

Singing Data Labeling Tool

Milestone 1

Avinash Persaud, Nandith Narayan, Carlos Cepeda

GUI

Top 3 choices for GUI Development:

1. Gtkmm
2. **Qt**
3. wxWidgets

We needed to choose GUI development kits that supported C++, and that supported cross-platform development.

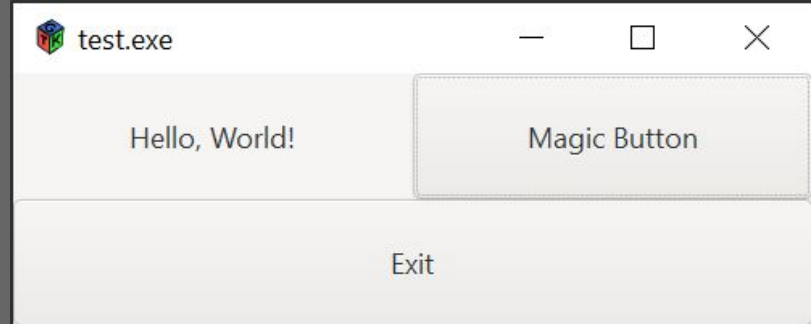
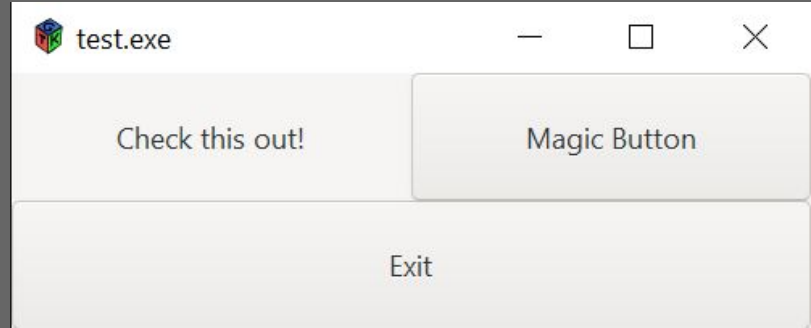
Gtkmm

Pros:

- Known as excellent linux UI development kit
- Provides glade UI for designing GUIs
- Healthy documentation

Cons:

- Not cross-platform oriented
- Is actually a C programming language kit (Gtk+) which is easier to implement than its C++ interface



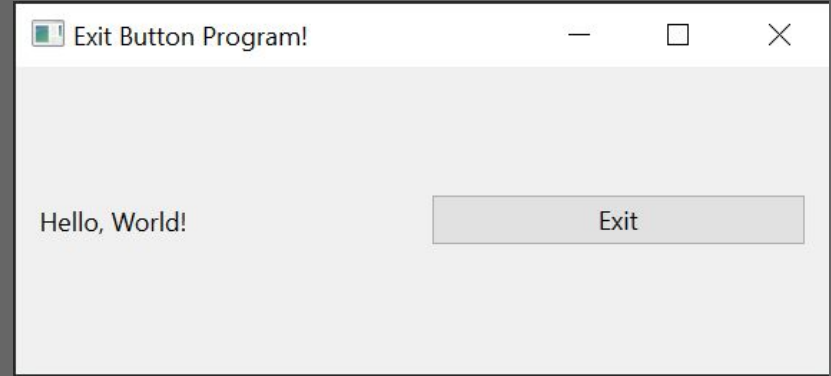
Qt

Pros:

- Industry standard
- Easy to read documentation
- Supports cross-platform development
- Provides UI for designing GUIs
- Provides IDE for easy class creation
- Made to be object oriented unlike Gtk which uses object C

Cons:

- Costly for closed-source software



wxWidgets

Pros:

- Open source
- Supports cross-platform development
- Backed by a very supportive community, and is always kept up to date

Cons:

- Bigger learning curve than the other 2 kits
- More limited documentation and tutorials available online



Need for Parsing

- We want to allow the user to create a template for the label format.
- To use the user defined template, we need to parse the user's input.

The screenshot shows a user interface for creating a label format template. The main area is a large text input field containing a template string: `phonemes[0].start * 10000000` `phonemes[0].end * 10000000`
`phonemes[0].type` `@` `phonemes[-2]` `^` `phonemes[-1]` `-` `phonemes[0]` `+`
`phonemes[1]`. The text is displayed in a monospaced font. Below the input field, there are several controls: a button labeled "Insert dynamic entry" on the right; a section with "On failure insert:" followed by a text box containing "xx"; a "Name:" label followed by a text box containing "HTS Song Full Label"; a "Repeat for each:" label followed by two buttons labeled "Phoneme" and "V"; and at the bottom, "Cancel" and "Save" buttons.

phonemes[0].start * 10000000 phonemes[0].end * 10000000

phonemes[0].type @ phonemes[-2] ^ phonemes[-1] - phonemes[0] +

phonemes[1]

Insert dynamic entry

On failure insert: xx

Name: HTS Song Full Label

Repeat for each: Phoneme V

Cancel Save

Parsing

- Created a simple grammar for our parser
- Tested the “Bison” parser generator with c++
- Tested the “Antlr v4” parser generator with java
- Tested the “Antlr v4” parser generator with c++

Grammar for parsing the template

```
1 grammar Labelformat
2 labelformat : statement+;
3 statement : value | expression;
4 value : Identifier index (Dot attribute)?;
5 index : openSqrBracket Integer closeSqrBracket;
6 attribute : Identifier;
7 expression : value binOp number;
8 binOp : Plus | Minus | Times | Division | Power;
9 number : Integer | Real;
10 Dot : '.';
11 Plus : '+';
12 Minus : '-';
13 Times : '*';
14 Division : '/';
15 Power : '^';
16 openSqrBracket : '[';
17 closeSqrBracket : ']';
18 Integer : Minus? [0-9]+;
19 Real : Integer (Dot [0-9]+)?;
20 Identifier : [a-zA-Z]*;
```


Selected parser generator

- We selected Antlr v4.9.2 as our parser generator
- Antlr is written in java but supports other languages
- We had to compile the antlr c++ runtime to get it to work with c++

Phoneme Alignment

- Successfully tested CMU Sphinx
 - Does not work well with singing
- SHIRO was not fully evaluated yet
- Potential Phoneme Sampa restrictions

CMU Sphinx Talking Demo

```
Frames: 323
Word: SIL 0 2
Word: AO 3 19
Word: TH 20 29
Word: ER 30 45
Word: DH 46 56
Word: AH 57 63
Word: D 64 71
Word: EY 72 85
Word: N 86 93
Word: SH 94 99
Word: UW 100 108
Word: T 109 119
Word: R 120 129
Word: EY 130 140
Word: L 141 158
Word: F 159 170
Word: IH 171 177
Word: L 178 186
Word: AH 187 192
Word: P 193 199
Word: S 200 211
Word: D 212 217
Word: IY 218 233
Word: L 234 248
Word: Z 249 254
Word: IH 255 263
Word: T 264 270
Word: S 271 279
Word: AE 280 288
Word: T 289 293
Word: ER 294 302
Word: AH 303 321
```

Audio from CMU Arctic database

“Author of the Danger Trail, Philip Steeles, etcetera”

Approximate english for phonemes:

“Author the deynshuw trail filep sdeels atera”

Notes:

- Model can be tuned
- System was made for words

CMU Sphinx Singing Demo

```
Frames: 1432
Word: SIL 0 126
Word: T 127 132
Word: AE 133 182
Word: NG 183 188
Word: IY 189 203
Word: DH 204 206
Word: EH 207 279
Word: N 280 282
Word: L 283 287
Word: EH 288 302
Word: HH 303 317
Word: T 318 323
Word: IH 324 327
Word: F 328 330
Word: ER 331 366
Word: M 367 369
Word: AE 370 402
Word: M 403 415
Word: N 416 418
Word: AH 419 426
Word: M 427 439
Word: L 440 446
Word: IH 447 463
Word: D 464 470
Word: F 471 487
Word: ER 488 522
Word: M 523 525
Word: AE 526 556
Word: M 557 559
Word: UW 560 586
Word: T 587 589
Word: L 590 602
Word: EH 603 616
Word: SIL 617 631
Word: T 632 639
Word: OW 640 673
Word: SIL 674 698
Word: M 699 717
Word: ZH 718 746
Word: D 747 763
Word: N 764 766
Word: AE 767 814
Word: N 815 823
Word: IY 824 833
Word: HH 834 838
Word: EH 839 872
Word: DH 873 879
Word: OW 880 907
Word: L 908 916
Word: IH 917 930
Word: D 931 937
Word: SIL 938 952
Word: T 953 957
Word: IH 958 973
Word: L 974 994
```

```
Word: AE 995 1017
Word: M 1018 1029
Word: UW 1030 1058
Word: Z 1059 1073
Word: L 1074 1082
Word: EY 1083 1100
Word: S 1101 1112
Word: HH 1113 1118
Word: UH 1119 1138
Word: Z 1139 1150
Word: L 1151 1168
Word: EY 1169 1191
Word: N 1192 1197
Word: IH 1198 1219
Word: S 1220 1236
Word: N 1237 1244
Word: OW 1245 1328
Word: SIL 1329 1430
```

Audio recorded at home

“Mary had a little lamb, little lamb, little lamb. <breath> Mary had a little lamb whose fleece was white as snow”

Approximate english for phonemes:

“Tangy then leht if erm am nom lid firm am /uw/ / /t/ // /eh/ <sil>to <sil> /m/ /zh/ /d/ nanny head owl ihd <sil> till”

SHIRO

- Supplied example model fails
- Example dataset is known
- Known to be used for singing alignment

Phoneme Alignment Conclusions

- Further research necessitated
- Specialized dataset may be needed

Milestone 2 Tasks

- Implement and demo preliminary UI layout
- Ascertain phoneme alignment method
- Start building and testing underlying data structures
- Implement file I/O for project file