

```

/* ----- LISTING 1 -----*/

/* GEM demonstrations from the 'C' programming language */
/* Written using the Metacomco Lattice C compiler version 3.04 */
/* For Page 6 Magazine, England */

/* This listing details the lines to be added to the first version ( 0.2 )
of this program, published in issue 28 of Page 6 Magazine */

/* ----- new defines ----- */

#define CENTRED 1
#define SPACE 0x3920

/* ----- new declarations ----- */

WORD demo2(), demo3();

/* NOTE: if you are using Lattice C version 3.04, you should:
1) delete the definitions of the global VDI arrays contrl[],
intin[], intout[], ptsin[] and ptsout[]
2) add two definitions - WORD work_in[ 11 ], work_out[ 57 ];
3) replace all references to intin[] and intout[] in the function
init_gem() with work_in[] and work_out[] respectively */

/* ----- new main() function ----- */

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;
        }
    }
    finish_gem();
}

/* ----- new finish_gem() function ----- */

finish_gem()
{
    WORD reset;
    reset=0;
    v_clrwk( work_handle );
    Setpalette( old_pal );
    v_show_c( work_handle, reset);
    v_clsvwk( work_handle );
    appl_exit();
}

```

```

/* GEM demonstrations from C */
/* Current version: 0.2, 11/7/87 */

/* ----- LISTING 2 ----- */

/* This listing contains the two new demos to be added to version 0.2 of
the program once the changes contained in Listing 1 have been made. */

/* ----- demonstration #2 ----- */
/* create illusion of movement by drawing and redrawing circles */
WORD demo2()
{
    int i;

    WORD c, x_incr, y_incr, last_x_incr, last_y_incr, centre_x,
    centre_y, last_x, last_y, color, radius;

    color=1;
    new_pal[ 0 ]=0; /* black background */
    new_pal[ 15 ]=0x777;
    Setpalette( new_pal );

    v_clrwk( work_handle );

    vst_color( work_handle, 1 );
    v_gText( work_handle, 8, 196, "Demo 2 - Moving circles" );
    vsf_color( work_handle, color );
    vsf_interior( work_handle, 0 ); /* no fill */
    vsf_perimeter( work_handle, 1 ); /* visible border */

    x_incr=last_x_incr-7;
    y_incr=last_y_incr+5;
    radius=10;
    last_x=153;
    last_y=95;

    for( i=0, centre_x=160, centre_y=100; i<10; ++i, centre_x+=x_incr, cent
re_y+=y_incr ) {
        v_circle( work_handle, centre_x, centre_y, radius );
    }
}

```

continued overleaf

```

button=0;
while( button==0 ){
    if( centre_x<14 || centre_x>306 ) {
        x_incr*=-1;
        color+=1;
        if( color>15 ) color=1;
    }

    if( centre_y<14 || centre_y>179 ) {
        y_incr*=-1;
        color+=1;
        if( color>15 ) color=1;
    }

    vsf_color( work_handle, color );
    v_circle( work_handle, centre_x, centre_y, radius );
    centre_x+=x_incr;
    centre_y+=y_incr;

    if( last_x<14 || last_x>306 ) last_x_incr*=-1;
    if( last_y<14 || last_y>179 ) last_y_incr*=-1;
    last_x+=last_x_incr;
    last_y+=last_y_incr;

    vsf_color(work_handle, 0 ); /* background colour */
    v_circle( work_handle, last_x, last_y, radius );
    vq_mouse( work_handle, &button, &dum, &dum );
}
c=button;
while( button!=0 ){
    vq_mouse( work_handle, &button, &dum, &dum );
}
return( c );
}

/* ----- demonstration #3 ----- */
/* printing text on the screen */
WORD demo3()
{
    WORD c, pxyarray[ 4 ];

    char str[ 3 ];
    str[ 0 ]=0x0e; /* the two halves of the Atari logo */
    str[ 1 ]=0x0f;
    str[ 2 ]=0x00; /* null terminator */

    v_clrwk( work_handle );
    vst_color( work_handle, 1 );
    v_gtext( work_handle, 8, 185, "Demo 3 - Text output" );

    vst_height( work_handle, 50, &dum, &dum, &dum, &dum );
    vst_color( work_handle, 2 );
    v_gtext( work_handle, 25, 150, "ATARI ST" );
    v_gtext( work_handle, 170, 150, str );

    vst_effects( work_handle, 12 ); /* 12 = italic & underline */
    vst_height( work_handle, 8, &dum, &dum, &dum, &dum );
    v_gtext( work_handle, 25, 170, "Page 6 Magazine - England" );

    vst_color( work_handle, 3 );
    vst_effects( work_handle, 32 ); /* 32=shadowed text */
    vst_height( work_handle, 13, &dum, &dum, &dum, &dum );
    vst_alignment( work_handle, CENTRED, 0, &dum, &dum );
    v_justified( work_handle, 165, 25, "stretched text", 280, 1, 1 );
    vst_alignment( work_handle, 0, 0, &dum, &dum );

    vst_effects( work_handle, 0 ); /* 0=standard text */
    vst_color( work_handle, 4 );
    vst_height( work_handle, 4, &dum, &dum, &dum, &dum );
    v_gtext( work_handle, 50, 102, "very small text indeed!!" );

    vst_height( work_handle, 8, &dum, &dum, &dum, &dum );
    vst_color( work_handle, 5 );
    v_gtext( work_handle, 100, 120, "normal.." );
    vst_effects( work_handle, 2 ); /* 2=light text */
    v_gtext( work_handle, 25, 120, "Light.." );
    vst_effects( work_handle, 1 ); /* 1=bold text */
    v_gtext( work_handle, 185, 120, "and bold." );

    vst_rotation( work_handle, 2700 ); /* baseline rotation 2700 degree
es */
    vst_height( work_handle, 8, &dum, &dum, &dum, &dum );
    vst_color( work_handle, 6 );
    vst_effects( work_handle, 16 ); /* 16=outlined text */
    v_gtext( work_handle, 5, 5, "OUTLINED TEXT" );

    vsi_color( work_handle, 7 );
    vst_color( work_handle, 7 );
    vst_rotation( work_handle, 0 ); /* normal baseline */
    vst_effects( work_handle, 0 );
    vst_height( work_handle, 7, &dum, &dum, &dum, &dum );
    vsi_width( work_handle, 1 );
    pxyarray[ 0 ]=25;
    pxyarray[ 1 ]=40;
    pxyarray[ 2 ]=305;
    pxyarray[ 3 ]=80;
    v_rbox( work_handle, pxyarray );

    v_gtext( work_handle, 30, 55, "Press any key to continue..." );
    v_gtext( work_handle, 30, 75, "or space bar to exit program" );

    vst_color( work_handle, 1 ); /* these 3 lines reset values */
    new_pal[ 0 ]=0x777; /* needed by demo1() */
    new_pal[ 15 ]=0;

    return( c=evnt_keybd() );
}

```