

## Kodename: An Overview by Miracle Okubor

The idea for this project was to create an application that can be used to teach children programming concepts without emphasis on language syntax. The instructive language is *Karel* which is an educative programming language used to create instructions that a robot called Karel uses to navigate, and solve problems within, its World. The Karel language is similar in structure and syntax to existing programming languages and so gives the young programmer an idea of what programming is like in general. However, since Karel, like any other programming language, requires valid syntax to create programs, there can be challenges with teaching young children the techniques of programming while avoiding memorizing syntax. During research, we looked at Scratch whose syntax is more conversational and organized into code pieces much like a jigsaw puzzle. Unlike Karel, the user needs no prior knowledge to put together “puzzles pieces” and then begin solving problems. Unfortunately Scratch syntax is substantially different from common programming languages.

Kodename uses the Karel programming language to teach the user programming and with its user interface, Kodename aims to eliminate the need for the user to focus on syntax. It combines and improves upon the best features of Karel and Scratch for the purposes of teaching children. It bridges the application gaps present in Karel and Scratch by both avoiding the need to understand language syntax while still creating a programming vocabulary that bears resemblance to other mainstream languages. It is designed to teach problem solving, condition and loop statements, modularity, code reusability, and other skills that can be easily applied for any other programming languages they may encounter in the future.

The application allows the user to create a World for the Karel robot based on specifications determined by the user. The Karel robot's World is a grid/maze-like environment that contains the Karel robot, walls that the robot cannot walk through, and beepers, which the robot can either pick up or set down. Once a World has been created, the user is presented with the main interface which has Code Blocks that contain Karel language instructions, a Simple View panel to show the Code Blocks and a World which is the World the user created at the start of the application. The user drags and drops Code Blocks into the Simple View; these Code Blocks are the chosen instructions that will help the user solve problems related to the World they have created. Once the user is done and they run their program they see the Karel robot move within the World and they can also see how each movement is related to each Code Block previously placed within the Simple View.

Other features of the Kodename application include:

- Creating custom functions that allow the user to divide their program and easily reuse these functions instead of repeating multiple lines of code.
- Editing and deleting custom functions.
- Removing Code Blocks from the Simple View.
- Clearing ALL the Code Blocks previously added to the Simple View.
- Step-by-Step execution of the users program.
- Continuous execution of the users program.
- Displaying status messages during execution.
- Saving a session, which can be returned to at another time.

The underlying operating environment for the Kodename application is JAVA, using features from the Java Swing library, such as lists, buttons, drag-and-drop, panels, etc, to build the user interface. The project implements the model-view-control architecture design pattern to control the flow of information between the user and the software. The application was designed for a classroom setting or personal learning and is not suitable for mobile devices such as smart phones, smart watches and tablets. It is suitable for desktops and laptops. The application does not require a connection to the internet to work.

The developing team was a 5-person group and I was an integral part of the user interface design team. The application has 5 basic instructions; Move, Turn Right, Turn Left, Pick Up Beeper, Put Down Beeper. Kodename has an Action panel which displays the Karel code pieces, which are represented as strings, and uses the Java DataTransfer packages drag and drop feature to move code pieces from this panel to the Simple View or Pseudocode panel. The Simple View is a Swing Jlist, which allowed a user to be able to insert or delete code pieces at any position in the Simple View. Dialogs were created to ensure the user chooses boolean statements to go with If-Else and numerical counts with For statements. These dialogs also appear when the user is creating custom functions. There is a World in which the Karel robot moves. This panel is designed with step-by-step and continuous execution buttons. As the user runs a program step-by-step, the line of code being executed at the moment is highlighted and a message is displayed in a text console below the World. The programming objective could be for Karel to pick up all the beepers in its World, put down beepers at certain locations or move in a circle a certain number of times. The user has free reign over the objective of the program they create.