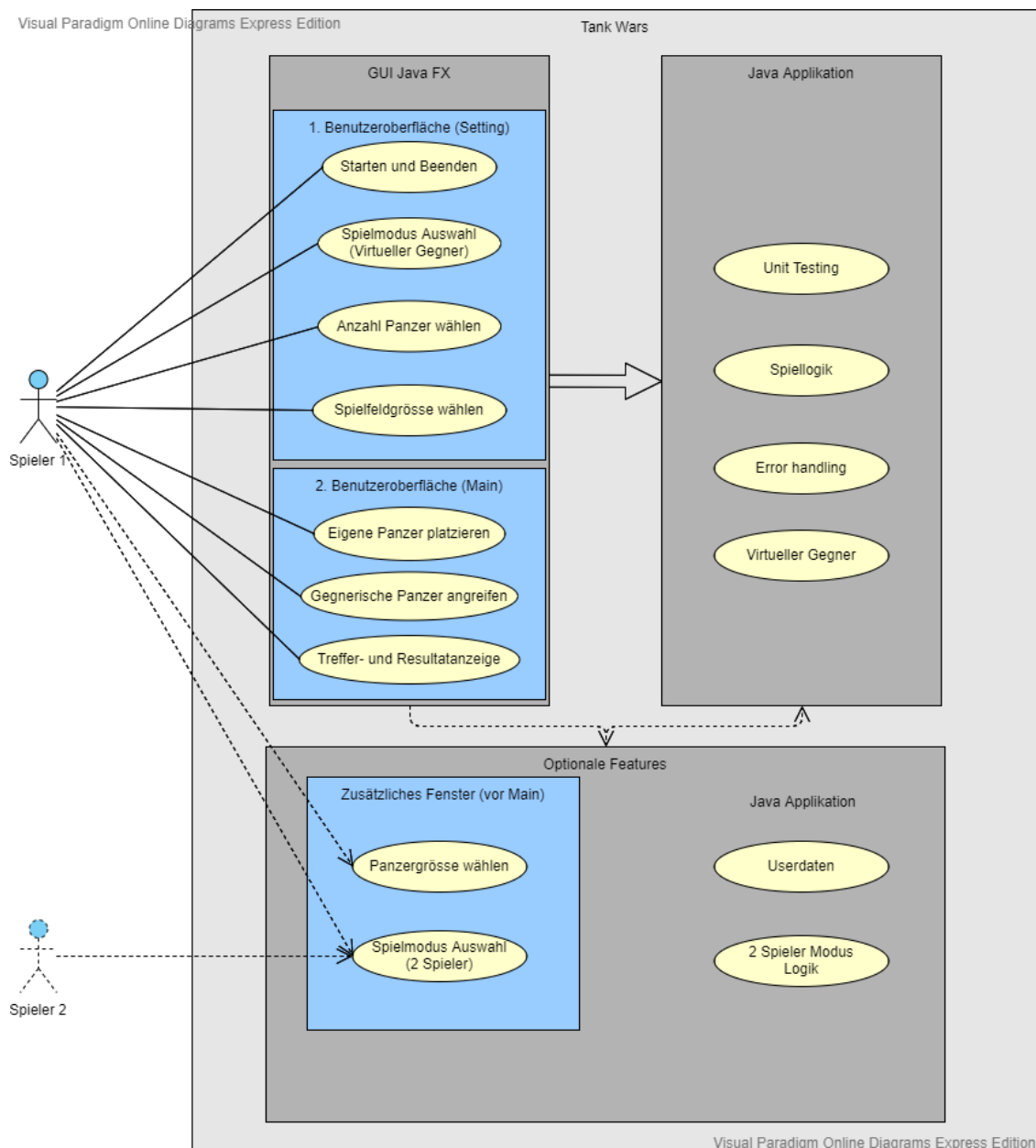


Abbildung 1: Use Case der Applikation Tank Wars

2 UI Mockups und Diagramme

Die Mockups dienen als Visualisierungsvorlage für die Erstellung der Benutzeroberflächen. Dabei werden drei Benutzeroberflächen realisiert:

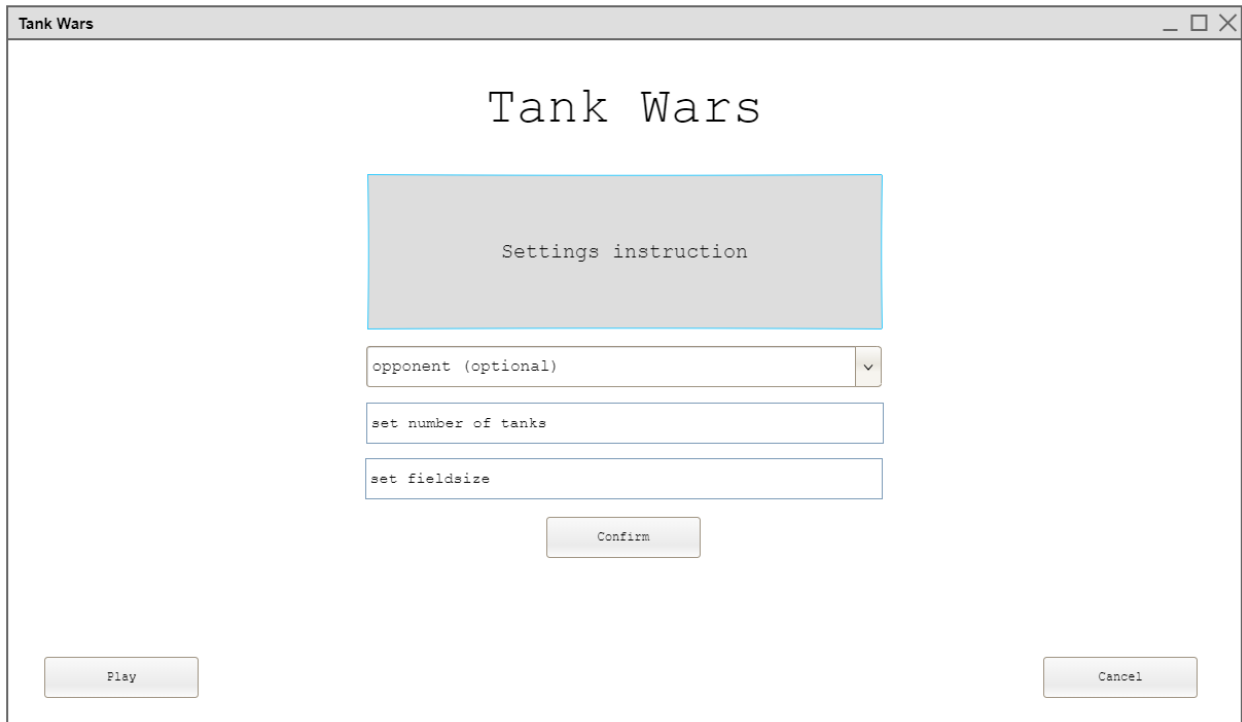


Abbildung 2: Benutzeroberfläche Spieleinstellungen

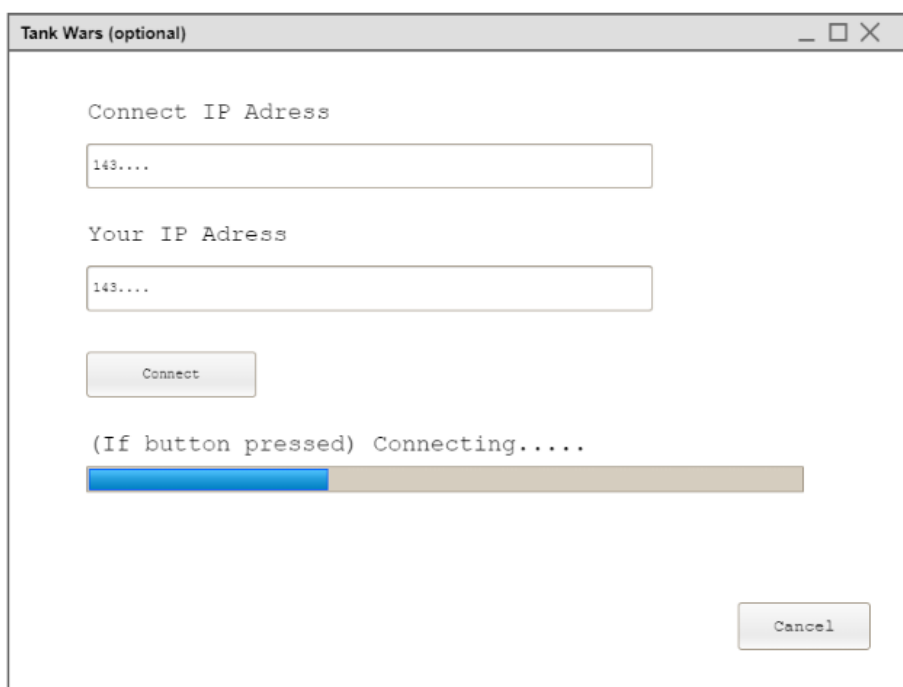


Abbildung 3: Benutzeroberfläche Mehrspieler-Modus (optional)

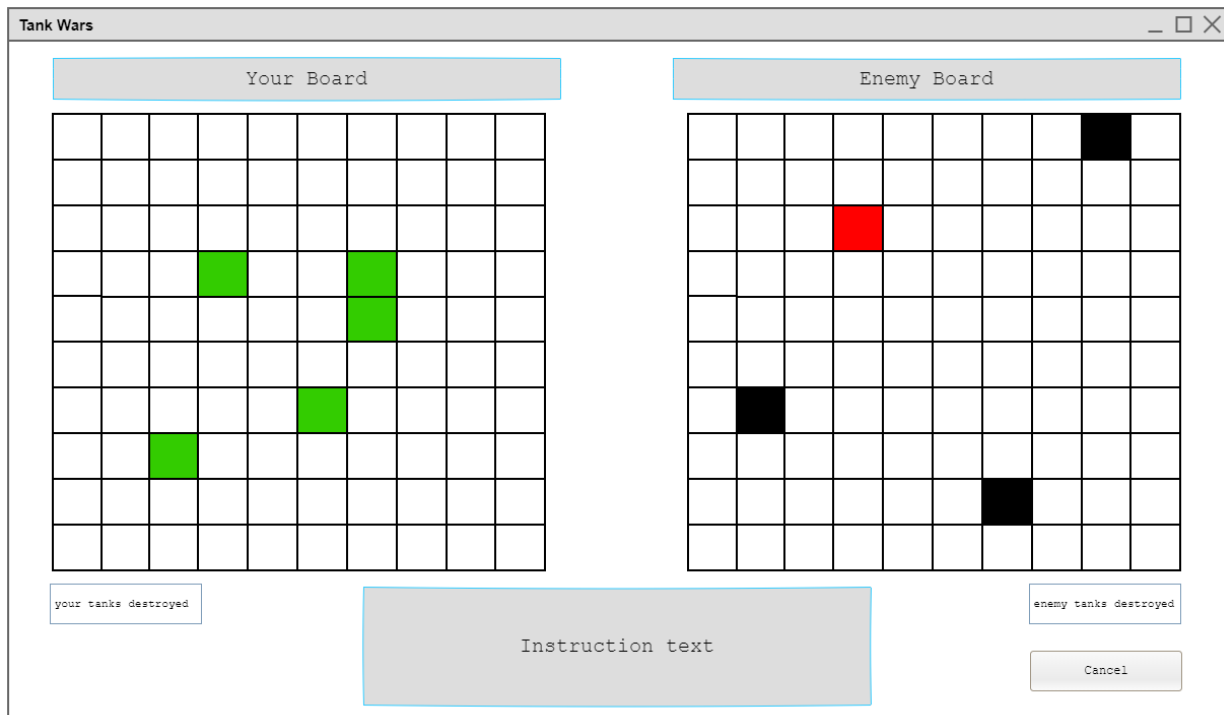


Abbildung 4: Benutzeroberfläche Spielverlauf

Die Projektplanung wird anhand des Gantt-Diagramms realisiert. Parallel zur Erstellung der Projektskizze verläuft die Testphase, welche erste Einblicke in die Applikation liefert. Anschliessend werden in der Entwicklungsphase die User Stories abgearbeitet und zugleich die Dokumentation erstellt.

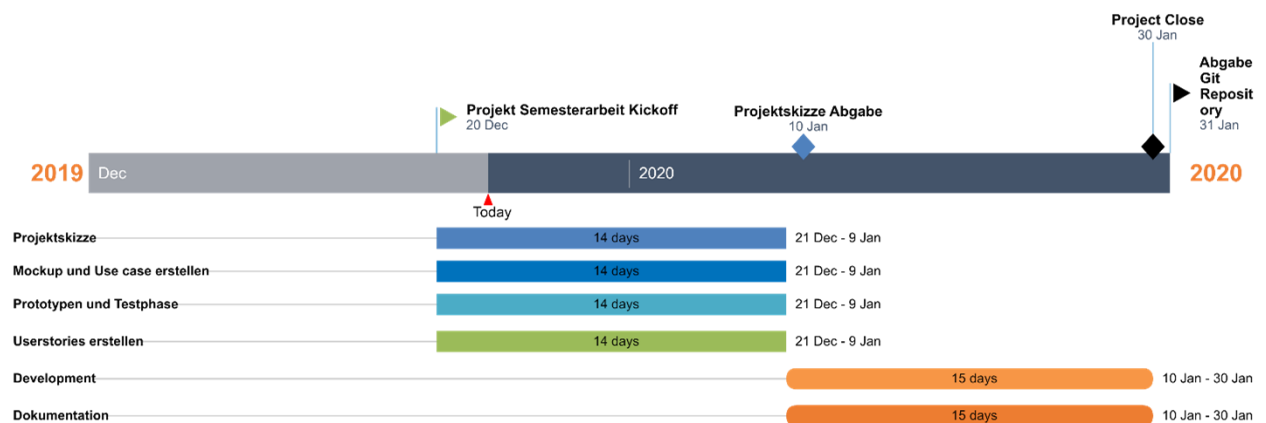


Abbildung 5: Gantt-Diagramm - Zeitplan Semesterarbeit

3 Priorisiertes Backlog

Das Backlog wird in funktionale und nicht funktionale User Stories unterteilt. Der Aufwand der User Stories wurde geschätzt und dementsprechend Story Points vergeben. Ein Story Point entspricht einer Arbeitsstunde.

Die Priorisierung der Einträge erfolgt nach dem MoSCoW Schema. Die Einträge werden folgendermassen klassifiziert:

- **MUST** (unbedingt für die Umsetzung erforderlich)
- **SHOULD** (Umsetzen, wenn alle Must have's erledigt sind)
- **COULD** (kann umgesetzt werden, wenn höherwertige Anforderungen nicht beeinträchtigt werden)
- **WON'T** (wird nicht umgesetzt, evtl. für spätere Umsetzungen interessant)

3.1 Funktionale User Stories

ID	Priorisierung	Beschreibung	Thema/Story	Story Points
1	MUST	Als User kann ich das Game direkt auf einem GUI spielen, wodurch die Bedienung erleichtert wird.	GUI Spielverlauf	15
2	MUST	Als User möchte ich das Game gegen einen Computergegner spielen, um alleine spielen zu können.	Applikation	6
3	MUST	Als User möchte ich die Anzahl Panzer und Feldgrösse selbst bestimmen, um das Game variantenreich zu spielen.	GUI Spieleinstellung	4
4	MUST	Als User möchte ich eine farbliche Darstellung für die ausgeführten Angriffe bzw. die getroffenen Panzer und Fehlschüsse auf dem jeweiligen Spielfeld, um den Spielverlauf besser nachvollziehen zu können.	GUI Spielverlauf	10
5	MUST	Als User möchte ich anhand einer Anzeige erkennen, wer während dem Spielverlauf wie viele Panzer bereits zerstört hat.	GUI Spielverlauf	2

6	COULD	Als User möchte ich die Möglichkeit die Panzergrösse zu wählen, um das Game variantenreicher zu spielen	Applikation	15
7	COULD	Als User möchte ich die Möglichkeit Mehrspieler Modus, um mit meinen Freunden spielen zu können.	Applikation	15

3.2 Nichtfunktionale User Stories

ID	Priorisierung	Beschreibung	Thema/Story	Story Points
8	SHOULD	Als User möchte ich eine intuitiv gestaltete Benutzeroberfläche haben, damit das Game schnell verständlich ist.	GUI allgemein	10
9	COULD	Als User möchte ich während dem Spielen Musik hören, um eine passende Atmosphäre zu schaffen	Applikation	3

4 Weitere relevante Projekthinhalte

GitHub wird als Repository für die gesamte Arbeit verwendet. Darin ist ein README File enthalten, welches die Regeln des Repositories beschreibt. Zusätzlich werden die User Stories im GitHub Repository mithilfe eines Kanban-Boards abgearbeitet. Das Kanban-Board ist unterteilt in die Spalten To Do, In Progress, In Review und Done. Dadurch können wir, als kleines Team, unserem Projekt mehr Struktur geben und somit die Effektivität steigern.