

## 14:18

考虑了下, 还是不加 `command` 太多功能, 就用于 `confirm` 就行.

增加类型 `Confirm`, 增加成员 `confirm`:

```
type Confirm struct {  
    ans  textinput.Model  
    hint string  
}
```

## 相关函数

- `confirmView()` string, 在底部展示 `confirm` 的信息, 如:

`Confirm: Do you want to delete tree [hello]? <yes|no> [your input]` 这里的 `tree, [hello]` 虽行为不同而改变. `<yes|no>` 及其之前的内容不可更改.

- `confirmOk()` bool, 会切换 `mode` 并卡住程序 `if confirmOk() { // delete or other }`, 根据 `[your input]` 的内容返回 `true` 或 `false`

流程, 按下 `d` 进入 `confirm mode`, 执行 `confirmView()` 显示输入, 根据输入判断是否进入 `del mode` 且改变 `operation` 字符串, 之后进行接下来的操作.

## bug

现在 `m.mode = del` 和 `m.mode = confirm` 死循环了 ☹