Animated Cursor

```
/*
 * Animated Cursor
 *generic animated cursor support.
 */

// Assumes inclusion of <MacHeaders>

Boolean InitAnimatedCursors(short acurID, short interval);
void ReleaseAnimatedCursors(void);
void SpinCursor(void );

typedef struct
                /* The structure of an 'acur' resource */
{
      short numberOfFrames;                    /* number of cursors to animate */
      short whichFrame;                        /* current frame number */
      CursHandle frame[];                      /* Pointer to the first cursor */
} acur, *acurPtr, **acurHandle;

/* Redefine HiWord() as a macro to increase speed. */
#define HiWrd(aLong)     (((aLong) >> 16) & 0xFFFF)

/* Some module-local globals */
static short gTickInterval; /* number of ticks between a frame switch */
static long gLastTick;          /* tick count of last call to SpinCursor */
static acurHandle gFrameList;                  /* our cursor list */

/* Try to get the acur record and the cursor list for acurID, returning TRUE
   if everything goes as planned. */
Boolean
InitAnimatedCursors(short acurID, short interval)
{
      register short i=0;
      register short cursID;
      Boolean noErrFlag = FALSE;

      if((gFrameList = (acurHandle) GetResource('acur',acurID))) {
                /* got it! */
                noErrFlag = TRUE;
                while((i<(*gFrameList)->numberOfFrames) && noErrFlag) {
                /* The id of the cursor is stored in the high word of the frame handle */
                          cursID = (short) HiWrd((long) (*gFrameList)->frame[i]);
                          (*gFrameList)->frame[i] = GetCursor(cursID);
                          if((*gFrameList)->frame[i])
                                    i++;                       /* get the next one */
                          else
                                    noErrFlag=FALSE;     /* foo! we couldn't find the cursor
                                                         */
                }
      }
      if(noErrFlag) {
                /* We have the cursors, now initialize the other fields */
                gTickInterval = interval;
                gLastTick = TickCount();
```

```
            (*gFrameList)->whichFrame = 0;
    }
    return noErrFlag;
}

/* Free up the storage used by the current animated cursor and all
  of its frames */
void
ReleaseAnimatedCursors()
{
    short i;

    for(i=0;i<(*gFrameList)->numberOfFrames;i++)
            ReleaseResource((Handle) (*gFrameList)->frame[i]);
    ReleaseResource((Handle) gFrameList);
}

/*  Display the next frame in the sequence, if it's time. If not, do nothing.
    This code should be pretty tight, but it might be possible to bum a few
    instructions; ideally it should be as fast as possible. I doubt it will
    be necessary to hand-hack it though.

    I chose this implementation over using a VBL task for the following reason:
    the whole point of using an animated cursor is to let the user know that
    your application is chugging away doing something. With this technique if
    the applications shoots off to never-never land, the cursor will probably
    stop spinning. A VBL task, on the other hand, would probably just sit and
    merrily go on spinning away, the user none the wiser. */
void
SpinCursor()
{
    register long  newTick;

    newTick = TickCount();

    /* Is it time? */
    if(newTick < (gLastTick + gTickInterval))
            return;             /* nope */

    /* Grab the frame, increment (and reset, if necessary) the count, and
            display the new cursor */
    SetCursor(*((*gFrameList)->frame[(*gFrameList)->whichFrame++]));
    if((*gFrameList)->whichFrame == (*gFrameList)->numberOfFrames)
            (*gFrameList)->whichFrame = 0;
    gLastTick = newTick;
}

#define countingHand        1000                /* id of the 'acur' resource */
#define watchCursor         1002
#define beachBall           1003
#define movieStrip          1004

#define rate        12      /* Make the animation faster by
                            * decreasing the rate
                            * and slower by increasing the rate
                            */
```

```
main()
{
    if(InitAnimatedCursors(beachBall, rate)) {
            while (!Button()) {
                    SpinCursor();
            };
            ReleaseAnimatedCursors();
    }
}

/* Rez description file for 4 spinning cursor types*/

#include "Types.r"

resource 'CURS' (300, preload) {
    $"0000 0000 0000 0000 0DB0 124C 100A 0802"
    $"1802 2002 2002 2004 1004 0808 0408 0408",
    $"0000 0000 0000 0000 0DB0 1FFC 1FFE 0FFE"
    $"1FFE 3FFE 3FFE 3FFC 1FFC 0FF8 07F8 07F8",
    {9, 8}
};

resource 'CURS' (301, preload) {
    $"0000 1800 2400 2580 1270 124C 080A 0802"
    $"1802 2002 2002 2004 1004 0808 0408 0408",
    $"0000 1800 3C00 3D80 1FF0 1FFC 0FFE 0FFE"
    $"1FFE 3FFE 3FFE 3FFC 1FFC 0FF8 07F8 07F8",
    {9, 8}
};

resource 'CURS' (302, preload) {
    $"0180 1A40 2640 2640 1270 124C 080A 0802"
    $"1802 2002 2002 2004 1004 0808 0408 0408",
    $"0180 1BC0 3FC0 3FC0 1FF0 1FFC 0FFE 0FFE"
    $"1FFE 3FFE 3FFE 3FFC 1FFC 0FF8 07F8 07F8",
    {9, 8}
};

resource 'CURS' (303, preload) {
    $"0180 1A70 2648 2648 1248 124C 080A 0802"
    $"1802 2002 2002 2004 1004 0808 0408 0408",
    $"0180 1BF0 3FF8 3FF8 1FF8 1FFC 0FFE 0FFE"
    $"1FFE 3FFE 3FFE 3FFC 1FFC 0FF8 07F8 07F8",
    {9, 8}
};

resource 'CURS' (304, preload) {
    $"0180 1A70 2648 264A 124D 1249 0809 0801"
    $"1802 2002 2002 2004 1004 0808 0408 0408",
    $"0180 1BF0 3FF8 3FFA 1FFF 1FFF 0FFF 0FFF"
    $"1FFE 3FFE 3FFE 3FFC 1FFC 0FF8 07F8 07F8",
    {9, 8}
};

resource 'CURS' (305, preload) {
    $"0180 1A70 2648 264A 124D 1249 6809 9801"
    $"8802 4002 2002 2004 1004 0808 0408 0408",
    $"0180 1BF0 3FF8 3FFA 1FFF 1FFF 7FFF FFFF"
    $"FFFE 7FFE 3FFE 3FFC 1FFC 0FF8 07F8 07F8",
    {9, 8}
};

resource 'CURS' (263, preload) {
    $"3F00 3F00 3F00 3F00 4080 8040 9040 8860"
    $"9C60 8040 8040 4080 3F00 3F00 3F00 3F",
    $"3F00 3F00 3F00 3F00 7F80 FFC0 FFC0 FFC0"
    $"FFC0 FFC0 FFC0 7F80 3F00 3F00 3F00 3F",
    {8, 8}
};

resource 'CURS' (262, preload) {
    $"3F00 3F00 3F00 3F00 4080 8040 8040 8060"
    $"BC60 8040 8040 4080 3F00 3F00 3F00 3F",
    $"3F00 3F00 3F00 3F00 7F80 FFC0 FFC0 FFC0"
    $"FFC0 FFC0 FFC0 7F80 3F00 3F00 3F00 3F",
    {8, 8}
};

resource 'CURS' (261, preload) {
    $"3F00 3F00 3F00 3F00 4080 8040 8040 8060"
    $"9C60 8840 9040 4080 3F00 3F00 3F00 3F",
    $"3F00 3F00 3F00 3F00 7F80 FFC0 FFC0 FFC0"
    $"FFC0 FFC0 FFC0 7F80 3F00 3F00 3F00 3F",
    {8, 8}
};

resource 'CURS' (260, preload) {
    $"3F00 3F00 3F00 3F00 4080 8040 8040 8060"
    $"9C60 8440 8440 4080 3F00 3F00 3F00 3F",
    $"3F00 3F00 3F00 3F00 7F80 FFC0 FFC0 FFC0"
    $"FFC0 FFC0 FFC0 7F80 3F00 3F00 3F00 3F",
    {8, 8}
};

resource 'CURS' (259, preload) {
    $"3F00 3F00 3F00 3F00 4080 8040 8040 8060"
    $"9C60 8240 8040 4080 3F00 3F00 3F00 3F",
    $"3F00 3F00 3F00 3F00 7F80 FFC0 FFC0 FFC0"
    $"FFC0 FFC0 FFC0 7F80 3F00 3F00 3F00 3F",
    {8, 8}
};

resource 'CURS' (258, preload) {
    $"3F00 3F00 3F00 3F00 4080 8040 8040 8060"
    $"9F60 8040 8040 4080 3F00 3F00 3F00 3F",
    $"3F00 3F00 3F00 3F00 7F80 FFC0 FFC0 FFC0"
    $"FFC0 FFC0 FFC0 7F80 3F00 3F00 3F00 3F",
    {8, 8}
};

resource 'CURS' (257, preload) {
    $"3F00 3F00 3F00 3F00 4080 8040 8140 8260"
    $"9C60 8040 8040 4080 3F00 3F00 3F00 3F",
    $"3F00 3F00 3F00 3F00 7F80 FFC0 FFC0 FFC0"
    $"FFC0 FFC0 FFC0 7F80 3F00 3F00 3F00 3F",
    {8, 8}
};

resource 'CURS' (783, "film1") {
    $"FFFF 8FF1 A005 8FF1 AFF5 8FF1 AFF5 8FF1"
    $"AFF5 8001 AFF5 8FF1 AFF5 8FF1 AFF5 FFFF",
    $"",
    {0, 0}
};

resource 'CURS' (784, "film2") {
    $"FFFF 8FF1 AFF5 8001 AFF5 8FF1 AFF5 8FF1"
    $"AFF5 8FF1 A005 8FF1 AFF5 8FF1 AFF5 FFFF",
    $"",
    {0, 0}
};

resource 'CURS' (785, "film3") {
    $"FFFF 8FF1 AFF5 8FF1 A005 8FF1 AFF5 8FF1"
    $"AFF5 8FF1 AFF5 8001 AFF5 8FF1 AFF5 FFFF",
    $"FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF"
    $"FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF",
    {0, 0}
};

resource 'CURS' (786, "film4") {
    $"FFFF 8FF1 AFF5 8FF1 AFF5 8001 AFF5 8FF1"
    $"AFF5 8FF1 AFF5 8FF1 A005 8FF1 AFF5 FFFF",
    $"FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF"
    $"FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF",
    {0, 0}
};

resource 'CURS' (787, "film5") {
    $"FFFF 8FF1 AFF5 8FF1 AFF5 8FF1 A005 8FF1"
    $"AFF5 8FF1 AFF5 8FF1 AFF5 8001 AFF5 FFFF",
    $"FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF"
    $"FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF",
    {0, 0}
};

resource 'CURS' (788, "film6") {
    $"FFFF 8FF1 AFF5 8FF1 AFF5 8FF1 AFF5 8001"
    $"AFF5 8FF1 AFF5 8FF1 AFF5 8FF1 A005 FFFF",
    $"FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF"
    $"FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF",
    {0, 0}
};

resource 'CURS' (789, "film7") {
    $"FFFF 8001 AFF5 8FF1 AFF5 8FF1 AFF5 8FF1"
    $"A005 8FF1 AFF5 8FF1 AFF5 8FF1 AFF5 FFFF",
    $"FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF"
    $"FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF",
    {0, 0}
};

resource 'CURS' (131) {
    $"07C0 1830 2008 701C 783C FC7E FEFE FFFE"
    $"FEFE FC7E 783C 701C 2008 1830 07C0",
    $"07C0 1FF0 3FF8 7FFC 7FFC FFFE FFFE FFFE"
    $"FFFE FFFE 7FFC 7FFC 3FF8 1FF0 07C0",
    {7, 7}
};

resource 'CURS' (128) {
    $"07C0 19F0 21F8 41FC 41FC 81FE 81FE FFFE"
    $"FF02 FF02 7F04 7F04 3F08 1F30 07C0",
    $"07C0 1FF0 3FF8 7FFC 7FFC FFFE FFFE FFFE"
    $"FFFE FFFE 7FFC 7FFC 3FF8 1FF0 07C0",
    {7, 7}
};

resource 'CURS' (129) {
    $"07C0 1FF0 3FF8 5FF4 4FE4 87C2 8382 8102"
    $"8382 87C2 4FE4 5FF4 3FF8 1FF0 07C0",
    $"07C0 1FF0 3FF8 7FFC 7FFC FFFE FFFE FFFE"
    $"FFFE FFFE 7FFC 7FFC 3FF8 1FF0 07C0",
    {7, 7}
};

resource 'CURS' (130) {
    $"07C0 1F30 3F08 7F04 7F04 FF02 FF02 FFFE"
    $"81FE 81FE 41FC 41FC 21F8 19F0 07C0",
    $"07C0 1FF0 3FF8 7FFC 7FFC FFFE FFFE FFFE"
    $"FFFE FFFE 7FFC 7FFC 3FF8 1FF0 07C0",
    {7, 7}

};

resource 'acur' (1000, "Counting Hand", preload) {
    {       /* array CursIdArray: 10 elements */
            /* [1] */
            300,
            /* [2] */
            301,
            /* [3] */
            302,
            /* [4] */
            303,
            /* [5] */
            304,
            /* [6] */
            305,
            /* [7] */
            304,
            /* [8] */
            303,
            /* [9] */
            302,
            /* [10] */
            301
    }
};

resource 'acur' (1002, "Watch", preload) {
    {       /* array CursIdArray: 7 elements */
            /* [1] */
            257,
            /* [2] */
            258,
            /* [3] */
            259,
            /* [4] */
            260,
            /* [5] */
            261,
            /* [6] */
            262,
            /* [7] */
            263
    }
};

resource 'acur' (1004, "MovieStrip", preload) {
    {       /* array CursIdArray: 7 elements */
            /* [1] */
            783,
            /* [2] */
            784,
            /* [3] */
            785,
            /* [4] */
            786,
            /* [5] */
            787,
            /* [6] */
            788,
            /* [7] */
            789
    }
};

resource 'acur' (1003, "BeachBall", preload) {
    {       /* array CursIdArray: 4 elements */
            /* [1] */
            128,
            /* [2] */
            129,
            /* [3] */
            130,
            /* [4] */
            131
    }
```

```
};
```