

# Q1 – A game

---

Modul

Objektorientierte Skriptsprachen

Medieninformatik Online

(MIB08 SS 2016)

Fachbereich Informatik und Medien der  
Beuth Hochschule für Technik Berlin

Autor: René Rose

Matrikelnummer: 774295

1. Spielanleitung.....	3
1.1 Willkommen bei Q1! .....	3
1.2 Karte .....	3
1.3 Objekte.....	4
1.4 Interaktionen .....	4
1.5 Zum Spiel .....	5
2. Aufgabenstellung .....	5
2.1 Ziel der Arbeit.....	5
2.2 Anforderungen.....	5
3. Umsetzung .....	7
4. Zusammenfassung.....	10
5. Anhänge und Verzeichnisse .....	10
5.1 Quellen.....	10
5.2 Abbildungsverzeichnis .....	10
5.3 Tabellenverzeichnis.....	10
5.4 Verwendete Software.....	11

# 1. Spielanleitung

## 1.1 Willkommen bei Q1!

Q1 spielt in einer fiktiven namenlosen Gegend, die vor haarsträubenden Gegnern nur so wimmelt. Hier geht es um das nackte Überleben!

Der bösartige Spawner, der Ursprung aller böstigen Wesen, muss beseitigt werden.

Erst dann kann dieses gepeinigste Land wieder Frieden haben ...

Doch zuerst ein paar Informationen, sonst hat das gefährliche Unterfangen wenig Aussicht auf Erfolg.

## 1.2 Karte

Auf der Karte befinden sich mehrere Landschaftstypen, die unterschiedliche Eigenschaften besitzen:








Typ	ID	Begehrbar <sup>1</sup>	Bearbeitbar <sup>2</sup>	Tödlisch <sup>3</sup>	Schüsse <sup>4</sup>	Bild
Erde	21	ja	nein	nein	ja	
Wasser	1	nein	nein	nein	ja	
Kalkstein	61	nein	ja	nein	nein	
Granit	62	nein	nein	nein	nein	
Lava	127	ja	nein	ja	ja	
Wiese	128	ja	ja	nein	ja	
Wald	161	ja	ja	nein	nein	

Tabelle 1: Landschaftstypen

1 Avatar kann diesen Typ betreten

2 Landschaft kann manipuliert werden (es wird Erde daraus)

3 Bei betreten stirbt der Avatar

4 Über diese Typen kann geschossen werden

## 1.3 Objekte

Folgende Objekte treten auf dem Spielfeld auf:



Name	Bild	Schaden	Beschreibung
Avatar		3	Der Protagonist. Schießt immer in Blickrichtung. Diese ist in nebenstehendem Bild nach unten gerichtet.
Schuss			Der Schuss. Einzige Verteidigungsmöglichkeit des Avatars. Steht unbegrenzt zur Verfügung
Grab			Stirbt der Avatar, wird er begraben. Sterben kann er durch betreten tödlicher Landschaftsteile oder er wird durch ein Monster gefressen.
Spawner		15	Der Endboss. Er muss unbedingt vernichtet werden, sonst spawnen in regelmäßigen Abständen immer neue Monster.
Bug of Death		5	Ein Abbild des Grauens. Fürchterlich. Allein bei seinem Anblick schlottern den Gegnern des Bug of Death die Knie.
Snake of Hell		5	Ein Abbild der Hölle. Immens grün. Fürchtet allerdings Wasser.

Tabelle 2: Spielobjekte

## 1.4 Interaktionen

Mit folgenden Tasten wird Q1 gespielt:

- Pfeiltasten: der Avatar wird über die Spielfläche bewegt.
- Leertaste: Landschaftstyp des aktuellen Felds wird in Erde umgewandelt. Nicht betretbare Felder können wie folgt abgebaut werden: der Avatar positioniert sich neben dem abzubauenen Feld. Dann die Pfeiltaste in Richtung des abzubauenen Feldes drücken und dazu die Leertaste.
- S-Taste: Avatar feuert einen Schuss in Blickrichtung ab. Dieser fliegt maximal acht Felder weit.

## 1.5 Zum Spiel

Der Avatar wird immer auf der linken Seite ins Spielfeld gesetzt. Der Spawner befindet sich irgendwo auf dem Spielfeld. Es gibt eine maximale Anzahl Monster, die nicht überschritten wird. Solange der Spawner existiert, wird in regelmäßigen Abständen ein Monster wiederbelebt, solange bis die Maximalzahl wieder erreicht ist. Das Ziel des Avatar sollte es also sein, zuerst den Spawner zu besiegen. Dann können alle Monster beseitigt werden. Möge das Glück auf deiner Seite sein, Avatar!

## 2. Aufgabenstellung

---

### 2.1 Ziel der Arbeit

Ziel des Projektes ist die Entwicklung eines Spieles mit der Programmiersprache Python und dem Modul Pygame.

### 2.2 Anforderungen

Q1 ist ein 2D-RPG. Es soll als Prototyp erstellt werden, um möglicherweise später weitere Funktionen einzufügen. In eine zufallsgenerierte Landschaft wird ein Avatar gesetzt. Dieser kann sich frei bewegen, einige Landschaftselemente können jedoch nicht betreten werden. Es werden Gegner auftauchen, die den Avatar angreifen. Irgendwo auf der Karte befindet sich ein Spawner, welcher die Gegner ins Leben ruft. Der Avatar muss den Spawner und die Gegner zerstören.

Folgende Anforderungen werden definiert:

Kriterium	ID	Datum	Autor	Beschreibung	Abhängigk.
Muss	001	04.06.2016	R. Rose	Aus einer per Perlin-Noise generierten Value-Map wird eine Tile-Map errechnet, die die „Landschaft“ darstellt.	-
Muss	002	04.06.2016	R. Rose	Die Tiles sind in quadratischer Form umgesetzt.	001
Muss	003	04.06.2016	R. Rose	Der gesamte Screen soll mit dieser Landschaft gefüllt werden. Die Landschaft besteht aus 70 x 30 Tiles, je 20x20 Pixel pro Tile	002
Muss	004	04.06.2016	R. Rose	Darstellung erste Stufe sind nur topologische Tiles (Stein, Erde, Wasser, (Lava?))	001

Muss	005	04.06.2016	R. Rose	Einige Landschaftselemente sollen nicht betreten werden können.	004
Muss	006	04.06.2016	R. Rose	Zu jeder Tile, auf die der Avatar gesetzt wird, wird eine textuelle Information ausgegeben.	004, 005, 009
Muss	007	04.06.2016	R. Rose	Es werden infrastrukturelle Tiles (Wiese, Wald) umgesetzt	002, 004
Muss	008	04.06.2016	R. Rose	Einige Landschaftselemente sollen manipulierbar sein, bspw. können entfernt werden	007
Muss	009	04.06.2016	R. Rose	Der Avatar kann sich frei bewegen	-
Muss	010	04.06.2016	R. Rose	Der Avatar hat eine Blickrichtung in die er immer agiert.	009
Muss	011	04.06.2016	R. Rose	Der Avatar wirft einen Gegenstand zum Bekämpfen der Gegner, der auch zum Zerstören von nicht betretbaren Tiles genutzt werden kann	005
Muss	012	04.06.2016	R. Rose	Aktionen werden mit der Leertaste ausgelöst.	-
Soll	013	04.06.2016	R. Rose	Es sollen zwei Arten von Gegnern geben, die sich per Zufallsalgorithmus über die Karte bewegen	001
Soll	014	04.06.2016	R. Rose	Die Gegner sollen bekämpft werden können	013
Soll	015	04.06.2016	R. Rose	Es wird ein Spawner gesetzt, der zerstört werden muss/kann	-
Soll	016	04.06.2016	R. Rose	Treffen die Gegner auf den Avatar, stirbt dieser	-
Soll	017	04.06.2016	R. Rose	Initial stirbt der Gegner beim ersten Treffer	009
Kann	018	04.06.2016	R. Rose	ESA 3 wird aus ESA 2 heraus aufgerufen	ESA 2
Kann	019	04.06.2016	R. Rose	Für ESA 3 wird ein neuer Thread gestartet.	018
Kann	020	04.06.2016	R. Rose	Die in der Anwendung von ESA 2 gewählte Map wird in ESA 3 geladen und zum Spielen verwendet	018
Wird	021	04.06.2016	R. Rose	Das Spiel wird gespeichert.	-

Tabelle 3: Anforderungen

### 3. Umsetzung

Das Spiel wird in einem eigenständigen Fenster in der Größe 1760 x 960 Pixeln dargestellt. Es kann sowohl mit Aufruf der eas3.py oder auch aus der GTK-Oberfläche der ESA 2 (esa2.py) gestartet werden. Für den Aufruf aus esa3.py muss eine transformierte Map mit dem Namen Map1 in der Datenbank existieren.

Die Landschaft ist aus einem Perlin-Noise ähnlichem Skript generiert (siehe Anhang) und als statische Liste zur Verfügung gestellt (siehe settings.py)

Über die Oberfläche der ESA 2 wird diese Liste um eine Dimension erweitert und speichert nun die Landschaftsinformationen wie in Tabelle 1 beschrieben (fieldtype, walkable, editable, deadly, text), (siehe auch esa2.py, def mapConverter()).

Aus dieser Liste wird die Landschaft in 32 x 32 Pixel großen Kacheln (Tiles) erstellt. Somit ist die Anzahl der Kacheln auf der X-Achse bei 55 und auf der Y-Achse auf 32 festgelegt.

Dieser Bildschirm verbleibt immer in dieser Darstellung und kann nicht verschoben, bzw.- verlassen werden.

Die Landschaftselemente sind mit verschiedenen Eigenschaften umgesetzt (siehe Tabelle1).

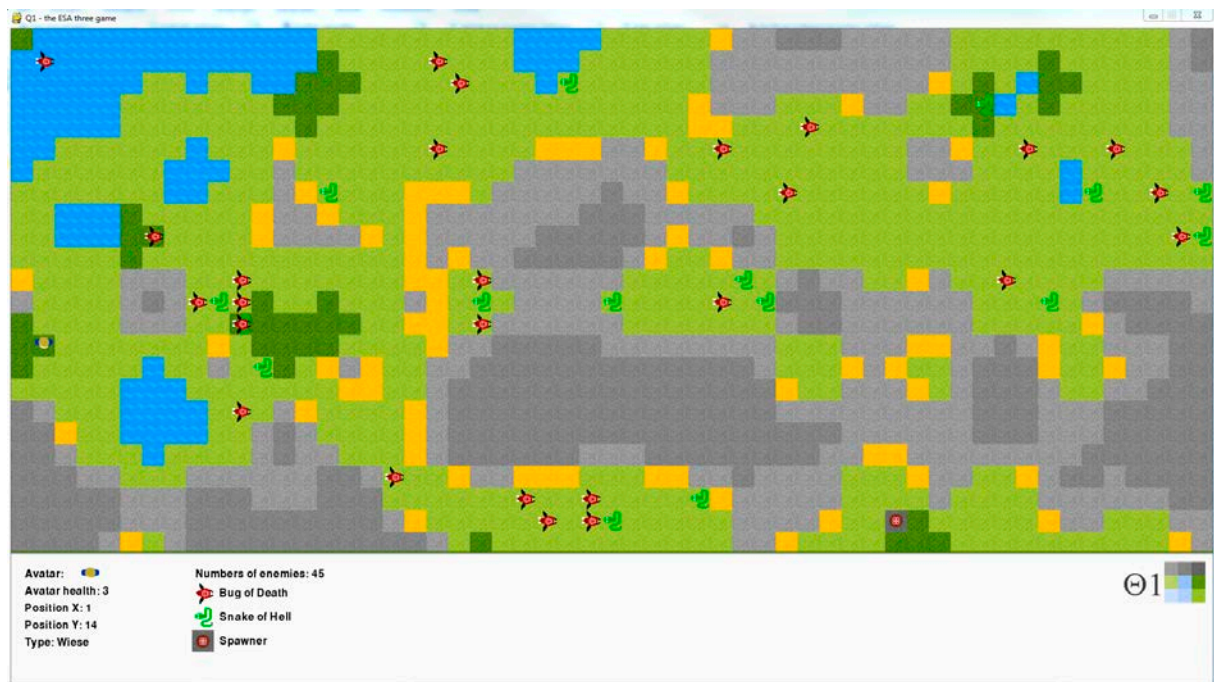


Abbildung 1: Spielfläche

Der Avatar kann sich frei bewegen, kann jedoch den sichtbaren Bereich nicht verlassen. Der Avatar kann manipulierbare Kacheln manipulieren mit der Leertaste, wenn diese begehbar sind oder mit Pfeiltaste und Leertaste, wenn diese nicht begehbar sind. Betritt der Avatar tödliche Kacheln, stirbt er.

Die Bewegung der Gegner ist zufallsbasiert. Trifft ein Gegner und der Avatar auf einem Feld zusammen, stirbt der Avatar. Der Avatar hat drei Leben, sind diese verwirkt, ist das Spiel verloren. Sind alle Gegner besiegt, ist das Spiel gewonnen.

Im unteren Fensterbereich ist eine Infobar umgesetzt, welche Informationen zum aktuellen Spielgeschehen abbildet.

Zusätzliche Umsetzung, die nicht in den Anforderungen definiert sind:

- Avatar dreht sich auch optisch in die Bewegungsrichtung,
- Avatar hat eine Anzahl Leben,
- Sind die Leben verwirkt, wird das Spiel beendet,
- Sind alle Gegner besiegt, wird das Spiel beendet,
- Solange der Spawner existiert werden Monster generiert, bis zu einer maximalen Anzahl,
- Ein beendetes Spiel startet erneut nach einigen Sekunden,
- Audioeffekte bei Interaktionen,
- Avatar und Gegner können nicht auf Kacheln spawnen, von denen aus keine Bewegungen möglich sind,
- Gegner können sich nicht auf Kacheln bewegen, die nicht begehbar sind,
- Gegner werden Lava nicht betreten und auch das Spielfeld nicht verlassen,
- Über Wald kann nicht geschossen werden,
- Einige Gegner können sich über Wasser bewegen, einige nicht.

Kriterium	ID	Beschreibung	Abhängigk.
Muss	001	Aus einer per Perlin-Noise generierten Value-Map wird eine Tile-Map errechnet, die die „Landschaft“ darstellt.	Nicht vollständig erfüllt. Der Noise wird statisch zur Verfügung gestellt. Es ist immer derselbe.
Muss	002	Die Tiles sind in quadratischer Form umgesetzt.	erfüllt
Muss	003	Der gesamte Screen soll mit dieser Landschaft gefüllt werden. Die Landschaft besteht aus 70 x 30 Tiles, je 20x20 Pixel pro Tile	Erfüllt, allerdings abweichende Tile-Maße und Tileanzahl
Muss	004	Darstellung erste Stufe sind nur topologische Tiles (Stein, Erde, Wasser, (Lava?))	erfüllt
Muss	005	Einige Landschaftselemente sollen nicht betreten werden können.	erfüllt
Muss	006	Zu jeder Tile, auf die der Avatar gesetzt wird, wird eine textuelle	erfüllt



		Information ausgegeben.	
Muss	007	Es werden infrastrukturelle Tiles (Wiese, Wald) umgesetzt	erfüllt
Muss	008	Einige Landschaftselemente sollen manipulierbar sein, bspw. können entfernt werden	erfüllt
Muss	009	Der Avatar kann sich frei bewegen	erfüllt
Muss	010	Der Avatar hat eine Blickrichtung in die er immer agiert.	erfüllt
Muss	011	Der Avatar wirft einen Gegenstand zum Bekämpfen der Gegner, der auch zum Zerstören von nicht betretbaren Tiles genutzt werden kann	Teilweise erfüllt. Der Wurfgegenstand tötet nur die Gegner, Tiles werden per Leertaste und Pfeiltaste manipuliert
Muss	012	Aktionen werden mit der Leertaste ausgelöst.	Teilweise erfüllt. Taste S löst einen Schuss aus.
Soll	013	Es sollen zwei Arten von Gegnern geben, die sich per Zufallsalgorithmus über die Karte bewegen	erfüllt
Soll	014	Die Gegner sollen bekämpft werden können	erfüllt
Soll	015	Es wird ein Spawner gesetzt, der zerstört werden muss/kann	erfüllt
Soll	016	Treffen die Gegner auf den Avatar, stirbt dieser	erfüllt
Soll	017	Initial stirbt der Gegner beim ersten Treffer	Erfüllt, Gegner haben sogar eine einstellbare Trefferanzahl.
Kann	018	ESA 3 wird aus ESA 2 heraus aufgerufen	erfüllt
Kann	019	Für ESA 3 wird ein neuer Thread gestartet.	erfüllt
Kann	020	Die in der Anwendung von ESA 2 gewählte Map wird in ESA 3 geladen und zum Spielen verwendet	erfüllt
Wird	021	Das Spiel wird gespeichert.	nicht umgesetzt

Tabelle 4: Status der Anforderungen

## 4. Zusammenfassung

---

Es sind nahezu alle Anforderungen erfüllt worden. Einige sind mit einer äquivalenten Lösung umgesetzt worden, in Tabelle 2 aber trotzdem als nicht vollständig erfüllt markiert. Lediglich die wird-Anforderung zum Speichern des Spiels aus esa3.py heraus wurde nicht erfüllt.

Somit kann das Projekt als den Anforderungen weitestgehend entsprechend umgesetzt betrachtet werden. Viele zusätzliche Funktionen sind umgesetzt (siehe 3.). Die Verknüpfung von ESA 2 und ESA3 rundet das Ganze ab.

Einige Sachen sind schon geplant, konnte aber aus Zeitgründen nicht mehr für die aktuelle Version umgesetzt werden, bspw.:

- Harmonisierung des Waldes und der Landschaftsdarstellung,
- Shot des Avatar sollte in eigener Klasse umgesetzt werden,
- Kollisionsabfrage Enemy auf Enemy, das nicht zwei Gegner auf einem Feld stehen,
- Animationen,
- Laufen bei permanentem Tastendruck,
- Gegner beschützen den Spawner besser.

Bekannter Bug:

Die textuelle Ausgabe gibt nicht immer den korrekten Kacheltypen an. Eine Lösung ist in Arbeit.

## 5. Anhänge und Verzeichnisse

---

### 5.1 Quellen

Audioeffekte von [www.soundbible.com](http://www.soundbible.com)

Noise from <https://gist.github.com/veb/7b9f5393d0c25977e4cb>

### 5.2 Abbildungsverzeichnis

Abbildung 1: Spielfläche.....7

### 5.3 Tabellenverzeichnis

Tabelle 1: Landschaftstypen.....3

Tabelle 2: Spielobjekte.....4

Tabelle 3: Anforderungen.....	6
Tabelle 4: Status der Anforderungen.....	9

## 5.4 Verwendete Software

Python 3.4.4

Pygame 1.9.2a0

PyCharm Community Edition 2016.1.2

PhotoShop CS 6

MS Word 2010

Subversion