**Softwareprojekt 2. Semester**

****

**Projektname:** Softwareprojekt: IS-Hunter

**Modul:** Softwaredesign und Softwareprojekte BTE5052d

|  |
| --- |
|  |

**Studierende:** Daniel Schären, Nicolas Takagawa und Ron Buntschu

**Dozent:** Dr. Lorenz Müller

**Institution:** Berner Fachhochschule Technik und Informatik

**Studiengang:** Elektro- und Telekommunikation

**Klasse:** E1d

**Semester:** Zwei

Inhalt

[1. Einleitung 2](#_Toc453063008)

[2. Planung 2](#_Toc453063009)

[2.1 Idee 2](#_Toc453063010)

[2.2 Spielidee 3](#_Toc453063011)

[2.2.1 Pseudocode 3](#_Toc453063012)

[2.2.2 3](#_Toc453063013)

[X.X Anhang 4](#_Toc453063014)

[X.1 Lastenheft 4](#_Toc453063015)

[X.2 Pflichtenheft 4](#_Toc453063016)

# 1. Einleitung

Das Ziel des Moduls Softwaredesign und Softwareprojekte war es, in einer Programmiersprache nach Wahl, ein Spiel zu programmieren. Die Entwicklungsumgebung und das Spiel konnten ebenfalls frei gewählt werden. Die Zeitspanne zog sich über die 16 Semesterwochen wobei etwa die erste Hälfte zur Vorbereitung und die zweite Hälfte zur Realisierung des Projektes diente.

Wir erhielten eine Einführung in C++, damit wir das Siel in dieser Programmiersprache realisieren können.

# 2. Planung

## 2.1 Idee

Bei der Suche nach einem passenden spiel trugen wir unsere Ideen zusammen. In eine engere Auswahl kamen ein an „FLAPPY BIRD“ angelehntes Spiel namens „Flying Hirsch“ und das an Moorhuhn angelehnte „IS-HUNTER“. Wir haben uns schlussendlich für letzteres entschieden, da es ein gewagtes und aktuelles Thema verkörpert. Wir möchten uns allerdings keineswegs über die Miesere im Nahen Osten lustig machen.

Es handelt sich um ein 2D-Shooter Game. Wir haben die Idee sehr allgemein im Lastenheft beschrieben. (Siehe Anhang)  
Das Spiel unterscheidet sich doch erheblich von einem Gewöhnlichen Moorhuhn und ist sehr gut erweiterbar. Ein Praktischer Punkt, da wir im Voraus nur schlecht einschätzen konnten, wie viele Punkte wir letztendlich realisieren können.

## 2.2 Spielidee

Die ersten Ideen des Ablaufs der Spiels haben wir im Lasten und Pflichtenheft erfasst. (Siehe Anhang)

Um einen ersten Strukturierten Ablauf des Spiels Aufzuzeigen, schrieben wir einen Pseudocode und erstellten verschiedene Diagramme. Die Funktion der Pseudocodes stimmt teilweise nicht komplett mit den Diagrammen überein. Dies liegt daran, dass der Code lange vor den Diagrammen entwickelt wurde und sich in der Zeit die Spielidee wieder verändert hat.

### 2.2.1 Pseudocode

1. Start des Programms

If(ctrl r){🡪2.;}

1. Menue; Hintergrund normal, Fenster mit: Play; Quit; Score

If(Play){🡪3.;}

If(Quit){🡪1.;}

1. Hintergrund mit Deckungen; Spieler nicht Verletzbar; nach t erscheinen Gegner hinter Deckungen

If(Taste aus Deckung){🡪4.;}

1. Fadenkreuz erscheint in Mitte von Hintergrund, mit Maus bewegbar;

If(Taste für in Deckung){🡪3.;}

If(Click){Schuss in Mitte von Fadenkreuz; Schussgeräusch1;}

If(Click & Fadenkreuz auf Gegner){Schuss in Mitte von Fadenkreuz;

Gegner verschwindet;

Score++;}

If(t vergeht & GegnerPistole nicht getroffen){GegnerPistole schiesst;

Schussgeräusch2;

Fadenkreuz blinkt 1x;

Leben --;

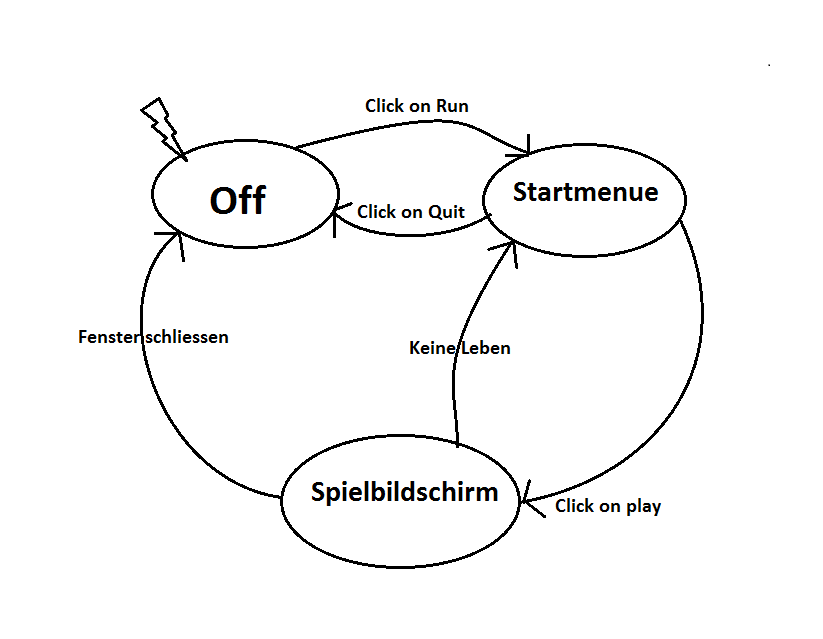
If(Leben==0){🡪2.;}}

If(t vergeht & GegnerSpreng nicht getroffen){Gegner bewegt sich nach vorne;}

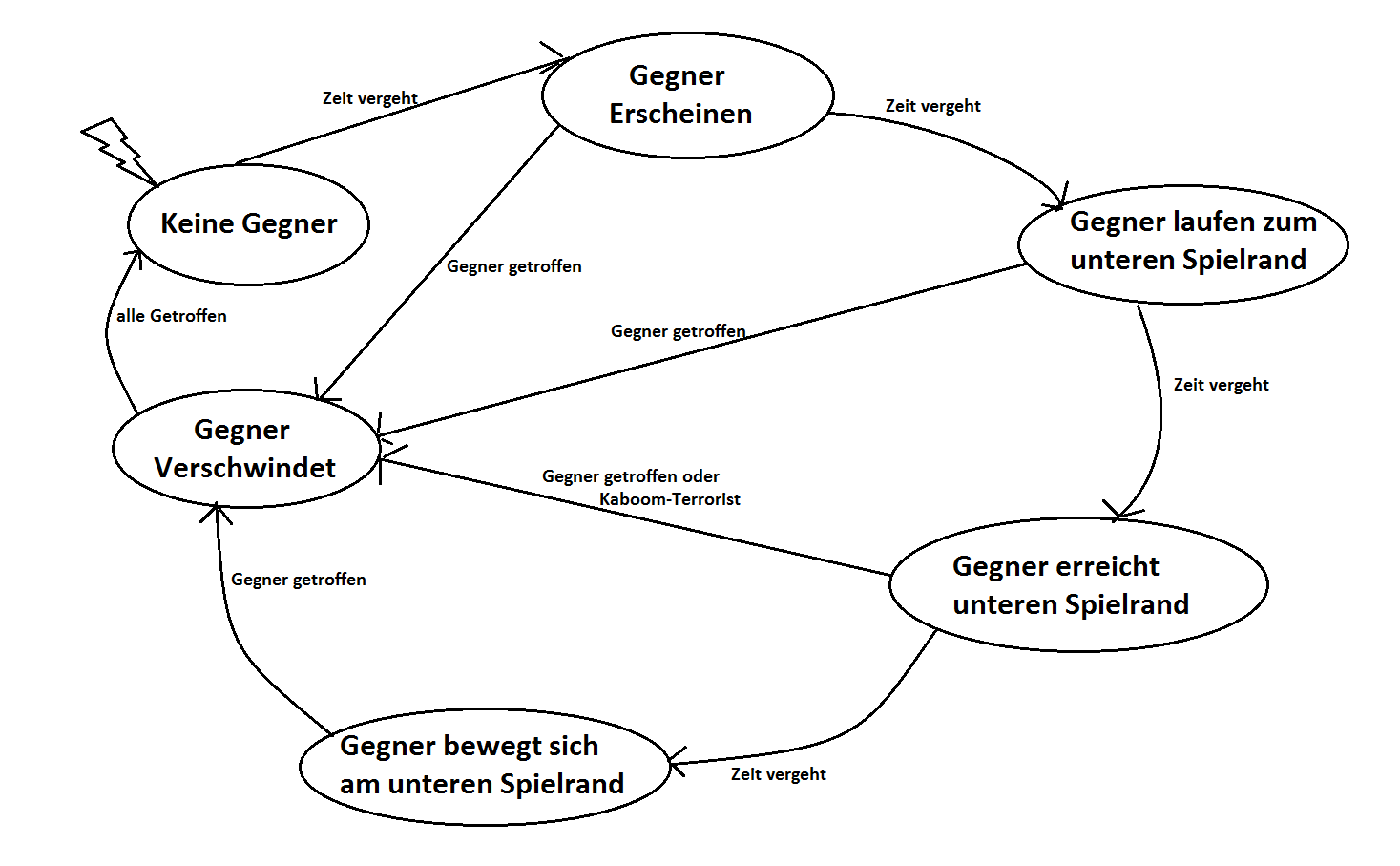
If((Abstand klein & Gegner getroffen)|(Abstand sehr klein){  
 Geräusch „Allahuakbar“;  
 Geräusch „Explosion“;  
 Leben--;  
 If(Leben==0){🡪2.;}}

### 2.2.2 State-Event Diagramm

Allgemeines Diagramm:



State Spielbildschirm:



## 2.3 Entwicklungsumgebung

Als Entwicklungsumgebung wählten wir den Qt-Creator. Einerseits weil er auf Mac, Linux und Windows funktioniert, andererseits weil wir bereits im ersten Semester alle C Projekte im Qt realisierten. Mittels Qt-Creator gibt es eine grosse Anzahl von möglichkeiten, ein solches Softwareprojekt zu realisieren. Wir entschieden uns für QML da wir auch im Unterricht eine kurze Einführung dazu erhielten. Einen grossen Teil des nötigen Wissens haben wir uns aus dem Internet und durch ausprobieren angeeignet.

# X.X Anhang

## X.1 Lastenheft

**Lastenheft**

Spielname: IS-Hunter

Nutzen:

Ein 2D-Shooterspiel, indem dem Benutzer ermöglicht wird auf IS-Karikaturen zu feuern.

Das Ziel besteht darin, aus der Deckung hervorkommende Terroristen zu erschiessen, ohne dabei zu sterben.

Es dient der Vergnügung.

Spielerlebnis:

In Anfangszustand befindet sich der Spieler in Deckung.

Durch das Betätigen einer Taste kann sich der Spieler aus der Deckung bewegen. Erst dann ist es möglich, mit der Maus das Fadenkreuz zu bewegen und per Mausklick zu schiessen.

Es existiert mehrere Kategorien von Feinden, die verschiedene Angriffseigenschaften besitzen.

Es existiert ein Spielhintergrund (Landschaft).

## X.2 Pflichtenheft

**Pflichtenheft**

Gruppe: Nicolas Takagawa, Daniel Schären, Ron Buntschu

**1. Zielbestimmung**

Das Spiel, IS-Hunter, dient der Unterhaltung des Benutzers. Er soll beim Spielen Spass haben, lachen können und den politischen, ironischen Hintergrund des Spiels sehen.

Musskriterien:

IS-Hunter ist ein 2D-Shooterspiel, indem dem Benutzer ermöglicht wird, auf IS-Karikaturen zu feuern.

Das Ziel besteht darin, aus der Deckung hervorkommende Terroristen zu erschiessen, ohne dabei zu sterben.

In Anfangszustand befindet sich der Spieler in Deckung.

Durch das Betätigen einer Taste kann sich der Spieler aus der Deckung bewegen. Erst dann ist es möglich, mit der Maus das Fadenkreuz zu bewegen und per Mausklick zu schiessen.

Es existiert mehrere Kategorien von Feinden, die verschiedene Angriffseigenschaften besitzen.

Wunschkriterien:

Es soll zwei Kategorien von Terroristen geben: -AK-Terrorist  
 -Kaboom-Terrorist

AK-Terroristen sollten aus der Deckung schiessen, wenn sie dies tun verliert man ein Leben.   
Kaboom-Terroristen tragen eine Bombe, bewegen sich nach vorne und wenn man sie zu spät trifft, detonieren sie, man verliert ein Leben.

Es soll verschiedene Hintergrundlandschaften geben, welche man im Menü auswählen kann.

**2. Produkt-Einsatz**

Anwendungsbereich:  
Das Spiel soll man auf Windows spielen können. Da wir Qt als Entwicklungsumgebung benutzen sollte die Anwendung auch auf anderen Plattformen funktionieren.

Das Spiel wird auf einem Fenster auf dem Bildschirm ausgegeben. Es ist mit der Maus und Tastatur zu bedienen. Die Auftraggeber (Entwickler) orientieren sich an dem Spiel Chicken-Shooter. Ausserdem haben manche Entwickler bereits mit Objektorientierten Hochsprachen gearbeitet.

Zielgruppe:  
Jugendliche und Erwachsene. Da im Spiel Waffengewalt und Gewaltverherrlichung vorkommen, sollte es eine Altersfreigabe geben. Wir denken diese sollte bei sechzehn Jahren liegen.

Das Spiel soll lokal auf dem PC, als Applikation, zur Verfügung stehen.

Betriebsbedingungen:  
Das Spiel sollte auf sämtlichen moderneren Betriebssystemen von Windows funktionieren.  
Die Altersbeschränkung welche beschrieben wurde müsste, im Falle einer Freigabe des Spiels für die Allgemeinheit, mit dem Gesetz vereinbart werden.

**3. Produktfunktionen**

Beim Spielstart soll ein Startmenü erscheinen, wo das Land und Spielstart ausgewählt werden kann.

Die Spielfigur soll mittels Taste aus der Deckung kommen und in Deckung gehen.  
Wenn sich die Figur nicht in der Deckung befindet, soll ein Gewehrlauf sichtbar sein, welcher sich mit der Mausbewegung mitbewegt, bez. auf die Maus zeigt.

Auch mit der Maus wird ein Fadenkreuz auf der Spieloberfläche bewegt. Das Fadenkreuz soll erscheinen, sobald die Figur aus der Deckung kommt. Es kann auf dem gesamten Spielfeld bewegt werden. Per Mausklick soll ein Schuss in die Mitte des Fadenkreuzes gefeuert werden, welcher akustisch von einem Knall begleitet wird.

Der Hintergrund soll wenn möglich gewählt werden können.

Auf dem Hintergrund sollen sich verschiedene Deckungen befinden. Aus diesen Deckungen werden dann die Gegner hervorkommen und Angreifen.   
Die schiessenden Gegner sollen, nach einer gewissen Zeit, mit ihren Gewehren losschiessen, auch bei ihnen sind Schussgeräusche zu hören. Wenn sie Treffen wird dies ebenfalls durch ein Geräusch signalisiert und man verliert ein Leben. Falls sie getroffen werden verschwinden sie hinter der Deckung.  
Die Gegner mit einer Bombe laufen von der Deckung in Richtung des Spielers. Wenn sie Getroffen werden Sprengen sie sich in die Luft. Wenn sie nahe beim Spieler sind ebenfalls. Werden sie erst getroffen wenn sie nahe beim Spieler sind, verliert man ein leben.  
Wenn man eine längere Zeit nicht getroffen wird, bekommt man verlorene Leben wieder dazu.

Hat man Sämtliche Leben verloren ist das Spiel vorbei.