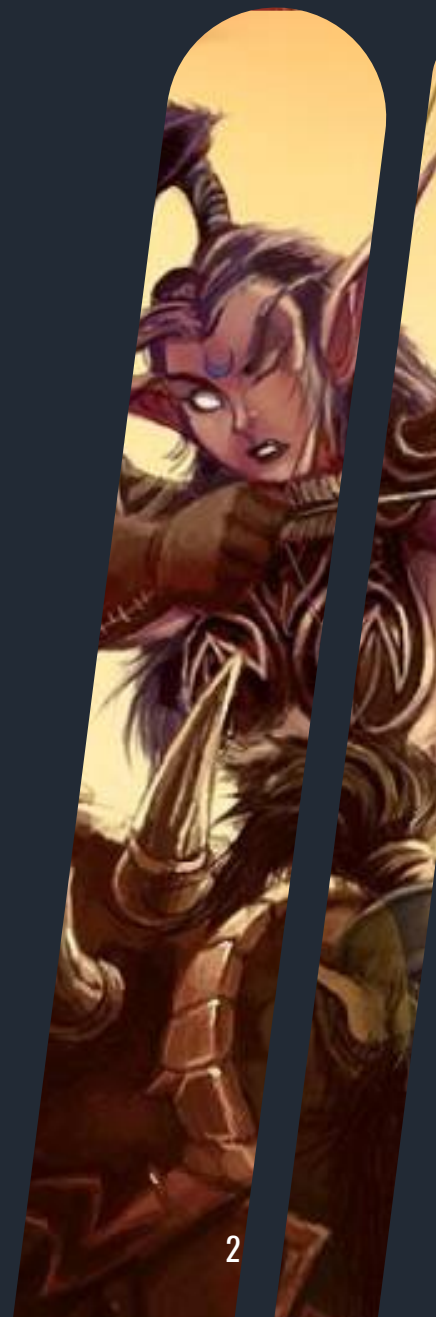
Noel Danz & Raoul Baron – WS18/19

Reverse-Engineering

Extracting Game Assets from Warcraft 1 (MacOS)

Overview

- What we had as a starting point
- What we‘ve done so far
- Which tools we were/are missing
- What we want to from here on



What we had as a starting point

BLIZZARD ARCHIVE

Warcraft: Orcs & Humans

Files/Versions Screenshots

Version History

Ver.	Date	Comment/Download	List File	Archive Version
Warcraft 1 PC				
	Oct 04, 1994	Warcraft Demo 1	Demo	Demo
	Oct 06, 1994	Warcraft Demo 2	Demo	Demo
	Oct 06, 1994	Warcraft Interplay Demo	Demo	Demo
1.12	Nov 03, 1994	Retail Floppy	Retail	Retail
1.12	Nov 03, 1994	Retail CD-ROM	Retail	Retail
1.12	Nov 23, 1994	Warcraft Demo 3	Retail	Demo
1.13				
1.14				
1.15	Dec 27?, 1994	Warcraft Patch 1.15		
1.16	Jan 25, 1995	Warcraft Patch 1.16		
1.20	Mar 06, 1995	Warcraft Patch 1.20		
1.21	Mar 15, 1995	Creative Labs Demo CD 1.21	-TODO-	-TODO-

BIN-FILE

Extras

How Warcraft 1 files work (PDF)

Warcraft 1 Insider's Guide (1995-08-27)

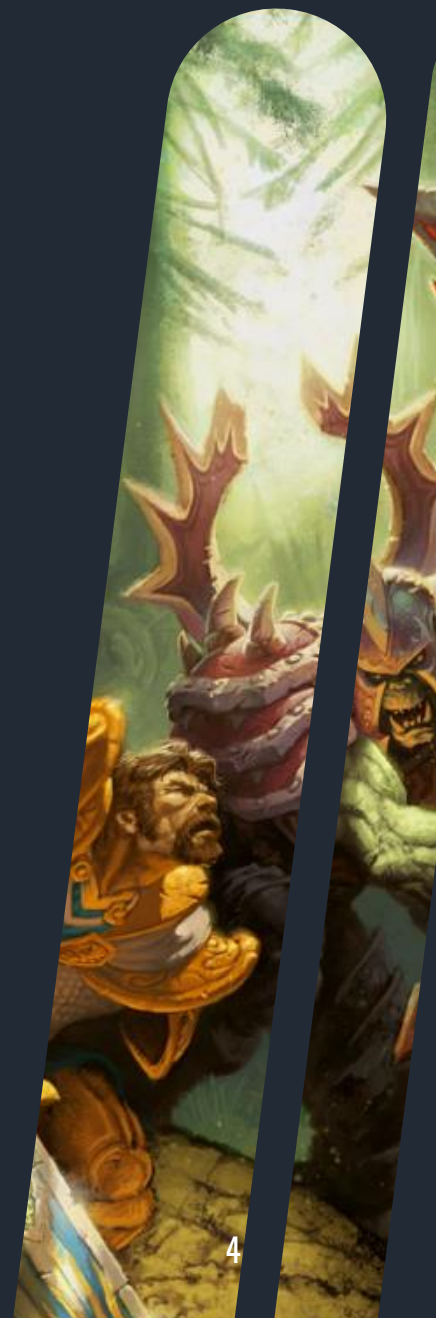
All DATA.WAR files (except MPlayer and TEN)

Wargus / war1gus

Code | Issues 26 | Pull requests 1 | Projects 0 | Ins

What we‘ve done so far

- (tried to) understand the WarData format
- rewrite some extraction tools in Python
- collected problems we‘ve encountered while working
- extracted text & audio
- developed some ideas for helpful tools



What we've done so far – the format

HEADER (MacOS)

Filetable Entry

```
Offset(d) 00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15
00000000  00 00 00 1A 00 00 02 47 00 00 12 40 00 00 00 00  ........G...@....
00000016  00 00 12 40 00 00 00 00 00 00 12 40 00 00 00 00  ...@.......@....
00000032  00 00 12 40 00 00 00 00 00 00 12 40 00 00 00 00  ...@.......@....
00000048  00 00 12 40 00 00 00 00 00 00 12 40 00 04 F7 05  ...@.......@..÷.
00000064  00 05 09 45 00 07 CA E6 00 0C D4 2B 00 06 01 FB  ...E..Êæ..Ô+...û
00000080  00 12 D6 26 00 07 21 CE 00 19 F7 F4 00 12 47 02  ..Ö&..!Î..÷ô..G.
00000096  00 2C 3E F6 00 14 30 05 00 40 6E FB 00 0F 34 2E  .,>ö..0..@nû..4.
00000112  [illegible]4F A3 29 00 0A 18 28 00 59 BB 51 00 09 F3 31  .O£]...[.Y»Q..ó1
00000128  00 63 AE 82 00 00 00 00 00 63 AE 82 00 00 00 00  .c®,.....c®,....
00000144  00 63 AE 82 00 00 00 00 00 63 AE 82 00 00 00 00  .c®,.....c®,....
00000160  00 63 AE 82 00 00 00 00 00 63 AE 82 00 00 00 00  .c®,.....c®,....
00000176  00 63 AE 82 00 00 00 00 00 63 AE 82 00 00 00 00  .c®,.....c®,....
00000192  00 63 AE 82 00 00 00 00 00 63 AE 82 00 00 00 00  .c®,.....c®,....
```

What we've done so far – the format

```
00000064  00 05 09 45 00 0
00000080  00 12 06 26 00 0
00000096  00 2C 3E F6 00 1
          4F A3 29 00 0
00000128  00 63 AE 82 00 0
00000144  00 63 AE 82 00 0
```

Filetable Entry

Placeholder: 00 00 00 00 || FF FF FF FF

Retail-Placeholder: 00 00 00 00 (increment with each occurence)

20 0A 00 00

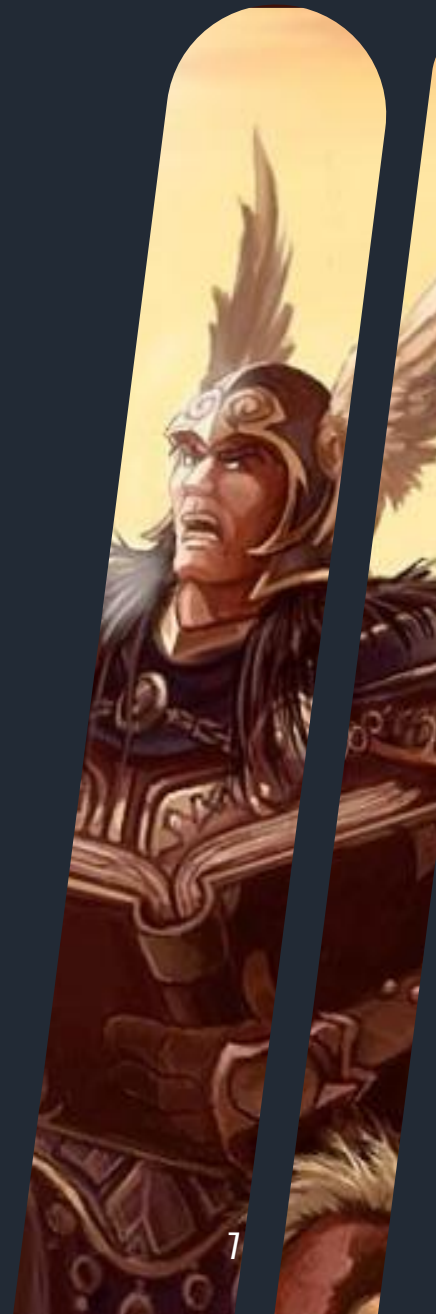
→ 3rd highest bit decides if compressed

→ & 0x1FFFFFFF gives (uncompressed) filesize

→ offsets[n+1]-offsets[n]-4 to get blobsize

What we‘ve done so far – extraction

- we‘ve collected some heuristics for deciding filetypes
  - classic: magic numbers, headers
  - „provable“ statements
  - File-Entropy
- dev. a scoring method to compare probabilities for different formats
  - use the „provable“ statements as indicators

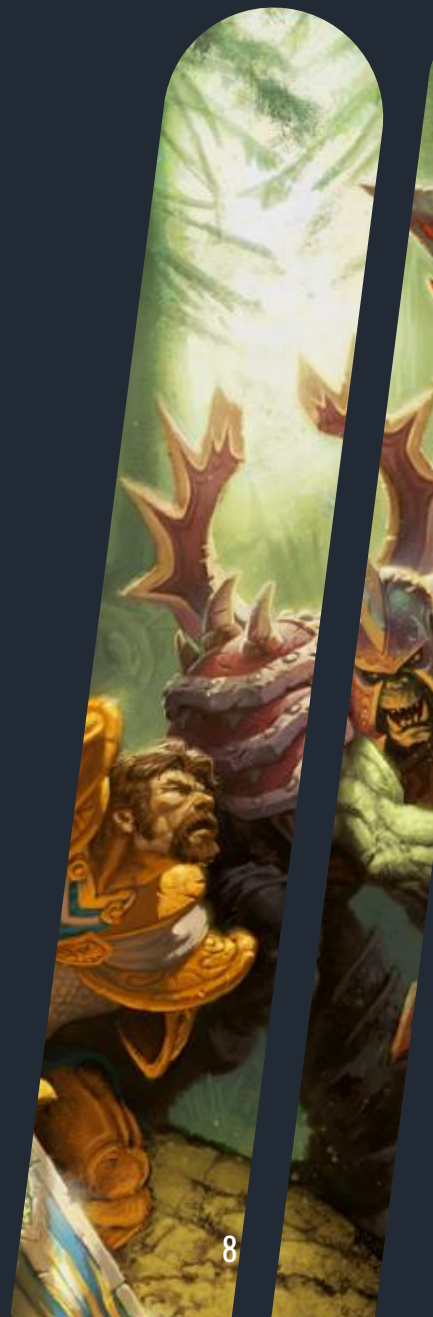


What we've done so far – Demo Time!

```
PS C:\Users\noeld> & python d:/OneDr
Removing empty blobs...
Found 583 blobs, 472 not empty
Filesize limit in kb: 400
Specify start index: 100
```

```
File 431 Size: 558 kb, entropy: 0.7021710342246725
Play audio file? (y/n) y
Playing file 431... was the file playable? (y/n) y
```

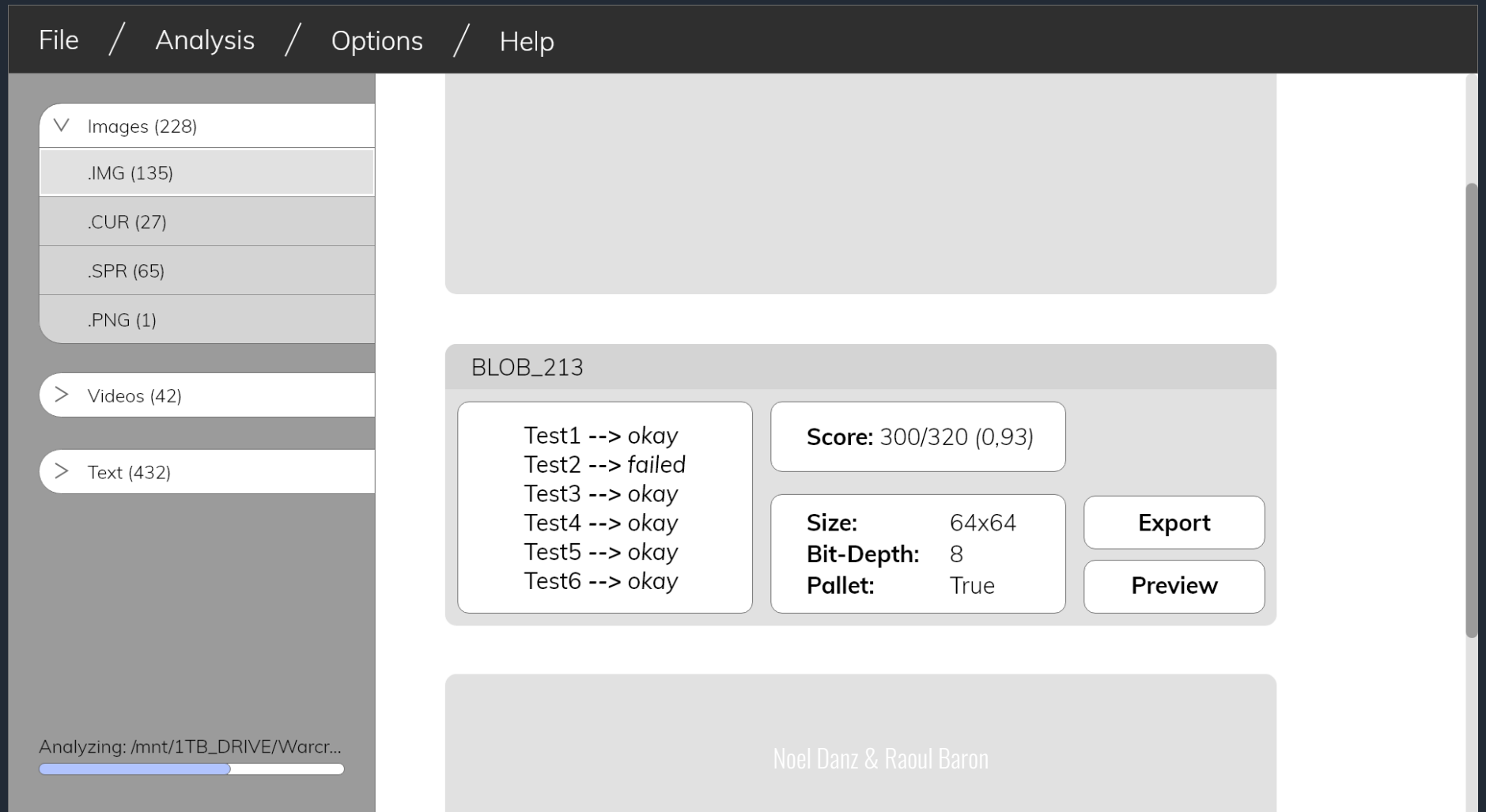
```
Average audio entropy: 0.6446205509876245
```

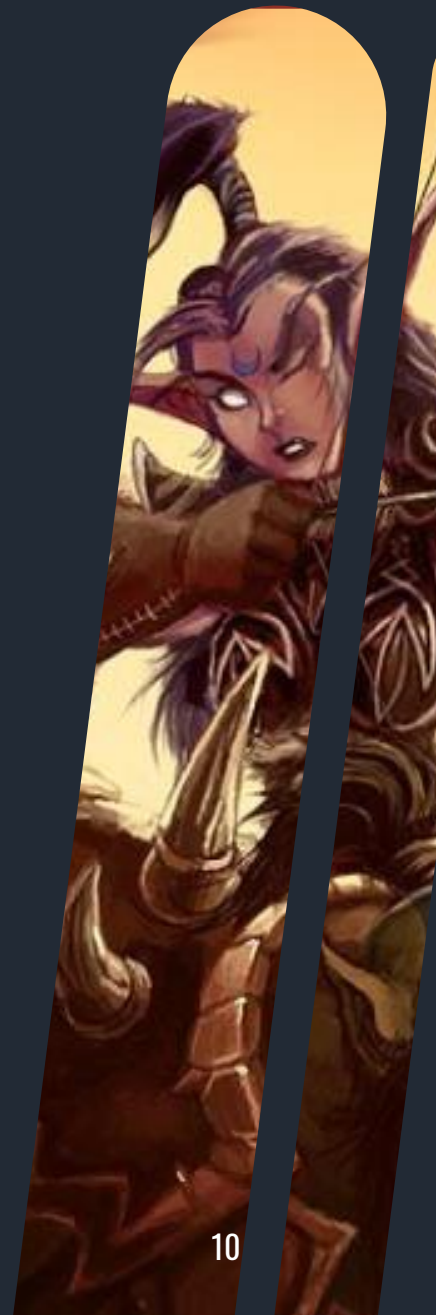


What we‘ve done so far – ideas for tools

- a GUI for analyzing the probabilities that a given blob of data contains a certain filetype
  - marking that part
- a way to describe the probability that a file is of a certain type

What we've done so far – ideas for tools





What we want to do from here on

- develop the presented tools (for better ease-of-use)
- extract all available assets from the Binary (as it is the original task)

Thanks!