MAL WARE TECHNICAL REPORT

text archaeology jp reader

Biyi Wen

Doctoral Student, University of Colorado at Boulder

Introduction

"text archaeology jp reader" is a customized research tool and a publishing project on the material history of Japanese text processing. Supported by the Race x Technology micro grant, the reader has developed two components: a dictionary and a visual archive. The dictionary collects the relevant Japanese words on text processing and the corresponding English translations; and the visual archive presents archival documentation on Japanese text processing. As my research continues, the dictionary and the archive will continue to collect more input. The scope of the reader covers the domains in which Japanese text is processed and circulated, such as typewriting, home computing, early internet history, and network culture.

Process

As I collected, translated, and read literature to research Japanese text processing, I identified two needs that call for the development of customized tools. I need a dictionary companion thematically centered on text processing to help me navigate literature; and a visual archive to store and present the visual documentation collected from literature. Hence, the project can be regarded as an annotation tool bound together with my research in text processing. At the same time, the project's level of customization,

inclusion of artistically interesting archival content, and interface design also make it a creatively driven project.

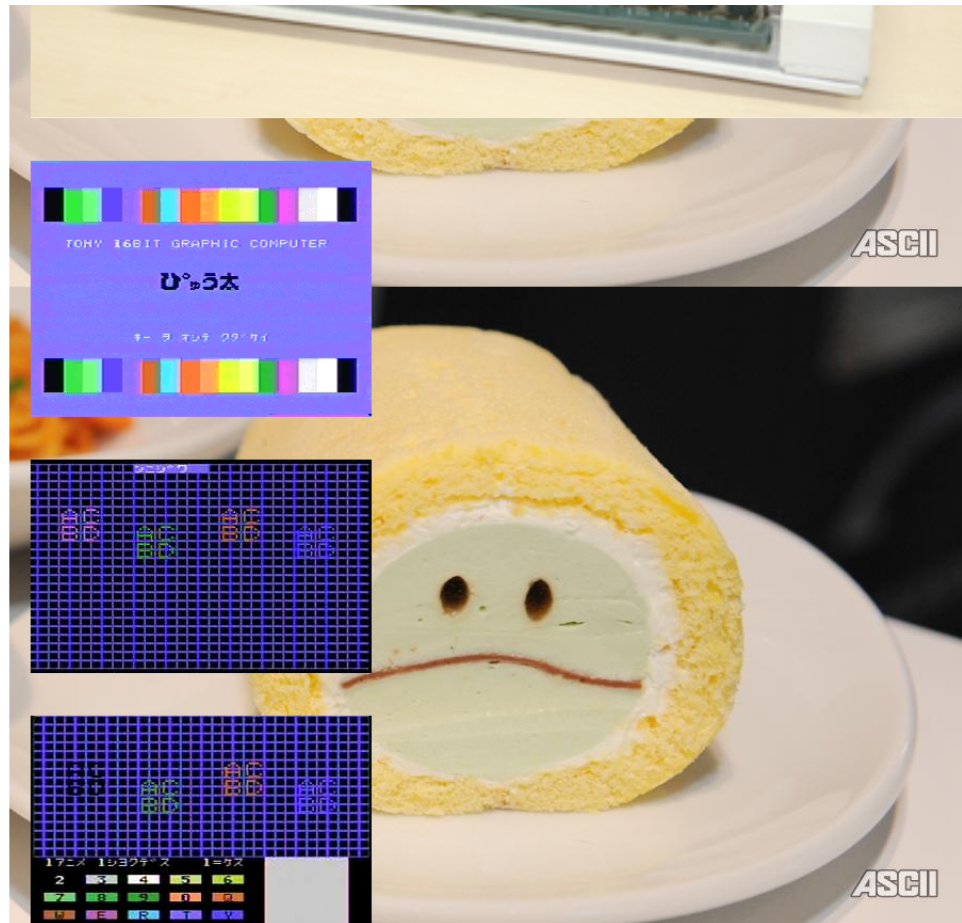## Examples of Collected Items In Visual Archive

3.1 Item: Tomy Pyuuta

Figure: Archival detail that showcases the manipulation of text from Tomy Pyuuta, a Japanese home computer model launched in 1982.
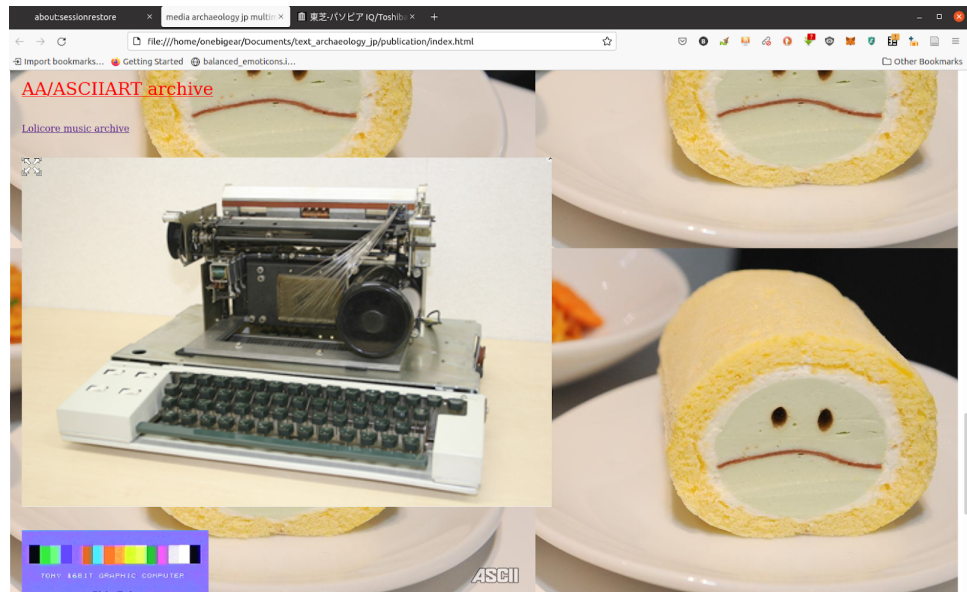
3.2 Item: OKI Wiredot Printer

Figure: Archival photograph of the Oki Electric Wiredot Printer, a dot matrix printer released in 1968.
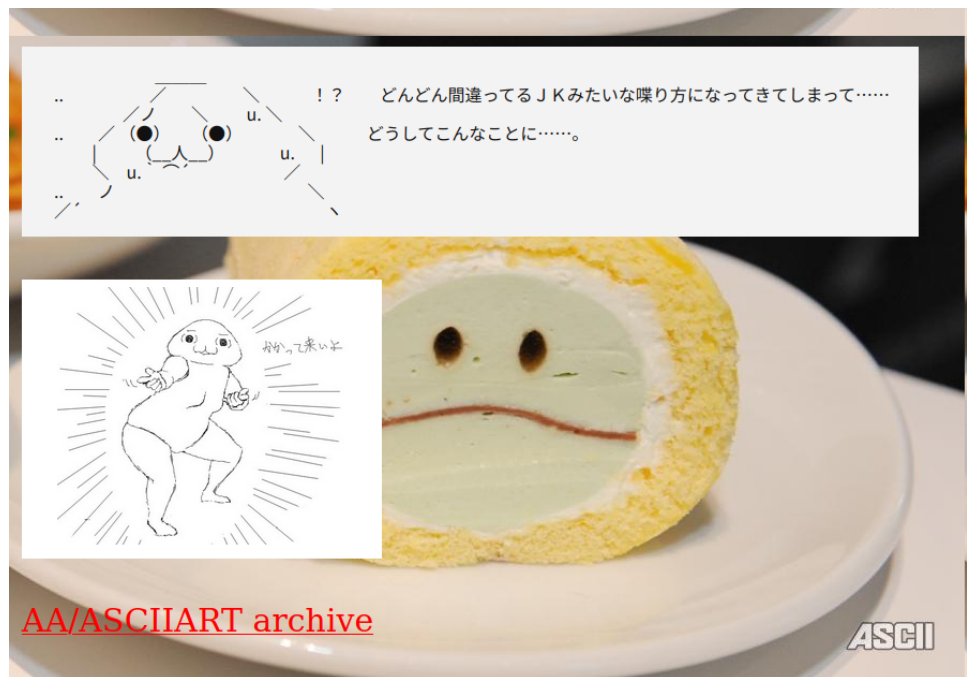
3.3 Item: Yaruo ASCII Meme

Figure: Yaruo, a text based ASCII meme that is viral in Japanese internet culture.

Technical Outline

Both components take advantage of the CSV and JSON formats to store metadata and content. In the dictionary, the Japanese words and the corresponding translations are entered into a CSV file. A script transforms the file into the JSON dictionary format. In the visual archive, content such as image, link, and video are entered and stored in a CSV file. A script transforms the file into the JSON format that is fed to an html generator. The html generator produces an html page that works as the visual interface of the archive. The visual archive uses the class attribute to differentiate between different document file types.

```
    "Byte": "バイト",
    "Computer": "コンピューター",
    "Arcade": "アーケード",
    "Internet": "インターネット",
    "Network": "ネットワーク",
    "Personal Computer": "パソコン",
    "Cyberspace": "サイバースペース",
    "Second Summer Of Love": "セカンドサマーオブラブ",
    "Centralization": "中央集権化",
    "Underground Subculture": "地下サブカルチャー",
    "Counterculture": "カウンターカルチャー",
    "Data": "データ",
    "Penpal": "ペンパル",
    "System": "システム",
    "Game magazine": "ゲーム雑誌",
    "Computer magazine": "パソコン雑誌",
    "Telephone line": "電話回線",
    "Telephone book": "電話帳",
    "Den Den Mushi telephone snails": "電伝虫",
    "California Ideology": "カリフォルニアイデオロジー",
    "Neuromancer": "ニューロマンサー",
    "TRON": "トロン",
    "Whole Earth Catalog": "全地球カタログ",
    "Hackers": "ハッカー",
    "Television Game Anthology": "",
    "Computer communication": "通信",
    "Eighth Grader Syndrome": "中二",
    "Trance": "鬼畜",
    "Host": "ホスト",
    "Ghost in the Shell": "攻殻機動隊",
    "Yaruo": "やる夫",
    "e-zine": "",
    "server": "サーバ",
    "mail art": "メールアート",
    "links": "リンク",
    "Cyberpunk": "サイバーパンク",
    "Denpa-kei": "電波系",
    "Otaku": "オタク",
    "Kawaii": "かわいい",
    "Decentralization": "地方分権化",
    "Distribution": "分布"
(base) onebigear@onebigear-ThinkPad-X1-Carbon-Gen-9:~/Documents/text_archaeology_jp/dictionary$
```

Figure: Dictionary output in the JSON file format.

Alternatively, both the dictionary and the visual archive can be thought of as customized and small scale alternatives to the more complex bibliography systems. The CSV works as a "back-end database" , in which I can add information on the fly; and the interface works as a "front-end", in which information is rendered tangibly in the browser.

The workflow is intended to be nimble, small, fast, and simple.