６．VC++ コーディング規約

1. コメントは日本語または英語のみとする。
2. 定数、マクロ、関数、変数宣言は概要が分かるようにコメントを記載すること。
3. デバッグ文はコンパイルオプションでON/OFF可能なこと。
4. 不要となったコードは削除すること。コメントアウトで残さないこと。
5. ネスト（段付け）はタブとすること。スペースは使用しないこと。
6. デバッグ文を除き、アプリケーション画面に表示する文字列は全てリソースファイルに記載すること。コード中にオンコーディングしないこと。
7. 定数（ファイルパス、ファイル名、レジストリキーなど）はプログラムに直接記載せず、プログラム冒頭で定数宣言または外部ファイルから読み込むこと。
8. 変数名は意味の分かる文字列とすること。単にstr1、str2などと記載しないこと。
9. VisualStudioで生成するコントロールの名前はデフォルトの連番の名称は使用しないこと。意味のあるコントロール名とすること。（例：Button1はPlayButtonなどとすること）

＜プログラム例＞

#define HC\_MAX\_PATH 128 // フルパス 最大文字列数

#define NUMBER\_OF(arg) \

( sizeof(arg) / sizeof(arg[0]) ) // 配列の要素数を得る

typedef struct Memory { // メモリ上のバッファ

char \*str; // 文字列

struct Memory \*next; // リスト構造ポインタ

} MEMORY;

//

// 機能 : 文字位置の取得

// 引数 :

const char \*tarstr 検索される対象の文字列

const char \*srcstr 検索する指定部分文字列

// 返り値 : 文字位置[byte](検索されなかった場合は 0を返す)

// 機能説明 : 文字列の中から，指定文字列を検索し位置を返す。

// 備考 : pStrInStr( "a#c#d#eeee", "#" ) -> 2

//　　　　　pStrInStr( "a#c#d##eeee", "##" ) -> 6

//　　　　　pStrInStr( "abcdeabcde", "AA" ) -> 0

//

int pStrInStr(const char \*tarstr, const char \*srcstr)

{

// ブロック単位で処理のコメントを書く

}

７．次期MCG開発時の留意事項

現行MCGの過去バグより以下に留意して開発をお願いします。

1. 現行MCGはUACをオフにし管理者権限でチェックインしないと正しくインストールできず、チェックイン時と別ユーザでログインするとMCGが体験版となります。特別な操作をせずにMCGをチェックインした時に別ユーザでログインしても製品版となっていることを確認してください。
2. 授業のライブ収録は最低90分以上の録画を行います。これ以上の時間動作してもメモリリークや強制終了とならないことを確認してください。
3. 録画した映像と音声は少しでもずれるとクレームとなります。ズレの無いように確認をお願いします。
4. 録画した映像と音声が最初はあっているが、段々ずれて後半になるほどズレが大きくなる場合があります。確認の際は動画の序盤、中盤、後半で音声と映像の確認をお願いします。
5. 動画挿入、カット時に映像と音声のズレが無いように確認をお願いします。
6. 高解像度のモニタを使用時に現行MCGの動作が遅くなる現象がまれに発生します。１９２０×１０８０以上の解像度でもテストをお願いします。
7. 操作時にプログラムが「応答なし」の状態にならないようにしてください。ロード中や処理中はマルチスレッド化を行い、ダイアログでプログレスバーを動かすなど止まっていないように見せてください。