ProjectOS Algorithms

A MOBILE APPLICATION FOR VISUALIZING OPERATING SYSTEM ALGORITHMS

—

Full Team Member List

1. Team-16(Page replacement)
    - Om Barshakhwala - 19BIT093
    - Ronak Lalakiya – 19BIT112
    - Nil Faldu - 18BIT080
    - Mayank Bhadaraka – 19BIT075
    - Vidit Odedra - 19BIT128

Overview

Page Replacment application which has been merged as a part of ProjectOS mobile application in order to prepare a more complete application for OS algorithms visualization, can run and visualize the various Concurrency and Deadlock algorithms.

Technical Specifications

Language: Flutter(Dart)

IDE: Android Studio or VS code

Getting your Windows Machine ready:

1) Installation of **Flutter**

1.1) System requirements:

To install and run Flutter, your development environment must meet these minimum

Requirements:

Operating Systems:

- Windows 7 SP1 or later (64-bit), x86-64 based *Disk Space:*
- 1.32 GB (does not include disk space for IDE/tools).

Tools:

- Flutter depends on these tools being available in your environment.
- Windows PowerShell 5.0 or newer
- Git for Windows 2.x, with the Use Git from the Windows Command Prompt option. If Git for Windows is already installed, make sure you can run `git` commands from the command prompt or PowerShell.

1.2) Download **Flutter SDK:**

I. Download the following installation bundle to get the latest stable release of the Flutter SDK: flutter_windows_2.0.5-stable.zip

II. Extract the zip file and place the contained `flutter` in the desired installation location for the Flutter SDK (for example, `C:\src\flutter`).

1.3) Update your path

If you wish to run Flutter commands in the regular Windows console, take these steps to add Flutter to the `PATH` environment variable:

- From the Start search bar, enter 'env' and select **Edit environment variables for your account.**
- Under **User variables** check if there is an entry called **Path**:
- If the entry exists, append the full path to flutter\bin using ; as a separator from existing values.
- If the entry doesn't exist, create a new user variable named Path with the full path to flutter\bin as its value.
- You have to close and reopen any existing console windows for these changes to take effect.

1.4) Run **flutter doctor:**

From a console window that has the Flutter directory in the path (see above), run the following command to see if there are any platform dependencies you need to complete the setup:

```
C:\src\flutter>flutter doctor
```

This command checks your environment and displays a report of the status of your Flutter installation. Check the output carefully for other software you might need to install or further tasks to perform (shown in **bold** text).

For example:

[-] Android toolchain - develop for Android devices

• Android SDK at D:\Android\sdk

✗ **Android SDK is missing command line tools; download from https://goo.gl/XxQghQ •**

Try re-installing or updating your Android SDK, visit

https://flutter.dev/setup/#android-setup for detailed instructions.

1.5) Install **Android Studio**

I. Download and install Android Studio.

II. Start Android Studio, and go through the 'Android Studio Setup Wizard'. This installs the latest Android SDK, Android SDK Command-line Tools, and Android SDK Build-Tools, which are required by Flutter when developing for Android.

1.6) Set up your **Android device**

To prepare to run and test your Flutter app on an Android device, you need an Android device running Android 4.1 (API level 16) or higher.

I. Enable **Developer options** and **USB debugging** on your device. Detailed instructions are available in the Android documentation.
II. Windows-only: Install the Google USB Driver.
III. Using a USB cable, plug your phone into your computer. If prompted on your device, authorize your computer to access your device.
IV. In the terminal, run the `flutter devices` command to verify that Flutter recognizes your connected Android device. By default, Flutter uses the version of the Android SDK where your `adb` tool is based. If you want Flutter to use a different installation of the Android SDK, you must set the `ANDROID_SDK_ROOT` environment variable to that installation directory.

1.7) Set up the **Android emulator:**

To prepare to run and test your Flutter app on the Android emulator, follow these steps:

I. Enable VM acceleration on your machine.
II. Launch **Android Studio**, click the **AVD Manager** icon, and select **Create Virtual Device…** ○ In older versions of Android Studio, you should instead launch **Android Studio > Tools > Android > AVD Manager** and select **Create Virtual Device….** (The Android submenu is only present when inside an Android project.)
    - If you do not have a project open, you can choose Configure > AVD Manager and select Create Virtual Device…
III. Choose a device definition and select **Next**.
IV. Select one or more system images for the Android versions you want to emulate, and select **Next**. An *x86* or *x86_64* image is recommended.
V. Under Emulated Performance, select **Hardware - GLES 2.0** to enable hardware acceleration. vi. Verify the AVD configuration is correct, and select **Finish**.

For details on the above steps, see Managing AVDs.

VI. In Android Virtual Device Manager, click **Run** in the toolbar. The emulator starts up and displays the default canvas for your selected OS version and device.

Set up an editor(using VS Code)

2.1) Install VS Code:

VS Code is a lightweight editor with Flutter app execution and debug support.

2.2) Install the **Flutter and Dart plugins**:

I. **Start VS Code.**
II. Invoke View > Command Palette….
III. Type “install”, and select Extensions: Install Extensions.
IV. Type “flutter” in the extensions search field, select Flutter in the list, and click Install. This also installs the required Dart plugin.

2.3) Validate your setup with flutter doctor:

I. Invoke View > Command Palette….
II. Type “doctor”, and select the Flutter: Run Flutter Doctor.
III. Review the output in the OUTPUT pane for any issues. Make sure to select Flutter from the dropdown in the different Output Options.

For more details and to solve some error you can visit to Flutter Installation

How to run?

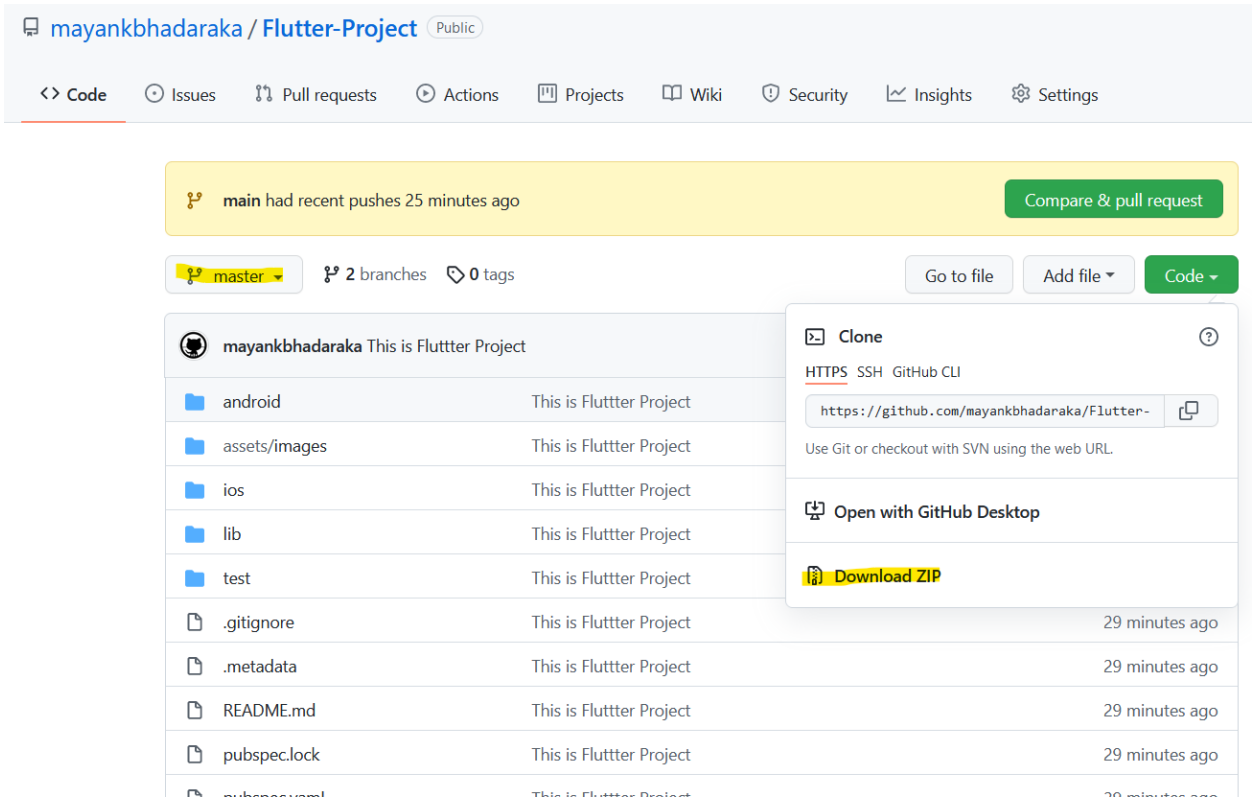
Method 1:

1. Download .apk file for ProjectOS
2. Install this .apk file in your Android Device and you can run Concurrency and Deadlock Algorithms directly on you Android Device

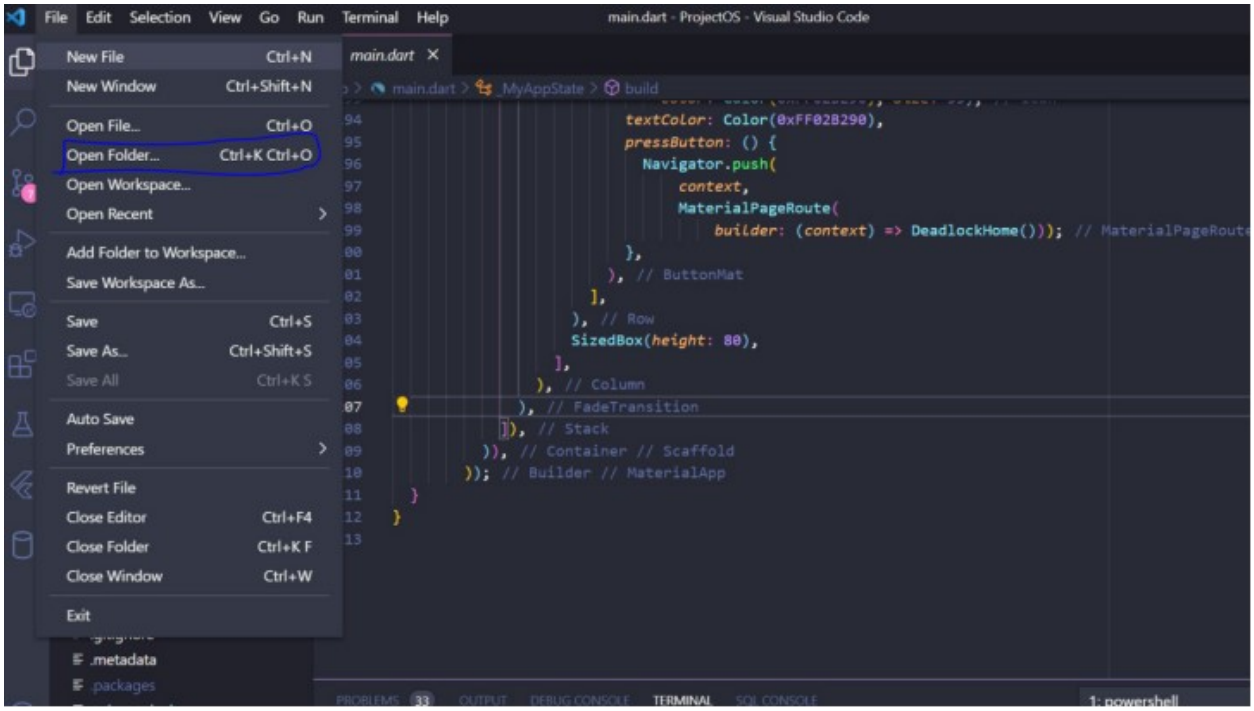
Method 2:

Follow the commands to run this application:

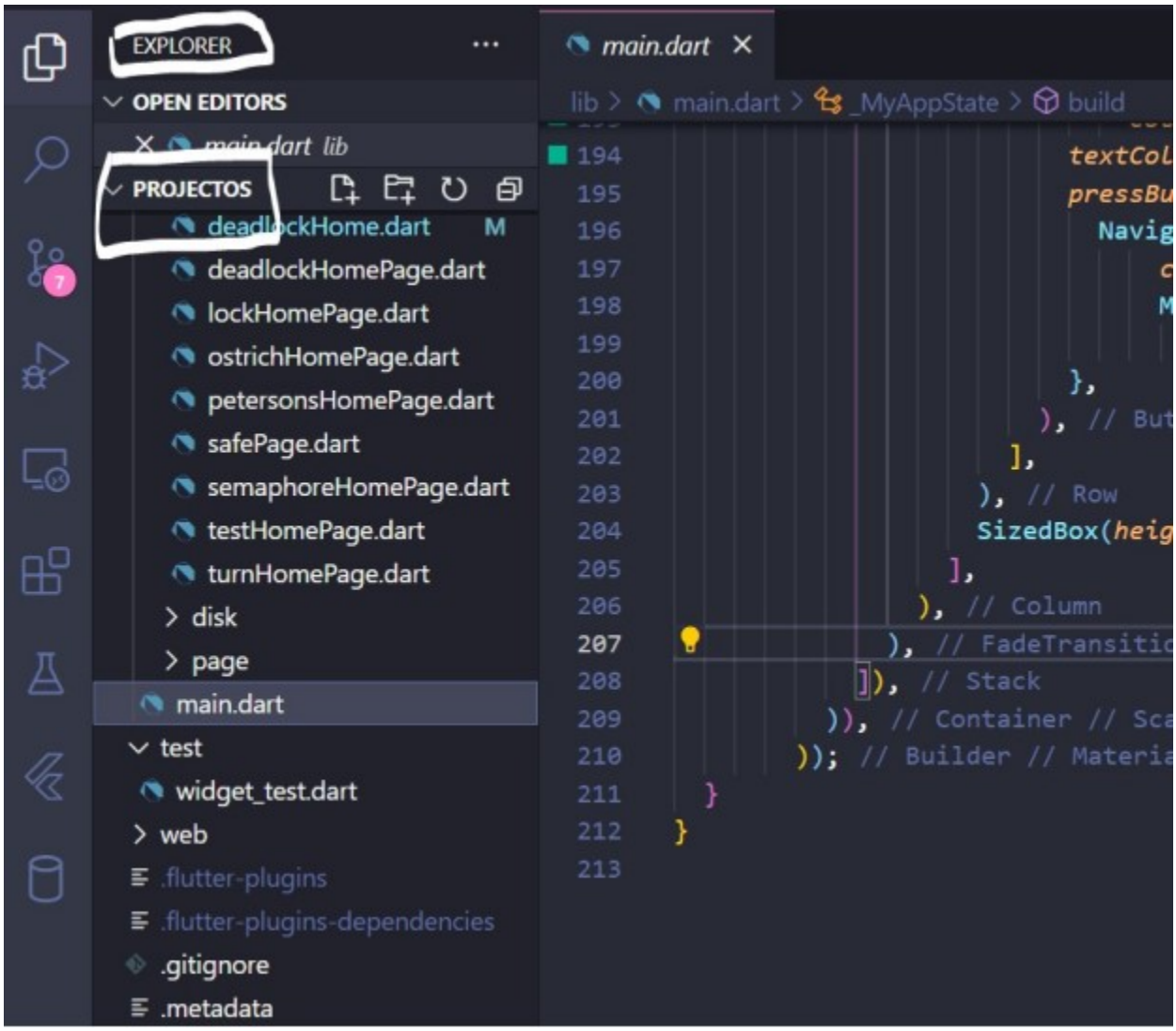
1. Go to GitHub to download the main files for the project.
2. Go to the merged branch.
3. Click on **Code** as shown below.



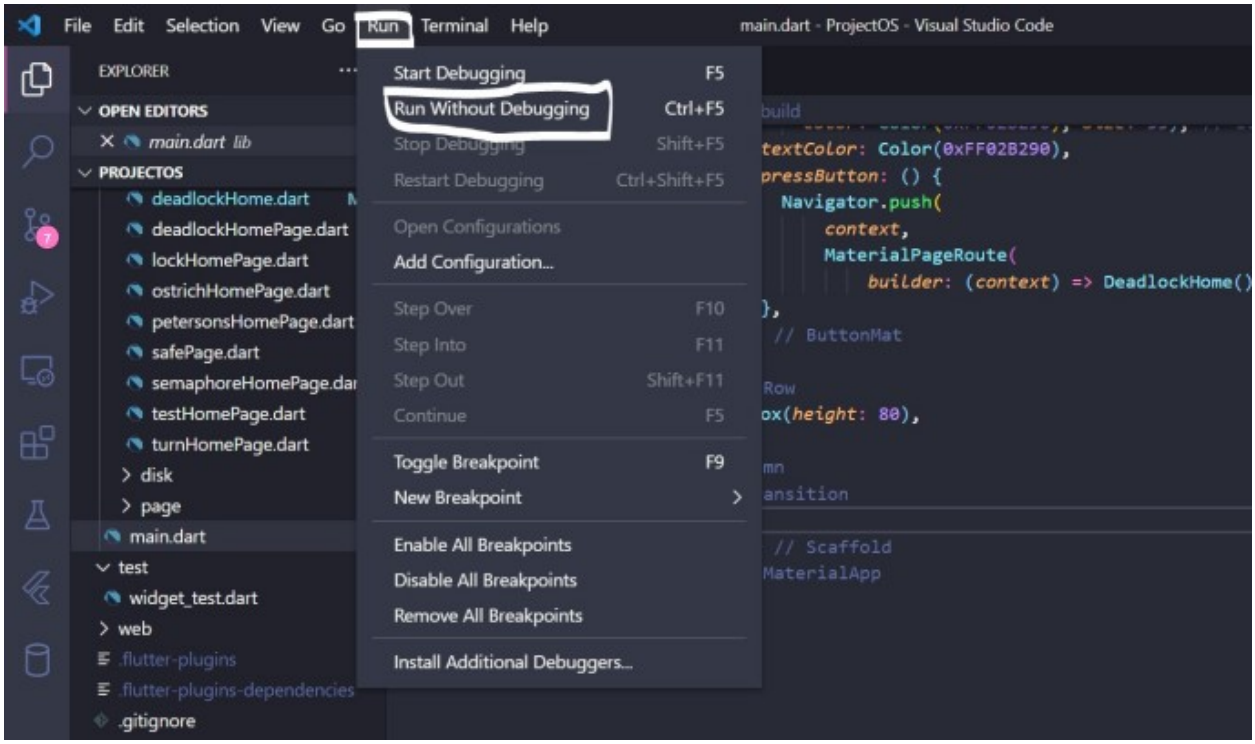
4. Click **Download ZIP** to download the Zip folder for our project.
5. **Extract file contents of Zip folder to any desired location.**
6. Now, open the terminal to move to the Flutter Project location using cd command.
7. Once you are in project's directory use command
   $ **flutter pub get**
8. **After completing all the above steps you can open VS Code and open your Project.**



9. To open your project click **File > Open Folder> Flutter Project. Now, our project is completely loaded your machine**
10. **By clicking on Explorer -> ProjectOS you can see all the files related to this project and using editor you can edit code as you wish.**



11. To run this project, go to Run -> Run Without Debugging for a quicker launch.



12. **The project will launch on the selected emulator.**