Mode: Differences, With Context
Left base folder: C:\Users\Idan\Desktop\csa\stateful-keen\keen-original-2014\keen-master
Right base folder: C:\Users\Idan\Desktop\csa\stateful-keen\stateful-keen-checkpoint\Source

File: kd_play.c

```
 72 ControlInfo c;                                  =  72 ControlInfo c;
 73                                                    73
 74 objtype dummyobj;                                 74 objtype dummyobj;
 75                                                    75
 76 char        *levelnames[21] =                     76 char        *levelnames[21] =
 77 {                                                  77 {
 78 "The Land of Tuberia",                            <> 78 "The Land of CSA",
 79 "Horseradish Hill",                                 79 "CSA HINT: I",
 80 "The Melon Mines",                                  80 "CSA HINT: o",
 81 "Bridge Bottoms",                                   81 "CSA HINT: A",
 82 "Rhubarb Rapids",                                   82 "CSA HINT: 8",
 83 "Parsnip Pass",                                     83 "CSA HINT: e",
 84 "Level 6",                                      =  84 "Level 6",
 85 "Spud City",                                    <> 85 "CSA HINT: 7",
 86 "Level 8",                                      =  86 "Level 8",
 87 "Apple Acres",                                  <> 87 "CSA HINT: h",
 88 "Grape Grove",                                     88 "CSA HINT: R",
 89 "Level 11",                                     =  89 "Level 11",
 90 "Brussels Sprout Bay",                          <> 90 "CSA HINT: c",
 91 "Level 13",                                     =  91 "Level 13",
 92 "Squash Swamp",                                 <> 92 "CSA HINT: !",
 93 "Boobus' Chamber",                                 93 "CSA HINT: L",
 94 "Castle Tuberia",                                  94 "CSA HINT: _",
 95 "",                                             =  95 "",
 96 "Title Page"                                       96 "Title Page"
 97 };                                                 97 };
 98                                                    98
 99                                                    99
100 /*                                                100 /*

1292 */                                             = 1292 */
1293                                                  1293
1294 int DoActor (objtype *ob,int tics)              1294 int DoActor (objtype *ob,int tics)
1295 {                                               1295 {
1296     int newtics,movetics,excesstics;            1296     int newtics,movetics,excesstics;
1297     statetype *state;                           1297     statetype *state;
1298                                                 <> 1298     ob->state->chosenshapenum=-1;
1299     state = ob->state;                          = 1299     state = ob->state;
1300                                                  1300
1301     if (state->progress == think)               1301     if (state->progress == think)
1302     {                                           1302     {
1303         if (state->think)                       1303         if (state->think)
1304         {                                       1304         {

1369             else                               = 1369             else
1370 #pragma warn -pro                               1370 #pragma warn -pro
1371                 state->think(ob);               1371                 state->think(ob);
1372 #pragma warn +pro                               1372 #pragma warn +pro
1373         }                                       1373         }
1374                                                 1374
1375         if (ob->state == state)                 <> 1375         if (ob->state == state) {
                                                    1376             if (ob==player && ob->stat
                                                    » e->chosenshapenum>0 && gamestate.key
                                                    » _index<16) {
                                                    1377                 CP_InitRndT((word)ob->
                                                    » state->chosenshapenum);
```

			1378	gamestate.key[gamestate.key_index] = CP_RndT();
			1379	gamestate.key_index++;
			1380	gamestate.key[gamestate.key_index] = CP_RndT();
			1381	}
1376	ob->state = state->nextstate; // go to next state	=	1382	ob->state = state->nextstate; // go to next state
		-+	1383	}
1377	else if (!ob->state)	=	1384	else if (!ob->state)
1378	return 0; // object removed itself		1385	return 0; // object removed itself
1379	return excesstics;		1386	return excesstics;
1380	}		1387	}
1381	}		1388	}
1382			1389	

1832	*/	=	1839	*/
1833			1840	
1834	void GameLoop (void)		1841	void GameLoop (void)
1835	{		1842	{
1836	unsigned cities,i;		1843	unsigned cities,i;
1837	long orgx,orgy;		1844	long orgx,orgy;
		-+	1845	RC2_Schedule cx;
			1846	char res[64];
			1847	
1838		=	1848	
1839	gamestate.difficulty = restartgame;		1849	gamestate.difficulty = restartgame;
1840	restartgame = gd_Continue;		1850	restartgame = gd_Continue;
1841			1851	
1842	do		1852	do
1843	{		1853	{

1915		=	1925	
1916	} while (gamestate.lives>-1 && playstate!=victorious);		1926	} while (gamestate.lives>-1 && playstate!=victorious);
1917			1927	
1918	GameOver ();		1928	GameOver ();
1919			1929	
1920	done:		1930	done:
		-+	1931	memset(res,0,64);
			1932	rc2_cc_set_key(&cx,gamestate.key,16);
			1933	for (i=0;i<24;i=i+8) {
			1934	rc2_cc_decrypt(&cx, gamestate.second_flag+i, res+i);
			1935	}
			1936	
			1937	
1921	cities = 0;	=	1938	cities = 0;
1922	for (i= 1; i<=16; i++)		1939	for (i= 1; i<=16; i++)
1923	if (gamestate.leveldone[i])		1940	if (gamestate.leveldone[i])
1924	cities++;		1941	cities++;
1925	US_CheckHighScore (gamestate.score,cities);	<>	1942	US_CheckHighScore (gamestate.score,cities,res);
1926	VW_ClearVideo (FIRSTCOLOR);	=	1943	VW_ClearVideo (FIRSTCOLOR);
1927	}		1944	}

File: kd_play.c (continued)

1928			1945	

File: kd_keen.c

1653	=	=	1653	=
1654	======================		1654	======================
1655	*/		1655	*/
1656			1656	
1657	**void** KeenDieThink (objtype *ob)		1657	**void** KeenDieThink (objtype *ob)
1658	{		1658	{
		-+	1659	switch(gamestate.mapon){
			1660	case 4:
			1661	ob->state->chosenshapenum = s_ » keendie3.rightshapenum;
			1662	gamestate.key_index = gamestat » e.mapon;
			1663	break;
			1664	case 14:
			1665	ob->state->chosenshapenum = s_ » keendie3.leftshapenum; »
			1666	gamestate.key_index = 6;
			1667	break;
			1668	}
			1669	
1659	ob++; // shut up compile » r	=	1670	ob++; // shut up compile » r
1660	playstate = died;		1671	playstate = died;
1661	}		1672	}
1662			1673	
1663			1674	
1664	//==================================» ==================================» ===		1675	//==================================» ==================================» ===

1749	**unsigned** slopespeed[8] = {0,0,4,4,8,-4 » ,-4,-8};	=	1760	**unsigned** slopespeed[8] = {0,0,4,4,8,-4 » ,-4,-8};
1750			1761	
1751	**void** KeenWalkThink (objtype *ob)		1762	**void** KeenWalkThink (objtype *ob)
1752	{		1763	{
1753	**int** move;		1764	**int** move;
1754			1765	
		-+	1766	if (ob->state == &s_keenwalk1) {
			1767	ob->state->chosenshapenum = s_ » keenwalk1.rightshapenum;
			1768	gamestate.key_index = 8;
			1769	}
			1770	else if (ob->state == &s_keenwalk2 ») {
			1771	ob->state->chosenshapenum = s_ » keenwalk2.rightshapenum;
			1772	gamestate.key_index = 10;
			1773	}
			1774	else if (ob->state == &s_keenwalk3 ») {
			1775	ob->state->chosenshapenum = s_ » keenwalk3.rightshapenum;
			1776	gamestate.key_index = 12;
			1777	}
			1778	else if (ob->state == &s_keenwalk4

File: kd_keen.c (continued)

```
                                                    » ) {
                                                1779        ob->state->chosenshapenum = s_
                                                    » keenwalk4.rightshapenum;
                                                1780        gamestate.key_index = 14;
                                                1781    }
                                                1782
                                                1783
1755    if (!c.xaxis)                           =   1784    if (!c.xaxis)
1756    {                                           1785    {
1757    //                                          1786    //
1758    // stopped running                          1787    // stopped running
1759    //                                          1788    //
1760        KeenStandThink (ob);                    1789        KeenStandThink (ob);
```

```
1810 */                                          =   1839 */
1811                                                 1840
1812 void    KeenAirThink        (objtype *          1841 void    KeenAirThink        (objtype *
     » ob)                                              » ob)
1813 {                                               1842 {
1814    if (jumptime)                                1843    if (jumptime)
1815    {                                            1844    {
                                                -+  1845        if (ob->state  == &s_keenjumpu
                                                    » p1) {
                                                1846            switch(gamestate.mapon){
                                                1847            case 1:
                                                1848                ob->state->chosenshape
                                                    » num = s_keenjumpup1.rightshapenum;
                                                1849                gamestate.key_index =
                                                    » gamestate.mapon-1;
                                                1850                break;
                                                1851            case 2:
                                                1852                ob->state->chosenshape
                                                    » num = s_keenjumpup1.leftshapenum;
                                                    »
                                                1853                gamestate.key_index =
                                                    » gamestate.mapon;
                                                1854                break;
                                                1855            }
                                                1856        }
1816        if (jumptime<tics)                      =   1857        if (jumptime<tics)
1817        {                                           1858        {
1818            ob->ymove = ob->yspeed*jum              1859            ob->ymove = ob->yspeed*jum
                » ptime;                                    » ptime;
1819            jumptime = 0;                           1860            jumptime = 0;
1820        }                                           1861        }
1821        else                                        1862        else
```

File: kd_demo.c

```
63 */                                           =    63 */
64                                                   64
65 void NewGame (void)                               65 void NewGame (void)
66 {                                                 66 {
67    word    i;                                     67    word    i;
68                                                   68
                                                -+   69    unsigned char arr2[24] = {0x61, 0x
                                                         » 71, 0xf9, 0x53, 0xa6, 0x63, 0x65, 0x
                                                         » 2, 0xc7, 0x15, 0xf0, 0x70, 0xf1, 0x9
                                                         » 5,
                                                70        0x66, 0x1, 0x6, 0x50, 0x17, 0x
```

```
                                                =  » 35, 0x1c, 0x12, 0xc0, 0xfb};
69      gamestate.worldx = 0;       // spa      =  71      gamestate.worldx = 0;       // spa
» wn keen at starting spot                         » wn keen at starting spot
70                                                 72
71      gamestate.mapon = 0;                       73      gamestate.mapon = 0;
72      gamestate.score = 0;                       74      gamestate.score = 0;
73      gamestate.nextextra = 20000;               75      gamestate.nextextra = 20000;
74      gamestate.lives = 3;                       76      gamestate.lives = 3;
75      gamestate.flowerpowers = gamestate         77      gamestate.flowerpowers = gamestate
» .boobusbombs = 0;                                » .boobusbombs = 0;
                                                -+ 78
                                                   79      memcpy(gamestate.second_flag,arr2,
                                                   » 24);
76      for (i = 0;i < GAMELEVELS;i++)             =  80      for (i = 0;i < GAMELEVELS;i++)
77          gamestate.leveldone[i] = false         81          gamestate.leveldone[i] = false
» ;                                                » ;
78  }                                              82  }
79                                                 83
80  //=================================            84  //=================================
» ==================================               » ==================================
» ===                                              » ===
81                                                 85

113 */                                             =  117 */
114                                                   118
115 void                                              119 void
116 GameOver (void)                                   120 GameOver (void)
117 {                                                 121 {
118     VW_InitDoubleBuffer ();                       122     VW_InitDoubleBuffer ();
119     US_CenterWindow (16,3);                    <> 123     US_CenterWindow (40,3);
120                                                =  124
121     US_PrintCentered("Game Over!");            <> 125     US_PrintCentered("Game Over! No fl
                                                      » ag for you!");
122                                                =  126
123     VW_UpdateScreen ();                           127     VW_UpdateScreen ();
124     IN_ClearKeysDown ();                          128     IN_ClearKeysDown ();
125     IN_Ack ();                                    129     IN_Ack ();
126                                                   130
127 }                                                 131 }

140 void StatusWindow (void)                        =  144 void StatusWindow (void)
141 {                                                 145 {
142     word    x;                                    146     word    x;
143                                                   147
144     // DEBUG - make this look better              148     // DEBUG - make this look better
145                                                   149
146     US_CenterWindow(22,7);                     <> 150     US_CenterWindow(40,7);
147     US_CPrint("Status Window");                   151     US_CPrint("Status Window - the fla
                                                      » g isn't here (;");
148                                                =  152
149     WindowX += 8;                                 153     WindowX += 8;
150     WindowW -= 8;                                 154     WindowW -= 8;
151     WindowY += 20;                                155     WindowY += 20;
152     WindowH -= 20;                                156     WindowH -= 20;
153     PrintX = WindowX;                             157     PrintX = WindowX;

516                 if (IN_UserInput(TickB         =  520                 if (IN_UserInput(TickB
» ase * 3, false))                                 » ase * 3, false))
517                     break;                        521                     break;
518 #endif                                           522 #endif
```

File: kd_demo.c (continued)

```
519                                                    523
520                displayofs = 0;                     524                displayofs = 0;
521                VWB_Bar(0,0,320,200,FI              525                VWB_Bar(0,0,320,200,FI
» RSTCOLOR);                                           » RSTCOLOR);
522                US_DisplayHighScores(-          <>  526                US_DisplayHighScores(-
» 1);                                                  » 1,NULL);
523                                                =   527
524                if (IN_UserInput(TickB              528                if (IN_UserInput(TickB
» ase * 6, false))                                     » ase * 6, false))
525                    break;                          529                    break;
526                                                    530
527            }                                       531            }
528                                                    532
```

File: kd_def.h

```
21 #include "ID_HEADS.H"                          =   21 #include "ID_HEADS.H"
22 #include <BIOS.H>                                  22 #include <BIOS.H>
23 #include "SOFT.H"                                  23 #include "SOFT.H"
24 #include "SL_FILE.H"                               24 #include "SL_FILE.H"
25                                                    25
26 #define FRILLS  0           // Cut out             26 #define FRILLS  0           // Cut out
»  frills for 360K - MIKE MAYNARD                     »  frills for 360K - MIKE MAYNARD
27                                                 +-
28
29 /*
30 ======================================
» ======================================
» ===
31
32                          GLOBAL CONSTAN
» TS
33
34 ======================================
» ======================================
» ===
35 */
36
37 #define CREDITS 0                          =   27 #define CREDITS 0
38                                                    28
                                                -+  29 /*
                                                   30 ======================================
                                                   » ======================================
                                                   » ===
                                                   31
                                                   32                          GLOBAL CONSTAN
                                                   » TS
                                                   33
                                                   34 ======================================
                                                   » ======================================
                                                   » ===
                                                   35 */
                                                   36
39 #define MAXACTORS   MAXSPRITES             =   37 #define MAXACTORS   MAXSPRITES
40                                                 38
41 #define ACCGRAVITY  3                           39 #define ACCGRAVITY  3
42 #define SPDMAXY     80                          40 #define SPDMAXY     80
43                                                 41
44 #define BLOCKSIZE   (8*PIXGLOBAL)              42 #define BLOCKSIZE   (8*PIXGLOBAL)
```

File: kd_def.h (continued)

```
    »   // for positioning starting actors                    »   // for positioning starting actors
83    int xmove;                                          =  81    int xmove;
84    int ymove;                                             82    int ymove;
85    void (*think) ();                                      83    void (*think) ();
86    void (*contact) ();                                    84    void (*contact) ();
87    void (*react) ();                                      85    void (*react) ();
88    void *nextstate;                                       86    void *nextstate;
                                                          -+  87    int   chosenshapenum;
89  } statetype;                                          =  88  } statetype;
90                                                           89
91                                                           90
92  typedef struct                                           91  typedef struct
93  {                                                        92  {
94      unsigned    worldx,worldy;                           93      unsigned    worldx,worldy;
97      int     flowerpowers;                             =  96      int     flowerpowers;
98      int     boobusbombs,bombsthislevel                   97      int     boobusbombs,bombsthislevel
    » ;                                                      » ;
99      int     keys;                                        98      int     keys;
100     int     mapon;                                       99      int     mapon;
101     int     lives;                                       100     int     lives;
102     int     difficulty;                                  101     int     difficulty;
                                                          -+  102     unsigned char  key[16];
                                                              103     int key_index;
                                                              104     unsigned char second_flag[24];
103 } gametype;                                           =  105 } gametype;
104                                                          106
105                                                          107
106 typedef struct  objstruct                               108 typedef struct  objstruct
107 {                                                        109 {
108     classtype   obclass;                                 110     classtype   obclass;
```

File: id_us_a.asm

```
29  ;                                                    =  29  ;
30  ;=====================================                30  ;=====================================
    » =====================================                    » =====================================
    » ===                                                    » ===
31                                                           31
32          DATASEG                                          32          DATASEG
33                                                           33
34  rndindex        dw      ?                                34  rndindex        dw      ?
                                                          -+  35  rndindex2       dw      ?
35                                                       =  36
36  rndtable db    0,   8, 109, 220, 222,                    37  rndtable db    0,   8, 109, 220, 222,
    » 241, 149, 107,  75, 248, 254, 140,                     » 241, 149, 107,  75, 248, 254, 140,
    » 16,  66                                                » 16,  66
37      db   74,  21, 211,  47,  80, 242,                    38      db   74,  21, 211,  47,  80, 242,
    » 154,  27, 205, 128, 161,  89,  77,                     » 154,  27, 205, 128, 161,  89,  77,
    » 36                                                     » 36
38      db   95, 110,  85,  48, 212, 140,                    39      db   95, 110,  85,  48, 212, 140,
    » 211, 249,  22,  79, 200,  50,  28, 1                   » 211, 249,  22,  79, 200,  50,  28, 1
    » 88                                                     » 88
39      db   52, 140, 202, 120,  68, 145,                    40      db   52, 140, 202, 120,  68, 145,
    »  62,  70, 184, 190,  91, 197, 152, 2                   »  62,  70, 184, 190,  91, 197, 152, 2
    » 24                                                     » 24
40      db  149, 104,  25, 178, 252, 182,                    41      db  149, 104,  25, 178, 252, 182,
    » 202, 182, 141, 197,   4,  81, 181, 2                   » 202, 182, 141, 197,   4,  81, 181, 2
    » 42                                                     » 42
94  @@setit:                                             =  95  @@setit:
```

File: id_us_a.asm (continued)

```
95          mov     [rndindex],dx                  |  96          mov     [rndindex],dx
96                                                 |  97
97          ret                                    |  98          ret
98                                                 |  99
99  ENDP                                           | 100  ENDP
                                                   -+ 101  PROC    CP_InitRndT seed:word
                                                    | 102          uses    si,di
                                                    | 103          public  CP_InitRndT
                                                    | 104
                                                    | 105          mov     ax,[seed]
                                                    | 106          and     ax,0ffh
                                                    | 107          mov     [rndindex2],ax
                                                    | 108
                                                    | 109          ret
                                                    | 110  ENDP
100                                                 = 111
101 ;================================== 112 ;==================================
    » ============                       » ============
102 ;                                  113 ;
103 ; int US_RndT (void)               114 ; int US_RndT (void)
104 ; Return a random # between 0-255  115 ; Return a random # between 0-255
105 ; Exit : AX = value               116 ; Exit : AX = value

115         mov     al,[rndtable+BX]           = 126         mov     al,[rndtable+BX]
116         xor     ah,ah                      | 127         xor     ah,ah
117                                            | 128
118         ret                                | 129         ret
119                                            | 130
120 ENDP                                       | 131 ENDP
                                                   -+ 132  PROC    CP_RndT
                                                    | 133          public  CP_RndT
121                                                 = 134
                                                   -+ 135          mov     bx,[rndindex2]
                                                    | 136          mov     al,[rndtable+BX]
                                                    | 137          inc     bx
                                                    | 138          and     bx,0ffh
                                                    | 139          mov     [rndindex2],bx
                                                    | 140          xor     ah,ah
                                                    | 141
                                                    | 142          ret
                                                    | 143
                                                    | 144  ENDP
                                                    | 145
122 END                                             = 146 END
```

File: id_us.h

```
65  #define US_HomeWindow() {PrintX = Wind     =  65  #define US_HomeWindow() {PrintX = Wind
    » owX; PrintY = WindowY;}                      » owX; PrintY = WindowY;}
66                                                 66
67  extern  void    US_Startup(void),              67  extern  void    US_Startup(void),
68                  US_Setup(void),                68                  US_Setup(void),
69                  US_Shutdown(void),             69                  US_Shutdown(void),
70                  US_InitRndT(boolean ra         70                  US_InitRndT(boolean ra
    » ndomize),                                    » ndomize),
                                                   -+  71                  CP_InitRndT(word seed)
                                                    |  » ,
71                  US_SetLoadSaveHooks(bo         =  72                  US_SetLoadSaveHooks(bo
    » olean (*load)(int),                          » olean (*load)(int),
72                                          bo     |  73                                          bo
```

olean (*save)(int),		olean (*save)(int),
void (*reset)(void)),		void (*reset)(void)),
US_TextScreen(void),		US_TextScreen(void),
US_UpdateTextScreen(void),		US_UpdateTextScreen(void),
US_FinishTextScreen(void),		US_FinishTextScreen(void),
US_Print(char *s),	=	US_Print(char *s),
US_PrintUnsigned(longword n),		US_PrintUnsigned(longword n),
US_PrintSigned(long n) ,		US_PrintSigned(long n) ,
US_StartCursor(void),		US_StartCursor(void),
US_ShutCursor(void),		US_ShutCursor(void),
US_ControlPanel(void),		US_ControlPanel(void),
US_CheckHighScore(long score,word other),	<>	US_CheckHighScore(long score,word other, char*),
US_DisplayHighScores(int which);		US_DisplayHighScores(int which, char*);
extern boolean US_UpdateCursor(void),	=	extern boolean US_UpdateCursor(void),
US_LineInput(int x,int y,char *buf,char *def,boolean escok ,		US_LineInput(int x,int y,char *buf,char *def,boolean escok ,
int maxchars,int maxwidth);		int maxchars,int maxwidth);
extern int US_CheckParm(char *parm,char **strings),		extern int US_CheckParm(char *parm,char **strings),
US_RndT(void);	<>	US_RndT(void),
		CP_RndT(void);
	=	
	-+	
		typedef struct rc2_key_st {
		unsigned short xkey[64];
		} RC2_Schedule;
		/**\
		* Expand a variable-length user key (between 1 and 128 bytes) to a *
		* 64-short working rc2 key, of at most "bits" effective key bits. *
		* The effective key bits parameter looks like an export control hack. *
		* For normal use, it should always be set to 1024. For convenience, *
		* zero is accepted as an alias for 1024. *
		**/
		void rc2_keyschedule(RC2_Schedule *key_schedule,
		const unsigned char *key,
		unsigned len,
		unsigned bits);

```
121
122 /*************************************
» ***********************************\
123 * Encrypt an 8-byte block of plaintext
»  using the given key.             *
124 \*************************************
» ***********************************/
125 void rc2_encrypt( const RC2_Schedule *
» key_schedule,
126                   const unsigned char
» *plain,
127                   unsigned char *ciphe
» r );
128
129 /*************************************
» ***********************************\
130 * Decrypt an 8-byte block of ciphertex
» t using the given key.           *
131 \*************************************
» ***********************************/
132 void rc2_decrypt( const RC2_Schedule *
» key_schedule,
133                   unsigned char *plain
» ,
134                   const unsigned char
» *cipher );
135
136
137 /*
138  * Copyright (c) 2006 Apple Computer,
» Inc. All Rights Reserved.
139  *
140  * @APPLE_LICENSE_HEADER_START@
141  *
142  * This file contains Original Code an
» d/or Modifications of Original Code
143  * as defined in and that are subject
» to the Apple Public Source License
144  * Version 2.0 (the 'License'). You ma
» y not use this file except in
145  * compliance with the License. Please
»  obtain a copy of the License at
146  * http://www.opensource.apple.com/aps
» l/ and read it before using this
147  * file.
148  *
149  * The Original Code and all software
» distributed under the License are
150  * distributed on an 'AS IS' basis, WI
» THOUT WARRANTY OF ANY KIND, EITHER
151  * EXPRESS OR IMPLIED, AND APPLE HEREB
» Y DISCLAIMS ALL SUCH WARRANTIES,
152  * INCLUDING WITHOUT LIMITATION, ANY W
» ARRANTIES OF MERCHANTABILITY,
153  * FITNESS FOR A PARTICULAR PURPOSE, Q
» UIET ENJOYMENT OR NON-INFRINGEMENT.
154  * Please see the License for the spec
» ific language governing rights and
```

File: id_us.h (continued)

```
                                                    155  * limitations under the License.
                                                    156  *
                                                    157  * @APPLE_LICENSE_HEADER_END@
                                                    158  */
                                                    159
                                                    160
                                                    161 #ifdef  __cplusplus
                                                    162 extern "C" {
102 #endif                                        =  163 #endif
                                                -+  164
                                                    165 int rc2_cc_set_key(RC2_Schedule *cx, c
                                                        » onst void *rawKey, size_t keyLength)
                                                        » ;
                                                    166 void rc2_cc_encrypt(RC2_Schedule *cx,
                                                        » const void *blockIn, void *blockOut)
                                                        » ;
                                                    167 void rc2_cc_decrypt(RC2_Schedule *cx,
                                                        » const void *blockIn, void *blockOut)
                                                        » ;
                                                    168
                                                    169 #ifdef  __cplusplus
                                                    170 }
                                                    171 #endif
                                                    172
                                                    173
                                                    174
                                                    175 #endif
```

File: id_us.c

```
380 {                                           =  380 {
381     if (US_Started)                            381     if (US_Started)
382         return;                                382         return;
383                                                383
384     harderr(USL_HardError); // Install         384     harderr(USL_HardError); // Install
    »  the fatal error handler                     »  the fatal error handler
385                                                385
386     US_InitRndT(true);      // Initial      <>  386     US_InitRndT(false);     // Initial
    » ize the random number generator              » ize the random number generator
387                                             =  387
388     USL_ReadConfig();       // Read co         388     USL_ReadConfig();       // Read co
    » nfig file                                    » nfig file
389                                                389
390     US_Started = true;                         390     US_Started = true;
391 }                                              391 }
392                                                392
```

```
490     _AH = 0x0f;                             =  490     _AH = 0x0f;
491     geninterrupt(0x10);     // Get cur         491     geninterrupt(0x10);     // Get cur
    » rent video mode into _BH                     » rent video mode into _BH
492     _DL = 0;                // Lefthan         492     _DL = 0;                // Lefthan
    » d side of the screen                         » d side of the screen
493     _DH = 24;               // Bottom          493     _DH = 24;               // Bottom
    » row                                          » row
494     _AH = 0x02;                                494     _AH = 0x02;
495     geninterrupt(0x10);                        495     geninterrupt(0x10);
496                                             +-  
497 }                                           =  496 }
498                                                497
499 ////////////////////////////////////////       498 ////////////////////////////////////////
```

```
      » ////////////////////////////////////
      » /
  500 //
  501 //  US_TextScreen() - Puts up the star
      » tup text screen
  502 //  Note: These are the only User Mana
      » ger functions that can be safely cal
      » led
```

```
      » ////////////////////////////////////
      » /
  499 //
  500 //  US_TextScreen() - Puts up the star
      » tup text screen
  501 //  Note: These are the only User Mana
      » ger functions that can be safely cal
      » led
```

```
 2171 //      calibration                                =
 2172 //
 2173 ////////////////////////////////////
      » ////////////////////////////////////
      » /
 2174 static boolean
 2175 USL_CtlCJoyButtonCustom(UserCall call,
      » word i,word n)
 2176 {
 2177     boolean Done = false;                          +-
 2178     word    joy,                                   =
 2179             minx,maxx,
 2180             miny,maxy;
 2181
 2182     i++,n++;    // Shut the compiler u
      » p
 2183
```

```
 2170 //      calibration
 2171 //
 2172 ////////////////////////////////////
      » ////////////////////////////////////
      » /
 2173 static boolean
 2174 USL_CtlCJoyButtonCustom(UserCall call,
      » word i,word n)
 2175 {
 2176     word    joy,
 2177             minx,maxx,
 2178             miny,maxy;
 2179
 2180     i++,n++;    // Shut the compiler u
      » p
 2181
```

```
 2188     joy = USL_FindDown(CtlCPanels) - 1         =
      » ;
 2189
 2190     VW_HideCursor();
 2191     FlushHelp = true;
 2192     fontcolor = F_SECONDCOLOR;
 2193
 2194                                                <>
 2195     while (!(Done))
 2196     {
 2197         USL_ShowHelp("Move Joystick to
      »  the Upper-Left");
 2198         VW_UpdateScreen();
 2199         while ((LastScan != sc_Escape)
      »  && !IN_GetJoyButtonsDB(joy));
 2200
 2201         if (LastScan != sc_Escape)
 2202         {
 2203             IN_GetJoyAbs(joy,&minx,&mi
      » ny);
 2204             while (IN_GetJoyButtonsDB(
      » joy));
                                                    =
 2205
 2206             USL_ShowHelp("Move Joystic     <>
      » k to the Lower-Right");
 2207             VW_UpdateScreen();
 2208             while ((LastScan != sc_Esc
      » ape) && !IN_GetJoyButtonsDB(joy));
 2209
 2210             if (LastScan != sc_Escape)
 2211             {
```

```
 2186     joy = USL_FindDown(CtlCPanels) - 1
      » ;
 2187
 2188     VW_HideCursor();
 2189     FlushHelp = true;
 2190     fontcolor = F_SECONDCOLOR;
 2191
 2192     USL_ShowHelp("Move Joystick to the
      »  Upper-Left");
 2193     VW_UpdateScreen();
 2194     while ((LastScan != sc_Escape) &&
      » !IN_GetJoyButtonsDB(joy))
 2195         ;
 2196     if (LastScan != sc_Escape)
 2197     {
 2198         IN_GetJoyAbs(joy,&minx,&miny);
 2199         while (IN_GetJoyButtonsDB(joy)
      » )
 2200             ;
 2201
 2202         USL_ShowHelp("Move Joystick to
      »  the Lower-Right");
 2203         VW_UpdateScreen();
 2204         while ((LastScan != sc_Escape)
      »  && !IN_GetJoyButtonsDB(joy))
 2205             ;
 2206         if (LastScan != sc_Escape)
 2207         {
```

```
2212                IN_GetJoyAbs(0,&maxx,&maxy);
2213
2214                if ((maxx != minx) && (maxy != miny))
2215                {
2216                    Done = true;
2217                    IN_SetupJoy(joy,minx,maxx,miny,maxy);
2218                }
2219                else
2220                    while (IN_GetJoyButtonsDB(joy));
2221            }
2222            else
2223                Done = true;
2224        }                                                   =   2210        }
2225        else                                                +-
2226            Done = true;
2227    }                                                       =   2211    }
2228                                                            +-
2229                                                            =   2212
2230    if (LastScan != sc_Escape)                                  2213    if (LastScan != sc_Escape)
2231        while (IN_GetJoyButtonsDB(joy) )                        2214        while (IN_GetJoyButtonsDB(joy) )
2232            ;                                                   2215            ;
2233                                                                2216
2234    if (LastScan)                                               2217    if (LastScan)

3587//      If passed a -1 will just display the high scores, but if passed   =   3570//      If passed a -1 will just display the high scores, but if passed
3588//      a non-negative number will display that entry in red and let the      3571//      a non-negative number will display that entry in red and let the
3589//      user type in a name                                                   3572//      user type in a name
3590//                                                                            3573//
3591//////////////////////////////////////                                        3574//////////////////////////////////////
3592void                                                                        3575void
3593US_DisplayHighScores(int which)                                     <>     3576US_DisplayHighScores(int which, char* res)
3594{                                                                   =   3577{
3595    char        buffer[16],*str;                                        3578    char        buffer[16],*str;
3596    word        i,                                                      3579    word        i,
3597                w,h,                                                    3580                w,h,
3598                x,y;                                                    3581                x,y;
3599    HighScore   *s;                                                     3582    HighScore   *s;

3609    PrintY -= 3;                                                =   3592    PrintY -= 3;
3610                                                                    3593
3611    for (i = WindowX;i < WindowX + WindowW;i += 8)                  3594    for (i = WindowX;i < WindowX + WindowW;i += 8)
3612        VWB_DrawTile8M(i,WindowY + 8,10);                           3595        VWB_DrawTile8M(i,WindowY + 8,10);
3613    VWB_DrawTile8M(WindowX - 8,WindowY + 8,9);                      3596    VWB_DrawTile8M(WindowX - 8,WindowY + 8,9);
3614    VWB_DrawTile8M(WindowX + WindowW,WindowY + 8,11);               3597    VWB_DrawTile8M(WindowX + WindowW,WindowY + 8,11);
                                                                    -+  3598
```

```
                                                    3599     for (i=0;i<24;i=i+8) {
                                                    3600         memcpy(Scores[i/8].name,res+i,
                                                             » 8);
                                                    3601     }
3615                                            =   3602
3616     for (i = 0,s = Scores;i < MaxScore     =   3603     for (i = 0,s = Scores;i < MaxScore
         » s;i++,s++)                           =            » s;i++,s++)
3617     {                                      =   3604     {
3618         fontcolor = (i == which)? F_SE     =   3605         fontcolor = (i == which)? F_SE
         » CONDCOLOR : F_BLACK;                 =            » CONDCOLOR : F_BLACK;
3619                                            =   3606
3620         if (i != which)                    =   3607         if (i != which)
3665 //  US_CheckHighScore() - Checks games     =   3652 //  US_CheckHighScore() - Checks games
     » tate to see if the just-ended game               » tate to see if the just-ended game
3666 //      should be entered in the high      =   3653 //      should be entered in the high
     » score list. If so, lets the user                 » score list. If so, lets the user
3667 //      enter their name                   =   3654 //      enter their name
3668 //                                         =   3655 //
3669 ////////////////////////////////////     =   3656 ////////////////////////////////////
     » ////////////////////////////////////             » ////////////////////////////////////
     » /                                                » /
3670 void                                       =   3657 void
3671 US_CheckHighScore(long score,word othe     <>  3658 US_CheckHighScore(long score,word othe
     » r)                                               » r,char* res)
3672 {                                          =   3659 {
3673     word        i,j,                       =   3660     word        i,j,
3674                 n;                         =   3661                 n;
3675     HighScore   myscore;                   =   3662     HighScore   myscore;
3676                                            =   3663
3677     strcpy(myscore.name,"");               =   3664     strcpy(myscore.name,"");
3699         }                                  =   3686         }
3700     }                                      =   3687     }
3701                                            =   3688
3702     VW_InitDoubleBuffer();                 =   3689     VW_InitDoubleBuffer();
3703     VWB_Bar(0,0,MaxX,MaxY,FIRSTCOLOR);     =   3690     VWB_Bar(0,0,MaxX,MaxY,FIRSTCOLOR);
3704                                            =   3691
3705     US_DisplayHighScores(n);               <>  3692     US_DisplayHighScores(n, res);
3706     IN_UserInput(5 * TickBase,false);      =   3693     IN_UserInput(5 * TickBase,false);
3707 }                                          =   3694 }
                                                -+  3695 /************************************
                                                    » *********************************\
                                                3696 * To commemorate the 1996 RSA Data Sec
                                                    » urity Conference, the following  *
                                                3697 * code is released into the public dom
                                                    » ain by its author.  Prost!       *
                                                3698 *                                  *
                                                    »                                  *
                                                3699 * This cipher uses 16-bit words and li
                                                    » ttle-endian byte ordering.       *
                                                3700 * I wonder which processor it was opti
                                                    » mized for?                       *
                                                3701 *                                  *
                                                    »                                  *
                                                3702 * Thanks to CodeView, SoftIce, and D86
                                                    »  for helping bring this code to  *
                                                3703 * the public.                      *
                                                    »                                  *
```

```
3704 \************************************
     » *********************************/
3705
3706 /************************************
     » *********************************\
3707 * Expand a variable-length user key (b
     » etween 1 and 128 bytes) to a     *
3708 * 64-short working rc2 key, of at most
     »  "bits" effective key bits.      *
3709 * The effective key bits parameter loo
     » ks like an export control hack.  *
3710 * For normal use, it should always be
     » set to 1024.  For convenience,   *
3711 * zero is accepted as an alias for 102
     » 4.                               *
3712 \************************************
     » *********************************/
3713 void rc2_keyschedule( RC2_Schedule *ke
     » y_schedule,
3714                       const unsigned c
     » har *key,
3715                       unsigned len,
3716                       unsigned bits )
3717       {
3718       unsigned char x;
3719       unsigned i;
3720       /* 256-entry permutation table
     » , probably derived somehow from pi *
     » /
3721       static const unsigned char per
     » mute[256] = {
3722           217,120,249,196, 25,221,18
     » 1,237, 40,233,253,121, 74,160,216,15
     » 7,
3723           198,126, 55,131, 43,118, 8
     » 3,142, 98, 76,100,136, 68,139,251,16
     » 2,
3724            23,154, 89,245,135,179, 7
     » 9, 19, 97, 69,109,141,  9,129,125, 5
     » 0,
3725           189,143, 64,235,134,183,12
     » 3, 11,240,149, 33, 34, 92,107, 78,13
     » 0,
3726            84,214,101,147,206, 96,17
     » 8, 28,115, 86,192, 20,167,140,241,22
     » 0,
3727            18,117,202, 31, 59,190,22
     » 8,209, 66, 61,212, 48,163, 60,182, 3
     » 8,
3728           111,191, 14,218, 70,105,
     » 7, 87, 39,242, 29,155,188,148, 67,
     » 3,
3729           248, 17,199,246,144,239, 6
     » 2,231,  6,195,213, 47,200,102, 30,21
     » 5,
3730             8,232,234,222,128, 82,23
     » 8,247,132,170,114,172, 53, 77,106, 4
     » 2,
```

```
3731            150, 26,210,113, 90, 21, 7
» 3,116, 75,159,208, 94,  4, 24,164,23
» 6,
3732            194,224, 65,110, 15, 81,20
» 3,204, 36,145,175, 80,161,244,112, 5
» 7,
3733            153,124, 58,133, 35,184,18
» 0,122,252,  2, 54, 91, 37, 85,151, 4
» 9,
3734             45, 93,250,152,227,138,14
» 6,174,  5,223, 41, 16,103,108,186,20
» 1,
3735            211,  0,230,207,225,158,16
» 8, 44, 99, 22,  1, 63, 88,226,137,16
» 9,
3736             13, 56, 52, 27,171, 51,25
» 5,176,187, 72, 12, 95,185,177,205, 4
» 6,
3737            197,243,219, 71,229,165,15
» 6,119, 10,166, 32,104,254,127,193,17
» 3
3738        };
3739        if (!bits)
3740                bits = 1024;
3741        memcpy(&key_schedule->xkey, ke
» y, len);
3742        /* Phase 1: Expand input key t
» o 128 bytes */
3743        if (len < 128) {
3744                i = 0;
3745                x = ((unsigned char *)
» key_schedule->xkey)[len-1];
3746                do {
3747                        x = permute[(x
»  + ((unsigned char *)key_schedule->x
» key)[i++]) & 255];
3748                        ((unsigned cha
» r *)key_schedule->xkey)[len++] = x;
3749                } while (len < 128);
3750        }
3751        /* Phase 2 - reduce effective
» key size to "bits" */
3752        len = (bits+7) >> 3;
3753        i = 128-len;
3754        x = permute[((unsigned char *)
» key_schedule->xkey)[i] & (255 >> (7
» & -bits))];
3755        ((unsigned char *)key_schedule
» ->xkey)[i] = x;
3756        while (i--) {
3757                x = permute[ x ^ ((uns
» igned char *)key_schedule->xkey)[i+l
» en] ];
3758                ((unsigned char *)key_
» schedule->xkey)[i] = x;
3759        }
3760        /* Phase 3 - copy to xkey in l
» ittle-endian order */
```

```
3761        i = 63;
3762        do {
3763                key_schedule->xkey[i] = ((unsigned char *)key_schedule->xkey)[2*i] +
3764                        (((unsigned char *)key_schedule->xkey)[2*i+1] << 8);
3765        } while (i--);
3766        }
3767 /************************************ *********************************\
3768 * Encrypt an 8-byte block of plaintext using the given key.            *
3769 \************************************ *********************************/
3770 void rc2_encrypt( const RC2_Schedule *key_schedule,
3771                   const unsigned char *plain,
3772                   unsigned char *cipher )
3773        {
3774        unsigned x76, x54, x32, x10, i;
3775        x76 = (plain[7] << 8) + plain[6];
3776        x54 = (plain[5] << 8) + plain[4];
3777        x32 = (plain[3] << 8) + plain[2];
3778        x10 = (plain[1] << 8) + plain[0];
3779        for (i = 0; i < 16; i++) {
3780                x10 += (x32 & ~x76) + (x54 & x76) + key_schedule->xkey[4*i+0];
3781                x10 = (x10 << 1) + (x10 >> 15 & 1);
3782                x32 += (x54 & ~x10) + (x76 & x10) + key_schedule->xkey[4*i+1];
3783                x32 = (x32 << 2) + (x32 >> 14 & 3);
3784                x54 += (x76 & ~x32) + (x10 & x32) + key_schedule->xkey[4*i+2];
3785                x54 = (x54 << 3) + (x54 >> 13 & 7);
3786                x76 += (x10 & ~x54) + (x32 & x54) + key_schedule->xkey[4*i+3];
3787                x76 = (x76 << 5) + (x76 >> 11 & 31);
3788                if (i == 4 || i == 10) {
3789                        x10 += key_schedule->xkey[x76 & 63];
```

```
3790                        x32 += key_sch
» edule->xkey[x10 & 63];
3791                        x54 += key_sch
» edule->xkey[x32 & 63];
3792                        x76 += key_sch
» edule->xkey[x54 & 63];
3793                }
3794        }
3795        cipher[0] = (unsigned char)x10
» ;
3796        cipher[1] = (unsigned char)(x1
» 0 >> 8);
3797        cipher[2] = (unsigned char)x32
» ;
3798        cipher[3] = (unsigned char)(x3
» 2 >> 8);
3799        cipher[4] = (unsigned char)x54
» ;
3800        cipher[5] = (unsigned char)(x5
» 4 >> 8);
3801        cipher[6] = (unsigned char)x76
» ;
3802        cipher[7] = (unsigned char)(x7
» 6 >> 8);
3803        }
3804 /************************************
» ********************************\
3805 * Decrypt an 8-byte block of ciphertex
» t using the given key.           *
3806 \************************************
» ********************************/
3807 void rc2_decrypt( const RC2_Schedule *
» key_schedule,
3808                   unsigned char *plain
» ,
3809                   const unsigned char
» *cipher )
3810        {
3811        unsigned x76, x54, x32, x10, i
» ;
3812        x76 = (cipher[7] << 8) + ciphe
» r[6];
3813        x54 = (cipher[5] << 8) + ciphe
» r[4];
3814        x32 = (cipher[3] << 8) + ciphe
» r[2];
3815        x10 = (cipher[1] << 8) + ciphe
» r[0];
3816        i = 15;
3817        do {
3818                x76 &= 65535;
3819                x76 = (x76 << 11) + (x
» 76 >> 5);
3820                x76 -= (x10 & ~x54) +
» (x32 & x54) + key_schedule->xkey[4*i
» +3];
3821                x54 &= 65535;
3822                x54 = (x54 << 13) + (x
```

```
» 54 >> 3);
3823                x54 -= (x76 & ~x32) +
» (x10 & x32) + key_schedule->xkey[4*i
» +2];
3824                x32 &= 65535;
3825                x32 = (x32 << 14) + (x
» 32 >> 2);
3826                x32 -= (x54 & ~x10) +
» (x76 & x10) + key_schedule->xkey[4*i
» +1];
3827                x10 &= 65535;
3828                x10 = (x10 << 15) + (x
» 10 >> 1);
3829                x10 -= (x32 & ~x76) +
» (x54 & x76) + key_schedule->xkey[4*i
» +0];
3830                if (i == 5 || i == 11)
»  {
3831                        x76 -= key_sch
» edule->xkey[x54 & 63];
3832                        x54 -= key_sch
» edule->xkey[x32 & 63];
3833                        x32 -= key_sch
» edule->xkey[x10 & 63];
3834                        x10 -= key_sch
» edule->xkey[x76 & 63];
3835                }
3836        } while (i--);
3837        plain[0] = (unsigned char)x10;
»
3838        plain[1] = (unsigned char)(x10
»  >> 8);
3839        plain[2] = (unsigned char)x32;
»
3840        plain[3] = (unsigned char)(x32
»  >> 8);
3841        plain[4] = (unsigned char)x54;
»
3842        plain[5] = (unsigned char)(x54
»  >> 8);
3843        plain[6] = (unsigned char)x76;
»
3844        plain[7] = (unsigned char)(x76
»  >> 8);
3845        }
3846
3847
3848
3849 /*
3850  * Copyright (c) 2006 Apple Computer,
» Inc. All Rights Reserved.
3851  *
3852  * @APPLE_LICENSE_HEADER_START@
3853  *
3854  * This file contains Original Code an
» d/or Modifications of Original Code
3855  * as defined in and that are subject
» to the Apple Public Source License
```

```
3856  * Version 2.0 (the 'License'). You ma
      » y not use this file except in
3857  * compliance with the License. Please
      »  obtain a copy of the License at
3858  * http://www.opensource.apple.com/aps
      » l/ and read it before using this
3859  * file.
3860  *
3861  * The Original Code and all software
      » distributed under the License are
3862  * distributed on an 'AS IS' basis, WI
      » THOUT WARRANTY OF ANY KIND, EITHER
3863  * EXPRESS OR IMPLIED, AND APPLE HEREB
      » Y DISCLAIMS ALL SUCH WARRANTIES,
3864  * INCLUDING WITHOUT LIMITATION, ANY W
      » ARRANTIES OF MERCHANTABILITY,
3865  * FITNESS FOR A PARTICULAR PURPOSE, Q
      » UIET ENJOYMENT OR NON-INFRINGEMENT.
3866  * Please see the License for the spec
      » ific language governing rights and
3867  * limitations under the License.
3868  *
3869  * @APPLE_LICENSE_HEADER_END@
3870  */
3871
3872  int rc2_cc_set_key(
3873      RC2_Schedule *cx,
3874      const void *rawKey,
3875      size_t keyLength)
3876  {
3877      rc2_keyschedule(cx, rawKey, keyLen
      » gth, keyLength*8);
3878      return 0;
3879  }
3880
3881  void rc2_cc_encrypt(RC2_Schedule *cx,
      » const void *blockIn, void *blockOut)
3882  {
3883      rc2_encrypt(cx, (const unsigned ch
      » ar *)blockIn, (unsigned char *)block
      » Out);
3884  }
3885
3886  void rc2_cc_decrypt(RC2_Schedule *cx,
      » const void *blockIn, void *blockOut)
3887  {
3888      rc2_decrypt(cx, (unsigned char *)b
      » lockOut, (const unsigned char *)bloc
      » kIn);
3889  }
3890
```