

נספח א' – קוד מבני

מטרת הכללים הבאים ליישם הלוקה לוגית וסגנון אחיד בתוך קבצים כדי להקל על ההתמצאות בקבצים של אנשים אחרים:

- יש לחלק את הקבצים (מימוש ו-Headers) לחלקים לוגיים שונים בתוך הקובץ ולתעד בהתאם (Definitions & Macros, User Defined Types, Prototypes, Globals).
- יש להפריד בין ביטויים ואופרטורים ע"י רווח: $iVal = 2 * (3 + Calc())$.
- כל פתיחה וסגירה של בלוך ('{' ו-'}'), תעשה ברמת הטבולציה הנוכחית. לדוגמא:

```
if (NULL == pBoard)
{
    for (i = 1; i < MOVES; i++)
    {
        ...
    }
    ...
}
```

- אין שימוש בהערות בוסה C (/* .. */) למעט במקרים שדרושה הערה באמצע פקודה.
- לפני כל פונקציה בקובץ המימוש יש לצרף תיעוד הכול: שם הפונקציה, מטרתה, הפרמטרים ומשמעותם, ערך החזרה.
- לפני קבוצת פקודות הממשות פונקציונליות אחת, תבוא הערה המסבירה את משמעות הפעולה. ההערה תיכתב באותה רמת טבולציה, וכן תתחיל ותסתיים בשורת הערה ריקה. לדוגמא:

```
//
// Choose the best move among the possible moves
//
for (pBoard = pSuccessors->GetFirst();
    NULL != pBoard; pBoard = pSuccessors->GetNext())
{
    //
    // Get the move rank using the MiniMax algorithm
    //
    iRank = MiniMax(pBoard, MIN, 0);
    ...
}
```

- יש לתת שמות משתנים לפי קונבנציית Hungarian Notation המפורטת בנספח ב'.

נספח ב' – קונבנציות Hungarian Notation

קונבנציה זו נועדה לאפשר למתכנת לדעת כמעט הכל על משתנים בתוכנית רק מהתבוננות בהם. כלליה הם:

1. שמות המשתנים מחולקים לשדות הבאים: <Name><Type><Pointer level><Array><Scope> כאשר-
 - <Scope> מציין את הסביבה בה המשתנה "חי" לפי הטבלה הבאה:

שדה ריק	משתנה לוקלי
g_	משתנה גלובלי
m_	Member של מחלקה
c_	קבוע

- נוסף תחילית 'a' כאשר המשתנה הוא מערך
- נוסף תחילית 'p' לכל רמת הצבעה
- נאפיין את טיפוס המשתנה לפי הטבלה הבאה:

Type	Prefix
char	c
int	i
long	l
void	v
“String terminated by Zero” (char array)	sz
unsigned int / short / long /char	ui / us / ul / uc
float	f
double	d

- נוכל להמציא תחיליות לטיפוסים ומבנים שלנו כל עוד הם לא מתנגשים עם ההגדרות לעיל.
- שם המשתנה יתחיל תמיד באות גדולה ושאר האותיות קטנות. אם השם מורכב ממספר מילים, הן יופיעו צמודות כאשר כל מילה מתחילה באות גדולה.
דוגמאות:

g_iReferenceCount – a global integer
pdValue – a pointer to a double
aiTable – an array of integers
m_szAddress – a member string
ppvTable – a pointer to a pointer to void
ulTimeOfDay – an unsigned long
c_lMagic – a constant long

נשים לב שניתן לדעת מה טיפוס המשתנה בביטויי הצבעה מרוכבים ע"י צמצום זוגות של 'p' ו-'*' :
אם בקטע תוכנית יש ביטוי *ppiID, ע"י "צמצום" נקבל 'piID' ולכן נדע שהביטוי מייצג מצביע ל-int.

2. שם של מחלקה יתחיל תמיד בתחילית 'C': CBoard
3. הגדרות מבנים, קבועים או מאקרואים (#define...), וכן טיפוסים חדשים יופיעו תמיד באותיות גדולות. אם ההגדרה מכילה מספר מילים הן יופרדו ע"י '_' :

```
#define MAGIC_NUMBER      0xFFFF7
#define MIN(x, y)         ((x) < (y)) ? (x) : (y)
typedef unsigned long     DWORD;
typedef struct _POINT {
    ...
} POINT, *PPOINT;
```