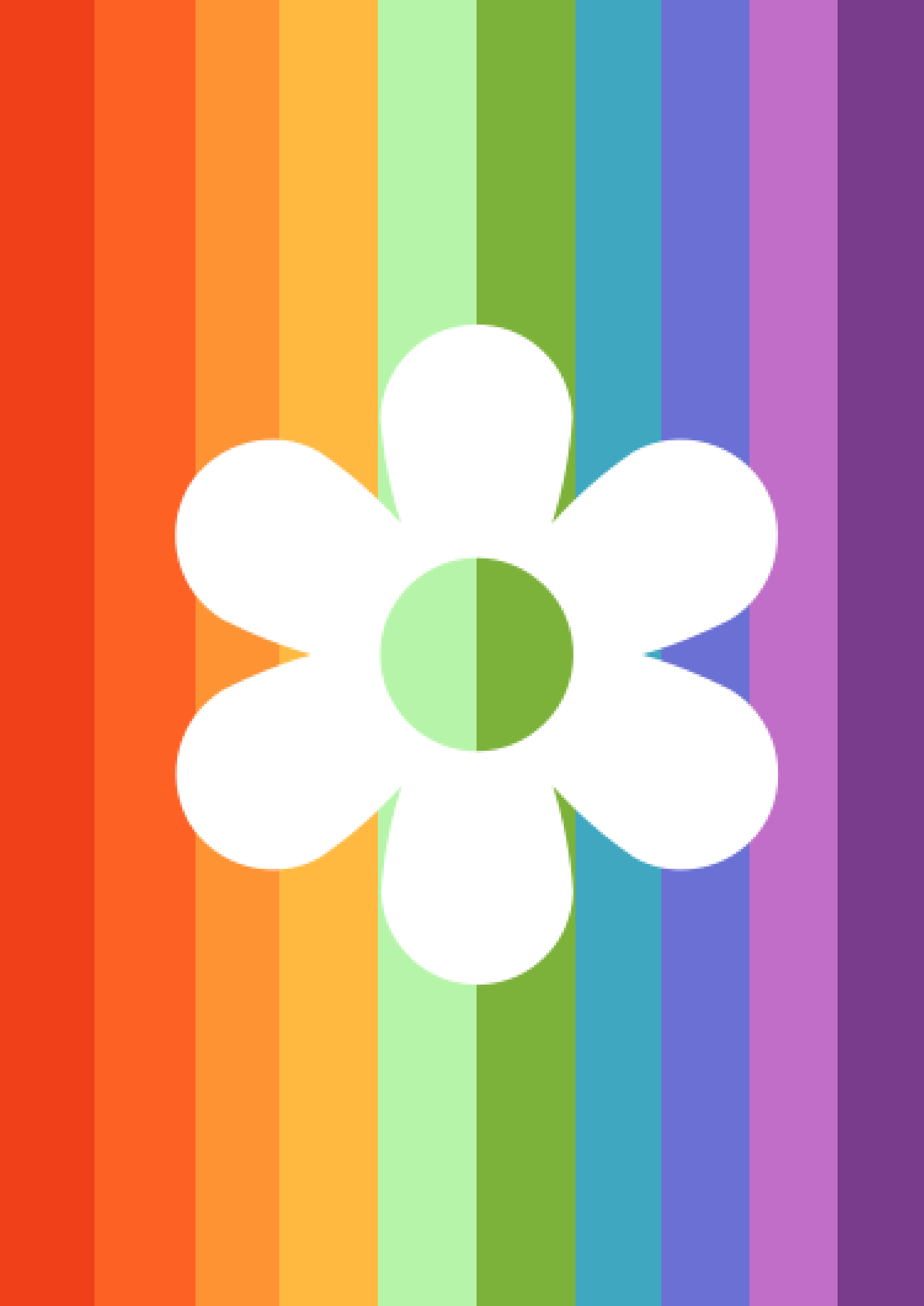

Contribution Statement – Drippy Flower Field

Introduction

The Drippy Flower Field is an interactive program in which users create a color palette using the camera, generate flowers on the screen by drawing on it and then simulate a drip painting. The program involves creativity with a bit of randomness. Users should have fun drawing flower fields that are pleasant to look at. Their time is limited in this section as perfection and attachment aren't the goal; just quick and fun art. In the dripping phase, they are encouraged to destroy the flowers which creates a new piece of art.

The work was divided among the three of us. We each worked on different portions of the code that we assigned each week and then the pieces were integrated by one person.

Team Contribution Overview

Every week during class on Monday we discussed what changes and features we wanted to add to the project. We created a goal for what we wanted to have implemented by the end of the week and divided up the work. For example, in the first week, Sidra created the mouse interaction in the flower drawing phase, Sophie created a class for flowers and Kathi created a drip painting program. The code was easily combined to create our first prototype. Communication throughout the week was done through WhatsApp and documentation of our design plans and feedback in class was shared on our mood board in Figma. Updated versions of the full code were shared in a Google Drive Folder.

Main tasks:

Katharina Borgs	**Sophie Kunz**	**Sidra Wernli**
Dripping Function	Flower drawing function	Fundamental drawing function
color palette implementation	sound implementation	Camera function
Assembling code		Optimization

Individual Contributions

Katharina Borgs

I worked on the dripping effect in the third phase of the program. I wanted the drips to flow over time and having a teardrop shape. The drops exist of tiny circles and one problem was that they rather flow very slowly or had gaps. I solved this problem through a calculation with lerp() that drew the intermediate circles.
I implemented the time-based switching between the three phases of the program and the two intermissions as well as the save function.
I was responsible for the color palette, that was assembled with the five most dominant colors of a picture. To prevent the color palette existing of too similar colors I inserted a short calcutation with the variable “dist” in the “equalsColor” function that grouped similar colors together. One issue was the quality of the camera, the photos were dull, as well as the color palette. The solution was to rewrite the code in HSB colormode and to enhance saturation and brightness of the palette colors.
We wanted to integrate the original photo even more, therefore some flowers function as a mask that displays a part of the original photo.
At the end of each week, I put the code samples from the other two members together. This was rather uncomplicated since everyone had their own subject area, therefore only small adjustments were necessary to make the code running.
I think I contributed evenly to the project and overall could help with organisation and time management.

Sidra Wernli

I worked on the main drawing in the flower generation phase that reacts to the speed of the mouse, the code that takes a picture of the user to be used in the color palette and the flower designs. I also helped Kathi with the transition between the flower and dripping phases because we wanted a transparent layer to cover the flowers with the text “Now give the flowers some drip!” Since the draw function continuously draws this, the transparent layer was being covered until it was no longer transparent. I was able to fix this with the delay() function. I faced similar issues trying to create a pause during the countdown before the program takes a picture. The delay() function would have prevented the screen from updating the camera input so I implemented a timer using millis() to keep each number in the countdown on the screen for one second.

Overall, I think I was able to contribute evenly to the project. I completed different tasks each week and was able to contribute to some key components of the project.

Sophie Kunz

My contribution to the project began with the creation and implementation of the first prototype class for the flowers. I designed the initial structure that allows flower objects to be created and

managed dynamically using an ArrayList. I tested two different flower types to explore design variations and structure, that were later expanded into even more complex flower design by Sidra. Later I took responsibility for the sound of the project. I selected audios that would enhance the experience and boost interactivity. I implemented these sounds into the code and decided to make the flower sound responsive to user input by tying the pitch to the mouse movement. Now the mouse doesn't only create and influence the flower´s appearance but also the sound they make. As for the drip sound effect, I decided to loop the audio as long as the user is clicking and holding an area on the screen.
In the final stage of the group project, I took on the task of creating the demo video, presenting and explaining our coding project.
For the sound I struggled with volume control when the amp() function didn't work reliably on such a short sound. To fix it, I lowered the volume manually in Audacity for a more comfortable experience.
When I first implemented the sound to react to user input, it was triggered too frequently, since many elements got created at once. This distorted the audio and made it overwhelming. I solved it using an if clause, that checks if the sound is already playing, so it only triggers once per interaction.
Another challenge I faced, was when I wanted the sound to be interactive. My first approach was very basic: I used an if clause that sets the rate either high or low depending on the speed of the mouse. But we wanted a smoother, continuous change. To solve this I used the map() function to tie the pitch to mouse speed, allowing for a more natural and dynamic sound.

Teamwork Reflection

We found that we were able to work together very effectively. The division and completion of tasks worked very well. Everyone finished their portion on time each week and there were no major difficulties in putting the code together. During brainstorming, we may have had different ideas on how to evolve the code but in the end, we always found a solution that everyone was satisfied with. Improvements we could have made is working more closely together. While the code works well, it could be cleaner because it is made up of many different pieces written by different group members.

Lesson that we have learned for future projects are that direct communication is important. Waiting for an issue to solve itself can make it worse. Updating the group, even for minor problems, can be helpful for general communication. Additionally, setting clear individual tasks and deadlines for each week was very helpful in keeping us on track with the project and dividing up the work evenly. Finally, getting everyone on the same page so each group member understands and agrees with what the project looks like. We found that Figma Board was a very helpful tool for creating a mood board as well as documenting and discussing ideas. It is a must have for creative group projects.