

# 艦これシミュレータ改の チュートリアル

書いた人：[YSR](#)

注意：

以下の記述には、[艦これ](#)のゲームのスクショの一部、  
および[艦載機厨](#)さんの制作物のスクショの一部が含まれます。

# 目次

- ▶ 最初的一步：サニティー・チェック
- ▶ 初級編：編成を組んで戦わせよう！
- ▶ 次に進む前に：JSONの基礎
- ▶ 中級編：編成データを編集しよう！
- ▶ 上級編：マップデータを作成しよう！




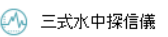





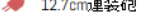
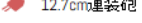
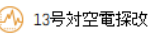
# 最初の一歩：サニティー・チェック

- ▶ run.batもしくはrun\_x64.batをクリックしてみましょう
- ▶ 「x64」と付いたexeおよびbatファイルは、64bit版のOS向けです。そうでない方は32bit版のOS向けです
- ▶ run.batをクリックした際に「MSVCP140.dllがないため～」などと警告が出た際は、「[Visual Studio 2015 の Visual C++ 再頒布可能パッケージ](#)」をダウンロードした後にインストールしてください
- ▶ log\*.txtおよびoutput\*.jsonが出力ファイルのサンプルです

# 初級編：編成を組んで戦わせよう！(1/3)

- ▶ まずは通常モードとして、艦隊と艦隊とを戦わせます
- ▶ 艦隊のデータはJSONで記述されています。よく分からなければ、「拡張子が\*.jsonなテキストファイル」とでも考えてください
- ▶ データの中身は、基本的に艦載機厨さんの「艦隊シミュレーター&デッキビルダー」で作成できます。第1艦隊の編成を組んだ後、下の方にあるテキストボックスから「{“version”:3}」で始まるテキストをコピーできるので、テキストエディタにでも貼り付けて保存してください。

# 初級編：編成を組んで戦わせよう！(2/3)

1/2 1/3 1/4						
1		Lv 20	 12.7cm連装砲	 12.7cm連装砲	 三式水中探信儀	★ <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
2		Lv 25	 12.7cm連装砲	 12.7cm連装砲	 新型高温高压缶	★ <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
3		Lv 30	 12.7cm連装砲	 12.7cm連装砲	 61cm五連装(酸素)魚雷	★ <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
4		Lv 35	 12.7cm連装砲	 12.7cm連装砲	 13号対空電探改	★ <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
5	五番艦					
6	六番艦					

- ▶ まず編成を組みます
- ▶ 第1艦隊に組むようにしましょう

<input type="text" value="{\" version\":3,\"f1\":{\"s1\":{\"ic\""=""/>	変換
<ul style="list-style-type: none"><li>• ユーザーはデッキが他の人に編集されないため、空白</li><li>• ユーザーが違くと自分のコピーが生成される。</li></ul>	

- ▶ このテキストボックスの中身をテキストエディタなどにコピーして、拡張子を「.json」にして保存します

## 初級編：編成を組んで戦わせよう！(3/3)

- ▶ 自艦隊および敵艦隊を編成して保存できたら、Readme通り「`KCS_CUI -i input1.json input2.json -f 0 0 -n 100000 -t 4 -o output.json`」などと指定して実行します
- ▶ もちろん上記操作はコマンドプロンプトなどで行います。GUI版は鋭意製作中です
- ▶ 結果は、標準出力およびファイル(-oを指定した場合)に出るはずですが、前者は「> log.txt」などとリダイレクトで書き出す方が良いでしょう
- ▶ 標準出力の結果は見たままで、ファイルの結果は標準出力に準じます(詳しくは中級編で)

## 次に進む前に：JSONの基礎(1/2)

- ▶ JSON(JavaScript Object Notation)は、テキストデータで複雑なデータ構造をやり取りするための形式の一つです
- ▶ 手で編集も可能ですが、見やすく整形したり編集したりするソフトが山のようにあります。是非利用しましょう
- ▶ 基本的には、`[1, 2, 3, 4, 5]`といった表記の配列と、`{"a": "3", "b": "hoge"}`といった表記の連想配列の2つを組み合わせでデータ構造を表現します。連想配列は、ざっくり言えば「添字に文字列を使える配列」です

## 次に進む前に：JSONの基礎(2/2)

名前：伊401改  
レベル：35  
艦種：潜水空母  
装備：試製晴嵐、  
61cm五連装(酸素)魚雷

元データ



```
{  
  "name" : "伊401改",  
  "level" : 35,  
  "type" : "潜水空母",  
  "weapon" : [  
    "試製晴嵐",  
    "61cm五連装(酸素)魚雷"  
  ]  
}
```

JSONデータ

※本来ならJSONはUTF-8で保存したいところですが、本ソフトで使用するJSONではASCII文字しか使いませんので実質関係ありません。



## 中級編：編成データを編集しよう！(1/4)

```
{  
  "version": 3,  
  "f1": {  
    "s1": {  
      "id": "403",  
      "lv": 35,  
      "luck": -1,  
      "items": {  
        "i1": {  
          "id": 62,  
          "rf": 7  
        },  
        "i2": {  
          "id": 58,  
          "rf": 10  
        }  
      }  
    }  
  }  
}
```

- ▶ では、実際のデータを読んでみましょう。  
先ほどの艦娘および装備をデッキビルダーに入力して、JSONを見てみます
- ▶ "f1"や"s1"や"i1"が、それぞれ「第1艦隊」「1番艦」「1番目の装備」を表しています。  
これらは連想配列のキーですので、同じ階層の第X艦隊・X番艦・X番目の装備と順番が入れ替わっても問題ありません。
- ▶ "version"は実質飾りですね。

## 中級編：編成データを編集しよう！(2/4)

```
{
  "version": 3,
  "f1": {
    "s1": {
      "id": "403",
      "lv": 35,
      "luck": -1,
      "items": {
        "i1": {
          "id": 62,
          "rf": 7
        },
        "i2": {
          "id": 58,
          "rf": 10
        }
      }
    }
  }
}
```

- ▶ “s○”や“i○”の値に当たる部分に“id”という項目がありますが、これはそれぞれ艦船ID・装備IDを指します。一覧はships.csvやslotitems.csvにありますが、これを弄れば**深海棲艦およびその装備も使用可能**です。
- ▶ “lv”は艦娘のレベル、“luck”は改修後の運(デフォルト値にする場合は-1とする)、“items”は装備を表す連想配列です
- ▶ “rf”は、艦載機(水偵を除く)なら熟練度、それ以外なら装備改修度を表します。

## 中級編：編成データを編集しよう！(3/4)

```
{
  ...,
  "lv": 110,
  "type": 2,
  "f1": {
    "s1": {
      ...,
      "cond": 85,
      "items": {
        "i1": {
          ...,
          "rf_detail": 120
        },
        ...,
      }
    },
    "s2": {(略)},
    ...
  },
  "f2": {(略)}
}
```

- ▶ 以下、デッキビルダーで弄れない独自項目について。図の”lv”(“f1”等と同じ階層)は司令部レベルを、”type”は艦隊形式(通常・空母機動・水上打撃・輸送護衛)を、condはcond値を、rf\_detailは内部熟練度を指します。
- ▶ 司令部レベル・艦隊形式・cond値・内部熟練度のデフォルトはそれぞれ120・1(通常艦隊)・49・？(通常熟練度に従う)です

## 中級編：編成データを編集しよう！(4/4)

- ▶ 前述のように、艦隊データを細かく弄ることで、深海棲艦との戦いや、連合艦隊による戦いなども再現できます
- ▶ デッキビルダーの使い勝手の良さを考えると、先にそちらで編集してから、細かいところを調整するべきでしょう
- ▶ 画面上で編集できるようになるGUIは現在鋭意製作中です
- ▶ ただしこれでは耐久・弾薬・燃料減少などは反映できませんので、より精密なシミュレートをしたい場合はマップモードを使用することになります

# 上級編：マップデータを作成しよう！(1/4)

- ▶ マップデータは、デッキビルダー形式という制約がない分よりシンプルに出来ています
- ▶ あくまで「一本道」しか記述できませんので、羅針盤要素はその確率を掛けることで反映してください  
(例えばボスマス率50%ならボス到達率などに0.5を掛ける)
- ▶ マップモードでは、-fによる陣形指定は開幕夜戦用となり、-oによる出力はボス戦での結果のみとなります
- ▶ したがって標準出力もしっかり読むことを推奨します

# 上級編：マップデータを作成しよう！(2/4)

- ▶ ここでは1-1(ボス戦ルート)用のデータを作成してみます
- ▶ まずはWiki等で編成を読み取ってメモします

出現場所	パターン	出現艦船	陣形
A: 敵偵察艦	パターン1	駆逐イ級	単縦
	パターン2	駆逐ロ級	
	パターン3	駆逐ハ級	
B: 敵はぐれ艦隊	パターン1	駆逐イ級、駆逐イ級	単縦
	パターン2	駆逐ロ級、駆逐イ級	
	パターン3	駆逐ハ級、駆逐ハ級	
C: 敵主力艦隊	パターン1	軽巡ホ級、駆逐イ級、駆逐イ級	単縦
	パターン2	軽巡ホ級、駆逐ロ級、駆逐ロ級	
	パターン3	軽巡ホ級、駆逐ハ級、駆逐ロ級、駆逐ロ級	

## 上級編：マップデータを作成しよう！(3/4)

- ▶ 敵の番号を見ながらJSONデータに落とし込みます
- ▶ 設定する内容としては、各マスの名前・各マスにおける戦闘モード(0～2が昼夜戦闘・昼戦のみ・開幕夜戦)・出てくる艦隊(編成および陣形)となります
- ▶ 構造としては、「マスごとに幾つかの編成が存在し、またマスの順番は上から下へ一定である」といったもの
- ▶ 性質上、ボスマスは旗艦を倒したいので、旗艦が潜水艦ではない場合は戦闘モードは当然「0」、道中は「1」が普通かと思われます

# 上級編：マップデータを作成しよう！(4/4)

```
{
  "version": "map",
  "position": [
    {
      "name": "A",
      "pattern": [{"fleets": [501], "form": 0},
                  {"fleets": [502], "form": 0},
                  {"fleets": [503], "form": 0}],
      "mode": 1
    },
    {
      "name": "C",
      "pattern": [{"fleets": [505, 501, 501], "form": 0},
                  {"fleets": [505, 502, 502], "form": 0},
                  {"fleets": [505, 503, 502, 502], "form": 0}],
      "mode": 0
    }
  ]
}
```