

VBAによる動画についての一考察

A Study of Animation Program using VBA

高林 茂樹

TAKABAYASHI Shigeki

In the animation programs using VBA that operates normally in Excel2003, there is something that doesn't operate normally in Excel2007 and Excel2010. Using the animation programs that I made of VBA, I compared the difference in operation by various versions of CPU, OS and Excel in the personal computers. As a result, it was possible to improve animation programs of VBA almost normally in operation by all present versions.

1. はじめに

Excelに含まれているVBA (Visual Basic for Applications) で作成した動画では、Excel2003で正常に動作していたプログラムが、Excel2007やExcel2010では、正常に動作しないものがある。VBAを使用して図形を動かす処理で、パソコンのCPU、OS、Excelのバージョンによってどのような違いがあるか比較した。VBAにあるDoEventsという制御をOSに移す命令の位置や図形の選択の有無などによる比較も行った。本論文では、これらの比較・検討により現在のバージョンで最適なVBAの動画作成プログラムの開発方法について考察したものである。

2. VBAの環境変化

VBAを使用して作成したプログラムに与える環境の変化について、関連のあるCPU、OS、プログラミング言語がどのように進化してきたか見ることにする。

2.1 CPU の変化

CPU は、コンピュータで演算と制御をする装置であるが、初期は、多くのトランジスタや集積回路 (IC) を組み合わせたものであった。1970年代に入ると、CPU の機能を1つの IC で実現したマイクロプロセッサ (MPU) が登場した。1971年に世界初の商用マイクロプロセッサである4ビット MPU が発表された。1974年に8ビット CPU インテル 8080、1978年には16ビット CPU インテル 8086発表され、これが x86系という言葉の元となった。1985年に32ビット CPU インテル 80386、1991年には64ビット RISC CPU MIPS R4000が発表された。1990年代後半になると、Windows95などの発売で、高性能の CPU が求められるようになり、高クロック周波数のインテル486、Pentium、MMX Pentium、Pentium II、Pentium IIIなどが登場した。2000年春には CPU のクロック周波数は1GHz 台になった。2004年にサン・マイクロシステムズ UltraSPARC IV では、同時マルチスレッディング (SMT) 機能を搭載し、ワンチップで2つのスレッドが実行可能になり、IBM POWER5ではワンチップで4つのスレッドが実行可能となった。

2000年代後半になると、クロック周波数の急激な増大に伴い発熱と消費電力が増大し、さらに半導体の微細化につれてリーク電流が加速度的に増大し、半導体回路を単純に微細化しても高速化につながりにくくなった。そして消費電力あたりの性能が重要視されるようになり、マルチコア CPU が普及し始めた。パソコン向けでは2コアが主流となっているが、サーバ向け CPU では UltraSPARC T1のようにマルチコアとハードウェアマルチスレッディングによりワンチップで数十のスレッドを実行する CPU が現れた。単一スレッドの実行速度は停滞気味となり、ハードウェアによる仮想化機能の搭載や、相対的に低いクロックでも高い性能を引き出しやすい SIMD の性能向上に力点が置かれるようになった。2006年にインテルの Intel Core 2ではパイプライン数を増やし、さらに128ビット処理が可能な広バンド幅の演算装置 ALU を搭載した。2008年にインテルは低消費電力プロセッサ Intel Atom を発表した。そしてインテルは2011年にさらに Core シリーズを発表し、Intel Core i7、Intel Core i5、Intel Core i3、Intel Pentium、Celeron および Xeon が発売された。なお、VBA でのマルチスレッド処理は、Access のクエリ以外では、現在行われていない。⁽¹⁾⁽²⁾

2.2 OS の変化

OS (オペレーティングシステム) という概念は1950年代に登場し、最初は、システム管理用ソフトウェアツールやハードウェアの使用を簡素化するツールであった。1960年代前半には、OS

機能の増強が進められ、スプール、ジョブ管理、記憶保護、マルチプログラミング、タイムシェアリングシステム、そして、仮想記憶の概念が登場した。1964年の IBM System/360シリーズに搭載された OS/360は世界初の商用 OS とされ、「オペレーティングシステム」という用語が一般化した。1970年代は、分散システムが普及し、ミニコンピュータ用 OS である VMS、オープンシステム UNIX は、ミニコンピュータからメインフレームまで広く使われた。1970年代には低価格なマイクロプロセッサが登場し、そのための OS として CP/M が8ビットのマイクロコンピュータでよく使われた。1980年代には16ビットの IBM PC 用に PC DOS が生まれ、これとほぼ同等のマイクロソフトの MS-DOS も普及した。そして GUI を標準装備したアップルコンピュータの Macintosh の OS が現れ、その後マイクロソフトに最初の Windows1.0 が現れた。1990年代にはオープンソースの UNIX 互換 OS が開発されはじめ、1991年に、Linux がフリーソフトウェアとして公開された。

1994年に、マルチタスク機能を備えた Windows NT が登場し、その後は WindowsNT の機能を限定的に取り入れた Windows 95、Windows 98との共存が続き、2001年には Windows は Windows NT に一本化された。2000年代後半からの Windows Vista、Windows 7や Mac OS X などの OS において、高機能のマルチコア CPU やビデオチップへの対応がされた。2010年代にはいると、狭い画面に最適化されたタッチパネルなどの組み込み、携帯機器の低消費電力の要求に応えた iOS、Android などのモバイルプラットフォームが使われるようになった。⁽²⁾

2.3 プログラミング言語の変化

プログラミング言語は、機械に動作を指示する符号であると考え、1801年にジャカード織機に指示を与えるパンチカードに遡ることができる。1890年には、国勢調査の情報を記録するため、パンチカードが利用された。1950年代にコンピュータの利用が本格的に始まり、多くのソフトウェアの開発が必要になった。それまでプログラムの作成はコンピュータに理解できる機械語と呼ばれる言語で行っていたが、機械語は人間には理解しにくい、人間にも理解しやすいコンピュータの言語の開発が始まり、1950年代後半から FORTRAN や COBOL などのプログラミング言語が登場した。1960年に MIT のジョン・マッカーシーが関数型言語である LISP を発表し、1964年には米ダートマス大学が BASIC (Beginner's All purpose Symbolic Instruction Code) を開発した。1970年代にマイクロソフトが MS BASIC を発表し、8ビットパソコンに ROM BASIC として搭載され広まった。1966年には IBM がシステム/360用に開発した PL/I を発表した。当時、多くの場合、科学技術計算には FORTRAN、事務処理には COBOL

が使用されていたが、PL/Iは双方の用途を1つの言語でカバーすることを目的とした。1969年スイス、チューリッヒ大学のニクラウス・ヴィルトが代表的な構造化言語の1つである Pascal を設計した。1970年頃 ベル研究所のケン・トンプソンらが B 言語を開発し、1972年にデニス・リッチーらが C 言語を開発した。1980年代にはオブジェクト指向の考え方を取り入れた C++へと発展した。

1991年にマイクロソフトが Microsoft Visual Basic を発表した。BASIC の名前がついているが、1980年代までの BASIC とはかなり異なるものである。また、1993年に Excel、Word など Microsoft Office に Visual Basic for Applications (VBA) が付いた。1995年にサン・マイクロシステムズが本格的なオブジェクト指向言語の1つ Java を発表した。2002年にマイクロソフトが C++と Java の中間的な特徴をもつ C#を発表した。2007年 C 言語の置き換えを目指して開発された D 言語が発表された。Excel 2010では、Microsoft Visual Basic for Applications 7.0、Excel2007と Excel2003では、Microsoft Visual Basic for Applications 6.5を使用している。⁽²⁾

3. VBA による動画の動作比較

3.1 動作比較のための VBA プログラム

(1)画像が1個の場合

動かす画像としては、図-1にあるように図形の四角と丸を組み合わせで作成した車を使用した。



図-1 動作比較の画面

準備ボタンは、車をスタートラインに合わせるためのもので最初の Sub プログラムに対応している。スタートボタンの0~5はケース0~5の動作を開始するためのもので2番目から6番目の Sub プログラムに対応している。ケース0は、DoEvents がない場合、ケース1は、DoEvents は1つで車の選択枠を消去するためにセルの選択を入れたもの、ケース2は車を選択する前後2か所

に DoEvents を入れたもの、ケース3は、DoEvents は1つで車を選択しないもの、ケース4は、DoEvents は1つで、車を選択した後のセルの選択がないもの、ケース5は、車を選択する前後2か所に DoEvents を入れ、車を選択した後のセルの選択がないものである。DoEvents はプログラムで占有していた制御を OS に一時的に渡すための関数である。⁽³⁾⁽⁴⁾ 以下は、画像が1個の場合のプログラムである。最初のプログラムは準備、その次からプログラムは、ケース0からケース5に対応する。

<pre> Sub kurumajunbi() Sheets("sheet1").Select i = 50 ActiveSheet.DrawingObjects("kuruma1").Select Selection.Left = i Selection.Top = 55 Range("D16").Select End Sub Sub kuruma() Sheets("sheet1").Select For i = 50 To 400 ActiveSheet.DrawingObjects("kuruma1").Select Selection.Left = i Selection.Top = 55 Next i Range("D16").Select End Sub Sub kuruma_de1() Sheets("sheet1").Select For i = 50 To 400 ActiveSheet.DrawingObjects("kuruma1").Select Selection.Left = i Selection.Top = 55 DoEvents Range("D16").Select Next i Range("D16").Select End Sub Sub kuruma_de2() Sheets("sheet1").Select For i = 50 To 400 DoEvents ActiveSheet.DrawingObjects("kuruma1").Select Selection.Left = i Selection.Top = 55 DoEvents </pre>	<pre> Range("D16").Select Next i Range("D16").Select End Sub Sub kuruma_de1s0() Sheets("sheet1").Select For i = 50 To 400 DoEvents With ActiveSheet.DrawingObjects("kuruma1") .Left = i .Top = 55 End With Next i Range("D16").Select End Sub Sub kuruma_de1r1() Sheets("sheet1").Select For i = 50 To 400 ActiveSheet.DrawingObjects("kuruma1").Select Selection.Left = i Selection.Top = 55 DoEvents Next i Range("D16").Select End Sub Sub kuruma_de2r1() Sheets("sheet1").Select For i = 50 To 400 DoEvents ActiveSheet.DrawingObjects("kuruma1").Select Selection.Left = i Selection.Top = 55 DoEvents Next i Range("D16").Select End Sub </pre>
--	--

(2)画像が2個および3個の場合

ケース0は、DoEvents がない場合、ケース1は、DoEvents は1つで車の選択枠を消去するためにセルの選択を入れたもの、ケース2はそれぞれの車を選択する前後2か所に DoEvents を入れたもの、ケース3は、DoEvents は1つで車を選択しないものである。以下は、画像が3個の場合

合のプログラムである。最初のプログラムは車をスタートラインにそろえる準備、その次からプログラムは、ケース0からケース3に対応する。

```

Sub kurumajunbi()
  Sheets("sheet3").Select
  i = 50
  ActiveSheet.DrawingObjects("kuruma1").Select
  Selection.Left = i
  Selection.Top = 55
  ActiveSheet.DrawingObjects("kuruma2").Select
  Selection.Left = i
  Selection.Top = 133
  ActiveSheet.DrawingObjects("kuruma3").Select
  Selection.Left = i
  Selection.Top = 211
  Range("D23").Select
End Sub

Sub kuruma()
  Sheets("sheet3").Select
  For i = 50 To 400
    ActiveSheet.DrawingObjects("kuruma1").Select
    Selection.Left = i
    Selection.Top = 55
    ActiveSheet.DrawingObjects("kuruma2").Select
    Selection.Left = i + i / 10
    Selection.Top = 133
    ActiveSheet.DrawingObjects("kuruma3").Select
    Selection.Left = i + i / 5
    Selection.Top = 211
  Next i
  Range("D23").Select
End Sub

Sub kuruma_de1()
  Sheets("sheet3").Select
  For i = 50 To 400
    DoEvents
    ActiveSheet.DrawingObjects("kuruma1").Select
    Selection.Left = i
    Selection.Top = 55
    ActiveSheet.DrawingObjects("kuruma2").Select
    Selection.Left = i + i / 10
    Selection.Top = 133
    ActiveSheet.DrawingObjects("kuruma3").Select
    Selection.Left = i + i / 5
    Selection.Top = 211
    Range("D23").Select
  Next i
  Range("D23").Select
End Sub

Sub kuruma_de2()
  Sheets("sheet3").Select
  For i = 50 To 400
    DoEvents
    ActiveSheet.DrawingObjects("kuruma1").Select
    DoEvents
    Selection.Left = i
    Selection.Top = 55
    Range("D23").Select
    DoEvents
    ActiveSheet.DrawingObjects("kuruma2").Select
    DoEvents
    Selection.Left = i + i / 10
    Selection.Top = 133
    Range("D23").Select
    DoEvents
    ActiveSheet.DrawingObjects("kuruma3").Select
    DoEvents
    Selection.Left = i + i / 5
    Selection.Top = 211
    Range("D23").Select
  Next i
  Range("D23").Select
End Sub

Sub kuruma_de1s0()
  Sheets("sheet3").Select
  For i = 50 To 400
    DoEvents
    With ActiveSheet.DrawingObjects("kuruma1")
      .Left = i
      .Top = 55
    End With
    With ActiveSheet.DrawingObjects("kuruma2")
      .Left = i + i / 10
      .Top = 133
    End With
    With ActiveSheet.DrawingObjects("kuruma3")
      .Left = i + i / 5
      .Top = 211
    End With
  Next i
  Range("D23").Select
End Sub

```

(3)画像が1個で割込み処理のある場合

ケース0は、DoEventsがない場合、ケース1は、DoEventsは1つで車の選択枠を消去するためにセルの選択を入れたもの、ケース2は車を選択する前後2か所にDoEventsを入れたもの、ケース3は、DoEventsは1つで車を選択しないものである。最初のプログラムは車をスタートラ

インにそろえる準備、2つ目が、割込みで車を停止させるもの、その次からプログラムは、ケース0からケース3に対応する。

```

Dim kurumaleft, kurumastop1

Sub kurumajunbi()
    Sheets("sheet1").Select
    i = 50
    kurumaleft = i
    kurumastop1 = 0
    ActiveSheet.DrawingObjects("kuruma1").Select
    Selection.Left = i
    Selection.Top = 55
    Range("D16").Select
End Sub

Sub kurumastop()
    kurumastop1 = 1
End Sub

Sub kuruma()
    Sheets("sheet1").Select
    Do Until kurumaleft >= 800
        kurumaleft = kurumaleft + 1
    ActiveSheet.DrawingObjects("kuruma1").Select
    Selection.Left = kurumaleft
    Selection.Top = 55
    If kurumastop1 = 1 Then
        kurumastop1 = 0
        Range("D16").Select
    End Sub
    End If
    Loop
    Range("D16").Select
End Sub

Sub kuruma_de1()
    Sheets("sheet1").Select
    Do Until kurumaleft >= 800
        kurumaleft = kurumaleft + 1
    DoEvents
    ActiveSheet.DrawingObjects("kuruma1").Select
    Selection.Left = kurumaleft
    Selection.Top = 55
    Range("D16").Select
    If kurumastop1 = 1 Then
        kurumastop1 = 0
        Range("D16").Select
    End Sub
    End If
    Loop
    Range("D16").Select
End Sub

Sub kuruma_de2()
    Sheets("sheet1").Select
    Do Until kurumaleft >= 800
        kurumaleft = kurumaleft + 1
    DoEvents
    ActiveSheet.DrawingObjects("kuruma1").Select
    DoEvents
    Selection.Left = kurumaleft
    Selection.Top = 55
    Range("D16").Select
    If kurumastop1 = 1 Then
        kurumastop1 = 0
        Range("D16").Select
    End Sub
    End If
    Loop
    Range("D16").Select
End Sub

Sub kuruma_de1s0()
    Sheets("sheet1").Select
    Do Until kurumaleft >= 800
        kurumaleft = kurumaleft + 1
    DoEvents
    With ActiveSheet.DrawingObjects("kuruma1")
        .Left = kurumaleft
        .Top = 55
    End With
    If kurumastop1 = 1 Then
        kurumastop1 = 0
    End Sub
    End If
    Loop
    Range("D16").Select
End Sub

```

(4)画像が2個で相互に割込み処理がある場合

動かす図形として、星型の図形とロケット型の図形を使用して、星型の図形をロケット型の図形でねらう処理を想定した。ケース1は、DoEventsは1つで星とロケットの図形の選択枠を消去するためにセルの選択を入れたもの、ケース2は星とロケットの図形を選択する前後2か所にDoEventsを入れたもの、ケース3は、DoEventsは1つで星とロケットの図形を選択しないものである。最初のプログラムは星を動かしてをロケットスタート位置にセットする準備、2つ目は、

終了のプログラム、3つ目は星が移動するプログラム4つ目は、割込みでロケットを発射するプログラムでロケットが移動中は星も割込み処理を行う。

```

Dim hosileft, rockettop, hosistop, rocketstop

Sub rocketjunbi()
    Sheets("sheet3").Select
    With ActiveSheet.DrawingObjects("rocket")
        .Left = 350
        .Top = 400
    End With
    hosileft = 50
    rockettop = 500
    hosistop = 0
    rocketstop = 0
    Call hosi
End Sub

Sub owari()
    hosileft = 1000
End
End Sub

Sub hosi()
    Sheets("sheet3").Select
    Do Until hosileft >= 900
        hosileft = hosileft + 1
        If hosileft > 799 Then hosileft = 50
        DoEvents
        With ActiveSheet.DrawingObjects("star")
            .Left = hosileft
            .Top = 10
        End With
    End With
    If hosistop = 1 Then
        hosistop = 0
        Call rocket
    End If
    Loop
End Sub

Sub rocket()
    Sheets("sheet3").Select
    Do Until rockettop <= 3
        rockettop = rockettop - 5
        DoEvents
        With ActiveSheet.DrawingObjects("rocket")
            .Left = 350
            .Top = rockettop
        End With
        hosistop = 1
        Call hosi
    Loop
    hosistop = 0
    rockettop = 400
    DoEvents
    With ActiveSheet.DrawingObjects("rocket")
        .Left = 350
        .Top = rockettop
    End With
End Sub

```

3.2 VBA プログラムごとの動作比較

(1)動作比較結果

Excel2003でCPUの処理速度が遅い場合では、DoEvents無しでも図形はなめらかに移動した。図形をSelectで選択した場合は、Excel2007およびExcel2010では、選択枠が表示された状態になる。図形とセルを交互に選択することで図形の選択枠の表示を見えないようにすることができるが、ちらつきが起こる。これは、車1台から車3台のケース3および割込みのケース3、相互割込みのケース3のように、図形をSelectで選択しないことでほぼ正常になる。

次の表-1は、パソコン1から10について、ケースごとに「○：正常の場合」、「×：図形が始点から終点まで飛んでしまう場合」、「□：選択枠が表示された状態になる場合」、「*：図形のちらつきが目立つ場合」、「□*：選択枠がちらつき場合」をまとめたものである。

表-1 動作比較

No.	1	2	3	4	5	6	7	8	9	10
PC機種	D750/A	HPL170	Lenovo PC	Aspire 1410	4520s	dc5800	I1520	I1520	DHP	AX/650LSBB
CPU	650	E6550	235e	U2300	P4600	E7300	T7100	T7100	Pentium 4	M processor
OS	Windows 7	Windows7	Windows7	Windows7	Windows7	WindowsVista	WindowsVista	WindowsVista	WindowsXp	WindowsXp
Office	Excel2010	Excel2010	Excel2010	Excel2007	Excel2007	Excel2007	Excel2010	Excel2007	Excel2007	Excel2003
車1台Case0	x	x	x	x	x	x	x	x	x	○
車1台Case1	○	○	□*	□*	○	○	○	○	□*	○
車1台Case2	○	○	□*	□*	○	○	○	○	□*	○
車1台Case3	○	○	○	○	○	○	○	○	○	○
車1台Case4	x	x	□	□	x	x	x	x	□	○
車1台Case5	□	□	□	□	□	□	□	□	□	○
車2台Case0	x	x	x	x	x	x	x	x	x	○
車2台Case1	○	○	○	○	○	○	○	○	○	○
車2台Case2	○	○	□*	□*	○	○	○	○	□*	○
車2台Case3	○	○	○	○	○	○	○	○	○	○
車3台Case0	x	x	x	x	x	x	x	x	x	○
車3台Case1	○	○	○	○	○	○	○	○	○	○
車3台Case2	○	○	□*	□*	○	○	○	○	□*	○
車3台Case3	○	○	○	○	○	○	○	○	○	○
割込みCase0	x	x	x	x	x	x	x	x	x	x
割込みCase1	○	○	□*	□*	○	○	○	○	□*	○
割込みCase2	○	○	□*	□*	○	○	○	○	□*	*
割込みCase3	○	○	○	○	○	○	○	○	○	○
相互割込みCase1	*	*	*	*	*	*	*	*	*	○
相互割込みCase2	*	*	□*	□*	*	*	*	*	□*	*
相互割込みCase3	○	○	○	○	○	○	○	○	○	○

(2)使用したパソコンの CPU および OS

比較のために使用したパソコンは9台である。No.7と No.8は同じパソコンであるが、Excel2007と Excel2010がインストールされているため分けて記述した。

次の表-2が使用したパソコンの CPU および OS である。

表-2 使用したパソコンの CPU および OS

No.	CPU および OS 等
1	Fujitsu C750/A Intel® Core™ i5 CPU 650 @3.20GHz Windows 7 Professional (SP1) 32bit OS
2	HP HPL170 Intel® Core™2 Duo CPU E6550 @2.33GHz Windows 7 Professional (SP1) 32bit OS
3	Lenovo Windows 7 PC AMD Athlon™ II X2 235e Processor 2.70GHz Windows7 Home Premium 64bit OS
4	Acer Aspire 1410 Genuine Intel™ CPU U2300 @1.20GHz Windows7 Home Premium 64bit OS
5	HP Probook 4520S Intel® CPU P4600 @2.00GHz Windows 7 Professional 32Bit OS
6	HP dc5800 Intel® Core™ 2 Duo CPU E7300 @2.66GHz Windows Vista™ Enterprise(SP1) 32bit OS
7	Dell Inspiron Intel® Core™ 2 Duo CPU T7100 @1.80GHz Windows Vista™ Home Premium
8	Dell Inspiron Intel® Core™ 2 Duo CPU T7100 @1.80GHz Windows Vista™ Home Premium
9	Dell DHP Intel™ Pentium® 4 CPU @2.60GHz Windows Xp Professional (SP3)
10	Toshiba AX/659LSBB Intel® Celeron® M processor 1.40GHz Windows Xp Home Edition(SP3)

4. おわりに

VBA で図形を動かす処理でも Excel のバージョンの違いで表示や動きにかなりの違いがあることがわかった。図形を Select で選択しないことでちらつきが減少し、ほぼ正常になることもわかった。VBA に限らず、バージョンアップには、よい面がもちろん多くあるが、そのたびにプログラムやシステムが動作しなくなったり、動作が不安定になったりすることがある。CPU や OS、Excel 等のソフトウェアの複雑化に伴って、以前の動いていたプログラムがバージョンアップで正常に動作しなくなった場合でも、以前の環境に戻って、正常に処理ができるとよい。一部にはこのような機能のあるものもあるが、より簡単に可能になることが望まれる。

参考文献

- (1) Intel 「製品情報」 <http://www.intel.co.jp/2011>
- (2) Microsoft 「製品一覧」 <http://www.microsoft.com/ja/jp/default.aspx2011>
- (3) E-Trainer.jp : 中村俊 「Excel VBA 辞典」 秀和システム 2010
- (4) 近田伸矢 「Excel VBA アクションゲーム作成入門」 インプレスジャパン 2011