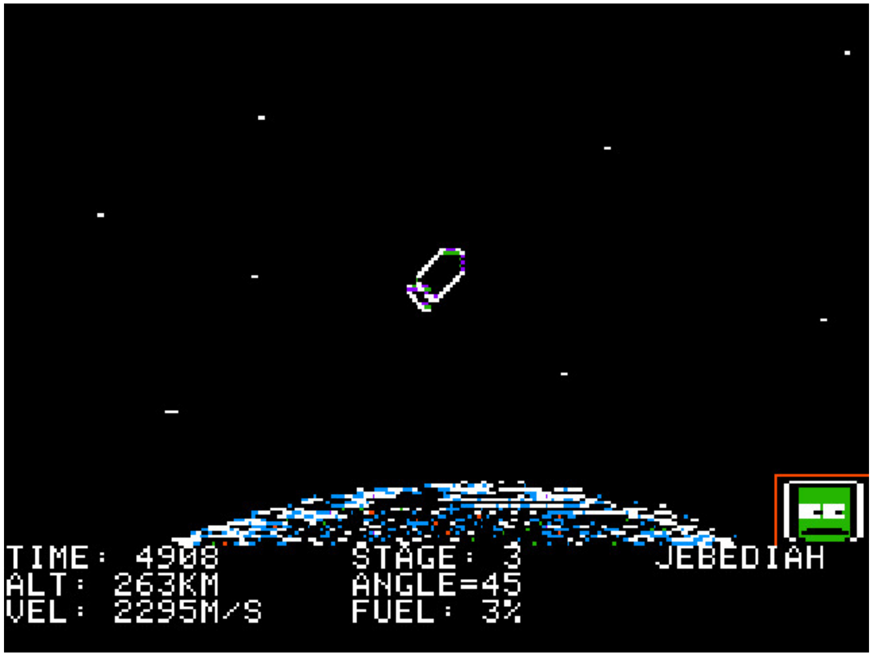
# Fast Loading Times on the Apple II: Pushing the Limits of the Disk ][

**Vince "DEATER" Weaver**

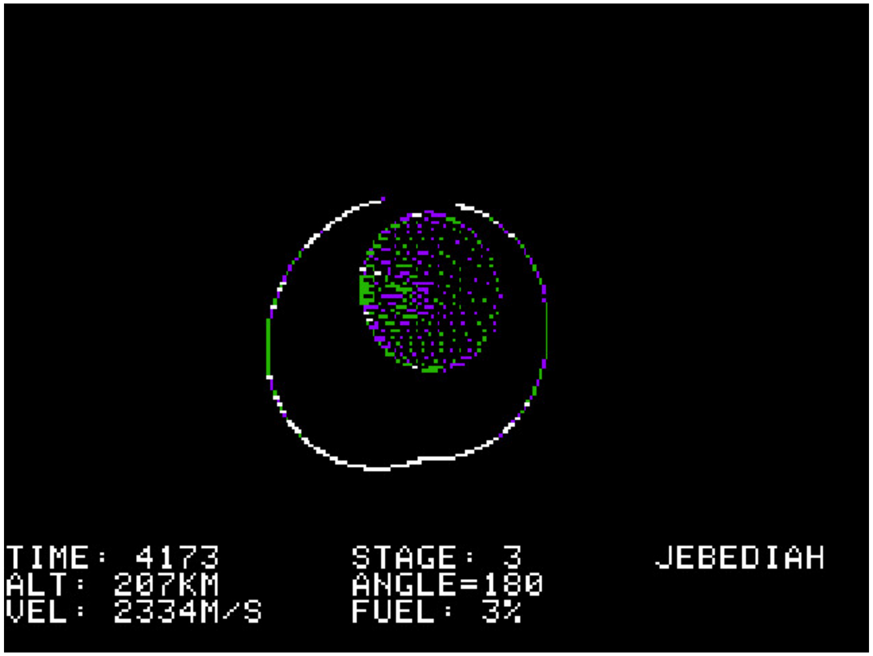
vince@deater.net

Demosplash 2020 — 20 November 2020

# My Demakes – Kerbal Space Program

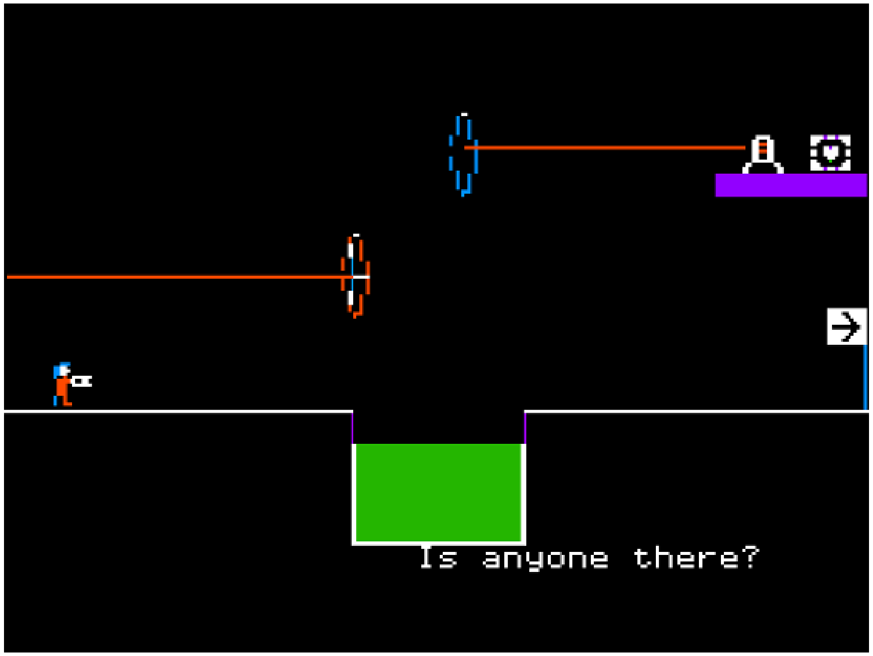


```
TIME: 4908        STAGE: 3        JEBEDIAH
ALT: 263KM        ANGLE=45
VEL: 2295M/S      FUEL: 3%
```

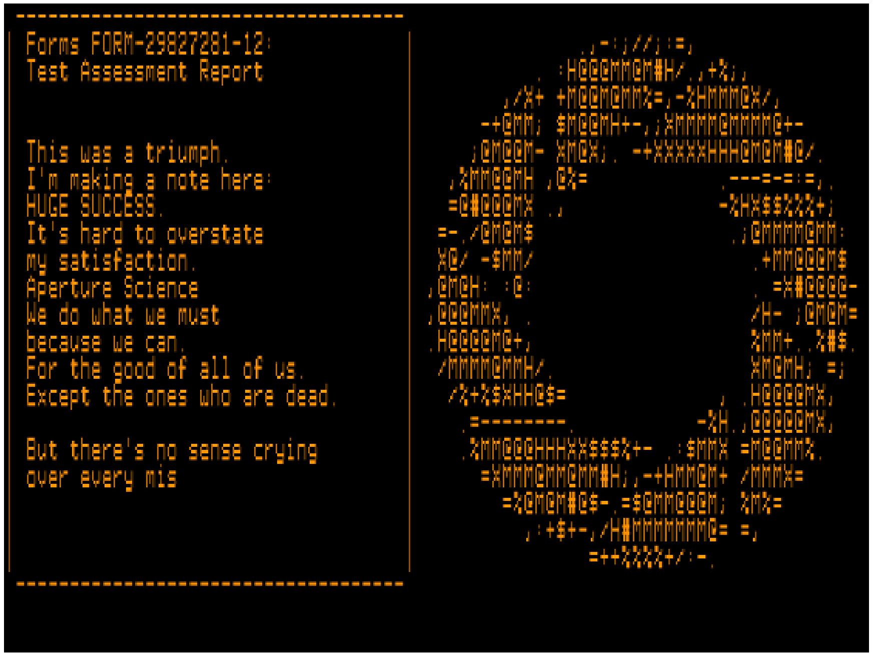


```
TIME: 4173        STAGE: 3        JEBEDIAH
ALT: 207KM        ANGLE=180
VEL: 2334M/S      FUEL: 3%
```

# My Demakes – Portal

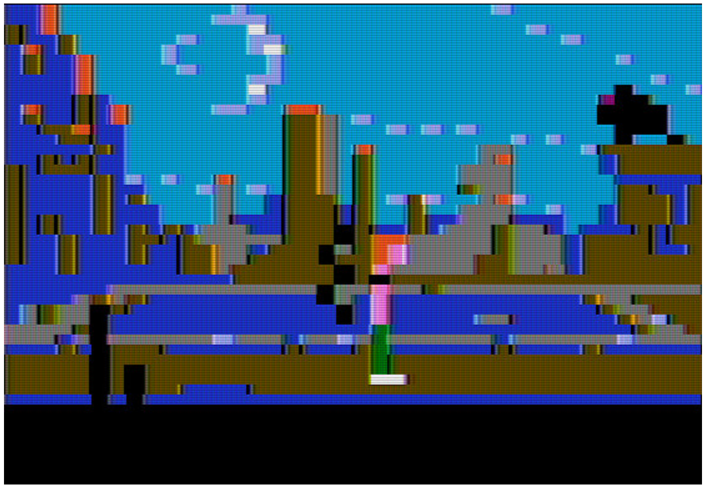


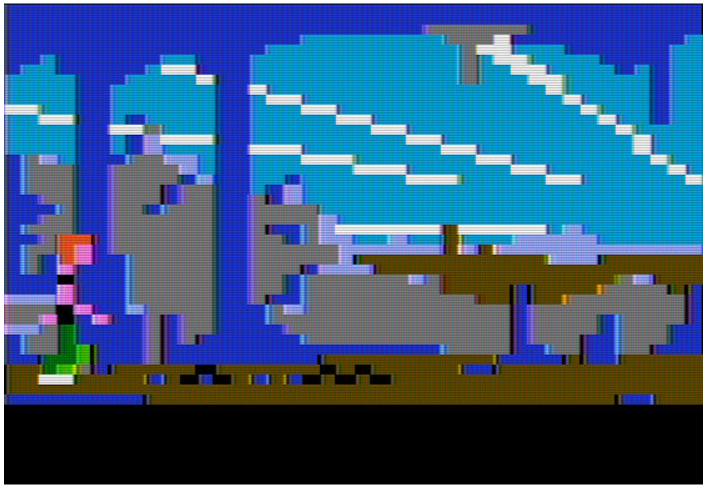
Is anyone there?

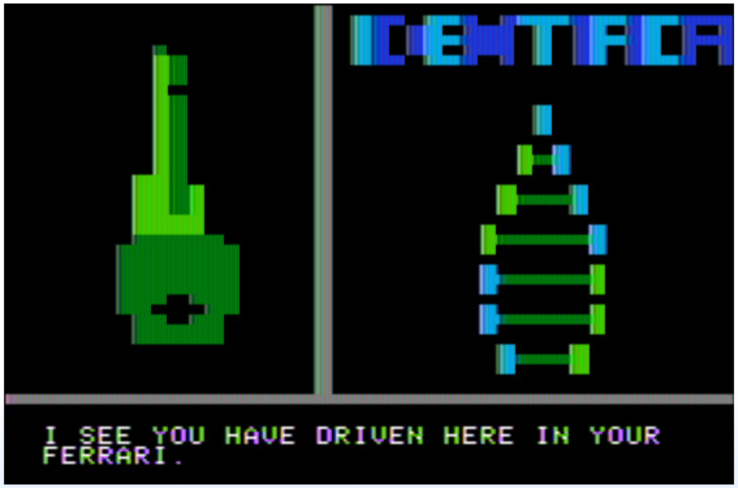


```
Forms FORM-29827281-12:
Test Assessment Report


This was a triumph.
I'm making a note here:
HUGE SUCCESS.
It's hard to overstate
my satisfaction.
Aperture Science
We do what we must
because we can.
For the good of all of us.
Except the ones who are dead.

But there's no sense crying
over every mis
```

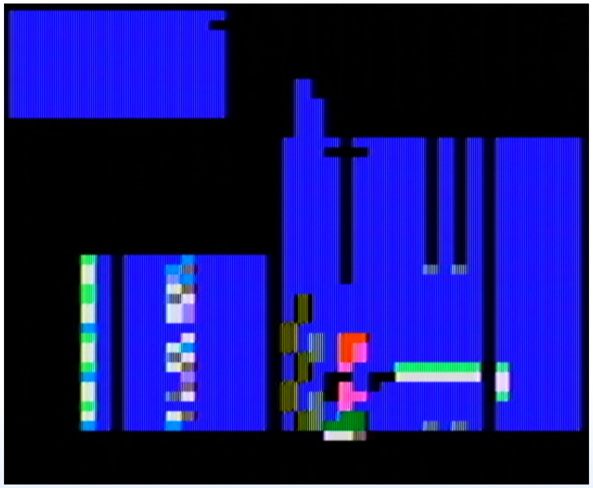
# My Demakes – Another World



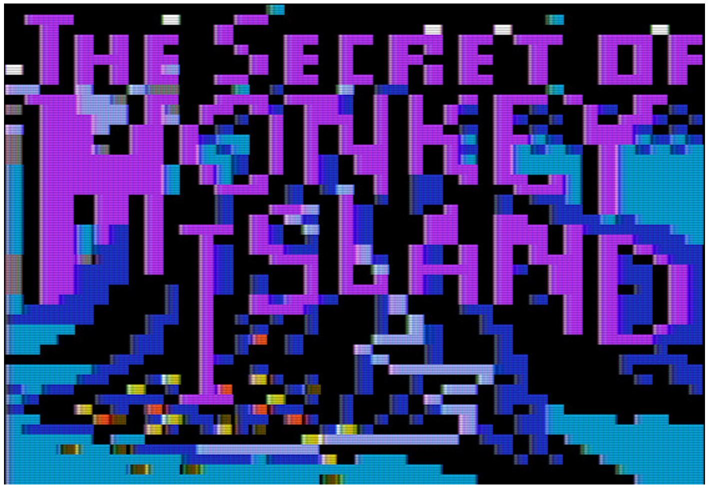


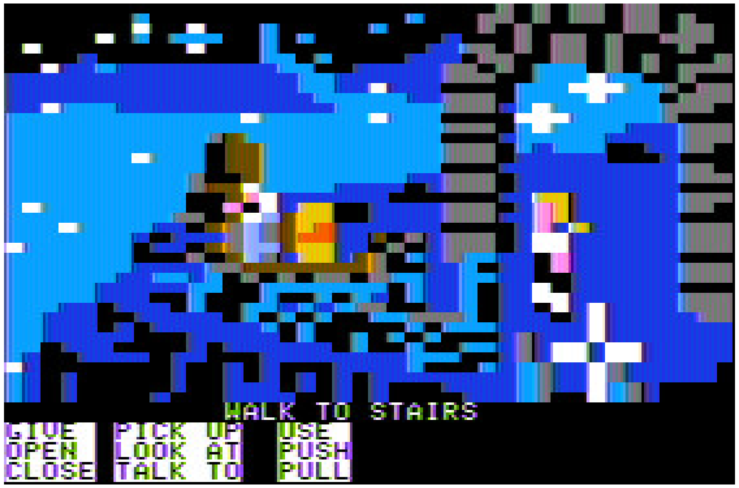


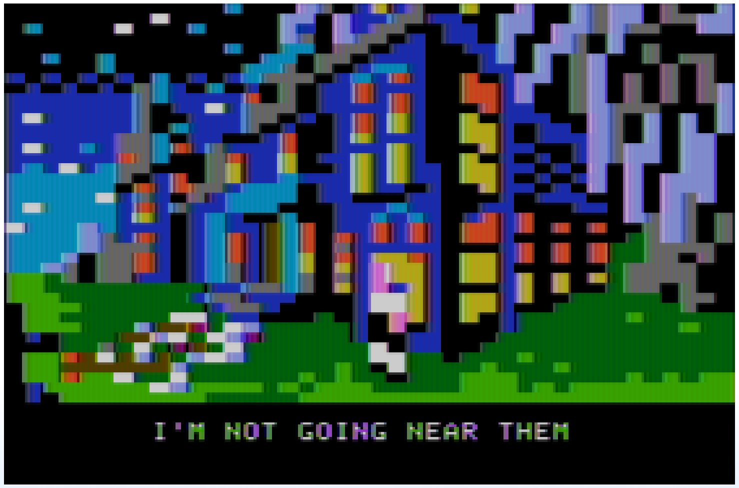
I SEE YOU HAVE DRIVEN HERE IN YOUR FERRARI.

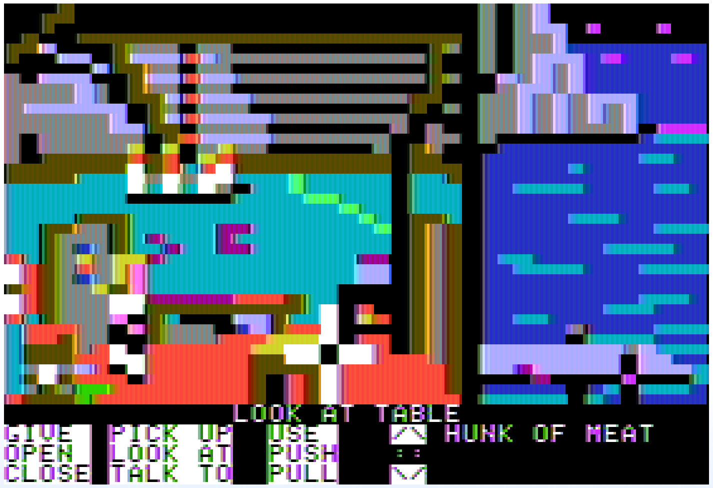


# My Demakes – Monkey Island

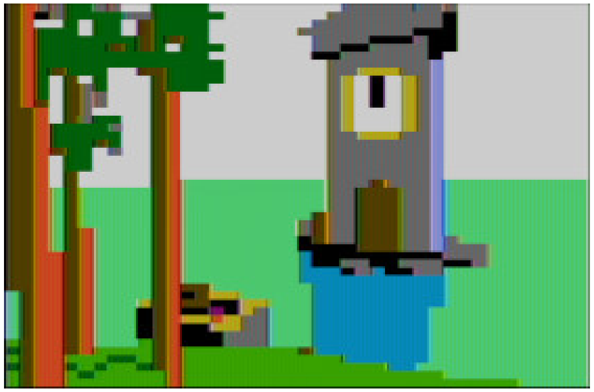


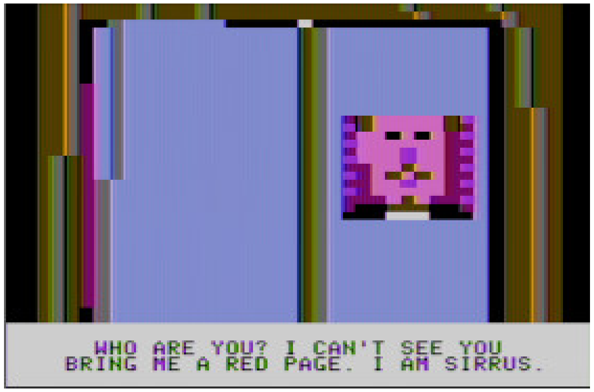

WALK TO STAIRS


I'M NOT GOING NEAR THEM

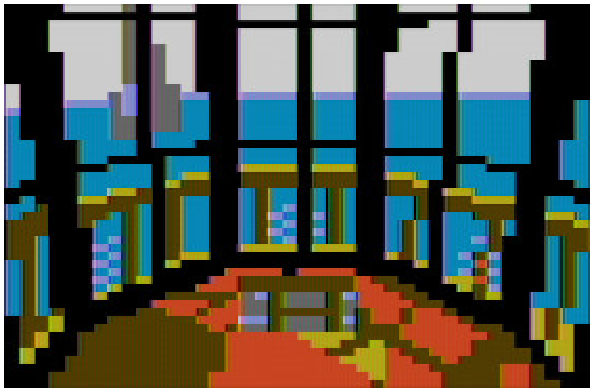

LOOK AT TABLE

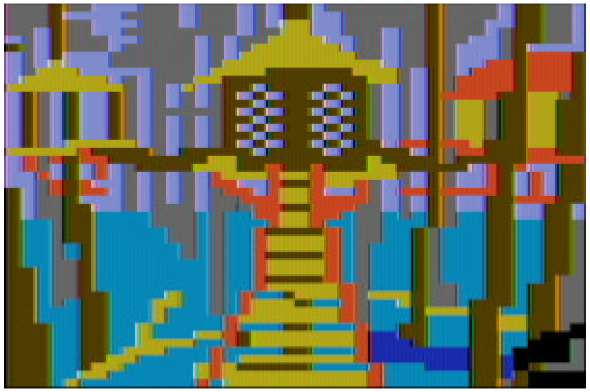
# My Demakes – Myst

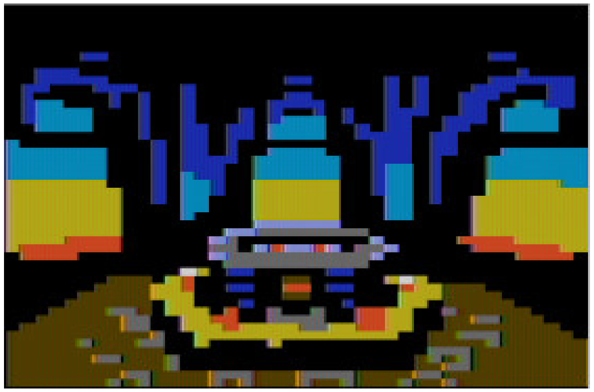


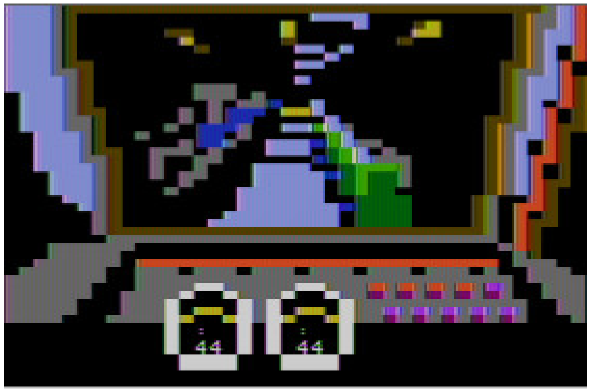


WHO ARE YOU? I CAN'T SEE YOU
BRING ME A RED PAGE. I AM SIRRUS.



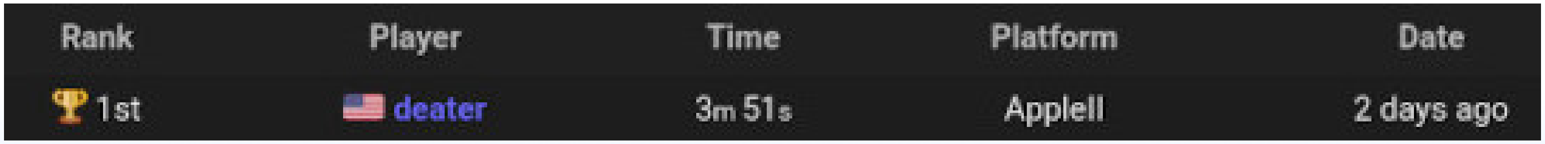






# Myst Demake

- `http://www.deater.net/weave/vmwprod/mist/`
- Fully Playable: All Ages, All Endings
- Apple II – 6502 assembly language
- 879 hand-rotoscoped 15-color 40x48 graphics
- Fits on 3 140k 5 ¼” disks
- For full details see my Kansasfest Presentation
- Copy lives in the Cyan vault



| Rank | Player | Time | Platform | Date |
|---|---|---|---|---|
| 🏆 1st | 🇺🇸 deater | 3m 51s | AppleII | 2 days ago |

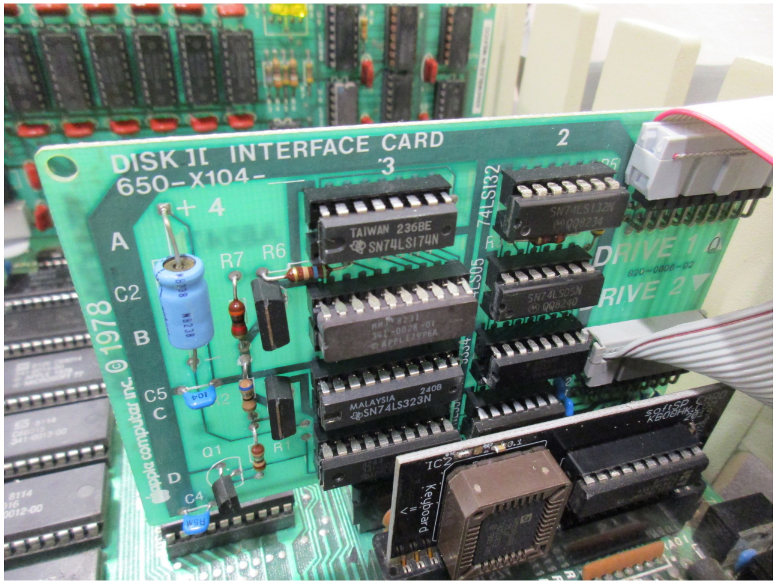
# How did I fit a CD game (500MB) onto 3 floppy disks (420k)?

## Can we make it not horribly slow?

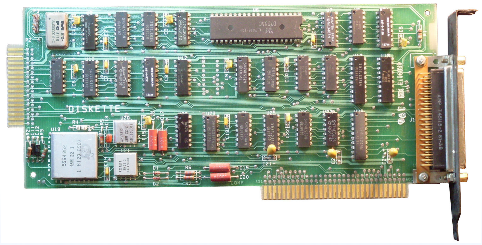
# Disk II drive

- Introduced in June 1978
- 5 $^1/_4$"
- Single-sided, double density
- 35 tracks
- Sectors: 256 bytes
- Originally 13-sector Tracks
- Later 16-sector Tracks (140k)
- Two drives can be connected
- 19-pin connector
- No track-0 sensor

# Disk II Controller



- Usually in slot 6
- Careful! Cable not keyed
- Woz spent Christmas 1977
- Woz got it to 8 chips
- Jobs demanded Shugart sell them stripped drive mechanism

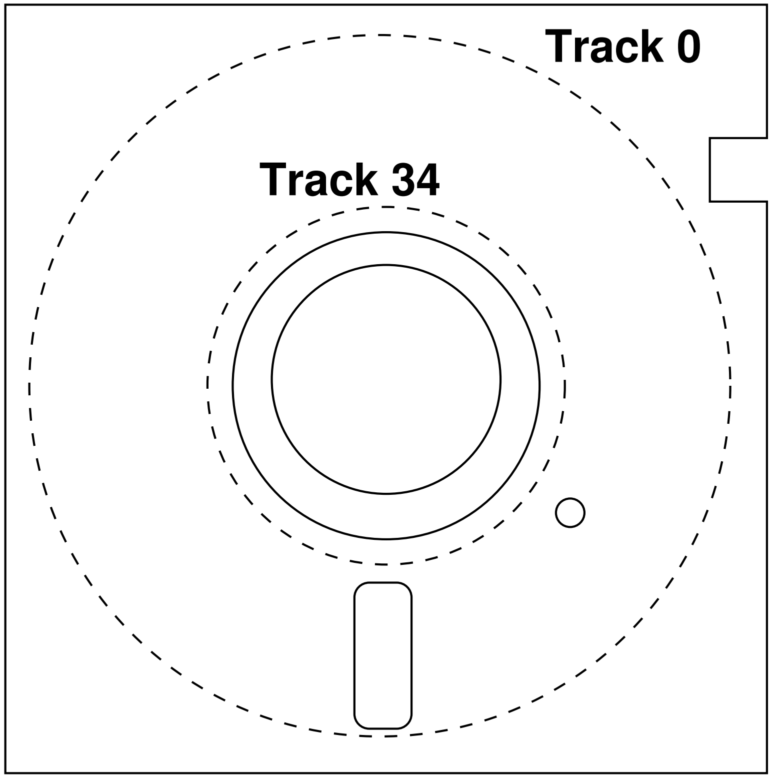


IBM PC Floppy controller for comparison

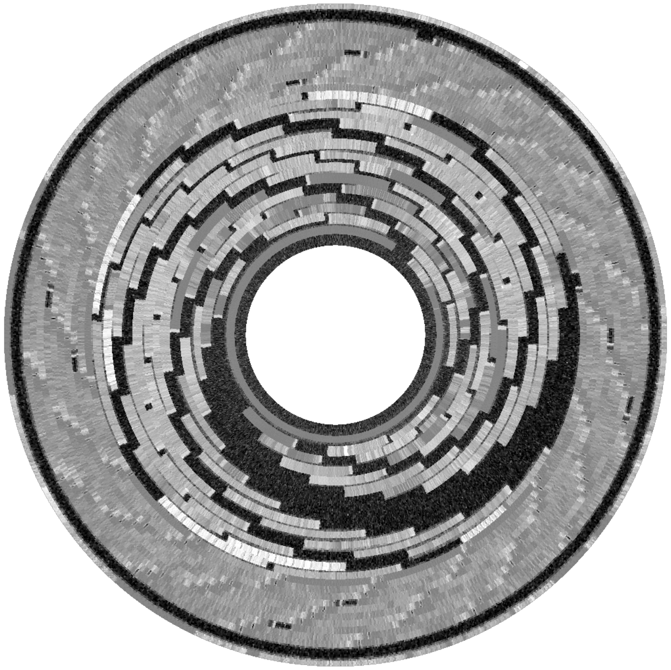
# Disk II low-level

- 35 tracks
- Soft sectored
- No track 0 sensor
- Stepper motor, step 1/4 tracks, full software control
- Can do lots of obscure copy protection

  See @a2_4am



# Disk II low-level



Applesauce Flux Image

Lode Runner

- Real time
- State machine @2MHz
- R/W every 32 cycles otherwise data loss
- Cycle-counted code
- What happens on faster CPUs?
- What about interrupts?
- Can you run code in background?

# Disk II Booting (DOS3.3)

Apple ][

- Original: boot to monitor, manually `C600G`
- "Autostart Rom" look for disk card, jump to boot ROM
- Loads boot sector T0S0 (256 bytes) to \$800, jumps to \$801
- Possibly multiple sectors (specified by 1st byte) usually 1
- Need to turn off floppy motor
- Then load second stage

# Disk II Disk Layout (16 sector)

- 6+2 GCR encoding
- Bytes need high bit, only one pair consecutive zeros
- 64 valid nibbles plus control
- Self syncing $FF gaps
- Address `D5 AA 96`, VOL, TRACK, SECTOR, CHECKSUM, `DE AA EB`
- Gap1 to give time to process
- Data `D5 AA AD`, 342 bytes, CHECKSUM, `DE AA EB`
- Gap2 adjusted at FORMAT to fit

# Disk II Interleave

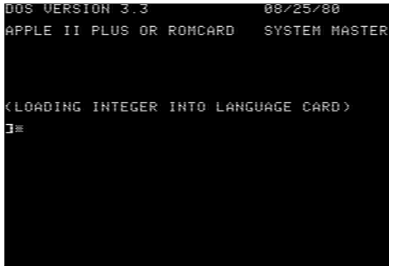
- Sectors in track not always 0 1 2 3 ... 15 as might expect
- Various filesystems interleave this (so have time to finish processing sector before disk spins around to next)
- ProDOS uses 512-byte blocks (joins together 2 sectors) but they aren't contiguous

| Physical | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| DOS3.3 | 0 | 7 | E | 6 | D | 5 | C | 4 | B | 3 | A | 2 | 9 | 1 | 8 | F |
| ProDOS | 0 | 8 | 1 | 9 | 2 | A | 3 | B | 4 | C | 5 | D | 6 | E | 7 | F |
| Pascal | 0 | 8 | 1 | 9 | 2 | A | 3 | B | 4 | C | 5 | D | 6 | E | 7 | F |
| CP/M | 0 | B | 6 | 1 | C | 7 | 2 | D | 8 | 3 | E | 9 | 4 | F | A | 5 |

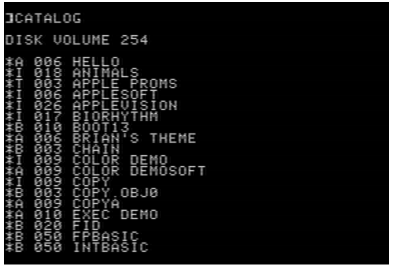
# Apple DOS3.3 History

- Initial release was DOS3 (?)
- Designed by HW Engineer and high school student (Woz and Wigginton) who apparently never used a real (UNIX) filesystem
- Releases 3.1, 3.2, 3.21 all 13-sector
- DOS3.3 Released August 1980
- 16-sector, required swapping ROMs on controller to update state machine

# DOS3.3 Low-Level



```
DOS VERSION 3.3                 08/25/80
APPLE II PLUS OR ROMCARD   SYSTEM MASTER


(LOADING INTEGER INTO LANGUAGE CARD)
]
```



```
]CATALOG
DISK VOLUME 254
*A 006 HELLO
*I 018 ANIMALS
*T 003 APPLE PROMS
*I 006 APPLESOFT
*I 026 APPLEVISION
*I 017 BIORHYTHM
*B 010 BOOT13
*A 006 BRIAN'S THEME
*B 003 CHAIN
*I 009 COLOR DEMO
*A 009 COLOR DEMOSOFT
*I 009 COPY
*B 003 COPY.OBJ0
*A 009 COPYA
*A 010 EXEC DEMO
*B 020 FID
*B 050 FPBASIC
*B 050 INTBASIC
```

- Tacks $0, $1, $2 hold the OS
- Includes minimal file utils, INIT, CATALOG
- Track 17 reserved for VTOC, free bitmap
- Track/Sector List: one sector each 122 sectors
- Metadata (filesize, address) stored in files, not in filesystem
- Filename 30 chars long.
  Right padded with space. Must start capital letter. No comma or colon. Control chars OK
- Access OS commands by intercepting stdout
  Inside of BASIC print CHR$(4) to send to DOS
- No subdirectory support

# So Why Not DOS3.3 For Myst?

- It's slow
- It takes up 4 tracks (16k) of disk space ($> 10\%$ of disk)
- By default reserves $9600–$BFFF 10k of RAM ($> 20\%$ of 48k)
- It's slow

# Why is DOS3.3 Slow?

- At least in part because it does a lot of extra copying between buffers (rather than just reading into one place)

# Are there Alternatives?

- ProDOS – Apple's replacement to DOS. Much more advanced
- Various third-party fast DOS3.3 replacements
  - Beagle Bros ProntoDOS
  - David Dos
  - Diversi-DOS
  - Quick DOS

# Qboot by Qkumba

- `https://github.com/peterferrie/qboot`
- Designed for loading game images from disk with minimal overhead
- No filesystem, just loads Track/Sector range from disk
- Only 3 sectors (768 bytes) to boot
- Can load up to 48k straight to memory with no buffering
- Can optionally load into language card
- 3 pages of memory, no ZP after init, 6 bytes stack
- Scatter read, full track read in one revolution

# Did Games modify RWTS for Speed?

- According to cracker/archivist @a2_4am most changes were for copy protection. Very few speed improvements.
- Roland Gustafson (Late Brøderbund SW) used a fast 18-sector RWTS (Prince of Persia)
- Burger Becky Heineman's graphical adventures (Borrowed Time, The Tracer Sanction, Mindshadow) used a custom RWTS that could load tracks as fast as qkumba's fastest

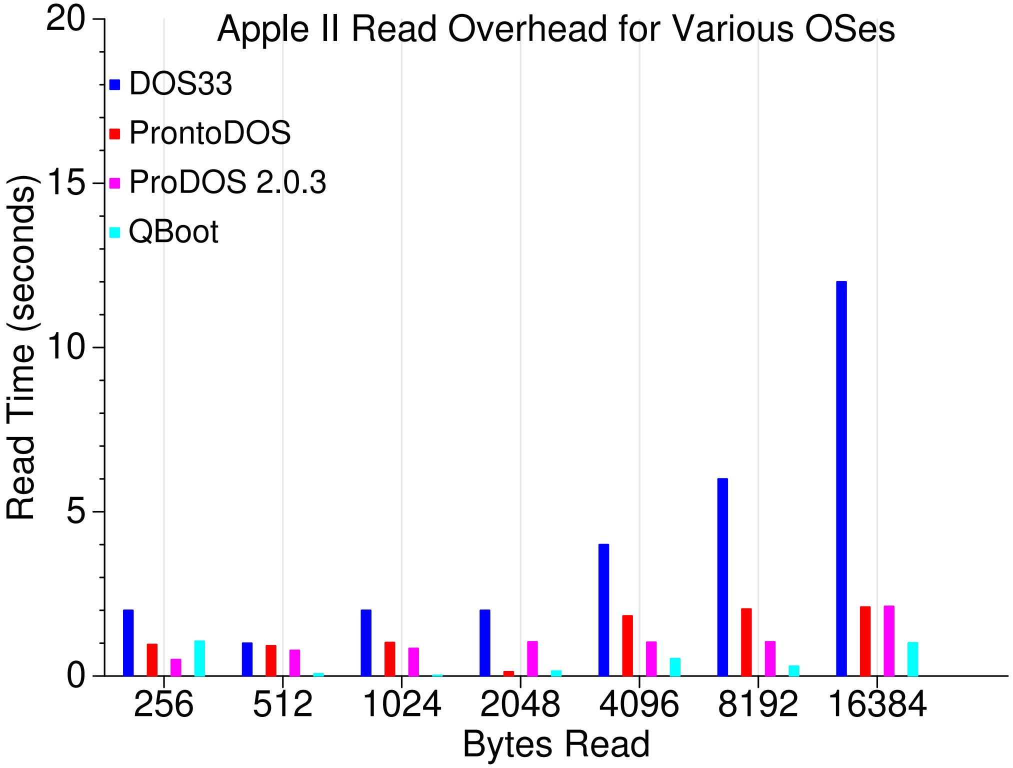
# Benchmarks – Methodology

- Write 256, 512, 1024, 2048, 4096, 8192, 16384 blocks to disk from memory $2000 (HGR) with freshly INIT version of OS
- Read back same values in loop
- Before/after each read/write click the speaker by X=PEEK(-16336) or similar
- Record this on AppleWin Emulator (with disk speed set to "accurate") [ran out of time to run on real hardware]
- Load into Kdenlive and measure distance between clicks

# Benchmarks – Why this isn't great metric

- If drive isn't spinning, takes 1s to spin it up
- Depends on disk layout of sectors, will vary with OS
- Having to seek to new track takes time
- Depends on (random) location of head on track when starting access what sector gets read first
- Drive *does* have Constant angular speed (300RPM) So read time is not different outer vs inner tracks

# Benchmark: Reads

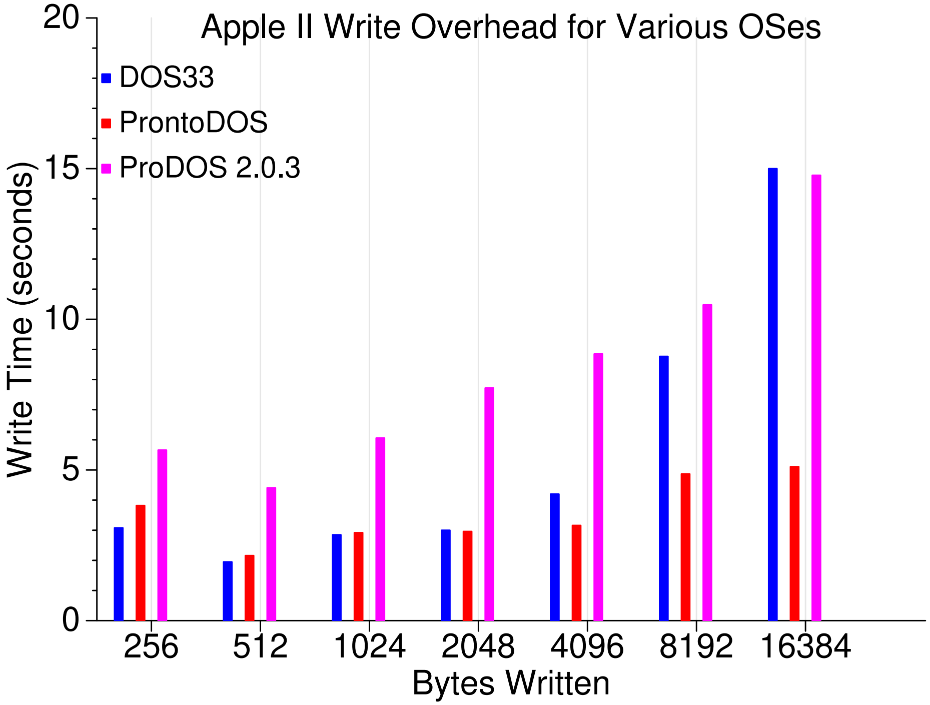


# Benchmark: Read Summary

| Setup | Avg Read Speed |
|---|---|
| DOS33 | 1,365 B/s |
| ProntoDOS | 7,801B/s |
| ProDOS 2.0.3 | 7,727B/s |
| Qboot | 16,221B/s |
| Apple II Cassette | 167B/s |
| C64 1541 drive | 400B/s |

# Benchmark: Writes

Note that the writes often do an immediate read for VERIFY.
ProDOS behavior makes more sense if you read *Beneath Apple ProDOS*



# Other Issues (besides Load Speed)

- Obtaining maximum space on disks (qboot low overhead, only 3 sectors)
- Telling what disk you are on (making sure right floppy inserted)
  - DOS33 puts volume number on each sector, but DSK emulator format does not include it
  - In end code just reads boot sector to (for Myst Demake each disk slightly different)
- Writes for save game

# Myst – Disk Progression

- First used DOS3.3 — had the tools, easy to set up
- Moved to FASTLD6/RTS, a fast RWTS by qkumba Still use DOS3.3 filesystem, but no DOS in memory. Custom compact routines for finding and loading files
- This still has disk overhead and is still relatively slow, as had to navigate the DOS3.3 file structures and seek to each track/sector
- Used QBoot instead: faster and lower memory footprint

# Myst – Uses Modified Qboot

- Modified to be callable as function
  - Specify new track/sector, will seek to it
  - Added code to start disk motor, wait 1s until stable
- Wrote utility to put data on disk, ProDOS order
- Manual layout of files due to lack of filesystem
- Save games on disk1
- No qboot on disk2/3: just an “insert disk1” message

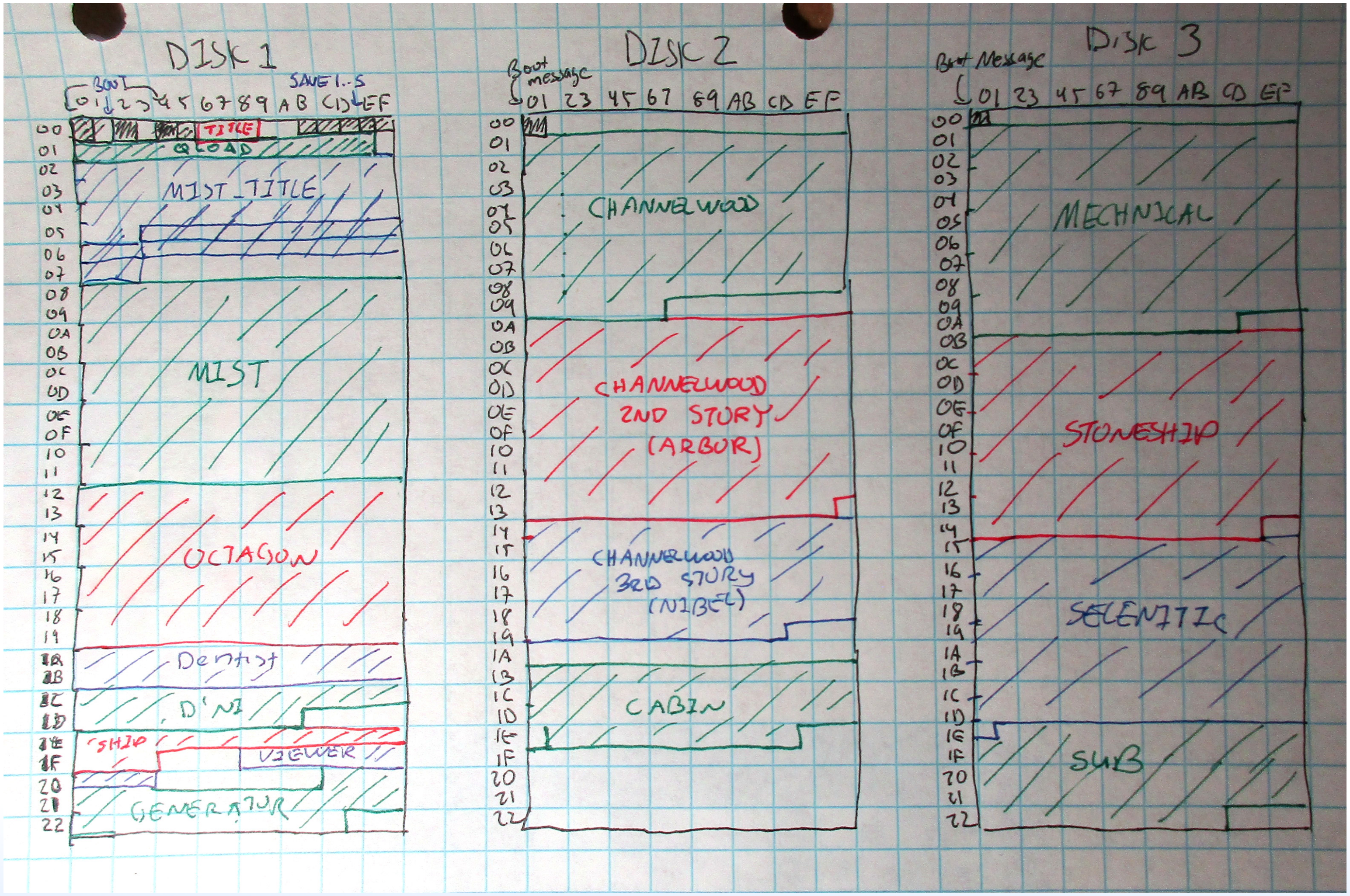
# Write support

- How do you write if no real filesystem?
- Luckily save game for Myst is less than 256 bytes
- A matter of re-using the seek code and then just writing a single sector
- POPW single-sector write code from qkumba
- Temporarily over-write graphics memory when saving
- Put sector to write in $d00, (copy from zp) encoded nibbles at $e00, bit2tbl at $f00

# Myst Memory Map – Not to Scale

| | | |
|---|---|---|
| $0000 - $00ff | 256B | Zero Page |
| $0100 - $01ff | 256B | Stack |
| $0200 - $02ff | 256B | grouped table |
| $0300 - $0369 | 112B | preshift table |
| $03d0 - $03ff | 48B | interrupt vectors |
| $0400 - $07ff | 1k | lores page 1 |
| $0800 - $0bff | 1k | lores page 2 |
| $0c00 - $0fff | 1k | bg graphics |
| $1000 - $11ff | 512B | qboot code |
| $1200 - $1fff | 3.5k | common routines |
| $2000 - $bfff | 40k | current level data |
| $c000 - $cfff | 4k | I/O |
| $d000 - $ffff | 12k | ROM |
| $d000 - $ffff | 12k | (lang card) link sound effect |

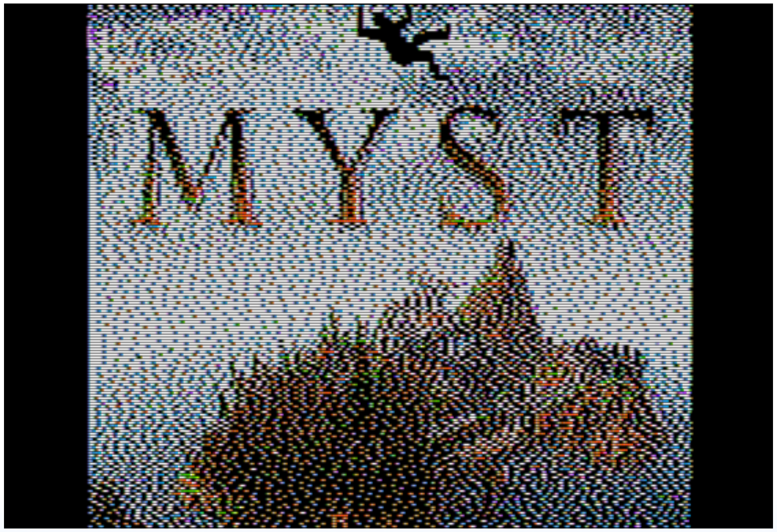
# Myst Disk Layout



# Conclusion

It *is* possible to get fast disk loads on Apple II

Just don't use DOS3.3



# Questions?

`vince@deater.net`

## More Info and Sourcecode

Apple II Myst Demake

`http://www.deater.net/weave/vmwprod/mist/`

Also check out Total Replay by @a2_4am and Qkumba

`https://archive.org/details/TotalReplay`

# End Notes

References:

- *Beneath Apple DOS*, Worth and Lechner
- *Beneath Apple ProDOS*, Worth and Lechner
- *Sophistication and Simplicity*, Weyhrich

Picture Credits

- Original 5 1/4 Diskette Drive Adapter found on the IBM PC (IBM 5150)
  By German, CC BY-SA 3.0, `https://commons.wikimedia.org/w/index.php?curid=18623045`
- Signed floppy – BigMessOfWires
- License Plate – Fabrice Sanglard's Prince of Persia Writeup
- Applesauce Image – 4am