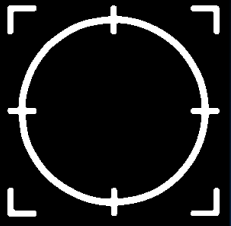


# CD Overview - Part One



Ben Fawcett  
Developer Support Engineer  
S.C.E.E.





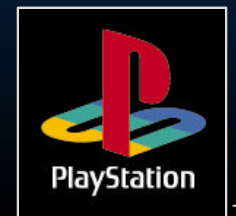
# Covered topics

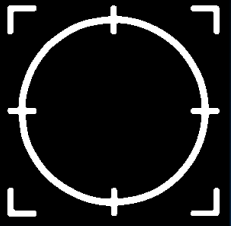
Functionality

Speed Considerations

CD Errors

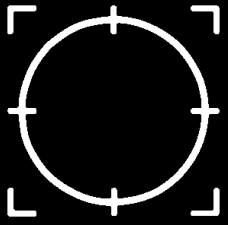
Multi-CD Games





# Functionality

- ▶ Play Music
  - ▶ XA-ADPCM
  - ▶ CD-DA
- ▶ Streaming
  - ▶ Movies - CD Overview : Part Two
  - ▶ Data / World



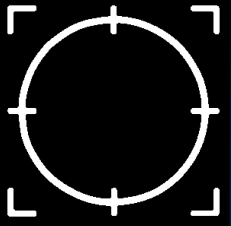
# Play Music

## ➤ ADPCM

- Compressed 8X to 32X
- Easy to do: See `\PSX\SAMPLE\CD\TUTO\TUTO5.C`
- Continued in Sound Seminar

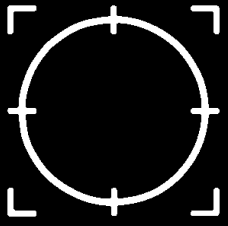
## ➤ DA / Red Book

- Lossless
- Easy to do: See `\PSX\SAMPLE\CD\TUTO\TUTO4.C`
- Continued in Sound Seminar



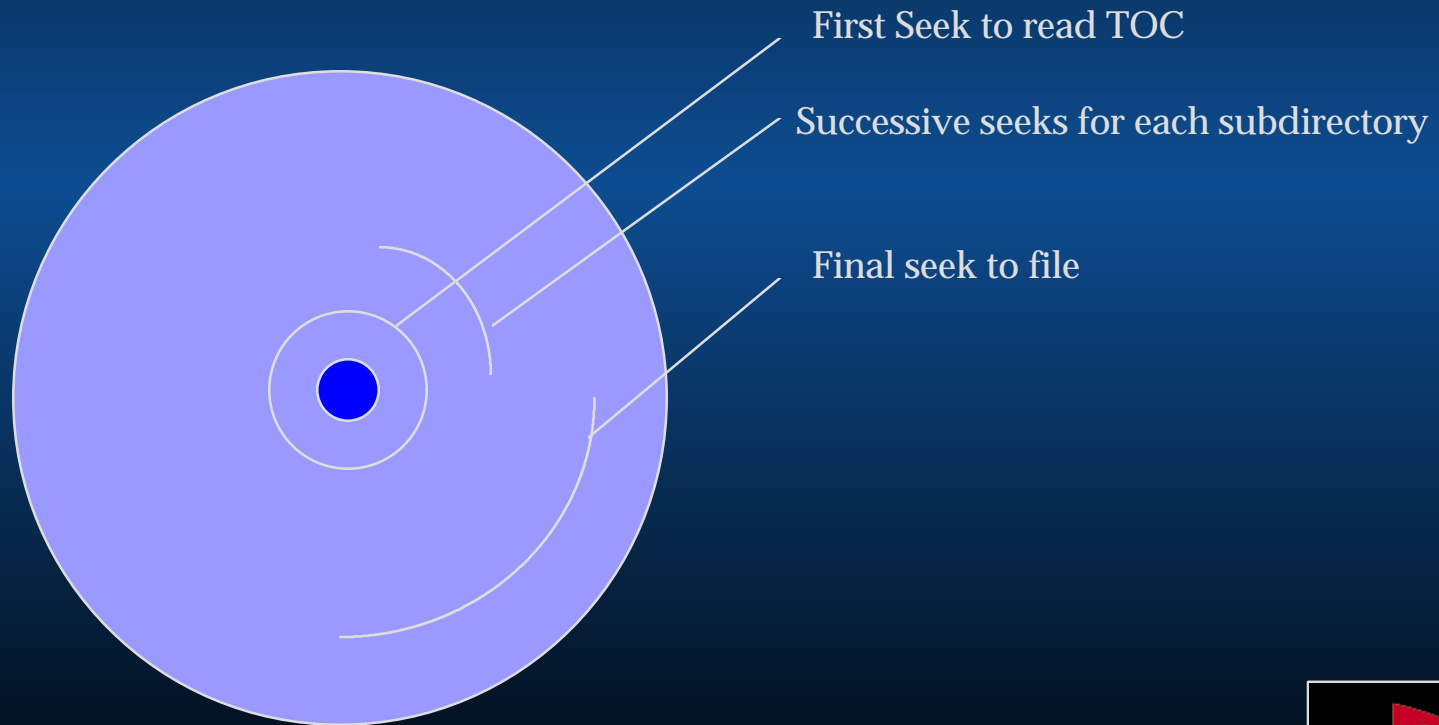
# Speed Issues

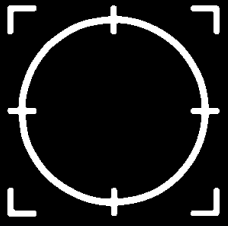
- Minimize seeks / Data organisation
- Read asynchronously
- Avoid speed changes



# CdSearchFile vs. Direct Seek

CdSearchFile() has a single directory buffer, which causes it to seek multiple times to go to any file.

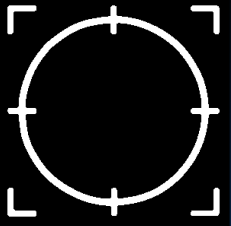




# Hard-code file locations - .CCS

## ➤ Four easy steps :

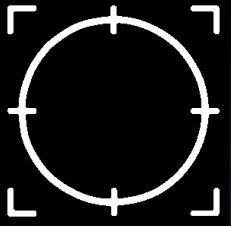
- ➊ Use CDGEN to do a layout (works without CDW-900E)
  - Put MAIN.EXE last in track 1 (so size can vary)
  - Save as .CCS file
- ➋ Write/use a utility to create POS.H with position info from CCS file
  - Absolute file location appears on the line following each sourcefile path
- ➌ Compile with POS.H
- ➍ Burn CD / Build emulation image



# Hard-code File Locations - .MAP

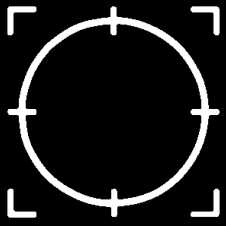
- ▶ Can also use Buildcd to generate .MAP file
- ▶ Includes more information : size, physical location, logical location
- ▶ Paths can be difficult to find without prior knowledge





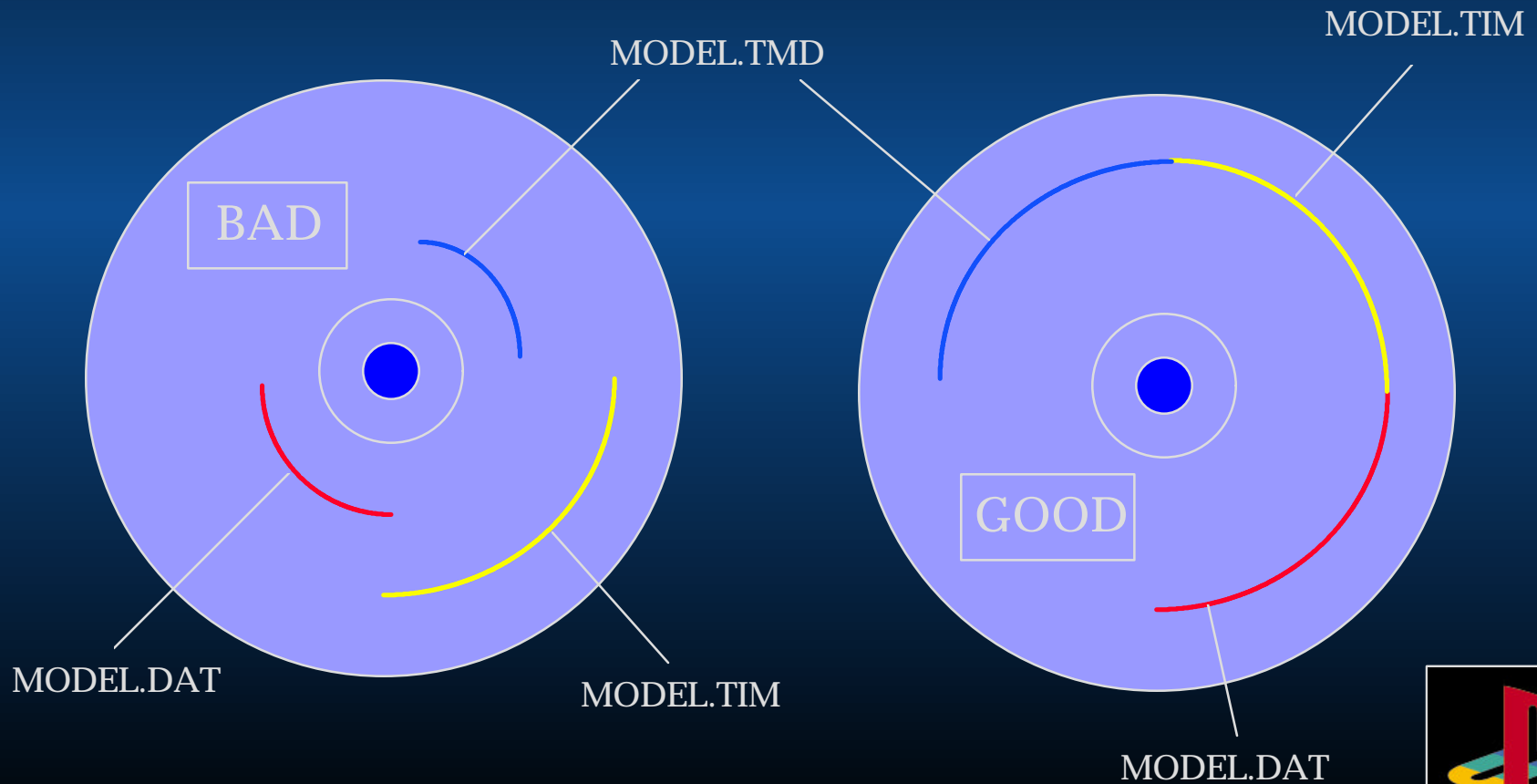
# Data Organisation

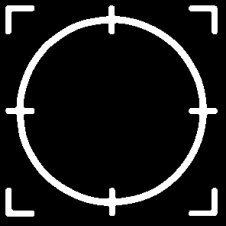
- +/- 100 sector seeks require rotation of CD read head, but not linear motion, so they are much faster than longer seeks
- Optimize data layout to take advantage of this
- Set up data in contiguous blocks



# Set data up in contiguous blocks

MODEL.TMD, MODEL.TIM, MODEL.DAT  
to be read at same time

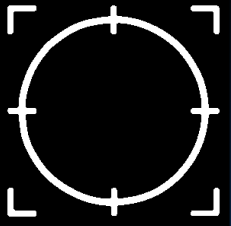




# Read asynchronously

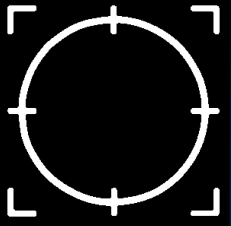
Main loop:

```
...  
ERROR:  
    while( CdControl(CdlSetloc, (u_char *)&fp.pos, 0) == 0 );  
    if( CdRead(nsector, sectbuf[], CdlModeSpeed) == 0 )  
        goto ERROR:  
  
    /* Since CdRead() runs in the background, the program can do  
    * another task in the foreground. The current reading status can  
    * be modified in CdReadSync().  
    */  
    /*  
    while ((cnt = CdReadSync(1, 0)) > 0)  
    {  
        :  
        :  
        /* Foreground routines */  
        :  
        :  
    }  
  
    if (cnt == -1) goto ERROR:
```



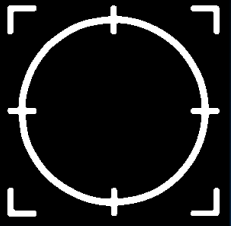
# Avoid speed changes

- ▶ Spin up and spin down takes a lot of time
  - ▶ Avoid using CD-DA with a lot of intermittent CD access
  - ▶ Use XA audio at double speed instead
- ▶ Do not use CdIStop, use CdIPause instead



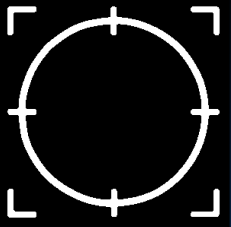
# CD Error issues

- Read errors
- Seek errors
  - Retry
  - Overshoot



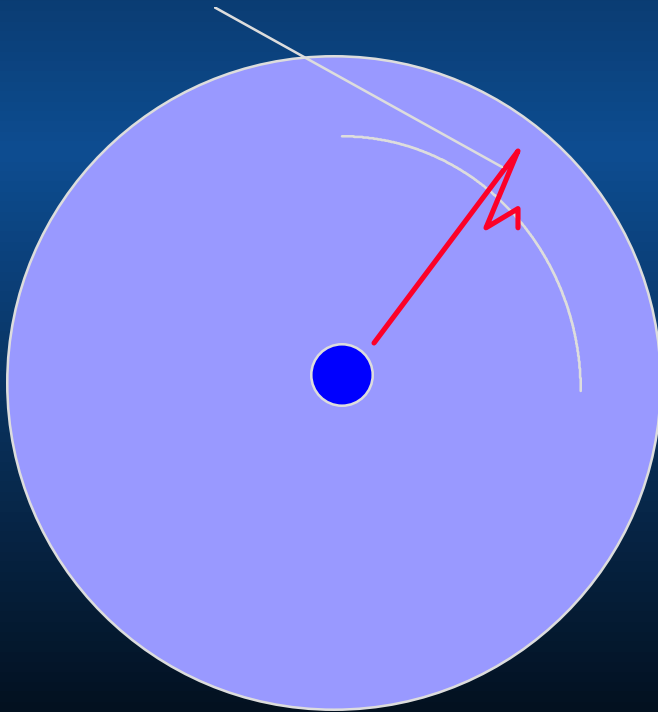
# Read Errors

- CdControl() only returns successful acceptance of CD command, not successful completion of CD command
- Bad Strategy:
  - if (CdControl(CdlReadN, pos, result) != 1) goto error;
- Good strategy
  - Set up a watchdog timer in VSyncCallback for retry
  - Example: CdRead waits 8 seconds

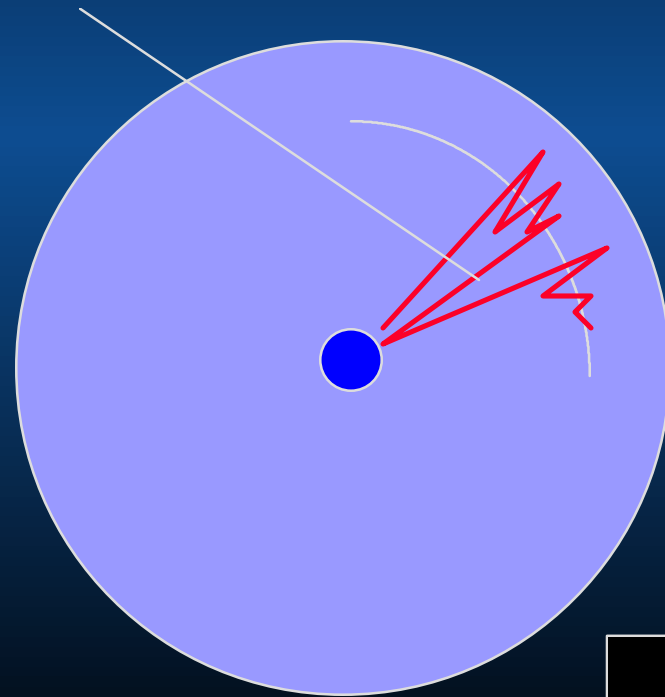


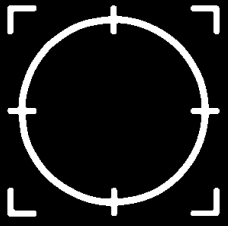
# Seek Errors

CD read head can overshoot a few times before success



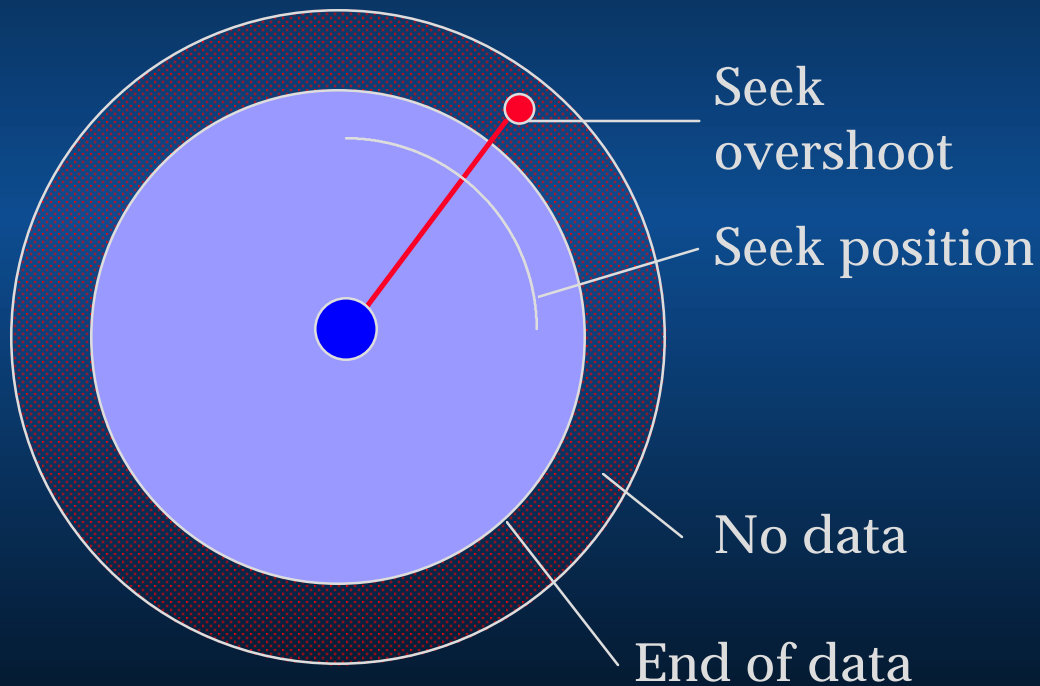
If the seek does not settle within a limited time, the head returns to the center and retries



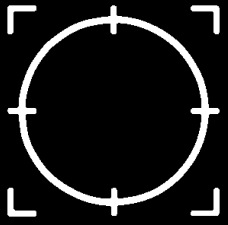


# Seek Errors (cont.)

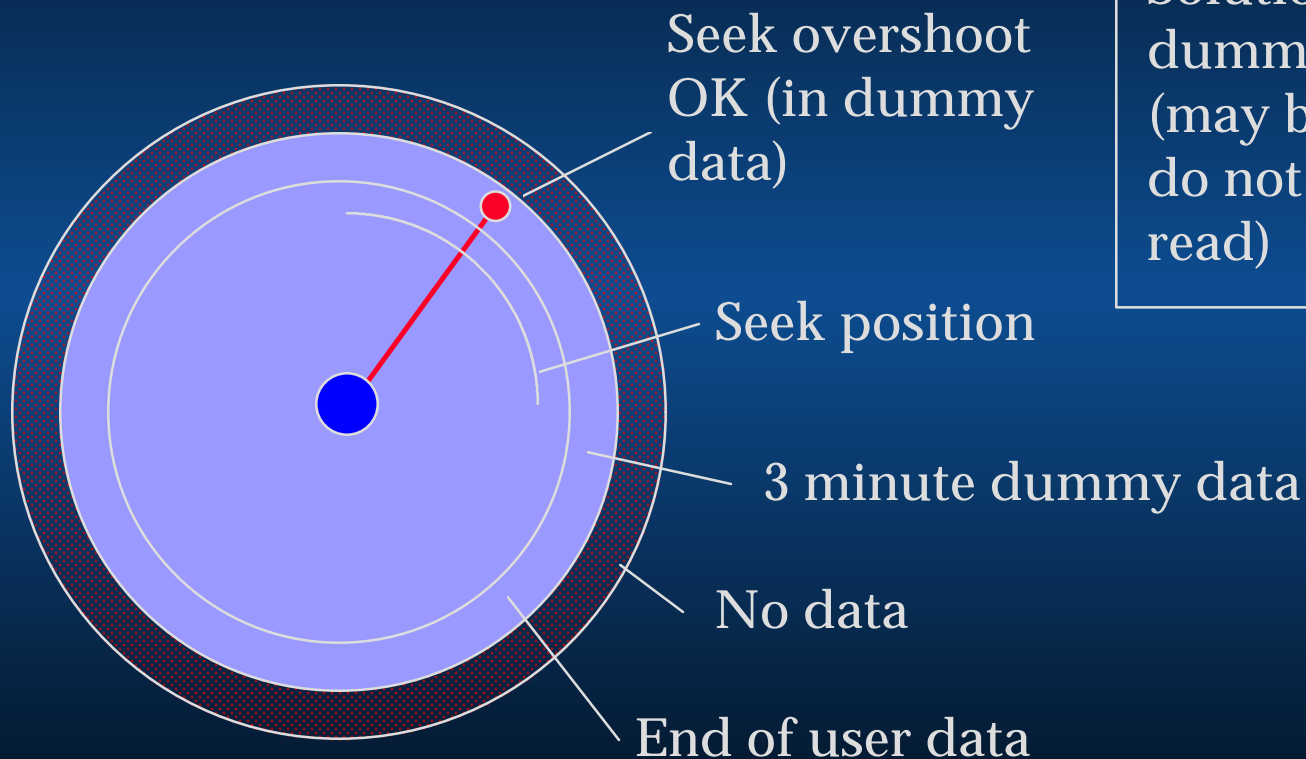
A seek past the end of data is fatal



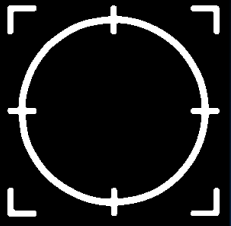




# Seek errors (cont.)

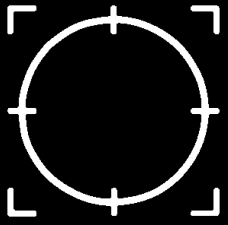


Solution: Put 3 minutes of dummy data at end of CD (may be real data that you do not seek into, but only read)



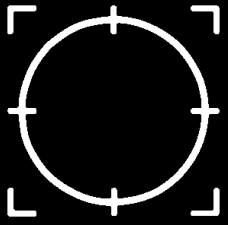
# Multi CD games

- When the CD cover is opened, the CD subsystem is put into an indeterminate state
- Two ways to tell when CD subsystem state is restored :
  - ❶ Polling shell open flag then timing
  - ❷ Seeking until a non-error is returned



# CD Status Flags

```
/* Status Contents */
#define CdlStatPlay          0x80    /* playing CD-DA */
#define CdlStatSeek         0x40    /* seeking */
#define CdlStatRead         0x20    /* reading data sectors */
#define CdlStatShellOpen    0x10    /* once shell open */
#define CdlStatSeekError    0x04    /* seek error detected */
#define CdlStatStandby      0x02    /* spindle motor rotating */
#define CdlStatError        0x01    /* command error detected */
```



# Polling Shell Open Flag

Psuedocode:

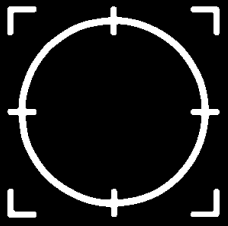
```
// Show "Put in Disk #2..." screen
```

```
while (!(status & CdIStatShellOpen)); // wait for lid to open
```

```
while (status & CdIStatShellOpen); // wait for lid to close
```

```
// Wait for a few seconds
```

```
// Continue ...
```

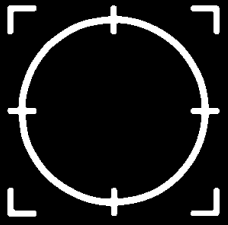


# Seeking until non-error returned

Use logical seek, because a physical seek will be successful for non-PlayStation CDs  
If DA CD is anticipated, use physical seek

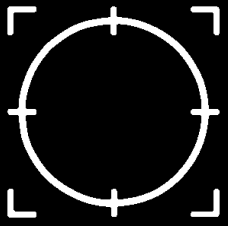
Pseudocode:

```
// Show "Put in Disk #2..." screen  
while (!(status & CdIStatShellOpen)); // wait for lid to open  
while (CdControlB(CdISeekL, pos, 0) == error); // wait for success  
// Continue ...
```



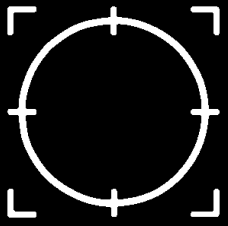
# New functionality

- New libraries will include functions which aid in the detection of a second CD
- These include :
  - CdDiskReady()
  - CdGetDiskType()



# Summary

- Reduce seeks / loading times
  - Optimise your data layout
  - Use hard-coded file locations
- CD errors will occur
  - Handle them appropriately
- Tomorrow - CD Overview Part Two



The End

