

```

/* Part 3 of demonstrations of the use of GEM from 'C' */
/* By Steve Pedler for Page 6 Magazine, England */

/* The following lines should be added to the source code published in the
previous issues of Page 6 */

/* new global declarations */

WORD calc_num(), demo4();

/* new main() function - substitute this function for that
found in the previous version, so that the fourth demo can be called */
main()
{
    WORD c;

    init_gem();
    while( finished !=0 ){
        c=demo1();
        if( c==3 ) {
            finished=0; break;
        }
        c=demo2();
        if( c==3 ) {
            finished=0; break;
        }
        c=demo3();
        if( c==SPACE ) {
            finished=0; break;
        }
        c=demo4(); /* new lines */
        if( c==3 ) {
            finished=0; break;
        }
    }
    finish_gem();
}

/* ----- demonstration #4 - draw a series of globes ----- */
WORD demo4()
{
    WORD centre_x, centre_y, perm_radius, temp_radius, color,
        starta, stopa, clip[ 4 ], c;

    int max_color, max_rad, high_x, high_y;

    starta=0;
    stopa=3600;

    color=2; /* These 5 parameters may be changed for */
    max_color=7; /* different effects */
    max_rad=50;
    high_y=180;
    high_x=305;

    Setpalette( old_pal );
    v_clrwk( work_handle );
    vst_height( work_handle, 6, &dum, &dum, &dum, &dum );
    vst_color( work_handle, 1 );
    v_gtext( work_handle, 8, 198, "Demo 4 - Globes" );
    vsi_color( work_handle, color );

    clip[ 0 ]=1;
    clip[ 1 ]=1;
    clip[ 2 ]=318;
    clip[ 3 ]=190; /* set up a graphics clipping */
    vs_clip( work_handle, 1, clip ); /* rectangle */

    button=0;
    while( button==0 ) {
        perm_radius=calc_num( 10, max_rad ); /* calculate the */
        centre_x=calc_num( 12, high_x ); /* 3 parameters needed */
        centre_y=calc_num( 12, high_y ); /* for each globe */

        vsf_color( work_handle, 0 );
        v_ellipse( work_handle, centre_x, centre_y,
            perm_radius, perm_radius );

        vsi_color( work_handle, color );
        temp_radius=perm_radius-5;
        v_ellarc( work_handle, centre_x, centre_y,
            perm_radius, perm_radius, starta, stopa );

        while( temp_radius>0 ){
            v_ellarc( work_handle, centre_x, centre_y,
                perm_radius, temp_radius, starta, stopa );
            v_ellarc( work_handle, centre_x, centre_y,
                temp_radius, perm_radius, starta, stopa );
            temp_radius-=5; /* note: this is
'temp_radius=temp_radius-5' */
        }
        color++;
        if( color>max_color ) color=2;
        vq_mouse( work_handle, &button, &dum, &dum );
    }
    c=button;
    while( button !=0 ) {
        vq_mouse( work_handle, &button, &dum, &dum );
    }
    vs_clip( work_handle, 0, clip ); /* disable clipping rectangle */
    return (c);
}

/* calculate the centres and radius */
WORD calc_num( small, large )
int small, large;
{
    double val_max, val_temp, ret_num;
    int random;

    val_max=2147483647.0;
    /* ***** NOTE: if your compiler uses 16-bit ints rather than the 32
bits used by Lattice C, the 'magic number' val_max should be changed
to 32767.0 - this is very important! ***** */

    random=rand(); /* C library function for random numbers */
    val_temp=random/val_max;

    ret_num=( val_temp*large )+small;
    return (WORD)ret_num;
}

```