LASTENHEFT

Artillery

# Abstarct

# Inhaltsverzeichniss

[Abstarct 2](#_Toc433821281)

[Inhaltsverzeichniss 3](#_Toc433821282)

[Einleitung 5](#_Toc433821283)

[1.1 Änderungen / Version 5](#_Toc433821284)

[1.2 Abkürzungen 5](#_Toc433821285)

[1.3 Gültigkeit und Zweck 5](#_Toc433821286)

[1.4 Ablage, Verweise 5](#_Toc433821287)

[1.5 Mitgeltende Unterlagen 5](#_Toc433821288)

[1.6 Verteiler und Freigabe 6](#_Toc433821289)

[Ausgangslage 6](#_Toc433821290)

[1.7 Ist-Konzept 6](#_Toc433821291)

[1.8 Projektidee 6](#_Toc433821292)

[1.9 Langfriste Strategie 6](#_Toc433821293)

[Konzept und Rahmenbedingungen 7](#_Toc433821294)

[1.10 Zielsetzung 7](#_Toc433821295)

[1.11 Muss 7](#_Toc433821296)

[1.11.1 Systemelemente 7](#_Toc433821297)

[1.11.2 Spielelemente 7](#_Toc433821298)

[1.12 Nice To Have 7](#_Toc433821299)

[1.12.1 Systemelemente 7](#_Toc433821300)

[1.12.2 Spielelemente 8](#_Toc433821301)

[1.13 Projektziele 8](#_Toc433821302)

[1.14 Bewertung der Zielerreichung 8](#_Toc433821303)

[1.15 Abgrezungskriterien 8](#_Toc433821304)

[Produkteinsatz 9](#_Toc433821305)

[1.16 Benutzer / Zielgruppe 9](#_Toc433821306)

[Produktumgebung 9](#_Toc433821307)

[1.17 Systemvoraussetzung 9](#_Toc433821308)

[1.17.1 Windows 9](#_Toc433821309)

[1.17.2 Mac OS 9](#_Toc433821310)

[1.17.3 Linux 10](#_Toc433821311)

[Anforderungen 11](#_Toc433821312)

[1.18 Funktionale Anforderungen 11](#_Toc433821313)

[1.18.1 Optionale Funktionen 12](#_Toc433821314)

[1.19 Daten Schnittstellen 12](#_Toc433821315)

[1.19.1 Optionale Funktionen 12](#_Toc433821316)

[1.20 Benutzer Schnittstellen 12](#_Toc433821317)

[1.21 Leitungsmerkmale 12](#_Toc433821318)

[1.22 Testfälle 12](#_Toc433821319)

[1.23 Nichtfunktionale Anforderungen Benutzbarkeit 13](#_Toc433821320)

[1.23.1 Design 13](#_Toc433821321)

[1.23.2 Zuverlässigkeit 13](#_Toc433821322)

[1.23.3 Effizienz 13](#_Toc433821323)

[1.23.4 Änderbarkeit; Übertragbarkeit; Wartbarkeit 13](#_Toc433821324)

[Lieferumfang 13](#_Toc433821325)

[Abnahmekriterien 13](#_Toc433821326)

# Einleitung

## Änderungen / Version

|  |  |  |  |
| --- | --- | --- | --- |
| **Task:** | **Wer:** | **Wann:** |  |
| Erstellt | WR | 27.10.2015 |  |
| Review | JK | 28.10.2015 |  |

## Abkürzungen

|  |  |  |
| --- | --- | --- |
| **Abkürzung:** | **Bedeutung:** | **Verweise** |
| JK | Joel Koch |  |
| RW | Raphael Waltenspül |  |
| Fxy | Produktfunktion Nummer xy |  |
| Dxy | Datenschnittstelle Nummer xy |  |
| Lxy | Leitungsmerkmal Nummer xy |  |
| Bxy | Benutzerschnittsteller Nummer xy |  |
| Exy | Testcase Nummer xy |  |
| Spielrunde | Entspricht dem Spielverlauf bis zu einem Treffer |  |
| Spielzug | Entspricht einem Feuerbefehl |  |

## Gültigkeit und Zweck

Das Dokument ist für die Dauer des Projektes, das heisst für die Dauer des Maltab Workshops gültig. Mit der Entlastung des Projektteams erlischt der Anspruch auf weiter Implementierungen gemäss Lastenheft. Zweck dieses Dokumentes ist es eine exakte Beschreibung der Erwartungen an die Software Artillerie zu liefern und den Programmierern den Auftrag über die gewünschten Funktionen zu formulieren. Das Lastenheft dient weiter zur Bewertung der Zielerreichung und gilt als Grundlage zur Entlastung des Projektteams.

## Ablage, Verweise

Die Ablage des gesamten Projektes befindet sich in einem Git- Repository. Berechtigung für das Repository wird allen beteiligten erteilt.

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

## Mitgeltende Unterlagen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Bezeichnung | Autor / Datum | Nummer | Verweis | Bemerkung |
| Skript mds |  |  |  |  |
| Skript mech |  |  |  |  |

## Verteiler und Freigabe

Der Verteiler des Lastenhefts 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

Die Freigabe des lasten Heftes erfolgt von mindestens einem der Dozenten:

* Prof. Dr. Norbert Hofmann
* Dr. Patrick Coray

# Ausgangslage

Ausgangslage ist der Auftrag, im Zuge des Moduls Matlab Workshop eine Software zu entwickeln.

## Ist-Konzept

Zum jetzigen Zeitpunkt besteht keine Software. Die Software wird neu geschrieben.

## Projektidee

Die Idee des Projektes Artillery entstand dadurch, dass nach einer Software gesucht wurde welche in der Komplexität und Menge der Funktionen stark Skalierbar ist, die uns einen Bezug zu anderen Modulen gibt (wie Mechanik oder Modelieren Dynamischer Systeme) und zu guter Letzt auch Spass macht.

## Langfriste Strategie

Die Langfristige Strategie besteht darin Wissen bezüglich dem Einsatz von Matlab zu erwerben und dieses in weitern Modulen einzusetzen und hoffentlich den Erfolg in diesen zu erleichtern.

# Konzept und Rahmenbedingungen

## Zielsetzung

Es wird ein Artilleriesimulator entwickelt. In diesem Simulator müssen die Spieler über die Eingabe von Parametern versuchen sich gegenseitig zu treffen. Dabei findet das Geschehen in einer zufällig generierten Umgebung statt, in welcher auch Parameter wie Wind veränderlich sind. Die Flugparabel wird unter Berücksichtigung der Luftwiderstand berechnet und gezeichnet. Ziel ist es, eine stabil funktionierende Anwendung in Matlab zu schreiben, welche mindestens alle Musskriterien erfüllt.

## Musskriterien

### Systemelemente

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

### Spielelemente

* Die Landschaft wird pro Spielrunde neu berechnet.
* Die Landschaft wird zufällig generiert.
* Der Wind weht und ändert in Abhängigkeit der Zeit die Stärke.
* Die Spieler werden zufällig platziert.
* Jeder Spieler hat einen Tank.
* Der Spieler der hinten liegt fängt an. In der ersten Runde wird gelost.
* Der Spieler kann die Parameter der Kanone ändern. Diese sind:
  1. Stärke der Treibladung (Skalare Grösse)
  2. Winkel des Kanonenrohrs (2- dimensionaler Einsvektor)
* Oder der Spieler kann diese über eine Mauseingabe per Geschicklichkeit steuern.
* Der Spieler gibt den Feuerbefehl
* Das Spiel zeichnet die exakte Flugparabel unter Berücksichtigung des Luftwiderstandes.
* Trifft das Geschoss die Umgebung, so bildet sich dort einen Krater.
* Trifft das Geschoss den Gegenspieler so wird dieser zerstört, der Spieler erhält einen Punkt und eine neue Spielrunde beginnt.

## Nice to Have Kriterien

### Systemelemente

* Das Spiel kann über das Netzwerk / Internet gespielt werden
* Es können mehr als zwei Spieler Teilnehmen.
* Das Spiel kann sowohl Rundenbasiert nacheinander, als auch Rundenbasiert gelichzeitig gespielt werden. (Gleichzeitig heisst, wenn alle Spieler ihren Zug eingegeben haben wird dieser ausgeführt.
* Das Spiel bietet Soundeffekte
* Das Spiel ist 3-dimensional

### Spielelemente

* Die Kanonen haben eine Toleranz. Die Schuss Parabel unterliegt einer Streuung
* Der Spieler kann verschiedene Geschosse wählen. Diese können sein:
  1. Grösser Sprengkraft, dafür Schwerer.
  2. Streumunition
* Die Kanone kann sich im Verlauf des Spiels erhitzen, dies erhöht die Ungenauigkeit und kann bis zum Ausfall der Kanone führen.
* Es können sich Schockeffekte Bilden. Trifft ein Geschoss in der unmittelbaren Umgebung so muss der Gegenspieler ein Spielzug aussetzen.
* Die Spieler können mehrere Tanks haben.
* Die Tanks haben individuelle Beschädigungen
* Das Spiel bietet weiter Aktionen wie
  1. Fahren
  2. Reparieren
* 3D Plot

## Bewertung der Zielerreichung

Die Bewertung der Zielerreichung erfolgt durch die Dozenten:

* Prof. Dr. Norbert Hofmann
* Dr. Patrick Coray

Als Grundlage zur Bewertung gilt die Liste der Funktionen und der Umsetzung dieser.

## Abgrezungskriterien

Das Programm wird

# Produkteinsatz

Das Programm ist zu Lehrzwecken im Zuge des Matlab Worksohop gedacht. Der stetig steigende Schwierigkeitsgrad im Aufbau der Funktionen soll uns erlauben unsere Fähigkeiten kontinuierlich zu steigern, anzuwenden und frühe Erfolge zu erzielen. Weiter soll es stets ein Ansporn sein möglichst viele schwiereige Nice to Have Funktionen zu implementieren.

Das fertige Produkt soll dann als Lückenfüller in nicht allzu anspruchsvollen Vorlesungen[[1]](#footnote-1) dienen.

## Benutzer / Zielgruppe

Die Benutzer sind primär die Dozenten welche die Software testen werden und wir selber. Weiter soll das Programm dem Studiengang 3Sbb zur verfügung gestellt werden.

# Produktumgebung

## Systemvoraussetzung

### Windows

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operating Systems** | **Processors** | **Disk Space** | **RAM** | **Graphics** |
| Windows 10  Windows 8.1  Windows 8  Windows 7 Service Pack 1  Windows Server 2012  Windows Server 2008 R2 Service Pack 1  Windows Server 2008 Service Pack 2 | Any Intel or AMD x86 processor supporting SSE2 instruction set\* | 1 GB for MATLAB only, 3–4 GB for a typical installation | 2 GB;  With Simulink Coder, 4 GB is recommended | No specific graphics card is required. Hardware accelerated graphics card supporting OpenGL 3.3 with 1GB GPU memory recommended. |

### Mac OS

Mac Os Systeme ohne Gewähr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operating Systems** | **Processors** | **Disk Space** | **RAM** | **Graphics** |
| Mac OS X 10.11 (El Capitan)  Mac OS X 10.10 (Yosemite)  Mac OS X 10.9.5 (Mavericks) | All Intel-based Macs with an Intel Core 2 or later | 1 GB for MATLAB only, 3–4 GB for a typical installation | 2 GB;  With Simulink Coder, 4 GB is recommended | No specific graphics card is required. Hardware accelerated graphics card supporting OpenGL 3.3 with 1GB GPU memory recommended. |

### Linux

Linux Systeme ohne Gewähr

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Operating Systems** | **Processors** | **Disk Space** | **RAM** | **Graphics** |
| Qualified distributions\*:  Ubuntu 14.04 LTS and 14.10  Red Hat Enterprise Linux 6 and 7 (Red Hat 6.6 or higher is recommended)  SUSE Linux Enterprise Desktop 11.3+  Debian 7.x | Any Intel or AMD x86 processor supporting SSE2 instruction set\*\* | 1 GB for MATLAB only,  3–4 GB for a typical installation | 2 GB;  With Simulink Coder, 4 GB is recommended | No specific graphics card is required. Hardware accelerated graphics card supporting OpenGL 3.3 with 1GB GPU memory recommended.  Use of vendor-supplied proprietary drivers is strongly recommended. |

# Anforderungen

## Funktionale Anforderungen

Nachfolgend sin die Funktionalen Anforderungen der Must Elemente beschrieben.

1. Starten des Programmes in Matlab über den Befehl >> ARTILLERY
2. Das Programm öffnet das Main Menue.
3. Im Main Munue können die Grundlegenden Parameter eingestellt werden.
4. 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);
5. Starten des Spiels über einen Button.
6. Eine zweidimensionale Landschaft wird zufällig generiert, Dabei wird die gewünschte Berghöhe berücksichtigt.
7. 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
8. Die Generierte Landschaft entspricht in der x Achse zwischen 500m und 2000m
9. Der Panzer jedes Spielers wird zufällig platziert. Dies aber auf der Landschaftoberfälche und an einem Abschnitt welcher keine Steigung >1 hat.
10. In den Ecken des Spielfeldes werden die Punktestände angezeigt.
11. Im Spielmodi Geschicklichkeit, erscheint ein Powerbalken in der Mitte des Feldes.
12. Im Spielmodi Taktik, erscheint ein Eingabefeld für Treibladungsstärke und Winkel.
13. Befindet sich das Spiel in der ersten Runde, oder herrscht gleichstand so ermittelt das Programm zufällig welcher Spieler startet.
14. Herrscht kein gleichstand, so startent der Spieler der zurückliegt.
15. 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.
16. 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.
17. 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
18. Trifft das Geschoss die Umgebung, so bildet sich dort einen Krater. Dies geschit durch das ändern des Landschaftsvektor an dieser Koordinate.
19. Trifft das Geschoss den Gegenspieler so wird dieser zerstört, der Spieler welcher trifft erhält einen Punkt und eine neue Runde beginnt.

### Optionale Funktionen

1. Loren Issum Dolor

## Daten Schnittstellen

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

1. Die Grundeinstellung der Daten sind in jeweiligen Klassen beschrieben. Dies soll das Handling mit grossen Mengen an Spielparametern erleichtern. Zum einen önnen diese dann einfach als Objekt übergeben werden zum andern können die Parameter einafch über Methoden eingestellt werden.
2. Die Spielzüge werden als Datei gespeichert. (Z. Bsp. .txt;.csv;.xml)

### Optionale Funktionen

1. Lorm Ipsum Dolor

## Benutzer Schnittstellen

1. Das Spiel wird in einer GUI Dargestellt

## Leitungsmerkmale

1. Das Spiel muss flüssig laufen
2. Die Berechnung der Parabel soll < 1s. dauern.

## Testfälle

1. Ausftarttest des Programs
2. Eingabe aller Parameter
3. Abbrechen nach eingabe Parameter.
4. Fehlhandlung durch neben Buttons cklicken
5. Eingabe unrealistischer Parameter
6. Parabel mathematisch nachprüfen. Weite maximale Höhe.
7. Punktezähler
8. Abbruch / Neustart des Programmes
9. Nächstes Level starten.
10. Win/Loss Bedingungen

## Nichtfunktionale Anforderungen Benutzbarkeit

### Design

Das Programm soll möglichst ansprechend gestaltet sein. Dabei ist speziell auf einem Benutzerführende resp. selbsterklärende Interface zu achten. Die Grafik des Spielfeld muss erkennen lassen was mit den einzelnen Elementen wie beispielsweise einem Panzer gemeint ist.

### Zuverlässigkeit

Das Programm soll stabil Laufen. Abstürze dürfen durch normals Handenln nicht verusracht werden.

### Effizienz

Die verwendetet rechner Leistung soll durch effizientes programmieren möglichst tief gehalten werden. (Wo möglich und sinnvoll und mit unserem Wissen zu bewältigen)

### Änderbarkeit; Übertragbarkeit; Wartbarkeit

Die Software muss gut dokumentiert sein. Die Software muss jederzeit durch dritte gelesen und erweitert werden können.

Die Mathematischen Berechnungen müssen in der Dokumentation nachvollzogen werden können. (Bsp. Formeln in Latex)

Zur Versionierung soll ein git repository verwendet werden.

# Lieferumfang

Der Lieferumfang beinhaltet die Software inklusiver der Dokumentation.

# Abnahmekriterien

1. Damit ist nicht das Modul matl gemeint. [↑](#footnote-ref-1)