

Maprotagenerator Manual

timbow, fletschoa, kappakay

Inhaltsverzeichnis

1	Quickstart	2
2	Einstell Parameter Übersicht	3
3	Neue Map einfügen	5

1 Quickstart

Zum schnellen loslegen mit dem Generieren einer Rotation müssen lediglich in der ‘config.json’ gewünschte Anzahl an Rotationen (`number_of_rotas`), die Anzahl an Layer pro Rotation (`number_of_layers`) und die gewünschte Anzahl an Seed Layern am Start der Rotation (`seed_layer`) eingestellt werden. Sämtliche sonstigen Parameter sind optional und können ersteinmal von den Voreinstellungen übernommen werden.

Bei erstmaligem ausführen wird außerdem empfohlen, ‘update_layers’ zu aktivieren, um automatisch eine aktuelle version der Layer und zugehöriger Votes abzurufen. Es wird generell Empfohlen, diese Option immer aktiviert zu lassen, ist zur reinen Funktionalität jedoch nicht zwingend erforderlich. Zum Generieren der Rotation muss nun lediglich die ‘main.js’ ausgeführt werden. Nach Abschluss der Generierung speichert diese die Generierte Rotation im Hauptordner unter dem Namen ‘layer_rota_nr.cfg’

2 Einstell Parameter Übersicht

number_of_rotas [int]

anzahl an Rotas, die generiert werden sollen

number_of_layers [int]

Wie viele Layer eine Rotation insgesamt beinhalten soll

seed_layer [int]

Wie viele Seed Layer am sich am Start der Rota befinden sollen

update_layers [bool]

Entscheidet, ob beim start layer/votes neu abgerufen werden sollen

mode_distribution [dict]

Struktur:

pools [dict]

Beliebige anzahl an Mode pools, die eine beliebige Anzahl an modes beinhalten und deren wahrscheinlichkeit innerhalb des Pools festlegen. Wichtig: Es muss immer mindestens den 'main' pool mit mindestens einem Modes geben. Struktur:

```
1 {
2     "pool_name":
3     {
4         "Mode1": probability (float),
5         "Mode2": probability (float)
6     }
7 }
```

pool_distribution [dict]

Wahrscheinlichkeiten für die mode pools, s.o Struktur:

```
1 {
2     "pool_name1": probability (float),
3     "pool_name2": probability (float)
4 }
```

pool_spacing [int]

Mindestabstand zwischen nicht 'main' pool modes

space_main [bool]

Entscheidet, ob 2x der selbe Mode aus dem Main pool hintereinander kommen dürfen oder sie sich abwechseln müssen.

Einstellungen ab hier haben enorme Auswirkungen auf die Generierung und Verteilung. Änderungen ohne genaueres Verständnis der Funktionsweise und des Ablaufs der Generierung nicht empfohlen

biom_spacing [int]

Wie lange ein Cluster gelockt wird.

min_biom_distance [float]

Bioms umkreis Abstand, in dem gelockt wird

mapvote_slope [float]

Slope der Mapvoteweight Sigmoid funktion

mapvote_shift [int]

Shift der Mapvoteweight Sigmoid funktion

layervote_slope [float]

Slope der Layervote Sigmoid funktion

layervote_shift [int]

Shift der Layervote Sigmoid funktion

use_vote_weight [bool]

Entscheidet, ob die layer nach Votes gewichtet werden sollen

Achtung: Deaktivieren sorgt nicht für eine Gleichverteilung der Maps.

use_map_weight [bool]

Entscheidet, ob die Maps nach Votes gewichtet werden sollen

save_expected_map_dist [bool]

Entscheidet, ob die erwartete Mapverteilung nach Mapvoteweight als Datei gespeichert werden soll

use_lock_time_modifier [bool]

Entscheidet, ob die Cluster-Locktime für eine bessere verteilung korrigiert werden darf.

auto_optimize [bool]

Entscheidet, ob bei Änderungen relevanter Werte vor Generierung einer Rota automatisch der Optimizer gestartet werden soll Dringend empfohlen.

3 Neue Map einfügen

Um eine neue Map in die Generierung aufzunehmen, müssen der Map in 'data/bioms.json' biom Parameter zugewiesen werden.

Beispiel Struktur der 'bioms.json':

```
1 {  
2   "Sumari": [0.0,0.0,0.0,0.0,0.8,0.0,1.0,0.0,0.3],  
3   "Logar": [0.03,0.0,0.0,0.0,1.0,0.0,0.9,0.4,0.2]  
4 }
```

Jede Map hat 9 Biom Parameter, jeder Parameter bildet einen float wert in dem Biom Array in folgender Reihenfolge:

1. Mapgröße
2. Wald
3. Schnee
4. Wasser
5. Wüste
6. Grasland
7. Stadt
8. Berge
9. Felder