



# FT联合开源交流群 代码组员培养教程 地图篇

来到另一个世界 |

注意：严禁使用中文标点和  
全角输入法

严禁在代码里用空格，如需  
使用，一律改成“\_”

讲师：Twisunshine Intelligent  
联系QQ：1261911579

我给小马国画了新地图!



# 1 认识地图文件

## 2 这个txt是什么

## 3 这个bmp是什么

## 4 剩下的是什么

花花短裤又尿床了?



# 认识地图文件

——工欲善其事，必先利其器

- strategicregions
- supplyareas
- terrain
- adjacencies.csv
- adjacency\_rules.txt
- airports.txt
- ambient\_object.txt
- buildings.txt
- cities.bmp
- cities.txt
- colors.txt
- continent.txt
- default.map
- definition.csv
- heightmap.bmp
- positions.txt
- provinces.bmp
- railways.txt
- rivers.bmp
- rocketsites.txt
- seasons.txt
- supply\_nodes.txt
- terrain.bmp
- trees.bmp
- unitstacks.txt
- weatherpositions.txt
- world\_normal.bmp

strategicregions: 战略区域文件夹，游戏中的战略区域在此定义  
supplyareas: 补给区域文件夹，补给中心出现后已失去实际功能  
terrain: 地形图像文件夹，游戏里显示的地形就是这里的图像

adjacencies.csv: 用于定义pros之间的特殊通道  
adjacency\_rules.txt: 用于定义游戏中海峡的可通行状态  
airports.txt : 用于定义游戏中机场模型的位置  
ambient\_object.txt: 用于定义游戏中上下边框  
buildings.txt: 用于定义游戏中建筑模型的位置  
cities.bmp: 用于定义游戏中城市模型的位置  
cities.txt: 用于将不同的城市模型分配给世界的不同国家  
colors.txt: 用于定义游戏中城市模型的灯光  
continent.txt: 用于定义游戏中的大洲  
default.map: 执行文件，辅助游戏读取其他文件  
definition.csv: 与provinces.bmp共同构成

地图的基本组成文件

钢丝地图  
都有哪些文件

# 游戏 地图

## 钢丝地图 都有哪些文件

heightmap.bmp: 地形高度图  
positions.txt: 空文件  
provinces.bmp: 地图的基本组成文件  
railways.txt: 开局时的铁路网  
rivers.bmp: 用于定义游戏中河流的位置和粗细  
rocketsites.txt: 用于定义游戏中火箭基地的位置  
seasons.txt: 用于定义游戏中四季的长度和始末  
supply\_nodes.txt: 开局时的补给中心  
terrain.bmp: 用于定义游戏中各地块的地形  
trees.bmp: 用于定义游戏中树的位置  
unitstacks.txt: 用于定义游戏中兵模的位置  
weatherpositions.txt: 用于定义游戏中的天气  
world\_normal.bmp: 法线地图, 决定了地图的3D 渲染中每个像素的确切斜率。用于照明

名称

- strategicregions
- supplyareas
- terrain
- adjacencies.csv
- adjacency\_rules.txt
- airports.txt
- ambient\_object.txt
- buildings.txt
- cities.bmp
- cities.txt
- colors.txt
- continent.txt
- default.map
- definition.csv
- heightmap.bmp
- positions.txt
- provinces.bmp
- railways.txt
- rivers.bmp
- rocketsites.txt
- seasons.txt
- supply\_nodes.txt
- terrain.bmp
- trees.bmp
- unitstacks.txt
- weatherpositions.txt
- world\_normal.bmp

# 2

# 这个txt是什么

## —— 不要用记事本修改



adjacency\_rules.txt



这是干嘛的?

ambient\_object.txt



这又是干嘛的?

supply\_nodes.txt



世上无难事，只要肯放弃！溜了溜了



我们先来看看哪些文件是txt吧！

简单翻译下文件名

## 第一节

 adjacencies.csv	通行
 adjacency_rules.txt	通行规则
 airports.txt	机场
 ambient_object.txt	环境物体
 buildings.txt	建筑
 cities.txt	城市
 colors.txt	颜色
 continent.txt	大洲
 positions.txt	位置
 railways.txt	铁路
 rocketsites.txt	火箭基地
 seasons.txt	季节
 supply_nodes.txt	补给点
 unitstacks.txt	部队位置
 weatherpositions.txt	天气位置

这两个文件看名字就知道肯定有关系，所以就放在一块讲

游戏中存在着大量不按常理的地形，比如非军事区，通行很复杂的海峡，莫名其妙可以跨海水上行走的岛屿，你一定很好奇，这是怎么做到的？



```
1 From;To;Type;Through;start_x;start_y;stop_x;stop_y;adjacency_rule_name;Comment
2 12251;10321;sea;8313;-1;-1;-1;-1;;Andaman
3 7413;1419;sea;5294;-1;-1;-1;-1;;Moluccas
4 10141;1419;sea;5267;-1;-1;-1;-1;;Moluccas
5 480;9431;sea;8621;-1;-1;-1;-1;;Netherlands outlier
6 8857;2284;sea;7617;1560;1009;1556;1016;PANAMA_CANAL;Panama
7 2729;2408;sea;6257;2949;1598;2962;1612;KIEL_CANAL;Kiel
8 5388;803;sea;1155;-1;-1;-1;-1;SUEZ_CANAL;Suez
9 299;211;sea;2282;-1;-1;-1;-1;;Småland-Öland
10 6345;6287;sea;2382;-1;-1;-1;-1;;Skåne-Sjaelland
11 3260;3325;sea;2408;-1;-1;-1;-1;;Sjaelland-Fyn
12 3374;13260;sea;2408;2971;1622;2973;1620;;SdFyn-Langeland
13 13260;13261;sea;2408;2975;1619;2977;1618;;Langeland-Lolland
14 13261;3305;sea;2408;2987;1617;2989;1619;;Lolland-Sjælland
```

这就是定义特殊通道的  
adjacencies.csv文件啦!  
官方还贴心的给了注释

翻译: 起始地块id; 目标地块id; 类型; 起始X; Y坐标; 目标X; Y坐标; 通行规则名字; 注释

官方注释: From;To;Type;Through;start\_x;start\_y;stop\_x;stop\_y;adjacency\_rule\_name;Comment

代码示例: 12251;10321;sea;8313;-1;-1;-1;-1;;Andaman

起始地块和目标地块可以互换, 就是单纯指出是这两个地块相连而已

Type包括海峡sea/非军事区impassable

X, Y坐标是指决定了在两个陆地省份之间穿越海峡而创建的红线的起点和终点位置

如果将X, Y坐标设置为-1, 则游戏将自动计算位置

adjacency\_rule\_name假如不是可通行的运河或者海峡, 即adjacency\_rules.txt需要引用

到, 则可以空置

此文件的最后一行必须是-1;-1;-1;-1;-1;-1;-1;-1;-1, 否则可能出现无法结束读取的bug

此文件必须以 UTF-8 编码

```
219 13196;13314;sea;9092;-1;-1;-1;-1;;Northern Quebec
220 -1;-1;;-1;-1;-1;-1;-1;-1
```

adjacency\_rules.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
#Suez Canal
adjacency_rule =
{
    name = "SUEZ_CANAL"

    contested =
    {
        army = no
        navy = no
        submarine = no
        trade = no
    }
    enemy =
    {
        army = no
        navy = no
        submarine = no
        trade = no
    }
    friend =
    {
        army = yes
        navy = yes
        submarine = yes
        trade = yes
    }
    neutral =
    {
    }
}
```

— □ ×

adjacency\_rules.t

文件(F) 编辑(E) 格式

#Suez Canal

adjacency\_rule =

{

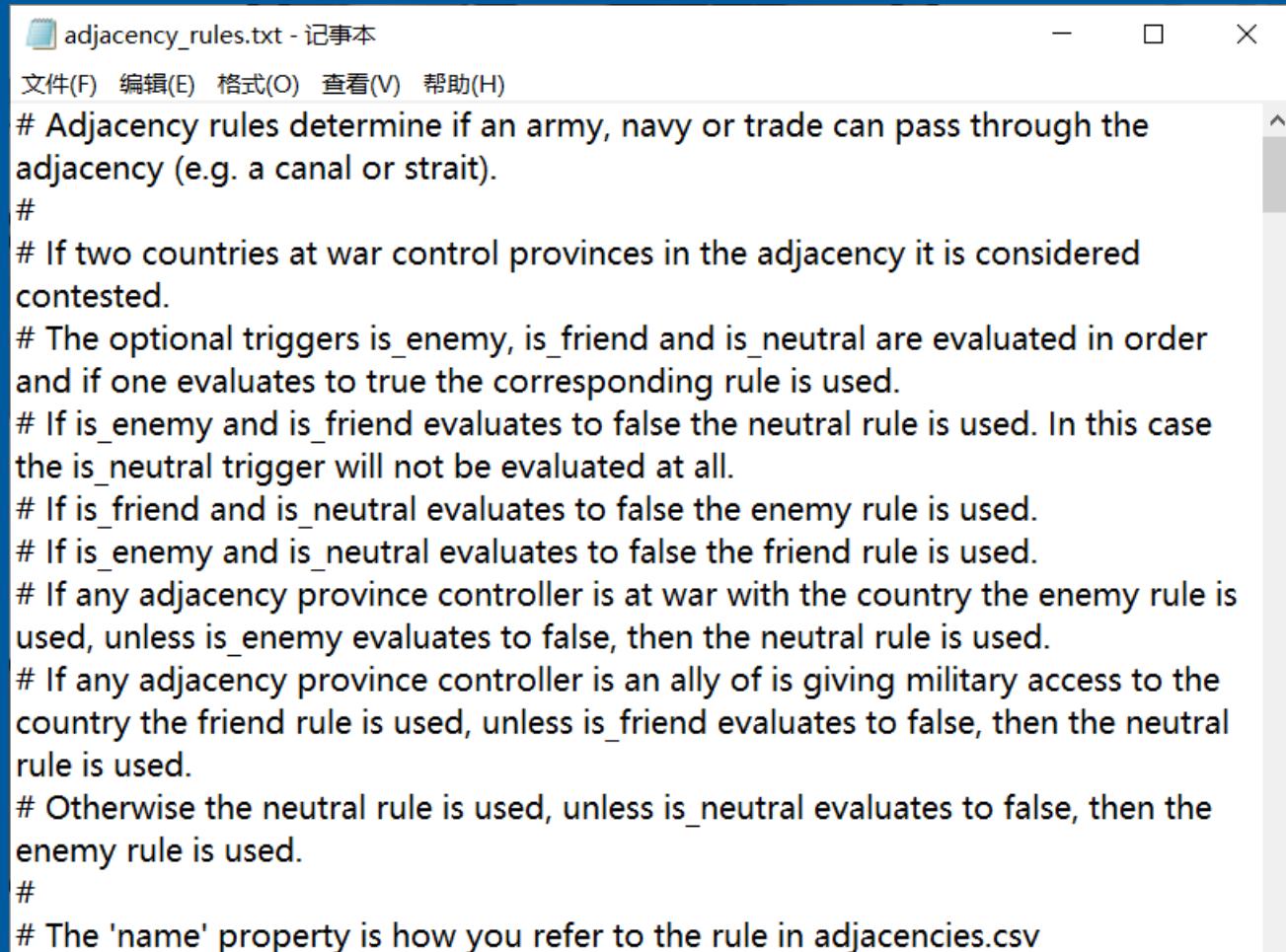
name = "SUEZ\_CANAL"

contested =

adjacency\_rules.txt决定了  
adjacencies.csv文件中海峡/运河这种  
特殊连接的通行规则

常见用法如图





```
adjacency_rules.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
# Adjacency rules determine if an army, navy or trade can pass through the
adjacency (e.g. a canal or strait).
#
# If two countries at war control provinces in the adjacency it is considered
contested.
# The optional triggers is_enemy, is_friend and is_neutral are evaluated in order
and if one evaluates to true the corresponding rule is used.
# If is_enemy and is_friend evaluates to false the neutral rule is used. In this case
the is_neutral trigger will not be evaluated at all.
# If is_friend and is_neutral evaluates to false the enemy rule is used.
# If is_enemy and is_neutral evaluates to false the friend rule is used.
# If any adjacency province controller is at war with the country the enemy rule is
used, unless is_enemy evaluates to false, then the neutral rule is used.
# If any adjacency province controller is an ally of is giving military access to the
country the friend rule is used, unless is_friend evaluates to false, then the neutral
rule is used.
# Otherwise the neutral rule is used, unless is_neutral evaluates to false, then the
enemy rule is used.
#
# The 'name' property is how you refer to the rule in adjacencies.csv
```

这是官方注释

翻译如下：

# 邻接规则决定了军队、海军或贸易是否可以通过邻接区域（例如运河或海峡）。

#

# 如果两个交战国家控制邻接省份，则视为有争议地区。

# 可触发选项按is\_enemy、is\_friend和is\_neutral的顺序进行验证，如果其中一个为真，则使用相应的规则。

# 如果is\_enemy和is\_friend验证为假，则使用is\_neutral进行规则。在这种情况下，根本不会触发is\_neutral验证。

# 如果is\_friend和is\_neutral验证为假，则使用is\_enemy规则。

# 如果is\_enemy和is\_neutral验证为假，则使用is\_friend规则。

# 如果有任何邻接省份控制者与国家交战，则使用is\_enemy规则；除非is\_enemy验证为假，如此则会使用is\_neutral规则。

# 如果任何邻接省份控制者是该国的盟友，并授予该国军事访问权限，则使用is\_friend规则，除非is\_friend评估为false，如此则会使用中立规则。

# 使用中立规则，除非is\_neutral评估为 false，如此则会使用is\_enemy规则。

#

# “name”的内容请您参阅[adjacencies.csv](#)

```
1 adjacency_rule ={
2     name = "" ..... adjacencies.csv文件中定义对象的name
3     通行规则在此 (见下页)
4     required_provinces = {} ..... 涉及到的prov地块, 以空格间隔
5     is_disabled = {} ..... 在什么情况下禁用
6     icon = ..... 图标
7     offset = {} ..... 偏移量, 以空格间隔
8 }
```

对**敌对国家、盟友和中立国家**的通行规则写法与**争议地区**的通行规则相同  
四个通行规则**不可省略**, 四个通行对象**也不可省略**, ppt里是简略的写法不要学

```
1  contested ={..... 争议地区的通行规则
2      army = no.....陆军能否通过yes/no
3      navy = no.....海军能否通过
4      submarine = no.....潜艇能否通过
5      trade = no.....民用舰船能否通过
6  }
7  enemy = {.....对敌对国家的通行规则
8      army = no
9      navy = no
10     submarine = no
11     trade = no
12 }
13 friend = {.....对盟友的通行规则
14     army = no
15     navy = no
16     submarine = no
17     trade = no
18 }
19 neutral = {.....对中立国家的通行规则
20     army = no
21     navy = no
22     submarine = no
23     trade = no
24 }
```

对敌对国家、盟友和中立国家的通行规则写法与争议地区的通行规则相同  
四个通行规则不可省略，四个通行对象也不可省略，假如没有写的话会按官方注释顺延，所以建议写一下

```
type={  
    type="ambient_wind_entity"  
    use_animation=no  
    always_visible=yes  
    object={  
        name="ambient_wind"  
        position={  
            0.000 0.000 0.000  
        }  
        rotation={  
            0.000 0.000 0.000  
        }  
    }  
}  
type={  
    type="ambient_water_entity"  
    use_animation=no  
    scale=10.000000  
    object={  
        name="ambient_water"  
        position={  
            205.110 10.000 1517.910  
        }  
        rotation={  
            0.000 0.000 0.000  
        }  
    }  
    object={
```

ambient\_object.txt决定了游戏上下边框和钢丝LOGO的位置和大小形状

格式为  
依次为

假如需要大量修改  
建议利用nudge模式  
进行制作

```

1;arms_factory;2946.00;11.63;1364.00;0.45;0
1;arms_factory;2948.00;10.05;1356.00;3.47;0
1;arms_factory;2948.00;10.38;1357.00;6.13;0
1;arms_factory;2950.00;11.20;1370.00;2.51;0
1;arms_factory;2945.00;12.03;1368.00;4.07;0
1;arms_factory;2945.00;11.88;1369.00;1.91;0
1;industrial_complex;2950.00;12.05;1361.00;3.12;0
1;industrial_complex;2950.00;12.05;1361.00;0.96;0
1;industrial_complex;2950.00;12.40;1364.00;3.98;0
1;industrial_complex;2944.00;9.50;1357.00;0.36;0
1;industrial_complex;2945.00;9.60;1357.00;3.02;0
1;industrial_complex;2948.00;10.82;1371.00;5.69;0
1;air_base;2948.38;11.60;1359.45;0.15;0
1;naval_base;2954.00;9.70;1370.00;0.98;5579
1;naval_base;2942.00;9.50;1371.00;-2.60;5555
1;naval_base;2944.00;9.50;1360.00;-0.79;5579
1;bunker;2946.00;11.63;1364.00;5.69;0
1;bunker;2945.00;10.03;1361.00;2.07;0
1;bunker;2953.00;10.15;1364.00;6.20;0
1;coastal_bunker;2953.00;9.78;1372.00;0.98;0
1;coastal_bunker;2941.00;9.50;1360.00;-0.79;0
1;coastal_bunker;2940.00;9.50;1368.00;-1.19;0
1;dockyard;2953.00;9.50;1375.00;-3.00;0
1;anti_air_building;2952.00;10.80;1364.00;6.20;0
1;anti_air_building;2952.00;10.80;1364.00;2.59;0
1;anti_air_building;2943.00;10.30;1365.00;0.43;0
1;synthetic_refinery;2946.00;9.73;1358.00;1.63;0
1;nuclear_reactor;2948.00;10.82;1371.00;1.03;0

```

[buildings.txt](#)文件决定了每个state的建筑模型的位置  
在哪个prov

格式为State ID (integer); building ID (string); X position; Y position; Z position; Rotation; [Adjacent sea province \(integer\)](#)

依次为省份id; 建筑代码; X轴; Y轴; Z轴位置;  
旋转; 省份相邻的海洋地块的id  
以分号间隔

[State ID](#)和[Adjacent sea province](#)必须为整数  
[Rotation](#)以弧度完成, 0 是默认状态, 正值按逆时针  
旋转。旋转一圈的数字约等于 6.28, 也即π乘以 2  
[building ID](#)包括[arms\\_factory](#) (军用工厂) ;  
[industrial\\_complex](#) (民用工厂) ; [air\\_base](#) (空军  
基地) ; [naval\\_base](#) (海军基地) ; [bunker](#) (要  
塞) ; [coastal\\_bunker](#) (海军要塞) ; [dockyard](#)  
(船坞) ; [anti\\_air\\_building](#) (防空炮) ;  
[synthetic\\_refinery](#) (合成精炼厂) ;  
[nuclear\\_reactor](#) (核反应堆) 等

假如需要大量修改  
建议利用[nudge模式](#)  
进行制作

```
1={3838 }
2={11751 }
3={11590 }
4={6690 }
5={11386 }
6={6560 }
7={68 }
8={6583 }
9={9421 }
10={3544 }
11={327 }
12={11302 }
13={83 }
14={6548 }
15={9578 }
16={9725 }
17={6698 }
18={6545 }
19={6762 }
20={641 }
21={11875 }
22={6962 }
23={655 }
24={3493 }
25={9751 }
26={590 }
27={11628 }
28={1346 }
29={11540 }
```

[airports.txt](#)决定了游戏内各个state内  
建造机场的prov地块

格式为[State ID = {Province ID}](#)  
依次为[州的ID= {省份/地块ID}](#)

假如需要大量修改  
建议利用[nudge模式](#)  
进行制作

```
1={3838 }
2={9904 }
3={11604 }
4={9679 }
5={6402 }
6={13068 }
7={3211 }
8={6583 }
9={9429 }
10={467 }
11={9404 }
12={9317 }
13={11291 }
14={9495 }
15={11521 }
16={9507 }
17={6698 }
18={6545 }
19={6762 }
20={3770 }
21={958 }
22={6770 }
23={9631 }
24={3493 }
25={625 }
26={590 }
27={9625 }
28={549 }
29={549 }
```

rocketsites.txt决定了游戏内各个state  
内建造火箭基地的prov地块

格式为State ID = {Province ID}  
依次为州的ID= {省份/地块ID}

假如需要大量修改  
建议利用nudge模式  
进行制作

```
types_source = "map/cities.bmp"
pixel_step_x = 2 #2
pixel_step_y = 2 #2

# Western cities
city_group = {
    color_index = 15 # color index in bmp palette
    density = 0.9 # 0.1 # in fraction of pixels. Negative=less dense.

    # The following should be sorted by distance (growing)
    building = {
        distance = 1 # distance to the edge of urban area (in map
pixels)
        mesh = {
            "westerngfx_house_1_1"
        }
    }
    building = {
        distance = 2
        mesh = {
            "westerngfx_house_1_2"
        }
    }
    building = {
        distance = 3
        mesh = {
            "westerngfx_house_1_3"
        }
    }
}
```

[cities.txt](#)用于将不同的城市模型分配给世界的不同国家

```
color = { 4 144 178 }
color = { 181 72 106 }
color = { 63 125 214 }
color = { 54 167 156 }
color = { 231 181 12 }
color = { 57 160 101 }
color = { 194 140 86 }
color = { 199 175 12 }
color = { 69 157 208 }
color = { 117 165 188 }
color = { 206 181 97 }
color = { 195 149 155 }
color = { 94 136 191 }
color = { 219 124 139 }
color = { 146 108 146 }
color = { 69 130 203 }
color = { 59 158 125 }
color = { 220 138 57 }
color = { 191 113 133 }
color = { 147 157 76 }
color = { 26 53 177 }
color = { 125 92 110 }
color = { 121 162 109 }
color = { 182 134 100 }
color = { 201 97 74 }
color = { 116 92 39 }
color = { 33 101 137 }
color = { 129 177 125 }
color = { 110 88 151 }
```

colors.txt决定了游戏内城市的灯光

格式为

color = { R G B }

RGB代表的是该颜色对应的RGB颜色表中的数字

三个数字之间以分号间隔

```
continents = {  
    europe  
    north_america  
    south_america  
    australia  
    africa  
    asia  
    middle_east  
}
```

continent.txt决定了游戏中总共有哪些大洲

格式为

```
continents = {  
    大洲A  
    大洲B  
    大洲C  
}
```

大洲在该文件中的顺序将被自动视为该大洲的数字代号如地块xx所处大洲位置被填写为1,则会被游戏自动读取为处于大洲A

大洲名字要去本地化定义

假如需要大量修改  
建议利用nudge模式  
进行制作

没有实际用途  
随着更新已经失去存在意义  
甚至P社自己已经把代码删光了  
蠢驴因为屎山代码不敢删文件

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
3 4 6521 11444 3312 6282
2 5 6282 349 3340 9388 321
2 5 6282 6334 9334 9361 11372
1 4 6332 6402 3380 362
1 3 6332 11245 6375
3 4 6521 9428 6441 514
2 5 6521 375 11219 11276 321
3 4 6521 3499 9456 3522
1 4 3522 9375 526 6377
1 4 6377 6263 6325 9347
2 3 9347 6257 6389
3 7 6521 11505 9496 3572 6595 552 9570
2 3 6282 3207 9496
1 7 9347 3368 3231 11366 332 13263 399
1 3 6377 3271 3326
2 4 9347 6325 6349 3326
1 2 3326 241
1 7 6469 6622 11346 9281 11264 11360 241
1 2 6469 587
3 2 6469 495
1 3 495 6535 11388
1 4 11388 247 309 3326
1 6 495 6535 9509 405 3395 6377
1 4 6469 6570 529 11494
2 4 11494 3423 9575 11531
3 3 11494 6444 6488
2 4 11531 11547 11560 6488
3 4 9517 6568 589 6488
2 5 11531 6568 589 6488
```

railways.txt文件决定了游戏开局时即存在的每条铁路

格式为Level; Amount of provinces;  
List of provinces

依次为铁路等级; 铁路经过的地块数量; 铁路经过的所有地块id  
以空格间隔

Level取值范围为1-5, 必须为整数  
错误的铁路可能导致游戏崩溃

假如需要大量修改  
建议利用nudge模式  
进行制作

```
1 67
1 101
1 121
1 122
1 158
1 191
1 192
1 241
1 271
1 306
1 310
1 321
1 351
1 362
1 377
1 383
1 385
1 391
1 394
1 399
1 402
1 418
1 476
1 485
1 492
1 495
1 512
1 514
1 516
```

supply\_nodes.txt文件决定了游戏开局时即存在的每个补给中心

格式为Level; Province

依次为补给中心等级; 所在地块id  
以空格间隔

Level必须为1

假如需要大量修改  
建议利用nudge模式  
进行制作

```
winter = {
    start_date=00.12.01
    end_date=00.02.10

    hsv_north= { 0 0.1 1 }
    colorbalance_north= { 0.9 0.9 1 }

    hsv_center= { 0.0 1.0 1.0 }
    colorbalance_center= { 1.0 1.0 1.0 }

    hsv_south= { 0.0 1.0 1.0 }
    colorbalance_south= { 1.0 1.0 1.0 }

}

spring = {
    start_date=00.03.10
    end_date=00.04.22

    hsv_north= { 0 0.1 1 }
    colorbalance_north= { 0.9 0.9 1 }

    hsv_center= { 0.0 1.0 1.0 }
    colorbalance_center= { 1.0 1.0 1.0 }

    hsv_south= { 0.0 1.0 1.0 }
    colorbalance_south= { 1.0 1.0 1.0 }

}
```

[seasons.txt](#)决定了游戏内各个季节的起始和结束时间等

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

2;0;3429.00;9.85;1699.00;0.00;0.30  
3;0;3367.00;10.40;1807.00;0.00;0.28  
4;0;3168.00;9.50;1858.00;0.00;0.88  
5;0;2891.00;9.50;1626.00;0.00;0.82  
6;0;3579.00;10.60;1758.00;0.00;0.26  
7;0;3274.00;10.35;1829.00;0.00;0.32  
8;0;3311.00;11.15;1677.00;0.00;0.18  
9;0;1964.00;9.50;232.00;0.00;1.24  
10;0;2999.00;12.12;1768.00;0.00;0.28  
11;0;3076.00;11.85;1872.00;0.00;0.34  
12;0;3504.00;9.80;1677.00;0.00;0.24  
13;0;1361.00;9.50;303.00;0.00;1.34  
14;0;2963.00;9.87;1723.00;0.00;0.24  
15;0;3288.00;10.92;1918.00;0.00;0.36  
16;0;5263.00;9.50;1552.00;0.00;1.18  
17;0;3075.00;9.75;1567.00;0.00;0.20  
18;0;3369.00;10.15;1796.00;0.00;0.24  
19;0;3881.00;9.50;651.00;0.00;1.12  
20;0;3132.00;9.50;1203.00;0.00;0.68  
21;0;2949.00;9.50;1816.00;0.00;0.44  
22;0;3166.00;9.50;2005.00;0.00;0.34  
23;0;3500.00;9.80;1709.00;0.00;0.24  
24;0;5231.00;9.50;500.00;0.00;0.98  
25;0;5348.00;9.50;1275.00;0.00;1.18  
26;0;3386.00;10.37;1794.00;0.00;0.32  
27;0;2955.00;13.20;1754.00;0.00;0.34  
28;0;4782.00;9.50;474.00;0.00;0.92  
29;0;4179.00;9.50;636.00;0.00;0.80

unitstacks.txt文件决定了每个省份内的士兵模型和胜利点的位置

格式为Province ID; Type; X position; Y position; Z position; Rotation; Offset

依次为地块id; 类型; X轴; Y轴; Z轴位置; 旋转; 偏移  
以分号间隔

Type是一个介于 0 到 38 之间的整数, 用于分配用途。详见下页

Rotation以弧度完成, 0 是默认状态, 正值按逆时针旋转。旋转一圈的数字约等于 6.28, 也即 $\pi$ 乘以 2

Offset用于动画, 以使所有兵模动画不是完全相同, 而是具有延迟。

假如需要大量修改  
建议利用nudge模式  
进行制作

ID	Internal name	Notes	ID	Internal name	Notes
0	Standstill		21	Standstill RG	RG stands for regrouping
1	Moving 0	Different numbers represent different needed levels of rotation	22	Moving 0 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
2	Moving 1	Different numbers represent different needed levels of rotation	23	Moving 1 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
3	Moving 2	Different numbers represent different needed levels of rotation	24	Moving 2 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
4	Moving 3	Different numbers represent different needed levels of rotation	25	Moving 3 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
5	Moving 4	Different numbers represent different needed levels of rotation	26	Moving 4 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
6	Moving 5	Different numbers represent different needed levels of rotation	27	Moving 5 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
7	Moving 6	Different numbers represent different needed levels of rotation	28	Moving 6 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
8	Moving 7	Different numbers represent different needed levels of rotation	29	Moving 7 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
9	Attacking		30	Disembark 0 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
10	Defending		31	Disembark 1 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
11	Disembarck 0	Different numbers represent different needed levels of rotation	32	Disembark 2 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
12	Disembarck 1	Different numbers represent different needed levels of rotation	33	Disembark 3 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
13	Disembarck 2	Different numbers represent different needed levels of rotation	34	Disembark 4 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
14	Disembarck 3	Different numbers represent different needed levels of rotation	35	Disembark 5 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
15	Disembarck 4	Different numbers represent different needed levels of rotation	36	Disembark 6 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
16	Disembarck 5	Different numbers represent different needed levels of rotation	37	Disembark 7 RG	Different numbers represent different needed levels of rotation. RG stands for regrouping
17	Disembarck 6	Different numbers represent different needed levels of rotation	38	Victory point	
18	Disembarck 7	Different numbers represent different needed levels of rotation			
19	Ship in port				
20	Ship in port moving				

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
1;2781.24;9.90;1571.49;small
1;2746.16;10.10;1548.63;small
2;2734.00;11.30;1576.00;small
2;2763.00;11.38;1612.00;small
3;2718.15;11.20;1677.80;small
3;2739.81;12.60;1645.90;small
4;2677.00;10.55;1604.00;small
4;2657.00;10.93;1577.00;small
5;2881.23;9.80;1535.80;small
5;2900.76;9.60;1566.91;small
6;2970.24;10.80;1565.60;small
6;2950.00;10.03;1568.00;small
7;2941.69;10.50;1516.84;big
7;2992.11;11.30;1492.10;big
8;3006.41;9.80;1553.92;big
8;3032.77;9.70;1579.70;small
8;3067.28;10.40;1542.92;small
9;3124.16;9.50;1706.70;small
9;3114.52;9.50;1793.31;small
10;3026.86;10.50;1664.72;small
10;3039.88;10.40;1717.47;small
11;2955.23;12.90;1777.71;big
11;2933.59;12.40;1727.03;big
12;3310.06;10.80;1940.46;small
12;3396.48;10.00;1919.81;small
13;3235.27;10.40;1816.89;big
13;3269.13;10.30;1767.68;big
14;3832.00;10.30;1761.00;big
14;3831.88;10.30;1760.88;big
```

weatherpositions.txt 文件决定了云等天气模型的位置。

格式为strategic region ID;X position;Y position;Z position;size  
依次为战略区域id; X轴; Y轴; Z轴  
位置, 模型大小  
以分号间隔

假如需要大量修改  
建议利用nudge模式  
进行制作

# 3

# 这个bmp是什么

——用ps也比画图强

地图的宽度和高度由provinces.bmp决定，并且其像素数值都要是256的倍数  
需要注意，其他图片文件也全都是必须与provinces.bmp大小相同的  
一般来说，其他图片文件也都是以provinces.bmp为基础进行制作和修改的

-  cities.bmp
-  heightmap.bmp
-  provinces.bmp
-  rivers.bmp
-  terrain.bmp
-  trees.bmp
-  world\_normal.bmp
-  definition.csv

城市  
高度图  
省份  
河流  
地形  
树  
世界 普通  
定义

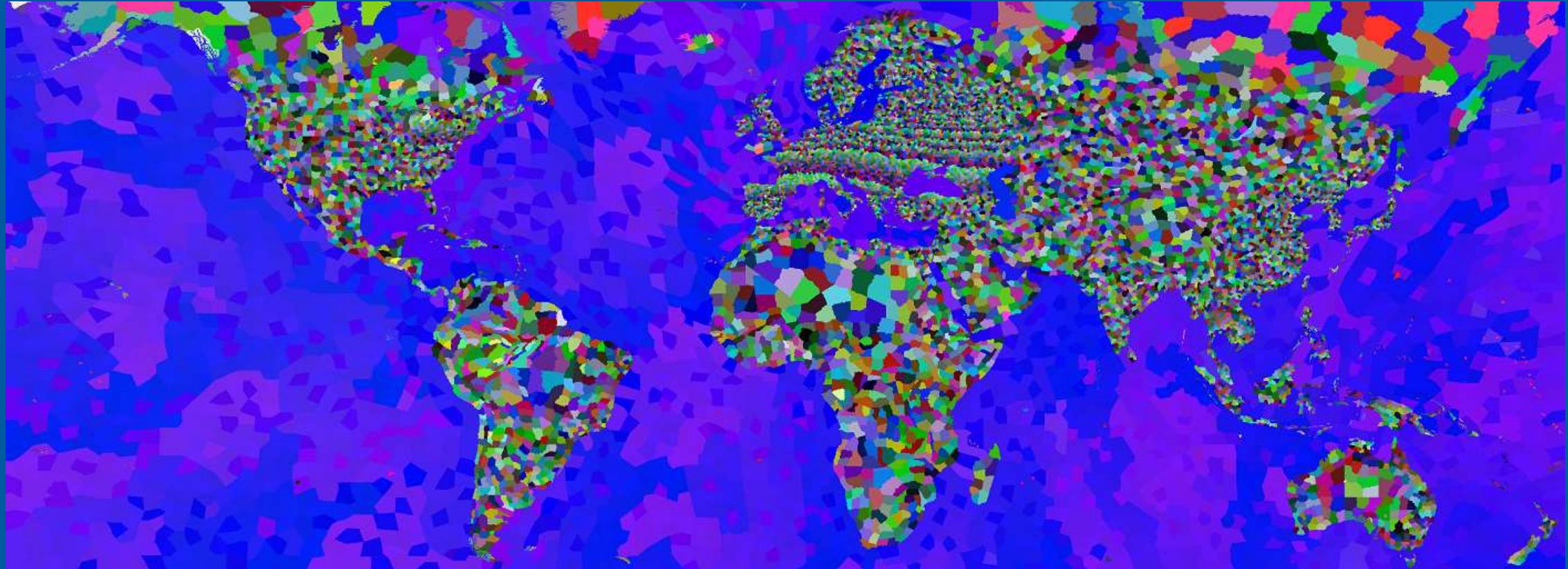
定义文件definition.csv  
因为和  
省份文件provinces.bmp  
必须搭配使用，所以放在一起讲

都有哪些bmp文件呢？

那么什么是provinces.bmp文件呢？我们一起来看看吧

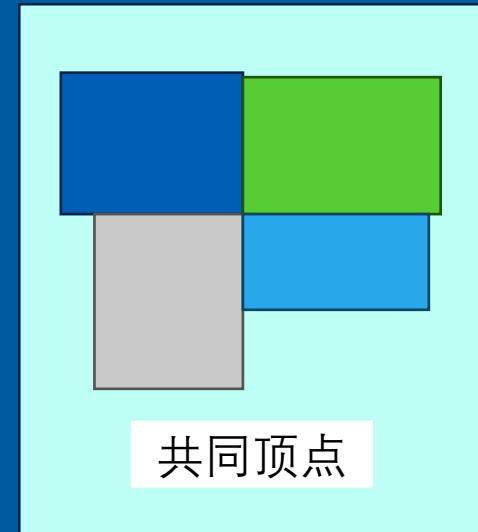


这是什么鬼？？？这不是一堆色块吗？马赛克吗？错！我们缩小看看



现在明白了吧，游戏里的一个个单独的地块就是在这个文件里定义了形状和位置，所以说provinces.bmp是整个maps文件夹的基石完全不为过。这也代表着，我们假如想要修改游戏中的地块大小和形状，动这个文件就可以了。

比如说，我们需要扩大左图中心那块草绿色的prov，让他不那么方方正正的同时，也让它更大一点。那么我们就可以打开画图软件，直接对它的形状和大小进行修改。在修改完毕后，保存文件就可以了。



共同顶点



虽然可以  
但是我并不推荐  
使用系统自带的  
画图软件  
推荐：PS

注意

1. 地块不能随意增添或删除，如有需要，请学习下一页
2. 修改地块的时候注意不要留白，也不要将地块涂没（删除）
3. 不推荐让非沿海地块分成好几块，地块也不要太大或太小
4. 不推荐让四个地块有共同的顶点，游戏会报错
5. 修改时尽量不要让原本不相邻/相邻的地块变得相邻/不相邻



但是你仍然可能好奇，  
左边的像素图是怎么变成游戏中右边的地块的呢？  
这就要打开我的文本啦！终于啊！  
我们前文中被反复鞭尸提及的  
定义文件definition.csv要派上用场了！  
瞪大眼睛仔细看！



	A	B	C
1	0;0;0;land;false;unknown;0		
2	1;230;81;119;lake;false;lakes;7		
3	2;0;0;55;land;false;forest;1		
4	3;0;0;205;land;false;forest;1		
5	4;0;0;232;sea;true;ocean;0		
6	5;0;2;240;sea;false;ocean;0		
7	6;0;3;20;land;false;forest;6		
8	7;0;3;170;land;false;forest;1		
9	8;0;3;225;land;false;hills;1		
10	9;0;4;248;sea;false;ocean;0		
11	10;0;6;85;land;false;forest;1		
12	11;0;6;135;land;false;forest;1		
13	12;0;6;190;land;false;plains;6		
14	13;0;8;244;sea;false;ocean;0		
15	14;0;155;land;false;forest;1		

Excel下打开

看不懂?  
数字是什么鬼?  
单词是干嘛的?  
没关系  
现在呢→

1	0;0;0;0;land;false;unknown;0
2	1;230;81;119;lake;false;lakes;7
3	2;0;0;55;land;false;forest;1
4	3;0;0;205;land;false;forest;1
5	4;0;0;232;sea;true;ocean;0
6	5;0;2;240;sea;false;ocean;0
7	6;0;3;20;land;false;forest;6
8	7;0;3;170;land;false;forest;1
9	8;0;3;225;land;false;hills;1
10	9;0;4;248;sea;false;ocean;0
11	10;0;6;85;land;false;forest;1
12	11;0;6;135;land;false;forest;1

Notepad++下打开

地块编号, RGB颜色, 是否陆地, 是否沿海, 地块类型, 所处大洲

0;0;0;0	land	false	unknown	0
1;230;81;119	lake	false	lakes	7
2;0;0;55	land	false	forest	1
3;0;0;205	land	false	forest	1
4;0;0;232	sea	true	ocean	0

地块编号	RGB颜色	是否陆地	是否沿海	地块类型	所处大洲
7;	0;3;170;	land;	false;	forest;	1

我们一一解释：

**地块编号**：每个地块独一无二的编号，游戏里地块的id即是，**每个地块唯一**

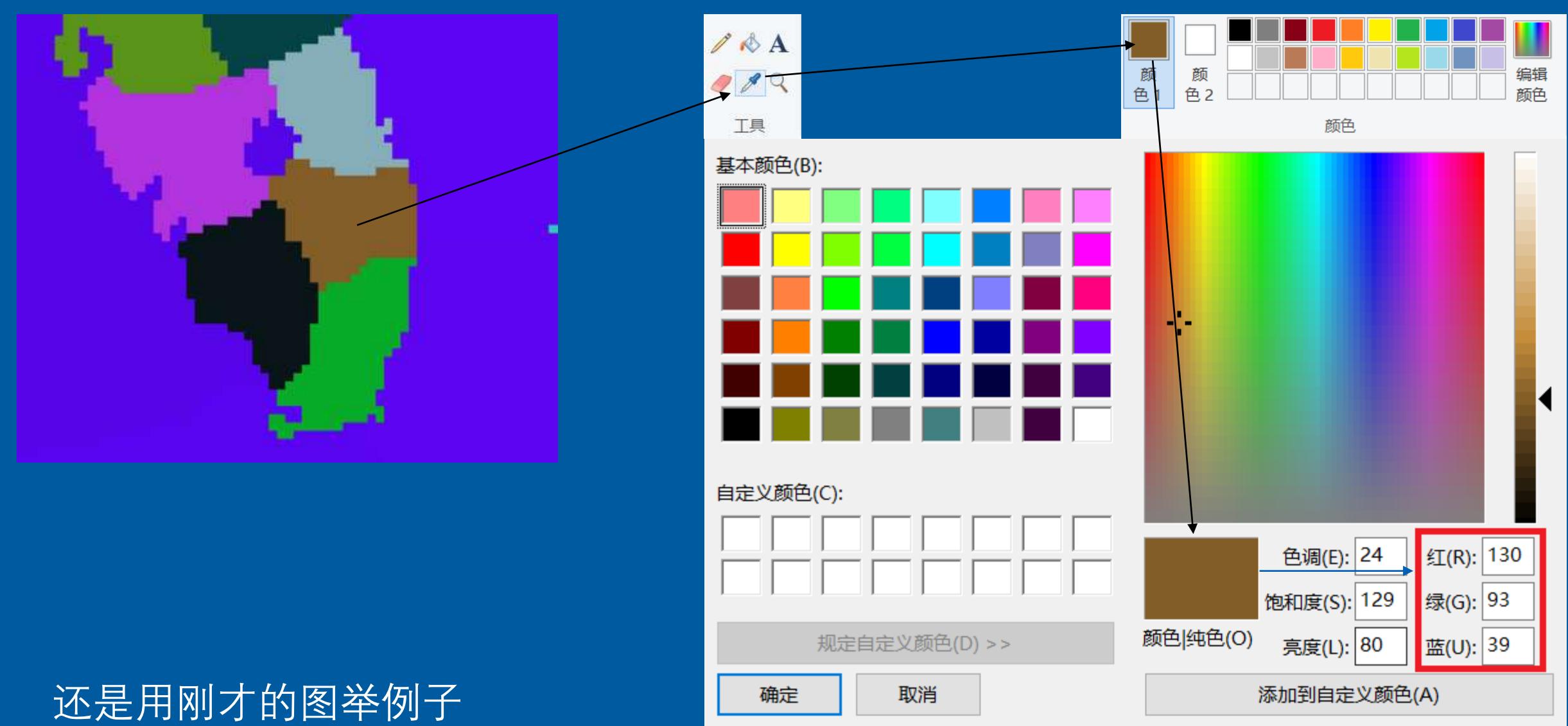
**RGB颜色**：地块在[provinces.bmp](#)之中对应的色块的颜色，**每个地块唯一**

**是否陆地**：这块地是**陆地**[land](#)/海洋[sea](#)/湖泊[lake](#)

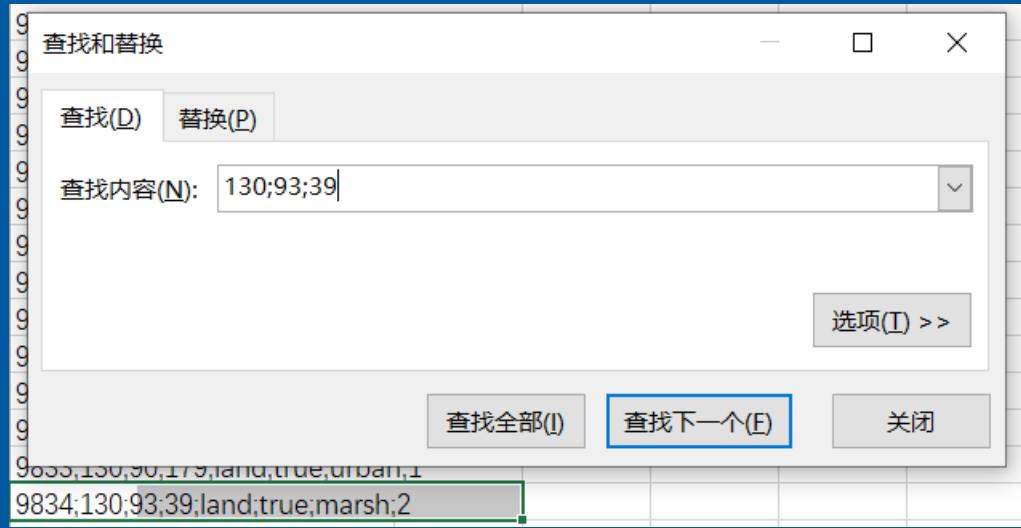
**是否沿海**：这块地**沿海**[true](#)/不沿海[false](#)

**地块类型**：这块地是什么类型？地块类型详见[Hearts of Iron IV/common/terrain](#)

**所处大洲**：这块地在哪个大洲？大洲详见[Hearts of Iron IV /map/continent.txt](#)



还是用刚才的图举例子  
我们看看这个地块的颜色是什么  
画图就可以做到，很轻松



找到了！  
根据数据来看，地块id是9834  
陆地省份，沿海，沼泽地形，  
处于continent.txt文件里排第二位的大洲  
打开游戏  
控制台tdebug看看数据  
完全一致！

```
continents = {  
    europe  
    north_america  
    south_america  
    australia  
    africa  
    asia  
    middle_east  
}
```



以此类推，其他的内容修改，如地貌，城市模型，高低图

也是找到对应的bmp文件修改就可以了

不过这些文件大多数需要专业工具进行更便利的修改就是了

terrain.bmp是一个控制地形分配和纹理BMP文件

trees.bmp控制地图上树木的摆放和密度

rivers.bmp控制地图上的河流

heightmap.bmp决定了地图的3D网格

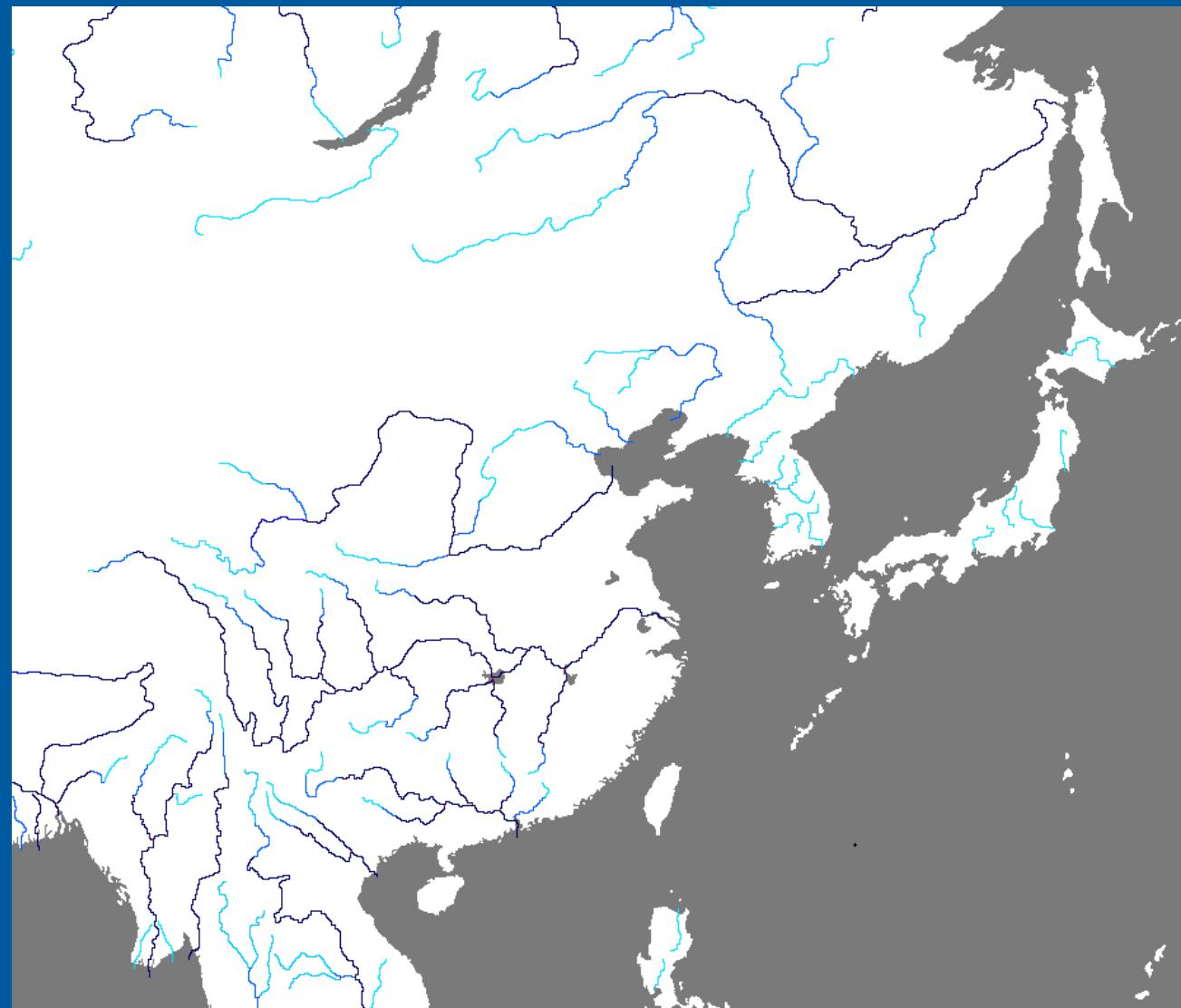
world\_normal.bmp定义了3D凹凸贴图

就用河流文件rivers.bmp做范例吧

同样是画图就可以打开 (不推荐)



黄河，长江，无需多言



Index	颜色	意义
0	(0, 255, 0)	河流的源头
1	(255, 0, 0)	流入源。用于把多条河流合并为一条干流。
2	(255, 252, 0)	流出源。用于把一条河流分流为多条支流。
3	(0, 225, 255)	外观最窄的河流
4	(0, 200, 255)	外观较窄的河流
5	(0, 150, 255)	
6	(0, 100, 255)	外观较宽的河流
7	(0, 0, 255)	
8	(0, 0, 225)	
9	(0, 0, 200)	
10	(0, 0, 150)	
11	(0, 0, 100)	外观最宽的河流



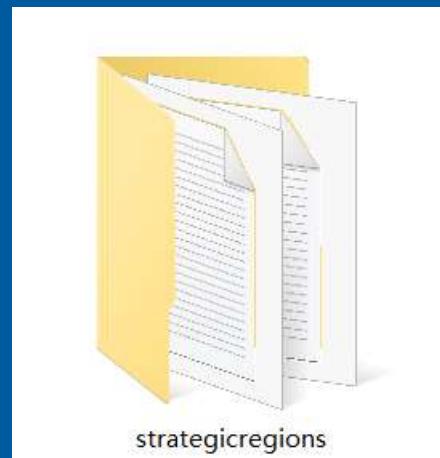
注意：

- 1.文件[rivers.bmp](#)需要保存为索引模式并保存为8位位图（.BMP格式）
- 2.每一条河都必须被唯一标识，即只有一个起点（绿色/黄色）或一个终点（红色）
- 3.地块的边界最好明确的以河为界或者远离河流
- 4.每条河流必须只有一格像素点的宽度，也必须连续不能断开

# 4

# 剩下的是什么

——还有点零碎的



游戏中的战略区域在此划定  
与states文件夹类似  
每个战略区域都被一个txt单独划分



随着 1.11 和 No Step Back 的发布  
补给区域被弃用  
取而代之的是supply\_nodes.txt和railways.txt定义  
现随更新已经失去存在意义



游戏中地形的图像  
一般不会涉及到此处的修改

colormap\_x.dds文件定义了应用于地图的整体颜色色调

colormap\_rgb\_cityemissivemask\_a.dds用于世界的整体色彩和改变城市灯光的不透明度  
colormap\_water\_0.dds用于世界海洋的着色

```
1 strategic_region={ ..... 代码抬头
2   id=1 ..... 战略区域id
3   name="STRATEGICREGION_1" ..... 战略区域名称 (本地化定义)
4   color = { <r>, <g>, <b> } ..... 战略区域的颜色 (未定义即随机)
5   provinces={ ..... 战略区域包含的prov (示例代表id为2024的prov属于此战略区域)
6     2024
7   }
8   weather={ ..... 天气
9     见下一章
10  }
11 }
```

现在我们看到的是一个  
尽可能完整的战略区域

每个战略区域id唯一, name同样唯一  
而每个prov只能存在于一个战略区域内  
也不要让prov不属于任何战略区域

颜色和天气并非必须的  
而其他的代码则是必须的

战略区域手写会累死你  
假如需要大量修改  
建议利用nudge模式  
进行制作

```
1 weather={ ..... 天气 (续上篇)
2   period={
3     between={ 0.0 30.0 } ..... 决定天气在哪个时期适用, 数字顺序为{日 月}, 从零开始
4     temperature={ -6.0 12.0 } ..... 指定该时间段的最高温度和最低温度
5     no_phenomenon=0.500 ..... 决定了战略区域中每个省份无现象的几率
6     rain_light=1.000 ..... 小雨
7     rain_heavy=0.150 ..... 大雨
8     snow=0.200 ..... 降雪
9     blizzard=0.000 ..... 暴风雪
10    arctic_water=0.000 ..... 洪水
11    mud=0.300 ..... 泥泞
12    sandstorm=0.000 ..... 沙尘暴
13    min_snow_level=0.000 ..... 最低降雪等级
14  }
15 }
```

此外, [/Hearts of Iron IV/map/weatherpositions.txt](#) 文件决定了云等天气模型的位置。

格式为[strategic region ID;X position;Y position;Z position;size](#)

我们组内

再见！

See you next.

See you next.