Happy Little Accidents(?)

Authors: Alex Zhang, Julia Jeong, Mira Haldar
Revision: 5/1/22

Introduction:

In the game, the user is an artist. The main goal of the program is for the user to just enjoy their time painting. They can buy paints and use them to color the piece they're working on. They can customize the paints they have by mixing black or white into their colors, creating a set of tints/shades. They can also write about art-related stuff on a blog (it's a typing test, basically) to earn money for more paints! If they successfully complete one painting, the game ends.

The game is directed at people who like a little variety in their zen coloring apps.

Instructions (has to be specific):

Some aspects of the game will run on mouse interaction: painting (clicking on the color you want, dragging to draw lines, shape in the drawing the user wants to select, selecting the option to mix the colors), clicking on the start button, clicking on the instruction button, clicking on the finish button, clicking on the start-over button, clicking on the mini game option.

Other aspects of the game will run on keyboard interaction: the typing game itself

HOME PAGE: The user will start with a home page with two options in the center that the user can click.

1. Button to see the instructions (leads to different screen)
2. Button to start the game (also leads to different screen)

The home page will also have a background picture that represents the game

IN GAME: during the game, the majority of the screen will be the drawing canvas. On the top right side, there will be option (button) to:

1. Go back to the instructions without losing the user's progress
    a. New screen (same screen as home page instructions)
2. Create shades and tints of one color
    a. Click on color then click on Mix
    b. New screen [color is pre-selected]
    c. Select "very little", "little", "a bit", or "a lot" of the white or black
3. Fill in a closed shape (drawing mode 1)
    a. Click on the "fill" option and then select a closed shape on the canvas
    b. Able to keep selecting a different shape without pressing on "fill" option every time
4. Draw freestyle lines (drawing mode 2)
    a. Click on "draw" option, select a color, and then drag your mouse on the canvas
    b. Able to keep dragging and drawing without pressing on "draw" option every time

5. Buy the paint
    a. Select a color and then click buy on the window
    b. New screen
    c. Typing game: the user has to type a paragraph about art related things as fast as they can within a minute
    d. the faster you type and the more accurate, the more paint the user receives
6. Start over
    a. Press a start over button
    b. Loses all the colors the user bought or created
    c. New blank canvas
7. Press done to finish the drawing
    a. New screen
    b. Go to end screen where the final drawing is framed and displayed to the user with a congratulations message
8. Exit the game
    a. Loses all progress and will go back to home screen when user clicks on the program again

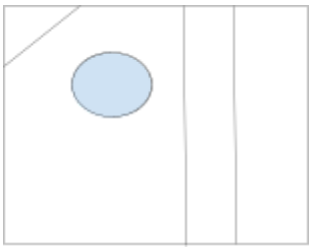
The beginning of the game should follow a certain order, though the game after the first few steps should be open-ended, meaning the user could paint, play the mini games, and mix colors in whatever order they want. The basic order and flow of the game should be in the following order: trying to get money by playing the mini game, purchasing the paint, creating shades and tints based on colors the user has, and painting an area by pressing a fill button and then selecting a color closed shape or just dragging with your mouse. This can be repeated until the user presses finish.

Features List (THE ONLY SECTION THAT CANNOT CHANGE LATER):

Must-have Features:

[These are features that we agree you will *definitely* have by the project due date. A good final project would have all of these completed. At least 5 are required. Each feature should be fully described (at least a few full sentences for each)]

1. The project should have an instruction screen. There will be an option to view these instructions on the home page and the drawing canvas screen. It will be a separate screen. Clicking on the instructions button during the game will not make the user lose their progress.
2. The project should contain the drawing canvas screen. This screen shows all the options the user can do while playing the game, the drawing canvas itself, and a color palette of all the colors the user currently has (below each color displays how much of the color is left)
3. The user should be able to choose a color and then fill a closed shape with it.



4. The project should have a mixing screen where the user can alter the colors they have by creating tints and shades. That way, they can have a set of colors that come from the same hue, but are lighter(the user added x amounts of white) and darker (the user added x amounts of black).
    a. However, the more colors the user has in the set, the less of each color they have and the faster they'll run out. For example, if they originally had 5 oz. of Red and then they create 4 shades, they'll have 1 oz. of each. That's also the max amount of shades they can create from one color.
5. The project should have an option for the user to buy the paint. They would select the color they want from a color bank displayed to them, then play the typing game The amount of color they receive depends on their performance on the typing test. The color will now be displayed on the drawing canvas screen.

Want-to-have Features:
[These are features that you would like to have by the project due date, but you're unsure whether you'll hit all of them. A good final project would have perhaps half of these completed. At least 5 are required. Again, fully describe each.]

1. The user has an erase button for the painting. If they fill an area with color, they should be able to erase it by pressing the erase button and then clicking the area they wish to erase.
2. They might also want an undo button. The undo button undoes the color on the last area the user painted, or it undoes an erase move if that was what the user's last move was instead.
3. The user picks between a landscape drawing or portrait
4. Gradient fill for a specific area or blending colors (not just white and black)
5. a zoom-in/zoom-out button on the painting in order for the user to color in the painting as best as they can.
6. Pressing a "finished" button. The user would go to the end screen where the final drawing is framed and displayed to the user with a congratulations message
7. A starting over option where everything goes back to how it was in the beginning. The user loses all progress in their drawing, and colors.
8. An option to exit the game?

Stretch Features:
[These are features that we agree a *fully complete version of this program would have, but that you probably will not have time to implement*. A good final project does not necessarily need to have any of these completed at all. At least 3 are required. Again, fully describe each.]

1. Brush Strokes— airbrush, metallic, patterns, etc.

2. Extracting outlines from a bank of photos, or a photo that the user uploads
3. Having a gallery of all the different past finished drawings
4. Being able to import actual photos from the internet
5. Being able to download the paintings as picture files such as jpegs or pngs
6. The user has a 'studio' of unfinished paintings. That way, the user can switch back and forth and choose which painting they want to work on and have that on the main screen

Class List:

[This section lists the Java classes that make up the program and very briefly describes what each represents. It's totally fine to put this section in list format and not to use full sentences.]

- Palette– to store Colors. There'll be a Palette of all colors.
- MixedPalette- extends Palette and it's to store shades
- Paint– availability and amount: also, when it clicked on, it shows a window with all the relevant info + buttons: for going to the mixing window, getting already mixed shades, filling, or going to the buyingColor window depending on availability
- Painting - The actual painting. Contains the outline, and can be colored
    - stores the current Color being used
- TypingGame - the mini game that we will implement.
- Main
- Button-
- PaintButton – extends button, has a Paint, and creates a window when pressed
- IntroScreen- users can go to instructions, or start the
- InstructionsScreen
- PaintingScreen
- Window- happens when you click a PaintButton
- MixingScreen
- BuyingColorScreen
- TypingScreen
- EndScreen
- ScreenSwitcher (interface)
- Drawing surface
- Screen

Main uses DrawingSurface
DrawingSurface implements ScreenSwitcher
All the Screens extend the Screen class
All the Screens have a DrawingSurface??? i think???
PaintingScreen has a Painting and a Palette
Palette has Paint
MixedPalette extends Palette
Paint has/uses MixedPalette
Paint has windowButtons

MixingScreen has a MixedPalette
MixedPalette uses Paint(i feel like we can leave this relationship off the UML tho)
TypingScreen uses TypingGame

Credits:
[Gives credit for project components. This includes both internal credit (your group members) and external credit (other people, websites, libraries). To do this:

- and then a bunch of text files for the typing game

Things to do that might need different screens/classes:
Instructions, Outlines, Painting, Pricing the Painting, TypingGame, Mixing Color, Choosing Size of Color, Palette(stores all possible colors and all their properties— clicking on color leads to more buttons leads to diff screens), Bank Account things

- Julia:
    - Instructions of program, finalizing readme, IntroScreen, InstructionsScreen, MixedPalette, working on PaintingScreen
- Alex:
    - Classes of program, readme rough draft, TypingGame, UML Diagram
- Mira:
    - Features of program, readme rough draft, Painting, Palette, Button, PaintButton, working on PaintScreen

Give credit to all outside resources used. This includes downloaded images or sounds, external java libraries, parent/tutor/student coding help, etc.]
for text:
https://xd.adobe.com/ideas/process/ui-design/what-is-color-theory/