



**HOCHSCHULE TRIER**

Trier University of Applied Sciences

**Informatik - Computer Science**

---

Entwicklung eines rundenbasierten Strategiespiels

Dokumentation

NAMEN PLACEHOLDER

Interdisziplinäres Teamprojekt

Betreuer: Prof. Dr. Linda Breitlauch, Prof. Dr. Christof Rezk-Salama

Trier, DATUM PLACEHOLDER

---

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b> .....	<b>1</b>
<b>2</b>	<b>Konzeptionierung</b> .....	<b>2</b>
<b>3</b>	<b>Squad Builder</b> .....	<b>3</b>
<b>4</b>	<b>Levelaufbau</b> .....	<b>4</b>
<b>5</b>	<b>Manager Objekt</b> .....	<b>5</b>
	5.1 Manager System .....	5
	5.2 Shooting System .....	5
	5.3 Inventory System & Inventory Component .....	5
	5.4 Player Assistance System .....	6
	5.5 Ability System .....	6
	5.6 Health System .....	6
<b>6</b>	<b>Spieler</b> .....	<b>7</b>
	6.1 Player Component .....	7
	6.2 Input Component .....	7
<b>7</b>	<b>Spielfiguren</b> .....	<b>8</b>
	7.1 Movement System .....	8
	7.2 Attribute Component .....	8
<b>8</b>	<b>Pathfinding</b> .....	<b>9</b>
<b>9</b>	<b>Kamera</b> .....	<b>10</b>
<b>10</b>	<b>User-Interface</b> .....	<b>11</b>
	10.1 Selektierte Spielfigur .....	11
	10.2 Action-Points Leiste .....	11
	10.3 Dynamische Ability-Icons .....	11

---

<b>11 3D Modelling</b> .....	12
11.1 Figuren.....	12
11.2 Waffen .....	12
11.3 Equipment .....	12
<b>12 Animationen</b> .....	13
12.1 Motion Capture Aufnahmen.....	13
12.2 Einbindung der Animationen .....	13
<b>13 Sounds</b> .....	14
<b>14 Effekte</b> .....	15
<b>15 Status quo</b> .....	16
<b>16 Fazit</b> .....	17

## Einleitung

## Konzeptionierung

## Squad Builder



## Manager Objekt

Um die verschiedenen Systeme, die für den korrekten Ablauf der Spielzüge und allgemein spielregeltechnischen Abläufe zu handeln, wurde ein Spielobjekt, das als Manager bezeichnet wird, erstellt. Im folgenden Kapitel wird auf die einzelnen Skripte die an diesem Manager Objekt hängen genauer eingegangen.

### 5.1 Manager System

Das Manager System ist für den korrekten Ablauf der einzelnen Züge zuständig. Es zählt die Runden hoch, stellt sicher, dass nur die Eingabe des Spielers, der aktuell an der Reihe ist, abgehandelt wird, merkt sich die aktuell ausgewählte Spielfigur, damit das User-Interface korrekt dargestellt wird, fügt jedem Spieler seine Spielfiguren zu und setzt die Spielfiguren zu Beginn der Sitzung an zuvor festgelegte Positionen.

### 5.2 Shooting System

### 5.3 Inventory System & Inventory Component

Das Inventory System wird aufgerufen sobald ein Spieler eine der folgenden Aktionen ausführt um die Anzahl der im Inventar enthaltenen Gegenstände zu verringern:

- Nachladen der Primärwaffe
- Einsatz von Handgranaten
- Einsatz von Tränengas
- Einsatz von Rauchgranaten
- Einsatz von Molotovcocktails



---

## **5.4 Player Assistance System**

## **5.5 Ability System**

## **5.6 Health System**

## Spieler

Das Spieler Objekt enthält als Kindobjekte seine Spielfiguren. Als Skripte hängen ihm eine Player Component, sowie eine Input Component an.

### 6.1 Player Component

```
1  GameObject[] figurines = new GameObject[3]; //Alle Figuren ueber die ein
    Spieler verfuegt
2  public int actionPoints = 0; //Anzahl an verfuegbaren Aktionspunkten
3  int maxAP; //Maxcap fuer AP
```

Das Skript speichert die maximale Anzahl an Aktionspunkten, die für die verschiedenen Fraktionen variieren, füllt nach dem Ende der Runde die Aktionspunkte beider Fraktionen auf und stellt sicher, dass dabei die Zahl der erhaltenen Aktionspunkte nicht die Grenze überschreiten.

#### Maximale Aktionspunkte Rebellen

$(n \text{ Figuren} + 4) * 2$

#### Maximale Aktionspunkte Regierungstruppen

$(n \text{ Figuren} + 2) * 2$

#### Aktionspunkteregeneration Rebellen

Aktionspunkte + Anzahl an Figuren + 4

#### Aktionspunkteregeneration Regierungstruppen

Aktionspunkte + Anzahl an Figuren + 2

**Abb. 6.1.** Berechnung der Aktionspunkte

### 6.2 Input Component

## Spielfiguren

### 7.1 Movement System

### 7.2 Attribute Component

## Pathfinding

”Civil War Nation” benutzt ein in Zellen aufgeteiltes Spielfeld. Um die Bewegung der Figuren auf diesem Spielfeld zu ermöglichen, müssen die günstigsten Pfade gefunden werden. Hierbei wird der ”Dijkstra Algorithmus” eingesetzt, der von der aktuell ausgewählten Figur die Entfernung zu allen anderen Zellen auf dem Spielfeld zu berechnen.

## Kamera

## User-Interface

### 10.1 Selektierte Spielfigur

### 10.2 Action-Points Leiste

### 10.3 Dynamische Ability-Icons

## **3D Modelling**

### **11.1 Figuren**

### **11.2 Waffen**

### **11.3 Equipment**

## **Animationen**

### **12.1 Motion Capture Aufnahmen**

### **12.2 Einbindung der Animationen**



## Sounds



Status quo

## Fazit