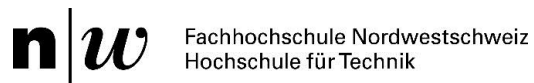


SOFTWARE DOKUMENTATION

Artillery



3. Semester Systemtechnik / 3Sbb

Fachhochschule Nordwestschweiz

Hochschule für Technik

Dozenten:

Prof. Dr. Norbert Hofmann

Dr. Patrick Coray

Studenten:

Joel Koch

Raphael Waltenspül

Final / V1.00.00 / 06.01.2016

I Abstract

Dieses Dokument, beschreibt im Allgemeinen die Software Artillery, deren Aufbau und Elemente. Weiter sind die Testfälle, die bekannten Fehler und die implementierten Anforderungen aus dem Lastenheft beschrieben. Detaillierte Informationen zu den implementierten Klassen und deren Methoden sind aus der Dokumentation des Source Codes zu beziehen. Dieser befindet sich im Anhang dieser Dokumentation.

II Inhaltsverzeichnis

I	Abstract.....	2
II	Inhaltsverzeichnis.....	3
III	Einleitung	5
III.1	<i>Änderungen / Version.....</i>	<i>5</i>
III.2	<i>Abkürzungen</i>	<i>5</i>
III.3	<i>Gültigkeit und Zweck.....</i>	<i>6</i>
III.4	<i>Ablage, Verweise.....</i>	<i>7</i>
III.5	<i>Mitgeltende Unterlagen.....</i>	<i>7</i>
III.6	<i>Verteiler und Freigabe.....</i>	<i>7</i>
IV	Die Software.....	8
IV.1	<i>Ort und Inbetriebnahme.....</i>	<i>8</i>
IV.2	<i>Struktogramm des Main.....</i>	<i>9</i>
IV.3	<i>Verwendete Klassen und deren Parameter und Methoden</i>	<i>12</i>
IV.4	<i>Visuelle Elemente des Programms Artillery</i>	<i>13</i>
IV.5	<i>Dokumentation Testfälle</i>	<i>14</i>
IV.6	<i>Buglist bei der Verison V01.00.00b.....</i>	<i>15</i>
IV.6.1	<i>Kurze Flugbahn, Prio 1 Fehler.....</i>	<i>15</i>
IV.6.2	<i>Alle Spieler gleichzeitig Zerstört, Prio 1 Fehler.....</i>	<i>15</i>
IV.6.3	<i>Beenden des Programms/ Menues, Prio 2 Fehler</i>	<i>15</i>
V	Zielerreichung im Projekt	16
V.1	<i>Must Kriterien.....</i>	<i>16</i>
VI	Systemelemente.....	16
VI.1.1	<i>Spielelemente</i>	<i>16</i>

VI.2	<i>Need Kriterien</i>	17
VI.2.1	Spielelemente	17
VI.3	<i>Nice to Have Kriterien</i>	17
VI.3.1	Systemelemente	17
VI.3.2	Spielelemente	18
VII	Erfüllung der Anforderungen	19
VII.1	<i>Funktionale Anforderungen</i>	19
VII.1.1	Need Funktionen	20
VII.2	<i>Daten Schnittstellen</i>	21
VII.2.1	Need Daten Schnittstellen	21
VII.3	<i>Benutzer Schnittstellen</i>	21
VII.4	<i>Leistungsmerkmale</i>	22
VIII	Anhänge	23
IX	Unterschriften	24

III Einleitung

III.1 Änderungen / Version

Task:	Wer:	Wann:	Bemerkung:
Erstellt V00.00.01	WR	04.01.2016	
Review V00.00.02	JK	05.01.2016	Ok
Anpassungen V01.00.00	WR	05.01.2016	

III.2 Abkürzungen

Abkürzung:	Bedeutung:
JK	Joel Koch
RW	Raphael Waltenspül
Fxy	Produktfunktion Nummer xy
Dxy	Datenschnittstelle Nummer xy
Lxy	Leistungsmerkmal Nummer xy
Bxy	Benutzerschnittsteller Nummer xy
Exy	Testcase Nummer xy
Spielrunde	Entspricht dem Spielverlauf bis zu einem Treffer
Spielzug	Entspricht einem Feuerbefehl
Software	Software bezeichnet die geschriebenen .m Files zur Ausführung in Matlab.

III.3 Gültigkeit und Zweck

Das Dokument dient der Dokumentation des Matlab Projektes und ist für die Software Version 01.00.00b gültig. Zweck dieses Dokumentes ist es, eine exakte Beschreibung der Software „Artillerie“ zu liefern und den Stand gemäss Lastenheft aufzuzeigen und etwaige Bugs zu beschreiben.

III.4 Ablage, Verweise

Die Ablage des gesamten Projektes befindet sich in einem Git- Repository. Berechtigung für das Repository wird allen beteiligten erteilt. Das Projekt unterliegt der „GNU General Public License / Version 3, 29 June 2007“

HTTPS clone URL: <https://github.com/fhnwRaphi/fhnwMatlProject>

III.5 Mitgeltende Unterlagen

Bezeichnung	Autor / Datum	Bemerkung
Skript mds, 2 Modellierungstechniken, S. 8	T. Petermann	Beispiel Schiefer Wurf mit Luftwiderstand
Skript mech, Dynamik des Massenpunktes, S.9	R. Lamprecht	Beispiel Schiefer Wurf
Skript matl,	N. Hofmann P. Coray	gesamtes Skript aus der Vorlesung
Lastenheft V01.00.00	J. Koch R. Waltenspül	Im Anhang befindlich

III.6 Verteiler und Freigabe

Der Verteiler des Projektes ist wie folgt:

- Ablage im Git- Repository
- norbert.hofmann@fhnw.ch
- patrick.coray@fhnw.ch
- joel.koch@students.fhnw.ch
- raphael.waltenspuel@students.fhnw.ch
- Abgabe in Printform

Die Bewertung des Projektes erfolgt durch mindestens einen der folgenden Dozenten:

- Prof. Dr. Norbert Hofmann
- Dr. Patrick Coray

IV Die Software

IV.1 Ort und Inbetriebnahme

Die Software wird in elektronischer Form in folgendem Ordner ausgeliefert:

```
...AblagePfad...\fhnwMatlProject\src00
```

Dieser Ordner muss vor der Ausführung des Programmes in den Matlab Pfad hinzugefügt werden. >> Rechtsklick auf Ordner src00 >> Add to Path >> Folder and Subfolders.

Die Dokumentation befindet sich im Ordner:

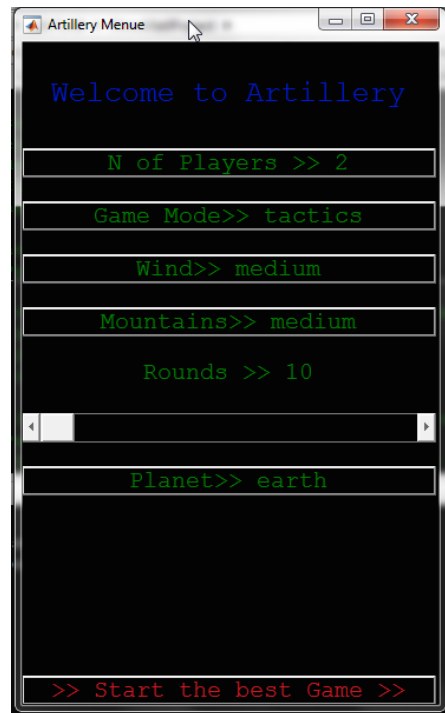
```
...AblagePfad...\fhnwMatlProject\Projektmanagement
```


IV.2 Struktogramm des Main

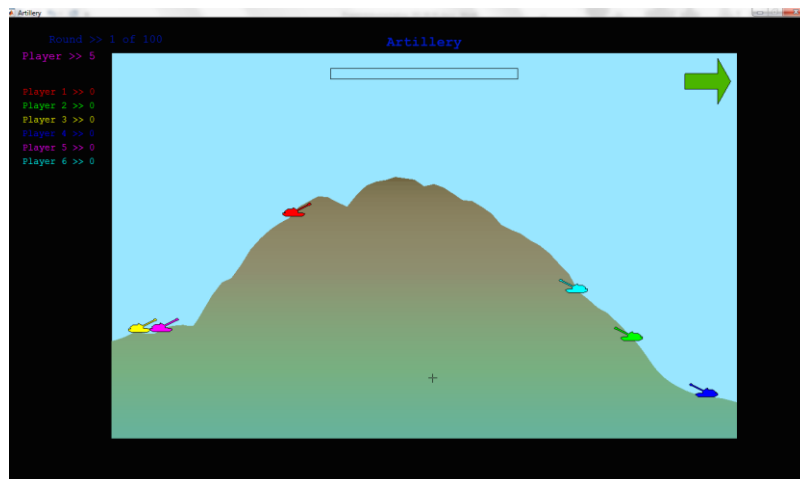
IV.3 Verwendete Klassen und deren Parameter und Methoden

IV.4 Visuelle Elemente des Programms Artillery

Nach dem Start des Programms wird folgendes Menü aufgerufen. Über dieses Menü können alle notwendigen Einstellungen für Spiel getätigt werden.



Nach dem Drücken der Starttaste wird das Hauptprogramm geöffnet.



IV.5 Dokumentation Testfälle

/E10/ Aufstarttest des Programms

Funktioniert einwandfrei

/E20/ Eingabe aller Parameter aus /F30/. (Ohne gegenseitige Abhängigkeiten)

Funktioniert einwandfrei

/E30/ Abbrechen nach Eingabe Parameter.

Das Programm startet mit den eingegebenen Parametern.

Schöner wäre das Programm zu beenden. Implementation in Version 2

/E40/ Fehlhandlung. Neben Buttons klicken, neben Spielfeld klicken.

Es konnten keine Fehler erzeugt werden.

/E50/ Eingabe unrealistischer Parameter.

Es konnten keine Fehler erzeugt werden.

/E60/ Parabel mathematisch nachprüfen. Breite, maximale Höhe.

Prüfung wurde nicht durchgeführt.

/E70/ Eingabe während Schussanimation.

Es konnten keine Fehler erzeugt werden.

/E80/ Punktezähler. Zählt dieser korrekt hoch.

Zählt für alle Anzahl der Spieler korrekt hoch.

/E90/ Abbruch / Neustart des Programmes

Funktioniert bedingt. Programm muss bei vorzeitigem Schließen des Fensters mit ctrl. C beendet werden. Implementation in Version 2

/E100/ Nächstes Level starten.

Funktioniert einwandfrei, nächste Runde wird korrekt hochgezählt

/E110/ Win/Loss Bedingungen testen.

Funktioniert, ausser wenn alle verbleibenden Spieler mit einem Schuss zerstört werden.

IV.6 Buglist bei der Version V01.00.00b

Folgende Bugs sind beim Release der Version V01.00.00b bekannt

IV.6.1 Kurze Flugbahn, Prio 1 Fehler

Dies ist ein Prio 1 Fehler, da der Spielablauf stark beeinträchtigt wird.

Beschreibung des Fehlers: Wird flach geschossen, so wird eine kurze Flugbahn korrekt berechnet. Diese Flugbahn wird aber von der Matlab Funktion comet extrem langsam animiert. Dies kann bis zu einer Minute dauern.

Empfohlener Workaround: Als Möglichkeit soll ausprobiert werden, das Array, falls es unter einer gewissen Länge ist, auf eine vorgegebene Länge zu extrapolieren.

Andernfalls muss eine eigene Methode zur Animation implementiert werden.

IV.6.2 Alle Spieler gleichzeitig zerstört, Prio 1 Fehler

Dies ist ein Prio 1 Fehler, da der Spielablauf unterbrochen wird.

Beschreibung des Fehlers: Wird mit einem Schuss, alle verbleibenden Panzer ausgeschaltet, so entsteht ein Unterbruch des Spielablaufs. Dieser wird in der Methode Player.nextPlayer verursacht.

Empfohlener Workaround: In der Methode Player.nextPlayer soll ein Watchdog implementiert werden. Weiter muss die Punktevergabe so erweitert werden, dass die nächste Runde ohne Vergabe von Punkten gestartet wird.

IV.6.3 Beenden des Programms/ Menues, Prio 2 Fehler

Dies ist ein Prio 2 Fehler, da der Spielablauf nicht beeinträchtigt wird, jedoch für den User umständlich ist

Beschreibung des Fehlers: Wird das Menue oder das Spielfenster geschlossen, wird das Programm nicht beendet.

Empfohlener Workaround: Implementieren des Beendens des Programms

V Zielerreichung im Projekt

Nachfolgend wird beschrieben welche grob umschriebenen Kriterien, welche im Lastenheft in Kapitel 4 definiert sind, implementiert wurden.

V.1 Must Kriterien

V.1.1 Systemelemente

- Zwei Spieler können gegeneinander spielen.
✓ Wurde implementiert
- Die Spieler spielen rundenbasiert an einem Rechner.
✓ Wurde implementiert
- Das Spiel ist 2-dimensional.
✓ Wurde implementiert

V.1.2 Spielelemente

- Die Landschaft wird pro Spielrunde neu berechnet.
✓ Wurde implementiert
- Die Landschaft wird zufällig generiert.
✓ Wurde implementiert
- Der Wind weht und ändert in Abhängigkeit der Zeit die Stärke.
✓ Wurde implementiert, wobei die Runde als Zeiteinheit gilt
- Die Spieler werden zufällig platziert.
✓ Wurde implementiert
- Jeder Spieler hat einen Tank.
✓ Wurde implementiert
- Der Spieler kann die Parameter der Kanone ändern. Diese sind:
 - ✓ Stärke der Treibladung (Skalare Grösse)
 - ✓ Winkel des Kanonenrohrs (Winkel in Grad)
- oder der Spieler kann diese über eine Mauseingabe per Geschicklichkeit steuern.
✓ Wurde implementiert

- Der Spieler gibt den Feuerbefehl
✓ Wurde implementiert
- Das Spiel zeichnet die exakte Flugparabel unter Berücksichtigung des Luftwiderstandes. Dies in einer 2 dimensional Ebene
✓ Wurde implementiert
- Trifft das Geschoss die Umgebung, so bildet sich dort einen Krater.
✓ Wurde implementiert
- Trifft das Geschoss den Gegenspieler so wird dieser zerstört. Das Spiel ist zu ende.
✓ Wurde implementiert

V.2 Need Kriterien

V.2.1 Spielelemente

- Das Spiel geht über mehrere Runden.
✓ Wurde implementiert
- Trifft der Spieler erhält einen Punkt und eine neue Spielrunde beginnt.
✓ Wurde implementiert, wenn alle anderen Spieler zerstört sind.
- Der Spieler der hinten liegt fängt an. In der ersten Runde wird gelost.
✗ Nicht implementiert, es wird immer gelost
- Die Kanonen haben eine Toleranz. Die Schuss- Parabel unterliegt einer Streuung
✗ Nicht implementiert

V.3 Nice to Have Kriterien

V.3.1 Systemelemente

- Das Spiel kann über das Netzwerk / Internet gespielt werden
✗ Nicht implementiert
- Es können mehr als zwei Spieler Teilnehmen.
✓ Wurde implementiert

- Das Spiel kann sowohl Rundenbasiert nacheinander, als auch Rundenbasiert gleichzeitig gespielt werden. (Gleichzeitig heisst, wenn alle Spieler ihren Zug eingegeben haben wird dieser ausgeführt.
✗ Nicht implementiert
- Das Spiel bietet Soundeffekte
✗ Nicht implementiert
- Das Spiel ist 3-dimensional
✗ Nicht implementiert

V.3.2 Spielelemente

- Der Spieler kann verschiedene Geschosse wählen. Diese können sein:
 - A. Grösser Sprengkraft, dafür Schwerer.
 - B. Streumunition
 - C. ...
✗ Nicht implementiert, vorbereitet
- Die Kanone kann sich im Verlauf des Spiels erhitzen, dies erhöht die Ungenauigkeit und kann bis zum Ausfall der Kanone führen.
✗ Nicht implementiert, vorbereitet
- Es können sich Schockeffekte bilden. Trifft ein Geschoss in der unmittelbaren Umgebung so muss der Gegenspieler einen Spielzug aussetzen.
✗ Nicht implementiert, vorbereitet
- Die Spieler können mehrere Tanks haben.
✗ Nicht implementiert, vorbereitet
- Die Tanks haben individuelle Beschädigungen
✗ Nicht implementiert, vorbereitet
- Das Spiel bietet weitere Aktionen wie
 - D. Fahren
 - E. Reparieren
✗ Nicht implementiert, vorbereitet
- Weitere Gadgets sind immer möglich...

VI Erfüllung der Anforderungen

Die detaillierte Implementation der Anforderung gemäss Lastenheft Kapitel 7 ist unten beschrieben. Die erfüllten Anforderungen wurden **Grün** Markiert, die nicht erfüllten Anforderungen wurden **Rot** Markiert. Anmerkungen sind in **Blau**.

VI.1 Funktionale Anforderungen

Nachfolgend sind die funktionalen Anforderungen der Must Elemente beschrieben.

- /F10/ Der User startet das Programm in Matlab über den Befehl `>> ARTILLERY`
- /F20/ Das Programm öffnet das Main Menue.
- /F30/ Im Main Menue können die grundlegenden Parameter eingestellt werden.
Die grundlegenden Parameter sind: Spielmodi (Geschicklichkeit; Taktik); Windstärke (low, medium, high); Einstellung der Berghöhe. (low, medium, high);
- /F40/ Starten des Spiels über einen Button.
- /F50/ Eine zweidimensionale Landschaft wird zufällig generiert. Dabei wird die gewünschte Berghöhe berücksichtigt.
- /F60/ Der Wind weht mit einer Stärke. Der Wind wird im Diagramm als **Zahl** und Richtungsvektor angezeigt. Der die Windstärke hängt von den eingestellten Parametern Windstärke ab. Anstelle der Zahl wurde dem Windshape eine veränderliche Farbe implementier.
- /F70/
- /F80/ Die generierte Landschaft entspricht in der x Achse zwischen 500m und 2000m **ca. 1000m**
- /F90/ Der Panzer jedes Spielers wird zufällig platziert. Dies aber auf der Landschaftsoberfläche und ~~auf einem flachen Plateau~~. Auf das flache Plateau wurde bewusst verzichtet um später ein realistisches fahren zu ermöglichen.
- /F100/ Im Spielmodi Geschicklichkeit, erscheint ein Powerbalken in der Mitte des Feldes.
- /F110/ Im Spielmodi Taktik, erscheint ein Eingabefeld für Treibladungsstärke und Winkel. Als Eingabefeld wurde ein Slider verwendet.

- /F120/Im Spielmodi Geschicklichkeit, kann der Spieler welcher am Zug ist, die Parameter der Kanone ändern. Für den Winkel geschieht dies über die Lage des Mauszeigers in Bezug zum Standort der Kanone. Für die Stärke geschieht dies über die Zeit welche der Spieler die Taste drückt.
- /F130/Im Spielmodi Taktik, kann der Spieler welcher am Zug ist, die Parameter der Kanone ändern. Dies geschieht über die Eingabe der Werte für Treibladung in Joul und Winkel in Grad in einem Eingabefeld. Den Feuerbefehl gibt der Spieler durch drücken einer Taste.
- /F140/Das Spiel zeichnet nun ausgehend von dem schießenden Spieler die exakte Flugparabel des Projektils unter Berücksichtigung des Luftwiderstandes. Es soll dafür eine möglichst exakte Flugbahn gezeichnet werden. Es werden die Methoden aus den Modulen Mechanik und Modellieren dynamischer Systeme verwendet
- /F150/Trifft das Geschoss die Umgebung, so bildet sich dort einen Krater. Dies geschieht durch das ändern des Landschaftsvektor an dieser Koordinate.
- /F160/Trifft das Geschoss den Gegenspieler so wird dieser zerstört, der Spieler welcher trifft, erhält einen Punkt und eine neue Runde beginnt. Im Multiplayer mode erhält der als Leztes verbleibende spieler einen Punkt.

VI.1.1 *Need Funktionen*

Nachfolgend sind die funktionalen Anforderungen der Need Elemente beschrieben.

- /F170/Ergänzend zu Punkt /F30/ Auswahl / Einstellen der Parameter: Spielmodi (Geschicklichkeit; Taktik); Windstärke (low, medium, high), Windböen (low, medium, high); Anzahl Runden (0 bis 100 St.); Einstellung der Berghöhe. (low, medium, high); Weiter wurde die Wahl eines Planeten auf welchem man spielen kann vorbereitet.
- /F180/Ergänzend zu Punkt /F60/ Der Wind weht und ändert in Abhängigkeit der Zeit die Stärke. Der Wind wird im Diagramm als ~~Zahl~~ und Richtungsvektor angezeigt. Der die Windstärke hängt von den eingestellten Parametern Windstärke ab. ~~Die Änderung der Windstärke während einer Runde hängt von den eingestellten Parametern Windböen ab~~ Die Änderung der Windstärke

während einer Runde wird mit einer Standardabweichung mal der zu beginn eingestellten Windstärke neu berechnet.

/F190/ In den Ecken des Spielfeldes werden die Punktestände angezeigt.

/F200/ Befindet sich das Spiel in der ersten Runde, oder herrscht Gleichstand so ermittelt das Programm zufällig welcher Spieler startet.

~~/F210/ Herrscht kein gleichstand, so starten der Spieler der zurückliegt.~~

VI.2 Daten Schnittstellen

Nachfolgend sind die Anforderungen und das Datenmanagement der Must Elemente beschrieben.

/D10/ Das Spiel hat feste Grundeinstellungen.

/D20/ Die Grundeinstellung der Daten sind in jeweiligen Klassen beschrieben. Dies soll das Handling mit grossen Mengen an Spielparametern erleichtern. Zum einen können diese dann einfach als Objekt übergeben werden zum andern können die Parameter einfach über Methoden eingestellt werden.

VI.2.1 *Need Daten Schnittstellen*

Nachfolgend sind die Anforderungen und das Datenmanagement der Need Elemente beschrieben.

~~/D30/ Die Spielzüge werden als Datei gespeichert. (Z. Bsp. .txt;.csv;.xml)~~

VI.3 Benutzer Schnittstellen

/B10/ Das Spiel wird in einer GUI Dargestellt

/B20/ Die Eingabe kann über die Maus erfolgen

~~/B30/ Die Eingabe kann über die Tastatur erfolgen.~~

VI.4 Leistungsmerkmale

/L10/ Das Spiel muss flüssig laufen. Das bedeutet die Kometenflugbahn darf nicht stocken. Pausen in welchen nichts passiert und keine Eingabe möglich ist sind zu vermeiden.

/L20/ Die Berechnung der Parabel soll $< 3s$. dauern. Die Animation des geschosses gehört zum Effekt des Spieles. Die Berechnung der Parabel läuft sehr schnell $< 1s$ ab.

VII Anhänge

- In Papierform: Ausdruck der Softwarecode
- In elektronischer Form Anhang in Ordner srcOO
 - A. Main Function Artillery()
 - B. Klasse Figure
 - C. Klasse Landscape
 - D. Klasse GameParameter
 - E. Klasse GameStates
 - F. Klasse Player
 - G. Klasse FlightPath
- In Papierform: Ausdruck des Lastenheftes
- In elektronischer Form Anhang in Ordner Projektmanagement

VIII Unterschriften

Student

Herr Joel Koch

.....

Ort, Datum

Unterschrift

Student

Herr Raphael Waltenspül

.....

Ort, Datum

Unterschrift