

# コード→状態遷移表→状態遷移図→ 仕様までのスペックアウトの実際

数年経ち、「状態って何スカ？」と言っていた新人A君は、知識を身に着け「中堅Aさん」と呼ばれるように成長していた。平和なある日、新人C君が質問にやってきてこう言ったのでした。



新人C君

「Aさん。研修で『このコードを解析して資料化しなさい』って言われたのですが、どうやって手を付けたらいいのかわからなくて…  
ヒントでいいので何か情報もらえませんか？」



現・中堅Aさん  
(元・新人A)

なぜ自分に質問に来たのだろうか？

と怪訝に思いながらコードを見るとそこには！

```
/*  
*  
* AUTHOR: Pinkuno,Usagi  
*  
***/
```

自分の名前がヘッダーコメントに記載されていたのだ。  
しかし、研修時の内容なんてもう覚えていない…  
一度自分でスペックアウトして、そこからヒント出すか…

# HOW TO スペックアウト

Step  
1

## スペックアウトの準備

- 関連資料の整理
- ソースファイルの整理



Step  
2

## スペックアウトと スペックアウト資料の作成

- 状態遷移表の作成



Step  
3

## 状態遷移表からダイアグラムの作成



Step  
4

## スペックアウト資料から仕様の作成

Step  
2

# スペックアウトと スペックアウト資料の作成

- 状態遷移表の作成

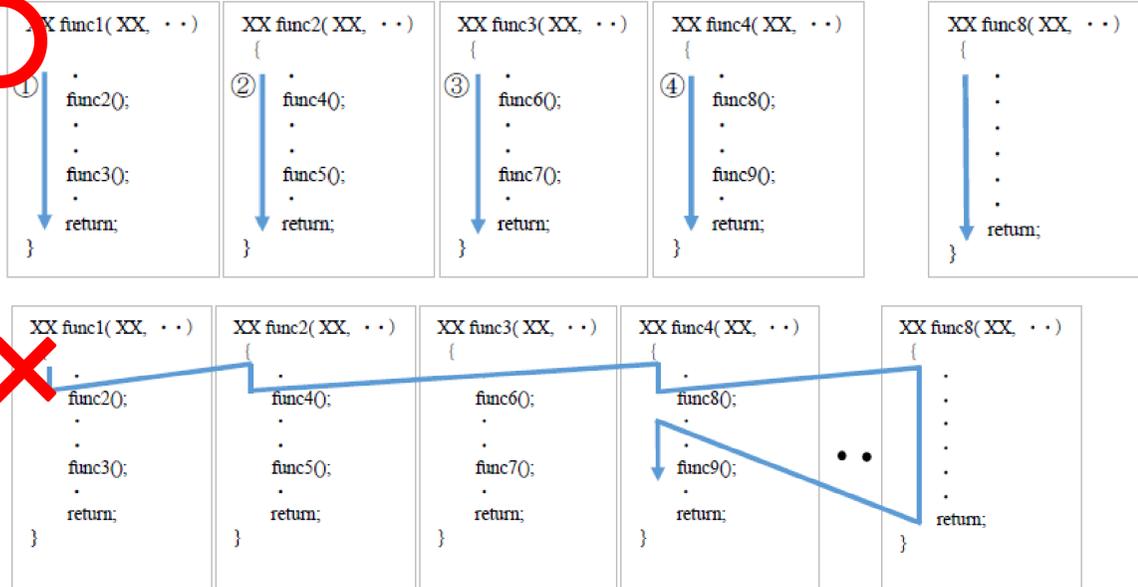
Q: 縦読みって?

## ソースコード抜粋(本戦)

```
switch(state)
{
case WAITE:
    wsprintf(language, "↓↓手を選んで下さい↓↓");
    wsprintf(gT_word, "じゃんけん");
    SelectObject(hdc, hFont2);
    TextOut(hdc, 550, 260, gT_word, strlen(gT_word));

    SetBkMode(hdc, OPAQUE);
    SelectObject(hdc, hFont1);
    SetTextColor(hdc, RGB(255, 255, 240)); // ivory
    SetBkColor(hdc, RGB(255, 99, 71)); // tomato
    TextOut(hdc, 660, 560, language, strlen(language));
    break;
case WAITE2:
    wsprintf(language, "↓↓手を選んで下さい↓↓");
    wsprintf(gT_word, "あいこで");
    SelectObject(hdc, hFont2);
    TextOut(hdc, 550, 260, gT_word, strlen(gT_word));

    SetBkMode(hdc, OPAQUE);
    SelectObject(hdc, hFont1);
    SetTextColor(hdc, RGB(255, 255, 240)); // ivory
    SetBkColor(hdc, RGB(255, 99, 71)); // tomato
    TextOut(hdc, 660, 560, language, strlen(language));
    break;
}
```



```
if((number == 5) && (gT_victory < gT_defeat))
{
    state = END;
    personid = WIN;
    wsprintf(gT_word, "YOU LOSE ! ");
}
else if((number == 5) && (gT_victory > gT_defeat))
{
    state = END;
    personid = LOSE;
    wsprintf(gT_word, "YOU WIN ! ");
}
else{
}
break;
```

ソースコードは縦読みで!

状態遷移図へ

イベント	WM_PAINT	WM_COMMAND	WM_LBUTTONDOWN
状態	—	—	—
WAITE	テキスト表示, 描画など	Win_Lose_Drawへ プログラムの手の判定 勝敗数の記録	なし
WAITE2	テキスト表示, 描画など	Win_Lose_Drawへ プログラムの手の判定 勝敗数の記録	なし
BATTLE	テキスト表示, 描画など 勝敗の判定	なし	勝敗数の判定 (2勝2敗なら) とキャラ・音楽の変更?
BATTLE2	—	—	WAITE2へ
EVEN	テキスト表示, 描画など 勝敗の判定	なし	なし
EVEN2	—	—	WAITE2へ
END	テキスト表示, 描画など	なし	なし
FIN	—	—	STAT_ENDへ (stat)
	テキスト表示, 描画など 処理の終了 (return 0)	なし	フラグの設定, 破棄, 処理の終了 (return 0)



Step  
4

# スペックアウト資料から仕様の作成

要求: 本戦において2勝2敗になると出てくるボスの出現条件をユーザが2勝した時点で出現するように変更したい

```
switch(state)
{
case WAITE:
wprintf(language, "↓↓↓手を選んで下さい↓↓↓");
wprintf(gT_word, "じゃんけん");
SelectObject(hdc, hFont2);
TextOut(hdc, 550, 260, gT_word, strlen(gT_word));

SetBkMode(hdc, OPAQUE);
SelectObject(hdc, hFont1);
SetTextColor(hdc, RGB(255, 255, 240)); //ivory
SetBkColor(hdc, RGB(255, 99, 71)); //tomato
TextOut(hdc, 660, 560, language, strlen(language));
break;

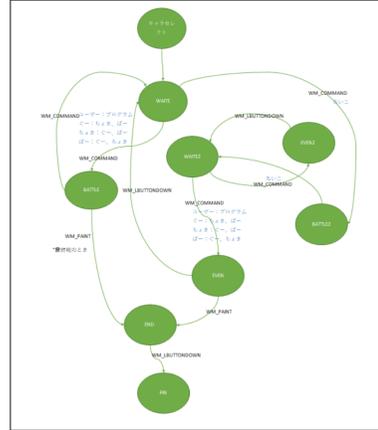
case WAITE2:
wprintf(language, "↓↓↓手を選んで下さい↓↓↓");
wprintf(gT_word, "あいこで");
SelectObject(hdc, hFont2);
TextOut(hdc, 550, 260, gT_word, strlen(gT_word));

SetBkMode(hdc, OPAQUE);
SelectObject(hdc, hFont1);
SetTextColor(hdc, RGB(255, 255, 240)); //ivory
SetBkColor(hdc, RGB(255, 99, 71)); //tomato
TextOut(hdc, 660, 560, language, strlen(language));
break;
}
```

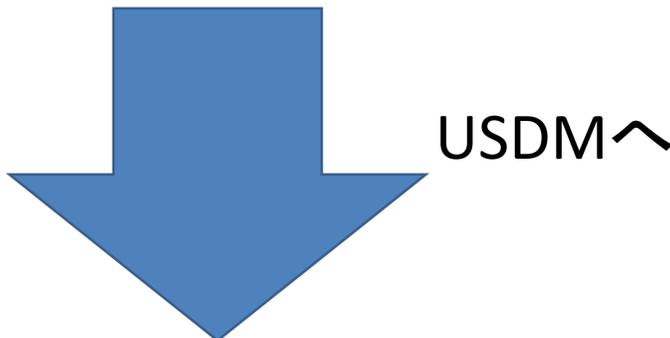
ソースコード

イベント	WM_PAINT	WM_COMMAND	WM_LBUTTONDOWN
状態	—	Win_Lose_Drawへ	—
WAITE	テキスト表示, 描画など	プログラムの手の判定 勝敗数の記録	なし
WAITE2	テキスト表示, 描画など	プログラムの手の判定 勝敗数の記録	なし
BATTLE	ENDへ	—	WAITEへ
BATTLE2	テキスト表示, 描画など 勝敗の判定	なし	勝敗数の判定 (2勝2敗なら) とキャラ・音楽の変更?
EVEN	ENDへ	—	WAITEへ
EVEN2	テキスト表示, 描画など 勝敗の判定	なし	勝敗数の判定 (2勝2敗なら) とキャラの変更?
END	テキスト表示, 描画など	なし	なし
FIN	テキスト表示, 描画など 処理の終了 (return 0)	なし	STAT_ENDへ (stat) フラグの設定, 破棄, 処理の終了 (return 0)

状態遷移表



ダイアグラム



要求	NEK	理由
	NEK	本戦において2勝2敗になると出てくるボスの出現条件をユーザが2勝した時点で出現するように変更したい
	理由	ユーザが2勝しないと出現しないため、1勝4敗の場合、ボスが出現せずにゲームが終わってしまうため
要求	NEK01	表示キャラクター、背景をボス(ねこまっちょ)仕様にする条件を、ユーザが2勝かつ2敗の場合という条件から、ユーザが2勝の場合に変更する。ボス表示に移行させているのは、状態 "BATTLE" と "EVEN" の2か所から移行しているため、両方の条件を変更する必要がある。
	理由	ボスに切り替える条件は、2勝かつ2敗の場合のみとなっている。これを2勝の場合とすることで変更要求を満たすことができるため
	<ボス切り替え条件>	
□□□	NEK01-1	状態 "BATTLE" 時、どのキャラクター表示をするか管理している変数 char_num へボス(3)を設定している個所の条件分岐からユーザの敗北数 (gT_defeat) の条件を削除する
□□□	NEK01-2	状態 "EVEN" 時、どのキャラクター表示をするか管理している変数 char_num へボス(3)を設定している個所の条件分岐からユーザの敗北数 (gT_defeat) の条件を削除する