Google Summer of Code 2024

International Catrobat Association

POCKET PAINT FLUTTER - MAJOR FEATURES IMPLEMENTATION AND UI ENHANCEMENTS

Submitted by:

Name	Bhav Khurana
University	Indian Institute of Technology, Varanasi
Degree	Bachelor of Technology
Year	Pre-Final (3rd year)
Email	bhav.khurana.cse21@itbhu.ac.in
GitHub ID	bhav-khurana
Timezone	UTC +5:30
Resume	Bhav Khurana – Resume
Postal Address	1108, Hoshiarpur Road, PO Dakoha, Rama Mandi, Jalandhar, Punjab, India, 144023
Phone Number	(+91) 7087341267

SHORT DESCRIPTION

I am Bhav Khurana, a dedicated pre-final year Computer Science student at IIT (BHU) Varanasi, India. With a strong background in Flutter development, I have actively contributed to open-source projects, particularly within the Catrobat organization. My proposal for the Paintroid-Flutter repository aims to:

- Implement a **comprehensive color picker** including the pipette tool
- Implement **drawing tools**: text, spray and watercolor
- Address other pending features: **hide buttons** and **share functionality**
- Introduce **guided tour, app theme customization** for improved user experience
- Introduce automatic **code covergage** reports: Integrating GitHub Actions and Codecov

GSoC PROPOSAL DATA

As a Computer Science student at **IIT BHU Varanasi**, India, I bring a wealth of experience in Flutter development, complemented by an **internship** in a startup SellerSetu focused on Flutter app development. My passion for **open-source contributions** is evident through my active involvement in various organizations, including Catrobat. I'm proficient in languages such as **C/C++, Python, Dart, and JavaScript** and have a keen interest in Software Development, Competitive Programming Machine Learning, Data Structures. I am well-versed in **GitHub** and **Visual Studio Code**. As a member of COPS, my institute's elite coding club, I am a self-driven individual committed to continuous learning. My adaptability and openness to new ideas make me a valuable candidate for the proposed contributions to the Pocket Paint Flutter project under GSoC 2024.

INTEREST

For my GSoC proposal, I aim to elevate the Pocket Paint Flutter repository within Catrobat. The plan includes implementing a **comprehensive color picker**, integrating **drawing tools** like text, spray and watercolor, and addressing pending features such as the **hide buttons and share functionality**. Additionally, I aim to introduce a **guided tour** for an improved user experience. The proposal also encompasses **app theme customization** to cater to diverse user preferences. Apart from these, I have also decided to integrate **GitHub Actions with Codecov** to generate automatic code covergage reports and keep an eye on the quality of the codebase. With my expertise in Flutter development and a proven track record in open-source contributions, I am enthusiastic about the opportunity to contribute meaningfully to the project.

SCRATCH

I have explored the Scratch visual programming language and have made a basic project on it called Honey Collector. Following is the link to the project uploaded on the Scratch Community:
Project link: Honey Collector

POCKET CODE

I have explored the PocketCode app and tried to learn it by making a slightly different version of the famous Flappy Bird game. Following is the link to the project published in the Catrobat community:
Project link: Rotating Flappy Bird

MOTIVATION

I'm excited about this project in Catrobat because it offers a chance to work on a dynamic mobile app. I'm particularly interested in the idea of migrating the app to Flutter from Kotlin, given my experience and enthusiasm for Flutter development. The proposed features, like the color picker and improved drawing tools, align perfectly with my goal of creating user-friendly apps. I also appreciate Catrobat's **collaborative and innovative environment**, which match my expectations from an open source community. I'm looking forward to applying my skills to enhance the repository, making it more **creative and user-friendly**.

CONTRIBUTION

I have actively contributed to Catrobat and have delved into the repositories and available resources. My involvement includes addressing issues, submitting pull requests, and participating in discussions within the community.

Some of the Pull Requests opened by me:

- PAINTROID-455: Add Color Picker
- PAINTROID-704: Remove launch_review package
- PAINTROID-583: Add PR number to apk file output
- PAINTROID-703: Fix bottom overflow
- PAINTROID-701: Addition of Tooltips

Some Issues opened by me:

- PAINTROID-717: Flutter: Run tests using GitHub Actions, Codecov
- PAINTROID-718: Flutter: Add `watch` command for whole project
- PAINTROID-701: Flutter: Addition of Tooltips
- PAINTROID-719: Flutter: Use `.fvmrc` config file
- PAINTROID-703: Flutter: Fix bottom overflow in tools list

MOBILE HARDWARE

For debugging and testing purposes, I will be utilizing my **OnePlus Nord CE2 Lite 5G** as the primary Android device. Additionally, I possess an iPhone 13 Pro, which can be dedicated to testing iOS applications if needed.

TEST-DRIVEN DEVELOPMENT

Test-driven development (TDD) is an approach where we write tests for our code before writing the actual code. We plan what our code should do and how it should behave, and then we create tests to check if it meets those expectations. The main motive of writing tests is to ensure that **known errors** do not get repeated. When we are adding some new features or even refactoring our code at a large scale, it is possible that some of the previously identified bugs reappear as a result of the changes. So, we write tests to ensure those **bugs do not appear** and we do not have to waste time resolving them again.

For example, if we are building a calculator, we would first write a test saying, "When I add 2 and 3, the result should be 5." Then, we write the code for the calculator, making sure it passes the test. This technique helps **catch errors early** and ensures that our code does what it's supposed to.

USABILITY

Usability and User-Centered Design are all about making things **easy and enjoyable** for people (users) to use. When something, like an app or a website, has good usability, it means it is **user-friendly** and does not confuse people. User-centred design is like putting the user at the heart of the creation process. Designers think about what users need, observe how they use things, and then build stuff that fits well with how people naturally want to do things. It is like making a chair comfortable for someone to sit in – we think about what makes it easy and comfy for them, not just what looks good. In short, usability and User-Centered Design make sure that whatever is created is not only good-looking but also easy and enjoyable for people to use.

EXPERTISE

I am currently pursuing Computer Science and Engineering at one of India's premier institutions, providing me with a **strong academic foundation**. My technical expertise extends to **significant contributions** to Catrobat and other open-source organizations like CCExtractor and The Palisadoes Foundation. I participated in the **Inter IIT Tech meets 11.0 and 12.0**, where I developed an interactive learning app called Memor.AI using Flutter and neuroscience principles during the former edition.

Additionally, I gained practical experience through an internship at a startup SellerSetu, where I built an app to optimize food and medicine delivery services for college students. This app will also be published on the **Play Store**.

My proficiency is further demonstrated by being a national finalist in prominent hackathons, including Code with Cisco and Mahindra Catapult Techathon, providing exposure to real-world industry projects. I have also bagged the **2nd runner-up** position at HackOn with Amazon 3.0. These experiences, along with my active involvement in open-source and innovative app development, position me well to contribute effectively to the Pocket Paint project.

HUMAN LANGUAGE SKILLS

I am fluent in **English**, **Hindi** and **Punjabi** languages, both verbally and in written format.

GRAPHIC DESIGN SKILLS

I have experience working with **Figma**, **Canva** and **InVision** for designing UI/UX of applications as well as designing posters/logos for events and workshops. Here are some of the samples:

- Design for a Smart Alarm Clock app
- Poster for a College Workshop

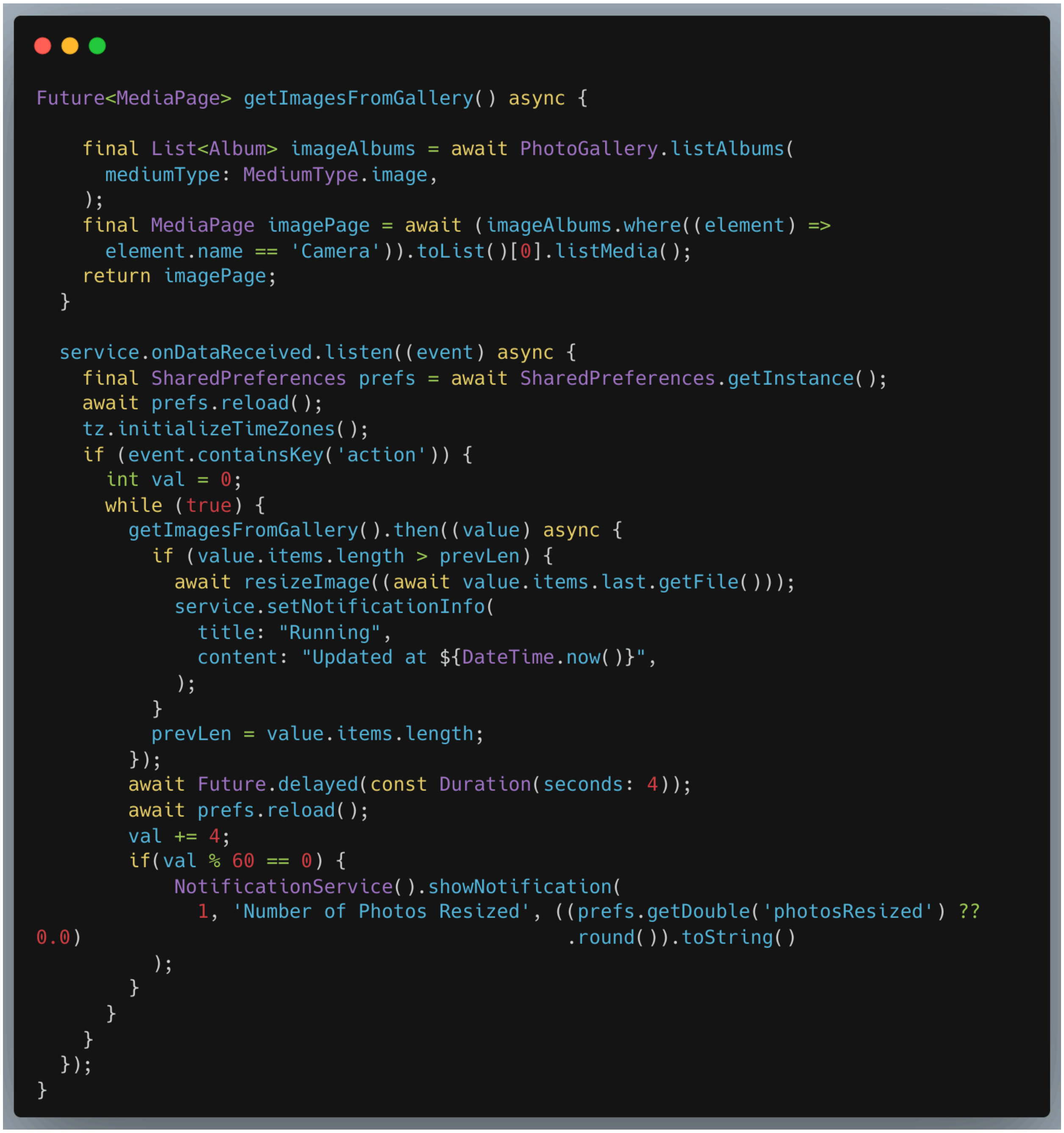
CODE SAMPLES

I have experience in working with different types of projects, starting from the most basic ones, like a ToDo application in Flutter to complicated ones like writing a complete food and medicine delivery app. Apart from Flutter, I have experience writing backend code with **Python and Golang**. Here are some of the interesting projects I have worked on:

- Inter IIT Prep project, which offers a multifaceted approach to address various aspects of health and education. It leverages **OpenAI** API to provide various recommendations and the ability to generate flashcards from PDF. This app was built keeping in mind the need for reusability of widgets in Flutter; hence it contains a lot of **reusable components** which can be used seamlessly in other applications.

- During my internship at a Health App startup, I worked on a **photo resizer** application which had the ability to **run in the background** and **automatically resize** or crop the images taken by the phone's camera. The app provided an interface to customize the resize/crop parameters and allow automatic deletion/non-deletion of the original images. One function which I worked on during the development was utilizing the FlutterBackgroundService plugin for running background tasks and periodically checking for new images in the 'Camera' album of the device's photo gallery. If new images are detected, the code resizes the latest image based on user-defined preferences and saves the resized version. Optionally, it can also delete the original image. Additionally, it displays a notification every 60 seconds (just for testing purposes), indicating the number of photos resized so far.

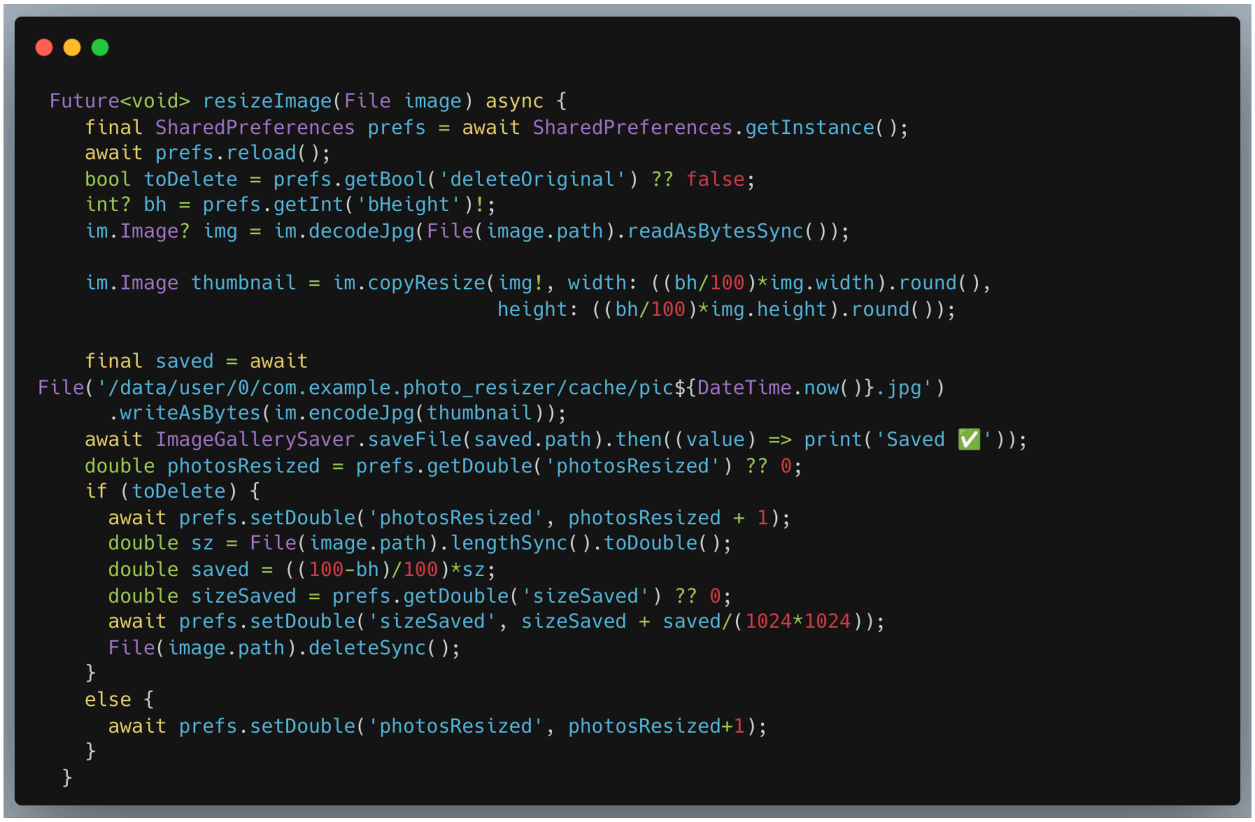
Here’s a code sample for the same:



```
Future<MediaPage> getImagesFromGallery() async {

    final List<Album> imageAlbums = await PhotoGallery.listAlbums(
      mediumType: MediumType.image,
    );
    final MediaPage imagePage = await (imageAlbums.where((element) =>
      element.name == 'Camera')).toList()[0].listMedia();
    return imagePage;
  }

  service.onDataReceived.listen((event) async {
    final SharedPreferences prefs = await SharedPreferences.getInstance();
    await prefs.reload();
    tz.initializeTimeZones();
    if (event.containsKey('action')) {
      int val = 0;
      while (true) {
        getImagesFromGallery().then((value) async {
          if (value.items.length > prevLen) {
            await resizeImage((await value.items.last.getFile()));
            service.setNotificationInfo(
              title: "Running",
              content: "Updated at ${DateTime.now()}",
            );
          }
          prevLen = value.items.length;
        });
        await Future.delayed(const Duration(seconds: 4));
        await prefs.reload();
        val += 4;
        if(val % 60 == 0) {
            NotificationService().showNotification(
              1, 'Number of Photos Resized', ((prefs.getDouble('photosResized') ??
0.0)                                          .round()).toString()
          );
        }
      }
    }
  });
}
```



```
 Future<void> resizeImage(File image) async {
    final SharedPreferences prefs = await SharedPreferences.getInstance();
    await prefs.reload();
    bool toDelete = prefs.getBool('deleteOriginal') ?? false;
    int? bh = prefs.getInt('bHeight')!;
    im.Image? img = im.decodeJpg(File(image.path).readAsBytesSync());

    im.Image thumbnail = im.copyResize(img!, width: ((bh/100)*img.width).round(),
                                       height: ((bh/100)*img.height).round());

    final saved = await
File('/data/user/0/com.example.photo_resizer/cache/pic${DateTime.now()}.jpg')
      .writeAsBytes(im.encodeJpg(thumbnail));
    await ImageGallerySaver.saveFile(saved.path).then((value) => print('Saved ✅'));
    double photosResized = prefs.getDouble('photosResized') ?? 0;
    if (toDelete) {
      await prefs.setDouble('photosResized', photosResized + 1);
      double sz = File(image.path).lengthSync().toDouble();
      double saved = ((100-bh)/100)*sz;
      double sizeSaved = prefs.getDouble('sizeSaved') ?? 0;
      await prefs.setDouble('sizeSaved', sizeSaved + saved/(1024*1024));
      File(image.path).deleteSync();
    }
    else {
      await prefs.setDouble('photosResized', photosResized+1);
    }
  }
```

- I have also worked on a project AeroEase for the Inter IIT Tech Meet 12.0 held at IIT Madras, which is a paradigm shift in aviation disruption management and employs **Quantum Algorithms**, specifically focusing on Quadratic Unconstrained Binary Optimization (QUBO) on annealing-based quantum computers.

TOOLCHAIN

I have extensive experience using **Git and GitHub** for version control, ensuring a seamless workflow and organized collaboration. During my contributions to Catrobat, I've effectively utilized the **Jira** issue tracker for streamlined issue management. For Flutter development, I rely on the versatility of **Visual Studio Code** and **Android Studio**. It allows me to contribute effectively to projects by ensuring version control, organized issue tracking, and a streamlined development environment.

REMOTE COOPERATION

Any **standard communication channels** (preferably Slack) are acceptable to me. I would be able to work full time on weekdays and would be available between **1:00 PM IST (7:30 AM UTC) to 1:00 AM IST (7:30 PM UTC)**, which is flexible and can be modified according to need.
On weekends, I'd rather like to **interact with the mentors** and the team to learn from them and work on any problems that may arise.
I'll also keep my mentor(s) updated in case of any emergency or modification in the schedule.

ROUGH TIME SCHEDULE

Period	Tasks
Community Bonding Period [May 1 - May 26]	- During this phase, I will engage in active communication with the Catrobat community to better understand the project's nuances and collaborate with mentors. I will familiarize myself with the existing codebase and finalize the design specifications for the proposed features.
Weeks 1 and 2 [May 27 - June 8] **Color Picker Implementation, Integrating Codecov**	- Refine and finalize the color picker module initiated in the Pull Request along with Pipette tool - Implement design adjustments to align with the original application. - Integrate GitHub Actions and Codecov for automatic code coverage reports
Weeks 3 and 4 [June 9 - June 22] **Drawing Tools Implementation: Text Tool**	- Develop the Text Tool, allowing users to add text with adjustable font size and color. - Create a custom painter class handling text properties and rendering logic. - Integrate the custom painter within the main widget for dynamic text rendering based on user input.

Weeks 5 and 6 [June 23 - July 7] **Drawing Tools Implementation: Spray Can Tool**	- Implement the Spray Can Tool, simulating a spray effect in the canvas rendering. - Develop a custom painter for randomized point rendering to achieve the spray effect. - Integrate touch event handlers in the main widget for user interaction and control.
Mid-Evaluation [July 8 - July 12]	- Review progress with mentors and receive feedback. - Address any necessary adjustments or refinements to the implemented features.
Weeks 7 and 8 [July 13 - July 27] **Drawing Tools Implementation: Watercolor Tool**	- Implement the Watercolor Tool using Flutter's shaders and blending modes. - Provide adjustable opacity, flow, and color options for a realistic watercolor painting simulation. - Ensure seamless integration with the existing canvas.
Weeks 9 and 10 [July 28 - August 11] **Share Image Functionality, Hide Buttons, and Other Unimplemented Features**	- Implement platform-specific sharing functionalities for Android and iOS. - Develop a dedicated share image button for effortless distribution. - Implement the ability to show or hide buttons dynamically. - Identify and address any other unimplemented features from the original application.
Weeks 11 and 12 [August 12 - August 26] **Splash Screen, Guided Tour, App Theme Customization**	- Set up a captivating splash screen for a polished app launch. - Create a guided tour using Flutter's IntroductionScreen package for a step-by-step user tutorial. - Enable app theme customization, allowing users to personalize the interface with dynamic color themes.

Final Evaluation [August 26 - September 2]	- Review the completed features and their integration into the app. - Address any final feedback from mentors and the community. - Ensure that the codebase is well-documented and follows the best practices.

INTEGRATION

To achieve the proposed features, I'll break down the project into **smaller tasks**. For example, within the "Color Picker Implementation," tasks include refining UI elements, implementing color selection logic, and integrating it with the canvas. Similarly, for "Drawing Tools Implementation," tasks encompass developing each drawing tool individually, handling user interactions, and ensuring seamless integration. I plan to submit pull requests for **code review at significant checkpoints**, such as completing a drawing tool or finishing a feature. This will allow **regular feedback** and **collaboration** with mentors and the community throughout the coding period. Implementation of all the features will be followed by **writing tests and documentation** for the same and following the procedures of **agile development**. I plan to devote 150 minutes (2.5 hours) every day, six days a week, to the project for 12 weeks, totaling **180 hours**. I would happily put in extra hours if the project demands the same.

COMMENTS

If any task or feature remains unimplemented, I will work to fulfil them after GSoC. I will continue contributing to improving the project and resolving any problems that may arise.
Apart from that, I will continue participating in Catrobat's discussion channels and get exposure to new ideas and technologies. I would love to be of any help to the community even after the GSoC period.

Thank You!