Minecode

NEA Project 2022/23

Jake Gaunt

Contents

1 | Analysis

1.1 Problem Identification & Proposed Solution

Currently, I believe there is a problem with teaching children computer programming. In my opinion, learning to code is a skill that is becoming increasingly more important as computers become more powerful and more jobs open up in the technology sector. From my own experience, I have found that many of the tools that are designed to teach computer programming are either too expensive or too boring for kids to enjoy. I intend to create a solution to this problem by integrating gaming and programming together.

In this project, my aim is to produce a mod for a game which will allow users to learn the basics of computer programming by creating their own programs and running them ingame.

1.2 Stakeholders

My software will be designed for people who enjoy computers and playing games. Due to this, I have chosen the target audience to be secondary school students and teachers. Students are an appropriate choice of user since a large proportion of my user base is represented by children from ages 11-17. Teachers are the perfect target clientele because they usually teach classes of around 15-35 students.

To appeal to this audience, my software will be designed to be appealing to schools and children. I have selected suitable stakeholders that reflect the user demographic to give me feedback on my software. They are:

- Trisha Gates - Year 7 student at Winterbourne Academy
- Josh Parker - Year 9 student at Bradley Stoke Community School
- Ms Josephine - GCSE Computer Science teacher

Over the course of development of my software, I will interview my stakeholders to gain ideas and feedback to improve my project.

1.3 Research

1.3.1 Game & Programming Language

Choosing a suitable game

For my project I will need to choose a game which meets the following criteria:

- Appeals to a wide age range
- Easy to understand
- Freedom for users to create
- Pre-existing modding API

This is why I have picked Minecraft to be the game that I will create my project for. Minecraft is a sandbox video game which allows players to build their own world out of blocks. It is an easy to understand game which can appeal to people of all ages and it has a large modding community with many modding APIs.

Choosing a platform

Minecraft has many different versions for different platforms. However the 2 main ones are Bedrock Edition and Java Edition.

Bedrock Edition

Minecraft Bedrock is designed to work on multiple different platforms such as PC, Xbox and Mobile. However, it does not support 3rd party mods. Players can install “Add-Ons” from the marketplace but these usually cost the player money and are made by larger developers.

Java Edition

Minecraft Java is the original version of the game and it is only designed to work on PC. It is much easier to make mods for Java edition compared to Bedrock Edition.

I will be creating my mod for the Java Edition of the game. I will be choosing this version since it supports modding and has PC support which is the device my target user base will be using.

Choosing a suitable programming language

Since I have chosen the Minecraft Java Edition, Java is the choice for my programming language. Java will be an excellent language for me to use since it is object oriented and I have over 2 years of experience using it. Java is also great because it is a multiplatform language, it compiles to bytecode which is then executed on the Java Virtual Machine (JVM for short). This means I will not have to compile multiple versions of my program to work on different operating systems.

1.3.2 Existing Solutions

Code.org

Overview

Code.org is a free to use website that aims to teach kids computational thinking and programming at a very basic level. Code.org is popular among schools since it is easy to access - it can be completed in a web browser with no need to install any software. Code kingdoms is aimed at younger children, with the problems being easy to solve and boring for older users.

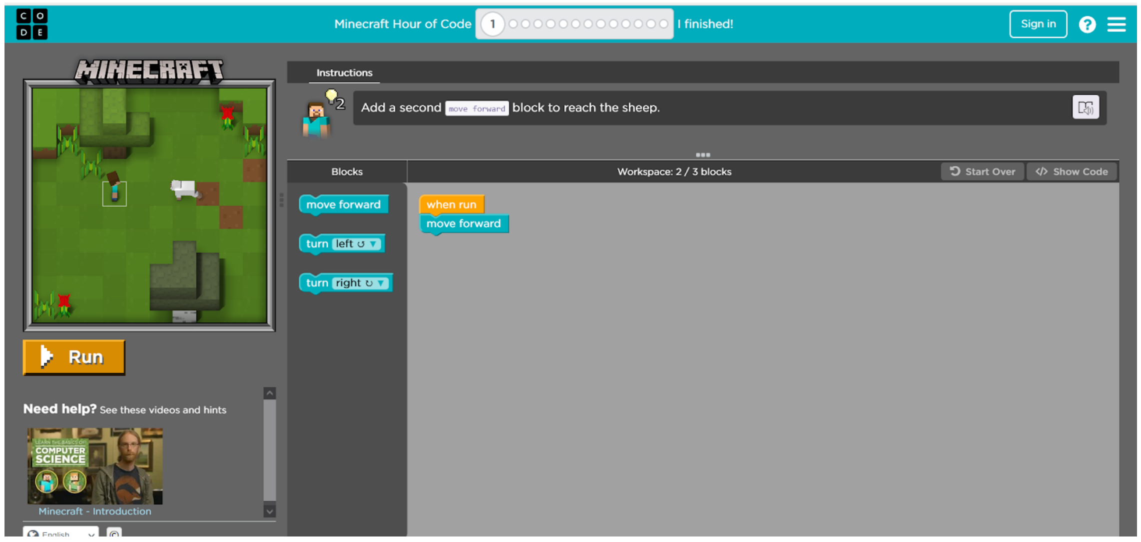
Benefits

- Simple drag and drop style makes learning easy
- Only need a web browser to use

Problems

- Problems too easy for older users
- 2D style of game quickly becomes boring

Content



Add a second move forward block to reach the sheep.

↑Code.org user interface

Code Kingdoms

Overview

Code kingdoms is a paid service that teaches children how to program using Minecraft. It offers users an online drag and drop interface where they can create code and run it on a private Minecraft server that is hosted by code kingdoms. Code kingdoms is beginner friendly but has options for older children who want a challenge.

Benefits

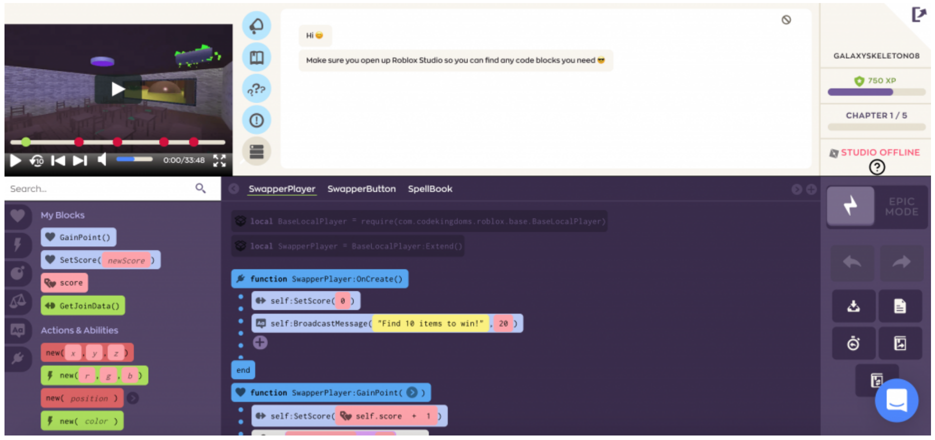
- Simple drag and drop style makes learning easy
- Only need a Minecraft account and web browser to get started
- Features more complex structures such as classes
- Drag and drop code becomes more similar to java or python for advanced users so transitioning from drag and drop to real code is easier

- Code is run on an online Minecraft[1] server which allows the user to play with friends

Problems

- Expensive
- Code cannot be used on other minecraft servers

Content

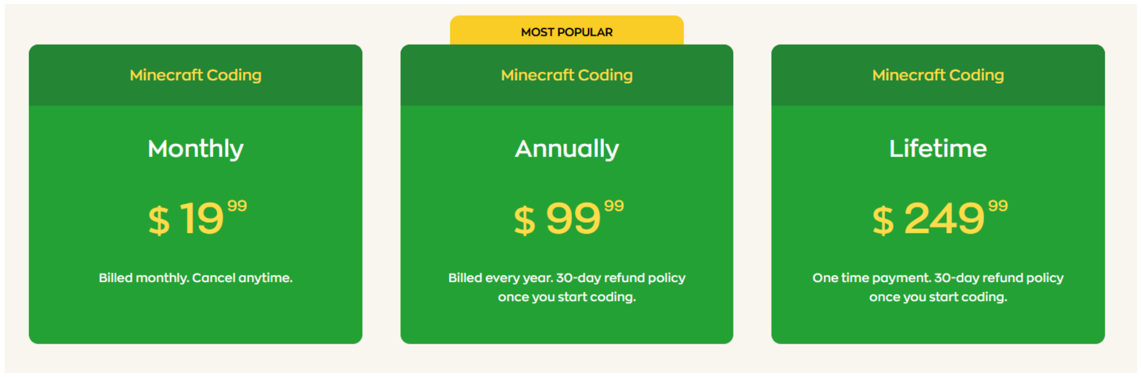


Hi 😊

Make sure you open up Roblox Studio so you can find any code blocks you need 😎

```
local BaseLocalPlayer = require(com.codekingdoms.roblox.base.BaseLocalPlayer)
local SwapperPlayer = BaseLocalPlayer:Extend()

function SwapperPlayer:OnCreate()
    self:SetScore( 0 )
    self:BroadcastMessage( "Find 10 items to win!" , 20 )
end
function SwapperPlayer:GainPoint( )
    self:SetScore( self.score + 1 )
```

↑ Code kingdoms user interface



Minecraft Coding	MOST POPULAR Minecraft Coding	Minecraft Coding
Monthly	Annually	Lifetime
$ 19 99	$ 99 99	$ 249 99
Billed monthly. Cancel anytime.	Billed every year. 30-day refund policy once you start coding.	One time payment. 30-day refund policy once you start coding.

↑ Code kingdoms pricing plan

Code Gym

Overview

Code Gym is an online learning platform that teaches students Java. Code Gym offers users lots of programming challenges that the user must complete in order to achieve “darkmatter” (points) so that the user can progress to harder levels and challenges. Code Gym's tasks can be completed in their basic online ide or by installing their plugin for Intellij IDEA and completing, submitting and testing tasks there. Intellij IDEA is a powerful ide for Java programming and helps the user to learn how to use and ide as well as learn to program. Code Gym's challenges cover a wide range of skill levels so that even experienced programmers can have fun.

Benefits

- Teaches user to use an ide
- Forums where users can see example solutions and ask for help from more experienced users
- Wide range of topics to interest all skill levels

Problems

- Expensive

1.3.3 Programming Languages

Criteria for a suitable programming language

For the needs of this project, I need a programming language that is easy to learn and simple to implement. For a language to be desirable for this problem, it must meet the following criteria:

Simple syntax - A simple syntax will make it easier for beginners to understand code

Interpreted - Since programs need to be modified quickly and regularly, a compiler is inappropriate

Easy to integrate with Java - There must be an easy way to integrate the language with Java since this is the language I have chosen to develop my software in

Due to these factors, I have chosen Lua to be the best option for the scripting system in my project.

Lua

Overview

Lua is a lightweight and fast scripting language which allows the user to create functional or procedural programs easily. Object oriented programming can also be implemented using tables (a data structure in Lua) although this is more complicated than in other programming languages. Lua has a simple syntax and is dynamically typed which makes it easy to learn. It features automatic memory management in the form of an incremental garbage collector that allows the developer to not have to worry about memory management which makes development simpler although it means that it lacks the level of control that languages such as C and C++ give the developer.

Why use Lua?

For my project I needed a programming language that would be easy for beginners to learn, compatible with Java (my programming language of choice and the language Minecraft[1] is written in) and fast enough to make the gameplay of the user unaffected. Lua is the obvious choice for this. At heart, Lua is designed to be a fast and embeddable scripting language. Some key features of Lua's design include:

- **Lua Interpreter** - Lua is an interpreted language, meaning that it does not need to be compiled before runtime. This makes it ideal for scripting since it allows the developer to change their code and then simply restart the program rather than having to recompile it. As there is no need to compile the code, the same .lua scripts can be used on different platforms which is useful since it can allow scripts to be shared between users even if they are on different operating systems. Lua is also regarded as the fastest interpreted scripting language.
- **Simple Syntax** - Lua's syntax is very easy to understand. It features only around 20 keywords, and is easily readable even for beginners.

- **Dynamic Typing** - Lua is dynamically typed which means the programmer does not need to specify the data type of a variable which makes scripts shorter and saves time for the programmer.
- **Garbage Collector** - Developers don't need to spend time implementing their own memory management system. This saves time and can help to optimise program's use of memory.
- **Small Footprint** - Lua can easily be added to a program without adding bloat. When compressed, the TAR archive containing all of Lua's source and documentation is only 353kB.
- **Well Documented API** - Lua has a well documented api which allows it to be integrated with other languages. Lua has been integrated with programs written in languages such as: C, C++, C#, Java.

1.3.4 Java Libraries

For my project I will be using several Java libraries created by other developers.

Gradle

To include dependencies in my project I will be using a build automation tool called Gradle. Gradle runs on the JVM (Java virtual machine) and allows developers to easily include libraries in their projects.

Luaj

Luaj is a Java library made to execute Lua code. It is built around the 5.2.x version of Lua and has these goals:

- Java-centric implementation of Lua vm built to leverage standard Java features.
- Lightweight, high performance execution of Lua.
- Multi-platform to be able to run on JME, JSE, or JEE environments.
- Complete set of libraries and tools for integration into real-world projects.

(For more information go to: github.com/luaj/luaj)

Luaj allows developers to create their own Lua libraries to integrate into their code. These libraries can then be imported and used in Lua scripts. Luaj also allows for Java methods to be called directly from Lua code.

Forge vs Fabric

I will need a Minecraft mod loader which will allow me to create mods for the game. There are 2 main mod loaders which will allow me to do this, Minecraft Forge and Fabric MC.

Forge

Forge is a reliable library for modding Minecraft Java Edition. It has been around for over 10 years and has support for Minecraft versions from 1.1 - 1.18 (latest version as of 15/05/22). Due to this, there are thousands of mods made to work with Forge for all versions of the game. I am very familiar with Forge as I have experience creating mods with it over the past 1.5 years. However, Forge has relatively bad performance.

Fabric

Fabric is a much newer modding library than forge. Fabric was created in 2018 and is focussed on having less boilerplate and better performance. For these reasons, many developers making mods for the more recent versions of the game choose to use Fabric. I have less experience working with Fabric but the API is simple and easy to learn.

I will be choosing to develop my mod with Fabric since I am developing for the latest version of Minecraft and I want my mod to have the best possible performance.

1.3.3 Initial Interviews

I will be conducting initial interviews with my stakeholders to understand what features should be included in my software. I have created a set of questions to ask each group of my stakeholders.

Student questions

1. Do you have previous experience with programming?
2. If so, what languages did you use and what would you rate your skills (1-10)?
3. Do you have previous experience with computer games?
4. If so, which games have you played and what did you enjoy about them?
5. What can you see yourself using this product for?
6. What are the features that you think must be included in the product?
7. What are the features that you think could be included in the product?
8. What are the features that you think should not be included in the product?

Teacher questions

1. Which programming languages do you have experience with and how much experience do you have with them (1-10)?

2. What would you say is the hardest aspect of teaching programming?
3. How effective do you think this product will be at teaching (1-10)?
4. Can you see yourself using this product in the classroom?
5. What are the features that you think must be included in the product?
6. What are the features that you think could be included in the product?
7. What are the features that you think should not be included in the product?

Initial Interview - Trisha Gates (Year 7 Student)

1. Do you have previous experience with programming?

“I have a little bit of experience with programming in primary school.”

2. If so, what languages did you use and what would you rate your skills (1-10)?

“I used scratch to make a few games and started to learn how to say hello in python. I would rate my skills 4/10”

3. Do you have previous experience with computer games?

“I sometimes play video games with my brother.”

4. If so, which games have you played and what did you enjoy about them?

“We usually play fortnite or terraria. I like the exploring aspect of the game and the dance moves.”

5. What can you see yourself using this product for?

“I want to play Minecraft and add cool new mobs. I hope I will be able to make mods like the ones I’ve seen on youtube”

6. What are the features that you think must be included in the product?

“I think there should be lots of tutorials that can show me how to make cool things”

7. What are the features that you think could be included in the product?

“I think that it would be good if you could include a way to make new mobs and blocks”

8. What are the features that you think should not be included in the product?

“I don’t think that it would be good if it was too complicated. I think you should not include anything too complicated”

Initial Interview - Josh Parker (Year 9 Student)

1. Do you have previous experience with programming?

“I’ve been learning programming in my computing class at the moment”

2. If so, what languages did you use and what would you rate your skills (1-10)?

“I’ve made a few simple games using python like blackjack. I would say that I am a 6/10”

3. Do you have previous experience with computer games?

“I love to play video games. I usually play for a few hours a night.”

4. If so, which games have you played and what did you enjoy about them?

“I mostly play Call of Duty and GTA. I like playing with my friends and unlocking new items.”

5. What can you see yourself using this product for?

“It seems like a cool idea since I can develop some useful skills while gaming. I can see myself making mods that will improve my gameplay skills on the game or automate things.”

6. What are the features that you think must be included in the product?

“I think there should be lots of ways to unlock different awards and achievements. There should also be a way that I can make mods that will help me win fights against my friend ”

7. What are the features that you think could be included in the product?

“There should definitely be the option to edit the mods from inside the game. It would be great to quickly change my program to enhance it.”

8. What are the features that you think should not be included in the product?

“I think that there is no need for a way to share scripts in a central place like a forum. I think this would overcomplicate things.”

Initial Interview - Ms Josephine (Teacher)

1. Which programming languages do you have experience with and how much experience do you have with them (1-10)?

"I have experience in Python, Java and C#. I teach python to my students and I would say that I am a 9/10. I did some Java and C# in university but I am a bit rusty now so I would say that I am a 7/10."

2. What would you say is the hardest aspect of teaching programming?

"I would say that the hardest aspect of teaching coding is getting people interested. Since most students are beginners we teach very simple stuff in lessons which is often boring for the students. I think that this is a problem because lots of kids find it boring and it puts them off from learning to code properly."

3. How effective do you think this product will be at teaching (1-10)?

"I think it will be very effective at teaching the basics. I would give it a 9/10."

4. Can you see yourself using this product in the classroom?

"I can see myself using this to teach. It would help to engage the students and teach them."

5. What are the features that you think must be included in the product?

"An easy way to install the software onto the computers. There should be lots of activities for lots of skill levels to interest all the kids in the class."

6. What are the features that you think could be included in the product?

"There could be a way for students to submit their work to the teacher"

7. What are the features that you think should not be included in the product?

"There should not be a way for the students to misuse the program to make inappropriate things."

1.4 Requirements and Limitations

1.4.1 Software & Hardware Requirements

My project requires Minecraft and Java to be installed on the system. The system must also meet the minimum software requirements for Minecraft:

OS - Windows (7 or higher), macOS (OS X Mavericks (10.9) or later) or Linux (Any modern 64-bit distros from 2014 onward)

CPU - Intel Core i3-3210 3.2 GHz/ AMD A8-7600 APU 3.1 GHz or equivalent

GPU - Nvidia GeForce 400 Series or AMD Radeon HD 7000 series with OpenGL 4.4

RAM - 4GB

HDD - At least 1GB

1.4.2 Features and Limitations

Based on my initial discussions with my stakeholders I have decided on a set of features and limitations for my software.

Features that will be included

- Tutorials & challenges
- Achievements
- Variety of skill levels
- Easy way to install
- Ingame script editor
- Script executing system
- Easy to understand UI

Features that could be included

- Way to submit work

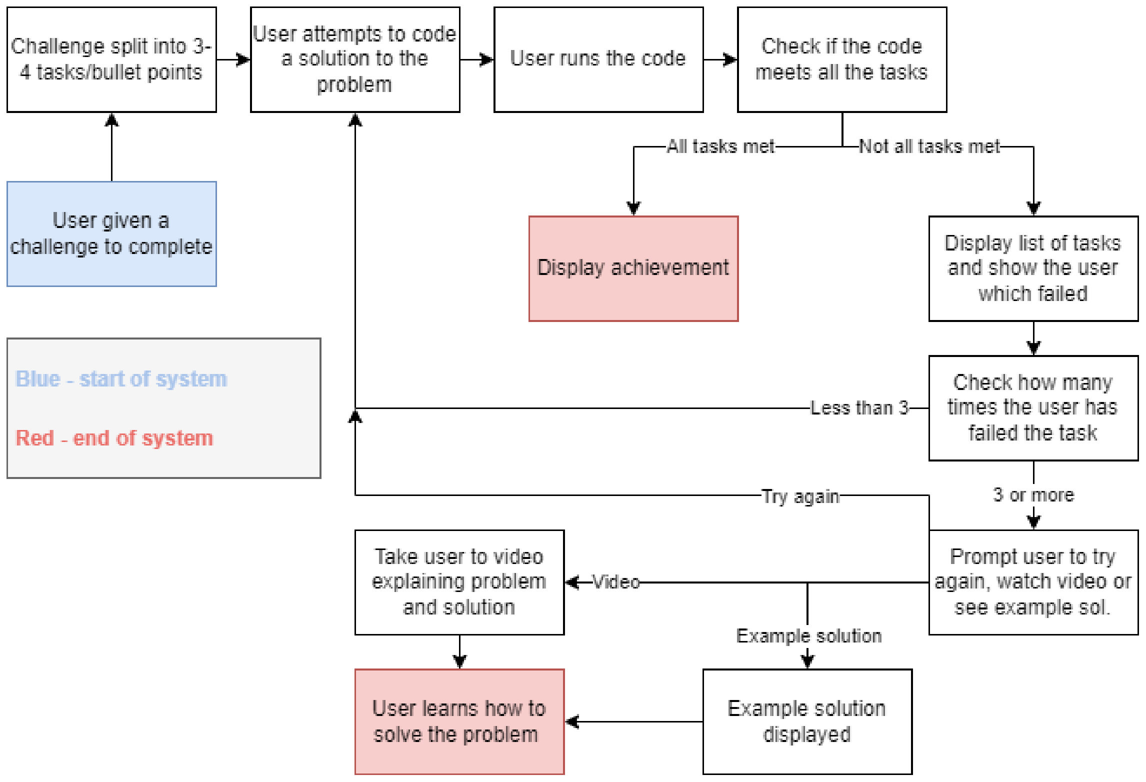
Features that will not be included

- Online forum

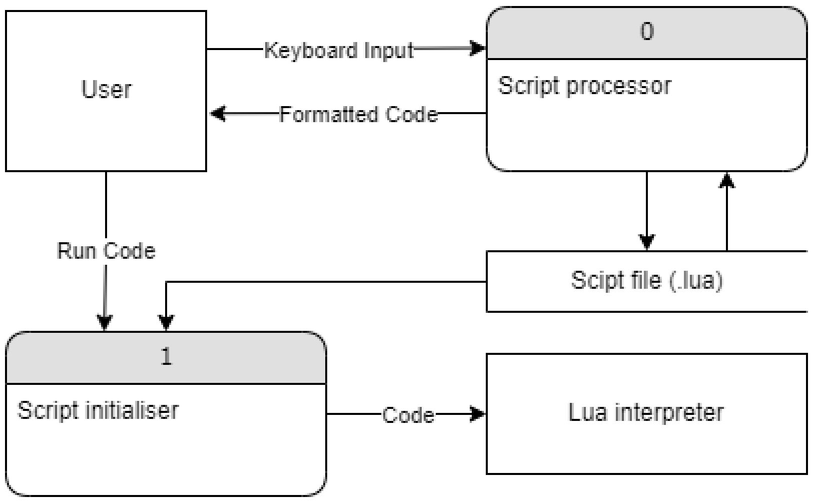
- Way for teacher to control students software

1.4.3 Models for features and systems

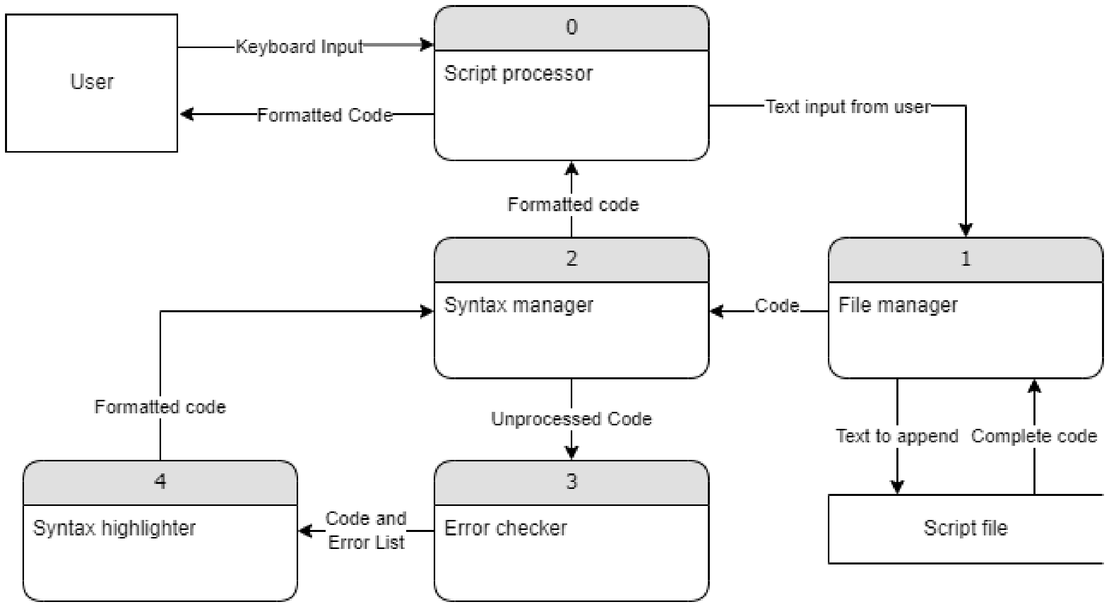
Example challenge/tutorial system


Challenge split into 3-4 tasks/bullet points
User attempts to code a solution to the problem
User runs the code
Check if the code meets all the tasks
All tasks met
Not all tasks met
User given a challenge to complete
Display achievement
Display list of tasks and show the user which failed
Blue - start of system
Red - end of system
Check how many times the user has failed the task
Less than 3
3 or more
Try again
Prompt user to try again, watch video or see example sol.
Take user to video explaining problem and solution
Video
Example solution
User learns how to solve the problem
Example solution displayed

Example scripting system



Example scripting processor



1.5 Objectives

Plain - Objective that must be met.

Italic - Objective that could be met.

X. - Main Objective

X.X. - Sub Objective

1. The software will be easy to install.

    1.1. Stakeholders will rate the ease of install 8+/10.

    1.2. There will be a tutorial on how to install the software

2. There will be an easy to understand menu.

    2.1. Stakeholders will rate how easy it is to navigate 7+/10.

    2.2. The user will be able to access all the features of the software

        2.2.1. The user will be able to access the script editor

        2.2.2. The user will be able to access the achievements

        2.2.2. The user will be able to access the tutorials list

3. There will be a way for the user to edit and use Lua scripts in a script editor.

    3.1. The user will be able to create a new script

        3.1.1. A new file will be created and save to the game directory

    3.2. The user will be able to edit existing scripts

        3.2.1. The user's code will be displayed

            3.2.1.1. There will be syntax highlighting for keywords in Lua

            3.2.1.2. There could be error highlighting

        3.2.2. The user will be able to write code into the script

        3.2.2. The user will be able to delete code from the script

        3.2.4. The user will be able to save the script

    3.3. The user will be able to delete scripts

3.3.1. The script file will be deleted from the game directory.

3.4. The user will be able to run scripts

3.4.1. The user's code will be checked for errors

3.4.2 The user's code will be executed

4. There will be a way for the user to earn achievements.

4.1. After each challenge there will be an achievement

4.2. Achievements will be stored in a gallery where the user can see them

5. There will be a range of tutorials and challenges

5.1. Stakeholders will score the range of challenges 7+/10.

5.2. There will be challenges with varying skill level

5.2.1. Challenges will have a skill level of 1-5

5.2.2. Each challenge will have an example solution

5.3. There will be a way for the user to check their work

5.3.1. A tick will be displayed if the code passes

5.4.1. A cross will be displayed if the code fails

5.4. A user will be able to access a tutorial if they're stuck

5.4.1. An example solution will be displayed

5.4.2. A walkthrough video could be displayed

6. There could be a way for students to submit work

6.1. Students could have their work sent to the teacher for review.

6.2. Teachers could reply with feedback to the student.

2 | Documented Design

After researching and creating objectives for my project, I am now ready to begin the design phase. I will be decomposing the design of my project into the following sections: Installer, Minecraft Mod.

2.1 Installer

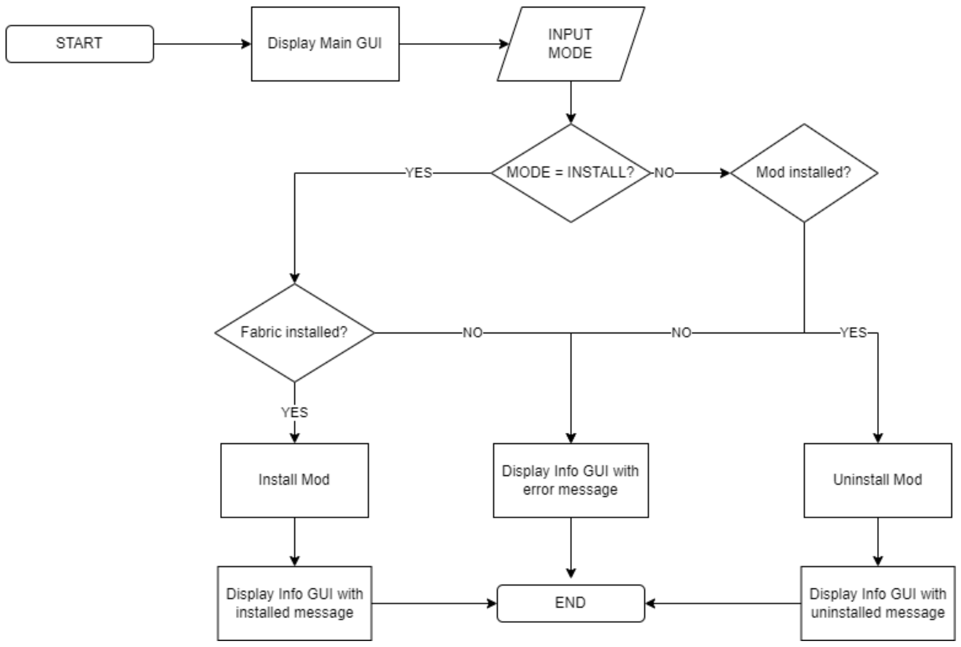
2.1.1 Overview

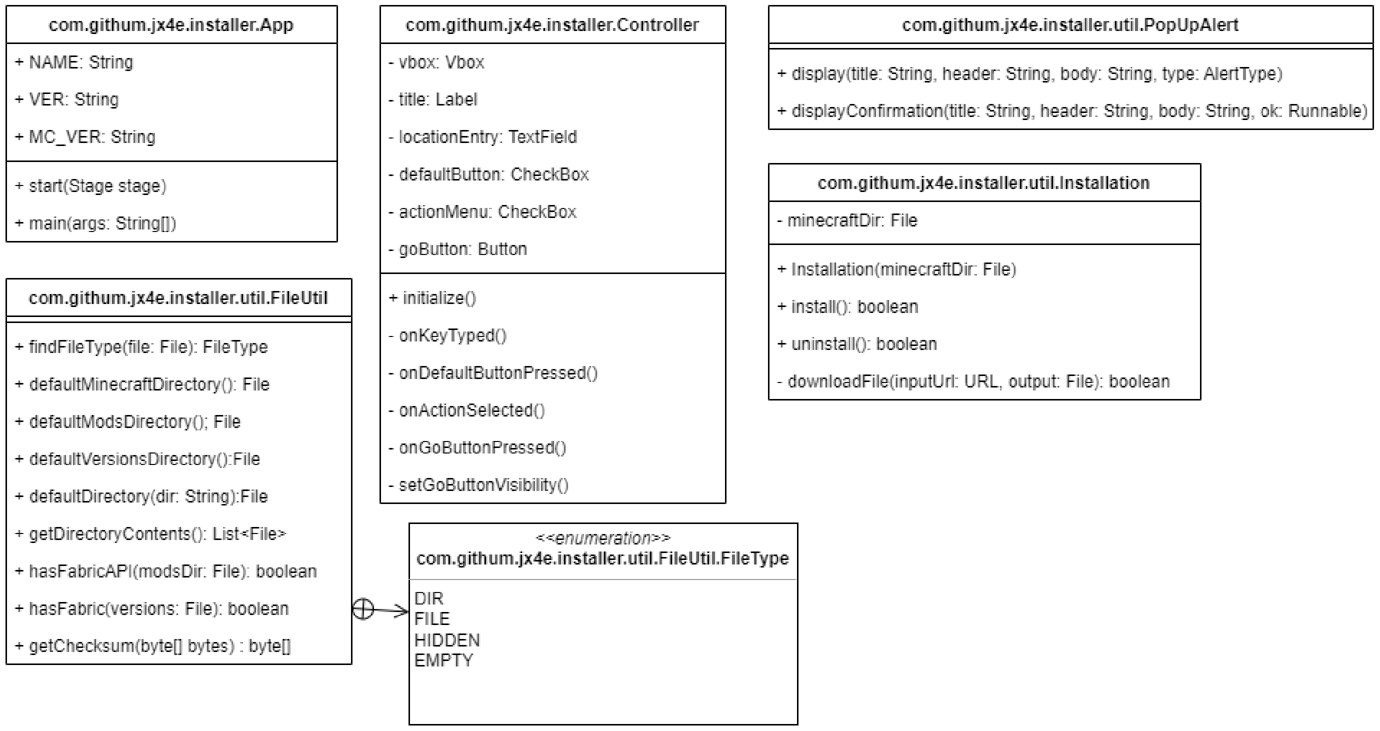
The installer is the first part of the software that the user will experience. The purpose of the installer is to automate the installation process to make it easier for the user.

The installer should:

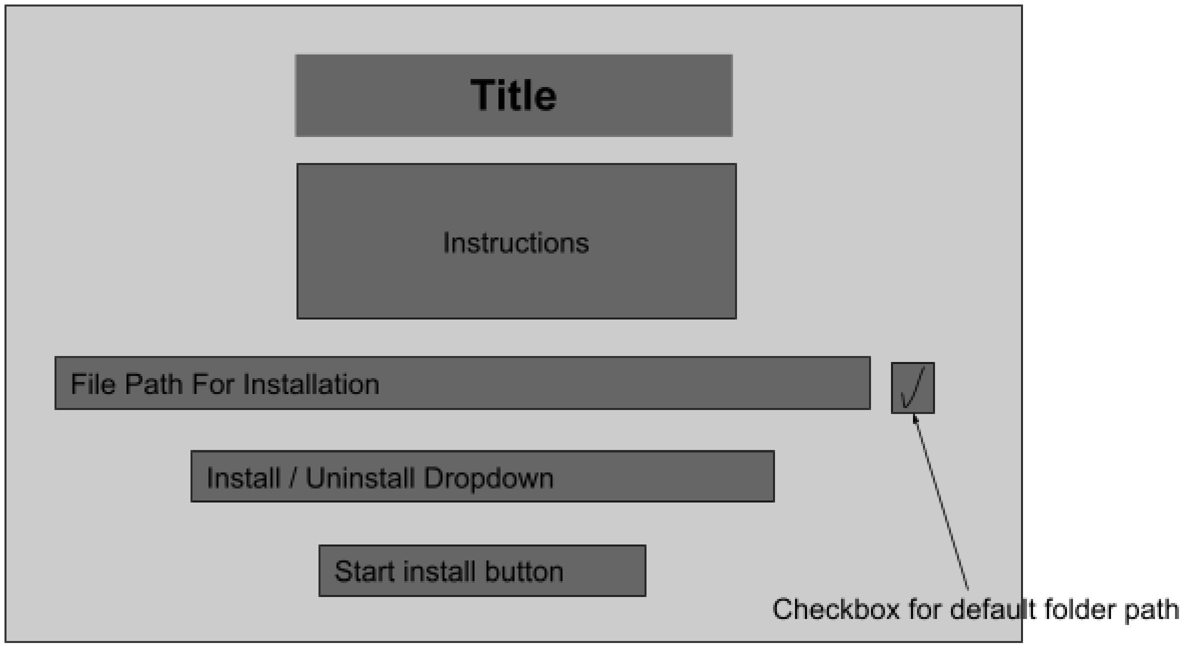
- Check the user has the right software installed
- Download the mod from the right URL
- Check the mod is authentic with a checksum
- Uninstall the mod

Class Diagram & System Flowchart




com.github.jx4e.installer.App
+ NAME: String
+ VER: String
+ MC_VER: String
+ start(Stage stage)
+ main(args: String[])
com.github.jx4e.installer.Controller
- vbox: Vbox
- title: Label
- locationEntry: TextField
- defaultButton: CheckBox
- actionMenu: CheckBox
- goButton: Button
+ initialize()
- onKeyTyped()
- onDefaultButtonPressed()
- onActionSelected()
- onGoButtonPressed()
- setGoButtonVisibility()
com.github.jx4e.installer.util.PopUpAlert
+ display(title: String, header: String, body: String, type: AlertType)
+ displayConfirmation(title: String, header: String, body: String, ok: Runnable)
com.github.jx4e.installer.util.Installation
- minecraftDir: File
+ Installation(minecraftDir: File)
+ install(): boolean
+ uninstall(): boolean
- downloadFile(inputUrl: URL, output: File): boolean
com.github.jx4e.installer.util.FileUtil
+ findFileType(file: File): FileType
+ defaultMinecraftDirectory(): File
+ defaultModsDirectory(); File
+ defaultVersionsDirectory():File
+ defaultDirectory(dir: String):File
+ getDirectoryContents(): List<File>
+ hasFabricAPI(modsDir: File): boolean
+ hasFabric(versions: File): boolean
+ getChecksum(byte[] bytes) : byte[]
<<enumeration>>
com.github.jx4e.installer.util.FileUtil.FileType
DIR
FILE
HIDDEN
EMPTY

2.1.2 GUI Design


Title
Instructions
File Path For Installation
Install / Uninstall Dropdown
Start install button
Checkbox for default folder path

2.1.3 Structure & Algorithms

App Class

The App class will be the entry point of the installer. It will be responsible for setting up and displaying the GUI.

Attributes

Name	**Description**
NAME	The name of the installer
MC_VER	Target Minecraft version of the mod
VER	Version of the installer

Methods

Name	**Description**
start	Method to create and display the GUI
main	Launches the application

Controller Class

The Controller class will be responsible for initialising all the components of the GUI and handling events (Eg: when a key is pressed). The controller will be linked to the GUI in the panel.fxml file and the components in the FXML file will be given the @FXML annotation. Javafx will then automatically create an instance of the Controller and create all of the attributes with the @FXML tag. Javafx will also automatically call any methods that have been added as events in the panel.fxml file.

File Util Class

The File Util class will contain utility methods that are file related. These methods will be used throughout the installer to make file handling easier. The File Util will also contain the method which will calculate the checksum for a file.

Methods

Name	Description
findFileType	Find the file type of the file. Will show if it is a file, directory, hidden or non-existent
defaultMinecraft Directory	Finds the default minecraft game directory on the user's computer
getDirectoryCon tents	Returns a list of all the files that are in that directory
hasFabric	Checks if the user has Fabric installed
getChecksum	Calculate a checksum from a file's bytes

Calculating & using the checksum

To generate a checksum, I have decided to use an algorithm called SHA-256. SHA-256 is better than other algorithms like MD5 and SHA-1 since it uses a longer key and is more secure. I will use the ***java.security.MessageDigest*** as this makes it easy to hash.

To use the checksum, I will have stored what the correct output of the algorithm should be then I can compare it to the value the user obtains at runtime. If they match, the file the user has must be the file I have so it is safe to run.

Key Algorithms

```
FUNCTION GetChecksum(byte[] bytes)
       md = MessageDigest.getInstance("SHA-256");
       md.update(bytes);
       Return md.digest();
```

```
FUNCTION hasFabric(File File)
       Return getDirectoryContents().stream()
   .anyMatch(WHERE file IS Fabric AND version is MC_VER)
```

Installation Class

The purpose of the Installation class is to install and uninstall the mod.

Attributes

Name	Description
minecraftDir	The directory of minecraft to create the installation

Methods

Name	Description
install	Install the mods. It should check the user has Fabric installed and if not it should display a popup and direct the user to the Fabric website. Any errors that occur should be handled by this method. It should also compare the checksums of the file it downloads to check it is safe
uninstall	Mod should be deleted from its folder if it exists.
downloadFile	Download a file from a URL. Returns if it has been successful or not

Key Algorithms

The DownloadFile method will download from a URL. It will create an inputstream from the url and an outputstream to the file. 1024 bytes of data are then read to the buffer before being written to the output. This will repeat while there is still data to be read.

```
FUNCTION DownloadFile(URL url, File file)
        IF url OR file IS NULL RETURN FALSE

        in = InputStream(url)
        out = OutputStream(file)

        buffer = byte[1024]
        WHILE ((read = in.read(buffer, 0, buffer.len)) != -1)
             out.write(buffer, 0, read);
         ENDWHILE

        IF ERROR OCCURS RETURN FALSE

        RETURN TRUE
```

2.2 Minecraft Mod

2.2.1 Overview

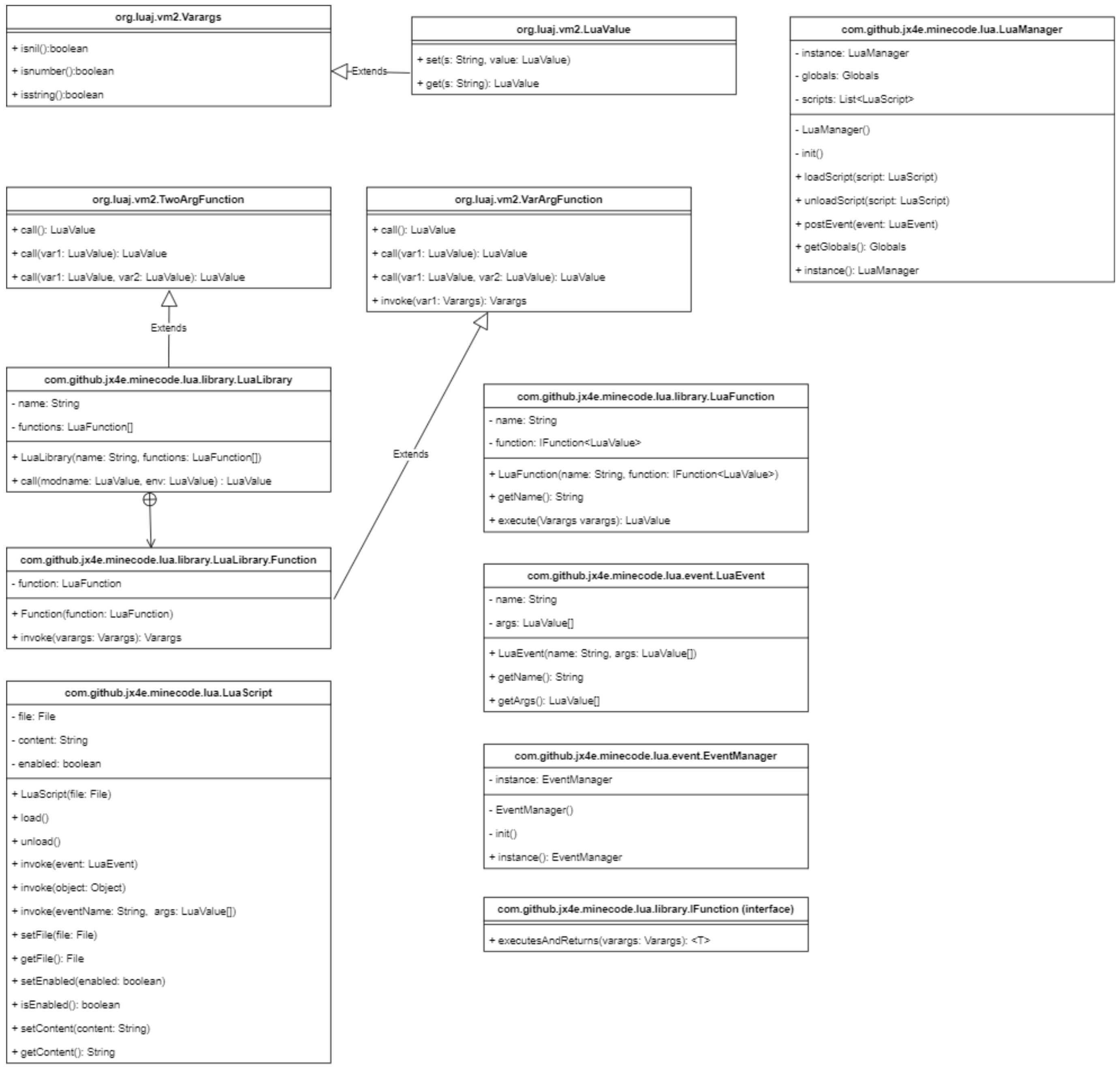
The minecraft mod is the software that will be loaded when the user starts minecraft. The mod will be built using minecraft fabric, a modding api. The purpose of the mod is to teach the user to code. The mod will be split into the following key systems:

- Lua Scripting System
- Project & Lesson System
- Rendering & UI System

2.2.2 Lua Scripting System

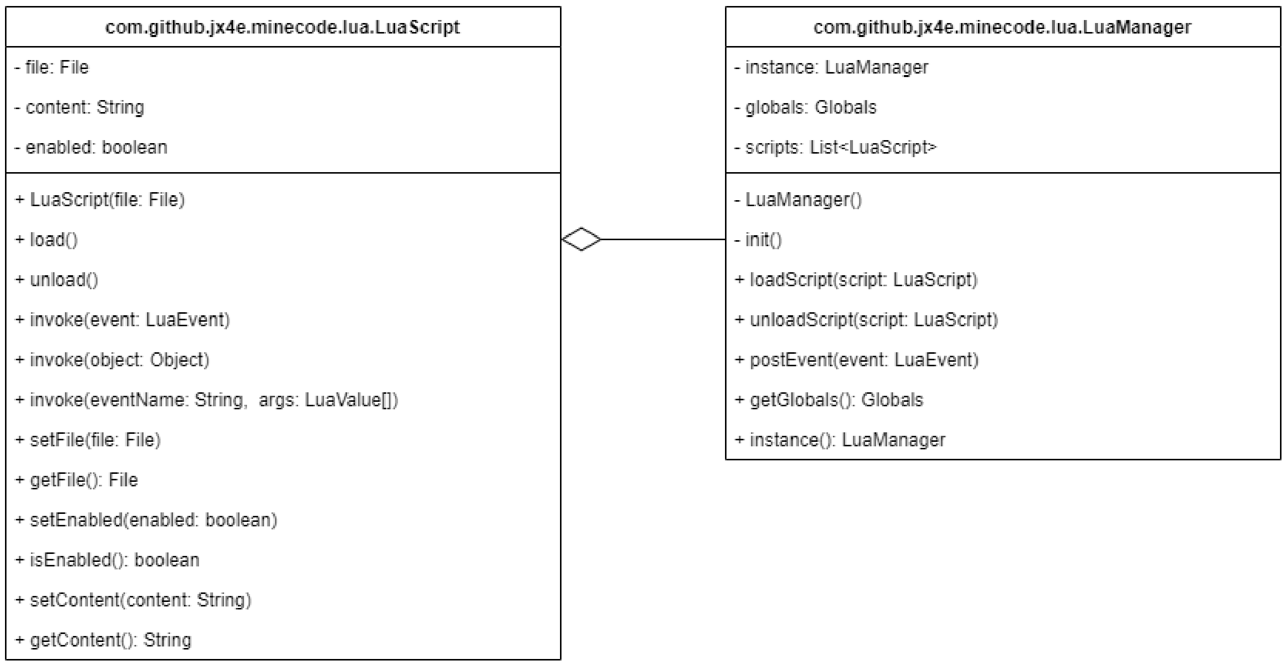
The Lua Scripting System is responsible for running Lua scripts, calling events in Lua scripts when an event happens ingame and creating custom libraries that can be used in Lua scripts.

Class Diagram & System Overview


org.luaj.vm2.Varargs
+ isnil():boolean
+ isnumber():boolean
+ isstring():boolean
Extends
org.luaj.vm2.LuaValue
+ set(s: String, value: LuaValue)
+ get(s: String): LuaValue
com.github.jx4e.minecode.lua.LuaManager
- instance: LuaManager
- globals: Globals
- scripts: List<LuaScript>
- LuaManager()
- init()
+ loadScript(script: LuaScript)
+ unloadScript(script: LuaScript)
+ postEvent(event: LuaEvent)
+ getGlobals(): Globals
+ instance(): LuaManager
org.luaj.vm2.TwoArgFunction
+ call(): LuaValue
+ call(var1: LuaValue): LuaValue
+ call(var1: LuaValue, var2: LuaValue): LuaValue
org.luaj.vm2.VarArgFunction
+ call(): LuaValue
+ call(var1: LuaValue): LuaValue
+ call(var1: LuaValue, var2: LuaValue): LuaValue
+ invoke(var1: Varargs): Varargs
Extends
com.github.jx4e.minecode.lua.library.LuaLibrary
- name: String
- functions: LuaFunction[]
+ LuaLibrary(name: String, functions: LuaFunction[])
+ call(modname: LuaValue, env: LuaValue) : LuaValue
Extends
com.github.jx4e.minecode.lua.library.LuaFunction
- name: String
- function: IFunction<LuaValue>
+ LuaFunction(name: String, function: IFunction<LuaValue>)
+ getName(): String
+ execute(Varargs varargs): LuaValue
com.github.jx4e.minecode.lua.library.LuaLibrary.Function
- function: LuaFunction
+ Function(function: LuaFunction)
+ invoke(varargs: Varargs): Varargs
com.github.jx4e.minecode.lua.event.LuaEvent
- name: String
- args: LuaValue[]
+ LuaEvent(name: String, args: LuaValue[])
+ getName(): String
+ getArgs(): LuaValue[]
com.github.jx4e.minecode.lua.LuaScript
- file: File
- content: String
- enabled: boolean
+ LuaScript(file: File)
+ load()
+ unload()
+ invoke(event: LuaEvent)
+ invoke(object: Object)
+ invoke(eventName: String, args: LuaValue[])
+ setFile(file: File)
+ getFile(): File
+ setEnabled(enabled: boolean)
+ isEnabled(): boolean
+ setContent(content: String)
+ getContent(): String
com.github.jx4e.minecode.lua.event.EventManager
- instance: EventManager
- EventManager()
- init()
+ instance(): EventManager
com.github.jx4e.minecode.lua.library.IFunction (interface)
+ executesAndReturns(varargs: Varargs): <T>

Scripts

The scripting system will need script objects to hold all the data of the .lua file and it will also need a class to manage the scripts. The LuaManager class has a list of LuaScripts that it will be responsible for managing. The LuaManager will be used to load & unload scripts from the Lua environment and post events to all of the Lua scripts. The LuaScript will be used to contain all of the data in a .lua file, store if the script is enabled and execute functions in the code when an event is posted.


com.github.jx4e.minecode.lua.LuaScript
- file: File
- content: String
- enabled: boolean
+ LuaScript(file: File)
+ load()
+ unload()
+ invoke(event: LuaEvent)
+ invoke(object: Object)
+ invoke(eventName: String, args: LuaValue[])
+ setFile(file: File)
+ getFile(): File
+ setEnabled(enabled: boolean)
+ isEnabled(): boolean
+ setContent(content: String)
+ getContent(): String
com.github.jx4e.minecode.lua.LuaManager
- instance: LuaManager
- globals: Globals
- scripts: List<LuaScript>
- LuaManager()
- init()
+ loadScript(script: LuaScript)
+ unloadScript(script: LuaScript)
+ postEvent(event: LuaEvent)
+ getGlobals(): Globals
+ instance(): LuaManager

LuaScript Class

Lua Scripts are what the user will use to create programs. They will contain the actual .lua code file, the content inside the file and whether it is enabled or not. Events can be posted to the Lua Scripts and they will find the correct function in the code and then run the function.

Attributes

Name	Description
file	The .lua file
content	The content inside the file

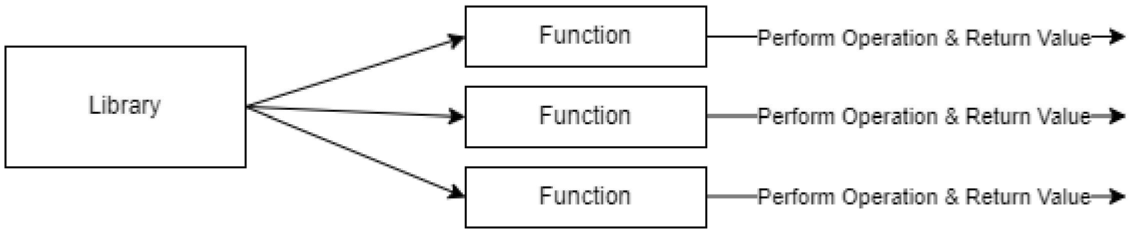
enabled	If the script is active or not

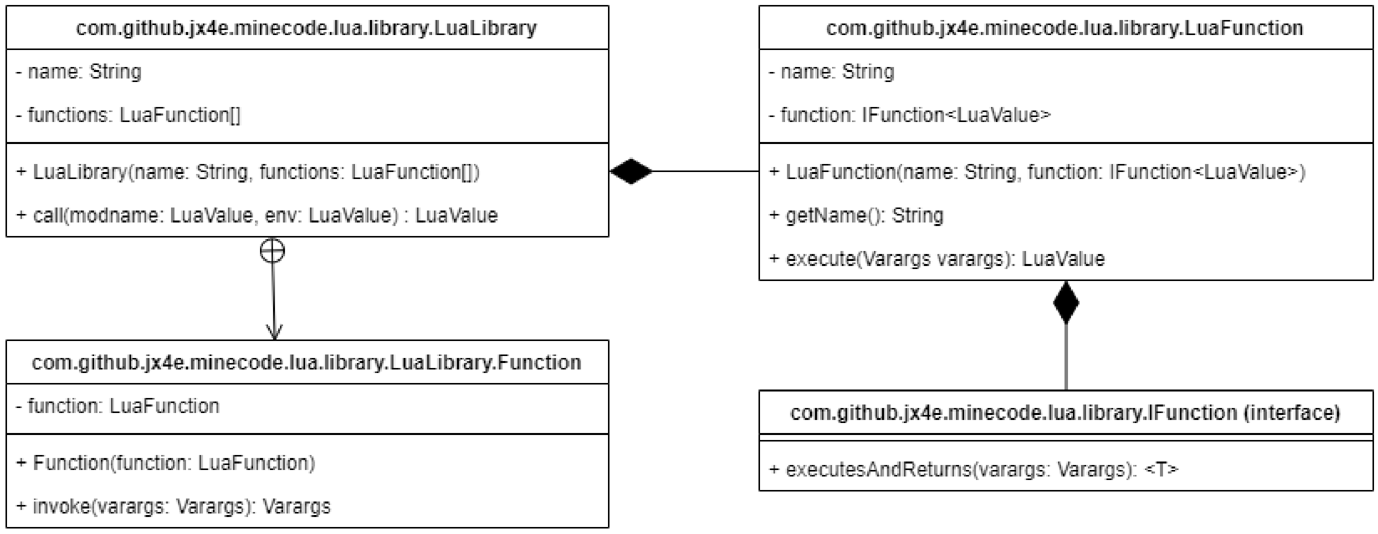
Methods

Name	Description
load	Loads the script into the Lua Environment
unload	Unloads the script into the Lua Environment
invoke	Calls function with the name of the Lua Event in the script
setFile	Sets the file
getFile	Gets the file
setEnabled	Set the script to enabled or disabled
isEnabled	Gets if the script is enabled
setContent	Write the new content to the file, then change the content attribute
getContent	Get the content

Lua Libraries

A Lua library is similar to a library in any other programming language. It has functions and can be imported into a Lua script so the functions can be used by the user.


Library
Function
Function
Function
Perform Operation & Return Value
Perform Operation & Return Value
Perform Operation & Return Value


com.github.jx4e.minecode.lua.library.LuaLibrary
- name: String
- functions: LuaFunction[]
+ LuaLibrary(name: String, functions: LuaFunction[])
+ call(modname: LuaValue, env: LuaValue) : LuaValue
com.github.jx4e.minecode.lua.library.LuaFunction
- name: String
- function: IFunction<LuaValue>
+ LuaFunction(name: String, function: IFunction<LuaValue>)
+ getName(): String
+ execute(Varargs varargs): LuaValue
com.github.jx4e.minecode.lua.library.LuaLibrary.Function
- function: LuaFunction
+ Function(function: LuaFunction)
+ invoke(varargs: Varargs): Varargs
com.github.jx4e.minecode.lua.library.IFunction (interface)
+ executesAndReturns(varargs: Varargs): <T>

LuaLibrary Class

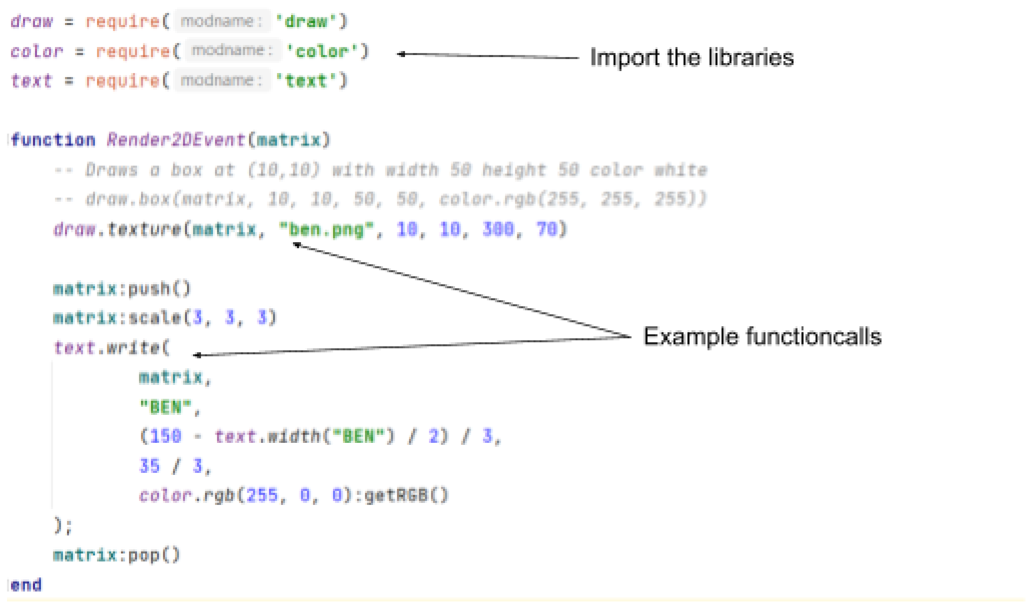
A LuaLibrary is a library that can be imported into a Lua script. It has a name that is used to import it and a list of LuaFunctions that can be called from the library.

Example Usage

The following creates a LuaLibrary called “text” with some arbitrary functions.

```java
public class LuaTextLibrary extends LuaLibrary {
   public LuaTextLibrary() {
       super("text", new LuaFunction[]{function1, function2, …});
   }
}
```

Example LuaLibrary in a Lua Script



```lua
draw = require( modname: 'draw')
color = require( modname: 'color')
text = require( modname: 'text')

function Render2DEvent(matrix)
    -- Draws a box at (10,10) with width 50 height 50 color white
    -- draw.box(matrix, 10, 10, 50, 50, color.rgb(255, 255, 255))
    draw.texture(matrix, "ben.png", 10, 10, 300, 70)

    matrix:push()
    matrix:scale(3, 3, 3)
    text.write(
            matrix,
            "BEN",
            (150 - text.width("BEN") / 2) / 3,
            35 / 3,
            color.rgb(255, 0, 0):getRGB()
    );
    matrix:pop()
end
```

Attributes

Name	Description
name	The name of the library
functions	The functions of the library

Methods

Name	Description
call	This method is called when the library is imported. It is responsible for loading the library and its methods so they can be called from the Lua script

Function Inner-Class

The Function class exists inside the LuaLibrary class since this is the only class that references it. It will have one attribute - the LuaFunction

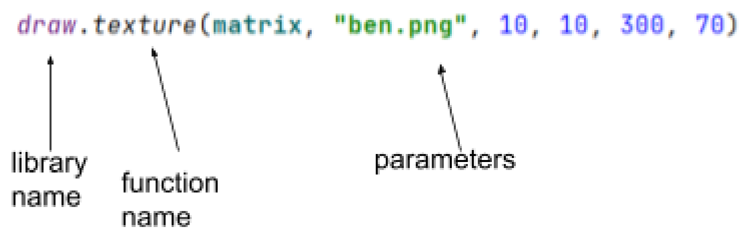
LuaFunction Class

LuaFunctions are functions that are stored as a part of a LuaLibrary (see above). Their purpose is to perform an operation and return a value to the Lua script. They have a name and an IFunction<LuaValue> function. They can be called in a Lua script by doing libraryName.functionName(params). This will execute the function with the arguments from the Lua script.

Example Usage

```
LuaFunction function = new LuaFunction("texture", varargs -> {code here...});
```

Example LuaFunction in a Lua Script



```
draw.texture(matrix, "ben.png", 10, 10, 300, 70)
```

Attributes

Name	Description
name	The name of the library
function	The IFunction that will perform an operation and return a value

Methods

Name	Description
getName	Get the name of the function
execute	Takes the varargs from the function in the Lua script and passes them into the IFunction to get a return value

IFunction Interface

IFunction is a simple interface used to take in Varargs, execute some code then return an output value. IFunction is a generic interface and takes some class, T, from the user. The use of generics will allow this interface to return a value of any specified type. Since this class is an interface and it has only 1 method, we can use a lambda to define it.

Example Usage

The following is an example that would return the string value of the varargs using a lambda.

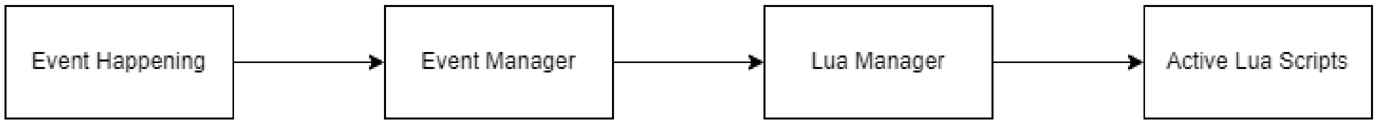
```
IFunction<String> printFunction = varargs -> varargs.toString();
```

Methods

Name	Description
executesAndReturns	Execute some code then return an output value

Lua Events

The Lua scripts need to know when something happens ingame such as a block breaking or a frame being rendered. This is why we need an Event System. The event system will communicate between the Lua scripts and the java code. Events are detected in the Event Manager then posted to the Lua Manager. The Lua Manager will then post the event to all of the enabled Lua scripts.



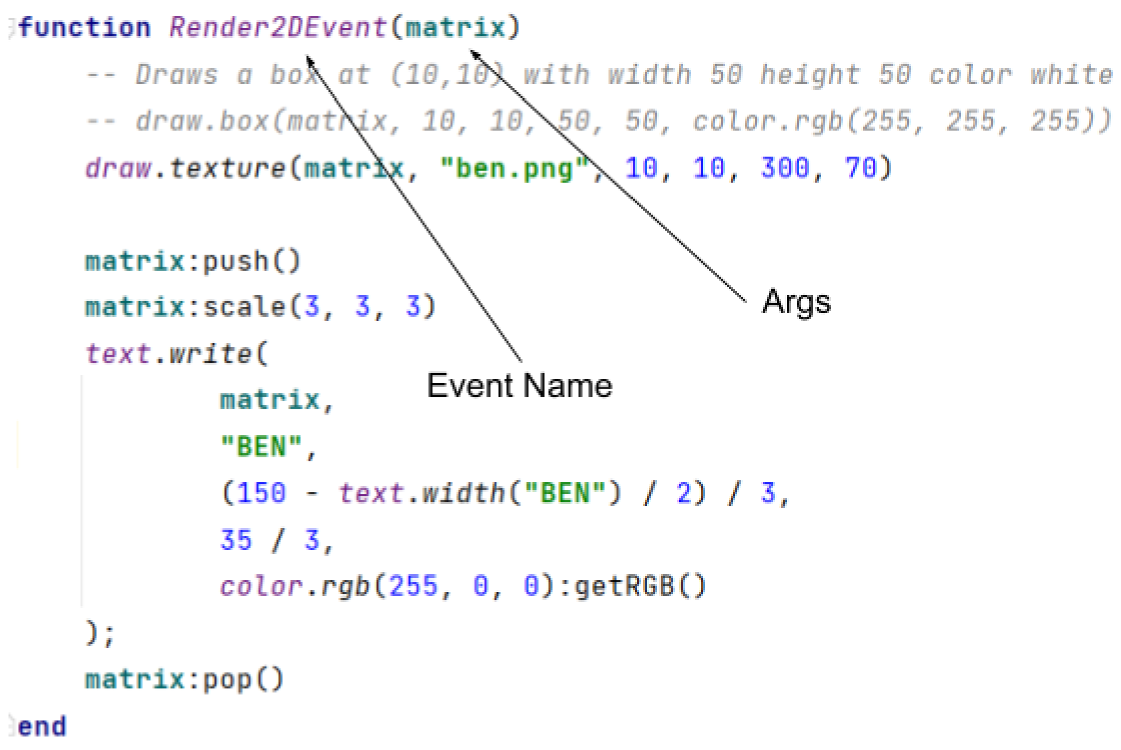
LuaEvent Class

The Lua event class is used as a framework for events in Lua scripts. For Example: The LuaEvent class can be used to create an event that is called every render tick. The name would be “RenderEvent” and its args would be the render matrix. Then the instance of the event is posted to the event system and it is called in all Lua scripts that contain that event.

Example Usage

```
public class Render2DEvent extends LuaEvent {
   public Render2DEvent(MatrixStack matrix) {
       super("Render2DEvent", new LuaValue[]{
               CoerceJavaToLua.coerce(matrix)
       });
   }
}

Note : CoerceJavaToLua.coerce() is a method that converts a java object to a LuaValue
```

Example Of LuaEvent in a Lua Script



```
function Render2DEvent(matrix)
    -- Draws a box at (10,10) with width 50 height 50 color white
    -- draw.box(matrix, 10, 10, 50, 50, color.rgb(255, 255, 255))
    draw.texture(matrix, "ben.png", 10, 10, 300, 70)

    matrix:push()
    matrix:scale(3, 3, 3)
    text.write(
            matrix,
            "BEN",
            (150 - text.width("BEN") / 2) / 3,
            35 / 3,
            color.rgb(255, 0, 0):getRGB()
    );
    matrix:pop()
end
```

Attributes

Name	Description

name	The name of the event
args	The arguments that should be taken into the event

Methods

Name	**Description**
getName	Get the name of the event
getArgs	Get the arguments of the event

EventManager Class

EventManager is responsible for detecting events as they happen and posting them to the LuaManager.

Attributes

Name	**Description**
instance	Instance of the EventManager. There can only be a single instance of this class.

Methods

Name	**Description**
initEvents	Register the callback methods with the fabric event system. Callbacks are run when events happen. Each callback will create a LuaEvent and post it to the LuaManager.
instance	Get the instance of the EventManager

2.2.3 Project & Lesson System

The project and lesson system will be responsible for downloading any resources that are needed for the mod and handling the project & lesson files.

Config Manager Class

The config manager is an important class which will be in charge of creating and loading files for the mod. It has several responsibilities which include creating directories, managing downloading resources and reading lessons and projects from their files. Below the responsibilities are explained in greater depth.

Resources

The resources that the project needs will be stored in a file called minecode.resources.json. Inside this file there will be a list of json objects with a name attribute and a url attribute. The name attribute will be the name of the file and the url will be where to download the file from. The config manager will have to be able to interpret and read this JSON. An example resource file could be:

```
{
  "resources" : [
      {
      "name" : "picture.png",
      "url" : "https://example.com/picture.png"
      },
      {
      "name" : "font.ttf",
      "url" : "https://example.com/font.ttf"
      }
]
}
```

Creating Directories

When the mod is loaded, the config manager must set up the directories which the mod will need. If the directories do not exist the config manager must create them.

Loading Lessons & Projects

The config manager will be responsible for passing the projects file and the lessons file into the ProjectManager and LessonManager. It will have to find and locate these directories.

Attributes

Name	Description
directory	The main directory for the mod. Located in the minecraft running directory.

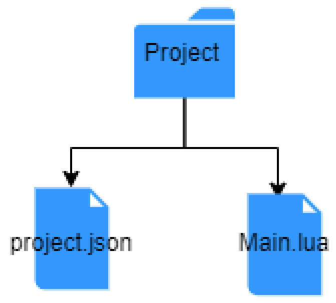
resource	The resources directory
project	The projects directory
lessons	The lessons directory

Methods

Name	**Description**
load	Called when the mod starts. Creates directories and downloads resources and loads projects
setupDirectory	Sets up directory in the correct way
loadResources	Loads resources. Called from the load method
loadProjectsAndLessons	Loads projects and lessons. Called from the load method

Projects

Projects will have their own folder in the Minecode directory. A project will have a project.json file detailing information about the project such as: name, status, main script. The main script is the .lua file that will be run when the project is enabled. A project will have the following file structure:



Lua Project Class

The Lua Project class will be used to create objects which contain the files in the project, info about the project and the main script which will be run when the project is enabled. A LuaProject object can be instantiated with a directory. The project.json file will then be read and the object will be created.

Attributes

Name	Description
directory	The project directory
projectFile	The project.json file
scriptFile	The main script file
name	The project name. Read from the project.json file
enabled	If the project is enabled. Read from the project.json file
mainScriptName	The name of the main script. Read from the project.json file
mainScript	The main script object. It is a LuaScript which will be run when project is enabled.

Methods

Name	Description
reloadScript	Reloads the main script from the project.json file
setEnabled	Sets the enabled attribute to a value. Will also enable/disable the mainScript.
toggle	Sets enabled to !enabled

Project Manager Class

The project manager will be responsible for storing all the loaded projects in a list. When the mod is loaded it will be passed a directory containing all of the project files and it will have to interpret that directory and create a list of LuaProject objects. These projects will be stored in an ArrayList called projects. This class will also have the functionality to create a new project while the mod is running.

Attributes

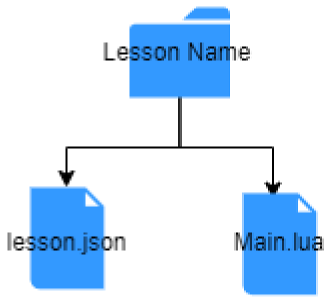
Name	Description
projects	The arraylist of LuaProjects

Methods

Name	Description
createProjectsFromDir	Creates projects from a directory. Adds the results to the projects ArrayList
createProject	Takes in a project name, main script name and if to use template. Then creates a new folder with the project.json file and main script .lua file. It will then clear all loaded projects and reload all the projects from scratch.

Lessons

Lessons will have their own folder in the Minecode directory. A lesson will have a lesson.json file detailing information about the lesson such as: name, description, content and tasks. The main.lua file is the file that will be edited when the user tries to complete the task.



Lua Lesson Class

The LuaLesson class will hold data about the files of the lesson as well as details like the name, description, content (what the lesson is teaching) and tasks (objectives for the user's code task). The constructor takes in a directory and will then create the object with data from the lesson.json file.

Lesson Content Record Class

The LessonContent class will be a record that is a subclass of the LuaLesson class. It will be located inside the LuaLesson class since it is only relevant to a LuaLesson. It will be a simple record class with attributes: text and code. I will use a record because it is immutable and I will not need to change any attributes after it has been created.

Attributes

Name	**Description**
dir	The directory of the lesson
lessonFile	The lesson.json file
mainScriptFile	The script file.
name	Name of lesson
description	Contains information about the lesson
content	ArrayList of LessonContent objects. Contains the actual "content" of the lesson that the user will learn. The "content" will contain a piece of code followed by an explanation so that the user can learn
tasks	ArrayList of LessonContent objects. Contains the "tasks" that the user must complete to beat the code task. The text attribute of the LessonContent will be what the user is asked to do. The code attribute will not be displayed to the user and it will be what is checked when the user submits their code.

Lesson Manager Class

The lesson manager will be responsible for storing all the loaded lessons in a list. When the mod is loaded it will be passed a directory containing all of the lesson files and it will have to interpret that directory and create a list of LuaLesson objects. These lessons will be stored in an ArrayList called lessons.

Attributes

Name	**Description**
lessons	The arraylist of LuaLessons

Methods

Name	Description
createLessonsFromDir	Creates lessons from a directory. Adds the results to the lessons ArrayList

IOUtil Class

Since many of the classes use repeated methods in the project package, I have decided to create a utility class to store all of these repeated methods. This class will allow the developer to easily work with input/output streams. All methods in this class will be public and static. If a developer tries to create an object of this class it will throw an UnsupportedOperationException() as this class is purely intended as a utility class (similar to a namespace in C++).

Methods

Name	Description
readResourceStream	Reads a resource stream and returns the contents as a string
writeToOutputStream	Takes in an InputStream and an OutputStream. Writes the contents of the InputStream to the OutputStream
downloadFile	Takes in a URL and a File. Creates a new BufferedInputStream with the URL and a new FileOutputStream with the File. Passes these into the writeToOutputStream method so that the contents of the URL are written to the file

Key Algorithms

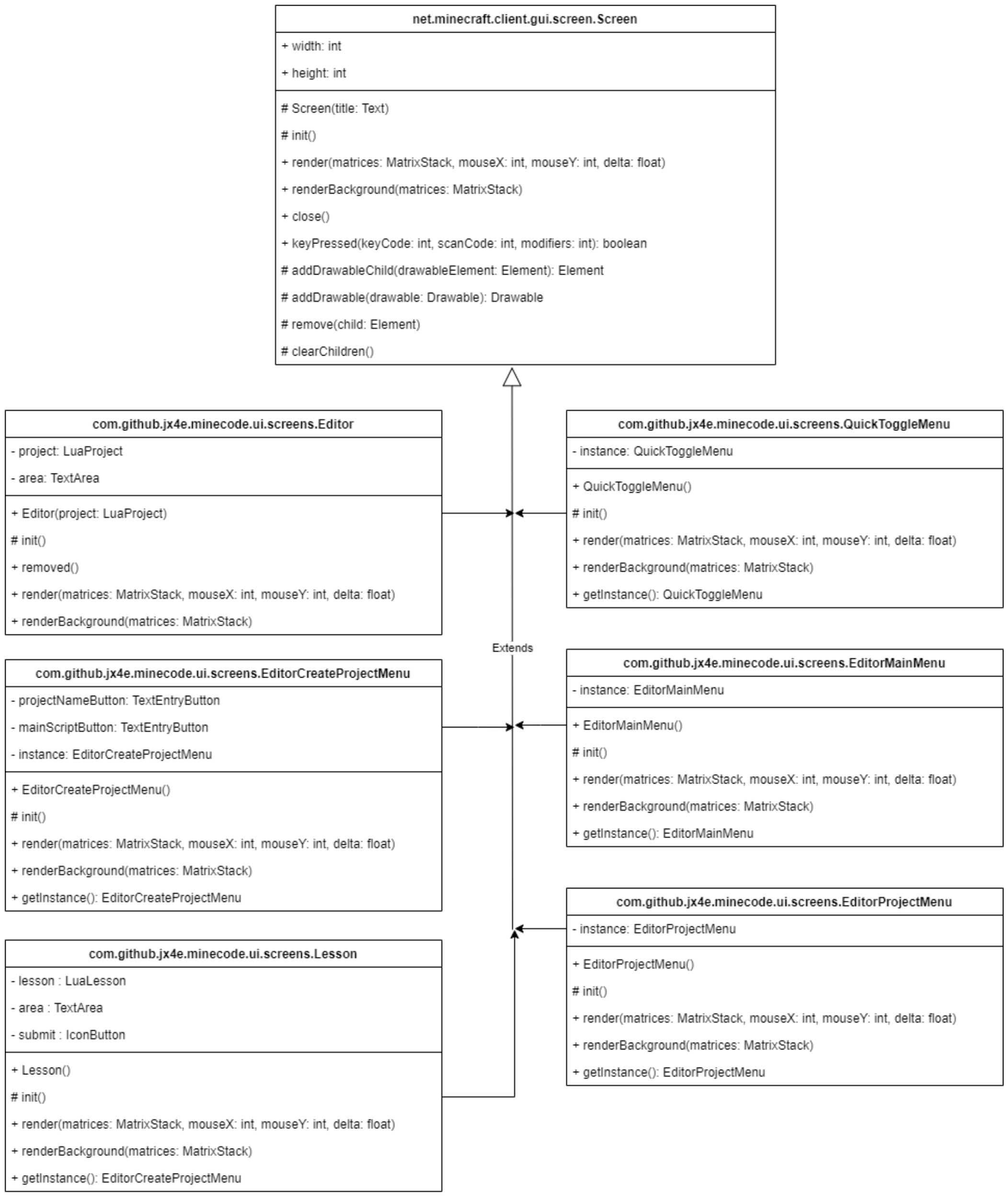
```
writeToOutputStream(in, out)
    TRY
       // Create empty byte array
       data = byte[1024]

       // Assign the amount of bytes read to "read"
       // Add the bytes read to data
       // Repeat this while there is still data to be read
       WHILE (read = in.read(data, 0, 1024)) != -1
             // Write the data to the output stream
             out.write(data, 0, read)
```

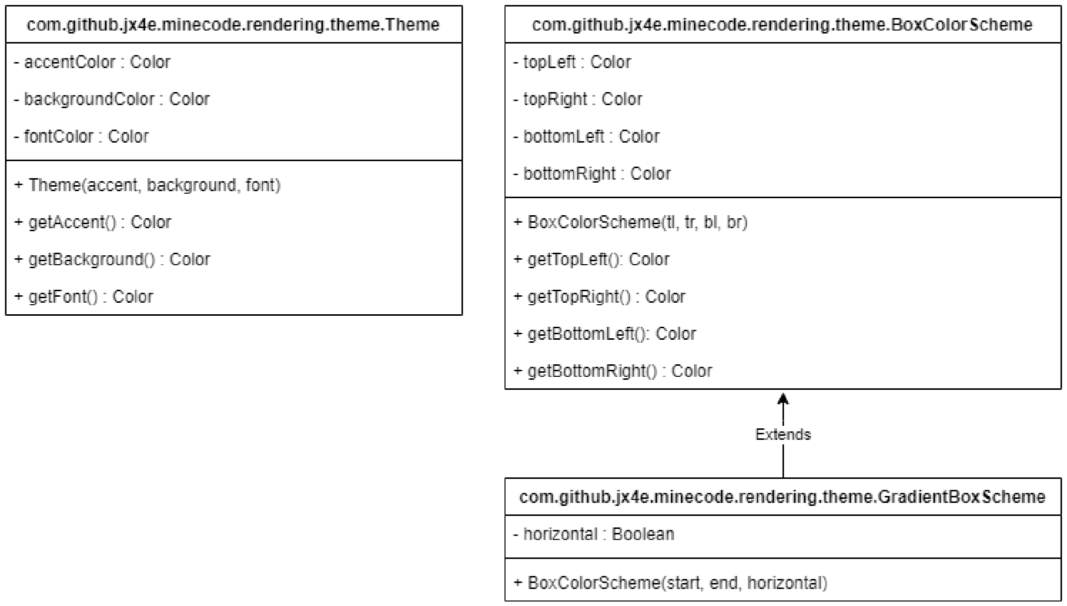
```
        ENDWHILE
    CATCH IOException
        // Handle Exception
```

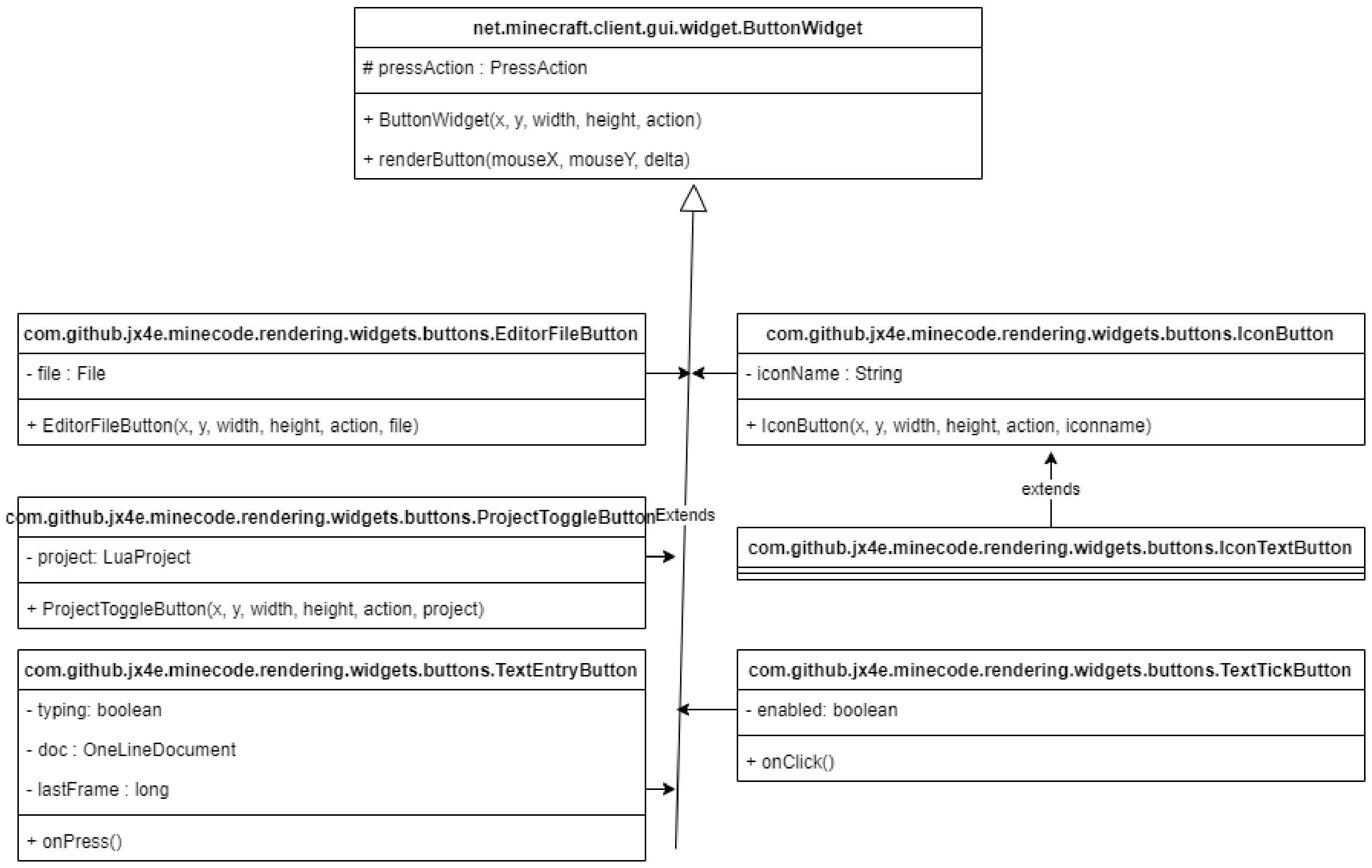
2.2.4 Rendering & UI System

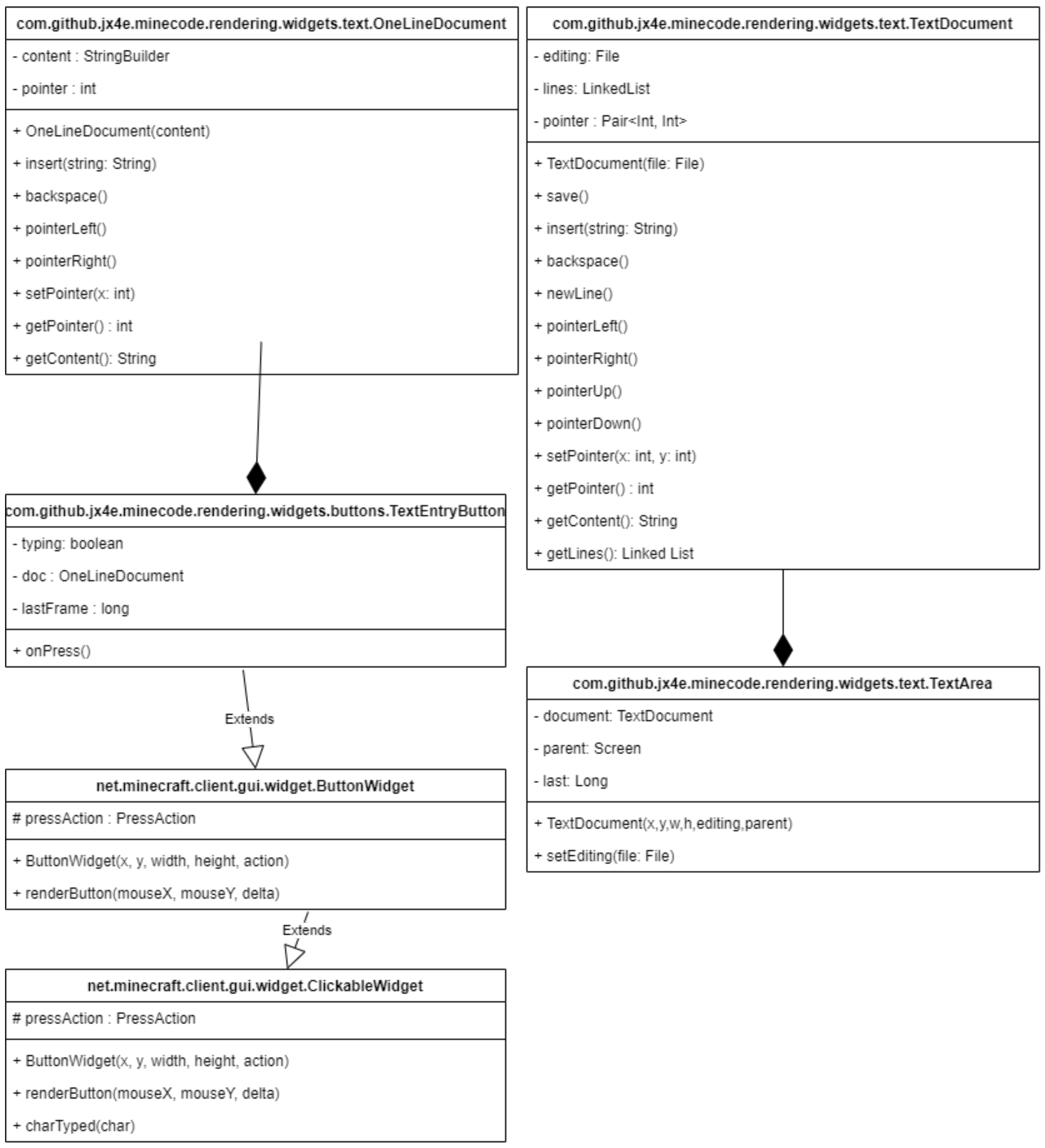
The rendering and UI system will be responsible for providing the user with a graphical interface. It should be able to display GUI Screens and handle user inputs like mouse clicks and typing.

Class Diagram & System Overview


net.minecraft.client.gui.screen.Screen
+ width: int
+ height: int
# Screen(title: Text)
# init()
+ render(matrices: MatrixStack, mouseX: int, mouseY: int, delta: float)
+ renderBackground(matrices: MatrixStack)
+ close()
+ keyPressed(keyCode: int, scanCode: int, modifiers: int): boolean
# addDrawableChild(drawableElement: Element): Element
# addDrawable(drawable: Drawable): Drawable
# remove(child: Element)
# clearChildren()
Extends
com.github.jx4e.minecode.ui.screens.Editor
- project: LuaProject
- area: TextArea
+ Editor(project: LuaProject)
# init()
+ removed()
+ render(matrices: MatrixStack, mouseX: int, mouseY: int, delta: float)
+ renderBackground(matrices: MatrixStack)
com.github.jx4e.minecode.ui.screens.QuickToggleMenu
- instance: QuickToggleMenu
+ QuickToggleMenu()
# init()
+ render(matrices: MatrixStack, mouseX: int, mouseY: int, delta: float)
+ renderBackground(matrices: MatrixStack)
+ getInstance(): QuickToggleMenu
com.github.jx4e.minecode.ui.screens.EditorCreateProjectMenu
- projectNameButton: TextEntryButton
- mainScriptButton: TextEntryButton
- instance: EditorCreateProjectMenu
+ EditorCreateProjectMenu()
# init()
+ render(matrices: MatrixStack, mouseX: int, mouseY: int, delta: float)
+ renderBackground(matrices: MatrixStack)
+ getInstance(): EditorCreateProjectMenu
com.github.jx4e.minecode.ui.screens.EditorMainMenu
- instance: EditorMainMenu
+ EditorMainMenu()
# init()
+ render(matrices: MatrixStack, mouseX: int, mouseY: int, delta: float)
+ renderBackground(matrices: MatrixStack)
+ getInstance(): EditorMainMenu
com.github.jx4e.minecode.ui.screens.Lesson
- lesson : LuaLesson
- area : TextArea
- submit : IconButton
+ Lesson()
# init()
+ render(matrices: MatrixStack, mouseX: int, mouseY: int, delta: float)
+ renderBackground(matrices: MatrixStack)
+ getInstance(): EditorCreateProjectMenu
com.github.jx4e.minecode.ui.screens.EditorProjectMenu
- instance: EditorProjectMenu
+ EditorProjectMenu()
# init()
+ render(matrices: MatrixStack, mouseX: int, mouseY: int, delta: float)
+ renderBackground(matrices: MatrixStack)
+ getInstance(): EditorProjectMenu

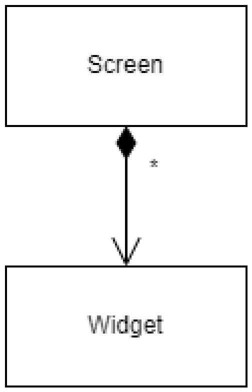

com.github.jx4e.minecode.rendering.theme.Theme
- accentColor : Color
- backgroundColor : Color
- fontColor : Color
+ Theme(accent, background, font)
+ getAccent() : Color
+ getBackground() : Color
+ getFont() : Color
com.github.jx4e.minecode.rendering.theme.BoxColorScheme
- topLeft : Color
- topRight : Color
- bottomLeft : Color
- bottomRight : Color
+ BoxColorScheme(tl, tr, bl, br)
+ getTopLeft(): Color
+ getTopRight() : Color
+ getBottomLeft(): Color
+ getBottomRight() : Color
Extends
com.github.jx4e.minecode.rendering.theme.GradientBoxScheme
- horizontal : Boolean
+ BoxColorScheme(start, end, horizontal)


net.minecraft.client.gui.widget.ButtonWidget
# pressAction : PressAction
+ ButtonWidget(x, y, width, height, action)
+ renderButton(mouseX, mouseY, delta)
com.github.jx4e.minecode.rendering.widgets.buttons.EditorFileButton
- file : File
+ EditorFileButton(x, y, width, height, action, file)
com.github.jx4e.minecode.rendering.widgets.buttons.IconButton
- iconName : String
+ IconButton(x, y, width, height, action, iconname)
extends
com.github.jx4e.minecode.rendering.widgets.buttons.ProjectToggleButton
Extends
- project: LuaProject
+ ProjectToggleButton(x, y, width, height, action, project)
com.github.jx4e.minecode.rendering.widgets.buttons.IconTextButton
com.github.jx4e.minecode.rendering.widgets.buttons.TextEntryButton
- typing: boolean
- doc : OneLineDocument
- lastFrame : long
+ onPress()
com.github.jx4e.minecode.rendering.widgets.buttons.TextTickButton
- enabled: boolean
+ onClick()


com.github.jx4e.minecode.rendering.widgets.text.OneLineDocument
- content : StringBuilder
- pointer : int
+ OneLineDocument(content)
+ insert(string: String)
+ backspace()
+ pointerLeft()
+ pointerRight()
+ setPointer(x: int)
+ getPointer() : int
+ getContent(): String
com.github.jx4e.minecode.rendering.widgets.text.TextDocument
- editing: File
- lines: LinkedList
- pointer : Pair<Int, Int>
+ TextDocument(file: File)
+ save()
+ insert(string: String)
+ backspace()
+ newLine()
+ pointerLeft()
+ pointerRight()
+ pointerUp()
+ pointerDown()
+ setPointer(x: int, y: int)
+ getPointer() : int
+ getContent(): String
+ getLines(): Linked List
com.github.jx4e.minecode.rendering.widgets.buttons.TextEntryButton
- typing: boolean
- doc : OneLineDocument
- lastFrame : long
+ onPress()
Extends
net.minecraft.client.gui.widget.ButtonWidget
# pressAction : PressAction
+ ButtonWidget(x, y, width, height, action)
+ renderButton(mouseX, mouseY, delta)
Extends
net.minecraft.client.gui.widget.ClickableWidget
# pressAction : PressAction
+ ButtonWidget(x, y, width, height, action)
+ renderButton(mouseX, mouseY, delta)
+ charTyped(char)
com.github.jx4e.minecode.rendering.widgets.text.TextArea
- document: TextDocument
- parent: Screen
- last: Long
+ TextDocument(x,y,w,h,editing,parent)
+ setEditing(file: File)

UI Screens and Widgets

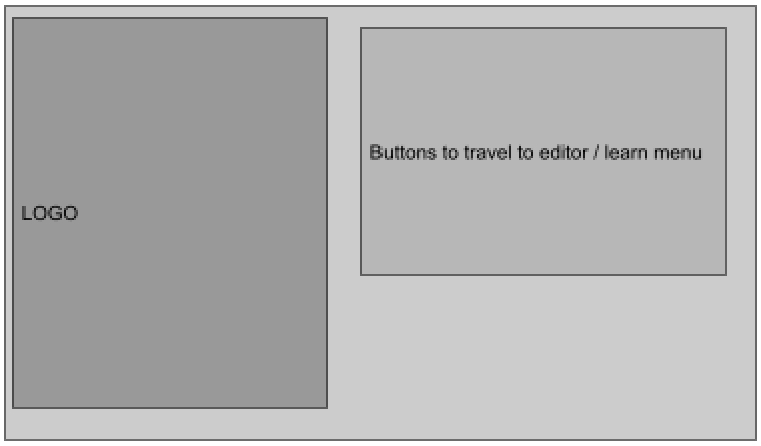
These are my designs for the Screens and widgets of the mod. Each screen contains many 'child' elements called widgets. These widgets can be clicked to perform actions.



Main Menu

The main menu for the mod. It will provide the user with options to either go to the projects menu or lessons menu.

Design



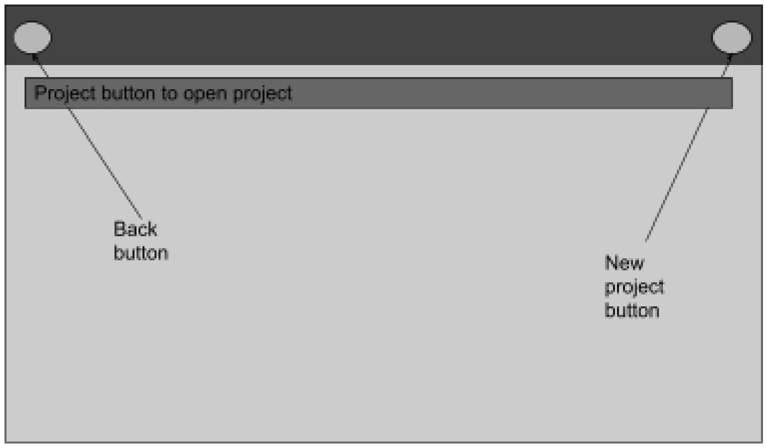
Methods

Name	Description
init	Inherited from the Screen class. This method is called when the screen is displayed. It will initialise all of the widgets on the screen.
render	Inherited from the Screen class. Draws the screen and all the widgets on the screen. It takes in the current render matrix and mouse position as parameters.

Select Project

The select project screen will let the user open an existing project or go to the create project screen to make a new one

Design



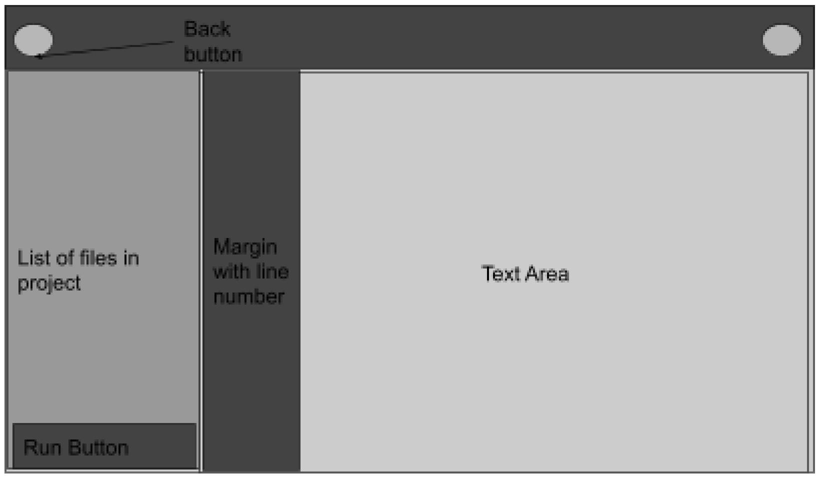
Methods

Name	Description
init	Inherited from the Screen class. This method is called when the screen is displayed. It will initialise all of the widgets on the screen.
render	Inherited from the Screen class. Draws the screen and all the widgets on the screen. It takes in the current render matrix and mouse position as parameters.

Project Editor

The editor is where the user will spend most of their time. It is for programming Lua scripts to perform ingame tasks. The screen will feature a list of files in the project as well as a text area to edit the contents from files.

Design

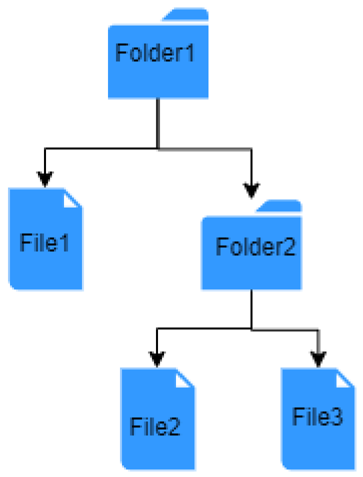


Attributes

Name	Description
project	The project that is currently open in the editor
area	The text area widget that the user will type their code into

Methods

Name	Description
init	Inherited from the Screen class. This method is called when the screen is displayed. It will initialise all of the widgets on the screen.
render	Inherited from the Screen class. Draws the screen and all the widgets on the screen. It takes in the current render matrix and mouse position as parameters.
removed	Inherited from the Screen class. This method is called when the screen is closed/removed. I will override this method to save the document that is currently being edited.
addChildFiles	This recursive method will iterate through all files and subdirectories in a file and create an EditorFileButton for the file. It works by taking in a string prefix, a file and the position to create the button at. It then uses a for loop to iterate over each file inside that directory. If the file it is checking is a directory, it will add the name of this directory to the end of the prefix and call itself with the new prefix. For Example:

	 Let’s say addChildFiles(“”, Folder1, 0) is called - The method iterates over each of the files that are in Folder1 - It finds File1. File 1 is not a directory (folder) so it creates a new EditorFileButton with text “File1”, increments the drawing position by the height of the button, then adds it to the list of widgets on the screen. - Next, It finds Folder2. Folder 2 **is** a directory, so it calls addChildFiles(prefix + Folder2, Folder2, drawing position) - The method iterates over each of the files that are in Folder2 - It finds File2. File2 is not a directory (folder) so it creates a new EditorFileButton with text “Folder2/File2”, increments the drawing position by the height of the button, then adds it to the list of widgets on the screen. - It finds File3. File3 is not a directory (folder) so it creates a new EditorFileButton with text “Folder2/File3”, increments the drawing position by the height of the button, then adds it to the list of widgets on the screen. - All files in the tree have been searched so the method ends.

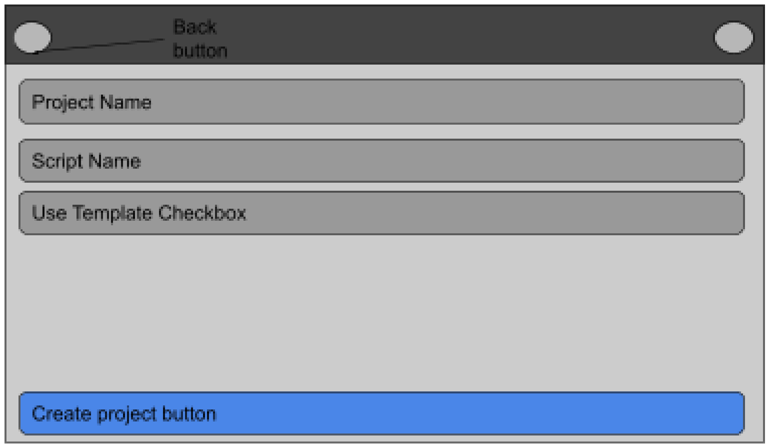
Key Algorithms

```
addChildFiles(String prefix, File dir, int drawPos)
    FOR file in dir DO
      IF file IS DIRECTORY
           addChildFiles(prefix + file.name + '/', file, drawPos)
      ELSE IF file IS FILE
           addButton(prefix + file.name)
           drawPos += buttonHeight
      ENDIF
    ENDFOR
```

Create Project

This screen will enable the user to create a new project.

Design



Attributes

Name	Description
projectNameButton	The text entry widget which will contain the name of the project the user wants to create
mainScriptButton	The text entry widget which will contain the name of the main script for the project the user wants to create
templateButton	The tick button/checkbox widget for if the user wants to use a template file

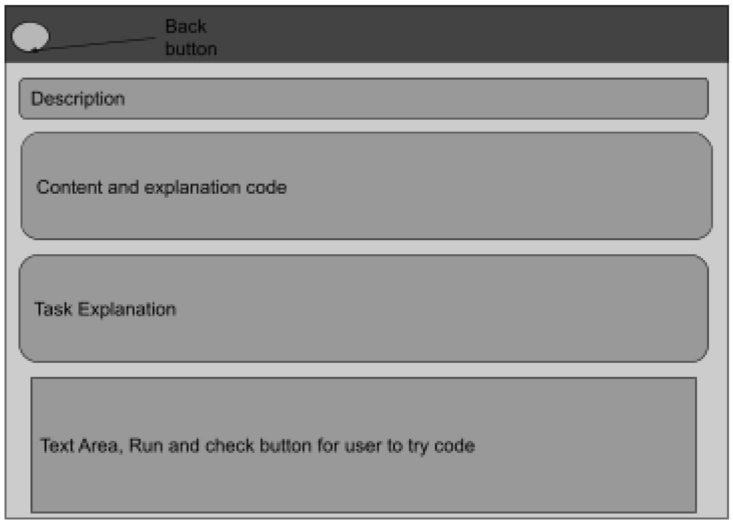
Methods

Name	Description
init	Inherited from the Screen class. This method is called when the screen is displayed. It will initialise all of the widgets on the screen.
render	Inherited from the Screen class. Draws the screen and all the widgets on the screen. It takes in the current render matrix and mouse position as parameters.

Lesson Screen

This screen will be where the user will be able to learn how to program. It will feature a description of the lesson, content of the lesson and a task. The user will be able to create a program for the task in the area below. They will be able to press a button to check their program and then they will be told if they have passed the task or not.

Design


Back button
Description
Content and explanation code
Task Explanation
Text Area, Run and check button for user to try code

Attributes

Name	Description
lesson	The LuaLesson that the user is completing
area	The text area for the users code
submit	The button for the user to submit their code

Methods

Name	Description
init	Inherited from the Screen class. This method is called when the screen is displayed. It will initialise all of the widgets on the screen.
render	Inherited from the Screen class. Draws the screen and all the widgets on the screen. It takes in the current render matrix and mouse position as parameters.

Theme Class

A theme object will hold colour data for a theme. It will contain an accent colour, 3 background tones, a button colour and a font colour. The Theme class will also have a static final Theme called DEFAULT. This will be a default theme for the project.

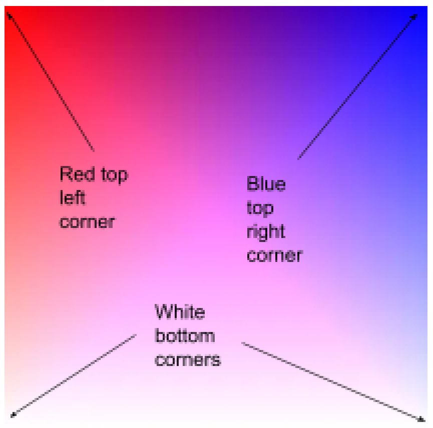
Attributes

Name	Description
accent	The accent colour
background	The background colours
button	The button colour
text	The text colour

Box Colour Scheme

A box colour scheme will contain colours for each vertex of a box. It can be used to render a quad with a specific colour scheme.

Example



This is an example of using a box colour scheme to draw a quad with a gradient. Each vertex is assigned a different colour then when it is drawn the colours are automatically blended by the rendering system to create a quad with a gradient.

Attributes

Name	Description

topLeft	Top left vertex colour
topRight	Top right vertex colour
bottomLeft	Bottom left vertex colour
bottomRight	Bottom right vertex colour

Editor File Button

The Editor File Button will extend net.minecraft.client.gui.widget.ButtonWidget. It will display the name of the file and an icon. The editor file button will be used to select the current file being edited in the Editor.

Document Name

Attributes

Name	Description
file	The file to be contained in the button

Icon Button

The Icon Button will extend net.minecraft.client.gui.widget.ButtonWidget. It will display an icon. The icon button will take a PressAction to perform an operation when clicked.

Attributes

Name	Description
iconName	The name of the icon. The icon will be located in the resources folder so the ResourceManager can easily create a texture from it.

Icon Text Button

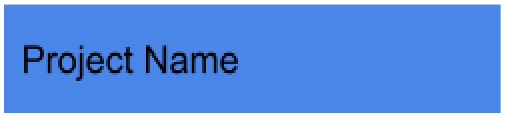
The Icon Text Button will extend net.minecraft.client.gui.widget.ButtonWidget. It will display an icon and some text. The button will take a PressAction to perform an operation when clicked.

Attributes

Name	Description
iconName	The name of the icon. The icon will be located in the resources folder so the ResourceManager can easily create a texture from it.
text	The text to render

Project Toggle Button

The Project Button will extend net.minecraft.client.gui.widget.ButtonWidget. It will display the name of the project and toggle the project when it is clicked.

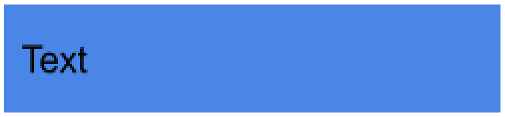


Attributes

Name	Description
project	The project of the button should toggle.

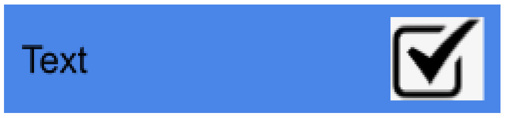
Simple Button

The Simple Button will extend net.minecraft.client.gui.widget.ButtonWidget. It will display some text. The button will take a PressAction to perform an operation when clicked.



Text Tick Button

The Tick button will be enabled or disabled when clicked. It will perform an operation when it is clicked. It will display some text and the tick box.



Attributes

Name	Description
enabled	If the button is enabled

Methods

Name	Description

onClick	Called when the button is clicked. It will change the value of enabled to !enabled (toggling the button)

OneLineDocument

The OneLine document will contain a string (content) and a cursor/pointer. The pointer can be moved throughout the string in order to insert/remove characters at that point in the string

Example

Let's create a OneLineDocument with the string "Hello World" and represent the pointer with |.

The pointer's index is set to 0 by default so our document would look like "|Hello World"

If we call pointerRight() the pointer is incremented by 1 and we have "H|ello World"

If we call backspace() the previous character is deleted and we have "|ello World"

If we call insert("J) "J" is inserted into the character and we have "J|ello World"

Attributes

Name	Description
content	The string that is in the document
pointer	The position of the pointer

Methods

Name	Description
insert	Inserts a string into the document
backspace	Removes the previous character
pointerLeft	Moves pointer left
pointerRight	Moves pointer right

TextEntry

The text entry is a widget which the user will be able to type into. It will contain a OneLineDocument which can be edited when the user types.

Attributes

Name	Description
enabled	If the button is enabled

Methods

Name	Description
onClick	Called when the button is clicked. It will change the value of enabled to !enabled (toggling the button)

TextDocument

The text document is similar to the OneLine document but it has multiple lines. This is more complex as the pointer has to navigate multiple lines. The TextDocument takes in a file, reads the contents and splits it into lines. The position of the pointer is stored as a Vector with the x representing the column and the y representing the row.

Attributes

Name	Description
file	The file that is being edited
lines	The list of lines in the document
pointer	The vector containing the position of the pointer

Methods

Name	Description

save	Writes the content from the document to the file (saving it)
insert	Inserts string into the document
backspace	Deletes the previous character
newLine	Adds a new line into the document. (Like pressing the enter button)
pointerUp	Moves pointer to the line above
pointerDown	Moves pointer to the line below
pointerLeft	Moves pointer left
pointerRight	Moves pointer right
getContent	Creates a single string with the lines by inserting a \n between each line.

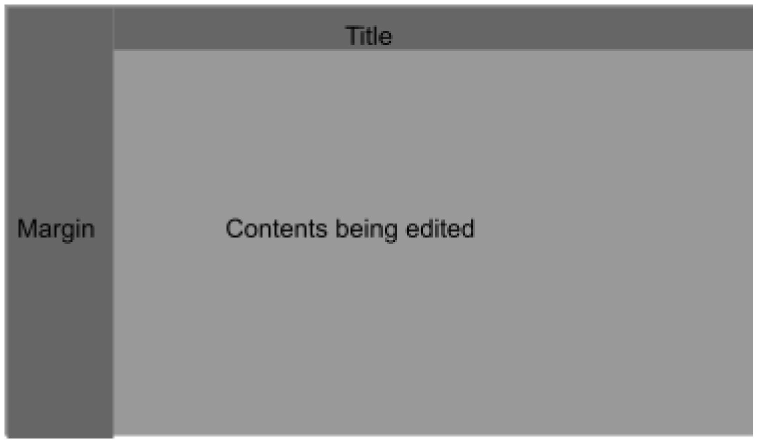
Key Algorithms

```
save()
    // Make sure to return if editing file is null
    // This will avoid any NullPointerExceptions
    IF editing = NULL
       REUTRN
    ENDIF

    // Try to write the content to the file
    TRY IOUtil.writeToOutputStream(
                   ByteArrayInputStream(getContent().getBytes(UTF-8)),
                   FileOutputStream(editing))

    // Handle exception if the file doesn't exist
    CATCH FileNotFoundException
       printException()
    ENDTRY

insert(string)
    // Create a sting builder for easy manipulation
    lineContent = StringBuilder(lines.get(pointer.y))
    // Insert the string at the pointer position in the line
    lineConetent.insert(pointer.x, string)
    // Replace the line in the lines array list
    lines.set(pointer.y, lineContent.toString())
    // Set the new pointer position
    setPointer(pointer.x + string.length, pointer.y)

backspace()
    // This is for deleting a line if we are at the start
    // Moves all the contents of current line to previous line
    IF pointer.x = 0
       // If we are in top corner there is nothing to do so return
       IF pointer.y = 0
             RETURN
       ENDIF

       // Move the pointer to the end of the previous line
       setPointer(lines.get(pointer.y - 1).length() - 1, pointer.y - 1);
       // Insert the line into the previous line
            lineContent = StringBuilder(lines.get(pointer.y));
            lineContent.append(lines.get(pointer.y + 1));
            lines.set(pointer.y, lineContent.toString());
       // Remove the line
            lines.remove(pointer.y + 1)
       // Nothing left to do so return
       RETURN
    ENDIF

    // Delete previous char and change the line in lines
    lineContent = StringBuilder(lines.get(pointer.y))
    lineContent.deletePrevious()
    lines.set(pointer.y, lineContent.toString())
    setPointer(pointer.x - 1, pointer.y)
```

TextArea

This class will extend the ClickableWidget class. The text area is a widget which the user will be able to type into. It will be used for editing text documents (documents with more than one line). The widget will have to display a margin with line numbers, the name of the file being edited, the contents of the file being edited as well as a cursor which will move with the arrow keys. The cursor will have to be rendered in different positions as well as blink to make it more easily visible.

Design



Attributes

Name	Description
document	The TextDocument that is open in the area
parentScreen	The Screen that this widget is present in
lastT	The last time in ms that the screen was rendered.

Methods

Name	Description
renderButton	Inherited and overridden from ClickableWidget. It is called to render the button to the screen.
keyPressed	Inherited and overridden from ClickableWidget. Called when a key is pressed by the user. In this case we will use it to move the pointer or backspace in the document.
charTyped	Inherited and overridden from ClickableWidget. Called when a character key is pressed by the user. In this case we will use it to insert the character into the document.
setEditingFile	Takes in a file as a parameter. Checks if the file is the same as the one being

	edited in the document. If it is different, it saves the document and makes a new document object with the new file.

Key Algorithms

```
renderButton()
    RENDER BACKGROUND

    // If there is no document being edited we can return
    IF document = NULL
       RETURN
    ENDIF

    RENDER MARGIN

    RENDER EACH LINE

    // We only want to display the cursor if the area is focused
    IF FOCUSED
       // We want the cursor to be visible for 500ms then be off for 250ms
       IF currentTime() - last < 500
             RENDER CURSOR
       ELSE IF currentTime() - last >= 750
             last <- currentTime()
       ENDIF
    ENDIF

keyPressed(key)
    // Handle our key actions
    SWITCH(key)
       case BACKSPACE -> document.backspace()
       case UP -> document.pointerUp()
       case DOWN -> document.pointerDown()
       case LEFT -> document.pointerLeft()
       case RIGHT -> document.pointerRight()
       case ENTER -> document.newLine()
    ENDSWITCH

    // Handle Paste
```

```
IF key = V && hasControlDown() && !hasShiftDown() && !hasAltDown()
   // Insert what is in our clipboard
   document.insert(getClipboard)
ENDIF
```

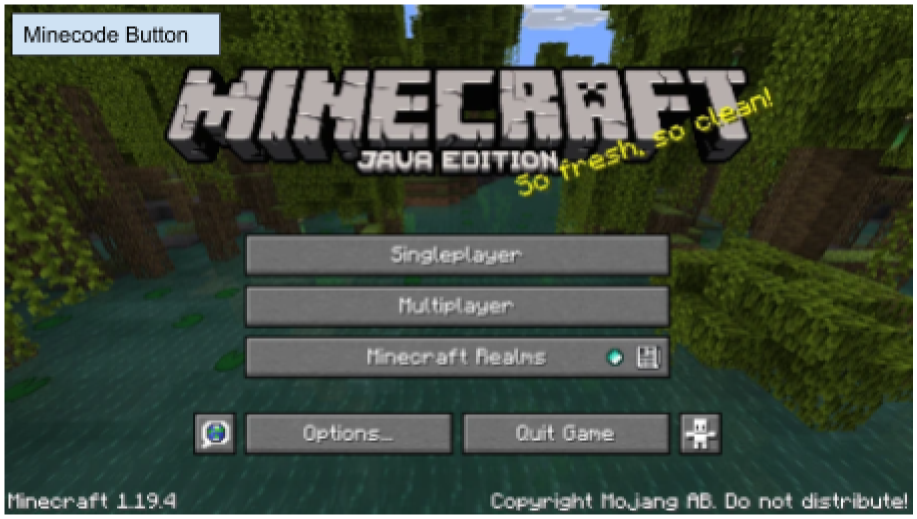
2.2.4 Mixins

Mixin is a useful library made by SpongePowered which will be included in the mod. It is a bytecode manipulation framework which will allow me to modify Minecraft's source code at runtime. I will need this to add buttons to screens so that the user will be able to access the mod's UI.

Adding a button to a GUI Screen

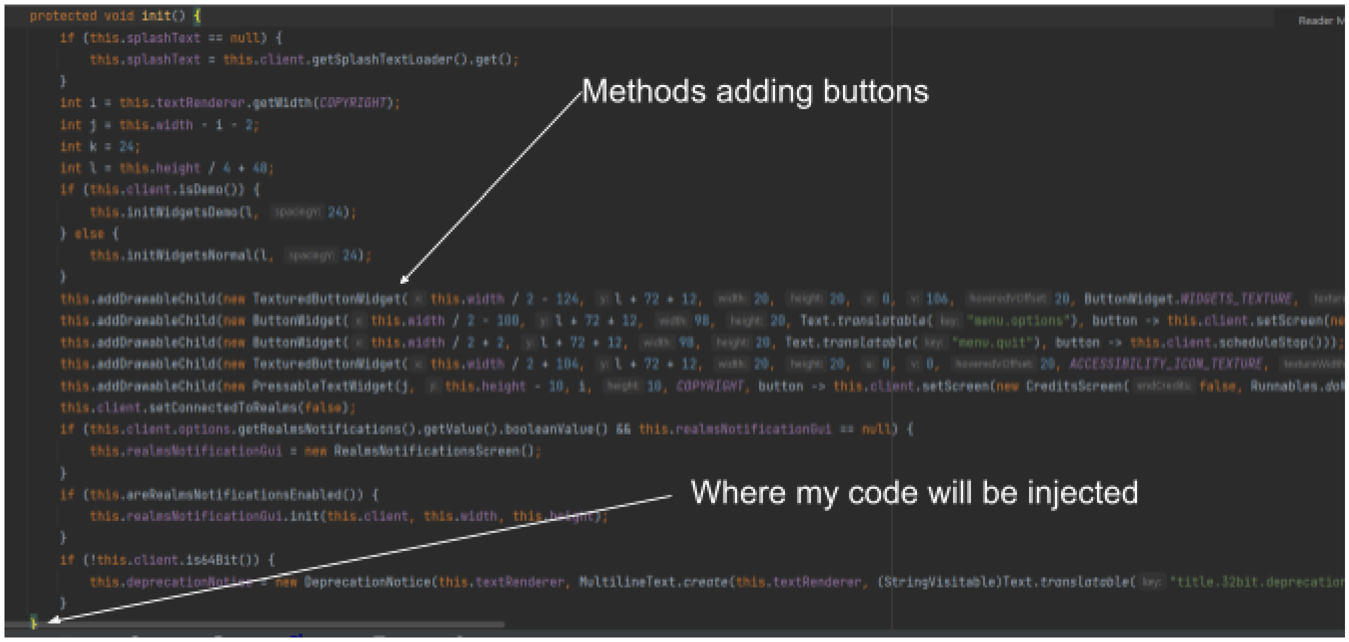
I intend to add a button to the top left of the Main menu so that the user can use my mod.

Design



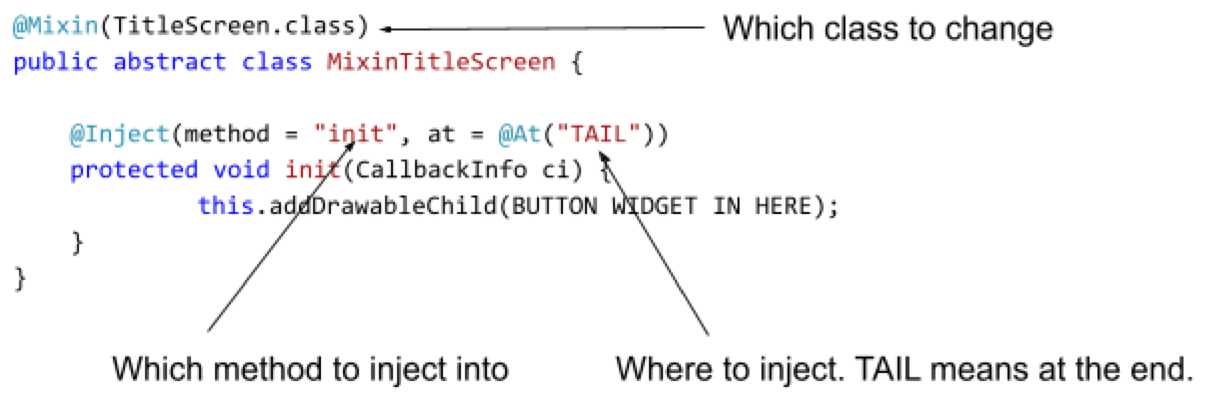
Implementation

To create this feature I will need to create a mixin which injects into the TitleScreen class. Buttons are added in the "init" method of the TitleScreen class so I will need to inject into there.



```java
protected void init() {
    if (this.splashText == null) {
        this.splashText = this.client.getSplashTextLoader().get();
    }
    int i = this.textRenderer.getWidth(COPYRIGHT);
    int j = this.width - i - 2;
    int k = 24;
    int l = this.height / 4 + 48;
    if (this.client.isDemo()) {
        this.initWidgetsDemo(l, spacingY: 24);
    } else {
        this.initWidgetsNormal(l, spacingY: 24);
    }
    this.addDrawableChild(new TexturedButtonWidget( x: this.width / 2 - 124, y: l + 72 + 12, width: 20, height: 20, u: 0, v: 106, hoveredVOffset: 20, ButtonWidget.WIDGETS_TEXTURE, textur
    this.addDrawableChild(new ButtonWidget( x: this.width / 2 - 100, y: l + 72 + 12, width: 98, height: 20, Text.translatable( key: "menu.options"), button -> this.client.setScreen(ne
    this.addDrawableChild(new ButtonWidget( x: this.width / 2 + 2, y: l + 72 + 12, width: 98, height: 20, Text.translatable( key: "menu.quit"), button -> this.client.scheduleStop()));
    this.addDrawableChild(new TexturedButtonWidget( x: this.width / 2 + 104, y: l + 72 + 12, width: 20, height: 20, u: 0, v: 0, hoveredVOffset: 20, ACCESSIBILITY_ICON_TEXTURE, textureWidth
    this.addDrawableChild(new PressableTextWidget(j, y: this.height - 10, i, height: 10, COPYRIGHT, button -> this.client.setScreen(new CreditsScreen( endCredits: false, Runnables.do
    this.client.setConnectedToRealms(false);
    if (this.client.options.getRealmsNotifications().getValue().booleanValue() && this.realmsNotificationGui == null) {
        this.realmsNotificationGui = new RealmsNotificationsScreen();
    }
    if (this.areRealmsNotificationsEnabled()) {
        this.realmsNotificationGui.init(this.client, this.width, this.height);
    }
    if (!this.client.is64Bit()) {
        this.deprecationNotice = new DeprecationNotice(this.textRenderer, MultilineText.create(this.textRenderer, (StringVisitable)Text.translatable( key: "title.32bit.deprecation
    }
}
```

The diagram above shows the “init” method. To add a button the addDrawableChild method needs to be used along with the button to add. I will add my button at the end of this method. I will use something similar to this to create my button:


Which class to change
Which method to inject into
Where to inject. TAIL means at the end.

```java
@Mixin(TitleScreen.class)
public abstract class MixinTitleScreen {

    @Inject(method = "init", at = @At("TAIL"))
    protected void init(CallbackInfo ci) {
            this.addDrawableChild(BUTTON WIDGET IN HERE);
    }
}
```

3 | Technical Solution

3.1 Contents

3.2 Installer

Controller.java

App.java

3.2 Mod

All code for the Mod section of the project can be found on GitHub at: https://github.com/jx4e/NEA-Mod

Minecode.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/Minecode.java

```
package com.github.jx4e.minecode;

import net.fabricmc.api.ModInitializer;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class Minecode implements ModInitializer {
    public static final String MOD_NAME = "Minecode";
    public static final String MOD_VER = "Beta-1.0";
    private static Minecode instance;
    private final Logger logger;

    public Minecode() {
        instance = this;
        logger = LoggerFactory.getLogger(MOD_NAME.toLowerCase() + "-logger");
    }

    @Override
    public void onInitialize() {
        logger.info("Initialising " + MOD_NAME + " " + MOD_VER);
    }

    public static Minecode getInstance() {
        return instance;
    }

    public Logger getLogger() {
        return logger;
    }
}
```

MinecodeClient.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/MinecodeClient.java

```
package com.github.jx4e.minecode;

import com.github.jx4e.minecode.lua.LuaManager;
import com.github.jx4e.minecode.project.ConfigManager;
import com.github.jx4e.minecode.lua.event.EventManager;
import net.fabricmc.api.ClientModInitializer;
import net.fabricmc.api.EnvType;
import net.fabricmc.api.Environment;
import net.minecraft.client.MinecraftClient;

@Environment(EnvType.CLIENT)
```

```java
public class MinecodeClient implements ClientModInitializer {
    public static MinecraftClient mc = MinecraftClient.getInstance();

    @Override
    public void onInitializeClient() {
        Minecode.getInstance().getLogger().info("Client Init");
        ConfigManager.instance().load();
        EventManager.instance();
        LuaManager.instance().init();
    }
}
```

LuaManager.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/lua/LuaManager.java

```java
package com.github.jx4e.minecode.lua;

import com.github.jx4e.minecode.lua.event.LuaEvent;
import com.github.jx4e.minecode.lua.impl.events.ScriptLoadEvent;
import com.github.jx4e.minecode.lua.impl.libs.LuaColorLibrary;
import com.github.jx4e.minecode.lua.impl.libs.LuaDrawLibrary;
import com.github.jx4e.minecode.lua.impl.libs.LuaInputLibrary;
import com.github.jx4e.minecode.lua.impl.libs.LuaTextLibrary;
import org.luaj.vm2.Globals;
import org.luaj.vm2.LuaValue;
import org.luaj.vm2.lib.jse.CoerceJavaToLua;
import org.luaj.vm2.lib.jse.JsePlatform;

import java.util.LinkedList;
import java.util.List;

import static com.github.jx4e.minecode.MinecodeClient.mc;

/**
 * @author Jake (github.com/jx4e)
 * @since 29/06/2022
 **/

public class LuaManager {
    private Globals globals;
    private final List<LuaScript> scripts = new LinkedList<>();

    private LuaManager() {}

    /**
     * Initialise the lua env and add all of the libraries
     */
    public void init() {
        globals = JsePlatform.standardGlobals();

        globals.load(new LuaTextLibrary());
        globals.load(new LuaInputLibrary());
        globals.load(new LuaDrawLibrary());
        globals.load(new LuaColorLibrary());

        // create a global lua variable
        getGlobals().set("mc", CoerceJavaToLua.coerce(mc));
    }

    /**
     * Loads a script to the environment
     * @param script
     */
    public void loadScript(LuaScript script) {
        if (!scripts.contains(script)) {
            try {
                scripts.add(script);
                LuaValue lv = getGlobals().load(script.getContent());

                lv.call();

                script.setEnabled(true);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }

    /**
     * Removes a script and calls the "Exit" event
     * @param script
     */
    public void unloadScript(LuaScript script) {
        if (scripts.contains(script)) {
            scripts.remove(script);
            script.invoke("Exit", null);
            script.setEnabled(false);
        }
    }

    public void postEvent(LuaEvent event) {
        scripts.forEach(script -> {
            script.invoke(event);
        });
    }

    public void postEvent(Object object) {
        scripts.forEach(script -> script.invoke(object));
    }

    public Globals getGlobals() {
        return globals;
    }

    public List<LuaScript> getScripts() {
        return scripts;
    }

    private static LuaManager instance;

    public static LuaManager instance() {
        if (instance == null) instance = new LuaManager();

        return instance;
    }
}
```

LuaScript.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/lua/LuaScript.java

```
package com.github.jx4e.minecode.lua;
```

```java
import com.github.jx4e.minecode.lua.event.LuaEvent;
import com.github.jx4e.minecode.project.IOUtil;
import org.luaj.vm2.LuaValue;
import org.luaj.vm2.lib.jse.CoerceJavaToLua;

import java.io.File;

/**
 * @author Jake
 * @since 03/03/2022
 **/

public class LuaScript {
    private File file;
    private String content;
    private boolean enabled;

    /**
     * Creates a Lua script
     * @param file - .lua file
     */
    public LuaScript(File file) {
        this.enabled = false;
        this.file = file;
        this.content = IOUtil.readFileToString(file);
    }

    /**
     * Load the script into the Lua Environment
     */
    public void load() {
        this.content = IOUtil.readFileToString(file);
        LuaManager.instance().loadScript(this);
    }

    /**
     * Unload the script from the Lua Environment
     */
    public void unload() {
        LuaManager.instance().unloadScript(this);
    }
    /**
     * Invoke an event
     * @param event
     */
    public void invoke(LuaEvent event) {
        invoke(event.getName(), event.getArgs());
    }

    /**
     * Invoke an event
     * @param object
     */
    public void invoke(Object object) {
        invoke(object.getClass().getSimpleName(), new
LuaValue[]{CoerceJavaToLua.coerce(object)});
    }

    /**
     * Invoke an event with arguments
     * @param eventName
     * @param args
     */
    public void invoke(String eventName, LuaValue[] args) {
        LuaValue func = LuaManager.instance().getGlobals().get(eventName);

        if (func == null || func == LuaValue.NIL) return;

        try {
            func.invoke(args);
        } catch (Exception e) {
            e.printStackTrace();
            unload();
        }
    }

    /**
     * Toggles the script
     */
    public void toggle() {
        setEnabled(!isEnabled());

        if (enabled) load();
        else unload();
```

```java
    }

    public void setFile(File file) {
        this.file = file;
    }

    public File getFile() {
        return file;
    }

    public void setEnabled(boolean enabled) {
        this.enabled = enabled;
    }

    public boolean isEnabled() {
        return enabled;
    }

    /**
     * Change the content in the file & reload the script
     * @param content
     */
    public void setContent(String content) {
        IOUtil.writeToFile(file, content);
        this.content = content;
    }

    public String getContent() {
        return content;
    }
}
```

EventManager.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/lua/event/EventManager.java

```java
package com.github.jx4e.minecode.lua.event;
```

```
import com.github.jx4e.minecode.lua.impl.events.Render2DEvent;
import com.github.jx4e.minecode.lua.impl.events.Render3DEvent;
import com.github.jx4e.minecode.lua.impl.events.TickEvent;
import com.github.jx4e.minecode.lua.LuaManager;
import net.fabricmc.fabric.api.client.event.lifecycle.v1.ClientTickEvents;
import net.fabricmc.fabric.api.client.rendering.v1.HudRenderCallback;
import net.fabricmc.fabric.api.client.rendering.v1.WorldRenderEvents;

/**
 * @author jake
 * @since 08/04/2022
 */
public class EventManager {
    private EventManager() {
        initEvents();
    }

    /**
     * Initialise all the events (Register them with the fabric event system)
     */
    private void initEvents() {
        HudRenderCallback.EVENT.register((matrixStack, delta) -> {
            Render2DEvent event = new Render2DEvent(matrixStack);
            LuaManager.instance().postEvent(event);
        });

        WorldRenderEvents.BEFORE_ENTITIES.register((context) -> {
            Render3DEvent event = new Render3DEvent(context,
Render3DEvent.Type.BeforeEntity);
            LuaManager.instance().postEvent(event);
        });

        WorldRenderEvents.AFTER_ENTITIES.register((context) -> {
            Render3DEvent event = new Render3DEvent(context,
Render3DEvent.Type.AfterEntity);
            LuaManager.instance().postEvent(event);
        });

        ClientTickEvents.END_CLIENT_TICK.register(client -> {
            TickEvent event = new TickEvent();
            LuaManager.instance().postEvent(event);
        });
    }
    private static EventManager instance;

    public static EventManager instance() {
        if (instance == null) instance = new EventManager();
        return instance;
    }
}
```

LuaEvent.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/lua/event/LuaEvent.java

```
package com.github.jx4e.minecode.lua.event;

import org.luaj.vm2.LuaValue;

/**
 * @author Jake (github.com/jx4e)
 * @since 08/06/2022
 **/

public abstract class LuaEvent {
    /**
     * Name of the event to be called in the Lua script
     */
    private final String name;

    /**
     * Arguments to be parsed into the function
     */
    private final LuaValue[] args;

    public LuaEvent(String name, LuaValue[] args) {
        this.name = name;
        this.args = args;
    }

    public String getName() {
        return name;
    }

    public LuaValue[] getArgs() {
        return args;
    }
}
```

Render2DEvent.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/lua/impl/events/Render2DEvent.java

```
package com.github.jx4e.minecode.lua.impl.events;

import com.github.jx4e.minecode.lua.event.LuaEvent;
import net.minecraft.client.util.math.MatrixStack;
import org.luaj.vm2.LuaValue;
import org.luaj.vm2.lib.jse.CoerceJavaToLua;

/**
 * @author Jake (github.com/jx4e)
 * @since 09/11/2022
 **/

public class Render2DEvent extends LuaEvent {
    /**
     * Render event to be called when the ingame HUD is rendered
     * @param matrix - the projection matrix
     */
    public Render2DEvent(MatrixStack matrix) {
        super("Render2DEvent", new LuaValue[]{
                CoerceJavaToLua.coerce(matrix)
        });
    }
}
```

Render3DEvent.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/lua/impl/events/Render3DEvent.java

```java
package com.github.jx4e.minecode.lua.impl.events;

import com.github.jx4e.minecode.lua.event.LuaEvent;
import net.fabricmc.fabric.api.client.rendering.v1.WorldRenderContext;
import org.luaj.vm2.LuaValue;
import org.luaj.vm2.lib.jse.CoerceJavaToLua;

/**
 * @author Jake (github.com/jx4e)
 * @since 09/11/2022
 **/

public class Render3DEvent extends LuaEvent {
    /**
     * Render event to be called when the world is rendered
     * @param context - the world render context
     * @param type - If it's before or after entities
     */
    public Render3DEvent(WorldRenderContext context, Type type) {
        super("Render3DEvent", new LuaValue[]{
                CoerceJavaToLua.coerce(context), LuaValue.valueOf(type.name())
        });
    }

    public enum Type {
        BeforeEntity, AfterEntity
    }
}
```

Tick Event

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/lua/impl/events/TickEvent.java

```
package com.github.jx4e.minecode.lua.impl.events;

import com.github.jx4e.minecode.lua.event.LuaEvent;
import org.luaj.vm2.LuaValue;

/**
 * @author Jake (github.com/jx4e)
 * @since 09/11/2022
 **/

public class TickEvent extends LuaEvent {
    /**
     * An event which is called every game tick (every 1/20th of a second)
     */
    public TickEvent() {
        super("TickEvent", new LuaValue[]{});
    }
}
```

LuaColorLibrary.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/lua/impl/libs/LuaColorLibrary.java

```
package com.github.jx4e.minecode.lua.impl.libs;

import com.github.jx4e.minecode.lua.library.LuaFunction;
import com.github.jx4e.minecode.lua.library.LuaLibrary;
import org.luaj.vm2.lib.jse.CoerceJavaToLua;

import java.awt.*;

/**
 * A Lua library based around color
 * @author Jake (github.com/jx4e)
 * @since 02/11/2022
 **/

public class LuaColorLibrary extends LuaLibrary {
    public LuaColorLibrary() {
        super("color", new LuaFunction[]{
                new LuaFunction(
                        "rgb",
                        varargs -> {
                            // Convert Our Values
                            int r = varargs.arg(1).toint();
                            int g = varargs.arg(2).toint();
                            int b = varargs.arg(3).toint();

                            Color color = new Color(r, g, b);

                            // Return Color
                            return CoerceJavaToLua.coerce(color);
                        }
                ),
                new LuaFunction(
                        "rgba",
                        varargs -> {
                            // Convert Our Values
                            int r = varargs.arg(1).toint();
                            int g = varargs.arg(2).toint();
                            int b = varargs.arg(3).toint();
                            int a = varargs.arg(4).toint();

                            Color color = new Color(r, g, b, a);

                            // Return Color
                            return CoerceJavaToLua.coerce(color);
                        }
                ),
                new LuaFunction(
                        "hsb",
                        varargs -> {
                            // Convert Our Values
                            float h = varargs.arg(1).tofloat();
                            float s = varargs.arg(2).tofloat();
                            float b = varargs.arg(3).tofloat();

                            Color color = new Color(Color.HSBtoRGB(h, s, b));

                            // Return Color
                            return CoerceJavaToLua.coerce(color);
                        }
                )
```

```
        });
    }
}
```

LuaDrawLibrary.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/lua/impl/libs/LuaDrawLibrary.java

```
package com.github.jx4e.minecode.lua.impl.libs;

import com.github.jx4e.minecode.lua.library.LuaFunction;
import com.github.jx4e.minecode.lua.library.LuaLibrary;
import com.github.jx4e.minecode.rendering.RenderManager;
import com.github.jx4e.minecode.rendering.ResourceManager;
import net.minecraft.client.texture.NativeImageBackedTexture;
import net.minecraft.client.util.math.MatrixStack;
import org.luaj.vm2.LuaValue;
import org.luaj.vm2.lib.jse.CoerceLuaToJava;

import java.awt.*;

/**
 * @author Jake (github.com/jx4e)
 * @since 02/11/2022
 **/

public class LuaDrawLibrary extends LuaLibrary {
    public LuaDrawLibrary() {
        super("draw", new LuaFunction[]{
                new LuaFunction(
                        "box",
                        varargs -> {
                            // Convert Our Values
                            MatrixStack matrix = (MatrixStack) CoerceLuaToJava.coerce(varargs.arg(1), MatrixStack.class);
                            int x = varargs.arg(2).toint();
                            int y = varargs.arg(3).toint();
                            int width = varargs.arg(4).toint();
                            int height = varargs.arg(5).toint();

                            if (varargs.arg(6).isint()) {
                                // Draw the box
                                RenderManager.instance().getRenderer().box(matrix, x, y, width, height,
                                        new Color(varargs.arg(6).toint()));
                            } else {
                                RenderManager.instance().getRenderer().box(matrix, x, y, width, height,
                                        (Color) CoerceLuaToJava.coerce(varargs.arg(6), Color.class));
                            }

                            // Return Nil
                            return LuaValue.NIL;
                        }
                ),
                new LuaFunction(
                        "texture",
                        varargs -> {
                            // Convert Our Values
                            MatrixStack matrix = (MatrixStack) CoerceLuaToJava.coerce(varargs.arg(1), MatrixStack.class);
                            String textureName = varargs.arg(2).tojstring();
                            int x = varargs.arg(3).toint();
                            int y = varargs.arg(4).toint();
                            int width = varargs.arg(5).toint();
                            int height = varargs.arg(6).toint();

                            // Draw the texture
                            NativeImageBackedTexture texture = ResourceManager.instance().getNativeImageTexture(textureName);
                            RenderManager.instance().getRenderer().image(matrix, texture.getGlId(), x, y, width, height);
                            texture.close();

                            // Return Nil
                            return LuaValue.NIL;
                        }
                ),
        });
    }
}
```

LuaTextLibrary.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/lua/impl/libs/LuaTextLibrary.java

```
package com.github.jx4e.minecode.lua.impl.libs;

import com.github.jx4e.minecode.lua.library.LuaFunction;
import com.github.jx4e.minecode.lua.library.LuaLibrary;
import com.github.jx4e.minecode.rendering.RenderManager;
import net.minecraft.client.util.math.MatrixStack;
import org.luaj.vm2.LuaInteger;
import org.luaj.vm2.LuaValue;
import org.luaj.vm2.lib.jse.CoerceLuaToJava;

import java.awt.*;

/**
 * @author Jake (github.com/jx4e)
 * @since 02/11/2022
 **/

public class LuaTextLibrary extends LuaLibrary {
    public LuaTextLibrary() {
        super("text", new LuaFunction[]{
                new LuaFunction(
                        "write",
                        varargs -> {
                            // Convert Our Values
                            MatrixStack matrix = (MatrixStack) CoerceLuaToJava.coerce(varargs.arg(1), MatrixStack.class);
                            String text = varargs.arg(2).tojstring();
                            int x = varargs.arg(3).toint();
                            int y = varargs.arg(4).toint();

                            // Draw the text
                            if (varargs.arg(5).isint()) {
                                RenderManager.instance().getDefaultFontRenderer().draw(matrix, text, x, y,
                                        varargs.arg(5).toint());
                            } else {
                                RenderManager.instance().getDefaultFontRenderer().draw(matrix, text, x, y,
                                        ((Color) CoerceLuaToJava.coerce(varargs.arg(5), Color.class)).getRGB());
                            }

                            // No Return Value
                            return LuaValue.NIL;
```

```
                    }
                ),
                new LuaFunction(
                        "width",
                        varargs -> {
                            // Convert Our Values
                            String text = varargs.arg(1).tojstring();

                            // Return Value
                            return LuaInteger.valueOf(RenderManager.instance().getDefaultFontRenderer().getWidth(text));
                        }
                ),
                new LuaFunction(
                        "height",
                        varargs -> {
                            // Return Value
                            return LuaValue.valueOf(RenderManager.instance().getDefaultFontRenderer().fontHeight);
                        }
                )
        });
    }
}
```

IFunction.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/lua/library/IFunction.java

```
package com.github.jx4e.minecode.lua.library;

import org.luaj.vm2.LuaValue;
import org.luaj.vm2.Varargs;

/**
 * An interface which recieves an argument, performs a function then returns a value of
type T
 * @author Jake
 * @since 02/03/2022
 **/

public interface IFunction<T> {
    T executesAndReturns(Varargs varargs);
}
```

LuaFunction.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/lua/library/LuaFunction.java

```java
package com.github.jx4e.minecode.lua.library;

import com.github.jx4e.minecode.lua.library.IFunction;
import org.luaj.vm2.LuaValue;
import org.luaj.vm2.Varargs;

/**
 * @author Jake
 * @since 02/03/2022
 **/

public class LuaFunction {
    /**
     * Name of the function
     */
    private final String name;

    /**
     * The IFunction to execute
     */
    private IFunction<LuaValue> function;

    public LuaFunction(String name, IFunction<LuaValue> function) {
        this.name = name;
        this.function = function;
    }

    public String getName() {
        return name;
    }

    public LuaValue execute(Varargs varargs) {
        return function.executesAndReturns(varargs);
    }
}
```

LuaLibrary.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/lua/library/LuaLibrary.java

```
package com.github.jx4e.minecode.lua.library;

import com.github.jx4e.minecode.lua.library.LuaFunction;
import org.luaj.vm2.LuaValue;
import org.luaj.vm2.Varargs;
import org.luaj.vm2.lib.TwoArgFunction;
import org.luaj.vm2.lib.VarArgFunction;

/**
 * @author Jake
 * @since 02/03/2022
 **/

public class LuaLibrary extends TwoArgFunction {
    /**
     * Name of the library
     */
    private String name;

    /**
     * The functions the library has
     */
    private LuaFunction[] functions;

    public LuaLibrary(String name, LuaFunction[] functions) {
        this.name = name;
        this.functions = functions;
    }

    /**
     * This is called when the library is imported
     * @param modname
     * @param env - the environment
     * @return this library with all the functions
     */
    @Override
    public LuaValue call(LuaValue modname, LuaValue env) {
        LuaValue library = tableOf();

        for (LuaFunction function : functions) {
            library.set(function.getName(), new Function(function));
        }

        env.set(name, library);
        env.get("package").get("loaded").set(name, library);
        return library;
    }

    /**
     * The function class which will be called
     */
    static class Function extends VarArgFunction {
        private LuaFunction function;

        public Function(LuaFunction function) {
            this.function = function;
        }

        @Override
        public Varargs onInvoke(Varargs varargs) {
            return varargsOf(new LuaValue[]{function.execute(varargs)});
        }

        @Override
        public String name() {
            return function.getName();
        }
    }
}
```

MixinGameMenuScreen.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/mixins/MixinGameMenuScreen.java

```
package com.github.jx4e.minecode.mixins;
```

```java
import com.github.jx4e.minecode.rendering.screens.EditorMainMenu;
import com.github.jx4e.minecode.rendering.screens.QuickToggleMenu;
import net.minecraft.client.gui.screen.GameMenuScreen;
import net.minecraft.client.gui.screen.Screen;
import net.minecraft.client.gui.widget.ButtonWidget;
import net.minecraft.text.Text;
import org.spongepowered.asm.mixin.Mixin;
import org.spongepowered.asm.mixin.injection.At;
import org.spongepowered.asm.mixin.injection.Inject;
import org.spongepowered.asm.mixin.injection.callback.CallbackInfo;

/**
 * @author Jake (github.com/jx4e)
 * @since 11/06/2022
 **/

@Mixin(GameMenuScreen.class)
public abstract class MixinGameMenuScreen extends Screen {
    protected MixinGameMenuScreen(Text title) {
        super(title);
    }

    /**
     * adds 2 buttons to the game menu screen at the end of the init method
     * @param ci
     */
    @Inject(method = "init", at = @At("TAIL"))
    protected void init(CallbackInfo ci) {
        this.addDrawableChild(
                new ButtonWidget(
                        10,
                        10,
                        98, 20,
                        Text.of("Editor"),
                        button -> this.client.setScreen(EditorMainMenu.getInstance())
                )
        );

        this.addDrawableChild(
                new ButtonWidget(
                        10,
                        40,
                        98, 20,
                        Text.of("Quick-Toggle"),
                        button -> this.client.setScreen(QuickToggleMenu.getInstance())
                )
        );
    }
}
```

MixinTitleScreen.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/mixins/MixinTitleScreen.java

```java
package com.github.jx4e.minecode.mixins;

import com.github.jx4e.minecode.rendering.screens.EditorMainMenu;
import net.minecraft.client.gui.screen.Screen;
import net.minecraft.client.gui.screen.TitleScreen;
import net.minecraft.client.gui.widget.ButtonWidget;
import net.minecraft.text.Text;
import org.spongepowered.asm.mixin.Mixin;
import org.spongepowered.asm.mixin.injection.At;
import org.spongepowered.asm.mixin.injection.Inject;
import org.spongepowered.asm.mixin.injection.callback.CallbackInfo;

/**
 * @author Jake (github.com/jx4e)
 * @since 11/06/2022
 **/

@Mixin(TitleScreen.class)
public abstract class MixinTitleScreen extends Screen {
    protected MixinTitleScreen(Text title) {
        super(title);
    }

    /**
     * Adds button to the screen
     * @param ci
     */
    @Inject(method = "init", at = @At("TAIL"))
    protected void init(CallbackInfo ci) {
        this.addDrawableChild(
                new ButtonWidget(
                        10,
                        10,
                        98, 20,
                        Text.of("Editor"),
                        button -> this.client.setScreen(EditorMainMenu.getInstance())
                )
        );
    }
}
```

ConfigManager.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/project/ConfigManager.java

```java
package com.github.jx4e.minecode.project;

import com.github.jx4e.minecode.Minecode;
import com.google.gson.GsonBuilder;
import com.google.gson.JsonArray;
import com.google.gson.JsonObject;
import net.minecraft.util.JsonHelper;

import java.io.File;
import java.net.MalformedURLException;
import java.net.URL;
import java.util.Arrays;
import java.util.List;

import static com.github.jx4e.minecode.MinecodeClient.mc;

/**
 * @author Jake (github.com/jx4e)
 * @since 29/06/2022
 **/

public class ConfigManager {
    private final File directory = new File(mc.runDirectory, Minecode.MOD_NAME.toLowerCase());
    private final File resources = new File(directory, "resources");
    private final File projects = new File(directory, "projects");
    private final File lessons = new File(directory, "lessons");

    private ConfigManager() {}

    /**
     * Load the config
     */
    public void load() {
        Minecode.getInstance().getLogger().info("Creating Directories...");
        setupDirectory(directory);
        setupDirectory(resources);
        setupDirectory(projects);

        Minecode.getInstance().getLogger().info("Downloading Resources...");
        loadResources();

        Minecode.getInstance().getLogger().info("Loading Projects And Lessons...");
        loadProjectsAndLessons();
    }
    /**
     * Create a directory
     * @param directory - location to make the dir
     * @return if the directory already exists
     */
    private boolean setupDirectory(File directory) {
        if (!directory.exists() || directory.isFile()) {
            directory.delete();
            directory.mkdir();

            return false;
        }

        return true;
    }

    /**
     * Download all the resources from the resources json
     */
    public void loadResources() {
        String json = IOUtil.readResourceStream("/minecode.resources.json");

        if (json == null) return;

        JsonArray resourceArray = JsonHelper.getArray(JsonHelper.deserialize(new
GsonBuilder().setPrettyPrinting().disableHtmlEscaping().create(), json, JsonObject.class), "resources");

        List<String> fileNames = Arrays.stream(resources.listFiles()).map(file -> file.getName()).toList();

        resourceArray.forEach(jsonElement -> {
            String name = JsonHelper.getString(jsonElement.getAsJsonObject(), "name");

            if (fileNames.contains(name)) return;

            String url = JsonHelper.getString(jsonElement.getAsJsonObject(), "url");

            try {
                IOUtil.downloadFile(new URL(url), new File(resources, name));
            } catch (MalformedURLException e) {
                throw new RuntimeException(e);
            }
        });
    }

    public void loadProjects() {
        ProjectManager.instance().createProjectsFromDirectory(projects);
    }

    /**
     * Load all the projects from files to LuaProject objects
     */
    public void loadProjectsAndLessons() {
        ProjectManager.instance().createProjectsFromDirectory(projects);
        LessonManager.instance().createLessonsFromDirectory(lessons);
    }

    public File getResources() {
        return resources;
    }

    public File getProjects() {
        return projects;
    }

    private static ConfigManager instance;

    public static ConfigManager instance() {
        if (instance == null) instance = new ConfigManager();

        return instance;
    }
}
```

IOUtil.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/project/IOUtil.java

```
package com.github.jx4e.minecode.project;

import javax.annotation.Nullable;
import java.io.*;
import java.net.URL;
import java.nio.charset.StandardCharsets;

/**
 * @author Jake (github.com/jx4e)
 * @since 29/06/2022
 **/

public class IOUtil {
    /**
     * Writes to byte array
     * @param in
     * @return
     */
    public static ByteArrayOutputStream writeToByteArray(InputStream in) {
        try (ByteArrayOutputStream out = new ByteArrayOutputStream()) {
            byte[] data = new byte[1024];
            int read;
            while ((read = in.read(data, 0, 1024)) != -1) {
                out.write(data, 0, read);
            }

            return out;
        } catch (IOException e) {
            e.printStackTrace();
        }

        return null;
    }

    /**
     * Writes inputstream to outputstream
     * @param in
     * @param out
     */
    public static void writeToOutputStream(InputStream in, OutputStream out) {
        try {
            byte[] data = new byte[1024];
            int read;
            while ((read = in.read(data, 0, 1024)) != -1) {
                out.write(data, 0, read);
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    /**
     * Reads a resource stream from a path
     * @param path
     * @return
     */
    @Nullable
    public static String readResourceStream(String path) {
        String content = null;

        try (InputStream in = IOUtil.class.getResourceAsStream(path)) {
            content = writeToByteArray(in).toString();
        } catch (IOException e) {
            e.printStackTrace();
        }

        return content;
    }
```

```java
    /**
     * Reads file to string
     * @param file
     * @return
     */
    public static String readFileToString(File file) {
        try (FileInputStream in = new FileInputStream(file)) {
            return IOUtil.writeToByteArray(in).toString();
        } catch (IOException e) {
            throw new RuntimeException();
        }
    }

    /**
     * Writes to file
     * @param file
     * @param content
     */
    public static void writeToFile(File file, String content) {
        try (FileOutputStream out = new FileOutputStream(file);
             ByteArrayInputStream in = new
ByteArrayInputStream(content.getBytes(StandardCharsets.UTF_8));
        ) {
           IOUtil.writeToOutputStream(in , out);
        } catch (IOException e) {
            throw new RuntimeException();
        }
    }

    /**
     * Downloads file from specified url
     * @param inputURL
     * @param outputFile
     * @return
     */
    public static boolean downloadFile(URL inputURL, File outputFile) {
        if (inputURL == null || outputFile == null) return false;

        try (BufferedInputStream in = new BufferedInputStream(inputURL.openStream());
             FileOutputStream out = new FileOutputStream(outputFile)) {
            IOUtil.writeToOutputStream(in, out);
        } catch (IOException e) {
            e.printStackTrace();
            return false;
        }
        return true;
    }

    private IOUtil() {
        throw new UnsupportedOperationException();
    }
}
```

LessonManager.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/project/LessonManager.java

```
package com.github.jx4e.minecode.project;

import java.io.File;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

/**
 * @author Jake (github.com/jx4e)
 * @since 29/06/2022
 **/

public class LessonManager {
    /**
     * List of all the lessons
     */
    private final List<LuaLesson> lessons = new ArrayList<>();

    private LessonManager() {}

    /**
     * Creates Lualessons from a directory.
     * @param dir - the directory to create the lessons from
     */
    public void createLessonsFromDirectory(File dir) {
        if (!dir.exists() || !dir.isDirectory()) return;
        Arrays.stream(dir.listFiles()).forEach(file -> {
            try {
                lessons.add(new LuaLesson(file));
            } catch (Exception e) {
                // If exception thrown we just do not add the project
            }
        });
    }

    public List<LuaLesson> getLessons() {
        return lessons;
    }

    private static LessonManager instance;

    public static LessonManager instance() {
        if (instance == null) instance = new LessonManager();

        return instance;
    }
}
```

LuaLesson.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/project/LuaLesson.java

```
package com.github.jx4e.minecode.project;

import com.github.jx4e.minecode.lua.LuaScript;
import com.google.gson.GsonBuilder;
import com.google.gson.JsonArray;
import com.google.gson.JsonObject;
import net.minecraft.util.JsonHelper;

import java.io.File;
import java.util.ArrayList;
import java.util.List;

public class LuaLesson {
    // Files
    private final File dir;
    private final File lessonFile;
    private final File mainScriptFile;

    // Lesson Info
    private final String name;
    private final String description;
    private final List<LessonContent> content;
    private final List<LessonContent> tasks;
    private final LuaScript script;

    /**
     * Takes a directory and makes the lua lesson from it
     * @param dir
     * @throws Exception
     */
    public LuaLesson(File dir) throws Exception {
        this.dir = dir;

        // Get the lesson.json file
        this.lessonFile = new File(dir, "lesson.json");
        this.mainScriptFile = new File(dir, "main.lua");

        // Read the lesson file
        String json = IOUtil.readFileToString(lessonFile);

        // Check if the file has been read correctly
        if (json == null) throw new Exception("Invalid Lesson File");

        JsonObject jsonObject = JsonHelper.deserialize(new
GsonBuilder().setPrettyPrinting().disableHtmlEscaping().create(), json, JsonObject.class);

        // Get the name and description from the json
        name = JsonHelper.getString(jsonObject, "name");
        description = JsonHelper.getString(jsonObject, "description");

        // Initialise array
        content = new ArrayList<>();

        // Iterate over the "content" array and create lesson content records
        // Adds them all to content
        JsonArray contentArray = JsonHelper.getArray(jsonObject, "content");
        contentArray.forEach(element -> {
            if (element.isJsonObject()) {
                LessonContent contentObject = new LessonContent(
                        JsonHelper.getString(element.getAsJsonObject(), "text"),
                        JsonHelper.getString(element.getAsJsonObject(), "code")
                );
                content.add(contentObject);
            }
        });

        // Create a JSON array of "tasks"
        JsonArray taskArray = JsonHelper.getArray(jsonObject, "tasks");
        tasks = new ArrayList<>();

        // Add all the tasks to our array
        taskArray.forEach(element -> {
            if (element.isJsonObject()) {
                LessonContent contentObject = new LessonContent(
                        JsonHelper.getString(element.getAsJsonObject(), "text"),
                        JsonHelper.getString(element.getAsJsonObject(), "code")
                );
                tasks.add(contentObject);
            }
        });

        // Lets create our lua script
        script = new LuaScript(mainScriptFile);
    }

    public File getLessonFile() {
        return lessonFile;
    }

    public File getDir() {
        return dir;
    }

    public File getMainScriptFile() {
        return mainScriptFile;
    }

    public String getName() {
        return name;
    }

    public String getDescription() {
        return description;
    }

    public List<LessonContent> getContent() {
        return content;
    }

    public List<LessonContent> getTasks() {
        return tasks;
    }

    public LuaScript getScript() {
        return script;
    }

    public record LessonContent(String text, String code) {
        @Override
        public String text() {
            return text;
        }

        @Override
        public String code() {
            return code;
        }
    }
}
```

LuaProject.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/project/LuaProject.java

```
package com.github.jx4e.minecode.project;

import com.github.jx4e.minecode.lua.LuaScript;
import com.google.gson.GsonBuilder;
import com.google.gson.JsonObject;
import net.minecraft.util.JsonHelper;
import java.io.*;
import java.util.Arrays;
import java.util.Objects;

public class LuaProject {
    // Files
    private File dir;
    private File projectFile;
    private File mainScriptFile;

    // Project Info
    private String name;
    private boolean enabled;
    private String mainScriptName;

    // Scripts
    private LuaScript mainScript;

    /**
     * Makes LuaProject from file
     * @param dir
     * @throws Exception
     */
    public LuaProject(File dir) throws Exception {
        this.dir = dir;

        Arrays.stream(Objects.requireNonNull(dir.listFiles())).forEach(file -> {
            if ("project.json".equals(file.getName())) {
                projectFile = file;
            }
        });

        // Read the project file
        String json = IOUtil.readFileToString(projectFile);

        if (json == null) throw new Exception("Invalid Project File");

        JsonObject jsonObject = JsonHelper.deserialize(new
GsonBuilder().setPrettyPrinting().disableHtmlEscaping().create(), json, JsonObject.class);

        name = JsonHelper.getString(jsonObject, "name");
        enabled = JsonHelper.getBoolean(jsonObject, "enabled");
        mainScriptName = JsonHelper.getString(jsonObject, "main-script");

        mainScript = new LuaScript(mainScriptFile = new File(dir, mainScriptName));
    }

    /**
     * Creates a new file object for the mainScript
     */
    public void reloadScript() {
        mainScript = new LuaScript(mainScriptFile = new File(dir, mainScriptName));
    }

    public File getDir() {
        return dir;
    }

    public File getProjectFile() {
        return projectFile;
    }

    public File getMainScriptFile() {
        return mainScriptFile;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public boolean isEnabled() {
        return enabled;
    }

    public void toggle() {
        setEnabled(!isEnabled());
    }

    public void setEnabled(boolean enabled) {
        this.enabled = enabled;
        if (enabled) {
            mainScript.load();
        } else {
            mainScript.unload();
        }
    }

    public String getMainScriptName() {
        return mainScriptName;
    }

    public void setMainScriptName(String mainScriptName) {
        this.mainScriptName = mainScriptName;
    }

    public LuaScript getMainScript() {
        return mainScript;
    }

    public void setMainScript(LuaScript mainScript) {
        this.mainScript = mainScript;
    }

    @Override
    public String toString() {
        return "LuaProject{" +
                "name='" + name + '\'' +
                ", enabled=" + enabled +
                ", mainScriptName='" + mainScriptName + '\'' +
                '}';
    }
}
```

ProjectManager.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/project/ProjectManager.java

```
package com.github.jx4e.minecode.project;
```

```java
import com.github.jx4e.minecode.Minecode;
import com.github.jx4e.minecode.rendering.ResourceManager;
import com.google.gson.JsonObject;

import java.io.*;
import java.nio.charset.StandardCharsets;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;

/**
 * @author Jake (github.com/jx4e)
 * @since 29/06/2022
 **/

public class ProjectManager {
    private final List<LuaProject> projects = new ArrayList<>();

    private ProjectManager() {

    }

    /**
     * Create projects from directory and adds them to projects
     * @param dir
     */
    public void createProjectsFromDirectory(File dir) {
        if (!dir.exists() || !dir.isDirectory()) return;

        Arrays.stream(dir.listFiles()).forEach(file -> {
            try {
                projects.add(new LuaProject(file));
            } catch (Exception e) {
                // If exception thrown we just do not add the project
            }
        });
    }

    /**
     * Creates project. Used when the user is on the EditorCreateProject screen
     * @param projectName
     * @param mainScriptName
     * @param useTemplate
     */
    public void createProject(String projectName, String mainScriptName, boolean
useTemplate) {
        File projectsDirectory = ConfigManager.instance().getProjects();

        // Dir to make this new project in
        File projectDirectory = new File(projectsDirectory, projectName);
        if (projectDirectory.exists()) {
            Minecode.getInstance().getLogger().info("Project Already Exists!");
            return;
        }
        projectDirectory.mkdirs();

        // Project JSON file created
        Minecode.getInstance().getLogger().info("Creating project json file");
        File projectJSON = new File(projectDirectory, "project.json");
        try {
            projectJSON.createNewFile();
        } catch (IOException e) {
            throw new RuntimeException(e);
        }

        // Write our json and add the properties that we need
        JsonObject jsonObject = new JsonObject();
        jsonObject.addProperty("name", projectName);
        jsonObject.addProperty("enabled", false);
        jsonObject.addProperty("main-script", mainScriptName);

        // Write it to the file
        try (ByteArrayInputStream in = new
ByteArrayInputStream(jsonObject.toString().getBytes(StandardCharsets.UTF_8));
                FileOutputStream out = new FileOutputStream(projectJSON)) {
            IOUtil.writeToOutputStream(in, out);
        } catch (IOException e) {

        }

        // Now we make the scripts file
        Minecode.getInstance().getLogger().info("Creating main script");
        File mainScript = new File(projectDirectory, mainScriptName);
        try {
            mainScript.createNewFile();
        } catch (IOException e) {
            throw new RuntimeException(e);
        }

        // If we want a template copy it from the template file
        if (useTemplate) {
            File templateFile = new File(ConfigManager.instance().getResources(),
"template.lua");
            String content = IOUtil.readFileToString(templateFile);
            IOUtil.writeToFile(mainScript, content);
        }

        // Reload the projects
        Minecode.getInstance().getLogger().info("Reloading all projects");
        projects.clear();
        ConfigManager.instance().loadProjects();
    }

    public List<LuaProject> getProjects() {
        return projects;
    }

    private static ProjectManager instance;

    public static ProjectManager instance() {
        if (instance == null) instance = new ProjectManager();

        return instance;
    }
}
```

CFontRenderer.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/CFontRenderer.java

```
package com.github.jx4e.minecode.rendering;

import com.github.jx4e.minecode.project.IOUtil;
```

```java
import com.google.gson.GsonBuilder;
import com.google.gson.JsonArray;
import com.google.gson.JsonObject;
import net.minecraft.client.font.Font;
import net.minecraft.client.font.FontStorage;
import net.minecraft.client.font.FontType;
import net.minecraft.client.font.TextRenderer;
import net.minecraft.util.Identifier;
import net.minecraft.util.JsonHelper;

import java.util.LinkedList;
import java.util.List;
import java.util.Objects;
import java.util.UUID;

import static com.github.jx4e.minecode.MinecodeClient.mc;

/**
 * @author Jake (github.com/jx4e)
 * @since 22/06/2022
 **/

public class CFontRenderer {
    private final TextRenderer textRenderer;
    private final Identifier identifier;

    public CFontRenderer(String config) {
        identifier = new Identifier(UUID.randomUUID().toString().toLowerCase());
        textRenderer = createContext(config, identifier);
    }

    public TextRenderer getTextRenderer() {
        return textRenderer;
    }

    /**
     * Creates a text renderer from a TTF file.
     * @param config
     * @param id
     * @return
     */
    public TextRenderer createContext(String config, Identifier id) {
        List<Font> fonts = new LinkedList<>();
        String json = IOUtil.readResourceStream("/assets/minecraft/minecode/font/" + config
+ ".json");

        if (json == null) return null;

        JsonArray jsonArray =
JsonHelper.getArray(Objects.requireNonNull(JsonHelper.deserialize(new
GsonBuilder().setPrettyPrinting().disableHtmlEscaping().create(), json, JsonObject.class)),
"providers");

        for(int i = jsonArray.size() - 1; i >= 0; --i) {
            JsonObject jsonObject = JsonHelper.asObject(jsonArray.get(i), "providers[" + i
+ "]");
            try {
                String stringType = JsonHelper.getString(jsonObject, "type");
                FontType fontType = FontType.byId(stringType);
                Font font =
fontType.createLoader(jsonObject).load(mc.getResourceManager());
                if (font != null)
                    fonts.add(font);
            } catch (RuntimeException e) {
                e.printStackTrace();
            }
        }

        FontStorage fontStorage = new FontStorage(mc.getTextureManager(), id);
        fontStorage.setFonts(fonts);
        return new TextRenderer(identifier -> fontStorage, true);
    }
}
```

RenderManager.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/RenderManager.java

```
package com.github.jx4e.minecode.rendering;
```

```
import net.minecraft.client.font.TextRenderer;
import net.minecraft.client.util.math.MatrixStack;

import static com.github.jx4e.minecode.MinecodeClient.mc;

/**
 * @author Jake (github.com/jx4e)
 * @since 29/06/2022
 **/

public class RenderManager {
    private final Renderer renderer = new Renderer();
    private final CFontRenderer codeFontRenderer = new CFontRenderer("code");
    private final CFontRenderer textFontRenderer = new CFontRenderer("text");
    private MatrixStack currentMatrix;

    private RenderManager() {

    }

    public Renderer getRenderer() {
        return renderer;
    }

    public TextRenderer getCodeFontRenderer() {
        return codeFontRenderer.getTextRenderer();
    }

    public TextRenderer getTextFontRenderer() {
        return textFontRenderer.getTextRenderer();
    }

    public TextRenderer getDefaultFontRenderer() {
        return mc.textRenderer;
    }

    public MatrixStack getCurrentMatrix() {
        return currentMatrix;
    }

    public void setCurrentMatrix(MatrixStack currentMatrix) {
        this.currentMatrix = currentMatrix;
    }
```

```java
    private static RenderManager instance;

    public static RenderManager instance() {
        if (instance == null) instance = new RenderManager();

        return instance;
    }
}
```

Renderer.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/Renderer.java

```java
package com.github.jx4e.minecode.rendering;

import com.github.jx4e.minecode.rendering.theme.BoxColorScheme;
import com.mojang.blaze3d.systems.RenderSystem;
import net.minecraft.client.gui.DrawableHelper;
import net.minecraft.client.render.*;
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.util.math.Matrix4f;

import java.awt.*;

/**
 * @author Jake (github.com/jx4e)
 * @since 11/06/2022
 **/

public class Renderer extends DrawableHelper {
    /**
     * Render a box at the selected position
     * @param matrices
     * @param x
     * @param y
     * @param width
     * @param height
     * @param scheme
     */
    public void box(MatrixStack matrices, int x, int y, int width, int height, BoxColorScheme scheme) {
        colorBox(matrices.peek().getPositionMatrix(), x, y, x + width, y + height, scheme);
    }

    public void box(MatrixStack matrices, float x, float y, float width, float height, BoxColorScheme scheme) {
        colorBox(matrices.peek().getPositionMatrix(), x, y, x + width, y + height, scheme);
    }

    public void box(MatrixStack matrices, int x, int y, int width, int height, Color color) {
        colorBox(matrices.peek().getPositionMatrix(), x, y, x + width, y + height, color);
    }

    public void box(MatrixStack matrices, float x, float y, float width, float height, Color color) {
        colorBox(matrices.peek().getPositionMatrix(), x, y, x + width, y + height, color);
    }

    public void image(MatrixStack matrices, int glID, float x, float y, float width, float height) {
        RenderSystem.setShaderTexture(0, glID);
        drawTexture(matrices, (int) x, (int) y, 0, 0f, 0f, (int) width, (int) height, (int) width, (int) height);
    }

    private void colorBox(Matrix4f matrix, int x1, int y1, int x2, int y2, BoxColorScheme colorScheme) {
        int i;
        if (x1 < x2) {
            i = x1;
            x1 = x2;
            x2 = i;
        }
        if (y1 < y2) {
            i = y1;
            y1 = y2;
            y2 = i;
        }

        BufferBuilder bufferBuilder = Tessellator.getInstance().getBuffer();
        RenderSystem.enableBlend();
        RenderSystem.disableTexture();
        RenderSystem.defaultBlendFunc();
        RenderSystem.setShader(GameRenderer::getPositionColorShader);
        bufferBuilder.begin(VertexFormat.DrawMode.QUADS, VertexFormats.POSITION_COLOR);
        bufferBuilder.vertex(matrix, x1, y2, 0.0f).color(colorScheme.getBottomLeft().getRGB()).next();
        bufferBuilder.vertex(matrix, x2, y2, 0.0f).color(colorScheme.getBottomRight().getRGB()).next();
        bufferBuilder.vertex(matrix, x2, y1, 0.0f).color(colorScheme.getTopRight().getRGB()).next();
        bufferBuilder.vertex(matrix, x1, y1, 0.0f).color(colorScheme.getTopLeft().getRGB()).next();
        BufferRenderer.drawWithShader(bufferBuilder.end());
        RenderSystem.enableTexture();
        RenderSystem.disableBlend();
    }

    private void colorBox(Matrix4f matrix, float x1, float y1, float x2, float y2, BoxColorScheme colorScheme) {
        float i;
        if (x1 < x2) {
            i = x1;
            x1 = x2;
            x2 = i;
        }
        if (y1 < y2) {
            i = y1;
            y1 = y2;
            y2 = i;
        }

        BufferBuilder bufferBuilder = Tessellator.getInstance().getBuffer();
        RenderSystem.enableBlend();
        RenderSystem.disableTexture();
        RenderSystem.defaultBlendFunc();
        RenderSystem.setShader(GameRenderer::getPositionColorShader);
        bufferBuilder.begin(VertexFormat.DrawMode.QUADS, VertexFormats.POSITION_COLOR);
        bufferBuilder.vertex(matrix, x1, y2, 0.0f).color(colorScheme.getBottomLeft().getRGB()).next();
        bufferBuilder.vertex(matrix, x2, y2, 0.0f).color(colorScheme.getBottomRight().getRGB()).next();
        bufferBuilder.vertex(matrix, x2, y1, 0.0f).color(colorScheme.getTopRight().getRGB()).next();
        bufferBuilder.vertex(matrix, x1, y1, 0.0f).color(colorScheme.getTopLeft().getRGB()).next();
        BufferRenderer.drawWithShader(bufferBuilder.end());
        RenderSystem.enableTexture();
        RenderSystem.disableBlend();
    }

    private void colorBox(Matrix4f matrix, int x1, int y1, int x2, int y2, Color color) {
        int i;
        if (x1 < x2) {
            i = x1;
            x1 = x2;
            x2 = i;
        }
        if (y1 < y2) {
            i = y1;
            y1 = y2;
            y2 = i;
        }

        BufferBuilder bufferBuilder = Tessellator.getInstance().getBuffer();
        RenderSystem.enableBlend();
        RenderSystem.disableTexture();
        RenderSystem.defaultBlendFunc();
        RenderSystem.setShader(GameRenderer::getPositionColorShader);
        bufferBuilder.begin(VertexFormat.DrawMode.QUADS, VertexFormats.POSITION_COLOR);
        bufferBuilder.vertex(matrix, x1, y2, 0.0f).color(color.getRGB()).next();
        bufferBuilder.vertex(matrix, x2, y2, 0.0f).color(color.getRGB()).next();
        bufferBuilder.vertex(matrix, x2, y1, 0.0f).color(color.getRGB()).next();
        bufferBuilder.vertex(matrix, x1, y1, 0.0f).color(color.getRGB()).next();
        BufferRenderer.drawWithShader(bufferBuilder.end());
        RenderSystem.enableTexture();
        RenderSystem.disableBlend();
    }

    private void colorBox(Matrix4f matrix, float x1, float y1, float x2, float y2, Color color) {
        float i;
        if (x1 < x2) {
            i = x1;
            x1 = x2;
            x2 = i;
        }
        if (y1 < y2) {
            i = y1;
            y1 = y2;
            y2 = i;
        }

        BufferBuilder bufferBuilder = Tessellator.getInstance().getBuffer();
        RenderSystem.enableBlend();
        RenderSystem.disableTexture();
        RenderSystem.defaultBlendFunc();
        RenderSystem.setShader(GameRenderer::getPositionColorShader);
        bufferBuilder.begin(VertexFormat.DrawMode.QUADS, VertexFormats.POSITION_COLOR);
        bufferBuilder.vertex(matrix, x1, y2, 0.0f).color(color.getRGB()).next();
        bufferBuilder.vertex(matrix, x2, y2, 0.0f).color(color.getRGB()).next();
        bufferBuilder.vertex(matrix, x2, y1, 0.0f).color(color.getRGB()).next();
        bufferBuilder.vertex(matrix, x1, y1, 0.0f).color(color.getRGB()).next();
        BufferRenderer.drawWithShader(bufferBuilder.end());
        RenderSystem.enableTexture();
        RenderSystem.disableBlend();
    }

    private void drawTexturedQuad(Matrix4f matrix, float x0, float x1, float y0, float y1, float z, float u0, float u1, float v0,
float v1) {
        RenderSystem.setShader(GameRenderer::getPositionTexShader);
        BufferBuilder bufferBuilder = Tessellator.getInstance().getBuffer();
        bufferBuilder.begin(VertexFormat.DrawMode.QUADS, VertexFormats.POSITION_TEXTURE);
        bufferBuilder.vertex(matrix, (float)x0, (float)y1, (float)z).texture(u0, v1).next();
        bufferBuilder.vertex(matrix, (float)x1, (float)y1, (float)z).texture(u1, v1).next();
        bufferBuilder.vertex(matrix, (float)x1, (float)y0, (float)z).texture(u1, v0).next();
        bufferBuilder.vertex(matrix, (float)x0, (float)y0, (float)z).texture(u0, v0).next();
        BufferRenderer.drawWithShader(bufferBuilder.end());
    }
}
```

ResourceManager.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/ResourceManager.java

```
package com.github.jx4e.minecode.rendering;

import com.github.jx4e.minecode.project.ConfigManager;
import net.minecraft.client.texture.NativeImage;
import net.minecraft.client.texture.NativeImageBackedTexture;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;

/**
 * @author Jake (github.com/jx4e)
 * @since 29/06/2022
 **/

public class ResourceManager {
    private ResourceManager() {

    }

    /**
     * Creates a texture in the GL context from a resource
     * @param resourceName
     * @return
     */
    public NativeImageBackedTexture getNativeImageTexture(String resourceName) {
        try {
            NativeImage image = NativeImage.read(new FileInputStream(new
File(ConfigManager.instance().getResources(), resourceName)));
            return new NativeImageBackedTexture(image);
        } catch (FileNotFoundException e) {
            throw new RuntimeException(e);
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }

    private static ResourceManager instance;

    public static ResourceManager instance() {
        if (instance == null) instance = new ResourceManager();

        return instance;
    }
}
```

Editor.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/screens/Editor.java

```
package com.github.jx4e.minecode.rendering.screens;

import com.github.jx4e.minecode.Minecode;
import com.github.jx4e.minecode.project.LuaProject;
import com.github.jx4e.minecode.rendering.RenderManager;
import com.github.jx4e.minecode.rendering.theme.Theme;
import com.github.jx4e.minecode.rendering.widgets.buttons.IconButton;
import com.github.jx4e.minecode.rendering.widgets.buttons.EditorFileButton;
import com.github.jx4e.minecode.rendering.widgets.text.TextArea;
import net.minecraft.client.gui.screen.Screen;
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.text.Text;

import java.io.File;

import static com.github.jx4e.minecode.MinecodeClient.mc;

/**
 * @author Jake (github.com/jx4e)
 * @since 11/06/2022
 **/

public class Editor extends Screen {
    private final LuaProject project;
    private TextArea area;

    public Editor(LuaProject project) {
        super(Text.of(Minecode.MOD_NAME));
        this.project = project;
    }

    /**
     * Initialise all of the buttons and widgets
     */
    @Override
    protected void init() {
        super.init();

        // Reload the script
        project.reloadScript();

        int barHeight = 2 * (RenderManager.instance().getTextFontRenderer().fontHeight +
10);
        int buttonSize = 2 * (int) (barHeight / 3f);

        // Add our buttons
        addDrawableChild(new IconButton(5,  barHeight / 2 - buttonSize / 2,
                buttonSize, buttonSize, Text.of("Back"),
                button -> mc.setScreen(EditorProjectMenu.getInstance()),
                "back.png"
        ));

        // Add the text area
        addDrawableChild(area = new TextArea(width / 5, barHeight, 4 * width / 5, height -
barHeight,
                project.getMainScriptFile(), this));

        // Recursive add child files method
        addChildFiles("", project.getDir(), barHeight);

        // Add the run and Setting buttons
        addDrawableChild(new IconButton(5,  height - barHeight + buttonSize / 3,
                buttonSize, buttonSize, Text.of("Run"),
                button -> project.toggle(),
                "run.png"
        ));

        addDrawableChild(new IconButton(width / 5 - 5 - buttonSize,  height - barHeight +
buttonSize / 3,
                buttonSize, buttonSize, Text.of("Settings"),
                button -> {},
                "settings.png"
        ));
    }

    /**
     * Adds all of the files in a directory to the screen as EditorFileButton widgets.
     * Calls itself recursively
     * @param prefix
     * @param dir
     * @param drawY
     */
    private void addChildFiles(String prefix, File dir, int drawY) {
        for (File file : dir.listFiles()) {
            if (file.isDirectory()){
                addChildFiles(prefix + file.getName() + "/", file, drawY);
            } else if (file.isFile()) {
                addDrawableChild(new EditorFileButton(0,  drawY,
                        width/5, 18, Text.of(prefix + file.getName()),
                        button -> area.setEditingFile(file),
                        file
                ));
                drawY += 18;
            }
        }
    }
    /**
     * Called when the screen is closed
     */
    @Override
    public void removed() {
        area.getDocument().save();
    }

    /**
     * Draws the background
     * @param matrices
     */
    @Override
    public void renderBackground(MatrixStack matrices) {
        float barHeight = 2 * (RenderManager.instance().getTextFontRenderer().fontHeight +
10);

        // Render background
        RenderManager.instance().getRenderer().box(matrices, 0, 0, width, height,
Theme.DEFAULT.getBackground2());

        // Render bars
        RenderManager.instance().getRenderer().box(matrices, 0, 0, width, barHeight,
Theme.DEFAULT.getBackground1());
        RenderManager.instance().getRenderer().box(matrices, 0, 0, width / 5, height,
Theme.DEFAULT.getBackground1());
        RenderManager.instance().getRenderer().box(matrices, 0, barHeight - 2, width, 2,
Theme.DEFAULT.getAccent());

        // Render the run buttons bar
        RenderManager.instance().getRenderer().box(matrices, 0, height - barHeight, width /
5, height, Theme.DEFAULT.getBackground2());
    }

    /**
     * Called when it is time to draw the screen
     * @param matrices
     * @param mouseX
     * @param mouseY
     * @param delta
     */
    @Override
    public void render(MatrixStack matrices, int mouseX, int mouseY, float delta) {
        renderBackground(matrices);

        // Render instructions at the top
        matrices.push();
        RenderManager.instance().getTextFontRenderer().draw(matrices, project.getName(),
                (width / 2f) -
RenderManager.instance().getTextFontRenderer().getWidth(project.getName()) / 2f,
                (RenderManager.instance().getTextFontRenderer().fontHeight + 5),
                Theme.DEFAULT.getFont().getRGB()
        );
        matrices.pop();

        super.render(matrices, mouseX, mouseY, delta);
    }
}
```

EditorCreateProjectMenu.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/screens/EditorCreateProjectMenu.java

```
package com.github.jx4e.minecode.rendering.screens;

import com.github.jx4e.minecode.Minecode;
import com.github.jx4e.minecode.rendering.widgets.buttons.IconButton;
import com.github.jx4e.minecode.rendering.widgets.buttons.SimpleButton;
import com.github.jx4e.minecode.rendering.widgets.text.TextEntry;
import com.github.jx4e.minecode.rendering.theme.Theme;
import com.github.jx4e.minecode.project.ProjectManager;
import com.github.jx4e.minecode.rendering.RenderManager;
import com.github.jx4e.minecode.rendering.widgets.buttons.TextTickButton;
import net.minecraft.client.gui.screen.Screen;
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.text.Text;
import net.minecraft.util.Formatting;

import static com.github.jx4e.minecode.MinecodeClient.mc;
```

```java
/**
 * @author Jake (github.com/jx4e)
 * @since 11/06/2022
 **/

public class EditorCreateProjectMenu extends Screen {
    private TextEntry projectNameButton;
    private TextEntry mainScriptButton;
    private TextTickButton useTemplateButton;

    public EditorCreateProjectMenu() {
        super(Text.of(Minecode.MOD_NAME));
    }

    /**
     * Init buttons and stuff
     */
    @Override
    protected void init() {
        super.init();

        int barHeight = 2 * (RenderManager.instance().getTextFontRenderer().fontHeight +
10);
        int buttonSize = 2 * (int) (barHeight / 3f);

        addDrawableChild(new IconButton(5,  barHeight / 2 - buttonSize / 2,
                buttonSize, buttonSize, Text.of("Back"),
                button -> mc.setScreen(EditorProjectMenu.getInstance()),
                "back.png"
        ));

        int entryHeight = RenderManager.instance().getTextFontRenderer().fontHeight * 2;

        addDrawableChild(projectNameButton = new TextEntry(10, 50, width - 20, entryHeight,
                Text.of("Name"), (action) -> {})
        );

        addDrawableChild(mainScriptButton = new TextEntry(10, 60 + entryHeight, width - 20,
entryHeight,
                Text.of("Script"), (action) -> {})
        );

        addDrawableChild(useTemplateButton = new TextTickButton(10, 70 + entryHeight * 2,
                (width - 20) / 5, entryHeight, Text.of("Use Template Lua Script"), (p) ->
{}));

        addDrawableChild(new SimpleButton(0, height - barHeight,
                width, barHeight, Text.of("Create Project"), (p) -> {
            ProjectManager.instance().createProject(projectNameButton.getContent(),
mainScriptButton.getContent(), useTemplateButton.isEnabled());
            mc.setScreen(EditorProjectMenu.getInstance());
        }));
    }

    /**
     * Renders the background
     * @param matrices
     */
    @Override
    public void renderBackground(MatrixStack matrices) {
        float barHeight = 2 * (RenderManager.instance().getTextFontRenderer().fontHeight +
10);

        // Render background
        RenderManager.instance().getRenderer().box(matrices, 0, 0, width, height,
Theme.DEFAULT.getBackground2());

        // Render bars
        RenderManager.instance().getRenderer().box(matrices, 0, 0, width, barHeight,
Theme.DEFAULT.getBackground1());
        RenderManager.instance().getRenderer().box(matrices, 0, height - barHeight, width,
barHeight, Theme.DEFAULT.getBackground1());
        RenderManager.instance().getRenderer().box(matrices, 0, barHeight - 2, width, 2,
Theme.DEFAULT.getAccent());
    }

    /**
     * Called to draw the screen
     * @param matrices
     * @param mouseX
     * @param mouseY
     * @param delta
     */
    @Override
    public void render(MatrixStack matrices, int mouseX, int mouseY, float delta) {
        renderBackground(matrices);
        // Render instructions
        matrices.push();
        RenderManager.instance().getTextFontRenderer().draw(matrices, "Create a new " +
Formatting.BOLD + "Project",
                (width / 2f) -
RenderManager.instance().getTextFontRenderer().getWidth("Create a new " + Formatting.BOLD +
"Project") / 2f,
                (RenderManager.instance().getTextFontRenderer().fontHeight + 5),
                Theme.DEFAULT.getFont().getRGB()
        );
        matrices.pop();

        super.render(matrices, mouseX, mouseY, delta);
    }

    private static EditorCreateProjectMenu instance;

    /**
     * Get the instance of this screen
     * @return
     */
    public static EditorCreateProjectMenu getInstance() {
        if (instance == null) instance = new EditorCreateProjectMenu();

        return instance;
    }
}
```

EditorLearnMenu.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/screens/EditorLearnMenu.java

```
package com.github.jx4e.minecode.rendering.screens;

import com.github.jx4e.minecode.Minecode;
import com.github.jx4e.minecode.project.LessonManager;
import com.github.jx4e.minecode.rendering.widgets.buttons.IconButton;
import com.github.jx4e.minecode.rendering.theme.Theme;
import com.github.jx4e.minecode.rendering.RenderManager;
import com.github.jx4e.minecode.rendering.widgets.buttons.IconTextButton;
import net.minecraft.client.gui.screen.Screen;
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.text.Text;
import net.minecraft.util.Formatting;

import java.util.concurrent.atomic.AtomicInteger;

import static com.github.jx4e.minecode.MinecodeClient.mc;

/**
 * @author Jake (github.com/jx4e)
 * @since 11/06/2022
 **/

public class EditorLearnMenu extends Screen {
    public EditorLearnMenu() {
        super(Text.of(Minecode.MOD_NAME));
    }

    /**
     * Add widgets and buttons
     */
    @Override
    protected void init() {
        super.init();

        int barHeight = 2 * (RenderManager.instance().getTextFontRenderer().fontHeight +
10);
        int buttonSize = 2 * (int) (barHeight / 3f);

        addDrawableChild(new IconButton(5,  barHeight / 2 - buttonSize / 2,
                buttonSize, buttonSize, Text.of("Back"),
                button -> mc.setScreen(EditorMainMenu.getInstance()),
                "back.png"
        ));

        int buttonWidth = width - 20;
        int buttonHeight = RenderManager.instance().getTextFontRenderer().fontHeight * 2;
        int buttonX = 10;
        AtomicInteger buttonY = new AtomicInteger(50);
        LessonManager.instance().getLessons().forEach(lesson -> {
            addDrawableChild(new IconTextButton(buttonX, buttonY.get(), buttonWidth,
buttonHeight,
                    Text.of(lesson.getName()),
                    button -> mc.setScreen(new Lesson(lesson)),
                    "learn.png"
            ));

            buttonY.addAndGet(buttonHeight + 2);
        });
    }

    @Override
    public void renderBackground(MatrixStack matrices) {
        float barHeight = 2 * (RenderManager.instance().getTextFontRenderer().fontHeight +
10);

        // Render background
        RenderManager.instance().getRenderer().box(matrices, 0, 0, width, height,
Theme.DEFAULT.getBackground2());

        // Render bars
        RenderManager.instance().getRenderer().box(matrices, 0, 0, width, barHeight,
Theme.DEFAULT.getBackground1());
        RenderManager.instance().getRenderer().box(matrices, 0, height - barHeight, width,
barHeight, Theme.DEFAULT.getBackground1());
        RenderManager.instance().getRenderer().box(matrices, 0, barHeight - 2, width, 2,
Theme.DEFAULT.getAccent());

    }

    @Override
    public void render(MatrixStack matrices, int mouseX, int mouseY, float delta) {
        renderBackground(matrices);

        // Render instructions
        matrices.push();
        RenderManager.instance().getTextFontRenderer().draw(matrices, "Please select a " +
Formatting.BOLD + "Lesson.",
                (width / 2f) -
RenderManager.instance().getTextFontRenderer().getWidth("Please select a " +
Formatting.BOLD + "Lesson.") / 2f,
                (RenderManager.instance().getTextFontRenderer().fontHeight + 5),
                Theme.DEFAULT.getFont().getRGB()
        );
        matrices.pop();

        super.render(matrices, mouseX, mouseY, delta);
    }

    private static EditorLearnMenu instance;

    public static EditorLearnMenu getInstance() {
        if (instance == null) instance = new EditorLearnMenu();

        return instance;
    }
}
```

EditorMainMenu.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/screens/EditorMainMenu.java

```
package com.github.jx4e.minecode.rendering.screens;

import com.github.jx4e.minecode.Minecode;
import com.github.jx4e.minecode.rendering.widgets.buttons.IconTextButton;
import com.github.jx4e.minecode.rendering.theme.Theme;
import com.github.jx4e.minecode.rendering.RenderManager;
import com.github.jx4e.minecode.rendering.ResourceManager;
import net.minecraft.client.gui.screen.Screen;
import net.minecraft.client.texture.NativeImageBackedTexture;
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.text.Text;
import net.minecraft.util.Formatting;

import static com.github.jx4e.minecode.MinecodeClient.mc;

/**
 * @author Jake (github.com/jx4e)
 * @since 11/06/2022
 **/

public class EditorMainMenu extends Screen {
    public EditorMainMenu() {
        super(Text.of(Minecode.MOD_NAME));
    }

    @Override
    protected void init() {
        super.init();

        int buttonWidth = width / 2 - 20;
        int buttonHeight = RenderManager.instance().getTextFontRenderer().fontHeight * 2;
        int buttonX = width / 2 + 10;

        addDrawableChild(new IconTextButton(buttonX, 50, buttonWidth, buttonHeight,
                Text.of("Open a " + Formatting.BOLD + "Project"),
                button -> mc.setScreen(EditorProjectMenu.getInstance()),
                "folder.png"
        ));

        addDrawableChild(new IconTextButton(buttonX, 50 + buttonHeight + 2, buttonWidth,
buttonHeight,
                Text.of("Choose a " + Formatting.BOLD + "Lesson"),
                button -> mc.setScreen(EditorLearnMenu.getInstance()),
                "learn.png"
        ));

        addDrawableChild(new IconTextButton(buttonX, 50 + 2 * buttonHeight + 4,
buttonWidth, buttonHeight,
                Text.of("Configure " + Formatting.BOLD + "Settings"),
                button -> mc.setScreen(EditorLearnMenu.getInstance()),
                "settings.png"
        ));
    }

    @Override
    public void renderBackground(MatrixStack matrices) {
        // Render background
        RenderManager.instance().getRenderer().box(matrices, 0, 0, width / 2f, height,
Theme.DEFAULT.getBackground1());
        RenderManager.instance().getRenderer().box(matrices, width / 2f, 0, width / 2f,
height, Theme.DEFAULT.getBackground2());
    }

    @Override
    public void render(MatrixStack matrices, int mouseX, int mouseY, float delta) {
        renderBackground(matrices);

        // Render Logo
        NativeImageBackedTexture logoTexture =
ResourceManager.instance().getNativeImageTexture("logo.png");
        RenderManager.instance().getRenderer().image(matrices, logoTexture.getGlId(), 0,
RenderManager.instance().getTextFontRenderer().fontHeight + 5, width / 2f, width / 8f);
        logoTexture.close();

        // Render instructions
        matrices.push();
        RenderManager.instance().getTextFontRenderer().draw(matrices, "Please select an " +
Formatting.BOLD + "Option" + Formatting.RESET + " or press " + Formatting.BOLD + "ESC" +
Formatting.RESET + " to exit.",
                (3 * width / 4f) -
RenderManager.instance().getTextFontRenderer().getWidth("Please select an " +
Formatting.BOLD + "Option" + Formatting.RESET + " or press " + Formatting.BOLD + "ESC" +
Formatting.RESET + " to exit.") / 2f,
                (RenderManager.instance().getTextFontRenderer().fontHeight + 5),
                Theme.DEFAULT.getFont().getRGB()
        );
        matrices.pop();

        super.render(matrices, mouseX, mouseY, delta);
    }

    private static EditorMainMenu instance;

    public static EditorMainMenu getInstance() {
        if (instance == null) instance = new EditorMainMenu();

        return instance;
    }
}
```

EditorProjectMenu.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/screens/EditorProjectMenu.java

```
package com.github.jx4e.minecode.rendering.screens;

import com.github.jx4e.minecode.Minecode;
import com.github.jx4e.minecode.rendering.widgets.buttons.IconButton;
import com.github.jx4e.minecode.rendering.widgets.buttons.IconTextButton;
import com.github.jx4e.minecode.rendering.theme.Theme;
import com.github.jx4e.minecode.project.ProjectManager;
import com.github.jx4e.minecode.rendering.RenderManager;
import net.minecraft.client.gui.screen.Screen;
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.text.Text;
import net.minecraft.util.Formatting;

import java.util.concurrent.atomic.AtomicInteger;

import static com.github.jx4e.minecode.MinecodeClient.mc;

/**
 * @author Jake (github.com/jx4e)
 * @since 11/06/2022
 **/

public class EditorProjectMenu extends Screen {
    public EditorProjectMenu() {
        super(Text.of(Minecode.MOD_NAME));
    }

    @Override
    protected void init() {
        super.init();

        int barHeight = 2 * (RenderManager.instance().getTextFontRenderer().fontHeight +
10);
        int buttonSize = 2 * (int) (barHeight / 3f);

        addDrawableChild(new IconButton(5,  barHeight / 2 - buttonSize / 2,
                buttonSize, buttonSize, Text.of("Back"),
                button -> mc.setScreen(EditorMainMenu.getInstance()),
                "back.png"
        ));

        addDrawableChild(new IconButton(width - buttonSize - 5, barHeight / 2 - buttonSize
/ 2,
                buttonSize, buttonSize, Text.of("Add"),
                button -> mc.setScreen(EditorCreateProjectMenu.getInstance()),
                "add.png"
        ));

        int buttonWidth = width - 20;
        int buttonHeight = RenderManager.instance().getTextFontRenderer().fontHeight * 2;
        int buttonX = 10;
        AtomicInteger buttonY = new AtomicInteger(50);

        ProjectManager.instance().getProjects().forEach(luaProject -> {
            addDrawableChild(new IconTextButton(buttonX, buttonY.get(), buttonWidth,
buttonHeight,
                    Text.of(luaProject.getName()),
                    button -> mc.setScreen(new Editor(luaProject)),
                    "folder.png"
            ));

            buttonY.addAndGet(buttonHeight + 2);
        });
    }

    @Override
    public void renderBackground(MatrixStack matrices) {
        float barHeight = 2 * (RenderManager.instance().getTextFontRenderer().fontHeight +
10);

        // Render background
        RenderManager.instance().getRenderer().box(matrices, 0, 0, width, height,
Theme.DEFAULT.getBackground2());

        // Render bars
        RenderManager.instance().getRenderer().box(matrices, 0, 0, width, barHeight,
Theme.DEFAULT.getBackground1());
        RenderManager.instance().getRenderer().box(matrices, 0, height - barHeight, width,
barHeight, Theme.DEFAULT.getBackground1());
        RenderManager.instance().getRenderer().box(matrices, 0, barHeight - 2, width, 2,
Theme.DEFAULT.getAccent());
```

```
    }

    @Override
    public void render(MatrixStack matrices, int mouseX, int mouseY, float delta) {
        renderBackground(matrices);

        // Render instructions
        matrices.push();
        RenderManager.instance().getTextFontRenderer().draw(matrices, "Please select a " +
Formatting.BOLD + "Project" + Formatting.RESET + " or press " + Formatting.BOLD + "+" +
Formatting.RESET + " to create one.",
                (width / 2f) -
RenderManager.instance().getTextFontRenderer().getWidth("Please select a " +
Formatting.BOLD + "Project" + Formatting.RESET + " or press " + Formatting.BOLD + "+" +
Formatting.RESET + " to create one.") / 2f,
                (RenderManager.instance().getTextFontRenderer().fontHeight + 5),
                Theme.DEFAULT.getFont().getRGB()
        );
        matrices.pop();

        super.render(matrices, mouseX, mouseY, delta);
    }

    private static EditorProjectMenu instance;

    public static EditorProjectMenu getInstance() {
        if (instance == null) instance = new EditorProjectMenu();

        return instance;
    }
}
```

Lesson.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/screens/Lesson.java

```
package com.github.jx4e.minecode.rendering.screens;

import com.github.jx4e.minecode.Minecode;
import com.github.jx4e.minecode.project.LuaLesson;
import com.github.jx4e.minecode.rendering.RenderManager;
import com.github.jx4e.minecode.rendering.theme.Theme;
import com.github.jx4e.minecode.rendering.widgets.buttons.IconButton;
import com.github.jx4e.minecode.rendering.widgets.text.TextArea;
import net.minecraft.client.gui.screen.Screen;
import net.minecraft.client.gui.widget.ButtonWidget;
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.text.Text;

import java.util.HashMap;
import java.util.Map;
import java.util.concurrent.atomic.AtomicInteger;
import java.util.function.Consumer;

import static com.github.jx4e.minecode.MinecodeClient.mc;

/**
 * @author Jake (github.com/jx4e)
 * @since 11/06/2022
 **/

public class Lesson extends Screen {
    private final LuaLesson lesson;
    private TextArea area;
    private IconButton run, submit;
    private float scroll = 0;

    public Lesson(LuaLesson lesson) {
        super(Text.of(Minecode.MOD_NAME));
        this.lesson = lesson;
    }

    @Override
    protected void init() {
        super.init();

        int barHeight = 2 * (RenderManager.instance().getTextFontRenderer().fontHeight +
10);
        int buttonSize = 2 * (int) (barHeight / 3f);

        addDrawableChild(new IconButton(5,  barHeight / 2 - buttonSize / 2,
                buttonSize, buttonSize, Text.of("Back"),
                button -> mc.setScreen(EditorLearnMenu.getInstance()),
                "back.png"
        ));
        // We do not want to add this to the children since it will be rendered when
super.render() is called
        // Instead we will draw it ourselves manually.
        addDrawableChild(
                area = new TextArea(width / 5, barHeight, 4 * width / 5, height -
barHeight,
                        lesson.getMainScriptFile(), this)
        );

        addDrawableChild(run = new IconButton(5, height - barHeight + buttonSize / 3,
                buttonSize, buttonSize, Text.of("Run"),
                button -> {
                    area.getDocument().save();
                    lesson.getScript().toggle();
                },
                "run.png"
        ));

        addDrawableChild(submit = new IconButton(5, height - barHeight + buttonSize / 3,
                buttonSize, buttonSize, Text.of("Check"),
                button -> {
                    Map<LuaLesson.LessonContent, Boolean> taskStatus = new HashMap<>();
                    String code = area.getDocument().getContent();

                    // Iterate through the tasks
                    for (LuaLesson.LessonContent task : lesson.getTasks()) {
                        // If the users code contains the correct code
                        boolean completed = code.contains(task.code());
                        // Add the task and if it was completed to our map
                        taskStatus.put(task, completed);
                    }

                    // display
                    mc.setScreen(new LessonSummary(this, taskStatus));
                },
                "tick.png"
        ));
    }

    @Override
    public void renderBackground(MatrixStack matrices) {
        // Render background
        RenderManager.instance().getRenderer().box(matrices, 0, 0, width, height,
Theme.DEFAULT.getBackground2());
    }

    @Override
    public void render(MatrixStack matrices, int mouseX, int mouseY, float delta) {
        // Draw the background
        renderBackground(matrices);

        float barHeight = 2 * (RenderManager.instance().getTextFontRenderer().fontHeight +
10);

        // Calculate the scroll amount
        scroll = Math.min(scroll, 0);

        float drawX = 10;
        float drawY = 10 + 2 * (RenderManager.instance().getTextFontRenderer().fontHeight +
10) + scroll;
        float drawWidth = width - drawX * 2;
        float drawHeight = 0;

        // Draw the lesson description
        matrices.push();
        drawHeight = RenderManager.instance().getTextFontRenderer().fontHeight + 10;
        RenderManager.instance().getRenderer().box(matrices, drawX, drawY, drawWidth,
drawHeight, Theme.DEFAULT.getBackground1());
        RenderManager.instance().getRenderer().box(matrices, 0, barHeight - 2, width, 2,
Theme.DEFAULT.getAccent());
        RenderManager.instance().getTextFontRenderer().draw(matrices,
lesson.getDescription(),
                (width / 2f) -
RenderManager.instance().getTextFontRenderer().getWidth(lesson.getDescription()) / 2f,
                drawY + 5,
                Theme.DEFAULT.getFont().getRGB()
        );
        matrices.pop();

        // Move our draw pointer so it draws lower down the page
        drawY += drawHeight + 10;

        // Draw the lesson content
        for (LuaLesson.LessonContent content : lesson.getContent()) {
            matrices.push();
            drawHeight = RenderManager.instance().getTextFontRenderer().fontHeight + 10;

            RenderManager.instance().getRenderer().box(matrices, drawX, drawY, drawWidth,
drawHeight, Theme.DEFAULT.getBackground1());

            RenderManager.instance().getTextFontRenderer().draw(matrices, content.text(),
                    (width / 2f) -
RenderManager.instance().getTextFontRenderer().getWidth(content.text()) / 2f,
                    drawY + 5,
                    Theme.DEFAULT.getFont().getRGB()
            );

            drawY += drawHeight;

            float codeBoxHeight = drawHeight;
            drawHeight = drawHeight + 4;

            RenderManager.instance().getRenderer().box(matrices, drawX, drawY, drawWidth,
drawHeight, Theme.DEFAULT.getBackground1());
            RenderManager.instance().getRenderer().box(matrices, drawX + 2, drawY + 2,
drawWidth - 4, codeBoxHeight, Theme.DEFAULT.getButton());

            RenderManager.instance().getCodeFontRenderer().draw(matrices, content.code(),
                    (width / 2f) -
RenderManager.instance().getCodeFontRenderer().getWidth(content.code()) / 2f,
                    drawY + 5,
                    Theme.DEFAULT.getFont().getRGB()
            );

            matrices.pop();

            drawY += drawHeight + 10;
        }

        // Draw the tasks title
        RenderManager.instance().getTextFontRenderer().draw(matrices, "Task: Create a
program that:",
                (width / 2f) -
RenderManager.instance().getTextFontRenderer().getWidth("Task: Create a program that:") /
2f,
                drawY + 5,
                Theme.DEFAULT.getFont().getRGB()
        );
        drawY += RenderManager.instance().getTextFontRenderer().fontHeight + 10;

        // Draw the lesson criteria
        for (LuaLesson.LessonContent content : lesson.getTasks()) {
            matrices.push();
            drawHeight = RenderManager.instance().getTextFontRenderer().fontHeight + 10;

            RenderManager.instance().getRenderer().box(matrices, drawX, drawY, drawWidth, 
drawHeight, Theme.DEFAULT.getBackground1());

            RenderManager.instance().getTextFontRenderer().draw(matrices, content.text(),
                    (width / 2f) - 
RenderManager.instance().getTextFontRenderer().getWidth(content.text()) / 2f,
                    drawY + 5,
                    Theme.DEFAULT.getFont().getRGB()
            );

            drawY += drawHeight + 10;
            matrices.pop();
        }

        // Draw the code editing area
        RenderManager.instance().getRenderer().box(matrices, drawX, drawY, drawWidth, 
area.getHeight() + 22, Theme.DEFAULT.getBackground1());

        run.x = (int) drawX + 2;
        run.y = (int) drawY + 2;
        run.setWidth(16);

        submit.x = (int) drawX + 20;
        submit.y = (int) drawY + 2;
        submit.setWidth(16);

        drawY += 20;

        area.x = (int) drawX + 2;
        area.y = (int) drawY;
        area.setWidth((int) drawWidth - 4);

        // Finally render the stuff that should be overlayed
        // Render bars
        RenderManager.instance().getRenderer().box(matrices, 0, 0, width, barHeight,
Theme.DEFAULT.getBackground1());

        // Render instructions
        matrices.push();
        RenderManager.instance().getTextFontRenderer().draw(matrices, lesson.getName(),
                (width / 2f) -
RenderManager.instance().getTextFontRenderer().getWidth(lesson.getName()) / 2f,
                (RenderManager.instance().getTextFontRenderer().fontHeight + 5),
                Theme.DEFAULT.getFont().getRGB()
        );
        matrices.pop();

        super.render(matrices, mouseX, mouseY, delta);
    }

    @Override
    public boolean mouseScrolled(double mouseX, double mouseY, double amount) {
        if (mouseY > 2 * (RenderManager.instance().getTextFontRenderer().fontHeight + 10))
{
            scroll += amount;
        }

        return super.mouseScrolled(mouseX, mouseY, amount);
    }
}
```

QuickToggleMenu.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/screens/QuickToggleMenu.java

```
package com.github.jx4e.minecode.rendering.screens;

import com.github.jx4e.minecode.Minecode;
import com.github.jx4e.minecode.rendering.theme.Theme;
import com.github.jx4e.minecode.project.ProjectManager;
import com.github.jx4e.minecode.rendering.RenderManager;
import com.github.jx4e.minecode.rendering.widgets.buttons.ProjectToggleButton;
import net.minecraft.client.gui.screen.Screen;
```

```java
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.text.Text;

import java.util.concurrent.atomic.AtomicInteger;

/**
 * @author Jake (github.com/jx4e)
 * @since 11/06/2022
 **/

public class QuickToggleMenu extends Screen {
    public QuickToggleMenu() {
        super(Text.of(Minecode.MOD_NAME));
    }

    @Override
    protected void init() {
        super.init();

        int buttonWidth = width - 20;
        int buttonHeight = RenderManager.instance().getTextFontRenderer().fontHeight * 2;
        int buttonX = 10;
        AtomicInteger buttonY = new AtomicInteger(50);

        ProjectManager.instance().getProjects().forEach(luaProject -> {
            // Add the button at the buttn X and Y
            addDrawableChild(new ProjectToggleButton(buttonX, buttonY.get(), buttonWidth,
buttonHeight,
                    Text.of(luaProject.getName()),
                    button -> luaProject.setEnabled(!luaProject.isEnabled()),
                    luaProject
            ));

            // Increment Y so that we get the buttons in a column
            buttonY.addAndGet(buttonHeight + 2);
        });
    }

    @Override
    public void renderBackground(MatrixStack matrices) {
        float barHeight = 2 * (RenderManager.instance().getTextFontRenderer().fontHeight +
10);
```

```
        // Render background
        RenderManager.instance().getRenderer().box(matrices, 0, 0, width, height,
Theme.DEFAULT.getBackground2());

        // Render bars
        RenderManager.instance().getRenderer().box(matrices, 0, 0, width, barHeight,
Theme.DEFAULT.getBackground1());
        RenderManager.instance().getRenderer().box(matrices, 0, height - barHeight, width,
barHeight, Theme.DEFAULT.getBackground1());
        RenderManager.instance().getRenderer().box(matrices, 0, barHeight - 2, width, 2,
Theme.DEFAULT.getAccent());
    }

    @Override
    public void render(MatrixStack matrices, int mouseX, int mouseY, float delta) {
        renderBackground(matrices);

        // Render instructions
        matrices.push();
        RenderManager.instance().getTextFontRenderer().draw(matrices, "Click to toggle a
project",
                (width / 2f) -
RenderManager.instance().getTextFontRenderer().getWidth("Click to toggle a project") / 2f,
                (RenderManager.instance().getTextFontRenderer().fontHeight + 5),
                Theme.DEFAULT.getFont().getRGB()
        );
        matrices.pop();

        super.render(matrices, mouseX, mouseY, delta);
    }

    private static QuickToggleMenu instance;

    public static QuickToggleMenu getInstance() {
        if (instance == null) instance = new QuickToggleMenu();

        return instance;
    }
}
```

BoxColorScheme.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/theme/BoxColorScheme.java

```
package com.github.jx4e.minecode.rendering.theme;

import java.awt.*;

/**
 * @author Jake (github.com/jx4e)
 * @since 22/06/2022
 **/

public class BoxColorScheme {
    private final Color topLeft;
    private final Color topRight;
    private final Color bottomLeft;
    private final Color bottomRight;

    public BoxColorScheme(Color topLeft, Color topRight, Color bottomLeft, Color
bottomRight) {
        this.topLeft = topLeft;
        this.topRight = topRight;
        this.bottomLeft = bottomLeft;
        this.bottomRight = bottomRight;
    }

    public Color getTopLeft() {
        return topLeft;
    }

    public Color getTopRight() {
        return topRight;
    }

    public Color getBottomLeft() {
        return bottomLeft;
    }

    public Color getBottomRight() {
        return bottomRight;
    }

    public static class Flat extends BoxColorScheme {
        public Flat(Color color) {
            super(color, color, color, color);
        }

        public Color getColor() {
            return super.getTopLeft();
        }
    }

    public static class Gradient extends BoxColorScheme {
        private final boolean horizontal;

        public Gradient(Color start, Color end, boolean horizontal) {
            super(start, horizontal ? end : start, horizontal ? start : end, end);
            this.horizontal = horizontal;
        }

        public boolean isHorizontal() {
            return horizontal;
        }

        public Color getStart() {
            return super.getTopLeft();
        }

        public Color getEnd() {
            return super.getTopLeft();
        }
    }
}
```

Theme.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/theme/Theme.java

```
package com.github.jx4e.minecode.rendering.theme;

import java.awt.*;

public class Theme {
    private Color accent = new Color(0x138BEC);
    private Color background1 = new Color(30, 30, 30,255);
    private Color background2 = new Color(43, 43, 43,255);
    private Color background3 = new Color(50, 50, 50,255);
    private Color button = new Color(60, 63, 65,255);
    private Color font = new Color(0xFFFFFF);

    public Color getAccent() {
        return accent;
    }

    public Theme setAccent(Color accent) {
        this.accent = accent;
        return this;
    }

    public Color getBackground1() {
        return background1;
    }

    public Theme setBackground1(Color background1) {
        this.background1 = background1;
        return this;
    }

    public Color getBackground2() {
        return background2;
    }

    public Theme setBackground2(Color background2) {
        this.background2 = background2;
        return this;
    }

    public Color getBackground3() {
        return background3;
    }

    public Theme setBackground3(Color background3) {
        this.background3 = background3;
        return this;
    }

    public Color getButton() {
        return button;
    }

    public Theme setButton(Color button) {
        this.button = button;
        return this;
    }

    public Color getFont() {
        return font;
    }

    public Theme setFont(Color font) {
        this.font = font;
        return this;
    }

    public static final Theme DEFAULT = new Theme();
}
```

EditorFileButton.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/widgets/buttons/EditorFileButton.java

```
package com.github.jx4e.minecode.rendering.widgets.buttons;

import com.github.jx4e.minecode.rendering.RenderManager;
import com.github.jx4e.minecode.rendering.ResourceManager;
import com.github.jx4e.minecode.rendering.theme.Theme;
import net.minecraft.client.gui.widget.ButtonWidget;
import net.minecraft.client.texture.NativeImageBackedTexture;
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.text.Text;

import java.awt.*;
import java.io.File;

public class EditorFileButton extends ButtonWidget {
    private final File file;
    public EditorFileButton(int x, int y, int width, int height, Text message, PressAction
onPress,
                            File file) {
        super(x, y, width, height, message, onPress);
        this.file = file;
    }

    @Override
    public void renderButton(MatrixStack matrices, int mouseX, int mouseY, float delta) {
        String icon = "document.png";

        // Decide which icon we want for this specific file
        if (file.getName().endsWith(".json")) icon = "json.png";
        else if (file.getName().endsWith(".lua")) icon = "code.png";

        // Render the background and text
        RenderManager.instance().getRenderer().box(matrices, x, y, width, getHeight(),
Theme.DEFAULT.getBackground2());
        RenderManager.instance().getTextFontRenderer().draw(matrices, getMessage(),
                x + 2,
                y + getHeight() / 2f -
RenderManager.instance().getTextFontRenderer().fontHeight / 2f,
                Color.WHITE.getRGB()
        );

        // Render the icon
        NativeImageBackedTexture texture =
ResourceManager.instance().getNativeImageTexture(icon);
        RenderManager.instance().getRenderer().image(matrices, texture.getGlId(),
                x + getWidth() - getHeight(),
                y + 2,
                getHeight() - 4, getHeight() - 4
        );
        texture.close();
    }
}
```

IconButton.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/widgets/buttons/IconButton.java

```
package com.github.jx4e.minecode.rendering.widgets.buttons;

import com.github.jx4e.minecode.rendering.RenderManager;
import com.github.jx4e.minecode.rendering.ResourceManager;
import com.github.jx4e.minecode.rendering.theme.Theme;
import net.minecraft.client.MinecraftClient;
import net.minecraft.client.gui.widget.ButtonWidget;
import net.minecraft.client.texture.NativeImageBackedTexture;
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.text.Text;

public class IconButton extends ButtonWidget {
    private final String iconName;

    public IconButton(int x, int y, int width, int height, Text message, PressAction
onPress, String iconName) {
        super(x, y, width, height, message, onPress);
        this.iconName = iconName;
    }

    public IconButton(int x, int y, int width, int height, Text message, PressAction
onPress,
                      TooltipSupplier tooltipSupplier, String iconName) {
        super(x, y, width, height, message, onPress, tooltipSupplier);
        this.iconName = iconName;
    }

    @Override
    protected void renderBackground(MatrixStack matrices, MinecraftClient client, int
mouseX, int mouseY) {
        RenderManager.instance().getRenderer().box(matrices, x, y, getWidth(), getHeight(),
Theme.DEFAULT.getButton());
    }

    @Override
    public void renderButton(MatrixStack matrices, int mouseX, int mouseY, float delta) {
        // Render the icon with the texture
        NativeImageBackedTexture texture =
ResourceManager.instance().getNativeImageTexture(iconName);
        RenderManager.instance().getRenderer().image(matrices, texture.getGlId(),
                x,
                y,
                getWidth(), getWidth()
        );
        texture.close();
    }
}
```

IconTextButton.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/widgets/buttons/IconTextButton.java

```
package com.github.jx4e.minecode.rendering.widgets.buttons;

import com.github.jx4e.minecode.rendering.theme.Theme;
import com.github.jx4e.minecode.rendering.RenderManager;
import com.github.jx4e.minecode.rendering.ResourceManager;
import net.minecraft.client.gui.widget.ButtonWidget;
import net.minecraft.client.texture.NativeImageBackedTexture;
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.text.Text;

import java.awt.*;

public class IconTextButton extends ButtonWidget {
    private final String iconName;

    public IconTextButton(int x, int y, int width, int height, Text message, PressAction
onPress,
                          String iconName) {
        super(x, y, width, height, message, onPress);
        this.iconName = iconName;
    }

    @Override
    public void renderButton(MatrixStack matrices, int mouseX, int mouseY, float delta) {
        RenderManager.instance().getRenderer().box(matrices, x, y, width, getHeight(),
Theme.DEFAULT.getButton());
        RenderManager.instance().getTextFontRenderer().draw(matrices, getMessage(),
                x + 2,
                y + getHeight() / 2f -
RenderManager.instance().getTextFontRenderer().fontHeight / 2f,
                Color.WHITE.getRGB()
        );

        NativeImageBackedTexture texture =
ResourceManager.instance().getNativeImageTexture(iconName);
        RenderManager.instance().getRenderer().image(matrices, texture.getGlId(),
                x + getWidth() - getHeight(),
                y + 2,
                getHeight() - 4, getHeight() - 4
        );
        texture.close();
    }
}
```

ProjectToggleButton.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/widgets/buttons/ProjectToggleButton.java

```
package com.github.jx4e.minecode.rendering.widgets.buttons;

import com.github.jx4e.minecode.rendering.RenderManager;
import com.github.jx4e.minecode.project.LuaProject;
import com.github.jx4e.minecode.rendering.theme.Theme;
import net.minecraft.client.gui.widget.ButtonWidget;
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.text.Text;

import java.awt.*;

public class ProjectToggleButton extends ButtonWidget {
    private LuaProject project;
    public ProjectToggleButton(int x, int y, int width, int height, Text message,
PressAction onPress, LuaProject project) {
        super(x, y, width, height, message, onPress);
        this.project = project;
    }

    @Override
    public void renderButton(MatrixStack matrices, int mouseX, int mouseY, float delta) {
        RenderManager.instance().getRenderer().box(matrices, x, y, getWidth(), getHeight(),
Theme.DEFAULT.getBackground1());
        RenderManager.instance().getTextFontRenderer().draw(matrices,
getMessage().getString(),
                x + getWidth() / 2f -
RenderManager.instance().getTextFontRenderer().getWidth(getMessage().getString()) / 2f,
                y + getHeight() / 2f -
RenderManager.instance().getTextFontRenderer().fontHeight / 2f,
                project.isEnabled() ? Theme.DEFAULT.getAccent().getRGB() :
Color.WHITE.getRGB()
        );
    }
}
```

SimpleButton.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/widgets/buttons/SimpleButton.java

```
package com.github.jx4e.minecode.rendering.widgets.buttons;

import com.github.jx4e.minecode.rendering.theme.Theme;
import com.github.jx4e.minecode.rendering.RenderManager;
import net.minecraft.client.gui.widget.ButtonWidget;
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.text.Text;

import java.awt.*;

public class SimpleButton extends ButtonWidget {
    public SimpleButton(int x, int y, int width, int height, Text message, PressAction
onPress) {
        super(x, y, width, height, message, onPress);
    }

    @Override
    public void renderButton(MatrixStack matrices, int mouseX, int mouseY, float delta) {
        RenderManager.instance().getRenderer().box(matrices, x, y, getWidth(), getHeight(),
Theme.DEFAULT.getAccent());
        RenderManager.instance().getTextFontRenderer().draw(matrices,
getMessage().getString(),
                x + getWidth() / 2f -
RenderManager.instance().getTextFontRenderer().getWidth(getMessage().getString()) / 2f,
                y + getHeight() / 2f -
RenderManager.instance().getTextFontRenderer().fontHeight / 2f,
                Color.WHITE.getRGB()
        );
    }
}
```

TextTickButton.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/widgets/buttons/TextTickButton.java

```
package com.github.jx4e.minecode.rendering.widgets.buttons;

import com.github.jx4e.minecode.rendering.RenderManager;
import com.github.jx4e.minecode.rendering.ResourceManager;
import com.github.jx4e.minecode.rendering.theme.Theme;
import net.minecraft.client.gui.widget.ButtonWidget;
import net.minecraft.client.texture.NativeImageBackedTexture;
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.text.Text;

import java.awt.*;

public class TextTickButton extends ButtonWidget {
    boolean enabled = false;
    public TextTickButton(int x, int y, int width, int height, Text message, PressAction
onPress) {
        super(x, y, width, height, message, onPress);
    }

    @Override
    public void renderButton(MatrixStack matrices, int mouseX, int mouseY, float delta) {
        // Render background and text
        RenderManager.instance().getRenderer().box(matrices, x, y,
                getWidth() - getHeight(),
                getHeight(), Theme.DEFAULT.getBackground3());
        RenderManager.instance().getRenderer().box(matrices, x + getWidth() - getHeight(),
y,
                getHeight(),
                getHeight(), Theme.DEFAULT.getButton());


        RenderManager.instance().getTextFontRenderer().draw(matrices, getMessage(),
                x + 2,
                y + getHeight() / 2f -
RenderManager.instance().getTextFontRenderer().fontHeight / 2f,
                Color.WHITE.getRGB()
        );

        // Only render the tick if it's enabled
        if (!enabled) return;

        NativeImageBackedTexture texture =
ResourceManager.instance().getNativeImageTexture("tick.png");
        RenderManager.instance().getRenderer().image(matrices, texture.getGlId(),
                x + getWidth() - getHeight() + 2,
                y + 2,
                getHeight() - 4, getHeight() - 4
        );
        texture.close();
    }

    @Override
    public void onClick(double mouseX, double mouseY) {
        enabled = !enabled;

        super.onClick(mouseX, mouseY);
```

```java
    }

    public boolean isEnabled() {
        return enabled;
    }
}
```

OneLineDocument.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/widgets/text/OneLineDocument.java

```java
package com.github.jx4e.minecode.rendering.widgets.text;

/**
 * @author Jake (github.com/jx4e)
 * @since 26/09/2022
 **/

public class OneLineDocument {
    private StringBuilder content;
    private int pointer = 0;

    public OneLineDocument(String content) {
        this.content = new StringBuilder(content);
    }

    /**
     * Inserts a string into the doc
     * @param string
     */
    public void insert(String string) {
        content.insert(pointer, string);
        setPointer(pointer + string.length());
    }

    /**
     * Deletes the char before the pointer
     */
    public void backspace() {
        if (pointer == 0) return;
        content.deleteCharAt(pointer - 1);
        setPointer(pointer - 1);
    }

    public void pointerLeft() {
        setPointer(pointer - 1);
    }

    public void pointerRight() {
        setPointer(pointer + 1);
    }

    public void setPointer(int x) {
        pointer = x;
    }

    public int getPointer() {
        return pointer;
    }

    public String getContent() {
        return content.toString();
    }
}
```

TextDocument.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/widgets/text/TextDocument.java

```
package com.github.jx4e.minecode.rendering.widgets.text;

import com.github.jx4e.minecode.project.IOUtil;
import com.mojang.datafixers.util.Pair;
import net.minecraft.util.math.MathHelper;

import java.io.ByteArrayInputStream;
import java.io.File;
```

```java
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.nio.charset.StandardCharsets;
import java.util.Arrays;
import java.util.LinkedList;

/**
 * @author Jake (github.com/jx4e)
 * @since 26/09/2022
 **/

public class TextDocument {
    private final File editing;
    private final LinkedList<String> lines = new LinkedList<>();
    private Pair<Integer, Integer> pointer = new Pair<>(0, 0);

    public TextDocument(File editing) {
        this.editing = editing;
        lines.addAll(Arrays.asList(IOUtil.readFileToString(editing).split("\n")));
    }

    /**
     * Saves the document by writing the updated content to the file
     */
    public void save() {
        if (editing == null) return;

        try {
            IOUtil.writeToOutputStream(
                    new
ByteArrayInputStream(getContent().getBytes(StandardCharsets.UTF_8)),
                    new FileOutputStream(editing)
            );
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        }
    }

    /**
     * Inserts a string into the doc
     * @param string
     */
    public void insert(String string) {
        StringBuilder lineContent = new StringBuilder(lines.get(pointer.getSecond()));
        lineContent.insert(pointer.getFirst(), string);
        lines.set(pointer.getSecond(), lineContent.toString());
        setPointer(pointer.getFirst() + string.length(), pointer.getSecond());
    }

    /**
     * Deletes the char before the pointer
     */
    public void backspace() {
        // This is for going back a line
        if (pointer.getFirst() == 0) {
            // return if we are on the very first line since there's nothing to do
            if (pointer.getSecond() == 0) return;
            // move pointer to last line
            setPointer(lines.get(pointer.getSecond() - 1).length() - 1, pointer.getSecond()
- 1);
            // Add the line to the previous line
            StringBuilder lineContent = new StringBuilder(lines.get(pointer.getSecond()));
            lineContent.append(lines.get(pointer.getSecond() + 1));
            // update
            lines.set(pointer.getSecond(), lineContent.toString());
            // Remove old lne
            lines.remove(pointer.getSecond() + 1);
            return;
        }

        // Delete the previous character
        StringBuilder lineContent = new StringBuilder(lines.get(pointer.getSecond()));
        lineContent.deleteCharAt(pointer.getFirst() - 1);
        lines.set(pointer.getSecond(), lineContent.toString());
        setPointer(pointer.getFirst() - 1, pointer.getSecond());
    }

    /**
     * Adds a new line
     */
    public void newLine() {
        String toMove = lines.get(pointer.getSecond()).substring(pointer.getFirst());
        lines.set(pointer.getSecond(), lines.get(pointer.getSecond()).substring(0,
pointer.getFirst()));
        lines.add(pointer.getSecond() + 1, toMove);
        setPointer(0, pointer.getSecond() + 1);
    }

    public void pointerUp() {
        setPointer(pointer.getFirst(), pointer.getSecond() - 1);
    }

    public void pointerDown() {
        setPointer(pointer.getFirst(), pointer.getSecond() + 1);
    }

    public void pointerLeft() {
        setPointer(pointer.getFirst() - 1, pointer.getSecond());
    }

    public void pointerRight() {
        setPointer(pointer.getFirst() + 1, pointer.getSecond());
    }

    public void setPointer(int x, int y) {
        // Validate the inputs so that they are within the right range
        y = MathHelper.clamp(y, 0, lines.size() - 1);
        x = MathHelper.clamp(x, 0, lines.get(y).length());

        pointer = new Pair<>(x, y);
    }

    public Pair<Integer, Integer> getPointer() {
        return pointer;
    }

    public String getContent() {
        StringBuilder content = new StringBuilder();

        for (int i = 0; i < lines.size(); i++) {
            content.append(lines.get(i));

            if (i != lines.size() - 1) {
                content.append("\n");
            }
        }

        return content.toString();
    }
    public LinkedList<String> getLines() {
        return lines;
    }

    public File getEditing() {
        return editing;
    }
}
```

TextEntry.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/widgets/text/TextEntry.java

Please note: This file has some formatting issues. Best viewed from GitHub.

```
package com.github.jx4e.minecode.rendering.widgets.text;

import com.github.jx4e.minecode.rendering.theme.Theme;
import com.github.jx4e.minecode.rendering.RenderManager;
import net.minecraft.client.gui.widget.ButtonWidget;
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.text.Text;

import java.awt.*;

import static org.lwjgl.glfw.GLFW.*;

public class TextEntry extends ButtonWidget {
    private boolean typing = false;
    private OneLineDocument document = new OneLineDocument("");
    private long last;

    public TextEntry(int x, int y, int width, int height, Text message, PressAction
onPress) {
        this(x, y, width, height, message, onPress, EMPTY);
    }
    public TextEntry(int x, int y, int width, int height, Text message, PressAction 
onPress,
                     TooltipSupplier tooltipSupplier) {
        super(x, y, width, height, message, onPress, tooltipSupplier);
        this.last = System.currentTimeMillis();
    }

    @Override
    public void onPress() {
        typing = !typing;
        super.onPress();
    }

    @Override
    public void renderButton(MatrixStack matrices, int mouseX, int mouseY, float delta) {
        RenderManager.instance().getRenderer().box(matrices, x, y, getWidth(), getHeight(),
Theme.DEFAULT.getButton());
        RenderManager.instance().getRenderer().box(matrices, x, y,
                getWidth() / 7,
                getHeight(), Theme.DEFAULT.getBackground3());

        RenderManager.instance().getTextFontRenderer().draw(matrices, getMessage(),
                x,
                y + getHeight() / 2f - 
RenderManager.instance().getTextFontRenderer().fontHeight / 2f,
                Theme.DEFAULT.getFont().getRGB()
        );

        RenderManager.instance().getCodeFontRenderer().draw(matrices, getContent(),
                x + getWidth() / 7,
                y + getHeight() / 2f - 
RenderManager.instance().getCodeFontRenderer().fontHeight / 2f,
                Color.WHITE.getRGB()
        );

        // Return if not typing
        if (!typing) return;


        // We want it to be visible for 500ms then gone for 250ms
        // While last<500ms it shows
        // Then it updates at 750ms so that between 500-750 it was not visible
        RenderManager.instance().getRenderer().box(matrices,
                x + getWidth() / 7f + (getContent().length() <= 0 || document.getPointer()
<= 0 ?
                        0 :
RenderManager.instance().getCodeFontRenderer().getWidth(getContent().substring(0,
document.getPointer()))),
                y + getHeight() / 2f -
RenderManager.instance().getCodeFontRenderer().fontHeight / 2f,
                1,
                RenderManager.instance().getCodeFontRenderer().fontHeight,
                System.currentTimeMillis() - last < 500 ? Color.WHITE : new
Color(0x0000000, true) // Render depending on the time
        );

        // If 750ms passed update last
        if (System.currentTimeMillis() - last >= 750) last = System.currentTimeMillis();
    }

    @Override
    public boolean keyPressed(int keyCode, int scanCode, int modifiers) {
        if (!typing) super.keyPressed(keyCode, scanCode, modifiers);

        // Actions to do when key pressed
        switch (keyCode) {
            case GLFW_KEY_BACKSPACE -> document.backspace();
            case GLFW_KEY_LEFT -> document.pointerLeft();
            case GLFW_KEY_RIGHT -> document.pointerRight();
            case GLFW_KEY_SPACE -> onPress();
        }

        return super.keyPressed(keyCode, scanCode, modifiers);
    }

    @Override
    public boolean keyReleased(int keyCode, int scanCode, int modifiers) {
        return super.keyReleased(keyCode, scanCode, modifiers);
    }

    @Override
    public boolean charTyped(char chr, int modifiers) {
        if (!typing) super.charTyped(chr, modifiers);


        document.insert(String.valueOf(chr));
        return super.charTyped(chr, modifiers);
    }

    public String getContent() {
        return document.getContent();
    }
}
```

TextArea.java

https://github.com/jx4e/NEA-Mod/blob/main/src/main/java/com/github/jx4e/minecode/rendering/widgets/text/TextArea.java

Please note: This file has some formatting issues. Best viewed from GitHub.

```
package com.github.jx4e.minecode.rendering.widgets.text;

import com.github.jx4e.minecode.rendering.RenderManager;
import com.github.jx4e.minecode.rendering.theme.Theme;
import net.minecraft.client.gui.screen.Screen;
import net.minecraft.client.gui.screen.narration.NarrationMessageBuilder;
import net.minecraft.client.gui.widget.ClickableWidget;
import net.minecraft.client.util.math.MatrixStack;
import net.minecraft.text.Text;

import java.awt.*;
import java.io.File;

import static com.github.jx4e.minecode.MinecodeClient.mc;
import static org.lwjgl.glfw.GLFW.*;

/**
 * @author Jake (github.com/jx4e)
 * @since 21/09/2022
 **/

public class TextArea extends ClickableWidget {
    private TextDocument document;
    private final Screen parent;
    private long last;
    public TextArea(int x, int y, int width, int height, File editing, Screen parent) {
        super(x, y, width, height, Text.of(editing.getName()));
        this.document = new TextDocument(editing);
        this.last = System.currentTimeMillis();
        this.parent = parent;
        parent.setFocused(this);
    }

    @Override
    public void renderButton(MatrixStack matrices, int mouseX, int mouseY, float delta) {
        RenderManager.instance().getRenderer().box(matrices, x, y, getWidth(), getHeight(),
Theme.DEFAULT.getBackground2());

        if (document.getEditing() == null) return;

        // margin
        float marginWidth = RenderManager.instance().getCodeFontRenderer().getWidth("00000") + 2;
        float marginHeight = RenderManager.instance().getCodeFontRenderer().fontHeight * 3;

        RenderManager.instance().getRenderer().box(matrices, x, y, marginWidth, getHeight(),
Theme.DEFAULT.getBackground3());

        // Saving
        RenderManager.instance().getTextFontRenderer().draw(matrices, "Currently Editing " +
document.getEditing().getName(),
                x + width - RenderManager.instance().getTextFontRenderer().getWidth("Currently
Editing " + document.getEditing().getName()) - 2,
                y + height - marginHeight / 2 - 5,
                Color.WHITE.getRGB()
        );

        float renderY = y + 1;
        for (int i = 0; i < document.getLines().size(); i++) {
            RenderManager.instance().getTextFontRenderer().draw(matrices, String.valueOf(i + 1),
                    x + 1, renderY, document.getPointer().getSecond() == i ? Color.WHITE.getRGB() :
Color.GRAY.getRGB()
            );

            RenderManager.instance().getCodeFontRenderer().draw(matrices, document.getLines().get(i),
                    x + marginWidth + 2, renderY, Color.WHITE.getRGB()
            );

            renderY = y + (i+1) * (RenderManager.instance().getCodeFontRenderer().fontHeight + 5);
        }

        //Render the cursor
        if (parent.getFocused().equals(this)) {
            String cursorLine = document.getLines().get(document.getPointer().getSecond());

            // We want it to be visible for 500ms then gone for 250ms
            // While last<500ms it shows
            // Then it updates at 750ms so that between 500-750 it was not visible
            RenderManager.instance().getRenderer().box(matrices,
                    x + marginWidth + 2 + (cursorLine.length() <= 0 ||
document.getPointer().getFirst() <= 0 ? 0 :
RenderManager.instance().getCodeFontRenderer().getWidth(cursorLine.substring(0,
document.getPointer().getFirst()))),
                    y + 1 + (document.getPointer().getSecond()) *
(RenderManager.instance().getCodeFontRenderer().fontHeight + 5),
                    1,
                    RenderManager.instance().getCodeFontRenderer().fontHeight,
                    System.currentTimeMillis() - last < 500 ? Color.WHITE : new Color(0x0000000,
true)
            );

            // Update as it has passed 750ms
            if (System.currentTimeMillis() - last >= 750) last = System.currentTimeMillis();
        }
    }

    @Override
    public void mouseMoved(double mouseX, double mouseY) {
        super.mouseMoved(mouseX, mouseY);
    }

    @Override
    public boolean mouseScrolled(double mouseX, double mouseY, double amount) {
        return super.mouseScrolled(mouseX, mouseY, amount);
    }

    @Override
    public boolean keyPressed(int keyCode, int scanCode, int modifiers) {
        switch (keyCode) {
            case GLFW_KEY_BACKSPACE -> document.backspace();
            case GLFW_KEY_UP -> document.pointerUp();
            case GLFW_KEY_DOWN -> document.pointerDown();
            case GLFW_KEY_LEFT -> document.pointerLeft();
            case GLFW_KEY_RIGHT -> document.pointerRight();
            case GLFW_KEY_ENTER -> document.newLine();
        }

        if (isPaste(keyCode)) {
            document.insert(mc.keyboard.getClipboard());
        }

        return super.keyPressed(keyCode, scanCode, modifiers);
    }

    public boolean isPaste(int code) {
        return code == GLFW_KEY_V && Screen.hasControlDown() && !Screen.hasShiftDown() &&
!Screen.hasAltDown();
    }

    @Override
    public boolean keyReleased(int keyCode, int scanCode, int modifiers) {
        return super.keyReleased(keyCode, scanCode, modifiers);
    }

    @Override
    public boolean charTyped(char chr, int modifiers) {
        document.insert(String.valueOf(chr));
        return super.charTyped(chr, modifiers);
    }

    public void setEditingFile(File file) {
        if (file.equals(document.getEditing())) return;

        document.save();
        document = new TextDocument(file);
    }

    public TextDocument getDocument() {
        return document;
    }

    @Override
    public void appendNarrations(NarrationMessageBuilder builder) {

    }
}
```

Minecode.mixins.json

https://github.com/jx4e/NEA-Mod/blob/main/src/main/resources/minecode.mixins.json

```
{
  "required": true,
  "minVersion": "0.8",
  "package": "com.github.jx4e.minecode.mixins",
  "compatibilityLevel": "JAVA_17",
  "mixins": [
    "MixinTitleScreen",
    "MixinGameMenuScreen"
  ],
  "client": [
  ],
  "injectors": {
    "defaultRequire": 1
  }
}
```

Minecode.resources.json

https://github.com/jx4e/NEA-Mod/blob/main/src/main/resources/minecode.resources.json

```
{
  "resources" : [
    {
      "name" : "close.png",
      "url" : "https://drive.google.com/uc?export=download&id=1qxu63sPoJU3QAeqPix_gphDXTdP76Qwf"
    },
    {
      "name" : "code.png",
      "url" : "https://drive.google.com/uc?export=download&id=1UxJq8P49EWmuILVHDjtpbhsT8kzZDOId"
    },
    {
      "name" : "document.png",
      "url" : "https://drive.google.com/uc?export=download&id=1grsCZWYrGktoMLgCVQDWlVJTlRsCn1qr"
    },
    {
      "name" : "folder.png",
      "url" : "https://drive.google.com/uc?export=download&id=1rtW0pRHqhL2tU7423LKlL1eYpQP-X-MZ"
    },
    {
      "name" : "gpu.png",
      "url" : "https://drive.google.com/uc?export=download&id=1yD_rOrSEQxpw_jPC4oPLad3DidDxqgnT"
    },
    {
      "name" : "image.png",
      "url" : "https://drive.google.com/uc?export=download&id=1vHi2LutAPcaNuc7A-tyUiFyz4C-WBmsH"
    },
    {
      "name" : "settings.png",
      "url" : "https://drive.google.com/uc?export=download&id=1jx8jY7qlNh2MWXjTBnpf0BsvLvGViyw2"
    },
    {
      "name" : "background.png",
      "url" : "https://drive.google.com/uc?export=download&id=1GibAanM-BX-d7InDB0sfqboSvcoO2McD"
    },
    {
      "name" : "add.png",
      "url" : "https://drive.google.com/uc?export=download&id=1_NE_lUsEynQYvx4EaOSecODcnjsW1W5N"
    },
    {
      "name" : "learn.png",
      "url" : "https://drive.google.com/uc?export=download&id=1wNE204-_q-8k2eh7k0J3IYfpMxVfv2cp"
    },
    {
      "name" : "logo.png",
      "url" : "https://drive.google.com/uc?export=download&id=1pt_3Qj4GqRYRKvQe7vjWYU3Bo8-P4fHq"
    },
    {
      "name" : "back.png",
      "url" : "https://drive.google.com/uc?export=download&id=18NPEUrfBzHiW2yvZYKtpeNeFWZ_3J_T3"
    },
    {
      "name" : "checked.png",
      "url" : "https://drive.google.com/uc?export=download&id=1tpB7DF0-uUY3H60c7VGl0DO1F2KPLbDA"
    }
  ,
    {
      "name" : "template.lua",
      "url" : "https://drive.google.com/uc?export=download&id=1GocGzFIhgBJD-7DlkSpqvVZlrr3UgfES"
    }
    ,
    {
      "name": "tick.png",
      "url": "https://drive.google.com/uc?export=download&id=1X7wDaUFqWaO8HFBkYIG6g6ukM33OfACJ"
    },
    {
      "name": "json.png",
      "url": "https://drive.google.com/uc?export=download&id=1cdyDdnTvmDKB_Ht7d6m07maUSbYQ56C3"
    },
    {
      "name": "run.png",
      "url": "https://drive.google.com/uc?export=download&id=1B91OLzUG-j2U5hELV-o3Pg7OqaSjlQCO"
    }
  ]
}
```

Fabric.mod.json

https://github.com/jx4e/NEA-Mod/blob/main/src/main/resources/fabric.mod.json

```json
{
  "schemaVersion": 1,
  "id": "minecode",
  "version": "${version}",
  "name": "Minecode",
  "description": "",
  "authors": [],
  "contact": {},
  "license": "MIT",
  "icon": "",
  "environment": "client",
  "entrypoints": {
    "client": [
      "com.github.jx4e.minecode.MinecodeClient"
    ],
    "main": [
      "com.github.jx4e.minecode.Minecode"
    ]
  },
  "mixins": [
    "minecode.mixins.json"
  ],
  "depends": {
    "fabricloader": ">=0.14.6",
    "fabric": "*",
    "minecraft": "~1.19",
    "java": ">=17"
  }
}
```

4 | Testing

5 | Evaluation