This is a small link up game. At the main menu, you can go to the game screen, instruction screen and leaderboard screen. At the game screen, you can go to the menu screen and leaderboard screen. At the instruction screen, you can go to the menu screen. At the leaderboard screen you can go to the game screen and menu screen.

The game is just a link up game we normally play. Find a pair of same patterns and link them. The path can't be more than three straight lines. If you suceed to link a pair, you will gain a point; otherwise, you will lose a point.

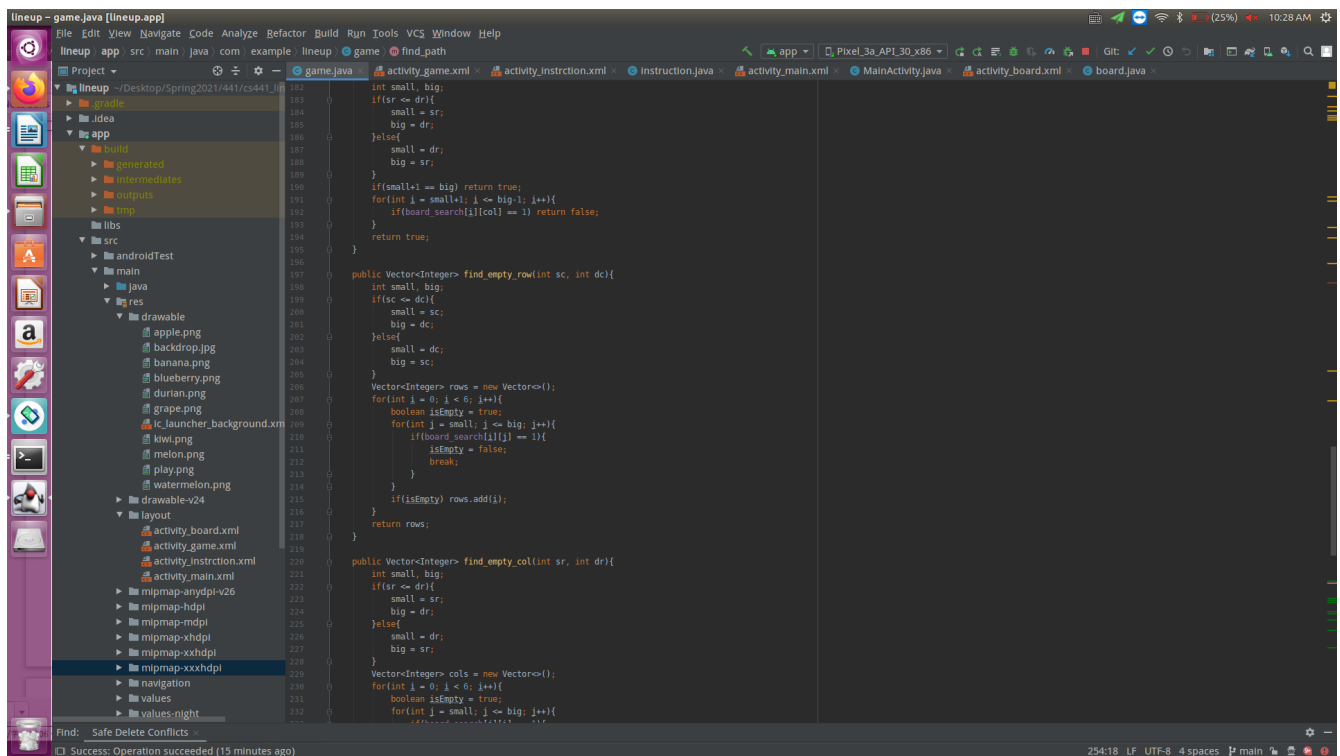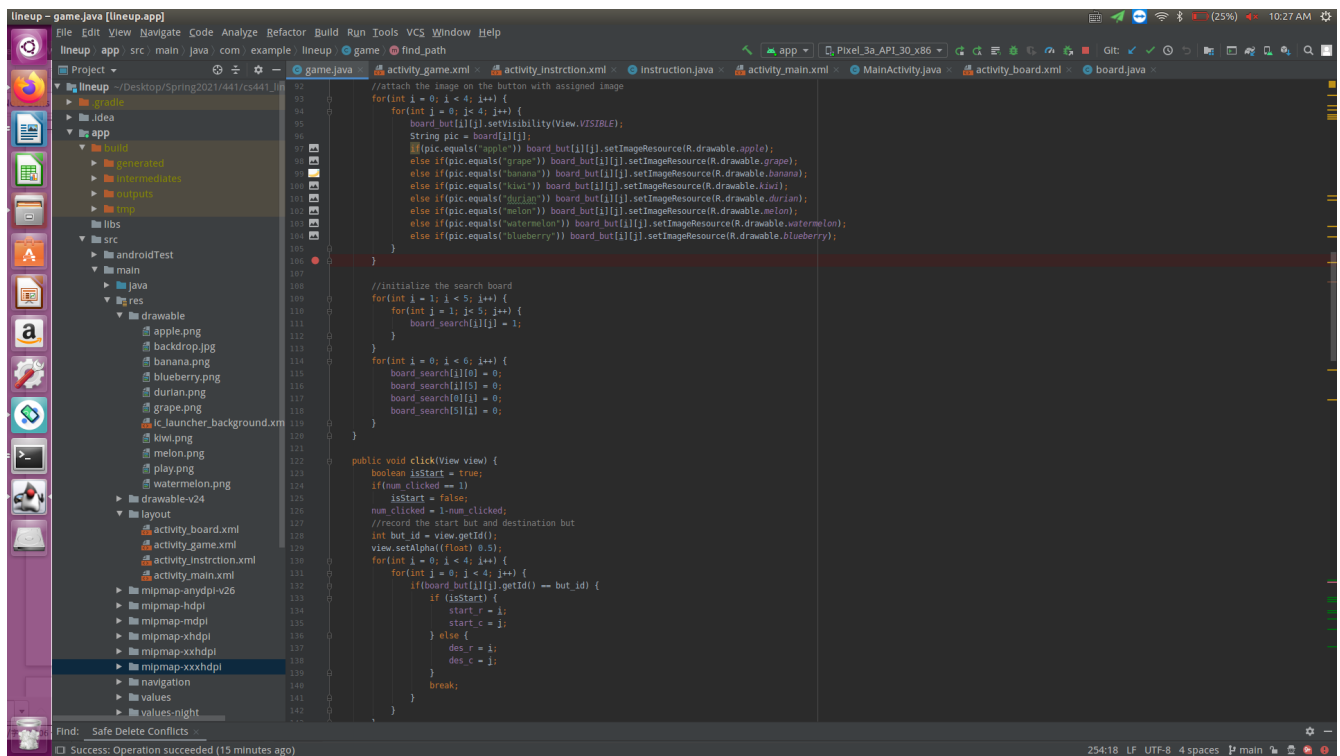The highest three scores will be recorded at the leaderboard.

Pictures:

lineup

score: 1

LEADERBOARD
RESTART
BACK TO MENU

lineup

LEADERBOARD

1st: 8

2nd: 3

3rd: 2

BACK TO GAME
BACK TO MENU

```java
package com.example.lineup;

import ...

public class game extends AppCompatActivity {

    //the matrix storing path obscures
    private int[][] board_search = new int[6][6];
    //the matrix storing different block information
    private String[][] board = new String[4][4];
    //the array stores black kind information
    private String[] pics = {"apple", "grape", "banana", "kiwi", "melon", "blueberry", "watermelon", "durian"};
    private ImageButton[][] board_but = new ImageButton[4][4];
    private ImageButton restart_but;
    private TextView score_board;
    private int score = 0;
    private int highest_score = 0;
    private int start_r = 0, start_c = 0, des_r = 0, des_c = 0, num_clicked = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_game);
        score_board = (TextView) findViewById(R.id.score);
        board_but[0][0] = (ImageButton) findViewById(R.id.b1);
        board_but[0][1] = (ImageButton) findViewById(R.id.b2);
        board_but[0][2] = (ImageButton) findViewById(R.id.b3);
        board_but[0][3] = (ImageButton) findViewById(R.id.b4);
        board_but[1][0] = (ImageButton) findViewById(R.id.b5);
        board_but[1][1] = (ImageButton) findViewById(R.id.b6);
        board_but[1][2] = (ImageButton) findViewById(R.id.b7);
        board_but[1][3] = (ImageButton) findViewById(R.id.b8);
        board_but[2][0] = (ImageButton) findViewById(R.id.b9);
        board_but[2][1] = (ImageButton) findViewById(R.id.b10);
        board_but[2][2] = (ImageButton) findViewById(R.id.b11);
        board_but[2][3] = (ImageButton) findViewById(R.id.b12);
        board_but[3][0] = (ImageButton) findViewById(R.id.b13);
        board_but[3][1] = (ImageButton) findViewById(R.id.b14);
        board_but[3][2] = (ImageButton) findViewById(R.id.b15);
        board_but[3][3] = (ImageButton) findViewById(R.id.b16);
        initializeboard();
    }

    //initialize the game
    public void initializeboard(){
        //reset the selected button
        start_r = 0; start_c = 0; des_r = 0; des_c = 0; num_clicked = 0;
        score = 0;
        score_board.setText(Integer.toString( 0));
        Vector<String> pics_copy = new Vector<>();
```

```java
//attach the image on the button with assigned image
for(int i = 0; i < 4; i++) {
    for(int j = 0; j < 4; j++) {
        board_but[i][j].setVisibility(View.VISIBLE);
        String pic = board[i][j];
        if(pic.equals("apple")) board_but[i][j].setImageResource(R.drawable.apple);
        else if(pic.equals("grape")) board_but[i][j].setImageResource(R.drawable.grape);
        else if(pic.equals("banana")) board_but[i][j].setImageResource(R.drawable.banana);
        else if(pic.equals("kiwi")) board_but[i][j].setImageResource(R.drawable.kiwi);
        else if(pic.equals("durian")) board_but[i][j].setImageResource(R.drawable.durian);
        else if(pic.equals("melon")) board_but[i][j].setImageResource(R.drawable.melon);
        else if(pic.equals("watermelon")) board_but[i][j].setImageResource(R.drawable.watermelon);
        else if(pic.equals("blueberry")) board_but[i][j].setImageResource(R.drawable.blueberry);
    }
}

//initialize the search board
for(int i = 1; i < 5; i++) {
    for(int j = 1; j < 5; j++) {
        board_search[i][j] = 1;
    }
}
for(int i = 0; i < 6; i++) {
    board_search[i][0] = 0;
    board_search[i][5] = 0;
    board_search[0][i] = 0;
    board_search[5][i] = 0;
}
}

public void click(View view) {
    boolean isStart = true;
    if(num_clicked == 1){
        isStart = false;
    num_clicked = 1-num_clicked;
    //record the start but and destination but
    int but_id = view.getId();
    view.setAlpha((float) 0.5);
    for(int i = 0; i < 4; i++){
        for(int j = 0; j < 4; j++) {
            if(board_but[i][j].getId() == but_id) {
                if (isStart){
                    start_r = i;
                    start_c = j;
                } else {
                    des_r = i;
                    des_c = j;
                }
                break;
            }
        }
    }
```

```java
    int small, big;
    if(sr <= dr){
        small = sr;
        big = dr;
    }else{
        small = dr;
        big = sr;
    }
    if(small+1 == big) return true;
    for(int i = small+1; i <= big-1; i++){
        if(board_search[i][col] == 1) return false;
    }
    return true;
}

public Vector<Integer> find_empty_row(int sc, int dc){
    int small, big;
    if(sc <= dc){
        small = sc;
        big = dc;
    }else{
        small = dc;
        big = sc;
    }
    Vector<Integer> rows = new Vector<>();
    for(int i = 0; i < 6; i++){
        boolean isEmpty = true;
        for(int j = small; j <= big; j++){
            if(board_search[i][j] == 1){
                isEmpty = false;
                break;
            }
        }
        if(isEmpty) rows.add(i);
    }
    return rows;
}

public Vector<Integer> find_empty_col(int sr, int dr){
    int small, big;
    if(sr <= dr){
        small = sr;
        big = dr;
    }else{
        small = dr;
        big = sr;
    }
    Vector<Integer> cols = new Vector<>();
    for(int i = 0; i < 6; i++){
        boolean isEmpty = true;
        for(int j = small; j <= big; j++){
```

```java
        //no turns
        if(beginR == endR){
            if(empty_row_between(beginR, beginC, endC)) return true;
        }else if(beginC == endC){
            if(empty_col_between(beginC, beginR, endR)) return true;
        }

        //one turns
        if(empty_row_between(beginR, beginC, endC) && empty_col_between(endC, beginR, endR) && board_search[beginR][endC] != 1) return true;
        else if(empty_row_between(endR, beginC, endC) && empty_col_between(beginC, beginR, endR) && board_search[endR][beginC] != 1) return true;

        //two turns
        Vector<Integer> empty_rows = find_empty_row(beginC, endC);
        Vector<Integer> empty_cols = find_empty_col(beginR, endR);
        if(empty_rows.size() != 0){
            for(int r : empty_rows){
                boolean findcol1, findcol2;
                findcol1 = empty_col_between(beginC, r, beginR);
                findcol2 = empty_col_between(endC, r, endR);
                if(findcol1 && findcol2) return true;
            }
        }
        if(empty_cols.size() != 0){
            for(int c : empty_cols){
                boolean findrow1, findrow2;
                findrow1 = empty_row_between(beginR, c, beginC);
                findrow2 = empty_row_between(endR, c, endC);
                if(findrow1 && findrow2) return true;
            }
        }
        return false;
    }

    public void restart_game(View view) { initializeboard(); }

    public void switch_to_menu(View view) {
        Intent act_action = new Intent( packageContext: this, MainActivity.class);
        startActivity(act_action);
    }

    public void switch_to_leaderboard(View view){
        // We need an Editor object to make preference changes.
        // All objects are from android.context.Context
        SharedPreferences settings = getSharedPreferences( name: "myPref", mode: 0);
        SharedPreferences.Editor editor = settings.edit();
        editor.putInt("key", highest_score);
        editor.apply();

        Intent act_action = new Intent( packageContext: this, board.class);
        startActivity(act_action);
        finish();
    }
```



```java
        int score, b1, b2, b3;

        @Override
        protected void onCreate(Bundle savedInstanceState) {
            super.onCreate(savedInstanceState);
            setContentView(R.layout.activity_board);

            button_menu = (Button) findViewById(R.id.menu1);
            first = (TextView) findViewById(R.id.best1);
            second = (TextView) findViewById(R.id.best2);
            third = (TextView) findViewById(R.id.best3);

            SharedPreferences prefs = getSharedPreferences( name: "myPref", Context.MODE_PRIVATE);
            score = prefs.getInt( key: "key", defValue: 0); //0 is the default value
            b1 = prefs.getInt( key: "best1", defValue: 0);
            b2 = prefs.getInt( key: "best2", defValue: 0);
            b3 = prefs.getInt( key: "best3", defValue: 0);

            if(score > b1){
                b3 = b2;
                b2 = b1;
                b1 = score;
            }else if(score < b1 && score > b2){
                b3 = b2;
                b2 = score;
            }else if(score < b2 && score > b3){
                b3 = score;
            }
            SharedPreferences.Editor editor = prefs.edit();
            editor.putInt("best1", b1);
            editor.putInt("best2", b2);
            editor.putInt("best3", b3);
            editor.apply();

            first.setText( text: "1st: " + Integer.toString(b1));
            second.setText( text: "2nd: " + Integer.toString(b2));
            third.setText( text: "3rd: " + Integer.toString(b3));
        }

        public void switch_to_game(View view){
            Intent act_action = new Intent( packageContext: this, game.class);
            startActivity(act_action);
        }

        public void switch_to_menu(View view){
            Intent act_action = new Intent( packageContext: this, MainActivity.class);
            startActivity(act_action);
        }
    }
```

Git log screen shot:

Here is Reference Link I used to write code:

https://www.youtube.com/watch?v=GtxVILjLcw8

https://www.youtube.com/watch?v=_cV7cgQFDo0