**Keyboard Scratch Doc**

**Competitor**

1. PRC Unity / MinSpeak: The PRC Unity / MinSpeak system uses a semantic compaction approach with icon-based sequences to generate text or speech output. The keyboard interface is designed around the selection of icons rather than traditional alphanumeric characters.
2. Tobii Dynavox Snap Core First: Snap Core First offers a symbol-based communication interface that provides a grid of symbols for users to select and form words or phrases. Although it doesn't have a traditional keyboard layout, the system is designed to be adaptable to various user needs.
3. Tobii Dynavox Communicator 5: The Communicator 5 software offers a variety of keyboard layouts, including QWERTY, ABC, and frequency-ordered layouts. The system also provides word and phrase prediction to facilitate faster text composition.
4. Smartbox Grid 3: Smartbox Grid 3 offers a range of keyboard layouts, such as QWERTY, ABC, and frequency-ordered layouts. The software includes word prediction to help users compose text more efficiently.
5. Avaz AAC: Avaz offers a picture-based communication interface, along with a traditional keyboard layout with word prediction to improve text composition efficiency.
6. Proloquo2Go: Proloquo2Go is a symbol-based AAC app that also provides a customizable QWERTY keyboard interface with word prediction for faster text composition.
7. Proloquo4Text: This text-based AAC solution offers a QWERTY keyboard layout with advanced word prediction and the ability to create custom keyboard shortcuts for frequently used phrases.
8. TouchChat HD: TouchChat HD provides a variety of keyboard layouts, including QWERTY and ABC, and comes with word prediction to support faster text composition.
9. LAMP Words for Life: This AAC app uses a variation of the Unity language system, with a grid-based keyboard interface focusing on icon selection instead of traditional alphanumeric characters.
10. GoTalk NOW: GoTalk NOW is primarily a symbol-based communication app, but it also provides a standard QWERTY keyboard layout with word prediction to facilitate text composition.
11. Predictable: Predictable is a text-based AAC app that offers a QWERTY keyboard layout, advanced word and phrase prediction, and customizable keyboard shortcuts.
12. CoughDrop: CoughDrop is a cloud-based AAC solution that supports a variety of keyboard layouts, including QWERTY, ABC, and customizable options, as well as word prediction.
13. Boardmaker: Boardmaker offers a grid-based keyboard interface, primarily focusing on symbol selection for communication. However, it does provide a QWERTY keyboard layout for text composition.
14. Speak for Yourself: This AAC app uses a grid-based keyboard interface with a focus on icon-based selection for communication, similar to the Unity language system.
15. Clicker Communicator: Clicker Communicator provides customizable keyboard layouts, including QWERTY, ABC, and frequency-ordered options, along with word prediction for efficient text composition.
16. Voice4U: Voice4U is primarily a symbol-based AAC app, but it also offers a QWERTY keyboard layout for users who prefer traditional text composition.
17. Speech Assistant AAC: This text-based AAC app provides a QWERTY keyboard layout with customizable keys and word prediction capabilities to support efficient text composition.
18. Chatable: Chatable is a symbol-based communication app that also offers a QWERTY keyboard layout with word prediction to help users compose text more quickly.
19. LetMeTalk: LetMeTalk is a picture-based AAC app that also provides a standard QWERTY keyboard layout for users who prefer text composition.

**Some existing ideas:**

* P300 Spelling Interface: Existing System: The P300 spelling interface uses a matrix of letters and symbols that flash sequentially. The system detects the P300 event-related potential in the user's EEG when they focus on the desired character. Evaluation: P300 spelling interfaces can be efficient due to their ability to utilize BCI input, making them suitable for users with limited motor function. However, the flashing matrix can be visually fatiguing, and typing speed may be limited by the speed at which the matrix flashes. Suggested Improvement: Implement an adaptive flashing speed, adjusting the rate based on the user's proficiency to reduce fatigue and increase typing speed.
* Frequency-based Layout: Existing System: This layout arranges characters based on their frequency of use in the language, with the most frequently used characters placed within the central region for easy access. Evaluation: The frequency-based layout minimizes the distance between frequently used characters, reducing the time it takes for users to input text. However, the layout may be less intuitive for new users, leading to a steeper learning curve. Suggested Improvement: Introduce customizable layouts, allowing users to modify the interface based on their preferences and communication needs.
* Predictive Text Interface: Existing System: This interface design incorporates a predictive text algorithm that suggests word completions based on the user's input, allowing users to select the desired word or character from a limited set of options. Evaluation: Predictive text interfaces reduce the number of selections needed to compose a message, making it easier and faster for users with limited motor function to communicate. However, predictive text algorithms may not always provide accurate suggestions, leading to potential frustration for users. Suggested Improvement: Enhance the predictive text algorithm's accuracy through machine learning techniques and allow users to provide feedback on suggested words to improve the system's performance over time.
* Radial Keyboard Layout: Existing System: This innovative layout arranges characters in concentric circles or sectors, with frequently used characters in the center. Users can access characters by navigating through the circles or sectors with eye gaze or BCI input. Evaluation: The radial keyboard layout minimizes eye or BCI movement, allowing users to quickly access characters and potentially increasing typing speed. However, the unconventional layout may require a significant learning curve for new users. Suggested Improvement: Offer a tutorial or interactive learning module to help users become familiar with the radial keyboard layout more quickly.
* Morse Code Interface: Existing System: Users input Morse code characters (dots and dashes) using eye gaze or BCI, which are then translated into text by the system. Evaluation: Morse code interfaces require only two input options (dot and dash), making them potentially faster and easier to use for users with limited motor function. However, learning Morse code may present a steep learning curve for users unfamiliar with the system. Suggested Improvement: Implement a gamified learning module to teach users Morse code, making the learning process more engaging and enjoyable.
* Dvorak Simplified Keyboard: Existing System: Users select characters using the Dvorak layout, which is designed for efficiency by placing frequently used characters in easily accessible positions. Evaluation: The Dvorak layout can improve typing speed by reducing the distance between frequently used characters, making it easier for users to input text. However, the non-standard layout may require a significant learning curve for users accustomed to QWERTY or other layouts. Suggested Improvement: Provide an interactive tutorial or learning module that helps users become familiar with the Dvorak layout more quickly, facilitating a smoother transition from traditional layouts.
* Colemak Keyboard Layout: Existing System: Users select characters using the Colemak layout, which is designed to minimize finger movement and increase typing speed. Evaluation: The Colemak layout can be beneficial for users with limited motor function, as it requires less movement to input text than traditional keyboard layouts. However, similar to the Dvorak layout, the non-standard arrangement may require a steep learning curve for new users. Suggested Improvement: Offer customizable layouts that allow users to adjust the Colemak layout based on their preferences and communication needs, making it more accessible to a wider range of users.
* Dynamic Keyboard Layout: Existing System: The layout adapts in real-time based on user input, placing the most likely next characters in easily accessible positions. Evaluation: Dynamic keyboard layouts can increase typing speed by reducing the time it takes for users to find and select the next character in a word or phrase. However, the constantly changing layout may be confusing for some users, increasing cognitive load. Suggested Improvement: Introduce a user-controlled adaptive mode that allows users to choose when the layout updates, providing a balance between efficiency and user comfort.
* Icon-based Interface: Existing System: Users select pre-defined icons representing common phrases, which are then translated into text. Evaluation: Icon-based interfaces reduce the number of selections needed to communicate common phrases, which can save time and effort for users with limited motor function. However, the limited set of predefined phrases may not adequately capture the full range of communication needs for all users. Suggested Improvement: Allow users to create custom icons and phrases, enabling more personalized and versatile communication options.
* Hierarchical Menu Interface: Existing System: Users navigate through a series of nested menus organized by categories, such as consonants and vowels, to select characters. Evaluation: Hierarchical menu interfaces limit the number of on-screen elements, potentially reducing cognitive load and making it easier for users to find and select characters. However, navigating through multiple menus may be time-consuming and frustrating for some users. Suggested Improvement: Implement an adaptive menu system that learns from the user's input patterns, streamlining the menu navigation process by prioritizing frequently used characters or categories
* Region-Based Layout: Specification: Divide the augmented reality (AR) display into different regions, each representing a group of related characters or symbols (e.g., vowels, consonants, punctuation, emojis). Users can select a region with eye gaze or BCI, causing it to zoom in and reveal its contents for further selection. Prediction algorithms can suggest the most probable characters or emojis based on the current input.
* Layered Wheel Design: Specification: Arrange characters and symbols in concentric circles or sectors, similar to a wheel, with frequently used characters and emojis placed in the inner layers. Users can navigate through the layers and select characters with eye gaze or BCI input. The prediction algorithm suggests likely next characters or emojis, automatically bringing them into the center for easy selection.
* Temporal Input Interface: Specification: Implement a time-based input system, where users focus on a single point in the AR display to cycle through characters and emojis at a specific speed. Users confirm their selection using a blink or BCI input. Prediction algorithms can adapt the cycling speed and sequence based on the user's typing history and provide likely next characters or emojis.
* Spatial Pattern Interface: Specification: Arrange characters and emojis in a grid or pattern that leverages spatial memory to facilitate faster selection. Users can navigate the grid using eye gaze or BCI input, and the prediction algorithm dynamically rearranges the grid based on the user's typing history to bring frequently used characters or emojis closer to the center.
* Gesture-Based Input: Specification: Users make specific eye gaze or BCI gestures to input alphanumeric characters and emojis. The AR display provides visual feedback for the gesture being made. The prediction algorithm offers suggestions based on the user's input history and adjusts the gesture recognition sensitivity to minimize errors.
* Semantic Emoji Influence: Specification: Users can select emojis with specific meanings or emotions to influence the prediction algorithm. For example, selecting a laughing emoji may prioritize words related to humor or happiness. The AR display highlights these contextually relevant suggestions for easy selection.
* Adaptive Zoom Interface: Specification: Characters and emojis are organized in a hierarchical structure, with the AR display zooming in on specific areas as users focus on them. The prediction algorithm adapts the zoom levels and selection areas based on the user's input history and current context, making frequently used characters and emojis more accessible.
* Morse Code Hybrid: Specification: Users input Morse code characters (dots and dashes) with eye gaze or BCI, which are then translated into text or emojis by the system. The prediction algorithm learns from the user's Morse code input patterns and suggests likely next characters or emojis, speeding up the input process.
* Phonetic Entry Interface: Specification: Users input phonetic representations of words or phrases using eye gaze or BCI. The prediction algorithm converts the phonetic input into standard text and suggests likely next characters or emojis based on the phonetic context.
* Customizable Visual Encoding: Specification: Users create custom visual encodings for frequently used characters, words, or phrases, which are then displayed on the AR interface. The prediction algorithm learns these custom encodings and suggests them alongside standard characters and emojis, making it easier for users to input their preferred content.
* T6 Interface: Specification: Adapt the T6 (Text-On-Six-Keys) input method, originally developed for numeric keypads, for eye gaze or BCI users. The AR display is divided into six zones, each containing a subset of characters and emojis. Users select a zone and then a specific character within that zone using eye gaze or BCI input. The prediction algorithm suggests likely next characters or emojis based on the current input and user's typing history.
* T9 Interface: Specification: Implement the T9 (Text-On-9-Keys) input method, traditionally used on mobile phones with a numeric keypad, for eye gaze or BCI users. The AR display is divided into nine zones, each representing a group of characters and emojis. Users input characters by selecting a zone multiple times according to the position of the desired character within the group. The prediction algorithm offers likely next characters or emojis based on the user's input history and current context.
* Chorded Keyboard Interface: Specification: Adapt the chorded keyboard concept, where users input characters by pressing multiple keys simultaneously, for eye gaze or BCI users. The AR display presents a small number of zones, and users select characters or emojis by "chording" multiple zones together. The prediction algorithm learns from the user's chording patterns and suggests likely next characters or emojis.
* Stenography-Based Input: Specification: Incorporate stenography principles, where users input phonetic syllables or combinations of characters simultaneously, into an AR interface for eye gaze or BCI users. The display is divided into zones representing phonetic components, and users select combinations of zones to create characters, words, or phrases. The prediction algorithm suggests likely next characters or emojis based on the phonetic context.
* Swype-Like Interface: Specification: Adapt the Swype input method, where users input text by "swiping" across a virtual keyboard, for eye gaze or BCI users. The AR display presents a grid of characters and emojis, and users select characters by focusing on the starting character and then "swiping" through subsequent characters in a continuous motion. The prediction algorithm offers suggestions based on the user's swiping patterns and current input.