NEA

[Introduction 1](#_Toc209599357)

[Aim 1](#_Toc209599358)

[Research 2](#_Toc209599359)

[Outlook 2](#_Toc209599360)

[Pre-requisites 2](#_Toc209599361)

[Features to include 2](#_Toc209599362)

[Partials solutions 2](#_Toc209599363)

[Libraries 3](#_Toc209599364)

[Stakeholders 3](#_Toc209599365)

[References: 3](#_Toc209599366)

Introduction:

Rust based block code editor: A coding editor that is written 100% in Rust. This project is an opportunity for me to learn not only how a coding editor works, but also why the best ones are the way they are, along with me expanding my understanding of coding languages by choosing a language that I am not close to mastering and allowing me to learn it through this project in able to expand my portfolio of coding languages for in the future. This editor aims to help coders who want to learn how to code as well as learning how to use Arduino and raspberry pi, it is its own stand-alone app that can provide a easy to use and intuitive interface to beginners to this area and allow them to learn through using the app.

## Aim

The overall aim of the project is to give me deeper understanding in how coding editors work and what is required to have a functioning editor. This project will also allow me to learn a new language to allow me to expand my portfolio of coding languages, which will benefit me in the future.

# Research

## Outlook

I believe this project will challenge me to work and understand how coding editors work and what components are needed within an editors to get them to work. This will allow me to use a computational approach to decompose the project down into approachable sections that I can work on and recombine to make a full working prototype. The project will require me to do deeper research into how individual component of a coding editor works and what each piece will need to be functional and how they will work together.

## Pre-requisites

For this project to be successful, I am going to need:

* A window library
* Access to filesystem to open/close/save files
* A way to compile code written to work for an Arduino

## Features to include

The features I would like to include in my project include:

* GUI for ease of use
* Inbuilt terminal to run/debug files using language servers on users’ machine (future)
* Debugging capabilities (future)
* Commands to have functionality
* Block based code

## Partials solutions

There are a few partial solution that are similar to what I am trying to make, these include:

* Zed, a rust based gui editor similar to vscode.

Zed Industries. (16 June 2025). *Zed*. [Online]. zed.dev. Last Updated: 10 September 2025. Available at: https://zed.dev/ [Accessed 18 September 2025].

* Helix, a text based coding editor written in rust, similar to vscode

Michael Murphy. (12 May 2021). *Helix*. [Online]. helix-editor.com. Last Updated: 18 June 2025. Available at: https://helix-editor.com/ [Accessed 18 September 2025].

* Blockly, a block based code editor written by google

Google. (13 May 2012). *Blockly*. [Online]. google.github.io/blockly. Last Updated: 28 August 2025. Available at: https://google.github.io/blockly/ [Accessed 21 September 2025].

All of these solutions use the Rust programming language, which is what I am also choosing to use for this project as it will help me to expand my knowledge of coding and allow me to have multiple options for later in life if I choose to code, allowing me to expand my breadth in coding to have multiple options for completing a project. Both solutions are text based, meaning that they only run in the terminal, whereas I would like to have this code editor as a GUI, so the interface is more useable to more people.

## Libraries

To make the program a GUI I have decided to look for multiple window libraries within the rust ecosystem. These libraries would allow me to produce a GUI that will serve as the basis for my code editor. The window library will need:

* to have the ability to create a menu to perform certain functions
* being able to accept keyboard input for the command mode
* Have the ability to create a area that serves as the editor where code is opened and edited

By searching for Rust Gui libraries, I have narrowed it down to:

* Egui, a window framework centred around simplicity and portability
* Slint, a rust native window library that is used a lot for embedded devices
* Gtk-rs, a rust port of the well know gtk framework, this allows for the power of the gtk framework to be used by rust

## Stakeholders

For my stakeholders I have chosen to contact the head of DT at my school, I have chosen to contact him as i feel this project

# References:

Zed Industries. (16 June 2025). *Zed*. [Online]. zed.dev. Last Updated: 10 September 2025. Available at: https://zed.dev/ [Accessed 18 September 2025].

Michael Murphy. (12 May 2021). *Helix*. [Online]. helix-editor.com. Last Updated: 18 June 2025. Available at: https://helix-editor.com/ [Accessed 18 September 2025].

Google. (13 May 2012). *Blockly*. [Online]. google.github.io/blockly. Last Updated: 28 August 2025. Available at: https://google.github.io/blockly/ [Accessed 21 September 2025].

What need doing:

* Email mr turner about stakeholders
* Speak to mr major about project as I am confused