4902 Project Deployment Artifacts

Kennedy Boynton, Jacob Franks, Brett Berglund, Kyle Minter

1. Overview:

The goal of this project is to give persons with visual impairments an accessible experience while not interfering with the workflow of people without any disabilities. Programming would be more convenient for everyone thanks to the extension's accessibility tools, which will enhance the whole experience. The most crucial accessibility elements for those with visual impairments will be included in the extension, such as:

* Outlining the scope of each individual line of code.
* Line-highlighting in a variety of styles and colors.
* Voice recognition enables time-saving code snippet installation as well as quick and simple accessibility option modifications.
* Filter intellisense suggestions.

1. Assumptions:

The user will have VScode and a screen reader (NVDA, JAWS, or VoiceOver)

* The user will have a Lego Mindstorm EV3, Spike Prime, or Thinker robot.
* The user will have a microphone and the system has a sound card.

1. Dependencies:

Node Dependencies:

"devDependencies": {

"@electron/rebuild": "^3.2.13",

"@types/bonjour": "^3.5.13",

"@types/glob": "^7.1.3",

"@types/midi": "^2.0.3",

"@types/mocha": "^8.2.2",

"@types/node": "16.x",

"@types/serialport": "^8.0.2",

"@types/ssh2": "^1.11.19",

"@types/ssh2-streams": "^0.1.12",

"@types/temp": "^0.9.4",

"@types/vscode": "^1.86.0",

"@types/ws": "^6.0.0",

"@types/zen-observable": "^0.8.7",

"@typescript-eslint/eslint-plugin": "^4.26.0",

"@typescript-eslint/parser": "^4.26.0",

"@vscode/test-electron": "^2.3.9",

"eslint": "^7.27.0",

"glob": "^7.1.7",

"mocha": "^10.0.0",

"prettier": "3.1.0",

"typescript": "^4.3.2"

},

"dependencies": {

"bonjour": "^3.5.0",

"compare-versions": "^6.1.0",

"dbus-next": "^0.10.2",

"duplexer": "^0.1.2",

"electron": "^24.2.0",

"electron-rebuild": "^3.2.9",

"jzz": "^1.7.4",

"say": "file:./dependencies/say",

"serialport": "^11.0.0",

"server": "^1",

"speech-synthesis": "^0.3.2",

"split": "^1.0.1",

"ssh2": "^1.15.0",

"ssh2-streams": "^0.4.10",

"standardized-audio-context": "^25.3.58",

"temp": "^0.9.4",

"tone": "^14.7.77",

"vscode-debugadapter": "^1.51.0",

"vscode-languageclient": "^5.1.0-next.9",

"web-audio-api": "^0.2.2",

"ws": "^6.2.2",

"zen-observable": "^0.10.0",

"zod": "^3.24.2"

}

Python Dependencies:

certifi==2022.12.7

charset-normalizer==3.1.0

click==8.1.3

colorama==0.4.6

idna==3.4

joblib==1.2.0

nltk==3.8.1

PyAudio==0.2.13

regex==2023.3.23

requests==2.28.2

SpeechRecognition==3.10.0

tqdm==4.65.0

urllib3==1.26.15

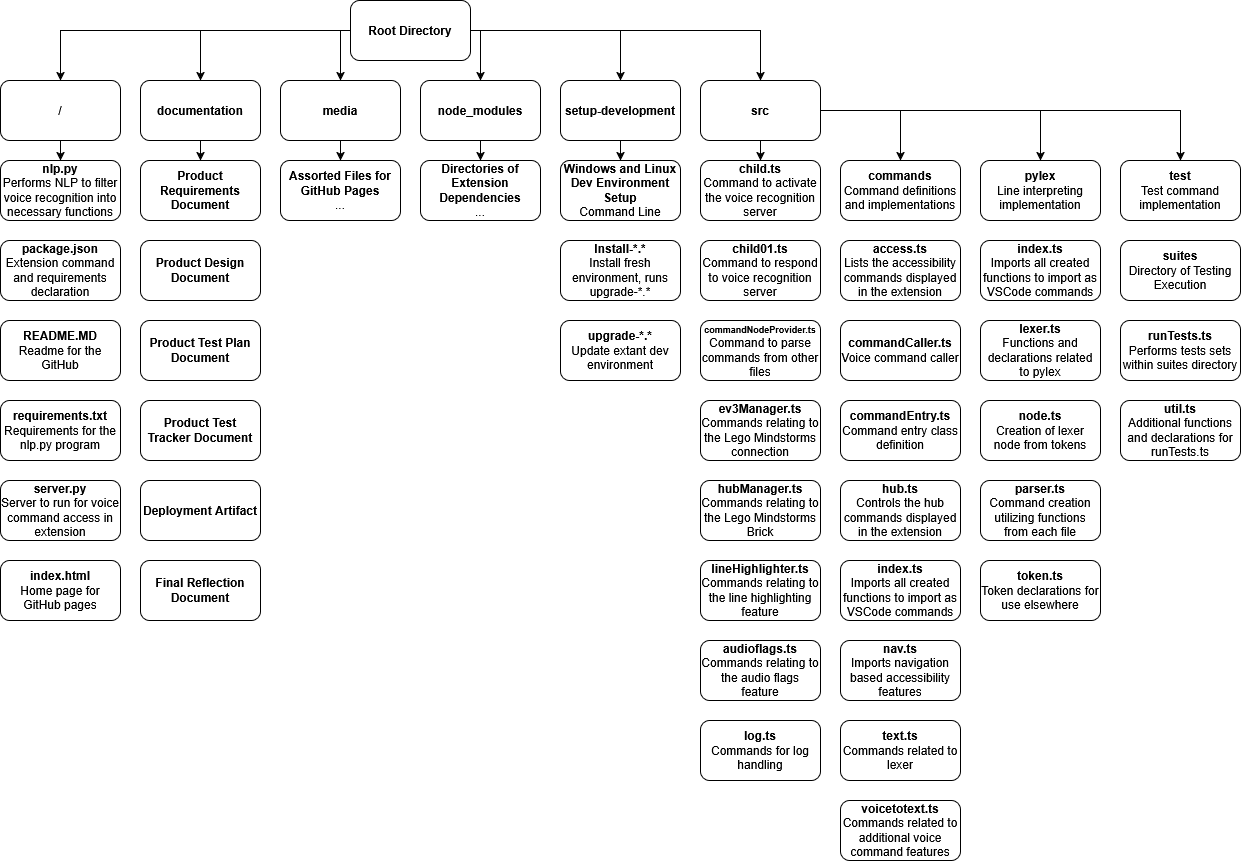
1. Constraints:

A constraint we face is when the user becomes disoriented within the editor, rendering them unable to navigate to their intended location. To address this issue, our mitigation plan involves implementing specific key bindings or commands that enable users to quickly access a designated home page for the extension. This home page will serve as a central hub from which users can easily return to their desired location within the editor, ensuring a seamless and efficient workflow. By providing these intuitive key bindings and a dedicated home page, we aim to empower users and alleviate the frustration caused by accidental disorientation, ultimately enhancing their overall experience with the extension.

In addition to the bloated workspace, another addition has been the audio flag feature. The user can mark lines with flags associated with a sound, and flags can be easily navigated to help improve the programming experience. This solution improves the visibility for specific lines of code with the visual flag and provided line-highlighter. In addition, audio queues can be played at any time which allows the user to play sounds when it will not conflict with other audio such as the screen reader.

Another constraint we face is the realization that VScode's compatibility with public screen readers falls short of its advertised capabilities. This shortfall becomes apparent when VScode incorrectly reads lines or fails to provide accurate information to the user. To tackle this challenge, our mitigation plan involves implementing specific functions within the extension that optimize content specifically for the user's chosen screen reader. By tailoring the output to suit the screen reader's requirements, we aim to enhance the accuracy and reliability of the information conveyed, ultimately ensuring a smoother and more accessible experience for users relying on screen readers while utilizing the extension.

1. Description of Deployment Artifacts:



1. Date Creation:

N/A

1. Admin Credentials:

N/A

1. Deployment Process:

Developer’s Install Guide:

Automated Install Script

Windows 10 or 11

1. Download [install-windows.ps1](../blob/master/setup-development/windows/install-windows.ps1)
2. Run `./install-windows.ps1` in PowerShell as a normal user, and accept any UAC prompts that pop up. The installation should take around 8 minutes. If an installer doesn't pop up, don't be alarmed.

Ubuntu, Linux Mint, Arch Linux, Manjaro

1. Download [install-linux.sh](../blob/master/setup-development/linux/linux-install.sh)
2. Run `install-linux.sh` as a normal user. It will use sudo when elevation is required. The installation should be done in 3 minutes.

Manual Install

Windows 10 or 11

1. Open Windows Powershell (Win+R, “powershell”, Enter).
2. Install/update Git for Windows.

`winget install Git.Git`

1. Open VS Code, and go to ‘Help’ > ‘About’ in the top toolbar.
2. Install the version of NodeJS.LTS corresponding to your Visual Studio Code version. This can be found in ‘Help’ > ‘About’ in the toolbar.

`winget install OpenJS.NodeJS.LTS --version <NodeJS version>`

This prevents a NODE\_MODULE\_VERSION conflict between Mind Reader and VSCode

1. Use node package manager to install ‘electron-rebuild’ and ‘vsce’

`npm install -g electron-rebuild vsce`

1. Navigate to your git directory, or create one. The following is recommend: ‘C:/Users/<username>/git’; i.e. ‘C:/Users/example/git’
2. Clone the repository using.

`git clone https://github.com/KyleMinter/Mind-Reader-Team-KJBK.git`

1. Install dependencies.

`npm install`

1. Rebuild Electron using the Electron Version from the About page.

`electron-rebuild --version <Electron version>`

This will rebuild the `serial\_port` Node Native Module. Without this module, Mind Reader will fail to register its commands.

1. Repeat steps 5-8 each time Visual Studio Code updates.

Extension Packaging Guide:

1. Open Windows Powershell (Win+R, “powershell”, Enter).
2. Globally install ‘@vscode/vsce’ using npm.

`npm install -g @vscode/vsce`

1. Navigate to the git repository that was cloned in the Developer’s Install Guide
2. Use ‘vsce’ to package the extension.

`vsce package --follow-symlinks`

User Installation Guide:

1. Download the VSIX file.

https://github.com/KyleMinter/Mind-Reader-Team-KJBK/releases/latest

1. Open VS Code and navigate to the Extensions tab.

‘View’ > ‘Extensions’

1. Click the 3 dot “...” icon located at the top right corner of the Extension tab.
2. Click “Install from VSIX…”
3. Using the file explorer, navigate to the downloaded VSIX file Step 1.
4. Click on the VSIX file to select it and then click install.