

バトロワプラグインの概要

yaml ファイルとデータパックを組み合わせることである程度自由なバトロワを作ることができます。

可動に必要な plugin は NBT-API のみです

またバトロワ可動にあたり、フィールドの設定を書いた yaml ファイルとルートテーブルの設定ファイル（後述）がそれぞれ最低一つ必要になります

ゲームの仕様上持っているアイテムは全て消えることになるので、ぜったにメインワールドに導入しないでください

一応現時点でも（多分）動く状態にはあると思いますが

- ・setgame コマンド（後述）を実行したワールドでゲームが行われるが、待機所の設定など何も書いてない

- ・バトロワが終了したときの演出を書いてない

って問題があります

フィールドの設定 yaml の要素と書き方

firstCenter:

- X:(double)

- Z:(double)

最初のエリアの中心

firstWidth: (double)

最初のエリアの広さ（一辺の 1/2 を設定してください 例：500 を設定すると 1000×1000 の範囲になる）

generatingRate: (double 0~1)

ルートチェストの生成率 コマンドで後からいじることもできます(その場合 yaml には保存されません)

randomTierWeight:

- Tier1: (int)

- Tier2: (int)

- ...

- TierN: (int)

ルートチェストの Tier をランダムに設定した場合でのそれぞれの Tier の重み

Tier の最大値計測も兼ねているのでランダムを使わないとしても設定してください

全て 0 にすると多分バグります

carePackage:

- frequency: (int)

- Tier: (int)

- rate: (double 0~1)

ケアパッケージが降るかどうか判定する頻度(frequency)、ケアパッケージの Tier、判定によってケアパッケージが降る確率(rate)

dropShipAltitude: (double)

ドロップシップの高さ

playerHealth: (int)

プレイヤーの初期体力（基本 20 でいいと思います）

firstAreaWaitTime: (int)

一番初めのエリア収縮が決定されるまでの時間

areaReduction:

```
-“1”:  
    -waitTime: (int)  
    -executeTime: (int)  
    -reductionRate: (double 0~1)  
    -spawnableCarePackage: (Boolean)  
    -areaDamage: (int)  
    -areaDamageBuffer: (int)  
  
-“2”:  
    -waitTime: (int)  
    -executeTime: (int)  
    -reductionRate: (double 0~1)  
    -spawnableCarePackage: (Boolean)  
    -areaDamage: (int)  
    -areaDamageBuffer: (int)  
  
...  
  
-“n”:  
    -waitTime: (int)  
    -executeTime: (int)  
    -reductionRate: (double 0~1)  
    -spawnableCarePackage: (Boolean)  
    -areaDamage: (int)  
    -areaDamageBuffer: (int)
```

初めの 1~n は「フェーズ」です、1 から順番に指定してください 理論上 int の最大値まで続けられます

そのほかは順番に「エリア収縮が始まるまでの時間」「エリア収縮を実行する時間」「エリアの収縮率(1 で変わらず、0 でエリア消滅)」「ケアパッケージが新しく出現するか」「エリア外のダメージ」「エリア外でダメージを受ける間隔」

chestPosition:

```
-X 座標|Y 座標|Z 座標:  
    -X: (double)  
    -Y: (double)  
    -Z: (double)  
    -Tier: (int)
```

...

ルートチェストの場所と Tier

この部分のみワールドから編集できます（コマンド説明のワールド編集関連を参照してください）

X 座標|Y 座標|Z 座標:の部分はコンフィグから直接編集する場合（ないとは思いますが）はどんな名前にしても OK です

各種コマンド説明

ゲーム進行関連

一般ピーポーが使えるのはここにあるものの一部だけです

以下一般ピーポー用

join

ゲームに参加登録します

leave

ゲームの参加登録を解除します

以下より op 用

setgame <使用するフィールドのデータを書いた yml ファイル(.yaml まで書いてください)> <使用するルートテーブル名>
バトロワの設定をロードします これを実行すると参加を促す文章が鯖に投下されます

setrate <double(0~1)>

ルートチェストの出現率を設定します

start

バトロワを開始します

cancel

-開催前のバトロワを削除します

stop

-開催中のバトロワを中断します

ワールド編集関連

editfield <編集したいファイル(.yaml まで)>

-指定したファイルを編集する準備をします

lootchest <get|visible|invisible|delete>

以下各種説明

get <(int)>

手に持ったブロックを「Tier(int)」(int は自然数または 0)という名前に変更します。これを設置することで editfield で設定したファイルにルートチェストの場所と Tier が保存されます（なので基本クリエ専用です）

visible <(int)|all>

editfield で設定したファイルの chestPosition の位置に中身が空のチェストを設置します (int)でどの Tier の位置に設置するかを指定することができ、all で全ての位置に設置します

invisible

上記の visible で設置したチェストを全て消します

delete

手に持ったアイテムを「削除」という名前にします。これを持って visible で表示したチェストを壊すことで editfield で設定したファイルからその場所のルートチェスト情報を削除します コンフィグから直接設定したルートチェストは削除できない場合があります

ルートテーブルの設定方法

データパックを用いてルートテーブルを作成します（データパックでのルートテーブル作成方法がわからない人はとりあえず gg ってください わからない部分は聞いてくれたら答えられる範囲で答えます）

git に「battleroyale」という名前のファイルが一緒に置いてあると思うので、その

dattleroyale/data/battleroyalepack/loot_tables

の中に＊新しいフォルダ（名前は小文字じゃないとダメだった気がする）を作ってください

その中に、作成したルートテーブルを「tier1」、「tier2」・・・と順にファイル名をつけて入れてください

setgame コマンドの<使用するルートテーブル名> に＊で設定した名前を入力することでこのルートテーブルが使用できま

