Assignment & Worksheet 4

CS 501, Mobile Application Development

Fall 2022

Gestures, Teamwork, Fundamental Programming

Date: 10/02/2022

Team Members: Lesley Chen, Tiffany Chen, Zizhuang Guo, Nick(Sangjoon) Lee, Jinpeng Lyu, Alex Wang

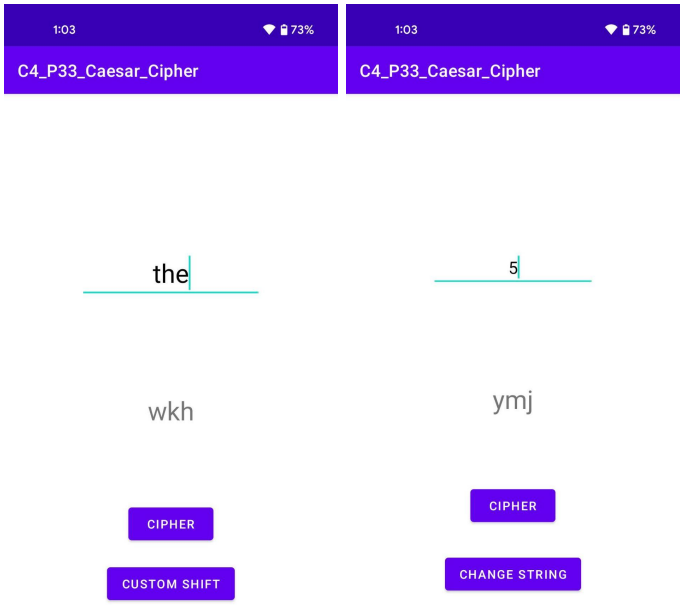
Assignment 4

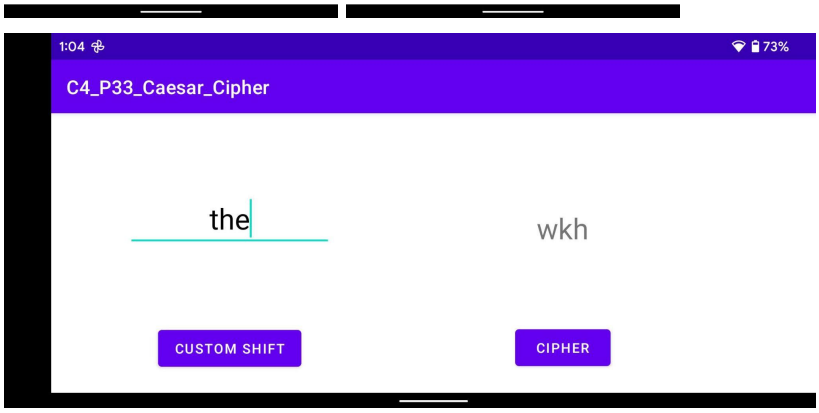
Ch4 Q1-10

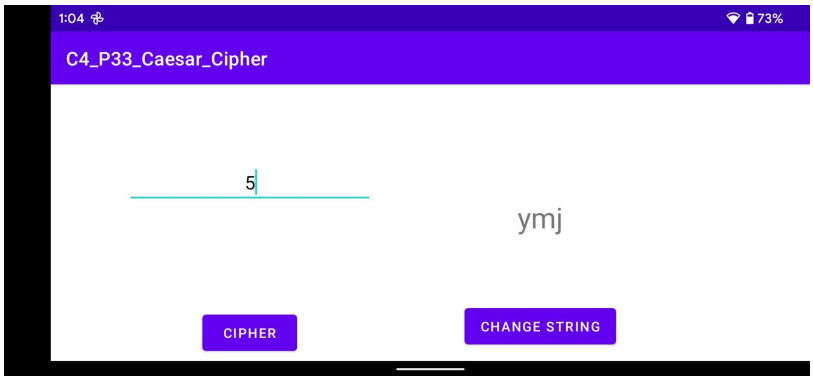
1. The TableLayout class can be used to organize various GUI components
   - Rows and columns
2. The direct superclass of LinearLayout and RelativeLayout is
   - View Group
3. TableLayout and TableRow are direct subclasses of
   - Linear Layout
4. The RelativeLayout class is a good choice to organize various GUI components
   - Because we can position components relative to other components
5. In what package is the Intent class?
   - android.content
6. After you have created an Intent for a new activity, what method of the activity class do you call with that Intent parameter in order to start a new activity?
   - startActivity
7. What method of the Activity class is automatically called when an activity is about to restart?
   - onRestart()
8. What methods of the Activity class (and in what order) are automatically called when an activity is first created?
   - onCreate, onStart, and onResume (in that order)
9. What method of the Activity class is automatically called when an activity becomes invisible to the user
   - onStop()
10. Two activities can share the same data:
    - Yes, for example by each accessing a public static instance variable from another class

Programming Task #33: C4_P33
Caesar Cipher, 2 activities

https://github.com/AlxWang9966/C4_P33







Worksheet 4

Part 0: Flashlight

https://github.com/AlAuB/W4_P0

Design Process

1. What design questions do you need to ask before beginning?

   What views we need to make (TextView, Button, etc)

   - SwitchCompat
   - ImageView
   - EditText

   What kind of layout we want

   - Constraint Layout and Linear Layout for Image and Switch

2. What are the functions that need to be written?

   - **setOnCheckedChangeListener()** for SwitchCompact
   - **setOnKeyListener()** for EditText
   - **onTouchEvent(), onFling()** for GestureDetector.OnGestureListener
   - **checkSelfPermission()** to get permission for flashlight
   - **getSystemService()** to check if flashlight is available or not.
   - **CameraManager** and **onTorchModeChanged(id, enabled)** to turn on or off the flashlight (This API is only available for Android 6.0 and above).
   - **Camera.open()**, **Camera.parameter**, **setFlashMode(FLASH_MODE_ON), SurfaceTexture**, and **Camera.startPreview()** to turn on or off the flashlight (This is only for Android 5.1 and these are deprecated APIs).

3. What exceptions do you need to handle and how should you handle them?

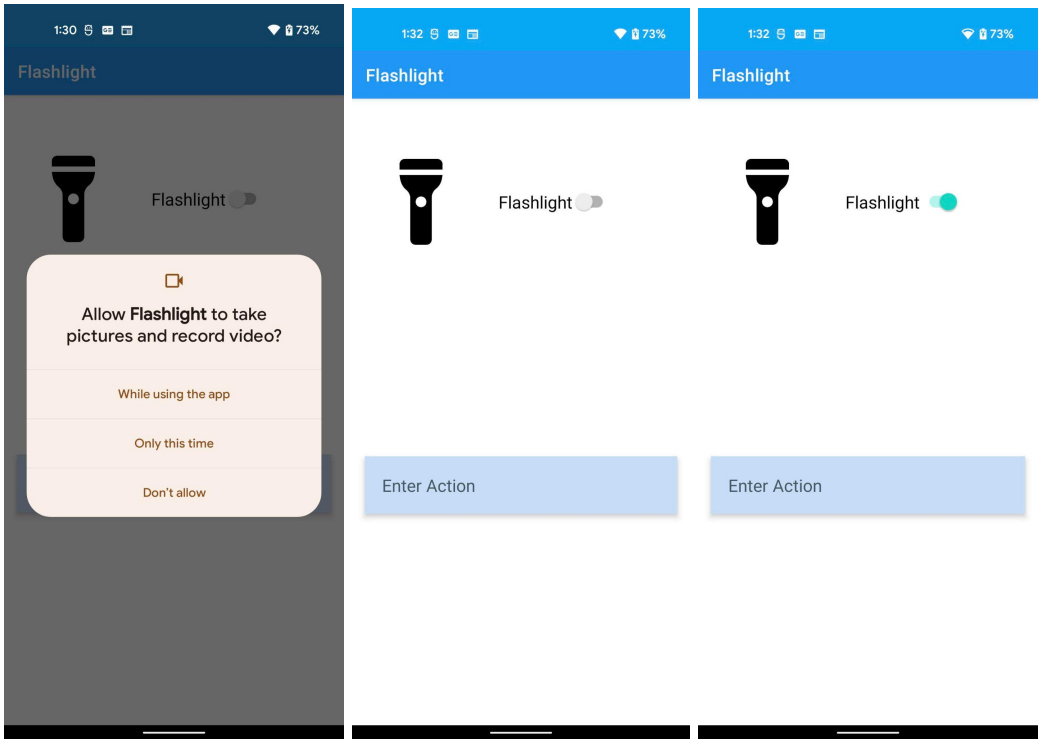
   Flashlight not available

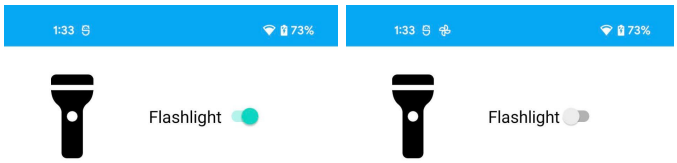
   Solution: A Toast message pops up and shows “This device does NOT have a flashlight”.

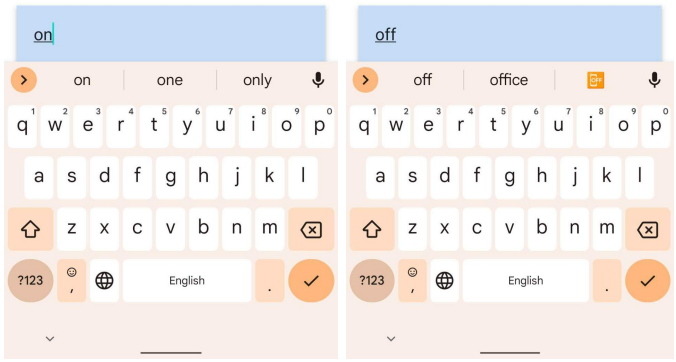
4. What types of feedback do you need to provide the user as they interact with your App.

   - Toggle on/off corresponds to flashlight on/off.
   - Flashlight not available → a Toast message shows up.
   - Fling up or down to turn on or off the flashlight.
   - Type “ON” or “OFF” in the text field also lets the user control the flashlight.

Screenshot:


1:30
73%
Flashlight
Flashlight
Allow Flashlight to take pictures and record video?
While using the app
Only this time
Don't allow
1:32
73%
Flashlight
Flashlight
Enter Action
1:32
73%
Flashlight
Flashlight
Enter Action


1:33
73%
Flashlight
1:33
73%
Flashlight


on
on
one
only
off
office
English
?123
English
?123

Part 1: Initial Project Ideas

Come up with 3 Ideas for projects; be as descriptive as possible

Describe Third party libraries you'd like to work with

Identify pinch points and what you would need to learn to implement your project ideas

a) Compare pricing of food delivery apps
   i) **Intro:** An useful app that helps you to compare delivery prices between various delivery apps for a certain restaurant of your choice.
   ii) **Scenario/How to Use:** Say you want to order food from *Raising Cane's Chicken Fingers*, and the source shows that UberEats, Grubhub and DoorDash all offer delivery options for it. Our app will pull from the UberEats' app, Grubhub's app and DoorDash's app to see if there is a price difference between all three choices based on the same combination of foods you ordered. So it makes it easier for you to compare the delivery prices in order to make the best deal.
   iii) **Pinch Points:** We need to apply various API from different apps. We need to receive permission from the food delivery apps to send data to this app. What's more, we will have to take into consideration deals/discounts presented by different apps. What if a certain amount of food will get better discounts in UberEats than Grubhub.
   iv) Revenue Generation: Ads on the app; premium users will not have to view ads

b) Comprehensive Scheduler
   i) A **very powerful scheduler App** that allows various features like:
      1) Record busy/free hours. [Auto In-advance Alarm: 5min, 30min, etc]
      2) Pass in information for the scheduled time slots, eg: Zoom links, location on maps, memos, etc.
      3) Friends Sharing!
         (a) Add your friends to schedule so both of you could have that time planned on both schedules.
         (b) Compare with your friends' schedule to better arrange meetings, etc.
      4) Set priorities, types, tags to arrange better solutions:
         (a) We developed a concept of 'Tag Schedules', meaning you will have a separate schedule that only displays the activities marked under a certain tag.
         (b) You will have a main schedule while you can choose to turn On/Off the different 'Tag Schedules' with toggles, so they would be shown with your main activities.
   ii) **Pinch Points:** We need to be able to merge different schedules between friends. We are also dynamically creating different activities when the user is creating a new 'Tag Schedule". Adding information to every event could be challenging. We will need to learn the backend.

iii) Revenue Generation: premium features: save schedule when changing phones, more tags, different themes, direct import from Google Calendar, and get date and link from email.

c) Wishlist:

i) **Intro:** still wondering what to choose for your friend's birthday? Don't! No need to question and spy. Just check his/her wishlist!

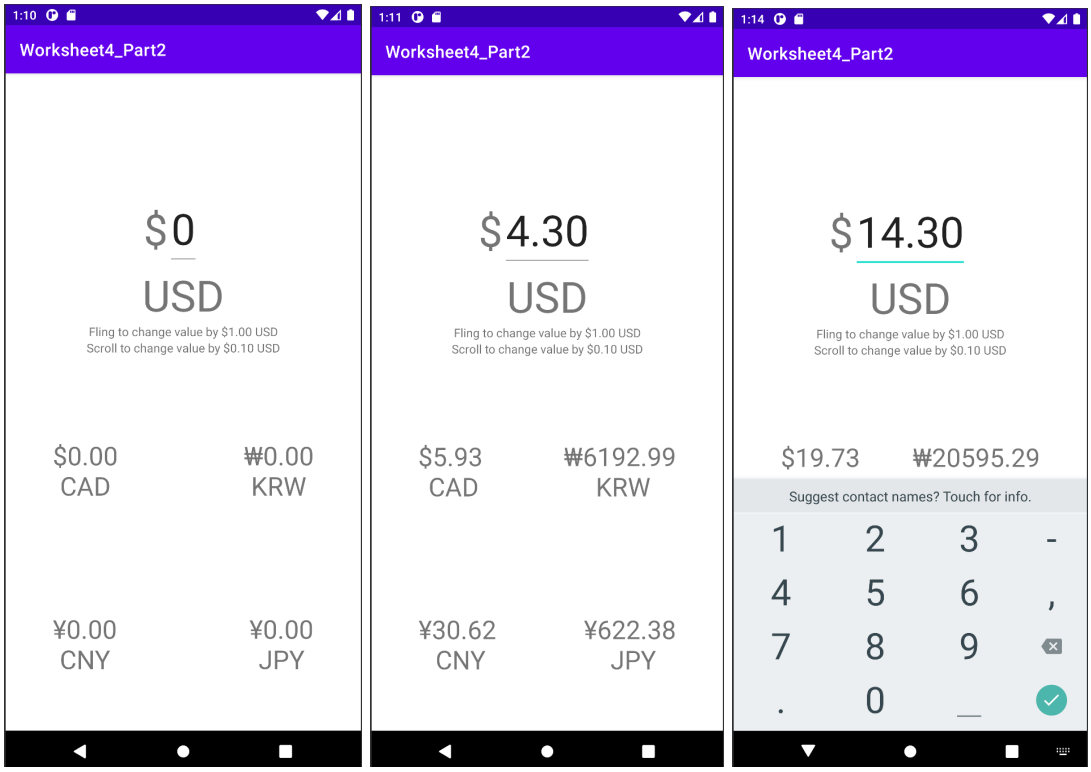
ii) **Features:**

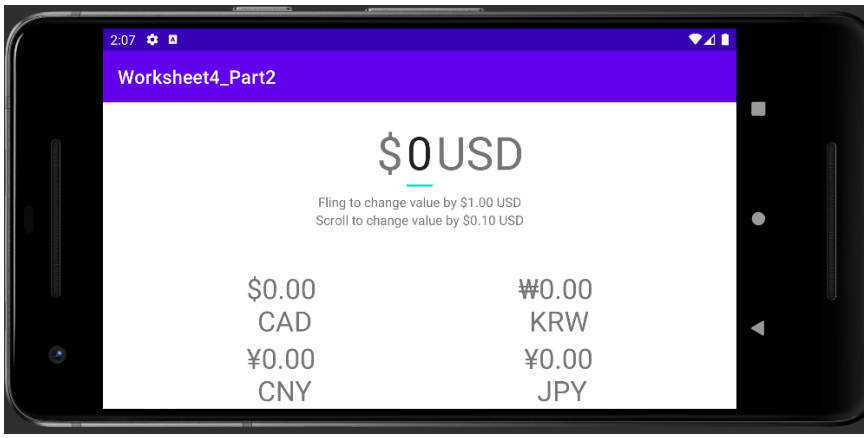
1) Add/Edit your list using simple descriptions, pics, etc. You can add a link if you want, so your friends know where to go!
2) Check your friend's list by asking them about the password. [This is a security issue. Assume you have healthy friend relationships and no one is going to mess up with your wishes]
3) Normal wishes are updated and will notify the host.
    (a) Revenue Generation: If User subscribe, you can see which Users have selected which gifts on your friend's wishlist
4) You will have a copy of the wishlist after entering the obtained password. You can check off the item you're giving as a gift. The updated version will be distributed to all the other friends who have the password. But the original list is maintained by the host only and will not be changing unless the host's actions delete/edit the list.

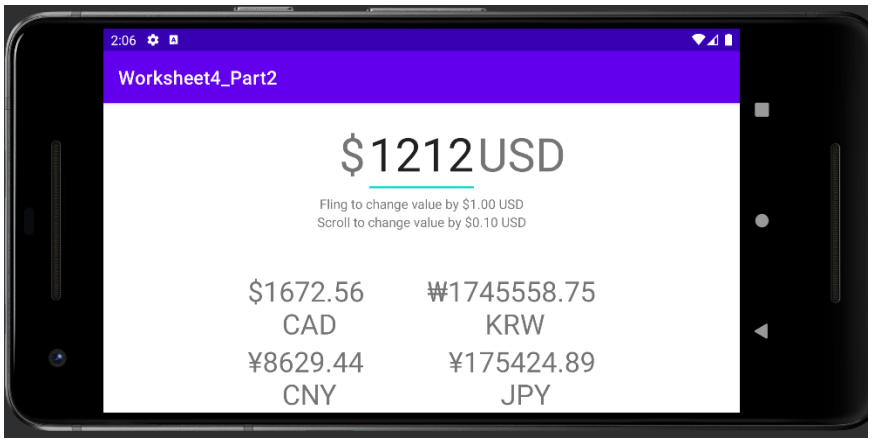
iii) **Pinch Points:** As the previous one, synchronicity between friends can be hard to apply. Security issues need more consideration.

Part 2: Flinging Money

https://github.com/sj0726/CS501_Worksheet4_Part2

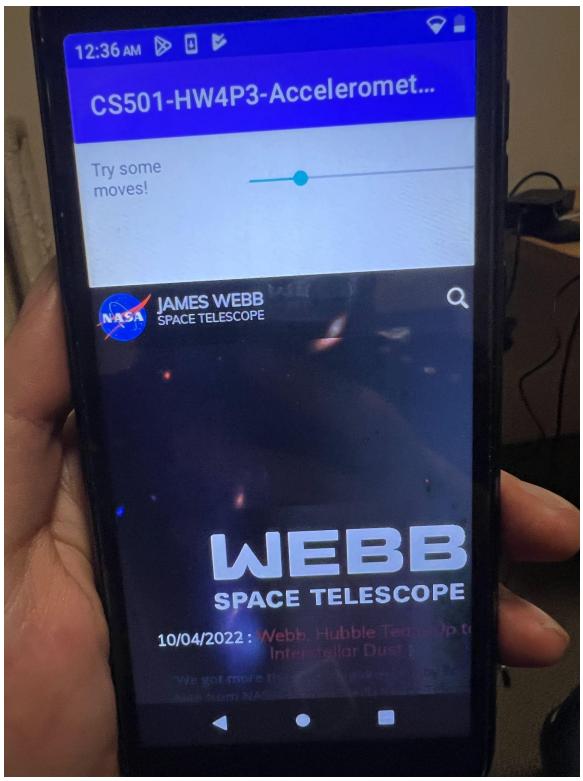


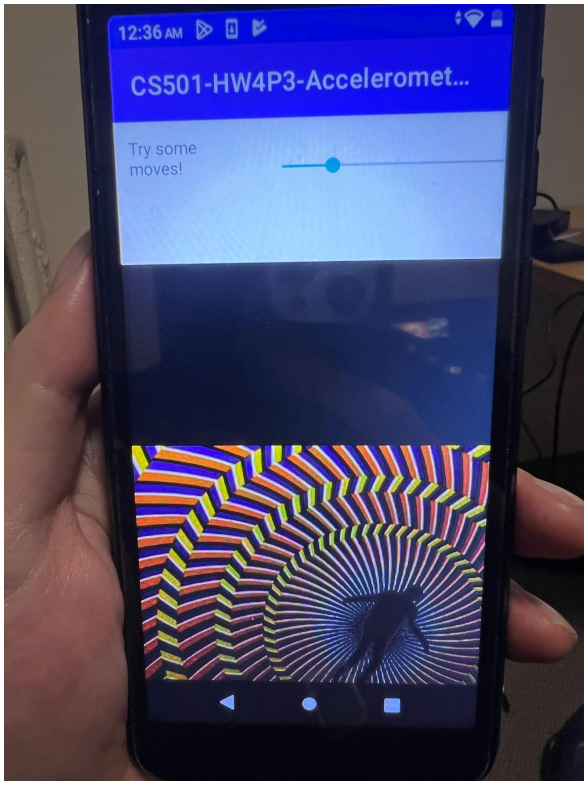

2:07
Worksheet4_Part2
$0USD
Fling to change value by $1.00 USD
Scroll to change value by $0.10 USD
$0.00
CAD
₩0.00
KRW
¥0.00
CNY
¥0.00
JPY

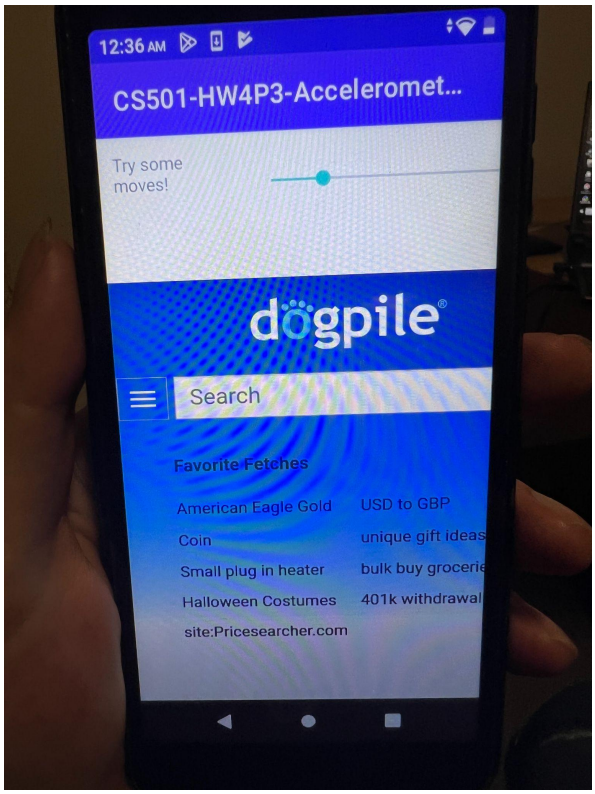

2:06
Worksheet4_Part2
$1212USD
Fling to change value by $1.00 USD
Scroll to change value by $0.10 USD
$1672.56
CAD
₩1745558.75
KRW
¥8629.44
CNY
¥175424.89
JPY

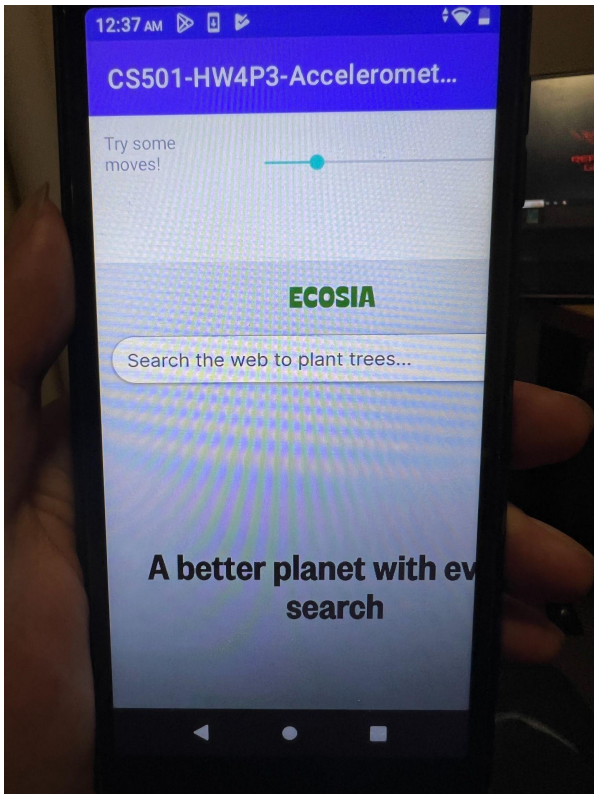
Part 3: Accelerometer Measurements

https://github.com/Guo-Zizhuang/CS501-HW4P3-AccelerometerMeasurement-Git



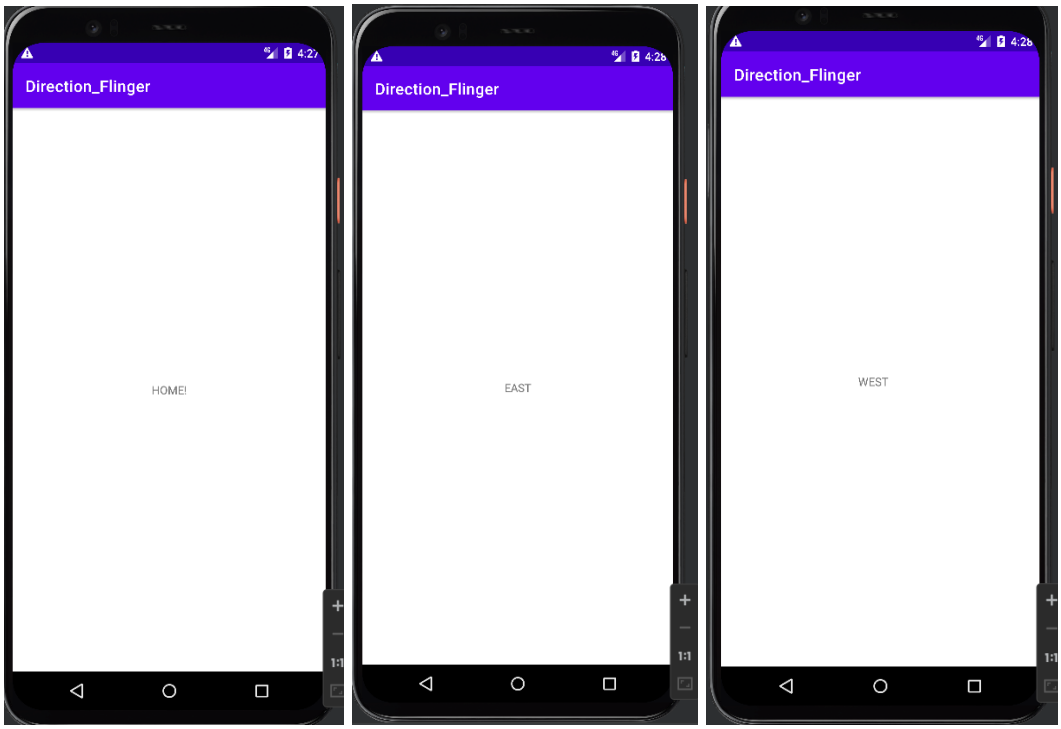


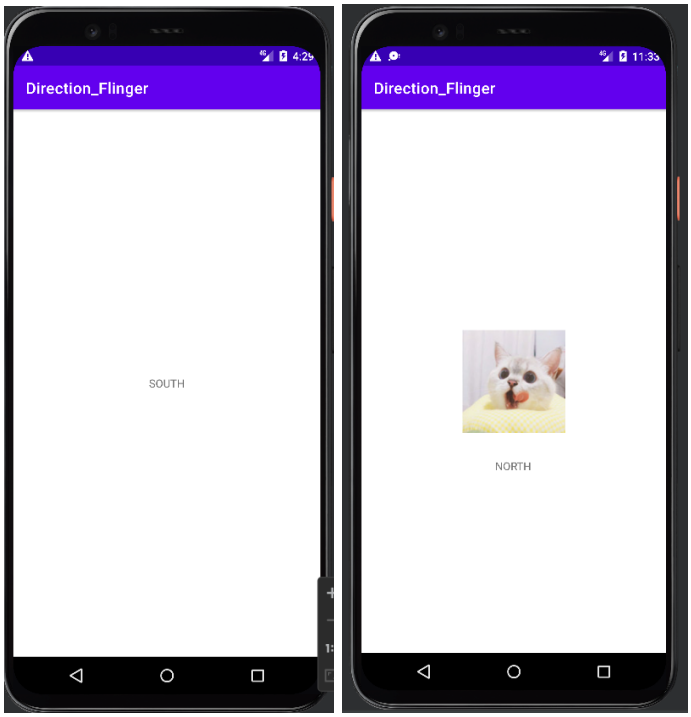


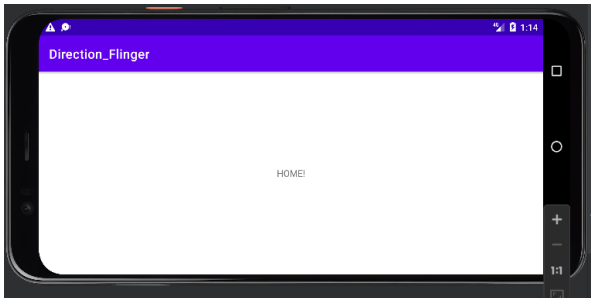


Part 4: Pulling it Altogether (North, South, East, West, Home)

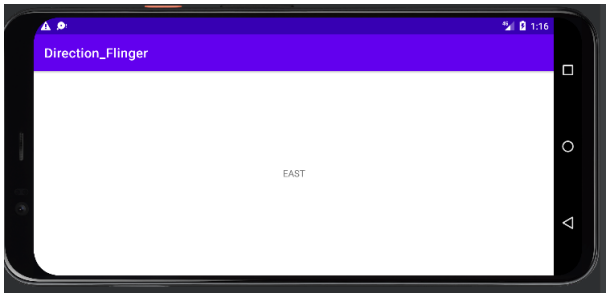
https://github.com/lchen456/W4_P4



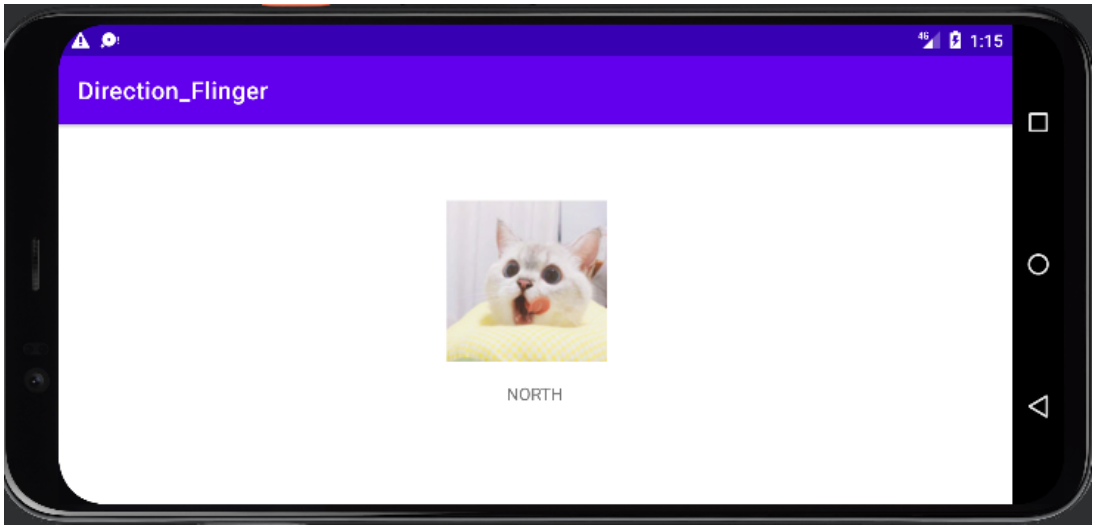



1:14
Direction_Flinger
HOME!


1:18
Direction_Flinger
SOUTH

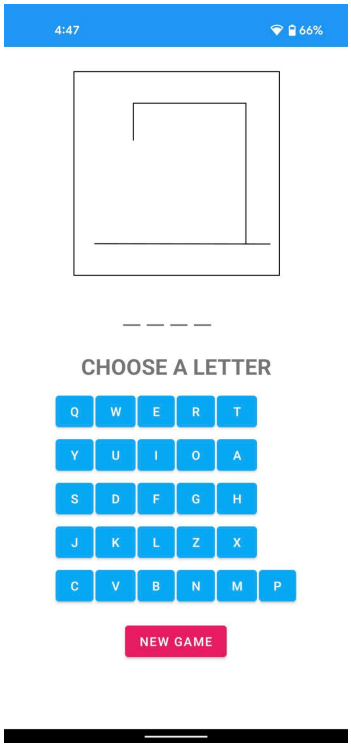

1:16
Direction_Flinger
EAST

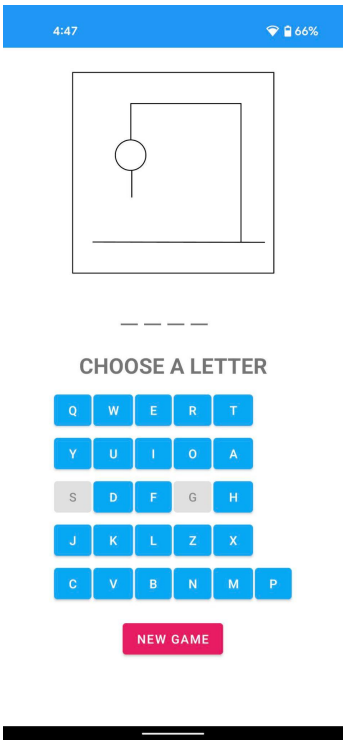

1:17
Direction_Flinger
WEST

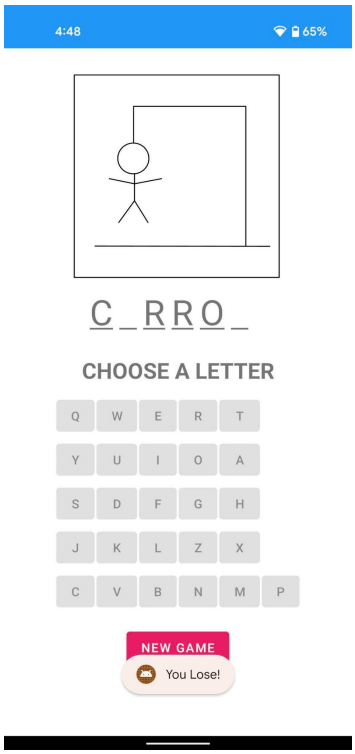

1:15
Direction_Flinger
NORTH

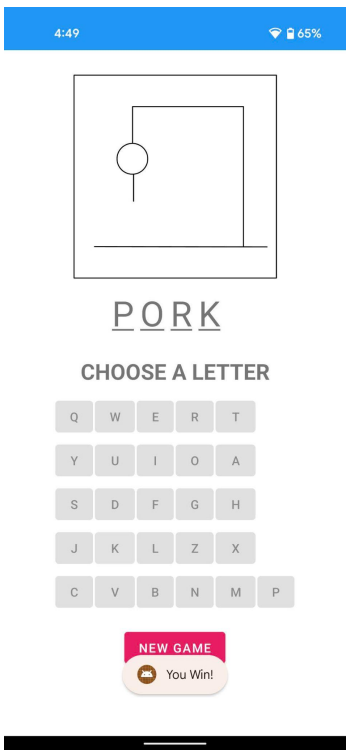
Part 5: Hangman

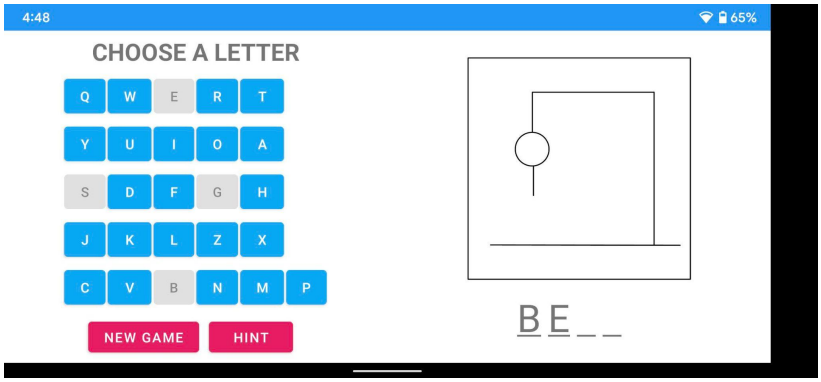
https://github.com/AlAuB/W4_P5

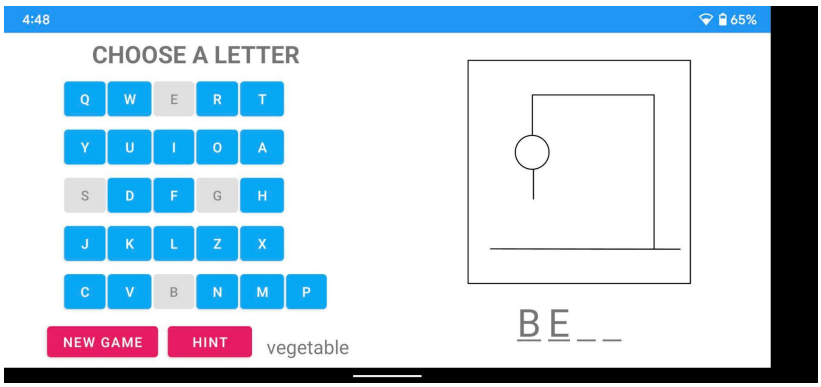




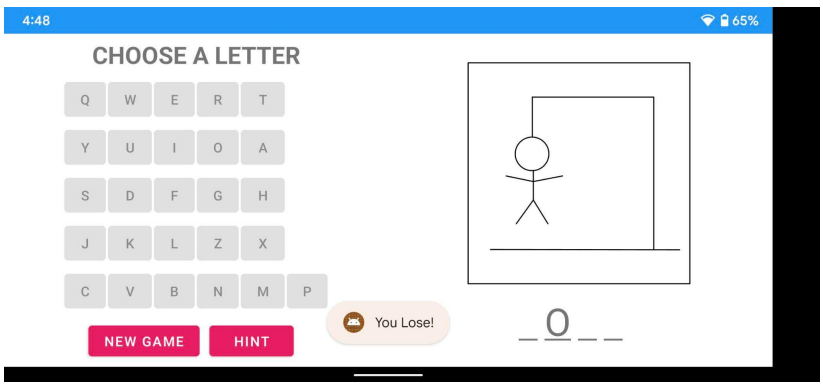


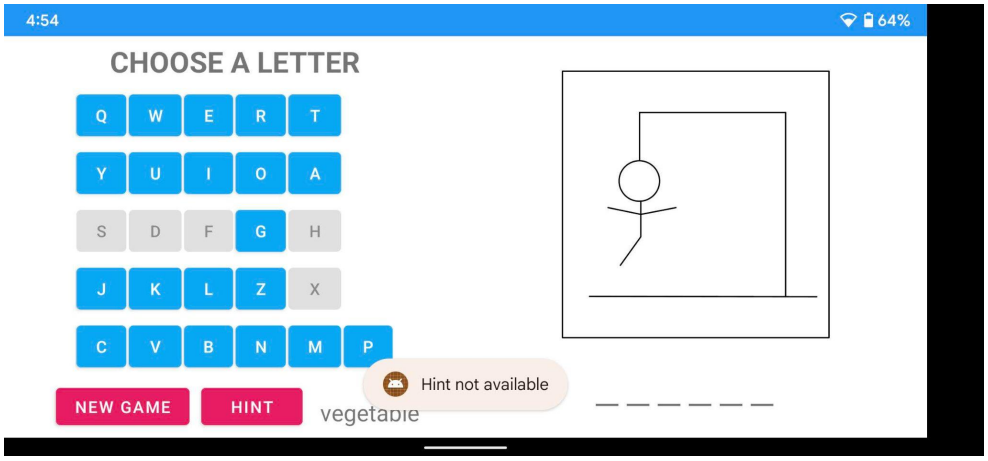


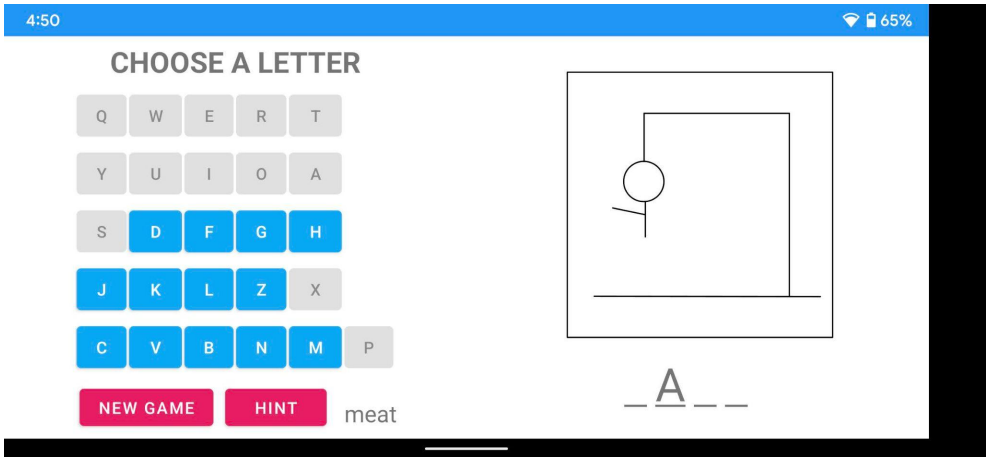

4:48
65%
CHOOSE A LETTER
Q W E R T
Y U I O A
S D F G H
J K L Z X
C V B N M P
NEW GAME
HINT
B E _ _

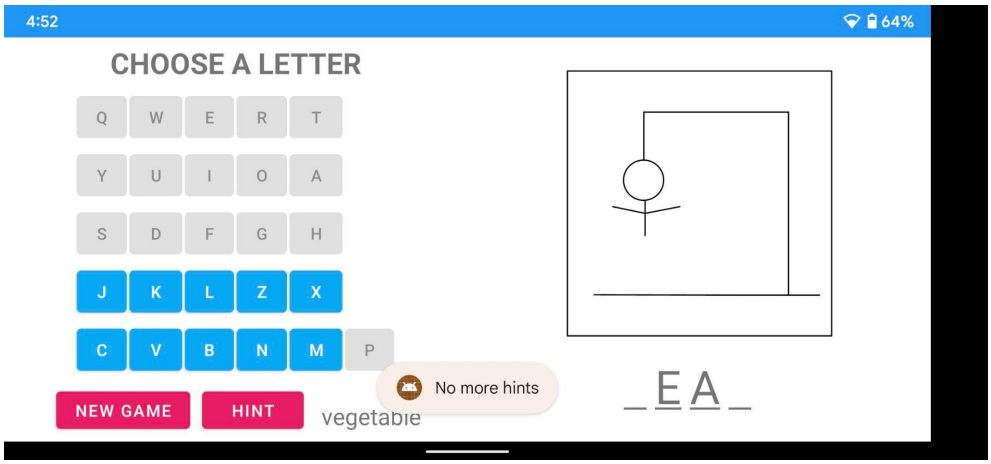

4:48
65%
CHOOSE A LETTER
Q W E R T
Y U I O A
S D F G H
J K L Z X
C V B N M P
NEW GAME
HINT
vegetable
B E _ _


4:48
65%
CHOOSE A LETTER
Q W E R T
Y U I O A
S D F G H
J K L Z X
C V B N M P
NEW GAME
HINT
vegetable
You Win!
B E A N


4:48
65%
CHOOSE A LETTER
Q W E R T
Y U I O A
S D F G H
J K L Z X
C V B N M P
NEW GAME
HINT
You Lose!
_ O _ _


4:54
64%
CHOOSE A LETTER
Q W E R T
Y U I O A
S D F G H
J K L Z X
C V B N M P
Hint not available
NEW GAME
HINT
vegetable
_ _ _ _ _ _


4:50
65%
CHOOSE A LETTER
Q W E R T
Y U I O A
S D F G H
J K L Z X
C V B N M P
NEW GAME
HINT
meat
_ A _ _


4:52
64%
CHOOSE A LETTER
Q W E R T
Y U I O A
S D F G H
J K L Z X
C V B N M P
No more hints
NEW GAME
HINT
vegetable
_ E A _