What You Say is What You Get

Handsfree Coding in 2023

BTW 2023, Dresden, Germany
March 08, 2023
Wolle
It's Simple, Really!

The requirements:

- **Microphone**: Every notebook has one!
- **Speech Recognition Software (SR)**: Included in Windows since 2007!
- **Voice Command Execution**: Available in every SR software!

LET'S G0000000!!!!!!
Let Me Just Show You How Easy It is
Let Me Just Show You How Easy It Is
Let Me Just Show You How Easy It is

Go Watch Emily’s Talk!
Where's the Challenge?

ONE DOES NOT SIMPLY

TALK TO A COMPUTER
Outline: What This Talk is Going to Cover

1. My Personal Background
   As data engineer & scientist, I use hands-free coding every day.

2. Demo & Usage Examples
   Hands-free coding is awesome and can be useful for everyone!

3. Setup & Best Practices
   No-cost base setup with optional upgrades (e.g., for eye tracking).

4. How to Get Started
   Videos, blogs, articles, support & community – engage now!
My Job is Data Science
I Am Wolle

I’m a data guy, not an ASR or HCI expert!

Wolfram Wingerath
Data Science

Research:
• Stream Processing
• Real-Time Databases
• NoSQL & Cloud Systems
• ...

Practice:
• Web Caching
• Big Data Analytics
• Anger Management
• ...

Carl von Ossietzky
Universität Oldenburg

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Baqend
Look, No Hands!
Basic Voice Control

- **Symbols, Modifiers & Navigation:**
  - Numbers, brackets, etc.: 
    - What you say: one space air bat shift one
    - What you get: 1_ab!

- **Spelling through a phonetic alphabet:**
  - NATO alphabet: alpha bravo charlie delta echo → abcde
  - Optimized alphabet: air bat cam drum each → abcde

- **Command Management** for efficiency, e.g.:
  - Chaining:
    - (paren close paren go left say hi) → (hi)
    1.) type: ()
    2.) move cursor: ←
    3.) dictate: hi

- **On-the-fly grammar prototyping** through Python live reloading!
Dynamic Scripting (Live Demo)

mode: sleep
-

hallo BTW:
  key(down)

"now two things will happen: the cursor will move and then text will appear"
Context-dependent behavior, for example:

- C#:    funky test funk → private void testFunk()
- JavaScript: funky test funk → function testFunk()

Intuitive IDE shortcuts such as

- "run code" instead of <shift-f10>
- "find usage" instead of <ctrl-alt-f7>

Powerful templates, e.g.:

```python
action(user.code_state_if):
    insert("if () {}")
    key(left enter up end left left left)
```
Handsfree Coding: Cursorless

- Available on GitHub: github.com/cursorless-dev/
- VSCode extension
- Spoken language for structural code editing
  - Decorates every token on screen with a mark
  - Tokens can be selected via combination of mark and scope
  - Actions operate on the specified tokens

Example:

```
bring call vest
```

(action scope mark)

(copy the function call with the marked „v“ to where my cursor is)
Handsfree Coding: Cursorless

(move code around)
Handsfree Coding: Cursorless

(selecting semantic entities)
Snappy Noise Control With Parrot

- Available on GitHub: github.com/chaosparrot/parrot.py
- Noise-controlled actions with latency <50ms
- Workflow
  1. Record sounds
  2. Train model for recognition
  3. Map sounds to actions
- Compatible (and recommended in combination with) with other tooling:
  - Often used with Project IRIS (eye tracking)
  - Can be used to produce **Talon-compatible** models

---

Custom noises for your Talon grammar!
Eye Tracking & Noise Recognition

- **Calibration** for adjusting your eye tracker to your current position
- **Noises** for actions (e.g. clicking & right-clicking):
  - Extremely low latency (<50ms)
  - Talon currently supports *pop* & *hiss*
  - Custom noise models available via Parrot
- **Different Modes** for convenience:
  - Zoom: (1) *pop* for zooming, (2) *pop* for clicking
  - Head tracking: eye gaze (jumps) + head movement (adjustment)
- **Debug** mode & camera overlay

[github.com/chaosparrot/parrot.py]
import React from 'react';
import styled from 'styled-components';

function IconButton() {
}

export default IconButton;
Handsfree Gaming: Eyes + Face + Voice/Noise

wolle.science/twitch
The Base Setup
Upgrades & Add-Ons
Multi-Computer Setup
Recognition Accuracy Issues

- **Microphone** determines accuracy!
  - *Build quality*: built-in < gaming headset < stage mic
  - *Positioning*: consistent, close to your mouth, away from all noise
  - *Mixed bag*: Noise canceling via hardware or software (e.g. RTX Voice)

- **Environment**: Minimize noise for you and annoyance for others!
  - Suspend ASR / mute mic accordingly (e.g. via push-to-talk pedal)

- **Homophones** should be avoided, e.g. through:
  - Grammar optimization to avoid ambiguity
  - Clear pronunciation
Potential Privacy Issues

- **Watch Your Tongue**: Passwords & confidential info may be leaked...
  - ... through plain acoustics (beware eavesdroppers!)
  - ... as they are stored your *command history!*
  - ... to involved third parties (e.g. with *Web Speech*)
- **Watch Your Transmitter**: Wireless solutions are often not encrypted!
- **Watch Your Eyes**: Your eye movement may give away a lot
  → perhaps avoid continuous eye tracking ;-)

<insert eye tracking challenge joke here>
Workflow & Anger Management Issues

- **Beware the Trolls**: Having an audience generally does not help!
  - Prepare to hear „Format C“ from your colleagues a lot

- **Keep your calm**: Shouting at the computer will not help, either!
  - Stay in your neutral voice, even when raging inside ...

- **Avoid Voice Strain**: Find a comfortable way to speak A LOT!
  - e.g. use your natural voice & drink a lot of tea

- **Command chaining**: Anticipate what is going to happen!
  - Practice, practice, practice!
• **Multilanguage support** is still in its infancy
  → Non-English language models all have their problems
  → Designing command libraries for different languages means effort

• **Complex setup** with many moving parts:
  → Random stuff sometimes just happens, get used to it!
  → Fallback to manual input sometimes necessary …

• **MACHINE LEARNING!!!**
  → Models often reflect typical issues (data bias, data quality issues, …)
  → Sometimes you have to just hope for the best …
Ryan Hileman @lunixbochs

I just used 3,000 GPU-hours to test all 9 new OpenAI Whisper speech recognition models, two Talon acoustic models, and NVIDIA Nemo large and xlarge models. Whisper has a peculiar failure case. Here's what I think.

9:50 nachm. · 27. Sep. 2022 · Twitter Web App

185 Retweets 26 Zitierte Tweets 1,418 „Gefällt mir“-Angaben

Ryan Hileman @lunixbochs · 27. Sep.
Antwort an @lunixbochs

Whisper sometimes exhibits what I would call "catastrophic" failures in recognition quality. I've provided some examples in the linked sheet, and I'll talk more about this downturn.
"short context" / vocabulary tests. It places somewhere between Whisper Small and Whisper Base on those tests.

Ryan Hileman @lunixbochs · 27. Sep.
Whisper was painfully slow compared to the other models tested. I achieved much higher throughput when running my GPU tests on the ESPnet 1B model and Nemo xlarge (600M) model than any Whisper model, including Whisper Tiny (39M).

Ryan Hileman @lunixbochs · 27. Sep.
Whisper output "feels great. It is very good at producing coherent speech, even when it is completely incorrect about what was said. While analyzing the highest error %, I saw an audio clip of only the word "partnerships" as transcribed by Whisper Large as:

"That's the end of the video. Thank you for watching. Lots of heat, December. Keep writing your comments below for new videos and feel free to contribute. If you have any questions, feel free to ask away, or make comments, post any of your comments. I'll see you next week."

What you say
What you get
Why This is Still Worth All the **Hassle**

**Productivity**
- Speed up input-heavy tasks
- Faster navigation through easy-to-remember shortcuts

**Convenience**
- Intuitive interfaces
- Relieve your hands

**Accessibility**
*Compensate handicaps:*
- Injuries (e.g. broken hand)
- Repetitive stress injury (RSI)
- Cubital Tunnel Syndrome
- ...

**General Awesomeness**
- Talk to your computer!!!

**It's Awesome!**
Helpful Resources & Outlook
Tooling Recommendations (Incomplete!)

- **Talon** (Free of Charge): [talonvoice.com](http://talonvoice.com) / [talon.wiki](http://talon.wiki)
  - Voice coding for Win / Linux / Mac!
- **parrot.py** (noise control): [github.com/chaosparrot/parrot.py](http://github.com/chaosparrot/parrot.py)
- **Cursorless** (code editing for VSCode): [github.com/cursorless-dev](http://github.com/cursorless-dev)
- **Paid Upgrades:**
  - Talon Premium Support: [patreon.com/join/lunixbochs](http://patreon.com/join/lunixbochs)
  - Dragon Speech Recognition: [nuance.com/dragon](http://nuance.com/dragon)
Alternatives: Speech Recognition

- Speech Recognition
  - WSR (Windows Speech Recognition): Built into Windows
  - Kaldi: [github.com/kaldi-asr/kaldi](https://github.com/kaldi-asr/kaldi)
  - Vosk (ASR on mobile devices!): [github.com/alphacep/vosk-api](https://github.com/alphacep/vosk-api)
  - Web Speech API (compatible with Talon through Chrome or Firefox)

- Scripting:
  - NatLink: [sourceforge.net/p/natlink/](http://sourceforge.net/p/natlink/)
  - Dragonfly: [github.com/dictation-toolbox/dragonfly](https://github.com/dictation-toolbox/dragonfly)
  - Caster: [github.com/dictation-toolbox/Caster](https://github.com/dictation-toolbox/Caster)
  - Vocola (Voice Command Language): vocola.net
Recommended Talks

Emily Shea. **Voice Driven Development: Who needs a keyboard anyway?, Strange Loop (2019)**

Boudewijn Aasman. **Coding by Voice with Dragonfly, PyGotham (2018)**

David Williams-King. **Coding by Voice with Open Source Speech Recognition, The Eleventh Hope (2016)**

Wolfram Wingerath

What You Say is What You Get: Hands-free Coding in 2022

Buzzing Technologies

hosted by SCAYLE™
Articles & Blogs

- Emily Shea: [whalequench.club/](https://whalequench.club/)
  - Talon user
  - Very good starter instructions
- James Stout: [handsfreecoding.org/](https://handsfreecoding.org/)
  - Dragonfly user
  - Huge collection of relevant blog posts
- Dusty Phillips (2020): [dusty.phillips.codes/2020/02/15/on-voice-coding/](https://dusty.phillips.codes/2020/02/15/on-voice-coding/)
Sprechen ist das neue Klicken

Dr. Wolfram Wingerath, Michaela Gebauer

Für die Bedienung des Computers brauchte man viele Jahre Maus und Tastatur – heute kann man mit Sprache, Gestik und Mimik sogar programmieren.
Closing Recommendations

- **Keep it simple**: Prioritize ease-of-use over efficiency at the start (in particular: get used to an existing grammar before optimizing it)

- **Keep it reasonable**: Try to find use cases that make sense for you (e.g.: I’m not giving this talk handsfree, since I can use my index finger)

- **Keep it in mind**: Handsfree coding might save you one day (revisit this talk when you struggle with RSI, broken hand, etc.)
Thanks! So What Now?

Slack
talonvoice.slack.com

Join the community!

Subscribe to the mailing list!

GI Initiative
handsfree-coding.gi.de

Try out handsfree coding!

Patreon
patreon.com/lunixbochs

Support Talon Development!

Slides Available at https://wolle.science

Wolfram Wingerath   wolle@uol.de