Basic calculation logic

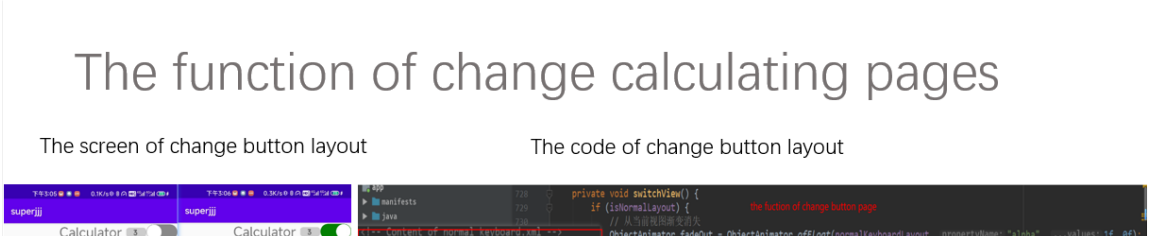
Result

- The basic operation interface and complex operation interface are switched, and the calculation result and expression are universal. Because i use the switch button to change the button layout, both layout are in the main_activity.xml file.
- Completed the basic operation +-×÷., % and negative number switching. Like 1+2 = 3, 7.2×2 = 14.4, 2-(-3) = 5 and so on
- Continuous calculations can be performed. Like 1+2+3×(-4) = -9
- Completed the special operation lg, ln, cos, sin, tan, √, (), reciprocal count and factorial. Like lg(10000) = 4, sin(30) = 0.5, tan(45) = 1, √9 = 3, 5! = 120
- Precise choices can be made to preserve a certain number of decimal places. If choose 3, 1%3 will get 0.333, if choose 7, 1%3 will get 0.3333333
- Clear button can clear expression and last result, back button can delete the last char of expression
- Swipe to the right from the screen to get the history page, which records the results of the calculator run

Code technique

1. **Design Layout Using DrawerLayout:** I used DrawerLayout as the parent container for my calculator interface. I included two different layouts for calculation buttons using the `<include>` tag. By clicking the button in the top right corner, I implemented the ability to switch between basic and advanced calculator layouts. Both layouts are contained within the same parent interface, so the results and expressions are shared between them.
2. **History Feature Implementation:** Inside the parent container, I set up a LinearLayout for the history interface, placed in the start direction of DrawerLayout. This allows swiping from the left to reveal the history page. Additionally, I implemented a gesture class in the Java main file, enabling the display of the history page. I stored the history records in a list, automatically saving the result into the list upon clicking the equal button. The related adapter is updated, ensuring the history interface always displays the most recent data.
3. **Calculation Logic Using Two Classes:** I employed two calculation classes to handle data computation. The first class handles basic operations such as addition, subtraction, multiplication, division, parentheses, and negative numbers. In the advanced calculation class, I iterate through the expression and prioritize special operators. I calculate the results of these sub-expressions, replacing the original sub-expression with the obtained result. Once all sub-expressions are resolved and replaced, the entire expression becomes a standard one, allowing for straightforward calculation to obtain the final result.
4. **Precision Handling with Sniper Class:** I used the Sniper class to store eight different precision options for users to choose from. Different precisions result in varying decimal places. The default precision is set to 3. Precision is passed as a parameter to the calculation function, allowing us to obtain the corresponding result based on the selected precision.
5. **Utilizing Toast, Precision, and Switch Utility Classes:** I created utility classes such as Toast (for displaying brief messages), Precision (for handling precision settings), and Switch (for designing the switch button) to simplify the code. These utility classes enhance the readability and maintainability of the codebase.

PPT pics

- 

- # The function of calculate result in normal page

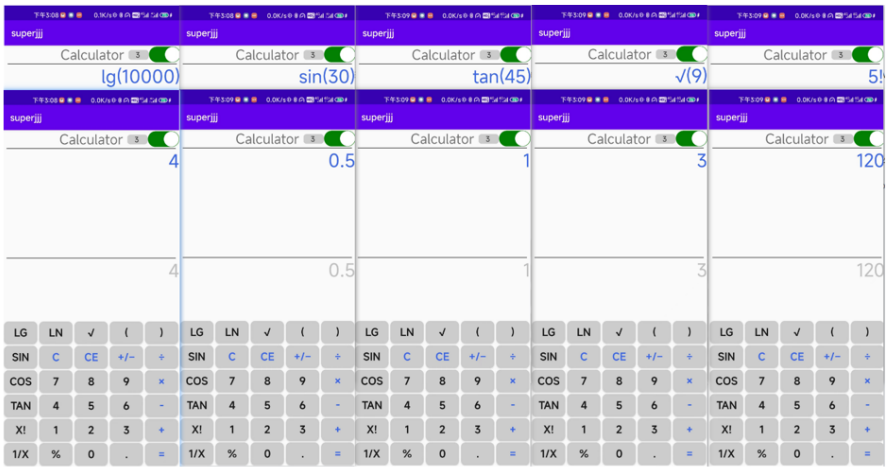
  The result of add, dot, negative number

  The result of continuous expression

- # The function of calculate result in special page

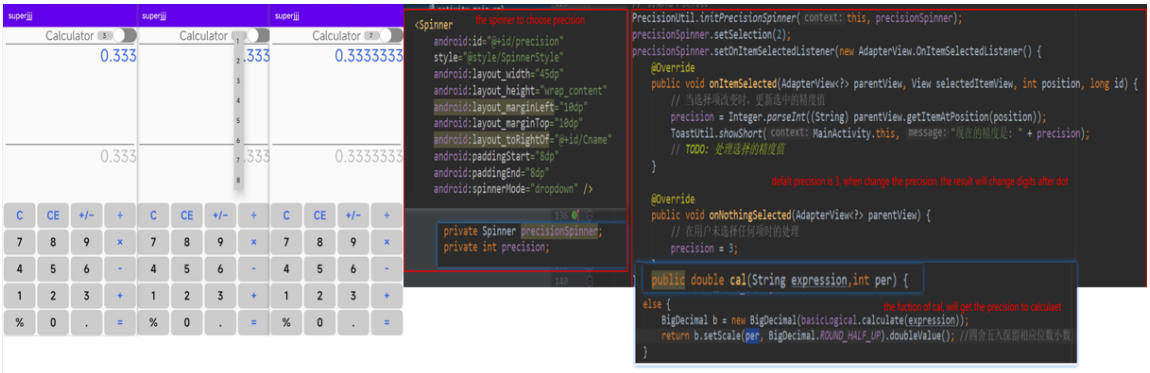
  The result of lg, sin, tan, sqrt, factorial

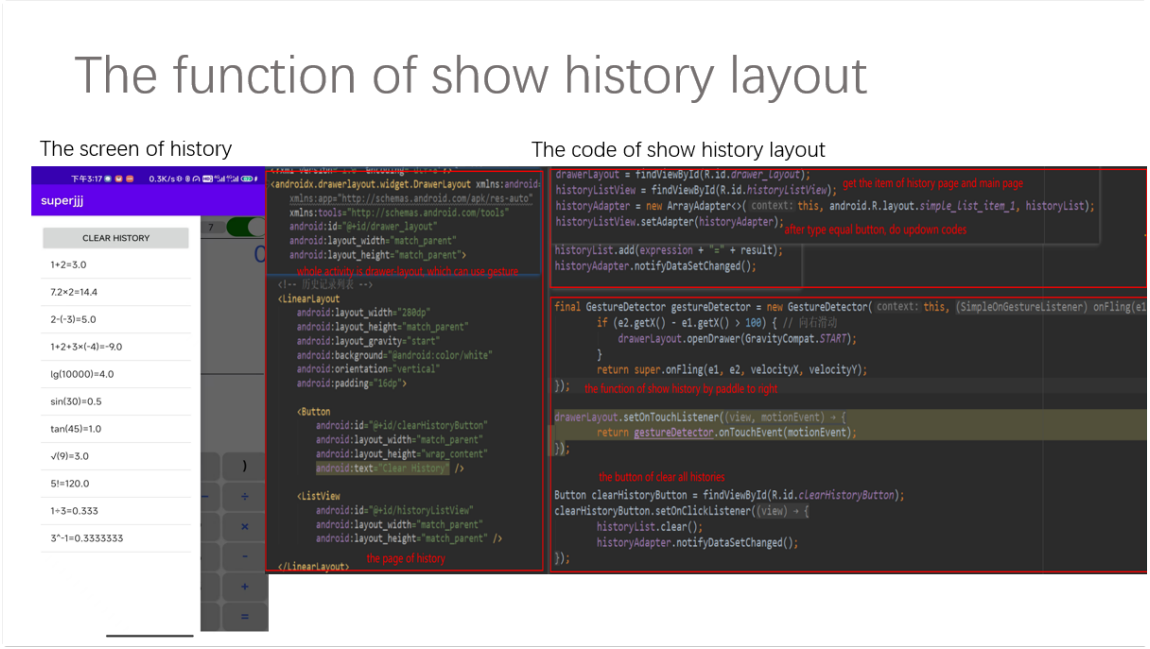
  

- # The function of change precision

  The screen of change precision

  The code of change button layout

- 

Transition logic

Results

- Apply the `fragments` to change 3 pages, the `first fragment is for calculator`, the `second fragment is for conversion functions and two small games`, the third fragment is for matrix (final implementation)
- Set logic of calculator in fragment, change main_activity to control Page-turning logic
- Conversion page contains 12 activities to apply:
  1. `length conversion`
     - set meter(m) as the basic operation unit
     - user can click other length and type in number to calculate for all length
     - when choose one length, the background will be white and set the blue border
     - the page can roll, but meter(m) is fixed on the top
     - Keep four decimal
     - just type **"="** can get the result, and if typing again after **"="**, the previous result will be cleaned
     - the reason of why after typing **"="** can show the result is result contains decimal number, it is difficult to add new number directly
     - if the expression is empty and type **"="**, the application will Toast a sentence
  2. `area conversion`
     - set meter²(m²) as the basic operation unit
     - other like length page
  3. `volume conversion`
     - set meter³(m³) as the basic operation unit
     - other like length page
  4. `weight conversion`
     - set kilogram(kg) as the basic operation unit
     - other like length page
  5. `conversion of number systems`
     - set binary(BIN) as the basic operation unit

- divide buttons into 4 arrays and get the layout by id of 4 kinds of number system, the corresponding buttons are allowed only when the corresponding number system is clicked
- Entering a value allows other number systems to get the corresponding value, because each button is bound to the method of calculating the result

6. `temperature conversion`
   - set degree centigrade(°C) as the basic operation unit
   - most aspects are similar to length pages
   - it allows typing dot and the dot just can type one time
   - it allows typing negative **"-"** and just can at the start of expression, if at other positions the application will Toast a sentence
7. `BMI calculating`
   - user should type height(cm) and weight(kg) and the region of them both is (0.01-300), if the value is wrong or has no value, the application will Toast a sentence
   - the result of BMI will show in the card layout
   - the BMI region i set between 14.0-34.0, the progress bar will show the BMI condition
   - the application will show the suggestion weight of your height
8. `number change to English`
   - most aspects are similar to length pages
   - it allows typing dot and the dot just can type one time
   - Entering a value allows to get the corresponding value, because each button is bound to the method of calculating the result
   - the most number is billion and the system will Toast a sentence
9. `calculate the gap between two time`
   - set two button to get the time dialog
   - it will show all minutes, hours, days, weeks, months, years between two time
   - use timestamp to get the result
10. `calculate the date after xx days or xx hours or xx minutes`
    - set one button to get the time dialog
    - user can type days or hours or minutes to get the deadline
11. `tictactoe game`
    - there are 3 kinds of result, O wins, X wins and draw
    - O is first to move, then is X, the cell that holds the content cannot be overwritten
    - when one's operation is the line, he will win, and the pic cannot move, the user can type button "play again" to restart the game
    - if no one win, the pic will show Draw, the user can type button "play again" to restart the game
12. `2048 game`
    - the game will show temporary score and history highest score
    - different number cells have different color
    - user can undo one step if the game panel structure is changed by typing **UNDO button**
    - the undo result changes with each swipe
    - user can restart game by typing **RESTART button**
    - the newly generated cell will realize the animation from small to large by scaleX and scaleY
    - the complex generated cell will realize the animation from large to small by scaleX and scaleY
    - the highest result is stored by SharedPreferences, if the temporary score is higher than history, the highest will changed

- if the game panel has no empty cells and each cell is not adjacent to a cell of the same value, the game is over and it will show dialog, the user can type EXIT to the conver page or type RESTART to play again

> the reason of set these two game
>
> - i have done 10 conversions but i have 2 cells have no function, and when i watch bilibili i notice one blogger 安卓knowledge creates a beautiful tictactoe page, but he did it by kotlin, so i just learn the page construction from him and do other things by canvas.
> - 2048's code is open source so i just learn the logic of one bilibili blogger named 极客学院, then i create the basic page of 2048, but the undo, score, style and animation design is by myself

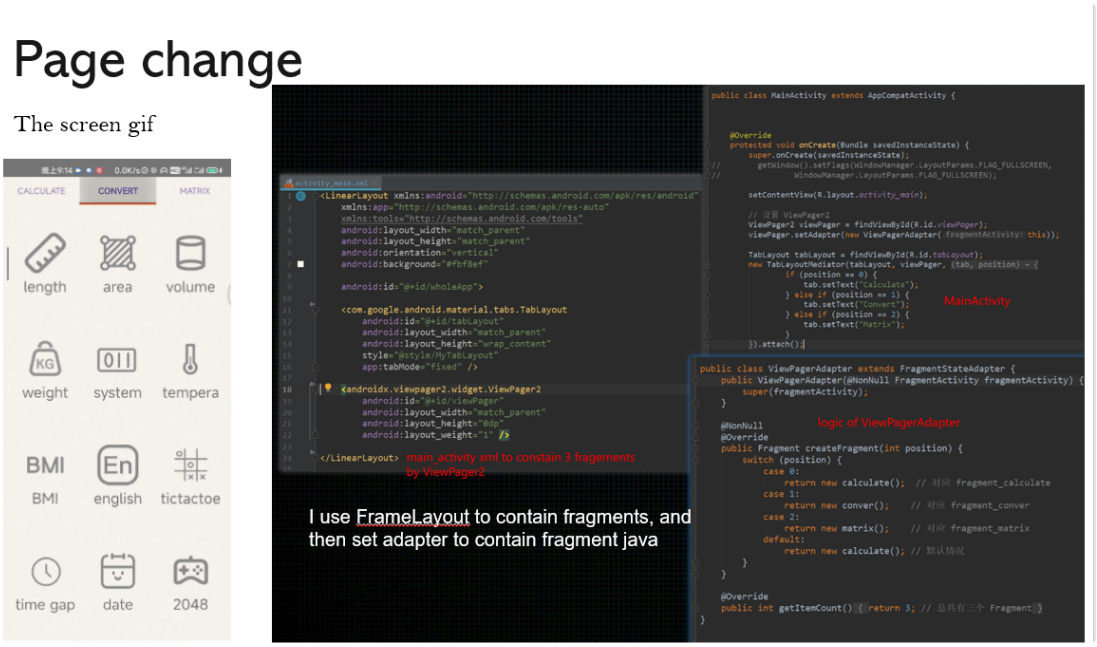
Code technique

- use ViewPager2 and TabLayout to control the switch fragments logic
- use lots of Toast to tell the user what should he does and what shouldn't do
- add lots of click event and click styles to highlight the choice after user choose
- use self-restrained dialog to show time picker
- use different conversion algorithm to get the length, weight, area, volume and other conversion objects
- set whether the button is available to converse number system
- create process bar to show the result of BMI; write number to English algorithm to get English of number
- design appropriate time calculating algorithm to get the gap between two times
- use canvas to write the tictactoe game, and use MVC structure to apply the logic
- use self-restrained layout to show 2048 game, and use SharedPreference to save data and use animate to make the page beautiful

Pics

> all of things are from PPT, PPT is the gif pic

page change

- 

Logic of fragment calculate

I set onCreateView to init all components

```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {
    view = inflater.inflate(R.layout.fragment_calculate, container,  attachTo

    scrollView = view.findViewById(R.id.scrollView);
    input = view.findViewById(R.id.input_view);
    output = view.findViewById(R.id.output_view);

    input.addTextChangedListener(new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence charSequence, int start, i
        }

        @Override
        public void onTextChanged(CharSequence charSequence, int start, int b
            scrollView.post(() -> {
                    scrollView.scrollTo( x: 0, input.getBottom());
            });
        }

    buttons[2] = button3;
    buttons[3] = button4;
    buttons[4] = button5;
    buttons[5] = button6;
    buttons[6] = button7;
    buttons[7] = button8;
    buttons[8] = button9;

    button0s = view.findViewById(R.id.number0s);
    button1s = view.findViewById(R.id.number1s);
    button2s = view.findViewById(R.id.number2s);
    button3s = view.findViewById(R.id.number3s);
    button4s = view.findViewById(R.id.number4s);
    button5s = view.findViewById(R.id.number5s);
    button6s = view.findViewById(R.id.number6s);
```

I set onViewCreated to init all functions

```
@Override
public void onViewCreated(View view, Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);

    // 这里你可以初始化视图和设置监听器
    // 例如:
    drawerLayout = view.findViewById(R.id.drawer_layout);
    historyListView = view.findViewById(R.id.historyListView);
    historyAdapter = new ArrayAdapter<>(getActivity(), android.R.layout.simple_
    historyListView.setAdapter(historyAdapter);

    // 其他的初始化代码...

    // 设置手势监听
    final GestureDetector gestureDetector = new GestureDetector(getActivity(),
            // 双击事件的处理
            drawerLayout.openDrawer(GravityCompat.START); // 打开Drawer
            return true;
    });

    // 设置触摸监听
    drawerLayout.setOnTouchListener((v, event) -> {
            // 处理触摸事件
            if (event.getAction() == MotionEvent.ACTION_UP) {
                // 当手指抬起时模拟点击
                v.performClick();
            }
            return gestureDetector.onTouchEvent(event);
    });

        initInput();
        initNumBtns();
        initBaseOpers();
        initBaseCalculatorFunction();
        initSuperFunction();
    }
```

Logic of fragment conversion

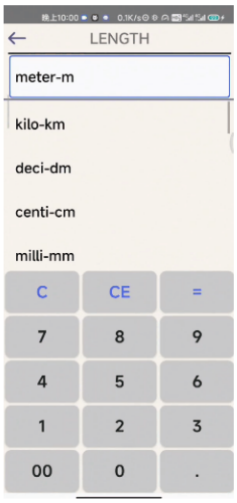
```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.fragment_conver, container,  attachToRoot: false);

    // 初始化视图
    l1 = view.findViewById(R.id.lengthView);
    l2 = view.findViewById(R.id.areaView);
    l3 = view.findViewById(R.id.volumeView);
    l4 = view.findViewById(R.id.weightView);
    l5 = view.findViewById(R.id.systemView);
    l6 = view.findViewById(R.id.temperatView);
    l7 = view.findViewById(R.id.BMIView);
    l8 = view.findViewById(R.id.englishView);
    l9 = view.findViewById(R.id.tictactoeView);
    l10 = view.findViewById(R.id.gapView);
    l11 = view.findViewById(R.id.timeView);
    l12 = view.findViewById(R.id.gameView);

    // 设置点击事件
    l1.setOnClickListener((v) -> {
            // 启动新的Activity
            Intent intent = new Intent(getActivity(), LengthActivity.class);
            startActivity(intent,
                    ActivityOptions.makeSceneTransitionAnimation(getActivity()).toBundle());
    });

    l2.setOnClickListener((v) -> {
            // 启动新的Activity
            Intent intent = new Intent(getActivity(), AreaActivity.class);
            startActivity(intent,
    l3.setOnClickListener((v) -> {
            // 启动新的Activity
            Intent intent = new Intent(getActivity(), VolumeActivity.class);
            startActivity(intent,
                    ActivityOptions.makeSceneTransitionAnimation(getActivity()).toBundle());
    });

    l4.setOnClickListener((v) -> {
            // 启动新的Activity
            Intent intent = new Intent(getActivity(), WeightActivity.class);
            startActivity(intent,
                    ActivityOptions.makeSceneTransitionAnimation(getActivity()).toBundle());
    });

    l5.setOnClickListener((v) -> {
            // 启动新的Activity
            Intent intent = new Intent(getActivity(), SystemActivity.class);
            startActivity(intent,
                    ActivityOptions.makeSceneTransitionAnimation(getActivity()).toBundle());
```

This just provide the change activity function and set a little creating animation

length

Length

The screen gif

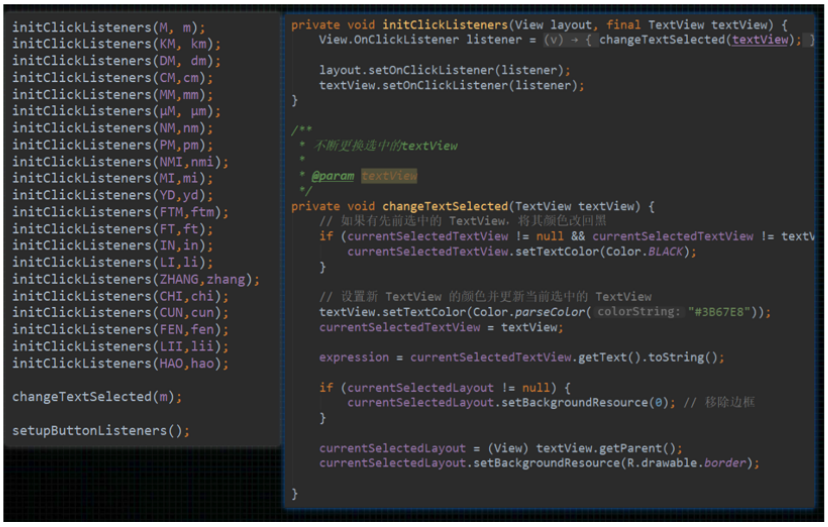


Firstly I get all texts and buttons

```
private View header;
private View converButtons;
private View back;
private TextView text;

private View M;
private View KM;
private View DM;
private View CM;
private View MM;
private View μM;
private View NM;
private View PM;
private View NMI;
private View MI;
private View YD;
private View FTM;
private View FT;
private View IN;
private View LI;
private View ZHANG;
private View CHI;
private View CUN;
private View FEN;
private View LII;
private View HAO;

private TextView m;
private TextView km;
private TextView dm;
private TextView cm;
private TextView mm;
private TextView μm;
private TextView nm;
private TextView pm;
private TextView nmi;
private TextView mi;
private TextView yd;
private TextView ftm;
private TextView ft;
private TextView in;
private TextView li;
private TextView zhang;
private TextView chi;
private TextView cun;
private TextView fen;
private TextView lii;
private TextView hao;

private TextView currentSelectedTextView = null;
private View currentSelectedLayout = null;

private String expression;
private boolean shouldResetInput = false;

private Button[] buttons = new Button[12];
private Button[] function = new Button[3];

Button button0, button00, button1;
Button button2, button3, button4, button5, button6, button7;
Button button8, button9;
Button buttonDot;
Button clearAll, clear, equal;

dm = findViewById(R.id.dmText);

CM = findViewById(R.id.cmView);
cm = findViewById(R.id.cmText);

MM = findViewById(R.id.mmView);
mm = findViewById(R.id.mmText);

μM = findViewById(R.id.μmView);
μm = findViewById(R.id.μmText);

NM = findViewById(R.id.nmView);
nm = findViewById(R.id.nmText);

PM = findViewById(R.id.pmView);
pm = findViewById(R.id.pmText);

NMI = findViewById(R.id.nmiView);
nmi = findViewById(R.id.nmiText);

MI = findViewById(R.id.miView);
mi = findViewById(R.id.miText);

YD = findViewById(R.id.ydView);
yd = findViewById(R.id.ydText);

FTM = findViewById(R.id.ftmView);
ftm = findViewById(R.id.ftmText);

FT = findViewById(R.id.ftView);
ft = findViewById(R.id.ftText);

IN = findViewById(R.id.inView);
in = findViewById(R.id.inText);
```

- # Length

Then I set the click function for all layouts and views, the currentSelectedView will calculate for others

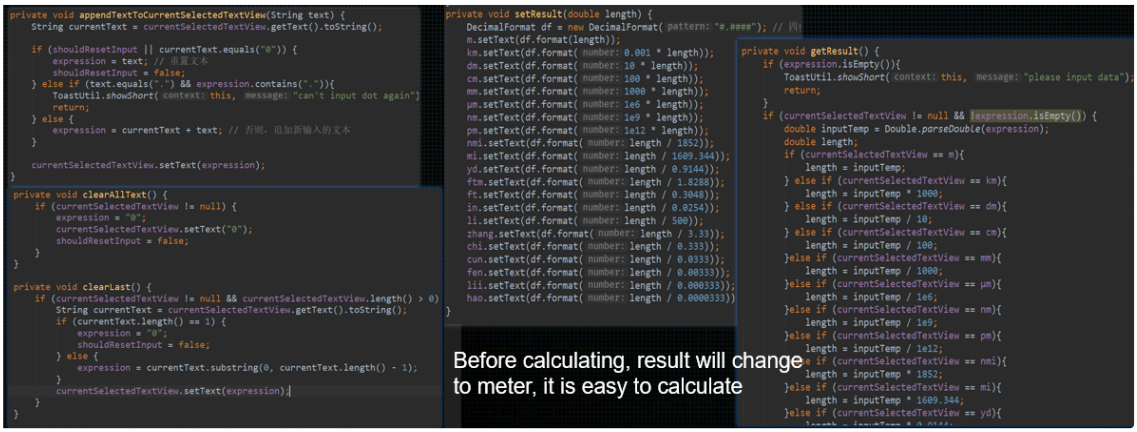


```
initClickListeners(M, m);
initClickListeners(KM, km);
initClickListeners(DM, dm);
initClickListeners(CM,cm);
initClickListeners(MM,mm);
initClickListeners(μM, μm);
initClickListeners(NM,nm);
initClickListeners(PM,pm);
initClickListeners(NMI,nmi);
initClickListeners(MI,mi);
initClickListeners(YD,yd);
initClickListeners(FTM,ftm);
initClickListeners(FT,ft);
initClickListeners(IN,in);
initClickListeners(LI,li);
initClickListeners(ZHANG,zhang);
initClickListeners(CHI,chi);
initClickListeners(CUN,cun);
initClickListeners(FEN,fen);
initClickListeners(LII,lii);
initClickListeners(HAO,hao);

changeTextSelected(m);

setupButtonListeners();
```

```
private void initClickListeners(View layout, final TextView textView) {
    View.OnClickListener listener = (v) -> { changeTextSelected(textView); };

    layout.setOnClickListener(listener);
    textView.setOnClickListener(listener);
}

/**
 * 不断更换选中的textView
 *
 * @param textView
 */
private void changeTextSelected(TextView textView) {
    // 如果有先前选中的 TextView，将其颜色改回黑
    if (currentSelectedTextView != null && currentSelectedTextView != textV
        currentSelectedTextView.setTextColor(Color.BLACK);
    }

    // 设置新 TextView 的颜色并更新当前选中的 TextView
    textView.setTextColor(Color.parseColor( colorString: "#3B67E8"));
    currentSelectedTextView = textView;

    expression = currentSelectedTextView.getText().toString();

    if (currentSelectedLayout != null) {
        currentSelectedLayout.setBackgroundResource(0); // 移除边框
    }

    currentSelectedLayout = (View) textView.getParent();
    currentSelectedLayout.setBackgroundResource(R.drawable.border);

}
```

- # Length

Then I set the click function for all buttons, expression will add the button text, clear button will make the expression empty, backspace button will shorten the expression, equal button will get the result of expression and set all of other unit result, I set meter as the basic calculate unit




```
private void appendTextToCurrentSelectedTextView(String text) {
    String currentText = currentSelectedTextView.getText().toString();

    if (shouldResetInput || currentText.equals("0")) {
        expression = text; // 重置文本
        shouldResetInput = false;
    } else if (text.equals(".") && expression.contains(".")){
        ToastUtil.showShort( context: this,  message: "can't input dot again");
        return;
    } else {
        expression = currentText + text; // 否则，追加新输入的文本
    }

    currentSelectedTextView.setText(expression);
}

private void clearAllText() {
    if (currentSelectedTextView != null) {
        expression = "0";
        currentSelectedTextView.setText("0");
        shouldResetInput = false;
    }
}

private void clearLast() {
    if (currentSelectedTextView != null && currentSelectedTextView.length() > 0)
        String currentText = currentSelectedTextView.getText().toString();
        if (currentText.length() == 1) {
            expression = "0";
            shouldResetInput = false;
        } else {
            expression = currentText.substring(0, currentText.length() - 1);
        }
        currentSelectedTextView.setText(expression);
    }
}
```

```
private void setResult(double length) {
    DecimalFormat df = new DecimalFormat( pattern: "#.####"); // 四
    m.setText(df.format(length));
    km.setText(df.format( number: 0.001 * length));
    dm.setText(df.format( number: 10 * length));
    cm.setText(df.format( number: 100 * length));
    mm.setText(df.format( number: 1000 * length));
    μm.setText(df.format( number: 1e6 * length));
    nm.setText(df.format( number: 1e9 * length));
    pm.setText(df.format( number: 1e12 * length));
    nmi.setText(df.format( number: length / 1852));
    mi.setText(df.format( number: length / 1609.344));
    yd.setText(df.format( number: length / 0.9144));
    ftm.setText(df.format( number: length / 1.8288));
    ft.setText(df.format( number: length / 0.3048));
    in.setText(df.format( number: length / 0.0254));
    li.setText(df.format( number: length / 500));
    zhang.setText(df.format( number: length / 3.33));
    chi.setText(df.format( number: length / 0.333));
    cun.setText(df.format( number: length / 0.0333));
    fen.setText(df.format( number: length / 0.00333));
    lii.setText(df.format( number: length / 0.000333));
    hao.setText(df.format( number: length / 0.0000333))
}
```

```
private void getResult() {
    if (expression.isEmpty()){
        ToastUtil.showShort( context: this,  message: "please input data");
        return;
    }
    if (currentSelectedTextView != null && !expression.isEmpty()) {
        double inputTemp = Double.parseDouble(expression);
        double length;
        if (currentSelectedTextView == m){
            length = inputTemp;
        } else if (currentSelectedTextView == km){
            length = inputTemp * 1000;
        } else if (currentSelectedTextView == dm){
            length = inputTemp / 10;
        } else if (currentSelectedTextView == cm){
            length = inputTemp / 100;
        }else if (currentSelectedTextView == mm){
            length = inputTemp / 1000;
        }else if (currentSelectedTextView == μm){
            length = inputTemp / 1e6;
        }else if (currentSelectedTextView == nm){
            length = inputTemp / 1e9;
        }else if (currentSelectedTextView == pm){
            length = inputTemp / 1e12;
        }else if (currentSelectedTextView == nmi){
            length = inputTemp * 1852;
        }else if (currentSelectedTextView == mi){
            length = inputTemp * 1609.344;
        }else if (currentSelectedTextView == yd){
```

area weight volume

- # Area, Volume, Wight

The screen gif

All of them are similar to Length logic, just change the basic calculating unit

number system

- # Number system

The screen gif

Many logic are similar to length, but just add the button allow logic before make them into 4 kinds of buttons. Different number system allow different buttons to enable to click


```
private Button[] function = new Button[3]; // 清0. 回退. 等号
private Button[] binary = new Button[3]; // 1. 0. 00 二进制
private Button[] octonary = new Button[6]; // 八进制 234567
private Button[] decimal = new Button[2]; // 十进制 89
private Button[] hexadecimal = new Button[6]; // 16进制ABCDEF


Button button0, button00, button1;
Button button2, button3, button4, button5, button6, button7;
Button button8, button9;
Button buttonA, buttonB, buttonC, buttonD, buttonE, buttonF;
Button clearAll, clear, equal;
```

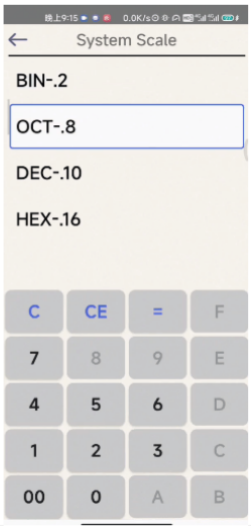
```
private void setButtonsEnabled(Button[] buttons, boolean enabled) {
    for (Button button : buttons) {
        if (button != null) {
            button.setEnabled(enabled);
        }
    }
}
```

```
if (textView == bin) {
    // BIN 被选中
    /*
    允许functions和0,1,00
     */
    setButtonsEnabled(binary, enabled: true);
    setButtonsEnabled(octonary, enabled: false);
    setButtonsEnabled(decimal, enabled: false);
    setButtonsEnabled(hexadecimal, enabled: false);
} else if (textView == oct) {
    // OCT 被选中
    /*
    允许functions和0,1,00,234567
     */
    setButtonsEnabled(binary, enabled: true);
    setButtonsEnabled(octonary, enabled: true);
    setButtonsEnabled(decimal, enabled: false);
    setButtonsEnabled(hexadecimal, enabled: false);
} else if (textView == dec) {
    // DEC 被选中
    /*
    允许functions和0,1,00,234567,89
     */
    setButtonsEnabled(binary, enabled: true);
    setButtonsEnabled(octonary, enabled: true);
    setButtonsEnabled(decimal, enabled: true);
    setButtonsEnabled(hexadecimal, enabled: false);
} else if (textView == hex) {
    setButtonsEnabled(binary, enabled: true);
    setButtonsEnabled(octonary, enabled: true);
    setButtonsEnabled(decimal, enabled: true);
    setButtonsEnabled(hexadecimal, enabled: true);
```

- # Number system

The screen gif

The logic of calculate, just use the existing algorithm of Android

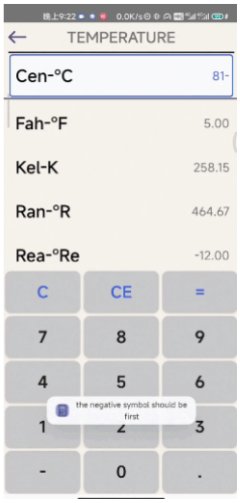



```
private void getResult() {
    if (expression.isEmpty()){
        ToastUtil.showShort( context: this,  message: "please input data");
        return;
    }
    if (currentSelectedTextView != null && !expression.isEmpty()) {
        try {
            long decimalValue;
            if (currentSelectedTextView == bin) {
                decimalValue = parseLongWithOverflowCheck(expression,  radix: 2);
                oct.setText(Long.toOctalString(decimalValue));
                dec.setText(Long.toString(decimalValue));
                hex.setText(Long.toHexString(decimalValue).toUpperCase());
            } else if (currentSelectedTextView == oct) {
                decimalValue = parseLongWithOverflowCheck(expression,  radix: 8);
                bin.setText(Long.toBinaryString(decimalValue));
                dec.setText(Long.toString(decimalValue));
                hex.setText(Long.toHexString(decimalValue).toUpperCase());
            } else if (currentSelectedTextView == dec) {
                decimalValue = parseLongWithOverflowCheck(expression,  radix: 10);
                bin.setText(Long.toBinaryString(decimalValue));
                oct.setText(Long.toOctalString(decimalValue));
                hex.setText(Long.toHexString(decimalValue).toUpperCase());
            } else if (currentSelectedTextView == hex) {
                decimalValue = parseLongWithOverflowCheck(expression,  radix: 16);
                bin.setText(Long.toBinaryString(decimalValue));
                oct.setText(Long.toOctalString(decimalValue));
                dec.setText(Long.toString(decimalValue).toUpperCase());
            }
        } catch (NumberFormatException e) {
            ToastUtil.showShort( context: this,  message: "Number too large or invalid format");
        }
    }
```

temperature

- # Temperature

The screen gif



All of them are similar to Length logic, just change the basic calculating unit
And allow typing negative "-" to calculate
It just can set at the start of expression and appear one time

English

- # English

The screen pic

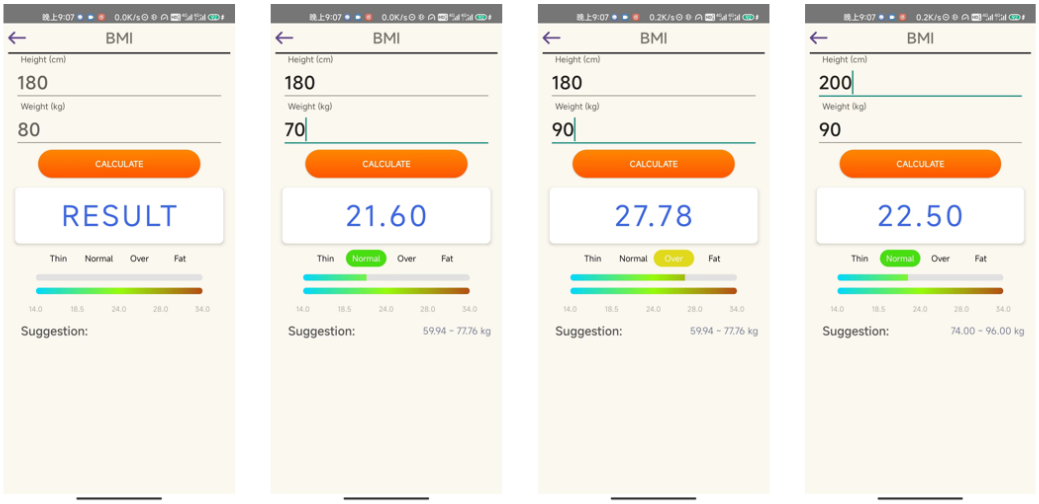
All of them are similar to Length logic, just change the basic calculating unit
And allow typing one number to get the result, the result will be calculated by function I created, and if the expression is too long, like 13 or more bits, the result will over billion, it will Toast the result is too large

BMI

- BMI

The result will show the BMI index, I set two progress bar, the first will show the index length, the second is just show the style, the BMI condition also have style. At the end of page, it will show the suggestion weight of user's height

The screen pic



- BMI

This is the style of bar

```xml
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <!-- Background Track -->
    <item android:id="@android:id/background">
        <shape>
            <corners android:radius="5dp" />
            <solid android:color="#E0E0E0" />
        </shape>
    </item>

    <!-- Progress -->
    <item android:id="@android:id/progress">
        <clip>
            <shape>
                <corners android:radius="5dp" />
                <gradient
                    android:angle="0"
                    android:centerColor="#97FC09"
                    android:startColor="#00D9FF"
                    android:endColor="#B44D13"
                    android:type="linear" />
            </shape>
        </clip>
    </item>
</layer-list>
```

This is the function to change the style of text and bar

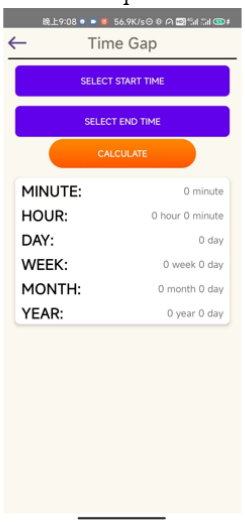
```java
private void updateTag(float bmiValue) {
    thin.setBackgroundResource(0);
    normal.setBackgroundResource(0);
    over.setBackgroundResource(0);
    fat.setBackgroundResource(0);
    thin.setTextColor(Color.BLACK);
    normal.setTextColor(Color.BLACK);
    fat.setTextColor(Color.BLACK);
    over.setTextColor(Color.BLACK);

    if (bmiValue < 18.5){
        thin.setBackgroundResource(R.drawable.thin);
        thin.setTextColor(Color.WHITE);
    } else if (18.5 <= bmiValue && bmiValue < 24){
        normal.setBackgroundResource(R.drawable.normal);
        normal.setTextColor(Color.WHITE);
    } else if (24 <= bmiValue && bmiValue < 28){
        over.setBackgroundResource(R.drawable.over);
        over.setTextColor(Color.WHITE);
    } else {
        fat.setBackgroundResource(R.drawable.fat);
        fat.setTextColor(Color.WHITE);
    }
    int progress = (int) ((bmiValue - 14) / (34 - 14) * MAX_BMI_SEEKBAR_VALUE);
    bmiProgressBar.setProgress(progress);
}
```

time gap

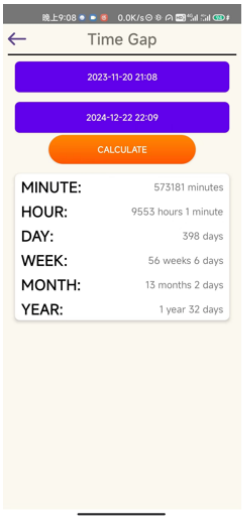
- # Time Gap

  User can choose the start time and end time, then get the minutes, hours, days, week and days, month and days, year and days between them. The day, week, month, year use zero point to calculate

  The screen pic

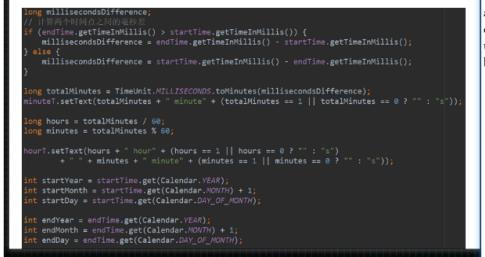
  

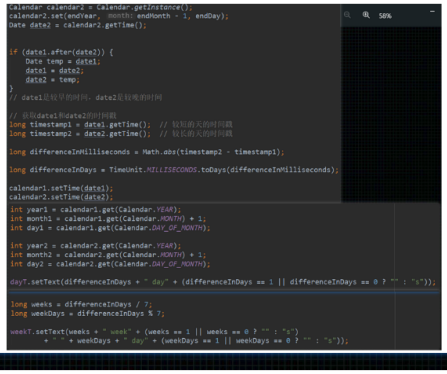
- ## Time Gap

  First get the timestamp of two time, then we can get the hours and minutes

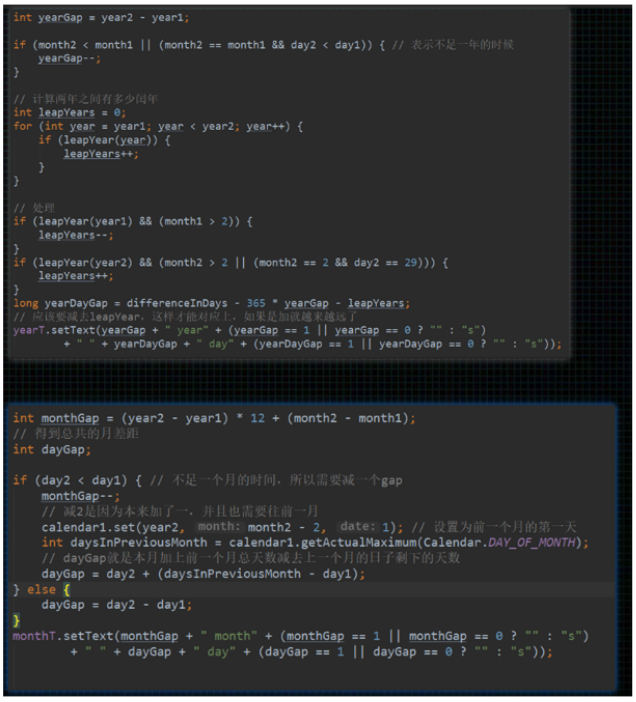
  ## Time Gap

  Then we can get the year, month, date of start time and end time, then get the earlier and later date, we will use them to get the days between and weeks between

- # Time Gap

  Calculate yearGap: firstly get the minus number between two years, then find all leap years' number because leap year has 366 days, when we get the dayGap we need minus leap years' number to get the right days

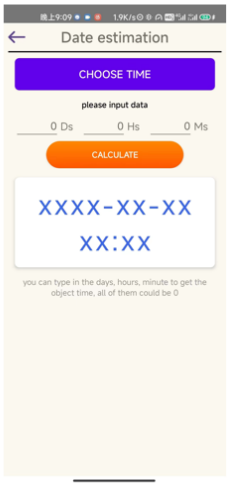
  Calculate monthGap

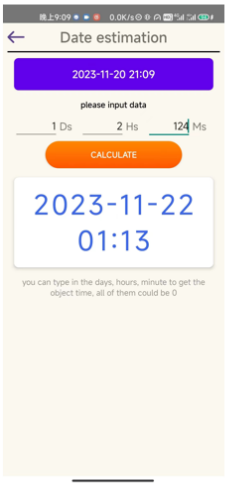
date

- # Date

The screen pic

User can choose the time, then type in days or hours or minutes to get time user wants to get, it is the convenient to know the deadline





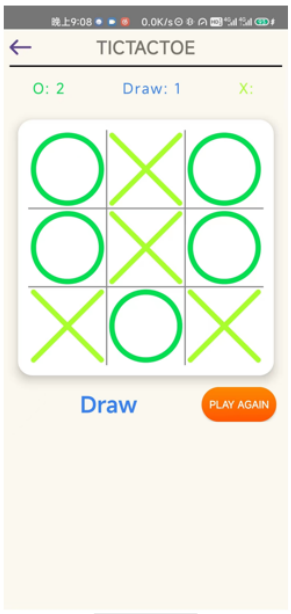
This is the code to get time, just add days, hours, minutes in DAY_OF_MONTH, HOUR_OF_DAY and MINUTE, the system will automatically set the correct time

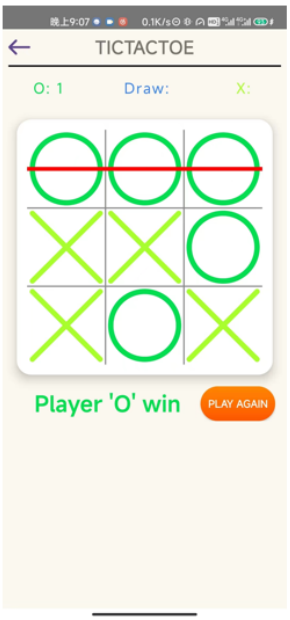
```
// 从EditText获取输入的天数、小时和分钟
if (!day.getText().toString().isEmpty()){
    addedDays = Integer.parseInt(day.getText().toString());
}
if (!hour.getText().toString().isEmpty()){
    addedHours = Integer.parseInt(hour.getText().toString());
}
if (!minute.getText().toString().isEmpty()){
    addedMinutes = Integer.parseInt(minute.getText().toString());
}


// 使用Calendar计算新的日期和时间
Calendar newDateTime = Calendar.getInstance();
newDateTime.set(year, month, dayOfMonth, hourOfDay, minuteOfHour);
// 加日期，需基于日的某一大的基础加
newDateTime.add(Calendar.DAY_OF_MONTH, addedDays);
// 需要在日期的时间基础加小时
newDateTime.add(Calendar.HOUR_OF_DAY, addedHours);
newDateTime.add(Calendar.MINUTE, addedMinutes);

// 格式化新的日期时间并设置到TextView
SimpleDateFormat format = new SimpleDateFormat( pattern: "yyyy-MM-dd HH:mm", Locale.getDefault());
String formattedNewDateTime = format.format(newDateTime.getTime());
date.setText(formattedNewDateTime);
tch (NumberFormatException e) {
ToastUtil.showShort( context: this,  message: "you input the num too large!!");
```

tictactoe

- # TicTacToe

The screen pic



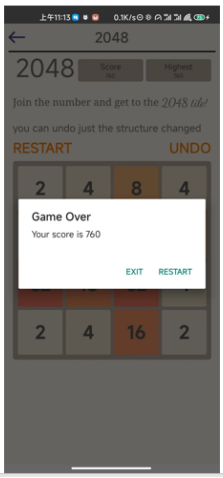


- # TicTacToe

I create one class TicTacToeBoardView entends View to draw the game panel, the activity of the tictactoe will switch data and information with this class. The game class will contains Rect[][] cells to contain all locations to choose which blank could draw the chess pieces. After click one position, the game panel will draw, and boolean[][] linesToDraw to check the win condition. In activity class, all textview and score and player status and game over flag will dynamically change always. When one chess piece draws, the system will redraw the whole panel and check the over flag

- # 2048

The screen pic



I create 2 new class to init the game, first is Card class, which will show the condition of number, and show different background color; one is GameView to init the layout and add game logic in it.

This is change background color logic

```
public void setNum(int num){
    this.num = num;
    if (num == 0){
        label.setText("");
    } else {
        label.setText(String.valueOf(num));
    }
    updateBackgroundColor();
}

private void updateBackgroundColor() {
    if (num == 0) {
        ((GradientDrawable) getBackground()).setColor(Color.parseColor( colorString: "
        return;
    }
    int color = Color.parseColor(TileColor.getColorByNumber(num));
    // 只改颜色不改样式
    ((GradientDrawable) getBackground()).setColor(color);

    // Change text color based on the background for better visibility
    if (num > 512 && num <= 8192) {
        label.setTextSize(24);
        label.setTextColor(Color.parseColor( colorString: "#f0f0f0")); // For darker b
    } else if (num < 100000 && num > 8192){
        label.setTextSize(20);
        label.setTextColor(Color.parseColor( colorString: "#f0f0f0")); // For darker b
    } else if (num > 100000){
        label.setTextSize(16);
        label.setTextColor(Color.parseColor( colorString: "#f0f0f0")); // For darker b
    } else {
        label.setTextSize(32);
        label.setTextColor(Color.parseColor( colorString: "#5f5d5d"));
    }
}
```

- # 2048

This is the logic init the panel, the size is calculated by itself

```
private void initGameView() {
    mTouchSlop = ViewConfiguration.get(getContext()).getScaledTouchSlop();

    post(() -> {
        int width = getWidth();
        tileMargin = 6dp;
        int paddingTotal = 2 * 8dp;
        tileSize = (width - paddingTotal - tileMargin * (GRID_SIZE - 1)) / GRID_SIZE;

        // 初始化卡片
        addCards(tileSize, tileMargin);
        start();
    });
}
```

```
private void addCards(int tileSize, int tileMargin) {
    for (int y = 0; y < GRID_SIZE; y++) {
        for (int x = 0; x < GRID_SIZE; x++) {
            Card card = new Card(getContext());
            card.setNum(0); // 初始化数字为0,同时不会显示出来数
            addView(card, tileSize, tileSize);

            // 设置边距
            GridLayout.LayoutParams params = (GridLayout.LayoutParams) card.getLayoutParams()
            int halfMargin = tileMargin / 2;
            params.setMargins(halfMargin, halfMargin, halfMargin, halfMargin);
            card.setLayoutParams(params);

            cardsMap[y][x] = card;
        }
    }
}
```

The logic of restart the game and add 2 random position I use Scale method to make the animate when one new card displays

```
private void addRandomCard() {
    emptyPoints.clear();
    for (int y = 0; y < GRID_SIZE; y++) {
        for (int x = 0; x < GRID_SIZE; x++) {
            // 如果当前位置是0的话就记录下来
            if (cardsMap[y][x].getNum() == 0) {
                emptyPoints.add(new Point(x, y));
            }
        }
    }
    if (emptyPoints.isEmpty()) {
        return; // 没有空位时直接返回
    }
    // 随机得到一个点的位置
    Point randomPoint = emptyPoints.get(new Random().nextInt(emptyPoints.size()));
    cardsMap[randomPoint.y][randomPoint.x].setNum(Math.random() > 0.15 ? 2 : 4); // 85%概率生成2, 10
    // 动画
    cardsMap[randomPoint.y][randomPoint.x].setScaleX((float) 0.4);
    cardsMap[randomPoint.y][randomPoint.x].setScaleY((float) 0.4);
    cardsMap[randomPoint.y][randomPoint.x].animate().scaleX(1).scaleY(1).setDuration(300).start();

}
```

```
public void start() {
    for (int y = 0; y < GRID_SIZE; y++) {
        for (int x = 0; x < GRID_SIZE; x++) {
            cardsMap[y][x].setNum(0);
        }
    }
    addRandomCard();
    addRandomCard();
    score = 0;
    updateScore();
}
```

- # 2048

The logic of save last step and undo the move, also show the move logic and animate

```
private void saveLastState() {
    for (int y = 0; y < GRID_SIZE; y++) {
        for (int x = 0; x < GRID_SIZE; x++) {
            if (lastCardsMap[y][x] == null) {
                lastCardsMap[y][x] = new Card(getContext());
            }
            lastCardsMap[y][x].setNum(cardsMap[y][x].getNum());
        }
    }
    tempScore = score;
}
```

```
public void undo() {
    boolean canUndo = false; // 增加了一个标志来检查是否可以回退
    for (int y = 0; y < GRID_SIZE; y++) {
        for (int x = 0; x < GRID_SIZE; x++) {
            if (lastCardsMap[y][x] != null && cardsMap[y][x].getNum() != lastCardsMap[y][x].getNu
                canUndo = true; // 如果存在状态改变，则可以回退
                cardsMap[y][x].setNum(lastCardsMap[y][x].getNum());
            }
        }
    }
    if (canUndo) {
        score = tempScore; // 恢复分数
        updateScore();
    } else {
        ToastUtil.showShort(getContext(), message: "you can't back");
    }
}
```

```
saveLastState();

boolean needAddCard = false;
for (int y = 0; y < GRID_SIZE; y++) {
    for (int x = GRID_SIZE - 1; x >= 0; x--) {
        for (int x1 = x - 1; x1 >= 0; x1--) {
            if (cardsMap[y][x1].getNum() > 0) {
                if (cardsMap[y][x].getNum() == 0) {
                    cardsMap[y][x].setNum(cardsMap[y][x1].getNum());
                    cardsMap[y][x1].setNum(0);
                    x++; // Move further to the right
                    needAddCard = true;

                } else if (cardsMap[y][x].equal(cardsMap[y][x1])) {
                    cardsMap[y][x].setNum(cardsMap[y][x].getNum() * 2);

                    cardsMap[y][x].setScaleX((float) 1.1);
                    cardsMap[y][x].setScaleY((float) 1.1);
                    cardsMap[y][x].animate().scaleX(1).scaleY(1).setDuration(200).start();
                    score += cardsMap[y][x].getNum() * 2;

                    cardsMap[y][x1].setNum(0);
                    needAddCard = true;

                }
                break;
            }
        }
    }
}

if (needAddCard) {
    addRandomCard();
    updateScore();
}
```

- # 2048

  The logic of whether can move, every time after swipe the screen will load this function

  Use SharedPreference to store the highest score

```
private boolean canMove() {
    // 检查是否有空的卡片
    for (int y = 0; y < GRID_SIZE; y++) {
        for (int x = 0; x < GRID_SIZE; x++) {
            if (cardsMap[y][x].getNum() == 0) {
                return true; // 存在空卡片，可以移动
            }
        }
    }

    // 检查是否有可合并的卡片
    for (int y = 0; y < GRID_SIZE; y++) {
        for (int x = 0; x < GRID_SIZE; x++) {
            int num = cardsMap[y][x].getNum();
            if ((x < GRID_SIZE - 1 && num == cardsMap[y][x + 1].getNum()) ||
                    (y < GRID_SIZE - 1 && num == cardsMap[y + 1][x].getNum())) {
                return true; // 存在可合并的卡片
            }
        }
    }

    // 既没有空卡片也无法合并，游戏结束
    return false;
}
```

```
public void updateScore(int newScore) {
    score.setText(String.valueOf(newScore));
    if (highScore < newScore) {
        highScore = newScore;
        height_score.setText(String.valueOf(newScore));
        setHighScore(newScore); // 保存新的最高分
    }
}


private int getHighScore() {
    SharedPreferences prefs = getSharedPreferences(PREFS_NAME, MODE_PRIVATE);
    return prefs.getInt(HIGH_SCORE, defValue: 0); // 默认为0
}

private void setHighScore(int score) {
    SharedPreferences prefs = getSharedPreferences(PREFS_NAME, MODE_PRIVATE);
    SharedPreferences.Editor editor = prefs.edit();
    editor.putInt(HIGH_SCORE, score);
    editor.apply();
}
```

Matrix logic

21207315 XuZhong
I also add the apk file to the zip, you can download it
The Github repository created by myself: Zhong0118/calculatorSuper (github.com)
The link i refer: Android 计算器-CSDN博客(this is just provide me the method of basic calculator - alpha version, the beta version and final version is created by myself)

Result

- At the last fragment page I create the function of matrix calculation, I will give the user 9 calculation expressions, `plus, minus, multiply between two matrixes`, `the determinant number, the inverse result, the adjoint result of one square matrix`, `the transpose, rank number of one matrix`, and `dual matirx of one vector`. (the largest dimension of matrix is 10, one reason is that we really don't need larger matrix, one reason is that the phone might not could show result clearly, and another reason is that the time complexity of some matrix algorithms increases exponentially for the resulting operation, and if the dimension is too large, the user experience will be affected)
- **Plus two matrix**
  - two matrix should have the same m×n structure, so i just allow the user to type in the Row number and Col number, then the screen will show 2 m×n EditText views and user could type in number, when type calculate button, the region of matrix will refresh and show the result
  - before calculate, if the cell of matrix user haven't input, the number is default 0
  - the input number could be integer, double number or negative number
  - the result will keep two decimals
- **Minus two matrix**
  - Everything is like **plus**, except one is adding and one is subtracting
- **Multiply two matrix**
  - two matrix should have the m×n and n×k structures, so i allow user to type in Row_A, Col_A, Row_B, Col_B, in fact ,i create a kind of function to made sure that Col_A would always be the same as Row_B, and no matter which input box I typed in, the other input box would change, then screen will show 2 views and allow user to input number.
  - before calculate, if the cell of matrix user haven't input, the number is default 0
  - the input number could be integer, double number or negative number

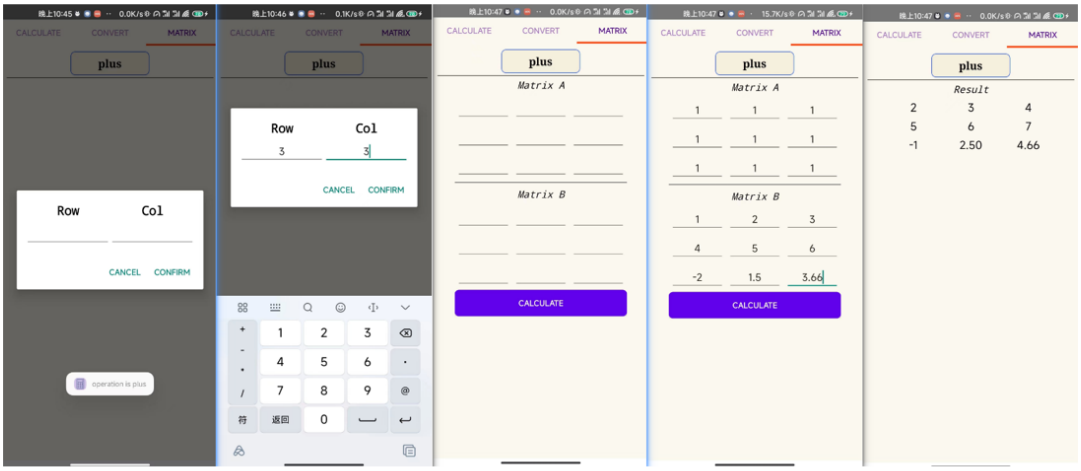
    - the result will keep two decimals
- **Transpose**
    - first the app will allow user to set the m×n structure, then after calculate, the screen will show the result after transpose with n×m structure
- **Rank**
    - it will allow user to type in the m×n structure and get the result number of matrix rank number
- **Inverse, Determinant, Adjoint**
    - all of these will allow user to type in the dimension of the square matrix to get the result of its determinant number, adjoint matrix and inverse matrix
- **Dual**
    - this will allow user to type in 3 numbers of one vector, it will get the 3×3 matrix. The reason I did this is because computer graphics uses this kind of matrix for linear calculations

Code technique

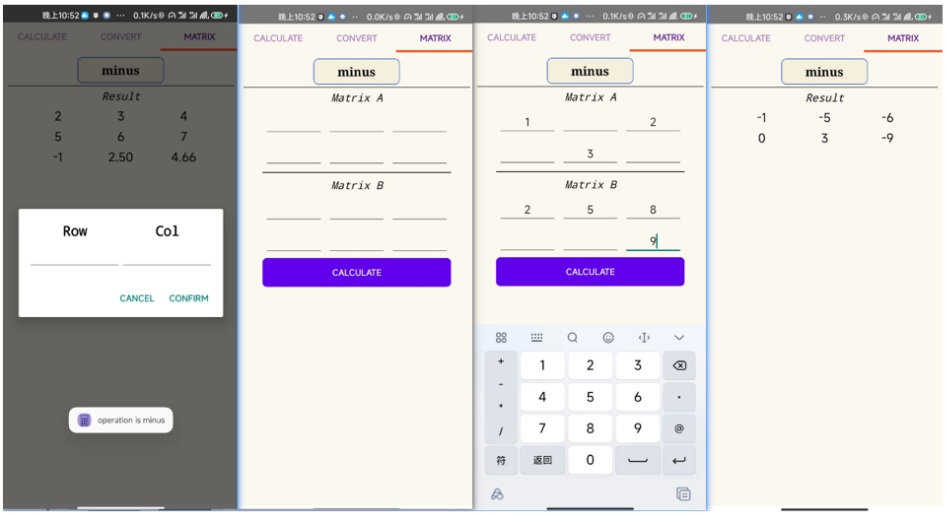
- Dynamically generate edit boxes corresponding to m×n structures
- Distinguish between page construction logic and computational logic
- Use the appropriate algorithm to extract the contents of the input box to the corresponding two-dimensional array for easy operation
- Rely on map to help separate page generation for different matrix calculations
- Much of the focus is on solving the logic of matrix calculations and using appropriate methods to present matrix dimensions and allow user input. It is important to design a reasonable and user-friendly display page

PPT pics

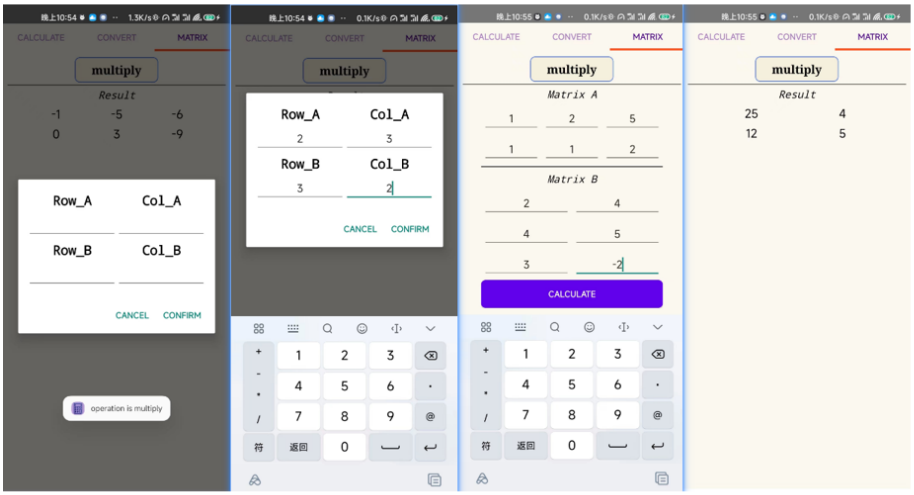
Plus page



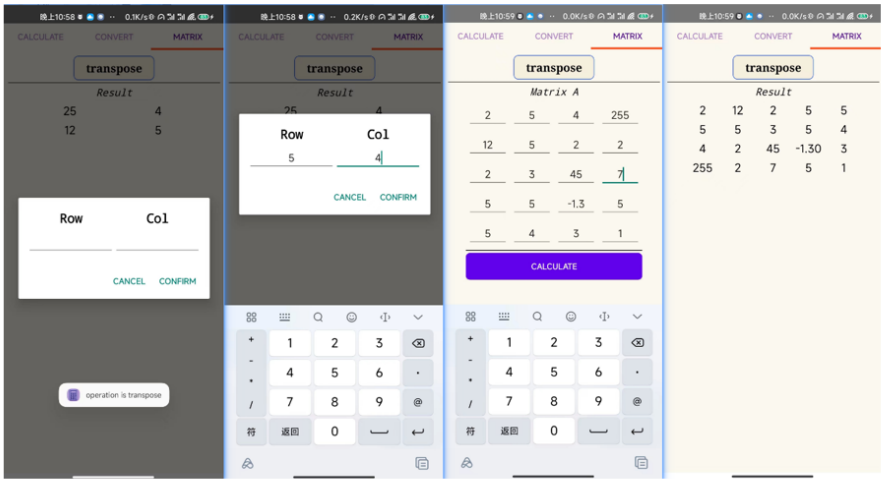
Minus page



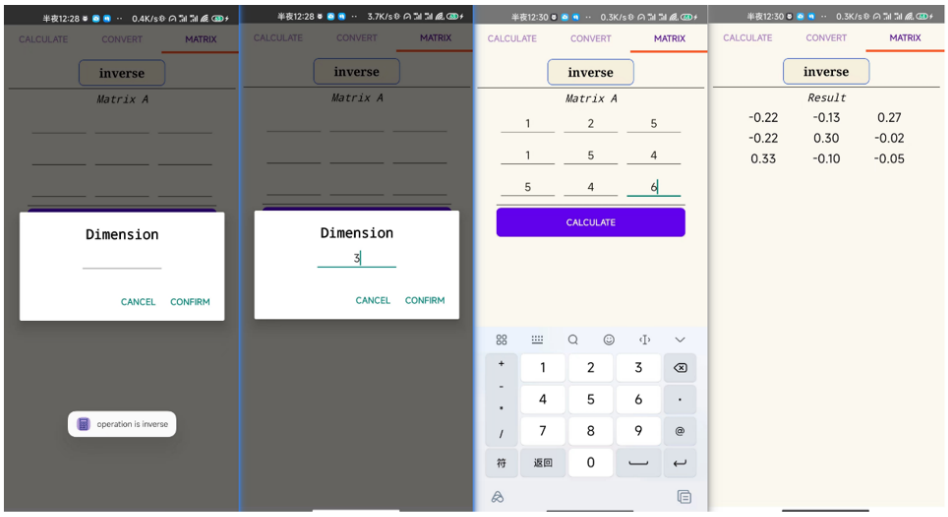
Multiply page



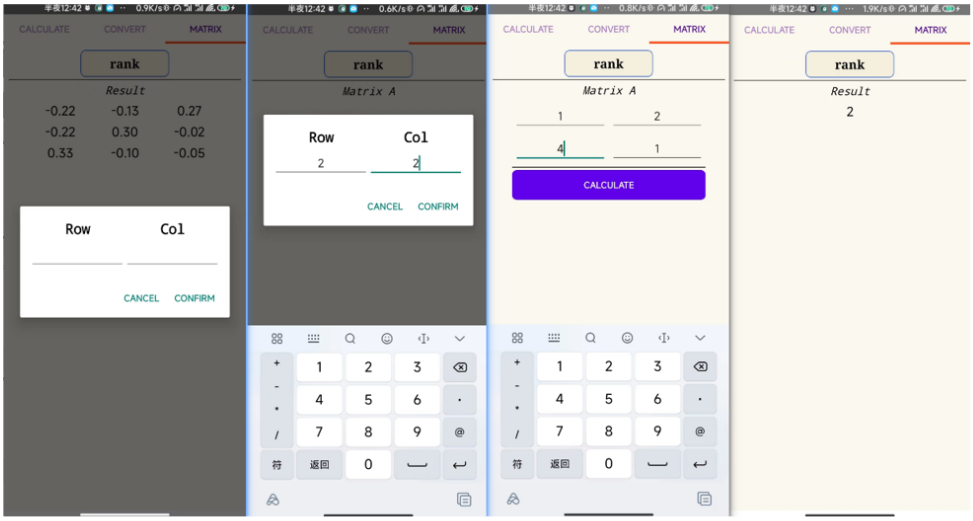
Transpose page



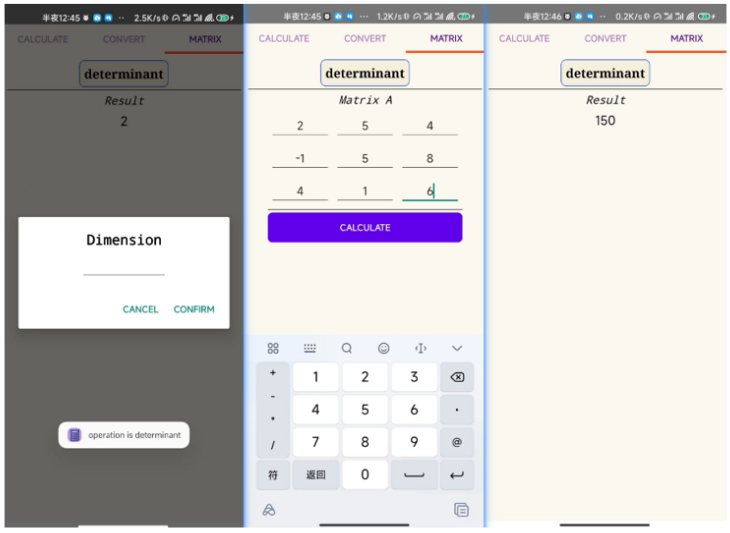
Inverse page



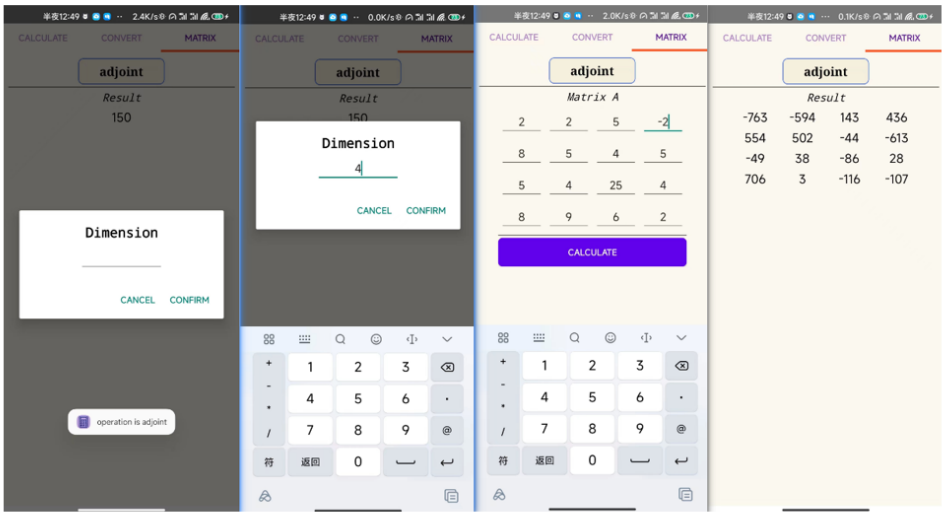
Rank page



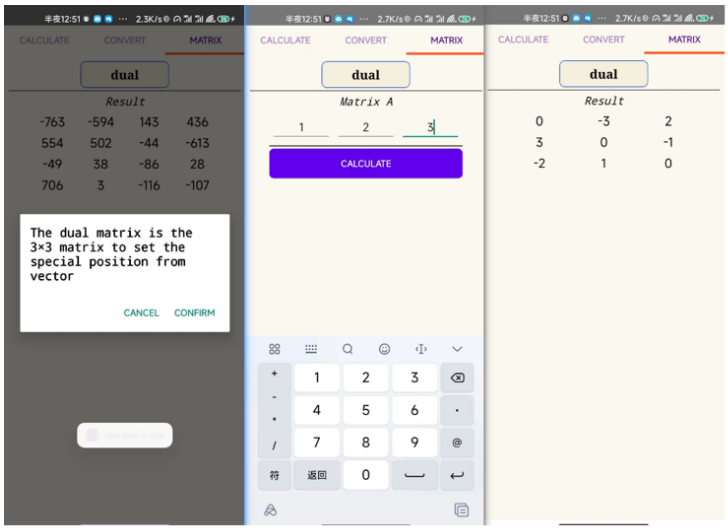
Determinant page



Adjoint page



Dual page



Matrix calculate logic of transpose, add, minus

```java
public static double[][] getTranspose(double[][] matrix) {
    int rows = matrix.length;
    int cols = matrix[0].length;

    double[][] result = new double[cols][rows];

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            result[j][i] = matrix[i][j];
        }
    }

    return result;
}
```

```java
/**
 * 加法
 *
 * @param aMatrix
 * @param bMatrix
 * @return
 * @throws IllegalArgumentException
 */
public static double[][] add(double[][] aMatrix, double[][] bMatrix) throws IllegalArgumentException {
    // 添加维度检查
    if (aMatrix.length != bMatrix.length || aMatrix[0].length != bMatrix[0].length) {
        throw new IllegalArgumentException("矩阵维度不匹配");
    }

    int rows = aMatrix.length;
    int cols = aMatrix[0].length;
    double[][] result = new double[rows][cols];

    // 执行加法操作
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            result[i][j] = aMatrix[i][j] + bMatrix[i][j];
        }
    }

    return result;
}
```

```java
/**
 * 减法
 *
 * @param aMatrix
 * @param bMatrix
 * @return
 * @throws IllegalArgumentException
 */
public static double[][] minus(double[][] aMatrix, double[][] bMatrix) throws IllegalArgumentException {
    // 添加维度检查
    if (aMatrix.length != bMatrix.length || aMatrix[0].length != bMatrix[0].length) {
        throw new IllegalArgumentException("矩阵维度不匹配");
    }

    int rows = aMatrix.length;
    int cols = aMatrix[0].length;
    double[][] result = new double[rows][cols];

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            result[i][j] = aMatrix[i][j] - bMatrix[i][j];
        }
    }

    return result;
}
```

Matrix calculate logic of multiply

```java
/**
 * 乘法
 *
 * @param aMatrix
 * @param bMatrix
 * @return
 * @throws IllegalArgumentException
 */
public static double[][] multiply(double[][] aMatrix, double[][] bMatrix) throws IllegalArgumentException {
    int aRows = aMatrix.length;
    int aCols = aMatrix[0].length;

    int bRows = bMatrix.length;
    int bCols = bMatrix[0].length;

    if (aCols != bRows) {
        throw new IllegalArgumentException("矩阵无法相乘，维度不匹配");
    }

    double[][] result = new double[aRows][bCols];

    for (int i = 0; i < aRows; i++) {
        for (int j = 0; j < bCols; j++) {
            for (int k = 0; k < aCols; k++) {
                result[i][j] += aMatrix[i][k] * bMatrix[k][j];
            }
        }
    }

    return result;
}
```

Matrix calculate logic of determinant

```java
public static double determinant(double[][] matrix) {
    int n = matrix.length;
    double[][] lu = new double[n][n];
    int swaps = 0;

    // 将输入矩阵复制到lu矩阵中
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            lu[i][j] = matrix[i][j];
        }
    }

    // 进行LU分解
    for (int k = 0; k < n; k++) {
        // 部分主元选取
        int pivot = k;
        for (int i = k + 1; i < n; i++) {
            if (Math.abs(lu[i][k]) > Math.abs(lu[pivot][k])) {
                pivot = i;
            }
        }

        // 交换行
        if (pivot != k) {
            double[] temp = lu[pivot];
            lu[pivot] = lu[k];
            lu[k] = temp;
            swaps++;
        }

        // 检查奇异矩阵
        if (Math.abs(lu[k][k]) < 1e-9) {
            return 0;
        }

        // 分解步骤
        for (int i = k + 1; i < n; i++) {
            lu[i][k] /= lu[k][k];
            for (int j = k + 1; j < n; j++) {
                lu[i][j] -= lu[i][k] * lu[k][j];
            }
        }
    }

    // 计算行列式
    double det = (swaps % 2 == 0) ? 1 : -1;
    for (int i = 0; i < n; i++) {
        det *= lu[i][i];
    }

    return det;
}
```

cofactor

```java
/**
 * 代数余子式
 *
 * @param matrix
 * @param temp
 * @param p
 * @param q
 * @param n
 */
private static void getCofactor(double[][] matrix, double[][] temp, int p, int q, int n) {
    int i = 0, j = 0;

    for (int row = 0; row < n; row++) {
        for (int col = 0; col < n; col++) {
            if (row != p && col != q) {
                temp[i][j++] = matrix[row][col];

                if (j == n - 1) {
                    j = 0;
                    i++;
                }
            }
        }
    }
}
```

adjoint

```java
/**
 * 伴随矩阵
 *
 * @param matrix
 * @return
 */
public static double[][] adjugate(double[][] matrix) {
    int n = matrix.length;
    double[][] adj = new double[n][n];
    double[][] cofactorMatrix = new double[n - 1][n - 1];

    if (n == 1) {
        adj[0][0] = 1;
        return adj;
    }

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            getCofactor(matrix, cofactorMatrix, i, j, n);
            int sign = ((i + j) % 2 == 0) ? 1 : -1;
            adj[i][j] = sign * determinant(cofactorMatrix);
        }
    }

    return getTranspose(adj); // 使用已有的转置方法
}
```

Matrix calculate logic of inverse and rank

```java
public static double[][] inverse(double[][] matrix) throws IllegalArgumentException {
    double det = determinant(matrix);
    if (Math.abs(det) < 1e-9) {
        double[][] a = {{0}};
        return a;
    }

    double[][] adj = adjugate(matrix);
    return multiplyByConstant(adj, constant: 1.0 / det);
}

/**
 * 矩阵乘以常数
 *
 * @param matrix
 * @param constant
 * @return
 */
private static double[][] multiplyByConstant(double[][] matrix, double constant) {
    double[][] result = new double[matrix.length][matrix[0].length];
    for (int i = 0; i < matrix.length; i++) {
        for (int j = 0; j < matrix[i].length; j++) {
            result[i][j] = matrix[i][j] * constant;
        }
    }
    return result;
}
```

```java
public static int rankOfMatrix(double[][] matrix) {
    int m = matrix.length; // number of rows
    int n = matrix[0].length; // number of columns
    int rank = 0;
    boolean[] rowSelected = new boolean[m];

    for (int col = 0; col < n; col++) {
        int i;
        for (i = 0; i < m; i++) {
            if (!rowSelected[i] && Math.abs(matrix[i][col]) > 1E-10) {
                break;
            }
        }

        if (i < m) {
            rank++;
            rowSelected[i] = true;
            for (int k = 0; k < m; k++) {
                if (k != i) {
                    double factor = matrix[k][col] / matrix[i][col];
                    for (int j = col; j < n; j++) {
                        matrix[k][j] -= factor * matrix[i][j];
                    }
                }
            }
        }
    }
    return rank;
}
```

The snipper

```java
public class MatrixChooseUtil {

    public static final Map<String, Integer> operationMap = new HashMap<>();

    static {
        operationMap.put("plus", 0);
        operationMap.put("minus", 1);
        operationMap.put("multiply", 2);
        operationMap.put("transpose", 3);
        operationMap.put("inverse", 4);
        operationMap.put("rank", 5);
        operationMap.put("determinant", 6);
        operationMap.put("adjoint", 7);
        operationMap.put("dual", 8);
    }

    public static void initSpinner(Context context, Spinner spinner) {
        List<String> operationValues = new ArrayList<>();
        operationValues.add("plus");
        operationValues.add("minus");
        operationValues.add("multiply");
        operationValues.add("transpose");
        operationValues.add("inverse");
        operationValues.add("rank");
        operationValues.add("determinant");
        operationValues.add("adjoint");
        operationValues.add("dual");
        ArrayAdapter<String> adapter = new ArrayAdapter<>(context, R.layout.matrixitem, operationValues);
        adapter.setDropDownViewResource(R.layout.matrixdrop);
        spinner.setAdapter(adapter);
    }
```

Will check different creating logic

```java
@Override
public Dialog onCreateDialog(Bundle savedInstanceState) {
    choise = getArguments().getInt( key: "choise");
    AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
    // Choose the right layout based on the choise
    switch (choise) {
        case 0:
        case 1:
            am = true;
            builder.setView(R.layout.dialog_two_inputs);
            condition = 0;
            break;
        case 3:
        case 5:
            am = false;
            builder.setView(R.layout.dialog_two_inputs);
            condition = 0;
            break;
        case 2:
            builder.setView(R.layout.dialog_four_inputs);
            condition = 1;
            am = true;
            // Add text changed listeners to sync Matrix A cols with Matrix B rows
            break;
        case 4:
        case 6:
        case 7:
            builder.setView(R.layout.dialog_one_input);
            condition = 2;
            am = false;
            break;
```

Multiply synchronize colA and rowB

```java
// Add the TextWatchers to synchronize rowB and colA
TextWatcher syncTextWatcher = new TextWatcher() {
    @Override
    public void beforeTextChanged(CharSequence s, int start, int count, int after) {
    }

    @Override
    public void onTextChanged(CharSequence s, int start, int before, int count) {
        rowB.removeTextChangedListener( watcher: this);
        colA.removeTextChangedListener( watcher: this);

        rowB.setText(s);
        colA.setText(s);

        rowB.addTextChangedListener( watcher: this);
        colA.addTextChangedListener( watcher: this);
    }

    @Override
    public void afterTextChanged(Editable s) {
    }
};

rowB.addTextChangedListener(syncTextWatcher);
colA.addTextChangedListener(syncTextWatcher);
```

Will check different creating logic

```java
if (condition == 0){
    rowA = dialogView.findViewById(R.id.editTextRow);
    colA = dialogView.findViewById(R.id.editTextCol);

    if (rowA.getText().toString().isEmpty() || colA.getText().toString().isEmpty()){
        ToastUtil.showShort(getActivity(), message: "please input all values");
        return;
    }
    row1 = Integer.parseInt(rowA.getText().toString());
    col1 = Integer.parseInt(colA.getText().toString());
    if (checkNum(row1) || checkNum(col1)){
        ToastUtil.showShort(getActivity(), message: "the num is too large");
        return;
    }
    listener.onMatrixSizeSelected(row1, col1, row1, col1, am);
    dismiss();
} else if (condition == 1){
    rowA = dialogView.findViewById(R.id.editTextRow1);
    colA = dialogView.findViewById(R.id.editTextCol1);
    rowB = dialogView.findViewById(R.id.editTextRow2);
    colB = dialogView.findViewById(R.id.editTextCol2);

    if (rowA.getText().toString().isEmpty() || colA.getText().toString().isEmpty()
            || rowB.getText().toString().isEmpty() || colB.getText().toString().isEmpty()){
        ToastUtil.showShort(getActivity(), message: "please input all values");
        return;
    }
    row1 = Integer.parseInt(rowA.getText().toString());
    col1 = Integer.parseInt(colA.getText().toString());
    row2 = Integer.parseInt(rowB.getText().toString());
    col2 = Integer.parseInt(colB.getText().toString());

    if (checkNum(row1) || checkNum(col1)
    || checkNum(row2) || checkNum(col2)){
        ToastUtil.showShort(getActivity(), message: "the num is too large");
        return;
    }

    listener.onMatrixSizeSelected(row1, col1, row2, col2, am);
    dismiss();
} else if (condition == 2){
    rowA = dialogView.findViewById(R.id.editTextDimension);
    if (rowA.getText().toString().isEmpty()){
        ToastUtil.showShort(getActivity(), message: "please input all values");
        return;
    }
    row1 = Integer.parseInt(rowA.getText().toString());

    if (checkNum(row1)){
        ToastUtil.showShort(getActivity(), message: "the num is too large");
        return;
    }
    listener.onMatrixSizeSelected(row1, row1, row2: 0, col2: 0,am);
    dismiss();
} else {
    listener.onMatrixSizeSelected( row1: 1, col1: 3, row2: 0, col2: 0,am);
    dismiss();
}
```

Logic of creating view dynamically

```
private void createGrids(int row1, int col1, int row2, int col2, boolean am) {
    container = view.findViewById(R.id.scrollViewContainer);
    container.removeAllViews();
    matrixAEditTexts.clear();
    matrixBEditTexts.clear();

    if (row1 > 0 && col1 > 0) {
        matrixA = new double[row1][col1];
        TextView matrixA = new TextView(new ContextThemeWrapper(getContext(), R.style.matrixText), a
        matrixA.setText("Matrix A");
        matrixA.setTextSize(TypedValue.COMPLEX_UNIT_SP,  size: 18); // 设置文本大小
        matrixA.setTextColor(Color.BLACK); // 设置文本颜色
        matrixA.setGravity(Gravity.CENTER); // 设置文本居中
        GridLayout gridLayout1 = createGridLayout(row1, col1,  isA: true);
        View divider = new View(getContext());
        LinearLayout.LayoutParams layoutParams = new LinearLayout.LayoutParams(
                LinearLayout.LayoutParams.MATCH_PARENT,
                1dp
        );
        int marginTop = 3sp;
        int marginBottom = 4sp;
        layoutParams.setMargins( left: 0, marginTop,  right: 0, marginBottom);
        divider.setLayoutParams(layoutParams);
        divider.setBackgroundColor(Color.BLACK);
        container.addView(matrixA);
        container.addView(gridLayout1);
        container.addView(divider);

    }

    if (row2 > 0 && col2 > 0 && am) {
        matrixB = new double[row2][col2];
        TextView matrixB = new TextView(new ContextThemeWrapper(getContext(), R.style.matrixText), a
        matrixB.setText("Matrix B");
        matrixB.setTextSize(TypedValue.COMPLEX_UNIT_SP,  size: 18); // 设置文本大小
        matrixB.setTextColor(Color.BLACK); // 设置文本颜色
        matrixB.setGravity(Gravity.CENTER); // 设置文本居中
        GridLayout gridLayout2 = createGridLayout(row2, col2,  isA: false);
        container.addView(matrixB);
        container.addView(gridLayout2);

    }
    Button calculate = new Button(getContext());
    calculate.setText("Calculate");
    calculate.setBackground(ContextCompat.getDrawable(getContext(), R.drawable.matrixbutton)); // 设
    calculate.setTextColor(Color.WHITE); // 设置文本颜色
    calculate.setOnClickListener((v) -> { calculate(); });
    container.addView(calculate);
}
```

```
private GridLayout createGridLayout(int rows, int cols, boolean isA) {
    GridLayout gridLayout = new GridLayout(new ContextThemeWrapper(getContext(), R.style.matrixGrid),
    gridLayout.setRowCount(rows);
    gridLayout.setColumnCount(cols);
    gridLayout.setLayoutParams(new LinearLayout.LayoutParams(
            LinearLayout.LayoutParams.MATCH_PARENT,
            LinearLayout.LayoutParams.WRAP_CONTENT
    ));
    int margin = 10;
    for (int i = 0; i < rows * cols; i++) {
        EditText editText = new EditText(getContext());
        GridLayout.LayoutParams params = new GridLayout.LayoutParams();

        params.width = 0;
        params.height = GridLayout.LayoutParams.WRAP_CONTENT;
        params.setMargins(margin, margin, margin, margin);
        params.setGravity(Gravity.FILL_HORIZONTAL);
        params.columnSpec = GridLayout.spec(GridLayout.UNDEFINED,  weight: 1f);
        params.rowSpec = GridLayout.spec(GridLayout.UNDEFINED);
        editText.setLayoutParams(params);

        editText.setGravity(Gravity.CENTER);
        editText.setInputType(InputType.TYPE_CLASS_NUMBER |
                InputType.TYPE_NUMBER_FLAG_DECIMAL |
                InputType.TYPE_NUMBER_FLAG_SIGNED);
        if (isA) {
            matrixAEditTexts.add(editText);
        } else {
            matrixBEditTexts.add(editText);
        }
        gridLayout.addView(editText);
    }

    return gridLayout;
```

Logic of show result

```
private void populateMatrix(double[][] matrix, ArrayList<EditText> editTexts, int rows, int cols) {
    for (int i = 0; i < rows * cols; i++) {
        int row = i / cols;
        int col = i % cols;
        String valueStr = editTexts.get(i).getText().toString();
        double value = valueStr.isEmpty() ? 0.0 : Double.parseDouble(valueStr);
        matrix[row][col] = value;
    }
}

private void calculate() {
    if (matrixAEditTexts.size() > 0) {
        populateMatrix(matrixA, matrixAEditTexts, matrixA.length, matrixA[0].length);
    }
    if (choise == 0) {
        result = Matrix.add(matrixA, matrixB);
    } else if (choise == 1) {
        result = Matrix.minus(matrixA, matrixB);
    } else if (choise == 2) {
        result = Matrix.multiply(matrixA, matrixB);
    } else if (choise == 3) {
        result = Matrix.getTranspose(matrixA);
    } else if (choise == 4) {
        result = Matrix.inverse(matrixA);
    } else if (choise == 5) {
        int rank = Matrix.rankOfMatrix(matrixA);
        result = new double[1][1];
        result[0][0] = rank;
    } else if (choise == 6) {
        double det = Matrix.determinant(matrixA);
        result = new double[1][1];
        result[0][0] = det;
    } else if (choise == 7) {
        result = Matrix.adjugate(matrixA);
    } else {
        double[] m = {matrixA[0][0], matrixA[0][1], matrixA[0][2]};
        result = Matrix.dual(m);
    }
    showResult(result);
}

private void showResult(double[][] result) {
    container.removeAllViews();
    TextView matrixA = new TextView(new ContextThemeWrapper(getContext(), R.style.matrixText),  att
    matrixA.setText("Result");
    matrixA.setTextSize(TypedValue.COMPLEX_UNIT_SP,  size: 18); // 设置文本大小
    matrixA.setTextColor(Color.BLACK); // 设置文本颜色
    matrixA.setGravity(Gravity.CENTER); // 设置文本居中
    GridLayout gridLayout1 = createResult(result.length, result[0].length, result);

    container.addView(matrixA);
    container.addView(gridLayout1);
}
```

```
private GridLayout createResult(int rows, int cols, double[][] result) {
    GridLayout gridLayout = new GridLayout(new ContextThemeWrapper(getContext(), R.style.matrixGrid)
    gridLayout.setRowCount(rows);
    gridLayout.setColumnCount(cols);
    gridLayout.setLayoutParams(new LinearLayout.LayoutParams(
            LinearLayout.LayoutParams.MATCH_PARENT,
            LinearLayout.LayoutParams.WRAP_CONTENT
    ));
    int margin = 10;
    int size = 0;

    if (rows <= 7){
        size = 20;
    } else if (rows == 8){
        size = 16;
    } else if (rows == 9){
        size = 14;
    } else {
        size = 12;
    }

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            TextView textView = new TextView(getContext());
            GridLayout.LayoutParams params = new GridLayout.LayoutParams();

            params.width = 0; // 修改宽度为自适应内容
            params.height = GridLayout.LayoutParams.WRAP_CONTENT; // 高度也设置为自适应内容
            params.setMargins(margin, margin, margin, margin);
            params.setGravity(Gravity.FILL_HORIZONTAL);
            params.columnSpec = GridLayout.spec(GridLayout.UNDEFINED,  weight: 1f);
            params.rowSpec = GridLayout.spec(GridLayout.UNDEFINED);
            textView.setLayoutParams(params);

            textView.setGravity(Gravity.CENTER);
            textView.setTextSize(size);
            textView.setTextColor(Color.BLACK);
            textView.setText(formatNumber(result[i][j])); // 使用自定义的格式化方法

            gridLayout.addView(textView);
        }
    }


    return gridLayout;
}

private String formatNumber(double number) {
    if (number == (long) number) {
```