

3. Welt-Service



Der Welt-Service erzeugt die Spielwelt und liefert Informationen über die Umwelt. Die Kommunikation erfolgt über einen Socket.

Initialisierung

Beim Start des Programms registriert sich dein Programm beim Discovery-Service.

- `action: register`
- `type: world`
- `ip`: Die IP-Adresse auf der dein Programm Nachrichten empfangen wird.
- `port`: Der Port über den dein Programm Nachrichten empfangen wird.

Als Antwort sendet der Discovery-Service OK oder ERROR. Bei ERROR muss das Programm abbrechen.

Das Programm erzeugt ein neues Objekt der Klasse „Map“.

Spielablauf

Das Programm öffnet einen Socket unter der oben registrierten Adresse/Port. Auf diesem Socket wartet das Programm auf Nachrichten.

Aktionen

Neue Welt

Das Programm erzeugt eine neue Spielwelt und platziert die Ameisenhügel auf der Karte.

Request

Ein JSON-Objekt mit den Attributen:

- `action: „new“`
- `count`: Die Anzahl der Ameisenvölker in der Welt.

Response

Ein JSON-Array mit den Koordinaten aller Ameisenhügel.

```
[  
  {"xcoord": n, "ycoord": m},  
  {"xcoord": n, "ycoord": m},  
  ...  
]
```

Umgebung

Das Programm liefert die Umgebung einer Ameise, abhängig von der Sichtweite.

Request

Ein JSON-Objekt mit den Attributen:

- action: „area“
- xcoord: Die X-Koordinate einer Ameise.
- ycoord: Die Y-Koordinate einer Ameise.
- range: Die Sichtweite einer Ameise.

Response

Ein JSON-Array mit den Zuständen der Felder. Die Reihenfolge der Felder ist immer von oben links nach unten rechts.

0	1	2	3	4
5	6	7	8	9
10	11		13	14
15	16	17	18	19
20	21	22	23	24

Dieses Beispiel zeigt die Umgebung einer Ameise mit einer Sichtweite 2. Die Ziffern entsprechen der Reihenfolge in der Response.

Karte

Das Programm liefert die vollständige Karte. Nur autorisierte Services dürfen diese Aktion ausführen.

Request

Ein JSON-Objekt mit den Attributen:

- `action`: „map“
- `token`: Ein verschlüsseltes Authentifikationstoken.

Response

Ein JSON-Array mit den Angaben zur Welt.

- `xsize`: Breite der Welt
- `ysize`: Höhe der Welt
- `fields`: Zweidimensionales Array mit allen Feldern der Welt.

Beenden

Erhält das Programm eine Nachricht mit „`action`“: „quit“, so wird das Programm beendet:

1. Sende eine Antwort mit OK über den Socket.
2. Schliesse den Socket
3. Beende das Programm.

Objekte und Methoden

- Klasse: [Map](#)
- Klasse: [Field](#)

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
<https://wiki.bzz.ch/modul/m321/projekt/welt>

Last update: **2024/03/28 14:07**

