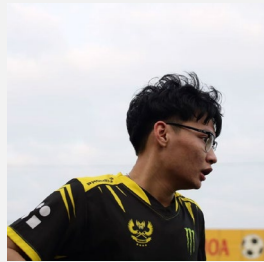




DDT Dictionary

Data Structures and Algorithms Advanced Lab



DDT 's here!

We consist of 3 members:

Pham Minh Dang 20194737

Mai Anh Duc 20194740

Tran Phuc Thanh 20194846

Table of Contents.

01- GUI

The project's interface

02 – Usages

Using BTree

03 - Autocomplete

Handling with GTK Events

04 - Suggestion

Using soundex library

05 –Performance

A millionword test



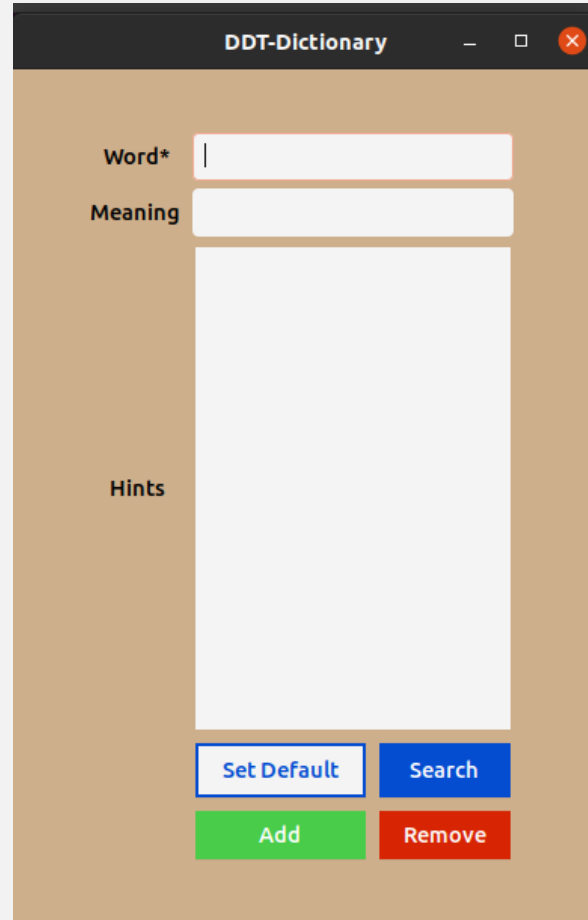
01

GUI

Using GTK library to create an User Interface

The Project's Interface

- GTK library used
- CSS Involved
- Buttons, Inputs, Labels

A screenshot of a desktop application window titled "DDT-Dictionary". The window has a dark title bar with standard window controls (minimize, maximize, close). The main content area has a light brown background. It features a "Word*" label next to a text input field. Below it is a "Meaning" label next to another text input field. A large, empty rectangular box is labeled "Hints". At the bottom, there are four buttons: "Set Default" (blue), "Search" (blue), "Add" (green), and "Remove" (red).

DDT-Dictionary

Word*

Meaning

Hints



02

Usages

What can the dictionary do?

Word*

Meaning

Remember to press this to
Search/Add another word

Set Default

Search

Add

Remove

Press this to search for
the word entered

Fulfill both "Word" and "Meaning"
fields to add new words

Remove the word in the
"Word" input field



03

Suggestion

Using ~~soundex~~ library to track similar words

Words with the same sound code
will appear in the “Hints” table below

Word*	as
Meaning	
Hints	age ago as ask

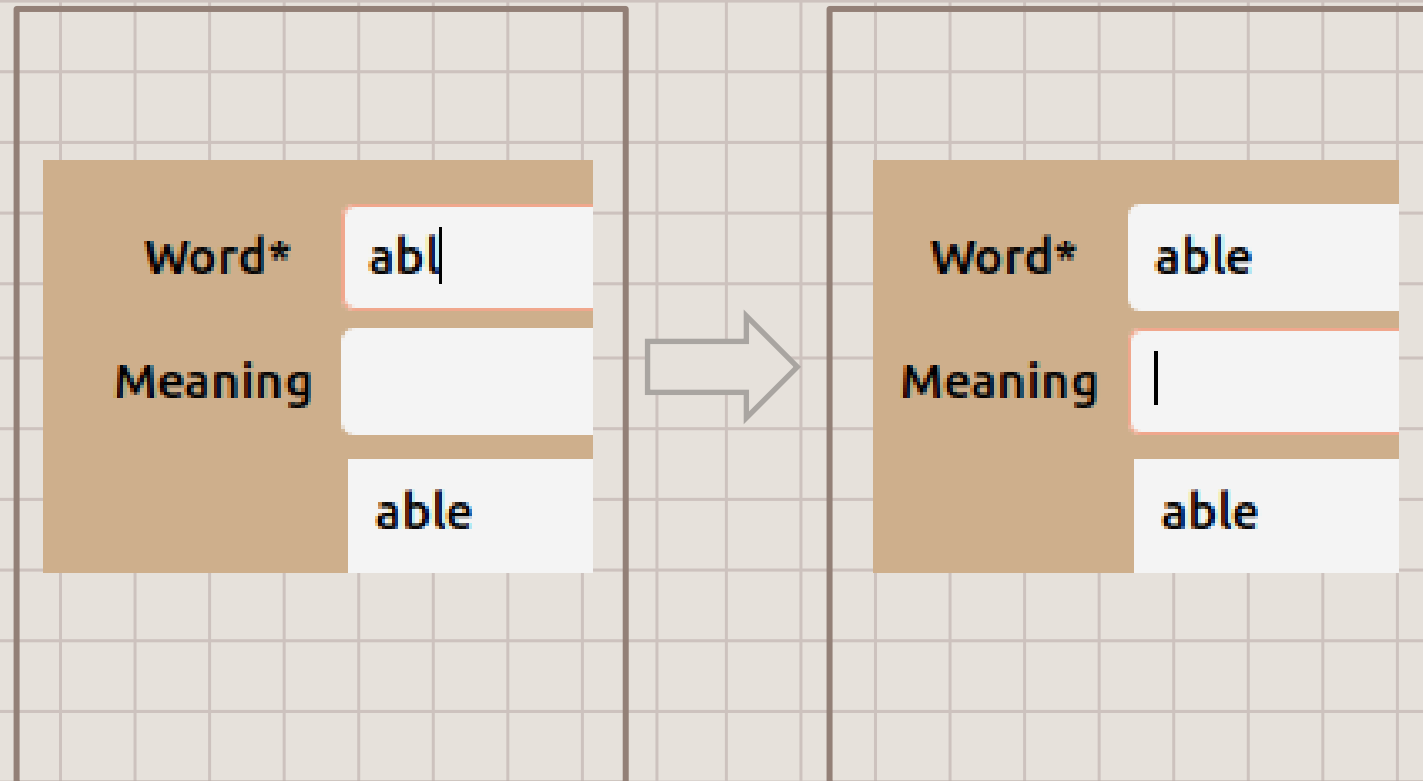


04

Autocomplete

Using the GTK Keypress events

Pressing “TAB” key





05

Performance

A millionword test

```
list.txt
999969 ghnwy gquzq
999970 dmhse fuudb
999971 kshfp sxiif
999972 rgjbi mzsoe
999973 mzfss pcsfx
999974 upxqu btgql
999975 uygxt lqhbc
999976 thowx adbaj
999977 fvtim bmqnp
999978 dgkaf drncy
999979 jhmsz ymack
999980 sshch hmxnp
999981 ueqwe fklmf
999982 xlolz plpzz
999983 bwc pz mbvot
999984 kpdfg eor pz
999985 reife babvv
999986 oaqib dotxt
999987 ljwle ldcye
999988 lvo jw tw nzi
999989 kwjwf ncqur
999990 zwbbr gmlfx
999991 nvpdn gkmex
999992 dxkau aubhn
999993 jyojp vzoam
999994 wysgp jilbt
999995 mxuno vnzmh
999996 awnos gxnte
999997 krvyi fpekt
999998 ktkfa trvac
999999 yjnrd vpkxd
1000000 bmzij ady zl
1000001
```

```
list.txt
1 phqgh umeay
2 lnlf d xfirc
3 vscxg gbwkf
4 nqdux wfnfo
5 zvsrt kjpre
6 pggx r pnrvy
7 stmw c ysyyc
8 qpevi keffm
9 znimk kasvw
10 srenz kycxf
11 xtls g ypsfa
12 dpooe fxzbc
13 oejuv pvabo
14 ygpoe ylfpb
15 npljv rvipy
16 amyeh wqnqr
17 qpmxu jjloo
18 vaowu xwhms
19 ncbxc oksfz
20 kvatx dknly
21 jy hfi xjsw n
22 kkufn uxxzr
23 zbmnm gqook
24 etlyh nkoau
25 gzqrc ddiut
26 eiojw ayyzp
27 vscmp sajlf
28 vgubf aaovl
29 zylnt rkdc p
30 wsrte sjwhd
31 izcob zcnfw
32 lqijt vdwvx
33 hrcbl dvgy l
34 webus bmbor
```

List.txt

A file consists of 1.000.000 words to test was created outside the project folder

66

```
chmod +x dictionary.sh  
./dictionary.sh
```

How to compile?

Thank you!

Have any questions?

