ZENDALONA
VISION BEYOND SIGHT

Project List

Or any new relevant proposals that come with good ideas or games

Project Title: Accessible-Bluff Bot - Enhancing AI Opponents for Inclusive Card Gaming

What is Accessible-Bluff

Accessible-Bluff is a web-based card game designed with accessibility in mind, ensuring that all actions and information are conveyed audibly using a screen reader. It mimics a physical card game with real-life experiences by enabling teams to establish private rooms for remote collaboration and gaming.

How to play the Bluff game?

Like a regular card game, a group of players join a virtual room and shuffle the cards. The game begins with the first player placing one or more cards by either telling the truth or lying. For instance, if the player places “the king”, he/she can claim it as “a queen”, “a jack”, or “any other card”, or truthfully state it as “a king” itself. Other players can challenge if they suspect deception/cheat. If a player falsely claims “a queen” as “a king” and is challenged, the player receives a penalty, while the challenger earns an extra chance. However, if the challenger is wrong, he/she receives the penalty. Players can choose to “Pass” if they don’t want to participate in the round, but they can still “Raise” a challenge. A round concludes only when all players “Pass”, or someone “Raise” the challenge. The winner is the player who clears his/her cards first.

Project Details
Project page: https://zendalona.com/accessible-bluff/
Game URL (Project): bluff.zendalona.com
Bluff GitHub repository: https://github.com/zendalona/accessible-bluff

This project is to develop an AI bot for the Accessible-Bluff card game, ensuring it is fully accessible to visually impaired players. The bot should be one of the players. The bot should support multiple difficulty levels, allowing players of different skill levels to enjoy the game. The design should be modular, enabling future improvements and enhancements. The bot should be seamlessly integrated with the existing Accessible-Bluff game environment and its actions must be fully conveyed via screen readers.

Expected Output

1. **AI Bot for Accessible-Bluff**

   1.1. On the homepage, players can select the number of bots and their difficulty level while creating a game room. Once the game starts, bots will be displayed with names like Bot1, Bot2, etc., alongside human players, according to the user's selection on the game page.

   1.2. A fully functional AI bot capable of playing the Accessible-Bluff card game, mimicking a real human player throughout the game.

   1.3. Multiple difficulty levels cater to different player skill levels, including the following.

      a. Beginner (Easy Mode) – The AI plays with basic strategies, making predictable moves and occasionally bluffing. This level helps new players understand the game mechanics.

b. Intermediate (Normal Mode) – The AI uses a mix of strategy and deception, analyzing previous moves to make more competitive decisions. It bluffs more effectively and detects bluffs with moderate accuracy.

c. Advanced (Hard Mode) – The AI employs complex strategies, adaptive learning, and advanced bluff detection. It bluffs unpredictably and adjusts its playstyle based on opponents' behaviour, making it challenging for experienced players.

2. Accessibility Features

Seamless integration with screen readers, ensuring all bot actions (bluffing with selected cards and bluff text, challenging opponents' actions) are audibly conveyed.

3. Modular Design & Integration

A flexible and expandable bot architecture for future improvements.

Seamless integration with the Accessible-Bluff game environment, preserving existing accessibility and gaming logic.

4. Comprehensive Documentation

Detailed documentation covering the bot's architecture, algorithms, and usage.

Skills required/preferred Bot, NodeJs, Javascript,
Expected size of the project: 175 hours
Difficulty: High
Possible mentors: Nalin Sathyan

Project Title: AI-Powered Agent to provide support for Zendalona products

Project Description:
Develop an AI-powered Agent that provides information and support related to Zendalona products and assistive technologies for visually impaired individuals. This project aims to enhance the agent’s capabilities by integrating advanced Natural Language Processing (NLP) techniques, improving accessibility features, and expanding its knowledge base to better serve users with visual impairments.

Project Goals:

1. **Advanced NLP Integration:**
    - Implement state-of-the-art NLP models (e.g., BERT, GPT) to improve the AI Agents understanding of user queries.
    - Enhance the Agents ability to handle complex, multi-turn conversations and provide contextually relevant responses.
2. **Accessibility Enhancements:**
    - Ensure the Agent is compatible with screen readers and other assistive technologies commonly used by visually impaired individuals.
    - Implement text-to-speech (TTS) and speech-to-text (STT) functionalities to facilitate seamless communication.
3. **Knowledge Base Expansion:**
    - Curate and integrate a comprehensive database of Zendalona, including detailed descriptions, usage instructions, and user reviews.
    - Enable the chatbot to provide personalized recommendations based on user preferences and needs.
4. **User Experience Improvements:**
    - Conduct user testing with visually impaired individuals to gather feedback and identify areas for improvement.
    - An intuitive and user-friendly user experience for all types of queries.

Expected Outcomes:

- An intelligent and responsive AI Agent capable of understanding and addressing a wider range of user queries.
- Enhanced accessibility features that make the Agent accessible for visually impaired individuals.

- A richer knowledge base that provides users with detailed and accurate information about assistive technologies.
- A user-friendly interface that improves overall satisfaction and engagement.

Skills required/preferred : AI, NLP, Python
Expected size of project: 175 hours
Difficulty: High
Possible mentors: Sivasailam vellaisamy, Mukundhan

ZENDALONA
VISION BEYOND SIGHT

Project Title: Accessible World-Map-Explorer V2

Description: World-Map-Explorer is an inclusive and educational mapping tool designed for both visually impaired and sighted users. Powered by OpenStreetMap, it provides a seamless way to explore the world using keyboard navigation, audio feedback, and interactive features. Users can search for places, move through locations using arrow keys, get real-time border alerts, measure distances, and access detailed geographic information. Additional features include an adjustable pointer for exploring surroundings, inbound navigation to limit movement within a selected region, and a road distance calculator.

The Application is live at map.zendalona.com

User guide - https://map.zendalona.com/src/pages/user-guide/index.html

GitHub repository: https://github.com/zendalona/world-map-explorer-v2

Outcome: Migrate the project to Next.js for better performance, implement server-side functionalities, and add the following features.

1. **User Login** – Log in to save your preferences and create a personalized experience.
2. **Search History and Undo/Redo – Access a history of your** previous actions to revisit locations easily.
3. **Dynamic Routing** – Navigate to places directly using external URLs when applicable
4. **Bookmarks** – Add and manage bookmarks, saving them in the database for future reference.
5. **Settings Management** – Save and manage your preferences for a customized experience:
    a. Profile – Edit your email and personal data.
    b. Key binding – Customize keyboard shortcuts for efficient navigation.
    c. Bookmark Management – Edit and organize your bookmarks.
    d. Other Settings – Adjust cursor pointer size, angle, and related features.
6. **Path Navigation** – Navigate through roads, railways, rivers, and other paths using the cursor to inform the user about the places passing through.
7. **Water Sound Alerts** – Receive auditory cues when cursor enters water bodies for enhanced interaction.

Skills required/preferred : JavaScript, TypeScript, Next.js, Leaflet.js, and PostgreSQL.
Expected size of project: 175 hours
Difficulty: High
Possible mentors: K Sathyaseelan, Nalin Sathyan

Project Title: Accessible TuxType and TuxMath Enhancement

Description

During Google Summer of Code (GSoC) 2013, we made TuxType and TuxMath accessible for visually impaired users. However, the current official versions lack these accessibility features. It's crucial to release a new official version of TuxType and TuxMath on all platforms.

TuxType

The keyboard is the primary input device for visually impaired users, making keyboard practice essential for effective computer use.

TuxType is an educational typing game designed to improve users' typing skills in an engaging way. It consists of multiple game modes:

Fish Cascade: Tux must eat fish while avoiding those with letters by typing them to make them disappear.

Comet Zap: Tux defends cities from falling comets by typing the letters on them, inspired by the classic math game "Tux, of Math Command."

Lessons and Phrase Typing: Players practice typing through structured lessons and phrases, tracking accuracy and speed. Future improvements aim to make it easier for educators to customize content.

GitHub Repository: https://github.com/tux4kids/tuxtype

TuxMath

TuxMath transforms math practice into an exciting game where players help Tux defend planets from asteroids by solving arithmetic problems. It fosters quick thinking, hand-eye coordination, and strategic planning while reinforcing mathematical skills.

GitHub Repository: https://github.com/tux4kids/tuxmath/

t4common library

Tuxmath and Tuxtype are coded in C and utilize the SDL library for graphics. They share the t4common library. Accessibility features can be activated via a new menu or keyboard shortcuts.

GitHub Repository: https://github.com/tux4kids/t4kcommon

Expected Output:

1. Improve documentation around the build
2. Fix Mac build for m-X chips
3. Check and update dependencies, this will include updating the SDL library to version 2
4. Expose accessibility features in Tuxmath and Tuxtype
5. Fix existing bugs that include supporting it over other Linux distros like Fedora, Arch Linux
6. Make an official release for GNU/Linux
7. Make an official release for Microsoft Windows
8. Make an official release for MacOS
9. Prepare and release official packages or installers for all platforms.
10. Releasing snap packages for the Ubuntu store
11. Add Ukrainian translation
12. Add fractions question support: There is an issue around it: https://github.com/tux4kids/tuxmath/issues/38

Release TuxType and TuxMath with accessibility features for visually impaired users on all platforms without compromising the comfort of regular users.

Skills preferred: C, SDL

Expected time of project: 175 hours

Difficulty: Intermediate

Possible mentors: Deepak Agarwal, Mukundhan Annamalai

ZENDALONA
VISION BEYOND SIGHT

Project Title: Liblouis table editor - Enhancement

Description: “The Liblouis software suite provides an open-source braille translator, back-translator, and formatter for many languages and braille codes. It is a set of libraries designed for use in several applications, both free and commercial.” - https://liblouis.io/

Numerous software applications like NVDA, JAWS, BrlTTY, TalkBack, make use of liblouis. The total global user count is in millions. Many projects developed in Zendalona utilize liblouis, including IBus-Braille, Sharada-Braille-Writer, Braille-Translator-GUI, and the Braille-Translator Web version.

Outcome: Creating Liblouis translation tables is a laborious and time-consuming task. Both software developers and end users at the Braille printing press require a more convenient method for editing the liblouis table. Furthermore, in numerous languages, the editing process involves inputting Unicode code points instead of directly writing the letters, adding an extra layer of complexity to the table editing process.

Liblouis manual: https://liblouis.io/documentation/liblouis.html#How-to-Write-Translation-Tables

As part GSoC 2024, We have created a free and open-source Liblouis table editor compatible with GNU/Linux, Microsoft Windows, and macOS, fulfilling the following criteria:

1. Display characters as they are alongside their Unicode code points.
2. Provide a drop-down list for selecting opcodes.
3. Enable the insertion of operands through entries with validity checks.
4. Include an option to test forward and back-translation on the fly.
5. Allow users to view and modify the Unicode character equivalent corresponding to the written Unicode code point.
6. Indicators to ensure the user is aware of duplicate entries.
7. Should be accessible to visually challenged people

GitHub repository: https://github.com/zendalona/liblouis-table-editor

Here is the list of tasks we need to complete to release the project

1. Implement the Forward/Backward table testing feature
2. Fix bugs related to the toolbar
3. Fix bugs

4. Showcase the project internally within the organization
5. Do internal testing and fix bugs
6. Create a package for GNU/Linux
7. Develop an installer for Windows
8. Present the project to potential users
9. Facilitate testing with end users
10. Perform accessibility testing with blind users
11. Fix bugs and release the final version

Skills required/preferred Python, Qt, Liblouis.
Expected size of project: 175 hours
Difficulty: Intermediate
Possible mentors: Samuel Thibault, Nalin Sathyan

Project Title: Enhancing Accessibility in the Ubuntu Installer for Visually Impaired Users to Support Accessible-Coconut 24.04

Description:
The Ubuntu installer is a critical tool for users to set up their systems, but it currently lacks sufficient accessibility features for visually impaired users. This issue directly impacts projects like **Accessible-Coconut**, a specialized Linux distribution developed by **Zendalona** over the past decade. Zendalona has been releasing Accessible-Coconut every two years, with the upcoming **Accessible-Coconut 24.04** planned to be based on **Ubuntu MATE 24.04**. Improving the accessibility of the Ubuntu installer is essential to ensure a smooth and inclusive experience for users of Accessible-Coconut.

This project aims to address the accessibility shortcomings of the Ubuntu installer(Ubuntu Desktop Provision) by improving compatibility with screen readers like Orca, enhancing keyboard navigation, and providing clear audio feedback. These improvements will not only benefit Ubuntu users but also directly support the efforts of **Zendalona**, the developer behind Accessible-Coconut, in releasing a fully accessible 24.04 version.

Ubuntu-Desktop-Provision GitHub repository:
https://github.com/canonical/ubuntu-desktop-provision

Objectives:

1. **Improve Orca Screen Reader Integration:**
   - Investigate and fix issues with Orca compatibility in the Ubuntu installer, ensuring that all installer components are properly labelled and navigable using the screen reader.
   - Collaborate with the Orca development team to address any upstream issues affecting the installer.
2. **Enhance Keyboard Navigation:**
   - Ensure that all installer elements can be accessed and interacted with using only the keyboard, as many visually impaired users rely on keyboard navigation instead of a mouse.
3. **Provide Clear Audio Feedback:**

   - Implement audio cues and feedback for critical actions (e.g., selecting a disk, confirming installation) to guide visually impaired users through the installation process.
4. **Simplify and Streamline the User Interface:**
   - Work with the Ubuntu design team to simplify the installer interface, making it easier for screen readers to interpret and for users to understand.
5. **Support Accessible-Coconut 24.04:**
   - Collaborate with **Zendalona**, who has been releasing Accessible-Coconut every two years for the past decade, to ensure that the improvements align with the needs of Accessible-Coconut users.
   - Test the installer changes in the context of Ubuntu MATE 24.04, which is the base for Accessible-Coconut 24.04.
6. **Documentation and Testing:**
   - Create detailed documentation for visually impaired users on how to use the improved installer.
   - Conduct usability testing with visually impaired users, including those from the Accessible-Coconut community, to gather feedback and ensure the changes meet their needs.

Expected Outcomes:

- A more accessible Ubuntu installer that provides a seamless experience for visually impaired users, including those using Accessible-Coconut.
- Improved compatibility with Orca and other screen readers.
- Clearer navigation and feedback mechanisms for users relying on assistive technologies.
- Comprehensive documentation and testing reports to guide future accessibility improvements.
- Direct support for **Zendalona's** efforts to release **Accessible-Coconut 24.04** based on Ubuntu MATE 24.04, continuing a decade-long tradition of providing accessible Linux solutions.

Skills Required:

- Familiarity with Linux systems and Ubuntu.

- Experience with accessibility tools like Orca and knowledge of accessibility standards (e.g., WCAG).
- Proficiency in Python (used in the Ubuntu desktop provisioner) and GTK (for UI improvements).
- Strong communication skills to collaborate with the Ubuntu community, Zendalona, and visually impaired users.

Related Links:

- Ubuntu Discourse Thread on Installer Accessibility
- Launchpad Bug Report on Accessibility Issues
- GitHub Issue on Installer Accessibility
- Ubuntu MATE Community Discussion on Orca Improvements

Impact:
This project will significantly improve the inclusivity of the Ubuntu installer, making it easier for visually impaired users to install and use Ubuntu. It will also directly support **Zendalona's** efforts to release **Accessible-Coconut 24.04**, ensuring that the distribution remains a leading choice for visually impaired users. Additionally, the improvements will set a precedent for future accessibility enhancements across Ubuntu and other Linux distributions. By supporting Zendalona's decade-long commitment to accessibility, this project will contribute to a more inclusive open-source ecosystem.

Skills preferred: Flutter, Dart, Python

Expected time of project: 175 hours

Difficulty: High

Possible mentors: K Sathyaseelan, Akshay S Dinesh

Project Title: Linux-Intelligence-OCR-Solution(LIOS) - Enhancement

Description: LIOS is a free and open-source software designed for converting printed text into digital format using scanners or cameras. It is also capable of generating text from scanned images sourced from various formats, including PDFs, images, image folders, or screenshots. The program ensures complete accessibility for visually impaired users through a graphical user interface (GUI). Leveraging OCR engines like Tesseract and Cuneiform, LIOS facilitates the conversion of images to text.

Significantly, LIOS is included in the Debian repository and distinguishes itself as the exclusively accessible OCR user interface within the GNU/Linux environment. Furthermore, it has been developed inclusively, allowing many sighted individuals to also benefit from its features.

Github page: https://github.com/zendalona/lios
Debian repository: https://packages.debian.org/bookworm/lios

Outcome: We need the following

1. Make the UI simpler more easier by moving/removing many UI components
2. Make the UI more readable for Low vision by providing easy switchable themes
3. Make items in preferences more reasonably categorized
4. Fix scanner driver issues
5. Refine the code
6. Fix all reported bugs
7. Enhance detection of Tesseract data paths
8. Make dialog boxes foolproof
9. Release the new version as RPM package

Skills required/preferred Python, Gtk, Cairo
Expected size of project: 175 hours
Difficulty: Intermediate
Possible mentors: Samuel Thibault, Nalin Sathyan

Project Title: Accessible World-Map-Explorer Android version - zMap

Description: Discover the world with world map explorer, a web app made with inclusivity in mind. Navigate through keyboard or via search and explore the world with real time voice over. Search for countries, states, places, rivers or even historic monuments. Control the exploration area to stay within a region or make sure you don't get lost in the map with real time border crossing alert. Explore the surroundings with the adjustable pointer by changing its distance and angle. Start your adventure with World Map Explorer today and navigate the world with ease and confidence!
The web version is currently availablemap.zendalona.com

The Application is live at map.zendalona.com

User guide - https://map.zendalona.com/src/pages/user-guide/index.html

GitHub repository (Web version): https://github.com/zendalona/world-map-explorer-v2

Expected Outcome:

The android version can be a webview which should achieve the same experience as the web version by using gestures instead of keys of the keyboard. The features of the app are:

1. **Gesture-Assisted navigation:** The user should be able to use the app by using the gestures and no gesture should overlap with other gestures.
2. **Capturing User Location:** User should be able to start navigation from a place near user's actual location.
3. **Navigation Using Search:** User should be able to
4. discover and navigate to any places, rivers and historic monuments by using the search feature.
4. **Boundary setting for search and navigation:** The user should be able to select to select a place to learn more about it and restrict the navigation within it's boundaries.

5. **Adjustable pointer:** The user should be able to customize the distance and angle of a pointer to explore nearby locations with precision.
6. **Distance to Borders:** The user should instantly know the distance to the borders in the north, south, east and west direction.
7. **Zoom in:** The user should be able to zoom in zoom out.
8. **Distance Finder:** The user should be able to find the distance between any two places with ease.
9. **Map layout options:** The user should be able to switch between political and geographical views.
10. **Altitude awareness:** The user should instantly know the altitude of the current place.
11. **Managing notifications and calls**: Managing many activities that can occur while using the apps such as notifications, calls etc.

Skills preferred: Android, Java

Expected time of project: 350 hours

Difficulty: Intermediate

Possible mentors: Nalin Sathyan, Mukundhan Annamalai

Project Title: Maths-Tutor QT Version and Enhancement

Description: Accessible Maths-Tutor is a software tool that combines the enjoyment of gaming with the essential skill of mathematics. Imagine having a friendly mentor right on your computer screen, guiding you through math problems in a fun and interactive way. This is precisely what Maths Tutor offers a unique and engaging learning experience.

Information about the previous version(1): https://zendalona.com/accessible-maths-tutor/

In the previous year's GSoC, we migrated the project from GTK to Qt to address accessibility issues on Windows and macOS.

Repository link: https://github.com/zendalona/maths-tutor-v2/tree/development

Here is the list of tasks to be completed:

Bugs in Maths-Tutor V2:

- Upload option not implemented in Home
- Settings Option not implemented
- Even though we answered correctly, the animation for the wrong answer appears and announces the wrong answer and moves to the next question.
- In Subject Time the Questions are not fully visible
- Only for the first right answer the clap audio is heard but for the second right answer
- Note overrides over the subject menu
- The Bell Ringing option was not implemented
- The difficulty level not working as intended same questions are repeated in every level

Things Remaining to be implemented:

- Provide Speech Sound using Human Voice instead of Text-To-Speech Engine
- Intermediate score announcement for Appreciation
- Provide Answer clues in situations like the User taking too much time
- User Interface for creating Question File
- Clock Tick Sound for indicating Time consumption

- Introduce the "Assessment Mode" Checkbox to Enable/Disable exposing Score, Clock, etc.
- User Guide
- Themes for Low Vision - High Contrast, Low Contrast
- Lessen the Lock to make the User finish for the Next Lesson
- Increase sound playing Speed to Speech Rate
- Profile Preferences for Each Kid and Use their Name While Appreciating
- Pair Question in Percentage
- Dataset - Add number typing of Lakhs / Millions
- Dataset for teaching multiple ways to make the same Digit
- Fix announcement of Lakhs / Millions

Skills required/preferred: Python, QT
Expected size of project: 175 hours
Difficulty: Intermediate
Possible mentors: Sai Saravan, K Sathyaseelan

Project Title: Maths-Mantra Enhancement

Description: In GSoC 2024, we developed a comprehensive smartphone application incorporating all the features of the Math-Tutor computer version. Additionally, we integrated the functionalities outlined in the 'What We Need' section. The app utilizes various input methods, including touch screen, accelerometer, microphone, GPS, clock, and magnetometer, to create an immersive and interactive learning experience. To enhance user engagement and effectiveness, we implemented feedback mechanisms such as stereo sound, vibrations, and other sensory cues.

This year, we need to fix existing bugs and add the remaining features to complete the project.

GithHub repository: https://github.com/zendalona/Math-Mantra

Outcome: Include the following concepts in the application:

1. Utilize phone sensors and stereo sound to teach cardinal directions (North, East, West, South) through interactive methods.
2. Integrate lessons on Angle X, Y, Z, and incorporate simple distance/height calculations using trigonometric functions.
3. Enable user engagement by allowing kids to input answers through sound or clapping using the device's microphone.
4. Leverage GPS functionality to teach distance-related concepts, providing a real-world context for understanding spatial relationships.
5. Implement multi-touch functionality on the screen to teach number concepts and facilitate the drawing of various shapes such as triangles, rectangles, rounds, ovals, etc.

Skills required/preferred: Android, Java, UI development
Expected size of project: 350 hours
Difficulty: Intermediate
Possible mentors: Nalin Sathyan, Mukundhan