

csファイルの構造について。
とてあえずメインゲームに関

Main
実際に呼び出す部分

Transition
場面遷移。Scenesディレクトリ
以下に各csファイルを持つ。

Generic
汎用関数・変数・定数。
あらゆるファイルから呼び出
せる。

Character
当たり判定を持つ一単位。
Objectを継承したクラス。

Object
当たり判定を持たない一単位。

UI
HPなどプレイヤーに情報を伝
えるもの。最前面に表示。

Stage
ステージをリストとして保持。
各ステージは、「所持マップ
数・始まりマップ番号・始まり
座標」などの情報を持つ。

Map
リスト構造。「背景・地形・前
景」の、オリジナルエディタを
用いて作ったマップチップに
よるマップを保存。

Event
リスト構造。ステージイベント。
Character.Bossを呼び出す。
「マップ番号・座標・起動条
件・内容」。内容に関しては
Editクラスが必要。

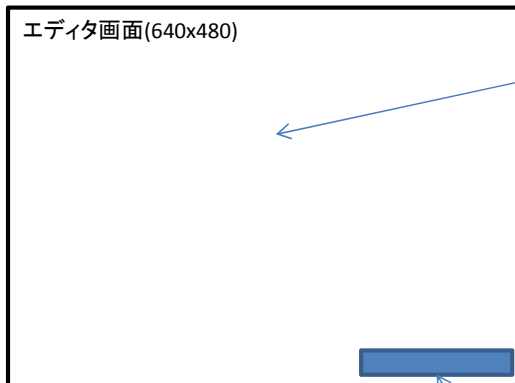
Unit
リスト構造。敵配置。「マップ
番号・座標・敵番号」。
Character.Enemyを呼び出す。

Stageフォルダ構造

```

Stage000
├─info.txt (ステージ名・マップ数・BGMなど)
├─event.css
├─unit.css
├─map000.dat (マップエディタによって生成)
├─map001.dat
├─...
└─map000.dat

```



エディタ画面(640x480)

編集マッピングの表示。
区画ごとに基盤の目状に区切られる。

右クリックごとに、「背景・地形・前景・敵配置・全て合わせて表示」モードに変更。全て合わせて表示モードではマッピングチップ変更不可で、マウスホイールを回すことで表

ドラッグによって表示位置をずらすことができる。

上段には編集マッピング番号と使用チップセット番号を表示。
中段にはカーソル位置の座標と現在のマッピングチップ番号を表示。
下段では「Load・Save・New」が選択可能。
Loadを押したあと、読み込むマッピング番号を入力してEnterで読み込む。
Saveを押したあと、「セーブしました」のメッセージを出し、右述の仕様でファイルを書き出す。

作成されるファイル(map000.dat など)
1行目...マッピング名(20文字まで)
2行目...使用する地形チップセット番号
3行目...使用する背景チップセット番号
4行目...使用する前景チップセット番号
5行目...Xサイズ
6行目...Yサイズ(以下、nと表記)
7行目~7+n-1行目...カンマ区切りで地形マッピングチップ番号
7+n行目~7+2n-1行目...カンマ区切りで背景マッピングチップ番号

↓↓↓↓↓↓↓↓↓↓作成されるファイル

(例)↓↓↓↓↓↓↓↓↓↓

テストマップ

0

0

0

3

2

0,0,1

1,1,1

2,2,2

1,1,1

1,1,1

1,1,1