Klasse: Map

Die Klasse „Map“ verwaltet die Spielkarte der Welt.

Attribute

- `width`: Breite der Karte
- `height`: Höhe der Karte
- `fields`: Zweidimensionales Array mit `Field`-Objekten.

Methoden

Methode "create_world"

Parameter

- `hive_count`: Anzahl der Ameisenvölker

Returnwert

- Liste von Dictionaries mit den Koordinaten der Ameisenhügel.

Beispiel

```
[
    {
        "xcoord": 7,
        "ycoord": 12
    },
    {
        "xcoord": 45,
        "ycoord": 3
    },
]
```

Beschreibung

Alle Ameisenvölker sollen vergleichbare Bedingungen beim Start des Spiels haben. Um eine Chancengleichheit zu gewährleisten, müssen für alle Völker ...

- ... gleiche viele Nahrungsressourcen erreichbar sein,
- ... andere Ameisenhügel im gleichen Abstand zu einander sein.

Variante 1: Zufallskarte
Die Karte kann zufällig generiert werden.

- Grösse der Karte: Als Grösse der Karte wird die Anzahl der Völker mit einem Faktor (z.B. 8) multipliziert. Dazu addieren wir in der Höhe und Breite jeweils 2 Felder für das Wasser, welches den Rand bildet.
- Ameisenhügel: Positioniere den ersten Ameisenhügel mit 2-3 Feldern Abstand zum Rand der Karte. Die weiteren Hügel werden nun im Uhrzeigersinn mit gleichen Abstand (+/- 1) am Rand positioniert.
- Nahrung: Für eine gerechte Verteilung der Nahrung muss die Menge (1-99) und Distanz zu den verschiedenen Hügeln berücksichtigt werden.
    - Du kannst dazu gleich verfahren wie bei den Hügeln.
    - Man könnte auch eine Formel entwickeln, die für jeden Hügel aus Abstand und Menge einen Wert ermittelt. Nun werden solange zufällige Verteilungen der Nahrung ausprobiert, bis alle Hügel einen gleichen Wert erreichen.

Variante 2: Vordefinierte Kartenteile
Bei diesem Ansatz werden mehrere Varianten von Kartenteilen mit jeweils einem Hügel und 3-4 Nahrungsquellen definiert. Aus diesen Varianten dieser Teilkarten wählt das Programm eine Teilkarte aus, wobei für jedes Volk die gleiche Teilkarte verwendet wird. Diese Teilkarten werden gleichmässig angeordnet und allfällige Lücken mit zufälligen Feldern aufgefüllt.

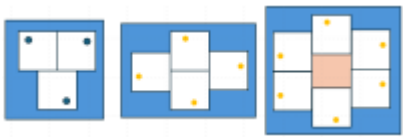

Abbildung: Anordnung der Kartenteile für 3,4 und 6 Spieler

Methode "show_area"

Parameter

- `xcoord`: Die X-Koordinate einer Ameise.
- `ycoord`: Die Y-Koordinate einer Ameise.
- `color`: Die Farbe des Ameisenvolks
- `range`: Die Sichtweite einer Ameise

Response

Ein JSON-Array mit den Zuständen der Felder. Jedes Feld kann einen dieser Zustände haben:

- **water**: Das Wasser bildet eine natürliche Barriere, die deine Ameisen nicht überwinden können.
- **home**: Dein eigener Ameisenhügel.
- **hill**: Der Ameisenhügel eines anderen Volks.
- **friend**: Eine eigene Ameise.
- **foe**: Eine Ameise eines anderen Volks.

- **food**: Auf diesem Feld liegt 1 - 99 Nahrung, die Menge an Nahrung ist nicht bekannt.
- **empty**: Ein leeres Feld auf das deine Ameise laufen kann.

Es wird immer nur der erste zutreffende Zustand ausgegeben.

Ist also in einem Feld eine eigene Ameise **und** Nahrung, wird nur **ant** angezeigt.

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.

Methode "show_map"

Parameter

Keine

Returnwert

Ein zweidimensionales Array (list) mit den Feldern der Karte. Jedes Feld kann einen dieser Zustände haben:

- **water**: Das Wasser bildet eine natürliche Barriere, die deine Ameisen nicht überwinden können.
- **hill *COLOR***: Der Ameisenhügel eines anderen Volks.
- **ant *COLOR***: Eine Ameise eines anderen Volks.
- **food**: Auf diesem Feld liegt 1 - 99 Nahrung, die Menge an Nahrung ist nicht bekannt.
- **empty**: Ein leeres Feld auf das deine Ameise laufen kann.

Es wird immer nur der erste zutreffende Zustand ausgegeben. Zusätzlich wird bei **ant** und **hill** die Farbe mitgegeben, z.B. „ant red“, „hill blue“.

Ist also in einem Feld eine eigene Ameise **und** Nahrung, wird nur **ant** angezeigt.

Methode "add_ant"

Diese Methode setzt eine neue Ameise auf ein Feld.

Parameter

- `xcoord`: Die X-Koordinate des Felds.
- `ycoord`: Die Y-Koordinate des Felds.
- `color`: Die Farbe des Ameisenvolks

Returnwert

Keiner

Methode "remove_ant"

Diese Methode entfernt eine Ameise auf einem Feld.

Parameter

- `xcoord`: Die X-Koordinate des Felds.
- `ycoord`: Die Y-Koordinate des Felds.
- `color`: Die Farbe des Ameisenvolks

Returnwert

Keiner

Methode "move_ant"

Diese Methode bewegt eine Ameise von einem Feld zu einem anderen Feld in der Weltkarte. Dazu erhält sie die Startposition (x/y-Koordinaten) und die Zielposition (x/y-Koordinaten).

1. Entferne die Ameise vom Startfeld.
2. Füge beim Zielfeld eine Ameise hinzu.
3. Ermittle die Angaben zum Zielfeld (siehe Returnwert).

Parameter

- `xcoord_start`: Die X-Koordinate der Startposition.
- `ycoord_start`: Die Y-Koordinate der Startposition.
- `xcoord_target`: Die X-Koordinate der Zielposition.
- `ycoord_target`: Die Y-Koordinate der Startposition.

Returnwert

Der Returnwert ist eine Angabe zum Zielfeld:

- „`food`“: Nahrung vorhanden
- „`water`“: Feld mit Wasser
- „`home`“: Eigener Ameisenhügel
- „`hill`“: Fremder Ameisenhügel
- „`empty`“: Leeres Feld

Methode "change_food"

Die Methode ändert die Menge an Nahrung auf einem Feld.

- Positive Menge: Die Menge an Nahrung wird hinzugefügt.
- Negative Menge: Die Menge an Nahrung wird weggenommen.
- 0: Die Menge an Nahrung wird auf 0 gesetzt.

Parameter

- `xcoord`: Die X-Koordinate des Felds.
- `ycoord`: Die Y-Koordinate des Felds.
- `amount`: Die Menge an Nahrung

Returnwert

Keiner