

Fallstudie Rollenspiel

Arbeitsauftrag

1 Einleitung

In dieser Fallstudie entwickeln Sie ein Konsolenspiel in der Programmiersprache C, das die grundlegenden Funktionen eines textbasierten Rollenspiels simuliert. Ziel dieser Aufgabe ist es, die erlernten Kenntnisse der Programmier Technik anzuwenden. Ausserdem ist diese Fallstudie eine Vorbereitung auf die Praxisarbeit.

2 Aufgabenstellung

Entwickeln Sie ein text-basiertes Rollenspiel in C, bei dem der Spieler gegen verschiedene Monster kämpfen kann. Der Spieler soll auch die Angriffsart wählen und das Monster auswählen können, gegen das er kämpfen möchte.

Spielmechanik

Charakter erstellen

- Der Spieler soll seinen Namen eingeben können. Der Spieler hat eine Gesundheit von 100 und Angriffskraft von 20 Punkten.

```
----Character-----  
Name: Roman  
Gesundheit: 100  
Angriff: 20
```

Monsterwahl

- Der Spieler soll aus einer Liste von Monstern wählen können, gegen welches Monster er kämpfen möchte. Beispiel:

```
Wählen Sie ein Monster zum Kämpfen:  
1: Monster 1 (Gesundheit: 83, Angriff: 16)  
2: Monster 2 (Gesundheit: 77, Angriff: 15)  
3: Monster 3 (Gesundheit: 93, Angriff: 15)  
Geben Sie die Nummer des Monsters ein: 1
```

Angriffsart-Wahl

Implementieren Sie verschiedene Angriffsarten: Physischer Angriff, Magischer Angriff, Fernkampfangriff und Giftangriff. Bitte den physischen Angriff und den magischen Angriff wie unten beschrieben implementieren. Für den Fernkampfangriff und den Giftangriff können Sie eine eigene Schadensberechnung erfinden.

Physischer Angriff (PHYSICAL):

- Ein grundlegender Nahkampfangriff, der direkten physischen Schaden verursacht.
- Schadensberechnung: Der Schaden entspricht der Angriffskraft des Charakters.
- Formel: Schaden = Angriffskraft

Magischer Angriff (MAGICAL):

- Ein Angriff, der magischen Schaden verursacht, typischerweise stärker als ein physischer Angriff.
- Schadensberechnung: Der Schaden ist 1,5-mal die Angriffskraft des Charakters.
- Formel: Schaden = Angriffskraft * 1.5
- Beispiel: Ein Charakter mit 20 Angriffspunkten verursacht 30 Schaden.

```
Wählen Sie eine Angriffsart:  
1: Physischer Angriff  
2: Magischer Angriff  
3: Fernkampfangriff  
4: Giftangriff  
Geben Sie die Nummer der Angriffsart ein: 2
```

Kampfablauf:

- Der Spieler greift das Monster mit der ausgewählten Angriffsart an, und die Gesundheit des Monsters wird entsprechend der Angriffskraft und dem Schadensmultiplikator reduziert.
- Falls das Monster nicht besiegt wurde, greift es den Spieler an, und die Gesundheit des Spielers wird um den Angriffswert des Monsters reduziert.
- Dieser Vorgang wird wiederholt, bis entweder der Spieler oder das Monster besiegt ist.

```
----Kampf!!!!-----  
Roman greift Monster 1 mit einem magischen Angriff an!  
Schaden Monster: 40  
Monster 1 greift Roman an!  
Veränderung Gesundheit Character: -16
```

Kampfende:

- Wenn das Monster besiegt wird, wählt der Spieler das nächste Monster aus, gegen das er kämpfen möchte.

```

----Monster-----
Name: Monster 3
Gesundheit: -27
Angriff: 15
Sie haben Monster 3 besiegt!
Wählen Sie ein Monster zum Kämpfen:
1: Monster 2 (Gesundheit: 77, Angriff: 15)
Geben Sie die Nummer des Monsters ein: 1

```
- Wenn der Spieler oder alle Monster besiegt sind, endet das Spiel.

```

Sie haben Monster 2 besiegt!
Glückwunsch, Sie haben alle Monster besiegt!

```
- Oder:

```

----Character-----
Name: Roman
Gesundheit: -9
Angriff: 20
Status: Vergiftet: 0, Betäubt: 0
----Monster-----
Name: Monster 3
Gesundheit: 33
Angriff: 15
Sie wurden besiegt...

```

Projektstruktur:

- Erstellen Sie mehrere Dateien für die Module: main.c, character.c, character.h, monster.c, monster.h, game.c, game.h, attack.c, attack.h.

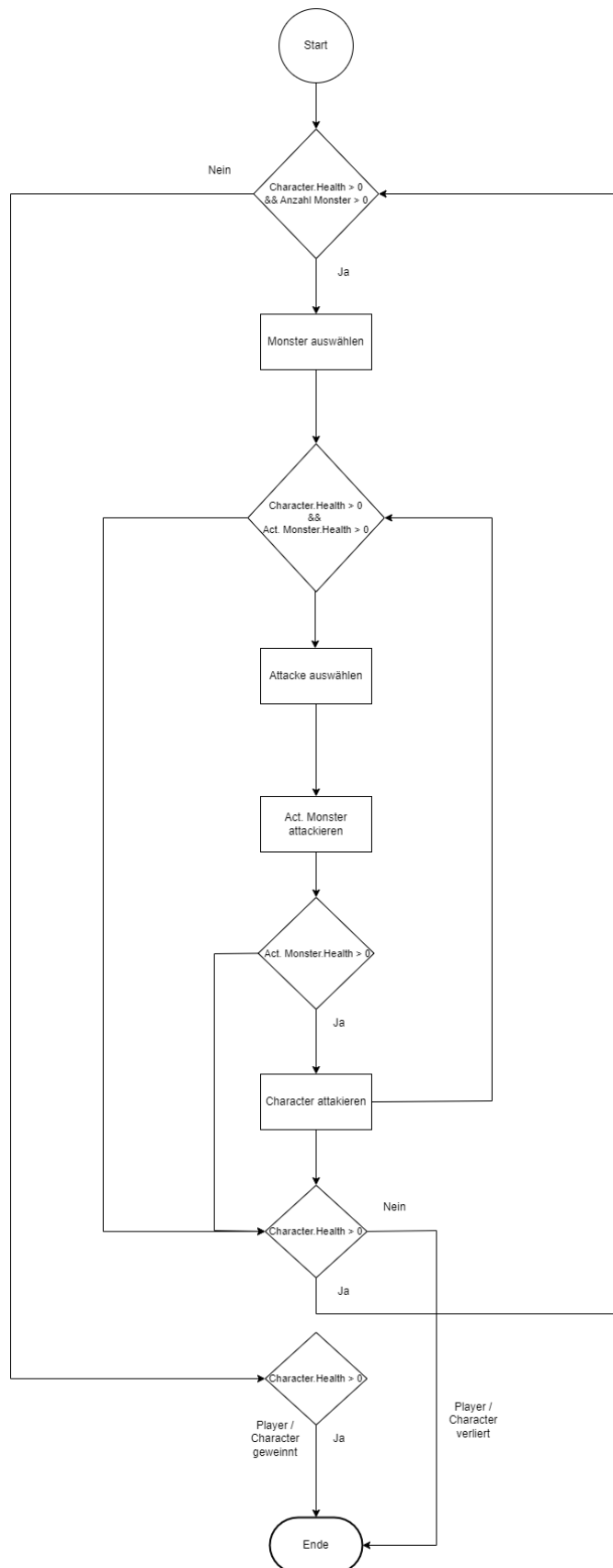
Module und Header-Dateien:

- **character.h:** Definieren Sie den Character-Struct und die Funktionen zur Erstellung und Anzeige des Charakters.
- **character.c:** Implementieren Sie die Funktionen zur Erstellung und Anzeige des Charakters.
- **monster.h:** Definieren Sie den Monster-Struct und die Funktionen zur Erstellung und Anzeige von Monstern.
- **monster.c:** Implementieren Sie die Funktionen zur Erstellung und Anzeige von Monstern.
- **attack.h:** Definieren Sie die verschiedenen Angriffsarten und die Funktionen zur Durchführung der Angriffe.
- **attack.c:** Implementieren Sie die Funktionen zur Durchführung der Angriffe.
- **game.h:** Definieren Sie die Funktionen zur Spielsteuerung.
- **game.c:** Implementieren Sie die Funktionen zur Spielsteuerung.
- **main.c:** Implementieren Sie die Hauptfunktion, die das Spiel startet.

Strukturen und Funktionen:

- **Character-Struct:** Enthält Name, Gesundheit, Angriffskraft und Bitfelder für Status-Effekte.
- **Monster-Struct:** Enthält Name, Gesundheit und Angriffskraft.
- **Funktionen:**
 - `createCharacter(Character *character)`: Erstellt den Charakter basierend auf den Eingaben des Spielers. (Modul character)
 - `displayCharacter(const Character *character)`: Zeigt die Eigenschaften des Charakters an. (Modul character)
 - `Monster createMonsters(int numMonsters)`: Erstellt eine Liste von Monstern.
 - `displayMonster(const Monster *monster)`: Zeigt die Eigenschaften eines Monsters an. (Modul Monster)
 - `performAttack(Character *attacker, Monster *defender, AttackType attackType)`: Führt einen Angriff auf das Monster aus. (Modul attack)
 - `performMonsterAttack(Monster *attacker, Character *defender)`: Führt einen Monsterangriff auf den Charakter aus. (Modul attack)
 - `selectAttack(AttackType *attackType)`: Ermöglicht dem Spieler die Auswahl der Angriffsart. (Modul game)
 - `selectMonster(Monster *monsters, int numMonsters, int *selectedMonster)`: Ermöglicht dem Spieler die Auswahl eines Monsters. (Modul game)
 - `startGame(Character *player, Monster *monsters, int numMonsters)`: Steuert den Ablauf des Spiels. (Modul game)

Spielablauf (Funktion startGame)



Teilaufgaben

1. Erstellen Sie die Projektstruktur und die Funktionsrumpfe (Prototypen)
2. Implementieren Sie das Spiel.
3. Testen Sie Ihr Spiel, ob es den Anforderungen entspricht.

Lieferobjekte

- Dokumentation der Arbeit
- Implementation des Spiels in einem main.c und mit den erwähnten C-Modulen.
- Zip-File einreichen mit Namen der Gruppenteilnehmer.