Roleplaye Game mit React (JS), HTML5 und CSS

Einfaches 2D RPG Game für den Browser

Semesterarbeit FWebT

von

Benjamin Käslin

Inhaltsverzeichnis

[1 Spielidee 4](#_Toc729642708)

[1.1 Regeln 4](#_Toc1665346851)

[1.2 Wireframe 4](#_Toc1451636243)

[2 Technologie Stack 5](#_Toc1247170993)

[3 Anforderungen 6](#_Toc1211742505)

[3.1 Kampfsystem 19](#_Toc1669394470)

[3.1.1 Berechnung Angriffsschaden 19](#_Toc1816283939)

[3.2 Bewegungssystem 19](#_Toc365181446)

[3.3 Monsters 20](#_Toc284693727)

[4 Struktur und Planung 21](#_Toc264900540)

[4.1 Struktur 21](#_Toc1190629479)

[4.2 Planung 22](#_Toc1955485820)

[4.3 PVA1 Nachbereitung 22](#_Toc177362280)

[4.4 PVA2 Nachbereitung 22](#_Toc809618336)

[4.5 PVA3 Nachbereitung 22](#_Toc1249306993)

[4.6 PVA4 Nachbereitung 22](#_Toc1220868512)

[5 Logbuch 23](#_Toc474269208)

[5.1 PVA2 23](#_Toc1287561289)

[5.2 PVA3 23](#_Toc744330098)

[5.3 PVA4 23](#_Toc1916213056)

**Abkürzungen**

|  |  |
| --- | --- |
| RPG | Role Play Game |
| JS | Javascript |
| NPC | Non Player Character |

# Spielidee

Ein einfaches Rollenspiel für den Browser. Der Spieler kann sich einen Charakter erstellen, mit welchem er die vorgegeben Dungeons im Spiel überstehen muss. Durch das erkunden und besiegen von Gegener, kann der Spieler das Levels seines Charakters erhöhen, dadurch werden die Basiswerte des Charakters erhöht.

Optional: Der Spieler kann verschiedene Gegenstände im Spiel finden und diese entsprechend anziehen oder verkaufen. Zusätzlich gibt es an bestimmten Orten einen Laden, in welchem er sich mit dem erspielten Gold, verschiedene Gegenstände erwerben kann.

## Regeln

Das Spiel ist zu Ende, wenn der Spieler stirbt (sprich keine Lebenspunkte mehr besitzt) oder alle Dungeons überstanden hat.

## Wireframe

Eine erste Idde wie das Spiel auf der Webseite angezeigt wird.

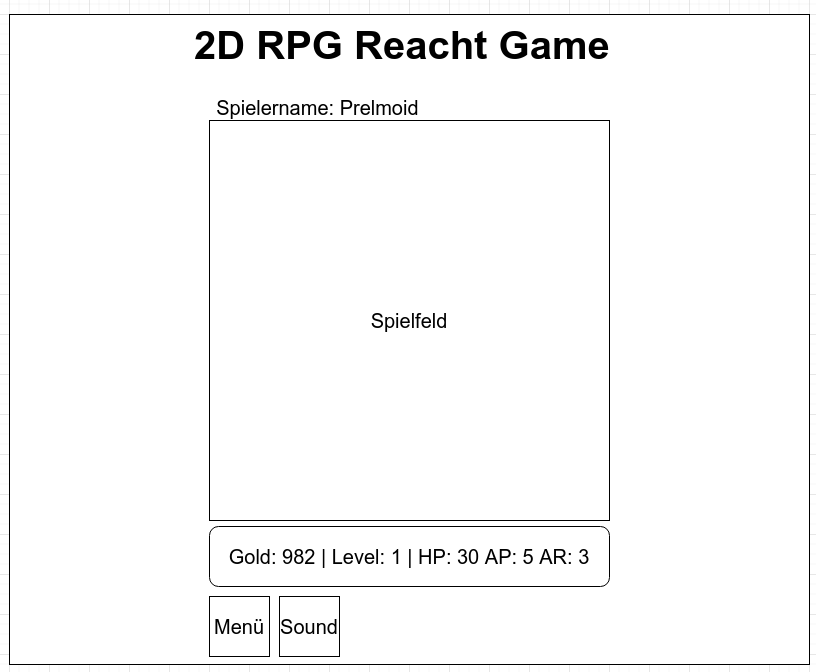


Image 1 Wireframe

# Technologie Stack

Für diese Arbeit wird ausschliesslich auf JavaScript (mit dem React Framework), sowie HTML5 und CSS verwendet.

# Anforderungen

Im folgenden Abschnitt werden die verschiedenen Anforderungen aufgeführt.

| **Name** | **Startseite / Spielstart** |
| --- | --- |
| ****ID**** | FA-00 |
| **Ziel** | Die Webseite wird geladen, der Spieler kann ein neues Spiel starten |
| ****Ereignis**** | Jemand besucht die Webseite. |
| ****Vorbedingung**** | Webbrowser geöffnet |
| ****Standardablauf**** | 1. Eingabe Charaktername 2. Spieler steht im ersten Dungeon |
| ****Alternativablauf**** | Falls es nicht der erste Besuch ist, und eine Speicherfunktion vorhanden ist. Kann ein Spielstand geladen werden. |
| ****Nachbedingung Erfolg**** | Der Spielcharakter befindet sich in der 2D Welt, am entsprechenden Ort (Startpunkt oder Ladestand |
| ****Nachbedingung Fehler**** | - |
| ****Klassifizierung**** | Funktional, MUSS |
| ****Aufwand**** | GROSS |

Table 1 FA-00 Spielstart

| **Name** | **Charakter Movement** |
| --- | --- |
| ****ID**** | FA-01 |
| **Ziel** | Der Spielcharakter kann sich auf der Karte bewegen. |
| ****Ereignis**** | Der Spieler betätigt eine der [BEWEGUNGS] Tasten (W[North],A[West],S[South],D[East]) |
| ****Vorbedingung**** | Unter welchen Bedingungen kann die Anforderung auftreten/erfüllt werden - Das Spiel ist gestartet - Der Spieler kann sich in eine Richtung bewegen |
| ****Standardablauf**** | Beschreibung des Ablaufs bei erfüllung der Vorbedingungen 1. Der Spieler betätigt die [BEWEGUNGS] Taste 2. Der Charakter bewegt sich in die entsprechende Richtung |
| ****Alternativablauf**** | Hindernis: Charakter bleibt stehen, dreht sich in die Richtung. |
| ****Nachbedingung Erfolg**** | Der Charakter hat sich in die entsprechende Richtung bewegt. |
| ****Nachbedingung Fehler**** | - |
| ****Klassifizierung**** | Funktional, MUSS |
| ****Aufwand**** | GROSS |

Table 2 FA-01 Charakter Movement

| **Name** | **Angriff (Nahkampf)** |
| --- | --- |
| ****ID**** | FA-02 |
| **Ziel** | Ein Spielercharakter kann ein Ziel angreiffen |
| ****Ereignis**** | Der Spieler betätigt die [ANGRIFF] Taste |
| ****Vorbedingung**** | - Das Spiel ist gestartet - Der Spieler ist am leben (> 0 Lebenspunkte) - Der Charakter ist in einem Gebiet mit Gegner (Dungeon) |
| ****Standardablauf**** | 1. Der Spieler betätigt die [ANGRIFF] Taste 2. Die Nahkampf Animation wird ausgelöst 3. Der Gegner befindet sich eine Bewegungseinheit vor dem Charakter |
| ****Alternativablauf**** | Ausserhalb Dungeon: Soundfile "sorry, can't do" wird abgespielt |
| ****Nachbedingung Erfolg**** | Gegnertreffer: Der Schaden des Gegners wird berechnet und dessen Lebenspunkte abgezogen.  Dazu werden die Angriffswerte des Spielers berücksichtigt, sowie der Rüstungswert des Gegners. |
| ****Nachbedingung Fehler**** | - |
| ****Klassifizierung**** | Funktional, MUSS |
| ****Aufwand**** | GROSS |

Table 3 FA-02 Angriff (Nahkampf)

| **Name** | **Interaktion (Shop)** |
| --- | --- |
| ****ID**** | FA-03 |
| **Ziel** | Ein Spieler kann Waren kaufen / verkaufen |
| ****Ereignis**** | Der Spieler betätigt die [INTERAKTIONS] Taste |
| ****Vorbedingung**** | - Das Spiel ist gestartet - Der Spieler ist am leben (> 0 Lebenspunkte) - Der Charakter ist in einem Gebiet mit NPC’s(Dungeon) |
| ****Standardablauf**** | 1. Der Spieler betätigt die [INTERAKTIONS] Taste 2. Der Shopdialog öffnet sich 3. Spieler kann Gegenstände kaufen oder verkaufen |
| ****Alternativablauf**** | NPC kann nicht interagieren: Soundfile "sorry, can't do" wird abgespielt |
| ****Nachbedingung Erfolg**** | Gegenstände befinden sich im Inventar, oder sind aus dem Inventar entfernt. |
| ****Nachbedingung Fehler**** | - |
| ****Klassifizierung**** | Funktional, KANN |
| ****Aufwand**** | MITTEL |

Table 4 FA-03 Interaktion (Shop)

| **Name** | **Interaktion Loot (Schatztruhe/Gegner/etc.)** |
| --- | --- |
| ****ID**** | FA-04 |
| **Ziel** | Ein Spieler kann mit lootbaren Gegenstände interagieren. |
| ****Ereignis**** | Der Spieler betätigt die [INTERAKTIONS] Taste |
| ****Vorbedingung**** | - Das Spiel ist gestartet - Der Spieler ist am leben (> 0 Lebenspunkte) - Der Charakter befindet sich vor einem gestorbenen Gegner, einer Schatztruhe oder ähnlichem. |
| ****Standardablauf**** | 1. Inhalt der Truhe / Gegners wird angezeigt 2. Spieler wählt Gegenstände aus für in das Inventar, oder interagiert nicht mit ihnen. |
| ****Alternativablauf**** | - |
| ****Nachbedingung Erfolg**** | Gegenstände befinden sich im Inventar. |
| ****Nachbedingung Fehler**** | - |
| ****Klassifizierung**** | Funktional, KANN |
| ****Aufwand**** | MITTEL |

Table 5 FA-04 Interaktion Loot

| **Name** | **Erfahrungspunkte** |
| --- | --- |
| ****ID**** | FA-05 |
| **Ziel** | Ein Spieler kann Erfahrungspunkte erhalten, und das Level des Charakters erhöhen. |
| ****Ereignis**** | Gegner stirbt. |
| ****Vorbedingung**** | - Das Spiel ist gestartet - Der Spieler ist am leben (> 0 Lebenspunkte) - Der Charakter hat einen Gegner bezwungen |
| ****Standardablauf**** | 1. Ein NPC wurde bezwungen 2. Erfahrungspunkte, welcher der Gegner ergibt, werden dem Charakter zugerechnet 3. Falls die nötigen Punkte für das nächste Level erreicht werden, werden die Charakterwerte erhöht. Lebenspunkte: +15 | Angriffswert: +5 | Rüstungswert: +1 |
| ****Alternativablauf**** | Levelup: Soundfile "Levelup" wird abgespielt |
| ****Nachbedingung Erfolg**** | Erfahrungspunkte wurden dem Gesamtstand zugerechnet |
| ****Nachbedingung Fehler**** | - |
| ****Klassifizierung**** | Funktional, MUSS |
| ****Aufwand**** | MITTEL |

Table 6 FA-05 Erfahrungspunkte

| **Name** | **Charakter stirbt** |
| --- | --- |
| ****ID**** | FA-06 |
| **Ziel** | Das Spiel ist beendet. |
| ****Ereignis**** | Der Spieler hat weniger als einen Lebenspunkt |
| ****Vorbedingung**** | - Das Spiel ist gestartet - Der Spieler ist nicht am leben (< 1 Lebenspunkte) - Der Charakter ist in einem Gebiet mit NPC’s(Dungeon) |
| ****Standardablauf**** | 1. Der Charakter stirbt 2. Das Spiel ist beendet 3. Der Spieler befindet sich im Startbildschirm [FA-00] |
| ****Alternativablauf**** | - |
| ****Nachbedingung Erfolg**** | Dem Spieler stehen wieder die Optionen aus FA-00 zur Verfügung. |
| ****Nachbedingung Fehler**** | - |
| ****Klassifizierung**** | Funktional, MUSS |
| ****Aufwand**** | MITTEL |

Table 7 FA-06 Charakter stirbt

| **Name** | **Speichern** |
| --- | --- |
| ****ID**** | FA-07 |
| **Ziel** | Der Spielstand kann gespeichert werden |
| ****Ereignis**** | Im Menü wird das Spiel gespeichert |
| ****Vorbedingung**** | - Das Spiel ist gestartet - Der Spieler ist am leben (> 0 Lebenspunkte) - Der Spieler hat das Spielmenü geöffnet |
| ****Standardablauf**** | 1. Der aktuelle Spielstand wird gespeichert 2. Der vorherige Speicherpunkt wird überschrieben. |
| ****Alternativablauf**** | - |
| ****Nachbedingung Erfolg**** | Das Spiel kann forgesetzt werden |
| ****Nachbedingung Fehler**** | - |
| ****Klassifizierung**** | Funktional, KANN |
| ****Aufwand**** | MITTEL |

Table 8 FA-07 Speichern

| **Name** | **Spielstand laden** |
| --- | --- |
| ****ID**** | FA-08 |
| **Ziel** | Der aktuelle Spielstand wird mit dem geladenen Spielstand überschrieben. |
| ****Ereignis**** | Der Spieler wählt im Menü die Ladeoption aus |
| ****Vorbedingung**** | - Das Spiel ist gestartet |
| ****Standardablauf**** | 1. Im Spielmenü die Ladeoption auswählen 2. Der gespeicherte Spielstand wird geladen |
| ****Alternativablauf**** | - |
| ****Nachbedingung Erfolg**** | Das Spiel kann nun mit dem geladenen Spielstand fortgesetzt werden. |
| ****Nachbedingung Fehler**** | - |
| ****Klassifizierung**** | Funktional, KANN |
| ****Aufwand**** | MITTEL |

Table 9 FA-08 Spielstand laden

| **Name** | **Spielsound ein-/ausstellen** |
| --- | --- |
| ****ID**** | FA-09 |
| **Ziel** | Das Spiel spielt Sound ab oder nicht |
| ****Ereignis**** | Der Spieler togglet die Soundoption |
| ****Vorbedingung**** | - Das Spiel ist gestartet - Der Spielsound wird abgespielt oder  - Der Spielsound wird nicht abgespielt |
| ****Standardablauf**** | 1. Sound toggle 2. A) Sound war eingestellt -> Sound abstellen B) Sound war ausgestellt -> Sound einstellen |
| ****Alternativablauf**** | - |
| ****Nachbedingung Erfolg**** | Der Spielsound wird entsprechend der Einstellung abgespielt oder nicht. |
| ****Nachbedingung Fehler**** | - |
| ****Klassifizierung**** | Funktional, KANN |
| ****Aufwand**** | MITTEL |

Table 10 FA-09 Spielsound ein-/ausstellen

| **Name** | **Lebenswiederherstellung** |
| --- | --- |
| ****ID**** | FA-10 |
| **Ziel** | Der Charakter stellt Lebenspunkte wieder her |
| ****Ereignis**** | Der Spieler wählt einen Heilungstrank aus zur Benutzung |
| ****Vorbedingung**** | - Das Spiel ist gestartet - Der Charakter ist am leben (> 0 Lebenspunkte) - Der Charakter hat >= 1 Heilungstrank |
| ****Standardablauf**** | 1. Öffnen des Inventars 2. Auswahl eines Heiltranks zur Konsumation |
| ****Alternativablauf**** | - |
| ****Nachbedingung Erfolg**** | Der Charakter hat wieder zusätzliche Lebenspunkte, jedoch nicht mehr als das maximum seiner Lebenspunkte. |
| ****Nachbedingung Fehler**** | - |
| ****Klassifizierung**** | Funktional, KANN |
| ****Aufwand**** | WENIG |

Table 11 FA-10 Lebenswiederherstellung

| **Name** | **Ausrüstung anlegen** |
| --- | --- |
| ****ID**** | FA-11 |
| **Ziel** | Der Charakter rüstet sich mit unterschiedlicher Ausrüstung aus. |
| ****Ereignis**** | Der Spieler wählt ein Rüstungselement(z.b. Schulter) oder eine Waffe aus. |
| ****Vorbedingung**** | - Das Spiel ist gestartet - Der Charakter ist am leben (> 0 Lebenspunkte) - Der Charakter hat Ausrüstung erhalten |
| ****Standardablauf**** | 1. Öffnen des Inventars 2. Auswahl eines Rüstungsgegenstand |
| ****Alternativablauf**** | - |
| ****Nachbedingung Erfolg**** | Der Rüstungsgegenstand ist angezogen. Falls bereits ein Gegenstand an der Stelle vorhanden ist, dieses zurück ins Inventar legen. Die Werte des Charakters wurden entsprechend aktualisiert. |
| ****Nachbedingung Fehler**** | - |
| ****Klassifizierung**** | Funktional, KANN |
| ****Aufwand**** | MITTEL |

Table 12 FA-11 Ausrüstung anlegen

| **Name** | **Goldstand Charakter** |
| --- | --- |
| ****ID**** | FA-12 |
| **Ziel** | Der Spieler verfügt über Gold, welches er ausgeben kann. |
| ****Ereignis**** | Ein Gegner stirbt, oder es wird etwas gelootet [FA-04]. |
| ****Vorbedingung**** | - Das Spiel ist gestartet - Der Charakter ist am leben (> 0 Lebenspunkte) |
| ****Standardablauf**** | 1. Dem Goldstand des Charakters wird ein Wert X zugerechnet. 2. Der Goldstand entspricht dem alten Wert zuzüglich dem dazu gewonnen Gold. |
| ****Alternativablauf**** | - |
| ****Nachbedingung Erfolg**** | Der Goldstand wird dem Spieler korrekt angezeigt. |
| ****Nachbedingung Fehler**** | - |
| ****Klassifizierung**** | Funktional, KANN |
| ****Aufwand**** | MITTEL |

Table 13 FA-12 Goldstand Charakter

| **Name** | **Kollision mit soliden Elementen** |
| --- | --- |
| ****ID**** | FA-13 |
| **Ziel** | Der Spieler bewegt sich nicht, wenn etwas im Weg steht.. |
| ****Ereignis**** | Der Spielercharakter will sich bewegen. |
| ****Vorbedingung**** | - Das Spiel ist gestartet - Der Charakter ist am leben (> 0 Lebenspunkte) |
| ****Standardablauf**** | 1. Der Spieler bewegt sich in die gewünschte Richtung ausser: a) Ein Gegner steht auf de b) Eine Wand ist im Weg c) Eine Truhe befindet sich auf dem Feld d) Prinzipielle alle Elemente auf dem Level, welche als nicht ‘begehbar’ gekennzeichnet sind |
| ****Alternativablauf**** | - |
| ****Nachbedingung Erfolg**** | Dem Spieler wird ausgegeben, dass er sich nicht entsprechend bewegen kann. |
| ****Nachbedingung Fehler**** | - |
| ****Klassifizierung**** | Funktional, MUSS |
| ****Aufwand**** | WENIG |

Table 14 FA-13 Kollision mit soliden Elementen

## Kampfsystem

Das Kampfsystem wird möglichst einfach gehalten. Damit ein Angriff erfolgreich sein kann, muss der Spielcharakter jeweils in die Richtung schauen, in welcher der Gegner sich befindet. Zusätzlich darf sich der Gegner nur eine Bewegungseinheit [aktuell gehe ich von Tiles von 16x16 Pixel aus, wird sich bei der Endauswahl der Gameassets entscheiden] entfernt befinden.

### Berechnung Angriffsschaden

Der Schaden, welcher an einem Gegner entsteht wird wie folgt berechnet:

Schaden = [round]{(Angriffsschaden\*Angriffsschaden)/(Angriffschaden + Rüstungswert des angegriffenen Charakters)}

## Bewegungssystem

Der Spielcharakter, sowie die Monster, kann sich jeweils in eine Richtung bewegen. Er hat die Möglichkeit nach Norden, Süden, Osten oder Westen sich zu bewegen. Pro Tile kann sich nur ein Monster, oder der Spielcharakter etc. darauf befinden.

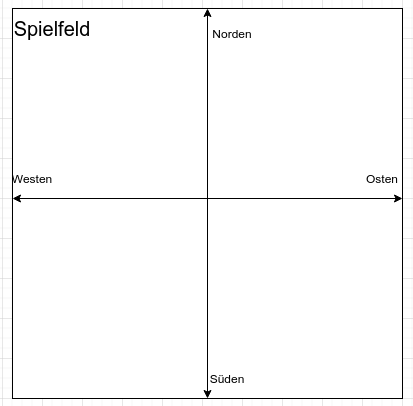


Image 2 Spielfeld

Wenn der Spieler sich um eine Tile[=Bewegungseinheit] bewegt, bewegen sich die Gegner entsprechend ihren Bewegungsmöglichkeiten, wenn diese im sichtbaren Bereich sind. Dadurch bewegen sich alle beweglichen Einheiten auf dem Spielfeld nur wenn sich auch der Spielcharakter bewegt.

## Monsters

In der folgenden Tabelle ist eine erste Auflistung von Gegner, sowie ihren Einschränkungen und Fähigkeiten.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bezeichnung** | **Lebenspunkte** | **Rüstungswert** | **Angriffswert** | **Bewegungseinheit** |
| Ratte | 10-45 | 3 | 4 | 2 |
| Skellet | 30-90 | 5 | 7 | 1 |
| Drache | 100-350 | 8 | 12 | 0.5 |

Table 15 Monsters

# Struktur und Planung

Vorstellung der Struktur und eine kurze Ablaufplanung.

## Struktur

src  
 data  
 items - icons and classes for all the items  
 maps - tiles for dungeons, city... as well as initial map designs[position of tiles, monsters, ..]  
 monsters - icons and classes for enemies  
 features  
 app-state - save/load game progress  
 dialog-manager - handle dialogs   
 game-menus - game menus (save, load, new game, options[sound on/off], inventory)  
 inventory - equip/drop item, sell item, buy item  
 monsters - handle monster actions..  
 player - handle player actions  
 stats - handle levels and experience  
 world - reset game state, start new game, load game  
 utils - if needed  
 docs  
 documents related to the game, such as roadmap or contributors etc.  
 public  
 for public stuff, like manifest.json for icons of the program if saved etc.

## Planung

Erste Grobplanung.

## PVA1 Nachbereitung

* Ordnerstruktur erstellen
* Anforderungen definieren
* Initialcommit
* Gameart beschaffen (vorselektieren)

## PVA2 Nachbereitung

* GUI Grundstruktur erstellen
* Gameassets finalisieren
* Programmierung der ersten Anforderungen
* Unittests Framework einbinden und für erstellte Logik abdecken

## PVA3 Nachbereitung

* Einen Gegner auf der Karte anzeigen und bewegen
* Backend für das Speichern des Spielstands erlauben
* ServiceWorker erstellen und ausprobieren
* Asynchronen Aufruf erstellen
* Weitere Funktionalitäten umsetzen [z.B. Kampfsystem]

## PVA4 Nachbereitung

* Präsentation vorbereiten
* Spiel finalisieren

# Logbuch

Veränderungen zwischen den Abgaben.

## PVA2

* Erstellung der Basisseite.
* Eine Map kann angezeigt werden und die Spielfigur
* Die Spielfigur kann über die Map bewegt werden, ohne dass sie auf Wänden stehen kann

## PVA3

## PVA4

Abbildungsverzeichnis

[Image 1 Wireframe 4](#_Toc1397782198)

[Image 2 Spielfeld 19](#_Toc1053349063)

Tabellenverzeichnis

[Table 1 FA-00 Spielstart 6](#_Toc1889127401)

[Table 2 FA-01 Charakter Movement 7](#_Toc439491518)

[Table 3 FA-02 Angriff (Nahkampf) 8](#_Toc1482046469)

[Table 4 FA-03 Interaktion (Shop) 9](#_Toc476608241)

[Table 5 FA-04 Interaktion Loot 10](#_Toc1411079391)

[Table 6 FA-05 Erfahrungspunkte 11](#_Toc1660833622)

[Table 7 FA-06 Charakter stirbt 12](#_Toc955557650)

[Table 8 FA-07 Speichern 13](#_Toc573993864)

[Table 9 FA-08 Spielstand laden 14](#_Toc1182976115)

[Table 10 FA-09 Spielsound ein-/ausstellen 15](#_Toc942142914)

[Table 11 FA-10 Lebenswiederherstellung 16](#_Toc1884235958)

[Table 12 FA-11 Ausrüstung anlegen 17](#_Toc80411824)

[Table 13 FA-12 Goldstand Charakter 18](#_Toc42726274)

[Table 14 Monsters 20](#_Toc1764441625)