Send A:
Project Deliverable 4

Team Members:
Anthony Le,
Birathan Somasundaram,
Pratana Atikhomkamalasai,
Suxin Hong,
Yuewen Ma

Table of Contents

Table of Contents 1

User Guide 2

Overview 2

Setup 2

How to use this new feature 2

Code Design 4

Existing components modified 4

Components be added 5

Acceptance Test 8

Unit Test 14

User Guide

Overview

This new feature allows users to have quick and intuitive control over Firefox-Focus. By clicking the expand button, bottom right of the screen, a pie menu of 3 sub menu buttons will pop up. This is for users to access common used functionalities such as erasing history, refreshing, or go back.

Setup

1. Make sure Android Studio has environment built correctly and open it
2. Clone the project from team repository
3. Check out branch expand-pie
4. In Android Studio, select the build variant as focusX86Debug
5. Run the project with an emulator that has API 21+

How to use this new feature

1. navigate to any website by typing the URL in the search address
2. the expand button will show up at the button right of the screen
3. It functions like Assistive Touch on iPhones
4. Drag buttons to the left or right of the screen, it will latch on to the screen.
5. The buttons are expanded towards the center of the screen (i.e if it's on the left edge it will expand to the right and vice versa.
6. Scrolling down the page will deactivate the expand button , it will reactivate when scrolling up.
7. In case of the expand disappeared, scrolling up the page will also reactivate the button.
8. This is the original design of the delete button in firefox, we chose to keep it for familiarity.

9. Once click, it will open a sub button menu
10. Click the refresh button (refresh icon) will refresh the page.
11. Clicking the delete sub Button (bin icon) will delete browser history and bring users back to home menu
12. Click the previous sub Button (left icon) will go back to the page.
13. Click again to close a sub button menu

For more information on usage, please refer to the gif.

Code Design

Existing components modified

widget/FloatingEraseButton (rename to **FloatingExpandButton**)[1]

We plan to use the original erase button as an "Expand" button. We need to change the icon's UI and the function from erasing history to expanding the sub buttons. We will keep the changes we made in the previous deliverable so that the button still can be moveable. Every time when the "Expand" button location changes, we will update the start angle and end angle in the "Expand" menu so that the sub buttons will expand in the correct direction. The visibility of sub buttons should follow the visibility of the "Expand" button, so when we show or hide the "Expand" button, we also need to set visibility to each sub button.

Fragment/BrowserFragment.kt -> **initialiseNormalBrowserUi**[1]

Initialize widgets the "Expand" button, sub buttons, and the expanded menu in BrowserFragment. Replace UI of "Expand" button to "X" instead of bin icon, add sub buttons (erase, refresh, and back) to the "Expand" menu, and also attach "Expand" button to it as well. Implement the setStateChangeListener interface so that we can add animation during the menu opening and closing.

Fragment/BrowserFragment.kt -> **onClick**[1]

Override the **onClick** method so that each button can listen to users' actions and do the corresponding functions based on the id of each sub button such as delete history, refresh the page, and go back to the last page in the onClick method.

res/value/dimens.xml[1]

Add necessary resources to define the size of SubActionButton, FloatingActionMenu, and FloatingExpandButton such that it fits the screen.

[1] before "/" is the package name, after "/" is the file name and "->" means the functions modified inside the file

Components be added

widget/*SubActionButton.java[2]
First, we need to add the icons for erase, back, and refresh buttons. We will reuse the icons from the original repo. SubActionButtons will be expanded from the "Expand" button. We need to extend FrameLayout using a builder design pattern. This is to standardized the code and makes it easier for us to initialize.

widget/*FloatingActionMenu.java[2]
The "Expand" button should be attached to the FloatingActionMenu. This new class contains the main listener for the expand button to delegate the tasks appropriately These tasks include opening and closing the expanding button, initializing sub buttons and attaching it. It also contains inner methods used in the expansion such e.g. angle calculation. We will implement it using a builder design pattern since there are many buttons to be initialized and all of them follow the same initialization process.

widget/*FloatingActionMenu.java -> **MenuStateChangeListener**[2]
We need an inner class to Set a listener for FloatingActionMenu to check when it is open or closed. This is for the menu animations to be selected appropriately during the expansion or closure.

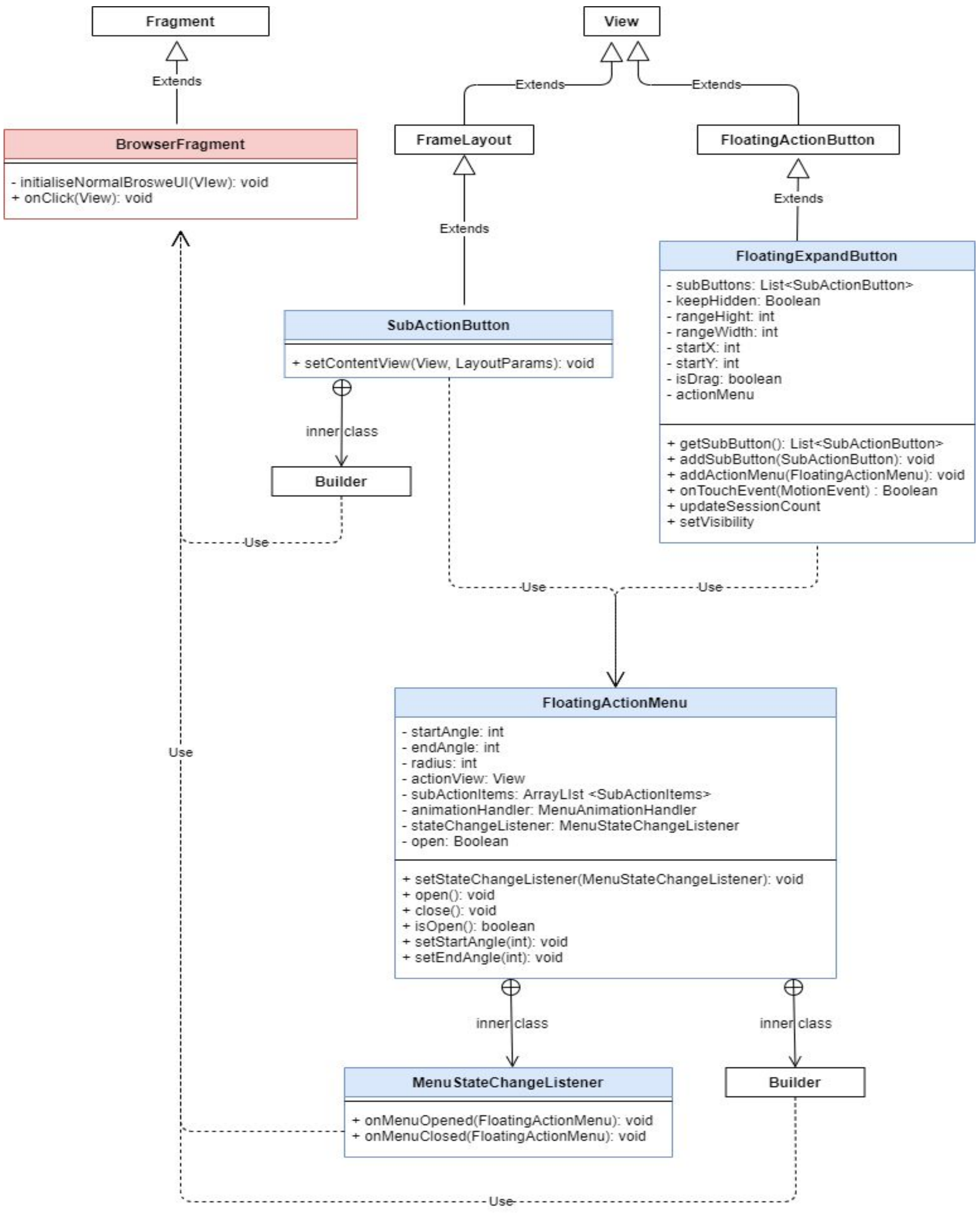
Fragment/BrowserFragment.kt -> **initialiseSubButton**[2]
This class initializes sub Buttons UI and functionality. We need to implement **setOnClickListener**, create a function id to each sub button and add the sub buttons to the "Expand button".

res/drawable/*button_action.xml, button_sub_action_selector.xml[2]
Use selectors to select and display the buttons in different states.

res/drawable-hdpi/*xxx.png[2]
Add front end resources for SubActionButton backgrounds in different states.

[2] "*" means the files added into the project

Overall, the design of our solution is similar to our initial plan, but we also made some changes. Since The FloatingActionMenu does not have a method for setting the expanded angle and We need the angle to calculate the direction of the expansion, We decided to add a method to set the expanded angle based on the button's current location. Everytime users release the button , the angle is reset. This is for the expanded button to expand correctly in the right direction. As a result, We do not need the getOnLeft method to track button position like planned.



Fragment
Extends
BrowserFragment
- initialiseNormalBrosweUI(View): void
+ onClick(View): void
View
Extends
Extends
FrameLayout
FloatingActionButton
Extends
Extends
FloatingExpandButton
- subButtons: List<SubActionButton>
- keepHidden: Boolean
- rangeHight: int
- rangeWidth: int
- startX: int
- startY: int
- isDrag: boolean
- actionMenu
+ getSubButton(): List<SubActionButton>
+ addSubButton(SubActionButton): void
+ addActionMenu(FloatingActionMenu): void
+ onTouchEvent(MotionEvent) : Boolean
+ updateSessionCount
+ setVisibility
SubActionButton
+ setContentView(View, LayoutParams): void
inner class
Builder
Use
Use
Use
Use
FloatingActionMenu
- startAngle: int
- endAngle: int
- radius: int
- actionView: View
- subActionItems: ArrayLIst <SubActionItems>
- animationHandler: MenuAnimationHandler
- stateChangeListener: MenuStateChangeListener
- open: Boolean
+ setStateChangeListener(MenuStateChangeListener): void
+ open(): void
+ close(): void
+ isOpen(): boolean
+ setStartAngle(int): void
+ setEndAngle(int): void
inner class
inner class
MenuStateChangeListener
+ onMenuOpened(FloatingActionMenu): void
+ onMenuClosed(FloatingActionMenu): void
Builder
Use

Classes in white are the classes in android or the builder class for construct objects. We are not going to show the attributes and methods since they are too many.
Classes in blue are those we need to add into the project, the FloatingExpandButton is the one to replace the original FolatingEraseButton.
Classes in red is the existing component in the project, we are not going to show all the existing methods of it since there are too many, so we just show the method we need to add or modify.

Acceptance Test

Acceptance Citeria

- Clicking the expand button should expand the sub menu-buttons back, delete, refresh in the right angle and correct order
- Each sub menu-button should work according to its corresponding function
- Clicking on the expand button again should close all the the sub menu-buttons
- The sub menu-button should be draggable and stick to the right side of the screen according to its drop position.

Pre-requisites

- Github
- Android Studio (java and sdk environment installed correctly)

Set up

- Git clone from our repository (branch name unknown for now)
- In the Android Studio, select the build variant as focusX86Debug
- Run the project with an emulator

Acceptance Test Steps and Results

No.	Steps	Expected Results	Actual Results	Passed/Failed
1	Type “google.com” in the URL toggle bar and press enter	The expand button only will show up when the user is redirected to google.com	The expand button showed up after redirecting to google.com	Passed
2	Drag and drop the expand button to right side of the screen	The expand button should stick to the right when dragging to the right.	The expand button was pinned to the right after dragging it to the right of the screen	Passed
3	Click the expand button	The expand button should expand to the left and show the 3 sub buttons: back, refresh and delete in that order, from top to button	the expand button expanded to the left of the screen and showed the 3 sub buttons back, refresh and delete in that order, from top to button	Passed

4	Scrolls down when the menu is open	All the sub buttons and the expand button should disappear	All the sub buttons and the expand button disappear	Passed
5	Scrolls up when the expand button is not displayed	Only the expand button will show up	The expand button shows up without menu open	Passed
6	Click the expand again to close the submenu	The expand button should close automatically	The expand button closed automatically when clicked the second time	Passed
7	Drag and drop the button to the left side of the screen	The button should stick to the left side of the screen	The expand button was pinned to the left after dragging it to the left of the screen	Passed

8	Click the expand button	The expand button should expand to the right and show the 3 sub buttons: back, refresh and delete in that order	The expand button expanded to the right of the screen and showed the 3 sub buttons back, refresh and delete in that order	Passed
9	Click the refresh sub button	The page should be refreshed	The page was refreshed when the refresh sub button was clicked	Passed
10	Type “reddit.com” in the URL toggle bar and press enter	The page is redirecting to reddit.com	The page was redirecting to reddit.com	Passed
11	Click the back sub button	The page should redirecting to google.com	The page was redirecting to google.com	Passed

12	Click the delete sub button	The browsing history should be deleted and the page should be redirected to the default main page	The browsing history was deleted and the page was redirected to the default main page	Passed

Accpetance Citeria Verification

No	Acceptance Citeria	Accepted/ Not Accepted
1	Clicking the expand button should expand the sub menu-buttons back, delete, refresh in the right angle and correct order	Accepted
2	Each sub menu-button should work according to its corresponding function	Accepted
3	Clicking on the expand button again should close all the the sub menu-buttons	Accepted
4	The sub menu-buttons should close automatically when dragging the expand button.	Accepted
5	The sub menu -button should be draggable and stick to the right size of the screen according to its drop position.	Accepted

Unit Test

Our initial plan for unit testing was manual test, since our feature is largely involved in front-end interactions, and the complete manual test flow can be found in our process outcomes document, starting from page 13.

Later, we found out we could use Espresso to do the test, thus we also created a test file to test the expanded angle of the menu and the drag button feature. The file can be found at here.