$PSDocId: Document Release 1.0 for Runtime Library Release 3.6$

PS-X Library Samples Directory October 16 1996

Please read the individual readme_e.txt files where applicable in each sample directory for additional details for the specific sample. Some of these files are still being translated and will be released in the near future. Please check the BBS for new updates.

This document contains the following:

- a tutorial on how to run the samples
- a listing of all sample programs available in the directory "psx\sample\"
- a brief synopsis of all of the sample programs.

If you are new to PlayStation programming, first run the brief tutorials below. Afterwards, you should build the samples in the "tuto" (short for "tutorial") directories, such as \psx\sample\graphics\tuto and \psx\sample\sound\tuto.

If you are an experienced PlayStation programmer, note that some new samples have been created:

psx\sample\dongle
psx\sample\graphics\clutfog\tuto3.c
psx\sample\mipmap
psx\sample\rotate\axesmime

Two new directories of samples have been added.

- psx\sample\scee (fomerly released under the "Dsupport" directory in CDROM 1.7) contains samples contributed to and by Sony Computer Entertainment Europe.

- psx\sample\scea samples from Sony Computer Entertainment America. Naturally, in the future, we hope to enlarge this.

Example: How to run a sample program
---
1. Your boards should already have been installed.

-DTL-H2000 users should read "\docs\technote\2000.doc" in the Technical Reference CD-ROM.

-DTL-H2500 users should read "\docs\technote\2500.doc" in the Technical Reference CD-ROM. Other documents that should be read in conjunction with the installation are

\docs\technote\decicons.doc
\docs\technote\flashbat.doc
\docs\technote\h25bios.doc

2. Make sure the appropriate drivers are running on your system.

- DTL-H2000 users should be running the "dexbios" in their

autoexec.bat. For more information, refer to the document \docs\technote\2000.doc in the Technical Reference CD-ROM.

- DTL-H2500 users should run the program "h25bios.com" at an MS-DOS command-line prompt. Drivers such as "h25drv.exe" and "dexbios.com" should NOT be running. At an MS-DOS command line prompt, type

```
h25bios.com
```

This loads the TSR into memory. If you are running from Windows 95, you should run h25bios.com in EACH MS-DOS window in which you will be compiling and running your programs.

For more information on h25bios.com, refer to the documents \docs\technote\2500.doc and \docs\technote\h25bios.doc in the Technical Reference CD-ROM.

2. Make sure the following path environment variables are valid:

```
PSYQ_PATH
COMPILER_PATH
PSX_PATH
C_INCLUDE_PATH
C_PLUS_INCLUDE_PATH
LIBRARY_PATH
GO32
GO32TMP
TMPDIR
```

If not, then you will have trouble compiling and running programs. Refer to the documents refer to the documents \docs\technote\2500.doc or \docs\technote\2000.doc.

3. Make sure your paths are pointing to the directories "psx\bin" and "psyq". If you don't have enough environment space, edit your config.sys

```
shell=command.com /E:1024 /p
```

to allocate more memory for the environment. The "/E:1024" sets the environment size to 1024 (valid ranges are from 160 to 32768), and "/p" makes this command.com the default command prompt. (See page 342 of "Peter Norton's Complete Guide to DOS 6.22 6th edition for further details. Or consult your favorite DOS manual.)

3. Make sure the Psy-Q security dongle is in place on PC's parallel printer port. WARNING: Do not connect any peripherals to the back of the security dongle. Although it was meant to be a pass through device, the dongle may be damaged when connect to certain devices, such as an external parallel-interface SCSI hard disk.

4. Type the following:

```
cd <parent-directory>\psx\sample\graphics\clutfog
psymake
```

The GNU-C compiler will produce (among other things) "tuto0.cpe", which is a PlayStation binary file.

5. Reset the board by typing

```
resetps 1
```

6. If (and only if) you are running the DTL-H2000 board, type the following:

```
run <parent-dir>\psyq\snpatch.cpe
```

   The file "snapatch.cpe" fixes a bug in the ROM of the DTL-H2000 board. DTL-H2500 users must not run this patch.

7. Load the "tim" files into the main memory of the development board by typing

```
psymake load
```

   "Tim" files are written in a format usable by the GPU library functions. Examine the file "makefile.mak". The "load" directive invokes the function "pqbload", which loads the specified binary data into the memory address (in hexadecimal notation) of the PlayStation's main memory. Routines within the code will transfer the data from the main memory to the video RAM.

6. Run the program. Type

```
run tuto0.cpe
```

   This loads the "tuto0.cpe" file into the memory of the PlayStation developer board and executes the program.

How to run the CD samples

In order to run the cdrom samples in "psx\sample\cd" and in "psx\sample\sound",

1. Put the Programmer Tools CD-ROM into the DTL-H2010 or the DTL-H2510 CD-ROM drives.

2. Change to the directory, "psx\sample\cd\movie". Type

```
cd <parent-directory>\psx\sample\cd\movie
```

3. Type

```
psymake
```

   to make all of the tutorials. After all the samples have finished compiling, you can do the next step.

4. Type

```
resetps 1
```

   to reset the board.

5. If (and only if) you are running the DTL-H2000 board, type the following:

```
run <parent-dir>\psyq\snpatch.cpe
```

   The file "snapatch.cpe" fixes a bug in the ROM of the DTL-H2000 board. DTL-H2500 users must not run this patch.

6. Type

```
run <parent-directory>\psyq\selcd
```

   This instructs the development board to use the CD-ROM drive (rather than the CD-ROM Emulator) during routines that use the CD-ROM (such as CdInit()).

7. Type

```
resetps  1
```

   to reset the board (it is now ready to run CD commands).

8. If (and only if) you are running the DTL-H2000 board, type the following:

```
run <parent-dir>\psyq\snpatch.cpe
```

   The file "snapatch.cpe" fixes a bug in the ROM of the DTL-H2000 board. DTL-H2500 users must not run this patch.

9. Type

```
run tuto0.cpe
```

You should see a vivid movie of a spaceship flying around the screen. If you look inside the file "tuto0.c", on line 32, is the file name "\\data\\mov.str;1". If you load your Programmer Tools CDROM into the PC's CDROM drive, you will indeed find the file "\data\mov.str".

Please ignore all comments in the readme_e.txt that specify that additional information can be found in a corresponding directory "doc/jp/..." This information has been incorporated into the respective Developer Reference Series.

Also additional information can be found in the "Change" document for the libraries.

Samples Directory (The samples with "(*)" are added this time.)

```
psx\kanji\
 |--asc2sjis
 |--fontdata
 |--kanjidiv
 |--kanjifnt
 +--sjiscode

psx\sample\
 |--cd
 |   |--earth
 |   |--movie
 |   |--str3d
 |   +--tuto
 |--cmplr
 |   +--scratch
 |--etc
 |   |--card
```

```
|   |   |--doc
|   |   |--lib
|   |   |--makecard
|   |   +--max
|   |--cman
|   |--comb
|   |--mouse
|   +--thread
|--graphics
|   |--2d
|   |--balls
|   |--bg
|   |   |--bgsample
|   |   +--fix32
|   |--clutfog (* tuto3)
|   |--data
|   |--diffuse
|   |--divide
|   |   |--active
|   |   +--clip
|   |--dmpsx
|   |--fballs
|   |--gsgpu
|   |--jimen
|   |--mat
|   |--mesh
|   |   |--qmesh
|   |   |--rmesh
|   |   +--smesh
|   |--mipmap (*)
|   |--misc
|   |   +--60frame
|   |--oden
|   |--phong
|   |--pmd
|   |--ppm
|   |--rgb24
|   |--rotate
|   |   |--arot
|   |   |--intrpol
|   |   |--mat2rot
|   |   +--axesmime (*)
|   |--shadow
|   |--tmd
|   |--tmdview
|   |   |--data
|   |   |--lowlevel
|   |   |--mime
|   |   |--rcube
|   |   |--shuttle
|   |   |--tmdview3
|   |   |--tmdview4
|   |   +--tmdview5
|   |--tod
|   |   |--janken
|   |   |   |--tmd
|   |   |   +--tod
|   |   +--todview
|   |--trr
|   |--tuto
|   |--walk
```

```
|   +--zimen
|--math
|   +--tree
|--module
|   |--cdexec
|   |--execmenu
|   |   |--anim
|   |   |--balls
|   |   +--rcube
|   +--overmenu
|--press
|   +--tuto
|       +--msiro
|--scea
|   +--cntrl(*)
|--scee (*)
    +---CD
    |   +---CDPLAYER
    |   \---XAPLAYER
    +---DEMO
    |   +---SOUND
    |   +---TIM
    |   +---ALERT
    |   \---SAVEDATA
    +---DONGLE*
    +---DEMODISC
    |   +---DEMO
    |   |   +---BS
    |   |   +---EXAMPL
    |   |   +---NONE2
    |   |   +---SETSP
    |   |   \---DOCS
    |   \---BSCODE
    |       \---DD
    +---ETC
    |   +---CARD
    |   +---CARDCONF
    |   \---MTAP
    +---KCHEATS
    |   +---SOUND
    |   +---TIM
    |   +---ALERT
    |   \---SAVEDATA
    +---PAL
    +---SUBDIV
    |   +---IFF
    |   \---DATA
    \---UTILS
        +---EXCEPT
        \---PROFILER
+--sound
    |--balls
    |--basic
    |--cdvol
    |--mutual
    |--simple (* jump)
    |--stream
    |--tuto
    +--xse

psx\utility\
```

```
 |--cdexec
 |--menu
 +--screen

------------------------------------------------------------------------------

Samples Index(alphabetical order)

[kanji] Font Data

.\asc2sjis: the ASCII code to the Shift-JIS code
        Converts the ASCII code to the Shift-JIS code

.\fontdata: Font data
        Size and type
                11/13/15 dots.
                non-kanji/first level/second level/vertical/half-size
                characters.
        Code conversion table

.\kanjidiv: Command to extract character data
        Extracting image data from font data in character units.
        Command and viewer for extracted image data.

.\kanjifnt: Use of font data by size
        Sample to use font data by size

.\sjiscode: KANJI Code Viewer Program
        Shift-JIS codes of the built-in fonts can be dispalyed.

[sample] Sample Program

.\cd\earth: Earth
        Example to map a moving picture on a curved surface

.\cd\movie: movie <libcd>
         A group of samples of moving pictures with streaming
        tuto0: simple streaming program
        tuto1: free resolutions
        tuto2: on memory streaming
        tuto3: avoiding frame skip

.\cd\str3d: combination
         Sample of the combination of moving pictures with streaming
        and 3D model display
         Spreading loads of animation with DecDCTvlcSize()

.\cd\tuto: CD tutorial <libcd>
        CD-ROM step-by-step tutorial
        tuto0:  simplest CD-Player (polling type)
        tuto1:  simplest CD-Player (interrupt type)
        tuto2:  auto repeat play among 2 points of CD-DA
        tuto3:  auto repeat play using CdlDataEnd
        tuto4:  fast operation using CdControlF
        tuto5:  auto repeat play among 2 point of CD-XA.
        tuto6:  interleaved audio/data channel
        tuto7:  background CD read
        tuto8:  multi file CdRead
        tuto9:  load and execute programs
        tuto10: high level CD-ROM file access
        tuto11: test CD type
```

```
.\cmplr\scratch: Scratch pad area used Sample
         Data is placed on the scratch pad area, and the difference of the
         processing speed can be seen with 3 access methods.

.\dongle: Illustrates how to use the memory card as a dongle.


.\etc\card: Memory card boot sample
        makecard: Program to create an icon for the memory card
                Creating a file with a 3-hand-gesture-image icon.
        max: Memory card operation screen
                Disclosing source codes of the OSD operation
                screen.
.\etc\cman: Memory card access sample
        Memory card file utility such as state-monitoring, formatting,
        creating.

.\etc\comb: Link cable sample
         Sample for operating each ball on the 2 machines connected to
         each other by a link cable

.\etc\mouse: Mouse control sample
        Sample to process cursor-movements and clicking with a mouse

.\etc\thread: Thread sample
        Sample to do other job until next VSync interruption

.\graphics\2d: 2D <libgpu>
        Texture mapping on a curved surface.
        Mapping a 512x256 texture pattern on a 3-dimensional curved
        surface.
        When 'select' is pressed, the pattern will come to pieces.

.\graphics\balls: Balls <libgpu>
        Displaying a lot of 16x16 sprites.
        The ways of using the following functions are described here.
            FntPrint KanjiFntPrint, VSyncCallback(), VSync()

.\graphics\bg: BG drawing function sample in libgs <libgs>
        TIM\CEL\BGD files made up with Sprite Editor may be read in
        and displayed.
        bgsample: Back ground sample
        fix32:    Back ground sample (fast)

.\graphics\clutfog: Fog sample with clut
        tuto0: A clut is generated frame by frame, and transferred to
              the vram.
        tuto1: Some cluts are placed on the vram, and switched according
              to the depth of fog.
        tuto2: Cluts are switched by DR_MOVE.
              Applicable to drawing by libgs as well.
        tuto3:textured polygon with depth queue (by enchanging CLUT
                with DR_LOAD primitive)" (*)

.\graphics\data: shared data

.\graphics\diffuse: Diffusion <libgte>
        16x16 balls or rectangle polygons are diffused from the
        origin in the world coordinate system.
        GTE performance demonstration.
```

```
.\graphics\divide: Polygon automatic division sample <libgte>
        clip: Divide function examples to avoid texture distortion.
               In readme.txt, PCpoly function examples included.
        active: Sub-division sample with the direct mapping.
                Crack problem and z-sorting by the maximum value not
                by the center of gravity are included.

.\graphics\dmpsx: DMPSX <libgte>
        High speed sample with dmpsx

.\graphics\fballs: Fast Balls <libgpu>
        Sample based on "balls" program for decreasing the drawing time

.\graphics\gsgpu: GSGPU <libgpu,libgs>
        Sample using libgs and libgte together.
        tuto0: Uses AddPrim() in libgs.
        tuto1: Draws libgs objects with DrawTag().

.\graphics\jimen: Undistorted texture mapping <libgte>
        Function sample for undistorted texture mapping.

.\graphics\mat: Matchang
        Sprite animation
        The Matchang animation is placed in the 3 dimension.

.\graphics\mesh: Mesh
        qmesh: two dimensional mesh
            tuto0: sample of QMESH function...screen clip
            tuto1: sample of QMESH function...terrain data
        rmesh: round mesh
        smesh: strip mesh
            tuto0: drawing performance of SMESH functions
            tuto1: browsing SMESH functions' drawing mode
            tuto2: icosahedron

.\graphics\mipmap:      mipmap sample  (*)

.\graphics\misc: graphics miscellaneous
        60frame: difference 60 frames from 30 frames

.\graphics\oden: Oden <libgs>
        Moving 3 light sources interactively, changing their colors,
        performing the real-time lighting calculation.

.\graphics\phong: phong shading <libgte>

.\graphics\pmd: PMD <libgte>
        pmdview (tmdview PMD version)
        tuto0   PMD file analysis and display
        tuto1   TMD file analysis and display
        tuto2   PMD display in the hierarchical coordinate system
        tuto3   Light-source-calculated PMD
        tuto5   DMPSX version of PMD function

.\graphics\ppm: undistorted mapping
        (perfect perspective mapping) <libgte>

.\graphics\rgb24: RGB24 <libgpu>
        Demo in the 24-bit mode.
        Examples with StoreImage(), LoadImage(), MoveImage() are here.
```

```
.\graphics\rotate: rotation <libgte>
        arot:    Rotation angle interpolation program
        intrpol: various kinds of interpolating about rotation
        mat2rot: Get Euler's angles from rotation matrix
        axesmime: Joint MIMe animation of articulated model(*)

.\graphics\shadow: Shadow <libgte>
         The shadow dropped from a triangle is calculated in this program.
        Since the clipping is performed accurately, the shadow can be
        dropped on the steps

.\graphics\tmd: handle TMD format using lowlevel functions
        tuto0: display simple POLY_F3 TMD data
        tuto1: display on hierachy coordinate system.
               display many TMDs located in each local coordinate.

.\graphics\tmdview: TMD view
        A group of samples with TMD and PMD data
        data:     Directory where the data used in the samples is included.
        lowlevel: Using lowlevel functions
            Lowlevel sample with GsTMDfast...() functions
            tuto0: 3 sided polygon, flat
            tuto1: 4 sided polygon, gouraud
            tuto2: eliminate a gap between polygons (with dmpsx)
            tuto3: mipmap version (with dmpsx)
        mime: MIMe Interactive Animation
            MIMe sample program with GsDOBJ5
            Controlling 4 MIMe parameters with L1,L2,R1,R2 buttons.
            The data is a simple gauraud-shaded polygon, and MIMe
            processing is performed for the normal.
        rcube: rotating cubes
            Variable effect samples for 3D
        shuttle: shuttle
            Hierarchical coordinate system sample with a space-
            shuttle model.
            The Animation such as opening/closing the hatch
            is displayed by setting the hatch and a satellite in
            the shuttle in its child coordinate.
        tmdview3: The simplest PMD data display program with GsDOBJ3
            tuto0: Simplest TMD data display program with GsDOBJ3
        tmdview4: The simplest TMD data display program with GsDOBJ2
            tuto0: simple tmdviewer using GsDOBJ2(GsSortObject4())
            tuto1: using GsSortObject4J()
            tuto2: active sub divide sample
            tuto3: sample code for split screen using GsDOBJ5
            tuto4: sample code for multi ot and using same object with
                   different hadlers.
            tuto5: multi screen coordinate sample
            tuto6: sample code of subjective move.
            tuto7: using GsSortObject4J() and using material attenuation
                   in GsDOBJ2
        tmdview5: TMD data display program with GsDOBJ5
            tuto0: Simplest TMD data display program with GsDOBJ5
            tuto1: Sample of split screen
            tuto2: With modeling data some objects are displayed.
                   More than one OT are used.
            tuto3: Automatic division with GsDOBJ5 attribute
            tuto4: Multi-screen coordinate system
            tuto5: Sample rewritten with GsSortObject5J
            tuto6: Sample where the viewpoint is moved subjectively (*)
```

```
.\graphics\tod: Animation with tod
        janken: Multiple interactive tod animation
        todview: Simple animation

.\graphics\trr: TransRot...functions sample <libgte>
         Sample of TransRot...() functions to eliminate a gap between
        polygons

.\graphics\tuto: Tutorial <libgpu>
        Step-by-step tutorial
        tuto0   Displaying sprites
        tuto1   Drawing test with OT
        tuto2   Drawing a rotating polygon with GTE
        tuto3   Drawing a rotating cube
        tuto4   Drawing a cube with the light source
        tuto5   Drawing multiple 3D objects
        tuto6   Testing a 1D scrolling BG
        tuto7   Drawing a cube with the depth cueing
        tuto8   Showing a cell-type BG
        tuto9   Showing a 3D-cell-type BG
        tuto10  3D cell type BG (bird view)
        tuto11  pseudo mosaic effect
        tuto12  pseudo line scroll effect
        tuto13  multi window operation

.\graphics\walk: An object walks on a polygon <libgte>
        Constraining an object on a polygon.
        On a object (object1) created by polygons, another object
        (object2) may move around. The object1 may take any shapes.
        The object2 changes its direction according to the direction of
        the object1's normal.

.\graphics\zimen: Terrain
         A group of programs to display the endless plane
        tuto0: Active primitive subdivision (with dmpsx)
        tuto1: Basic viewing volume clipping
        tuto2: Meshed ground pattern without height
        tuto3: Meshed infinite ground pattern
        tuto4: Meshed ground with active subdivision
        tuto5: Terrain sample with CLUT FOG (version with libgs)

.\math\tree: Math-libaray-used sample
        Drawing a tree curve by the trigonometric function.

.\module\execmenu: EXEC sample
        BALLS, RCUBE, and ANIM are activated from the menu.
        There are 2 examples in this sample.
           * Activated by LoadExec().
           * Activated by Exec() after reading to the available
             memory.

.\module\overmenu: Overlay sample
        BALLS, RCUBE, and ANIM are activated from the menu.
        3 BIN files are executed in order with the sound on.

.\press\tuto: Tutorial <libpress>
        MDEC step-by-step tutorial
        tuto1: simple VLC decode and MDEC on memory decompression
        tuto2: paralell execution of LoadImage() and DecDCTout()
        tuto3: simple on-memory movie operation
```

```
        tuto4: handshake using callback
        tuto5: parallel execution of LoadImage() and DecDCTout()
               using callback.
        tuto6: complete background on-memory movie decompression
        tuto7: fine tune-up for parameters

.\scea\cntrl (*) : Controller demonstration
        Demonstrates controller API.  Sprites are generated
        according to the controllers attached, such as a pad-sprite
        if a pad controller is being used, or a gun-sprite if
        a light-gun controller is in use.

.\scee (*) : Currently, the contents of this directory are unsupported,
        since they consist of contributions by other developers from
        Sony Computer Entertainment Europe.  You will not be
        able to run all of the examples, as noted in the descriptions
        below, due to a lack of data, or due to incompatibilites in the
        libraries.

.\scee\cd\cdplayer: Plays a music CD.
        When you put a music CD into your DTL-H2010 or DTL-H2510, you
        can play the tracks.

.\scee\cd\xaplayer: Shows how to use interleaved XA streams to
        store a large number of dialog lines and how to play
        them back nearly instantly. Unfortunately, we did not
        get the PACK1.XA and PACK2.XA files for Programmer Tools
        CD-ROM release 1.8. Please contact SCEE for more information on this.
        USA developers can contact SCEA, and we will forward your requests to SCEE.

.\scee\demodisc: Source code hook for usage in the Demo cd.
        Apparently, this is the demo cd used by the European developers.
        Please contact SCEE for more information on this. USA developers can
        contact SCEA, and we will forward your requests to SCEE.

.\scee\etc\card: Memory card access.
        Lists the contents of the memory card inserted
        in the controller box DTL-H2080. Please contact SCEE for more
        information on this. USA developers can contact SCEA, and we will
        forward your requests to SCEE.

.\scee\etc\cardconf: Memory card access from SCEE developer's conference.
        You will have to change the ".lnk" file to suit
        your development environment. It allows you
        to read and write from memory cards using optimized techniques,
        while displaying animated graphics and playing seq data.

.\scee\etc\mtap: Multitap example from the European Developers conference, 1996.
        Shows you you how to read data from the multitap.  Please
        contact SCEE for more information on this. USA developers can
        contact SCEA, and we will forward your requests to SCEE.
        (Note that another multitap example, written by Mike Fulton of
        SCEA, is available in .\scea\cntrl)

.\scee\PAL: Example for PAL users.
        Please contact SCEE for more information on this.
        USA developers can contact SCEA, and we will forward your requests to SCEE.

.\scee\kcheats: Complete source for the "killer cheats" program.  You
        must burn a CD or load the CD emulator with the appropriate
        files.  Please contact SCEE for more information on this. USA developers can
```

contact SCEA, and we will forward your requests to SCEE.

.\scee\subdiv: Alternative polygon subdivision routines.
Contributed by Derek Leigh-Gilchrist. This does a fast texture mapping to flat and shaded triangles and quadrilaterals. Use the following control pad buttons:

```
SELECT   = change poly type (Flat triangle, Gourad triangle,
                        flat quadrilateral, Gourad triangle)
START    = pause rotation
LEFT     = reduce amount of subdivision (0=no subdivide,
                                         1= split into 4 polys,
                                         2=split into 16 polys)
RIGHT    = increase amount of subdivision
UP       = ZOOM OUT
DOWN     = ZOOM IN
SQUARE   = rotate Y(-)
CIRCLE   = rotate Y(+)
CROSS    = rotate X(-)
TRIANGLE = rotate X(+)
```

.\scee\utils\except: Example exception handler.
Contributed by Brian Marshall, this is an exception handler that is compatible for versions 3.1 and 3.2 of the library. It is pretty old. A tester at SCEA was unable to get beyond the "red screen" as of November 1996. Please contact SCEE for more information on this. USA developers can contact SCEA, and we will forward your requests to SCEE.

.\scee\utils\profil: Example "profiler"
You will have to edit the "protest.lnk" file to match your development environment.

.\sound\balls: combination of the sound and graphics
Example of combination of sound and graphics.
While pressing a button, balls bounce in the screen.
When they hit against the wall, different sounds for each ball are generated.
SEQ data is used as the background music.

.\sound\basic: Basic play 1
SEQ/SEP data processing function sample.
SEQ and SEP data may be played simultaneously.
SEP data consists of 3-connected SEQ data.

.\sound\cdvol: SPU decoded data reading sample <libspu>
Music played on the CD is read as the "SPU decoded data" from the SPU in the background, and the volume is displayed with a graph (with the display of the peak level).

.\sound\mutual: Wave form data divided transfer sample <libsnd>
At a timing divided wave form data is read into the main memory, and transferred to the sound buffer. This process is repeated until all parts are transferred. As a result, 2 pieces of music may be played.

.\sound\tuto: Basic usage of basic sound library <libspu>
tuto1: Pitch designation/key-on/key-off
According to the control-pad operations, a sound is

```
                played with variable pitches.
        tuto2: Mute
                Performing the sound generation, mute-on, mute-off
        tuto3: SPU interrupt
                Setting a interrupt in the middle of piano sound data.
                When the piano sound starts, and the interrupt
                occurs, a sine wave is generated.
        tuto4: Noise sound source
                Generating a sine wave and noise by changing a pitch.
        tuto5: Divided transfer of wave form data
                Alternating divided transfer and sound
                generation after the transfer in 2 voices.
        tuto6: Reverb
                Generating a piano sound and designating 9 kinds
                of reverbs for the sound.

.\sound\simple: Basic play 2
        Example using the SEQ data processing functions.
        Playing SEQ endlessly.
        On the screen the current tempo, volume, status (playing,
        pausing, etc.) are displayed.

        jump: example of jump table. (*)

.\sound\stream: SPU streaming sample program <libspu>
        tuto1: The background is "balls".
        tuto2: The background is "movie".

        Sample using the SPU streaming library included in the basic
        sound library.
        Performing playback and halt of 7-channel (14 voices) SPU
        streams by operating the control pad.
        Displaying the state of SPU streaming on the screen.

.\sound\xse: Auto-effect
        Example using the sound utility functions.
        Such effects as pitch-bend, auto-panning, auto-volume are
        produced to the         keyed-on sounds.
        By moving a thumb in the scroll bar, auto-panning, auto-volume,
        and pitch-bend may be available.

[utility]

.\cdexec: Start-up utility from CD-ROM/Emulator
        Used when activating from CD-ROM/Emulator on H2000 with patch
        executed.
        Alternative module of 'resetps 0'.

.\menu: Sample Program Viewer which loads execution file.

        Sample execution files are activated from the menu.
        It is necessary that the program which can be activated from this menu
        should link "none2.obj" and be written in "menu.lst".

.\screen: Screen
        Frame buffer viewer.
        Demo to explain the display mode and display environment
        parameters.


================================================================================
```

SONY COMPUTER ENTERTAINMENT AMERICA
919 East Hillsdale Blvd, 2nd Floor
Foster City CA 94404
415-655-8000
E-mail: DevTech_Support@interactive.sony.com
Developer Support BBS: 415-655-8119
Developer Support Hotline: 415-655-8181

SONY COMPUTER ENTERTAINMENT EUROPE
Waverley House
7-12 Noel Street
London W1V 4HH
E-mail: dev_support@interactive.sony.com
WWW: http://www.scee.sony.co.uk
FAX: +44 (0) 171 390 4324
Developer Support Hotline: +44 (0) 171 390 1680

==========