

The universal DOS program environment applied for complex 3D computer visualization problems

A.F. Korniyushkin¹ and S. K. Sekatskii^{1,2}

¹Institute of Spectroscopy Russian Academy of Sciences, 142190 Troitsk, Moscow region, Russia.
E-mail: korn@isan.troitsk.ru (www.isan.troitsk.ru/dls/korn)

²Institut de Physique de la Matière Condensée Université de Lausanne, BSP, CH1015 Lausanne-Dorigny, Switzerland.
E-mail: sergey.sekatskii@ipmc.unil.ch

Abstract

We give a description of the simple DOS program environment (program DOS4GW written using the WATCOM C/C++ IDE language) which possess all the WINDOWS possibilities. Its realization is demonstrated for the designing of the complex many-object 3D engine.

1. INTRODUCTION

There are a number of objective drawbacks and difficulties, which one encounters when trying to write a program using the WINDOWS program environment. Among others, this is a non – transparency of the computer code which is difficult to use and to be fully familiar with. Moreover, there are a lot of problems where all the possibilities of WINDOWS are completely superfluous and for the practical programming it remains nothing else than to rely on the uninspiring “void main ()”.

It seems that it is not possible to design a more general approach to the problem. But is it really the case?

2. THE UNIVERSAL DOS ENVIRONMENT FOR THE RAPID PROGRAMMING (PROJECT “MICROWINDOWS”)

It is easy to note that literally all the interactive problems can be represented as follows:

```
int key;
while(key!='q') // main circle.
{
1). Description of the variables
(using the keys, mouse and so on)
2). Calculations
3). Videocontrol system
if(kbhit()) key=getch(); // exit program
}
```

Now let us propose that the main control is performed using the mouse and do the following. We select a few of the upper strings in the screen (so called «flashing keys» area) organized as it is shown in Fig. 1a and modify the program as follows:

```
int key;
unsigned char iInOut;

while(key!='q') // main circle.
{
1).
if(iInOut)
{
#include «book1.h»
#include «book2.h»
}
2). Calculations
3). Videocontrol system
(original programs to draw and eliminate lines, to project an
image onto the screen, etc.)

if(kbhit()) key=getch(); // exit program
}
```

Here iInOut defines the position of the mouse (is it in the «flashing keys» area or not) while «book1» and «book2» define two similarly constructed objects named «books».

These books are composed by the pages where the the functions and global variables to be controlled or changed are kept.

When the corresponding page is activated the program continue to work but with the diminished in accordance to this page part of the screen.

Thus we have a possibility to change any variable or apply any function while remaining fully inside our program. But this is exactly all what we need!

Adding to this scheme «drag and drop» Z-buffer, which enables to look at the drawing in full details, MATLAB headers, or music block or whatever you want (these are simply additional pages with the necessary comments), we obtain the powerful and universal system for such users (obviously, for by far the largest part of them) who have serious 3D computer imaging problems to be solved but do not know (and do not wish to know!) what is OWL-library and «invalidate rectangle».

The program is realized as a DOS4GW program for two resolutions (640x480 and 1024x768).

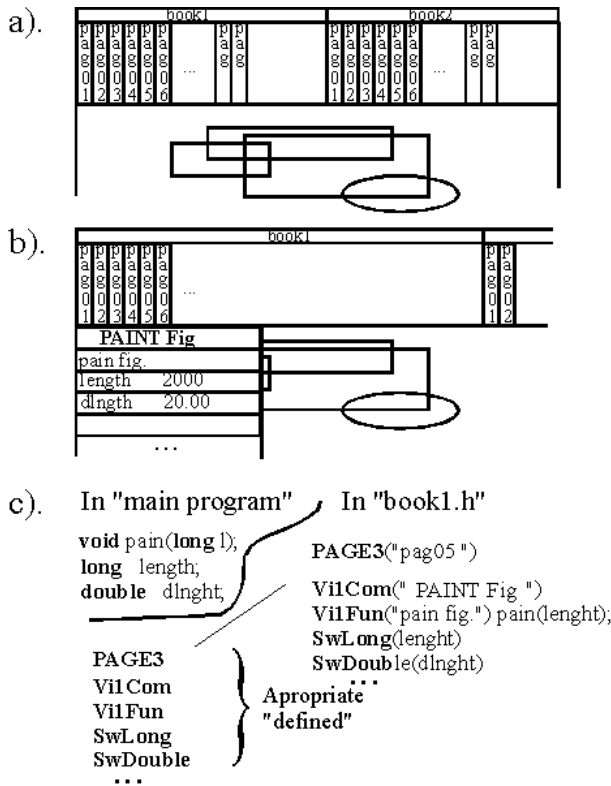


Figure 1:

- a) The book for closing
- b) The book for opening
- c) Scheme of the book

2. CONCLUSION

Both very simple and very complex programs can be written using the program environment described.

The complexity (Winsight using 300 variables!) was exactly what was needed to write the test program "Martian soccer", where vehicles are playing with the ball in a very complex landscape and trying to score in into the basketball ring.

Each vehicle is capable to look for the ball, to throw it into the ring, to avoid obstacles and so on (see Fig. 2 and Fig. 3).



Figure 2:
"MICROWINDOWS" at work.

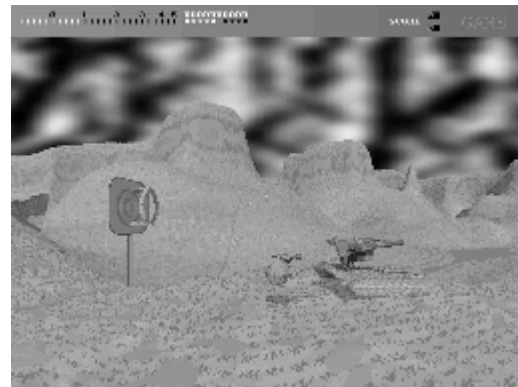


Figure 3:
"What an opportunity to score!".

3. REFERENCES

[1] A novel fast human vision adapted 3D visualization engine with an easy main memory access. A.F. Kornushkin and S. K. Sekatskii: GRAFICON 2000