

فئة سنست عرض ال مدونة، هذه في

4. بمساعدة ال مدونة هذه كاتابة تمت

مقدمة

تطبيقات في مخصصة رسم واجهة لتنفيذ شامل مثال وهي DrawActivity فئة سنست عرض ال مدونة، هذه في لتحقيق مَعًا عملها لكي فية مفصل شرح مع ال مستخدمة، وال خوارزميات مكون لكل بتفكيك سنقوم .
المطلوبة. ال وظائف

المحتويات جدول

على عامة نظرة
تهيئة
ال صورة عمليات مع ال
إدارة
الأحداث مع ال
وال إعادة التراجع ميزات
تخصيص
التاريخ إدارة
الخلاصة

على عامة نظرة

واجهة إدارة عن مسؤولية الفئة هذه بتطويره. نقوم ال ذي الرسم تطبيقي في رئيسية فئة هي DrawActivity القسم، هذا في الرسوميات. وتعديل لإنشاء لل مستخدم ال لازمة الأدوات وتوفر بالرسم الخاصة ال مستخدم ال مستخدم. مع تفاعلها ولكي فية DrawActivity لوظيفة ال رئيسية ال مكونات على عامة نظرة سنلقي

الرئيسية المكونات

1. اللوحة Canvas:

- في Canvas كائن بواسطة تمثيلها يتم والخطوط. الأشكال رسم فيها يتم التي المنطقة هي هذه الكود.
- رسومات، لإنشاء الفأرة أو اللمس باستخدام اللوحة مع التفاعل للخدمة يمكن.

2. الأدوات:

- الخطوط مثل مختلف أشكال برسم للخدمة تسمح التي الأدوات من مجموعة التطبيق يتضمن والستطيلات. والدوائر
- LineTool, CircleTool, RectangleTool. مثل به، خاصة فئة بواسطة أداة كل تمثيل يتم.

3. الألوان والفرش:

- الرسومات، لتخصيص المختلفة والفرش الألوان اختياري للخدمة يمكن.
- Paint فئة عبر والفرش الألوان إدارة يتم.

4. الرسومات حفظ:

- مثل مختلف ملفات صيغة في إنشائها قام التي الرسومات حفظ للخدمة يمكن.
- في الصور مع الواجهة مكتبات عبر الوظيفة هذه تنفيذ يتم.

الخدمة تفاعل

- اللوحة. على بالرسم يبدأ ثم الأدوات شريط من أداة باختيار الخدمة يقوم: الرسم
- مثل التعديل أدوات استخدام أو الألوان تغيير طريقي عن الرسومات تعديل للخدمة يمكن: التعديل المسح.
- الآخرين. مع مشاركته أو الجهاز في العمل حفظ للخدمة يمكن الرسم، من الانتهاء بعد: الحفظ

الكود على مثال

```
public class DrawActivity extends AppCompatActivity {  
    private CanvasView canvasView;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_draw);  
    }  
}
```

```

        canvasView = findViewById(R.id.canvas);
        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
    }

    public void clearCanvas() {
        canvasView.clearCanvas();
    }

    public void saveDrawing() {
        Bitmap bitmap = canvasView.getBitmap();
        // Save bitmap to file or share it
    }
}

```

اللوحة مسح مثل وظائف تنفيذي تم وال أدوات. اللوحة وإعداد DrawActivity تهيئة يتم لكي نرى المثل، هذا في محدة. طرق عبر الرسم وحفظ

فهم خل ال من الرسومات. لإنشاء ال مستخدم مع ها ي تفاعل ال تي الئيسي الواجهة هي DrawActivity ال خل ال ال استخدا م. وسهل قوي رسم تطبي ق تطوي ر ي مكن ن ا تفاعل ها، ولكي في الئيسي ال مكونات

المكونات مع والتفاعل الصور، اق تصاص الرسم، عمل يات مع ي تعامل ال ذي الئيسي ال نشاط هي DrawActivity ال ترا جع، الرسم، من ال مستخدم ت مكن مستخدم واجهة يوفر الصور. ورفع `onOptionsItemSelected` الأجزاء مثل الأخرى الصور. ومع الة التنفي ذ، إعادة

```

public class DrawActivity extends Activity implements View.OnClickListener {
    // (fragments)
    public static final int CAMERA_RESULT = 1;
    public static final int CROP_RESULT = 2;
    public static final int DRAW_FRAGMENT = 0;
    public static final int RECOG_FRAGMENT = 1;
    public static final int RESULT_FRAGMENT = 2;
    public static final int WAIT_FRAGMENT = 3;
    public static final int MATERIAL_RESULT = 4;
    public static final String RESULT_JSON = "resultJson";
    public static final int INIT_FLOWER_ID = R.drawable.flower_b;
    public static final int LOGOUT = 0;
}

```

```

public static final int IMAGE_RESULT = 0;

//
String baseUrl;
DrawView drawView;
Bitmap originImg;
public static DrawActivity instance;
View dir, clear, cameraView, materialView, scale;
ImageView undoView, redoView;
View upload;
String cropPath;
Tooltip toolTip;
int curFragmentId = -1;
int serverId = -1;
private Bitmap resultBitmap;
private RadioGroup radioGroup;
Fragment curFragment;
int curDrawMode;
RadioButton drawBackBtn;
private Activity cxt;
Uri curPicUri;
}

```

الأنشطة الرئيسية

الأنشطة حيّاة دورة استدعاء يتم، تطابق في جديد نشاط إنشاء عند الأنشطة. إنشاء عند استدعاء أي تم دالة أول وهي onCreate(). تُعرف المرحلة هذه التهيئة. بمرحلة تبدأ والتي الأولى. المتغيرات وتعني النسخة المستخدمة واجهة تهيئة مثل الأساسية المهام تنفيذ يتم الدالة، هذه في

```

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

```

```
//
}
```

المهام تنفذ لضمان الأصلية الدالة يستدعي `super.onCreate(savedInstanceState)` أعلاه: الكود في الذي التخطيط ملف يعين `setContentView(R.layout.activity_main)` النظام. بهذا يقوم التي الأساسية `activity_main.xml` ملف استخدام يتم المثال، هذا في المستخدم. واجهة لعرض استخدام سيتم

`onResume()` و `onStart()` مثل النشاط حياة دورة في الأخرى الدوال استدعاء يتم، `onCreate()` تنفذ بعد المستخدم. مع للفاعل وجاهزًا مرئيًا أصبح النشاط أن تعني والتي

الأولية، الصور تحميل العرض، إعداد مثل المختلطة التهئية عمليات تنفذ يتم، `onStart()` إنشاء عند الأحداث. مستمعي وتهئية

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    instance = this;
    cxt = this;
    cropPath = PathUtils.getCropPath();
    setContentView(R.layout.draw_layout);
    findView();
    setSize();
    initOriginImage();
    tooltip = new Tooltip(this);
    initUndoRedoEnable();
    setIp();
    initDrawmode();
}
```

ترجمة يتم لا `onCreate()` تطبق في طريقة من جزء وهو `onCreate()` بلغة مكتوب المقدم الكود ملاحظة: بشكل التطبيق عمل لضمان هي كما تبقى أن يجب الأسماء هذه أن حيث نفسه، الكود أو المتغيرات أو الدوال أسماء صحيحة.

```
onCreate()
```

النشاط في المستخدم `onCreate()` العرض لتهئية الطريقة هذه تستخدم

```
private void findView() {
    drawView = findViewById(R.id.drawView);
    undoView = findViewById(R.id.undo);
}
```

```

redoView = findViewById(R.id.redo);
scale = findViewById(R.id.scale);
upload = findViewById(R.id.upload);
clear = findViewById(R.id.clear);
dir = findViewById(R.id.dir);
materialView = findViewById(R.id.material);
cameraView = findViewById(R.id.camera);

dir.setOnClickListener(this);
materialView.setOnClickListener(this);
undoView.setOnClickListener(this);
scale.setOnClickListener(this);
redoView.setOnClickListener(this);
clear.setOnClickListener(this);
cameraView.setOnClickListener(this);
upload.setOnClickListener(this);
initRadio();
}

```

المتاليّة: للعنصر `dir` `materialView` `undoView` `scale` `redoView` `clear` `cameraView` `upload` `OnClickListener` للقر مسّمعين تعيّن تم

الراديوي. لتهيئة `initRadio()` الدالة استدعاء تم ثم

الرسـم. عرض حجم تعيّن

```

private void setSize() {
    setSizeByResourceSize();
    setViewSize(drawView);
}

```

إلى: أعلاه الكود ترجمة تمت

```

private void setSize() {
    setSizeByResourceSize();
    setViewSize(drawView);
}

```

ترجمته. يتم ولإعالمية برمجة لغة وهي `AndroidManifest.xml` بلغة مكتوب لأنه الكود في تغييري وجد لا

```
private void setSizeByResourceSize() {
    int width = getResources().getDimensionPixelSize(R.dimen.draw_width);
    int height = getResources().getDimensionPixelSize(R.dimen.draw_height);
    App.drawWidth = width;
    App.drawHeight = height;
}
```

إلى: أعلاه الكود ترجمة تمت

```
private void setSizeByResourceSize() {
    int width = getResources().getDimensionPixelSize(R.dimen.draw_width);
    int height = getResources().getDimensionPixelSize(R.dimen.draw_height);
    App.drawWidth = width;
    App.drawHeight = height;
}
```

`R.dimen.draw_width` الموارد مل في المحددة القيم على بناء والارتفاع العرض حجم تعييين يتم الدالة، هذه في `App.drawWidth` و `App.drawHeight` العامة المتغيرات في القيم هذه تعييين يتم ثم، `R.dimen.draw_height`

```
private void setViewSize(View v) {
    ViewGroup.LayoutParams lp = v.getLayoutParams();
    lp.width = App.drawWidth;
    lp.height = App.drawHeight;
    v.setLayoutParams(lp);
}
```

XXXXXXXXXXXXXXXXXXXX

لرسم. استخدمها سيتم التي الأولية الصورة تحمّل

```
private void initOriginImage() {
    Bitmap bitmap = BitmapFactory.decodeResource(getResources(), INIT_FLOWER_ID);
    String imgPath = PathUtils.getCameraPath();
    BitmapUtils.saveBitmapToPath(bitmap, imgPath);
    Uri uri1 = Uri.fromFile(new File(imgPath));
    setImageByUri(uri1);
}
```

العربية: إلى الكود ترجمة

```
private void initOriginImage() {  
    //  
    Bitmap bitmap = BitmapFactory.decodeResource(getResources(), INIT_FLOWER_ID);  
  
    //  
    String imgPath = PathUtils.getCameraPath();  
  
    //  
    BitmapUtils.saveBitmapToPath(bitmap, imgPath);  
  
    // Uri  
    Uri uri1 = Uri.fromFile(new File(imgPath));  
  
    // Uri  
    setImageByUri(uri1);  
}
```

الترجمة البرمجة. في تترجم لا والمتغيرات الدوال أسماء لأن الإنجليزية باللغة هو كما يبق الكود ملاحظة:
فقط. للتوضيح هي هنا

الصور عمليات معالجة

واقتصاص، عبر الصور تعيّن مثل المختلطة، الصور عمليات بمعالجة الصور النشاط يقوم
الصور. كالمرسومة الصور وحفظ الصور.

الصور

للرسم. وإعداده المحدد الصور من الصورة بتحميل يقوم

```
private void setImageByUri(final Uri uri) {  
    new Handler().postDelayed(new Runnable() {  
        @Override  
        public void run() {  
            curPicUri = uri;  
            Bitmap bitmap = null;  
        }  
    }, 100);  
}
```



```

        try {
            if (uri != null) {
                bitmap = BitmapUtils.getBitmapByUri(DrawActivity.this, uri);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private void setImageByUri(final Uri uri) {
        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                curPicUri = uri;
                Bitmap bitmap = null;
                try {
                    if (uri != null) {
                        bitmap = BitmapUtils.getBitmapByUri(DrawActivity.this, uri);
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                }

                int originW = bitmap.getWidth();
                int originH = bitmap.getHeight();
                if (originW != App.drawWidth || originH != App.drawHeight) {
                    float originRadio = originW * 1.0f / originH;
                    float radio = App.drawWidth * 1.0f / App.drawHeight;
                    if (Math.abs(originRadio - radio) < 0.01) {
                        Bitmap originBm = bitmap;
                        bitmap = Bitmap.createScaledBitmap(originBm, App.drawWidth, App.drawHeight, false);
                        originBm.recycle();
                    } else {
                        cropIt(uri);
                        return;
                    }
                }
            }
        }, 100);

        ImageLoader imageLoader = ImageLoader.getInstance();

```

```

imageLoader.addOrReplaceToMemoryCache("origin", bitmap);
originImg = bitmap;
serverId = -1;

```

العربية: إلى الكود مترجمة

```

int originW = bitmap.getWidth();
int originH = bitmap.getHeight();
if (originW != App.drawWidth || originH != App.drawHeight) {
    float originRadio = originW * 1.0f / originH;
    float radio = App.drawWidth * 1.0f / App.drawHeight;
    if (Math.abs(originRadio - radio) < 0.01) {
        Bitmap originBm = bitmap;
        bitmap = Bitmap.createScaledBitmap(originBm, App.drawWidth, App.drawHeight, false);
        originBm.recycle();
    } else {
        cropIt(uri);
        return;
    }
}
ImageLoader imageLoader = ImageLoader.getInstance();
imageLoader.addOrReplaceToMemoryCache("origin", bitmap);
originImg = bitmap;
serverId = -1;

```

عادةً. مترجمتها يتم ولا تقنية أسماء لها هي كما والوظائف المترجمات أسماء على الحفظ تم ملاحظة:

```

drawView.setSrcBitmap(originImg);
showDrawFragment(App.ALL_INFO);
curDrawMode = App.DRAW_FORE;
}
}, 500);
}

```

drawView الرسم لعنصر مصدريّة لصورة originImg الأصليّة الصورة تعيّن تم
App.ALL_INFO الملعومات جميع مع الرسم جزء لعرض showDrawFragment الدالة استدعاء تم
App.DRAW_FORE الأمامي الرسم وضع إلى curDrawMode الحالي الرسم وضع تعيّن تم
ثانية. مللي 500 قدره تأخير بعد الأوامر هذه تنفيذي تم

الخطوة الخامسة: حفظ الصورة

الخطوة السادسة: اقتصاص الصورة

```
public void cropIt(Uri uri) {
    Crop.startPhotoCrop(this, uri, cropPath, CROP_RESULT);
}
```

الخطوة السابعة: حفظ الصورة

الخطوة الثامنة: تحميل الصورة وإرسالها إلى الخادم.

```
public void saveBitmap() {
    Bitmap handBitmap = drawView.getHandBitmap();
    Bitmap originBitmap = drawView.getSrcBitmap();
    saveBitmapToFileAndUpload(handBitmap, originBitmap);
}
```

الخطوة التاسعة: تحميل الصورة وإرسالها إلى الخادم.

الخطوة العاشرة: تحميل الصورة وإرسالها إلى الخادم.

```
private void saveBitmapToFileAndUpload(Bitmap handBitmap, Bitmap originBitmap) {
    final String originPath = PathUtils.getOriginPath();
    BitmapUtils.saveBitmapToPath(originBitmap, originPath);
    final String handPath = PathUtils.getHandPath();
    BitmapUtils.saveBitmapToPath(handBitmap, handPath);
    new AsyncTask<Void, Void, Void>() {
        boolean res;
        Bitmap foreBitmap;
        Bitmap backBitmap;

        @Override
        protected Void doInBackground(Void... params) {
            // ...
        }
    }.execute();
}
```

```

@Override
protected void onPreExecute() {
    super.onPreExecute();
    showWaitFragment();
}

@Override
protected Void doInBackground(Void... params) {
    try {
        if (baseUrl == null) {
            throw new Exception("baseUrl is null");
        }

        String jsonRes = UploadImage.upload(baseUrl, serverId, Web.STATUS_CONTINUE, originPath, handPath, n
        getJsonData(jsonRes);
        res = true;
    } catch (Exception e) {
        res = false;
        e.printStackTrace();
    }
    return null;
}

private void getJsonData(String jsonRes) throws Exception {
    JSONObject json = new JSONObject(jsonRes);
    if (serverId == -1) {
        serverId = json.getInt(Web.ID);
    }

    String foreUrl = json.getString(Web.FORE);
    String backUrl = json.getString(Web.BACK);
    String resultUrl = json.getString(Web.RESULT);
    foreBitmap = Web.getBitmapFromUrlByStream1(foreUrl, 0);
    backBitmap = Web.getBitmapFromUrlByStream1(backUrl, 0);
    resultBitmap = Web.getBitmapFromUrlByStream1(resultUrl, 0);
}

@Override
protected void onPostExecute(Void aVoid) {

```

```

super.onPostExecute(aVoid);
if (res) {
    showRecogFragment(foreBitmap, backBitmap);
} else {
    Utils.toast(DrawActivity.this, R.string.server_error);
    recogNo();
}
}

}.execute();
}

```

إدارة `FragmentManager`

والتعريف للرسم مثل لل تطبيق، المخطلة الحالات مع لل تعامل المخطلة `FragmentManager` إدارة `FragmentManager` تدوير والانتظار.

`FragmentManager` `FragmentManager` الرسم جزء يعرض.

```

private void showDrawFragment(int infoId) {
    curFragmentId = DRAW_FRAGMENT;
    curFragment = new DrawFragment(infoId);
    showFragment(curFragment);
}

```

الترجمة:

```

private void showDrawFragment(int infoId) {
    curFragmentId = DRAW_FRAGMENT;
    curFragment = new DrawFragment(infoId);
    showFragment(curFragment);
}

```

`infoId` يسمى `int` النوع من معاملة تأخذ `showDrawFragment` تسمى خاصة دالة تعريف يتم أعلاه، الكود في النوع من جديد كائن إنشاء يتم ثم `curFragmentId` لل متغير `DRAW_FRAGMENT` قيمة تعيّن يتم الدالة، داخل

دالة استدعاء يتم أخيرًا، `curFragment` للامتغير تعييه ويتم كدعامل `infoId` باستخدام `DrawFragment` كدعامل. `curFragment` الكائن تمريه مع `showFragment`

الان تظار جزء عرض

```
private void showWaitFragment() {
    curFragmentId = WAIT_FRAGMENT;
    showFragment(new WaitFragment());
}
```

ترجمة:

```
private void showWaitFragment() {
    curFragmentId = WAIT_FRAGMENT;
    showFragment(new WaitFragment());
}
```

باللغة ودوال متغيرات أسماء على يحتوي لأنه ترجمته يتم لم البرمجية الكتلة في الموجود الكود ملاحظة:
البرمجة. في هي كما تبقى ما عادةً والتي الإنجليزية،

المحدد. بال- الحالي بال- استبدل

```
private void showFragment(Fragment fragment) {
    FragmentTransaction trans = getFragmentManager().beginTransaction();
    trans.replace(R.id.rightLayout, fragment);
    trans.commit();
}
```

العربية: إلى الكود ترجمة

```
private void showFragment(Fragment fragment) {
    //      Fragment
    FragmentTransaction trans = getFragmentManager().beginTransaction();

    //      Fragment      Layout      Fragment
    trans.replace(R.id.rightLayout, fragment);
}
```

}

commit. باستخدام العملية تأكيد ثم الاستبدال، لتنفذ

الأحداث مع الـ جة

الوقایم، واختیار الأضرار علی النقر مثلاً المستخدم، تفاعلات مختلف مع □□□□□□□□ تتعامل

□ □ □ □ □ □ □ □ □ □ □ □ □ □

المختلقة. العرض واجهات على النقر أحداث مع التامل

```
@Override
public void onClick(View v) {
    int id = v.getId();
    if (id == R.id.drawOk) {
        if (drawView.isDrawFinish()) {
            saveBitmap();
        } else {
            Utils.alertDialog(this, R.string.please_draw_finish);
        }
    } else if (id == R.id.recogOk) {
        recogOk();
    } else if (id == R.id.recogNo) {
        recogNo();
    } else if (id == R.id.dir) {
        Utils.getGalleryPhoto(this, IMAGE_RESULT);
    } else if (id == R.id.clear) {
        clearEverything();
    } else if (id == R.id.undo) {
        drawView.undo();
    } else if (id == R.id.redo) {
        drawView.redo();
    }
}
```

```

        drawView.redo();
    } else if (id == R.id.camera) {
        Utils.takePhoto(cxt, CAMERA_RESULT);
    } else if (id == R.id.material) {
        goMaterial();
    } else if (id == R.id.upload) {
        com.lzw.commons.Utils.goActivity(cxt, PhotoActivity.class);
    } else if (id == R.id.scale) {
        cropIt(curPicUri);
    }
}

```

□□□□□□□□□□□□□□□□ □□□□□□□□□□, □□□ □□□□□□□□□□, □□□□□□ □□□□□□
 اقتصاصها. أو الصور اختياريًا مثل الأخرى، الأنشطة نتائج معالجة

```

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    if (resultCode != RESULT_CANCELED) {
        Uri uri;
        switch (requestCode) {
            case IMAGE_RESULT:
                if (data != null) {
                    setImageByUri(data.getData());
                }
                break;
            case CAMERA_RESULT:
                setImageByUri(Utils.getCameraUri());
                break;
            case CROP_RESULT:
                uri = Uri.fromFile(new File(cropPath));
                setImageByUri(uri);
                break;
            case MATERIAL_RESULT:
                setImageByUri(data.getData());
        }
    }
}

```

والإعادة التراجع ميزات

الرسومية. للعمليات والإعادة التراجع وظيفة يوفر

مكتبة Android

. المكتبة توفر الـ `History.Callback` الـ استدعاء دوال تعيين خلال من والإعادة التراجع وظائف بتهيئة يقوم

```
void initUndoRedoEnable() {
    drawView.history.setCallBack(new History.Callback() {
        @Override
        public void onHistoryChanged() {
            setUndoRedoEnable();
            if (curFragmentId != DRAW_FRAGMENT) {
                showDrawFragment(curDrawMode);
            }
        }
    });
}
```

العربية: إلى الكود ترقية

```
void initUndoRedoEnable() {
    drawView.history.setCallBack(new History.Callback() {
        @Override
        public void onHistoryChanged() {
            setUndoRedoEnable();
            if (curFragmentId != DRAW_FRAGMENT) {
                showDrawFragment(curDrawMode);
            }
        }
    });
}
```

الكود: شرح

. المكتبة توفر الـ `History.Callback` الـ استدعاء دوال تعيين خلال من والإعادة التراجع وظائف بتهيئة يقوم

في التغيرات لتتبع `undoView.isEnabled` نداء رد تعييين يتم: `undoView.isEnabled = false`.
 التسجيل. `undoView.isEnabled`: في تعييين حدوث عند الدالة هذه استدعاء يتم: `undoView.isEnabled = true`.
 التسجيل. حالة على بناء وإعادة التراجع أضرار تعطيل أو تفعيل: `undoView.isEnabled = true`.
 التسجيل. الرسم جزء هو الحالي الجزء يمكن لم إذا: `undoView.isEnabled = true`؛ `undoView.isEnabled = false`.
 التسجيل. للرسم الحالي الوضع مع الرسم جزء عرض يتم.

ترجمته. يتم ولا برمجية أسماء لأنها هي كما والامتغيرات الدوال أسماء على الحفاظ تم ملاحظة:

```
void setUndoRedoEnable() {
    redoView.setEnabled(drawView.history.canRedo());
    undoView.setEnabled(drawView.history.canUndo());
}
```

تخصيص `DrawView`

للمستخدمين تسمح تفاعلية مستخدم واجهة لإنشاء `DrawView` تخصيص ليعرفية سن تعلم المنشور، هذا في ذلك. لتتبع `DrawView` سنستخدم الشاشة. على بالرسم

`DrawView` `DrawView` قالب باستخدام `DrawView` في جديد مشروع بإنشاء قم أولاً، جديد مشروع إنشاء: 1 الخطوة برمجة. كلغة `DrawView` واخترت تري، كما المشروع بتسمية قم. `DrawView`

التي `DrawView` فئة بتعريف وقم `DrawView.swift` باسم جديد ملف بإنشاء قم `DrawView` إنشاء: 2 الخطوة `UIView` من ترث.

```
import UIKit

class DrawView: UIView {
    var lines: [Line] = []
    var lastPoint: CGPoint!

    override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?) {
        lastPoint = touches.first!.location(in: self)
    }
}
```

```

override func touchesMoved(_ touches: Set<UITouch>, with event: UIEvent?) {
    let newPoint = touches.first!.location(in: self)
    lines.append(Line(start: lastPoint, end: newPoint))
    lastPoint = newPoint
    self.setNeedsDisplay()
}

override func draw(_ rect: CGRect) {
    let context = UIGraphicsGetCurrentContext()
    context?.setStrokeColor(UIColor.black.cgColor)
    context?.setLineWidth(2)
    context?.setLineCap(.round)

    for line in lines {
        context?.move(to: line.start)
        context?.addLine(to: line.end)
        context?.strokePath()
    }
}

struct Line {
    var start: CGPoint
    var end: CGPoint
}

```

قم الواجهة. إلى UIView بإضافة وقم Main.storyboard افتح الواجهة إلى DrawView إضافة 3: الخطوة
 قسّم في DrawView إلى UIView بهذه الخاصية الفئة بتغيّر.

على الرسم يمكنك الآن، فعله. جهّز أو محاكاة جهاز على التطبيق بتغيير قم التطبيق بتشغيل 4: الخطوة
 الشاشة. على إصبعك تحريك عن الشاشة

سمكه. أو الخط لون تغيّر مثلاً خصائص بإضافة DrawView تغيّر خصائص يمكنك DrawView تغيّر خصائص 5: الخطوة
 يلي: كما الكود تعديل يمكنك الخط، لون تغيّر خصائص لإضافة المثال، سبيل على

```

class DrawView: UIView {
    var lines: [Line] = []
    var lastPoint: CGPoint!
    var strokeColor: UIColor = .black
    var strokeWidth: CGFloat = 2

    override func touchesBegan(_ touches: Set<UITouch>, with event: UIEvent?) {
        lastPoint = touches.first!.location(in: self)
    }

    override func touchesMoved(_ touches: Set<UITouch>, with event: UIEvent?) {
        let newPoint = touches.first!.location(in: self)
        lines.append(Line(start: lastPoint, end: newPoint))
        lastPoint = newPoint
        self.setNeedsDisplay()
    }

    override func draw(_ rect: CGRect) {
        let context = UIGraphicsGetCurrentContext()
        context?.setStrokeColor(strokeColor.cgColor)
        context?.setLineWidth(strokeWidth)
        context?.setLineCap(.round)

        for line in lines {
            context?.move(to: line.start)
            context?.addLine(to: line.end)
            context?.strokePath()
        }
    }
}

```

strokeColor و strokeWidth خصائص تعديّل طريق عن وسمكه الخط لون تغيري يريمكنك الآن،

على بالرسم للخدمة يسمّح مخصص DrawView إنشاء كيفية تعملنا الممنشور، هذا في الخلصة احتياجك. حسب والوظائف الخصائص من المزيّد بإضافة الوظيفة هذه توسيع يمكنك الاشاعة.

والتحجيم. اللمس، أحداث الرسم، عمليّات لمعالجة يستخدم مخصص عرض هو DrawView

والآن نكتب كودنا للتعامل مع الأحداث

والآن نكتب كودنا للتعامل مع الأحداث

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    if (!scaleMode) {
        handleDrawTouchEvent(event);
    } else {
        handleScaleTouchEvent(event);
    }
    return true;
}
```

ترجمة:

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    if (!scaleMode) {
        handleDrawTouchEvent(event);
    } else {
        handleScaleTouchEvent(event);
    }
    return true;
}
```

ترجمته. يتم الآن التعامل مع الأحداث، كود لأنّه تغيّره يتم لم البرمجيّة الكتلة في الموجود الكود ملاحظة:

```
private void handleDrawTouchEvent(MotionEvent event) {
    int action = event.getAction();
    float x = event.getX();
    float y = event.getY();
    if (action == MotionEvent.ACTION_DOWN) {
        path.moveTo(x, y);
    } else if (action == MotionEvent.ACTION_MOVE) {
        path.quadTo(preX, preY, x, y);
    } else if (action == MotionEvent.ACTION_UP) {
        Matrix matrix1 = new Matrix();
    }
}
```

```

        matrix.invert(matrix1);
        path.transform(matrix1);
        paint.setStrokeWidth(strokeWidth * 1.0f / totalRatio);
        history.addToStack(path, paint);
        cacheCanvas.drawPath(path, paint);
        paint.setStrokeWidth(strokeWidth);
        path.reset();
    }
    setPrev(event);
    invalidate();
}

```

الترجمة:

```

private void handleDrawTouchEvent(MotionEvent event) {
    int action = event.getAction();
    float x = event.getX();
    float y = event.getY();
    if (action == MotionEvent.ACTION_DOWN) {
        path.moveTo(x, y);
    } else if (action == MotionEvent.ACTION_MOVE) {
        path.quadTo(preX, preY, x, y);
    } else if (action == MotionEvent.ACTION_UP) {
        Matrix matrix1 = new Matrix();
        matrix.invert(matrix1);
        path.transform(matrix1);
        paint.setStrokeWidth(strokeWidth * 1.0f / totalRatio);
        history.addToStack(path, paint);
        cacheCanvas.drawPath(path, paint);
        paint.setStrokeWidth(strokeWidth);
        path.reset();
    }
    setPrev(event);
    invalidate();
}

```

الشرح:

- `handleDrawTouchEvent`: اللمس أحداث ثعالج دالة لرسم.
- `MotionEvent`: حدث يمثّل اللمس.
- `action`: الرفع، الحركة، الضغط، مثّل الحدث بنوع يحدث بفظ
- `x, y`: اللمس. نقطة إحداثيات
- `path.moveTo(x, y)`: النقطة من جديد مساري بدأ.
- `path.quadTo(preX, preY, x, y)`: السابقة النقطة بين منحني يرسم `preX, preY` والنقطة `x, y` الحالية `x, y`.
- `Matrix`: الإحداثيات. لتحويلليست خدم
- `paint.setStrokeWidth`: العرض يحدد
- `history.saveToStack`: التاريخ. في والفرشاة المساري حفظ
- `cacheCanvas.drawPath`: المؤقتة. الرسم لوحة على المساري رسم
- `path.reset()`: المسار. تعيّن يعيد
- `setPrev(event)`: السابقة. الإحداثيات ي حفظ
- `invalidate()`: الواجهة. رسم يعيد

```
private void handleScaleTouchEvent(MotionEvent event) {
    switch (event.getActionMasked()) {
        case MotionEvent.ACTION_POINTER_DOWN:
            lastFingerDist = calFingerDistance(event);
            break;
        case MotionEvent.ACTION_MOVE:
            if (event.getPointerCount() == 1) {
                handleMove(event);
            } else if (event.getPointerCount() == 2) {
                handleZoom(event);
            }
            break;
        case MotionEvent.ACTION_UP:
        case MotionEvent.ACTION_POINTER_UP:
            lastMoveX = -1;
            lastMoveY = -1;
            break;
        default:
            break;
    }
}
```

```
}
```

العربية: إلى الكود ترحمة

```
private void handleScaleTouchEvent(MotionEvent event) {
    switch (event.getActionMasked()) {
        case MotionEvent.ACTION_POINTER_DOWN:
            lastFingerDist = calFingerDistance(event);
            break;
        case MotionEvent.ACTION_MOVE:
            if (event.getPointerCount() == 1) {
                handleMove(event);
            } else if (event.getPointerCount() == 2) {
                handleZoom(event);
            }
            break;
        case MotionEvent.ACTION_UP:
        case MotionEvent.ACTION_POINTER_UP:
            lastMoveX = -1;
            lastMoveY = -1;
            break;
        default:
            break;
    }
}
```

ترجمته. يتم ولا تقنيّة أسماء لأنها هي كذا والممتغي رات الدوال أسماء على الحفاظ تم ملاحظه:

```
private void handleMove(MotionEvent event) {
    float moveX = event.getX();
    float moveY = event.getY();
    if (lastMoveX == -1 && lastMoveY == -1) {
        lastMoveX = moveX;
        lastMoveY = moveY;
    }

    moveDistX = (int) (moveX - lastMoveX);
    moveDistY = (int) (moveY - lastMoveY);
}
```



```

    if (moveDistX + totalTranslateX > 0 || moveDistX + totalTranslateX + curBitmapWidth < width) {
        moveDistX = 0;
    }
    if (moveDistY + totalTranslateY > 0 || moveDistY + totalTranslateY + curBitmapHeight < height) {
        moveDistY = 0;
    }
    status = STATUS_MOVE;
    invalidate();
    lastMoveX = moveX;
    lastMoveY = moveY;
}

```

إلى: أعلاه الكود ترجمة تمت

```

private void handleMove(MotionEvent event) {
    float moveX = event.getX();
    float moveY = event.getY();
    if (lastMoveX == -1 && lastMoveY == -1) {
        lastMoveX = moveX;
        lastMoveY = moveY;
    }
    moveDistX = (int) (moveX - lastMoveX);
    moveDistY = (int) (moveY - lastMoveY);
    if (moveDistX + totalTranslateX > 0 || moveDistX + totalTranslateX + curBitmapWidth < width) {
        moveDistX = 0;
    }
    if (moveDistY + totalTranslateY > 0 || moveDistY + totalTranslateY + curBitmapHeight < height) {
        moveDistY = 0;
    }
    status = STATUS_MOVE;
    invalidate();
    lastMoveX = moveX;
    lastMoveY = moveY;
}

```

لا والتي، البرمجة بلغة محددة ووظائف متغيرات أسماء على يحتوي لأنه تغيري ي تم لم الكود ملاحظة:
عادةً، ترجمته ي تم

```

private void handleZoom(MotionEvent event) {
    float fingerDist = calFingerDistance(event);
    calFingerCenter(event);
    if (fingerDist > lastFingerDist) {
        status = STATUS_ZOOM_OUT;
    } else {
        status = STATUS_ZOOM_IN;
    }
    scaledRatio = fingerDist * 1.0f / lastFingerDist;
    totalRatio = totalRatio * scaledRatio;
    if (totalRatio < initRatio) {
        totalRatio = initRatio;
    } else if (totalRatio > initRatio * 4) {
        totalRatio = initRatio * 4;
    }
    lastFingerDist = fingerDist;
    invalidate();
}

```

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

للعرض. الحالية الحالة يرسم

```

@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    if (scaleMode) {
        switch (status) {
            case STATUS_MOVE:
                move(canvas);
                break;
            case STATUS_ZOOM_IN:
            case STATUS_ZOOM_OUT:
                zoom(canvas);
                break;
            default:
                if (cacheBm != null) {
                    canvas.drawBitmap(cacheBm, matrix, null);
                }
            }
        }
    }
}

```

```

        canvas.drawPath(path, paint);
    }
}
} else {
    if (cacheBm != null) {
        canvas.drawBitmap(cacheBm, matrix, null);
        canvas.drawPath(path, paint);
    }
}
}
}

```

الترجمة من العربية إلى الكود

الترجمة من العربية إلى الكود

```

private void move(Canvas canvas) {
    matrix.reset();
    matrix.postScale(totalRatio, totalRatio);
    totalTranslateX = moveDistX + totalTranslateX;
    totalTranslateY = moveDistY + totalTranslateY;
    matrix.postTranslate(totalTranslateX, totalTranslateY);
    canvas.drawBitmap(cacheBm, matrix, null);
}

```

الترجمة من العربية إلى الكود

```

private void move(Canvas canvas) {
    matrix.reset(); //
    matrix.postScale(totalRatio, totalRatio); //
    totalTranslateX = moveDistX + totalTranslateX; //
    totalTranslateY = moveDistY + totalTranslateY; //
    matrix.postTranslate(totalTranslateX, totalTranslateY); //
    canvas.drawBitmap(cacheBm, matrix, null); //
}

```

الترجمة من العربية إلى الكود

الترجمة من العربية إلى الكود

الترجمة من العربية إلى الكود

```

private void zoom(Canvas canvas) {
    matrix.reset();
    matrix.postScale(totalRatio, totalRatio);
    int scaledWidth = (int) (cacheBm.getWidth() * totalRatio);
    int scaledHeight = (int) (cacheBm.getHeight() * totalRatio);
    int translateX;
    int translateY;
    if (curBitmapWidth < width) {
        translateX = (width - scaledWidth) / 2;
    } else {
        translateX = (int) (centerPointX + (totalTranslateX - centerPointX) * scaledRatio);
        if (translateX > 0) {
            translateX = 0;
        } else if (scaledWidth + translateX < width) {
            translateX = width - scaledWidth;
        }
    }
    if (curBitmapHeight < height) {
        translateY = (height - scaledHeight) / 2;
    } else {
        translateY = (int) (centerPointY + (totalTranslateY - centerPointY) * scaledRatio);
        if (translateY > 0) {
            translateY = 0;
        } else if (scaledHeight + translateY < height) {
            translateY = height - scaledHeight;
        }
    }
    matrix.postTranslate(translateX, translateY);
    canvas.drawBitmap(cacheBm, matrix, null);
}

```

الترجمة:

```

private void zoom(Canvas canvas) {
    matrix.reset();
    matrix.postScale(totalRatio, totalRatio);

```

```

int scaledWidth = (int) (cacheBm.getWidth() * totalRatio);
int scaledHeight = (int) (cacheBm.getHeight() * totalRatio);
int translateX;
int translateY;
if (curBitmapWidth < width) {
    translateX = (width - scaledWidth) / 2;
} else {
    translateX = (int) (centerPointX + (totalTranslateX - centerPointX) * scaledRatio);
    if (translateX > 0) {
        translateX = 0;
    } else if (scaledWidth + translateX < width) {
        translateX = width - scaledWidth;
    }
}
if (curBitmapHeight < height) {
    translateY = (height - scaledHeight) / 2;
} else {
    translateY = (int) (centerPointY + (totalTranslateY - centerPointY) * scaledRatio);
    if (translateY > 0) {
        translateY = 0;
    } else if (scaledHeight + translateY < height) {
        translateY = height - scaledHeight;
    }
}
matrix.postTranslate(translateX, translateY);
canvas.drawBitmap(cacheBm, matrix, null);
}

Y = height - scaledHeight;
}
}

totalTranslateX = translateX;
totalTranslateY = translateY;
curBitmapWidth = scaledWidth;
curBitmapHeight = scaledHeight;
matrix.postTranslate(translateX, translateY);
canvas.drawBitmap(cacheBm, matrix, null);

```

```
}
```

التاريخ إدارة

والإعادة. التراجع وظائف لتمكين الرسم سجل تدير History فئة

التمديدات، التراجع التراجع

المكدس. في والفرشاة الحالي المساري حفظ

```
public void saveToStack(Path path, Paint paint) {  
    Draw draw = new Draw();  
    draw.path = new Path(path);  
    draw.paint = new Paint(paint);  
    saveToStack(draw);  
}
```

يلي: كما العربية إلى أعلاه الكود ترجمة تمت

```
public void saveToStack(Path path, Paint paint) {  
    Draw draw = new Draw();  
    draw.path = new Path(path);  
    draw.paint = new Paint(paint);  
    saveToStack(draw);  
}
```

Path، مثل الكائنات أو المتغيرات أو الدوال أسماء ترجمة يتم ولا، بلغة مكتوب المقدم الكود ملاحظة: Paint، Draw، إلخ. الترجمة. يتم ولا البرمجة لغة من جزء لأنها هي كما الـ أسماء هذه على الحفاظ يتم إلخ.

```
public void saveToStack(Draw draw) {  
    curPos++;  
    while (histroy.size() > curPos) {  
        histroy.pop();  
    }  
    histroy.push(draw);  
    if (callback != null) {  
        callback.onHistoryChanged();  
    }  
}
```

$$\left. \begin{array}{l} \} \\ \} \end{array} \right\}$$

إلى: أعلاه الكود ترجمة تمت

```
public void saveToStack(Draw draw) {
    curPos++;
    while (histroy.size() > curPos) {
        histroy.pop();
    }
    histroy.push(draw);
    if (callBack != null) {
        callBack.onHistoryChanged();
    }
}
```

كما عليها الحفاظ ويجب بالإنجليزية، ووظائف متغيرات أسماء على يحتوي لأنها تغيّره يتم لم الكود ملاحظه: صريح. بشكل الكود عمل لضم ان هي

[illegible]

..... الرسم سجل في المحددة النقطة حالة تم مثل نقطية صورة تخرج

```
public Bitmap getBitmapAtDraw(int n) {
    Canvas canvas = new Canvas();
    Bitmap bm = Utils.getCopyBitmap(srcBitmap);
    canvas.setBitmap(bm);
    for (int i = 0; i <= n; i++) {
        Draw draw = histroy.get(i);
        canvas.drawPath(draw.path, draw.paint);
    }
    return bm;
}
```

□ □ □ □ □ □

التراجع. عمليّة بتنفيذي قوم

```
public Bitmap undo() throws UnsupportedOperationException {
    if (canUndo()) {
```

```

        curPos--;
        if (callBack != null) {
            callBack.onHistoryChanged();
        }
        return getBitmapAtDraw(curPos);
    } else {
        throw new UnsupportedOperationException( " " );
    }
}

```

□□□□□□

□□□□□□□□. ال إعادة عملية بت تنفيذ في ذيقوم

```

public Bitmap redo() throws UnsupportedOperationException {
    if (canRedo()) {
        curPos++;
        if (callBack != null) {
            callBack.onHistoryChanged();
        }
        return getBitmapAtDraw(curPos);
    } else {
        throw new UnsupportedOperationException( " " );
    }
}

```

□□□□□□□□□□

ال أخير. ال إجراء عن ال تراجع بال إمكان كان إذا مم ا ي تحقق

```

public boolean canUndo() {
    return curPos > 0;
}

```

ال ترجمة:

```

public boolean canUndo() {
    return curPos > 0;
}

```


تتغير ولا عالمية برمجة لغة وهي `Boolean` بلغة مكتوب أنه حيث ترجمة إلى يحتاج لا المقدم الكود ملاحظة:
الللغة. باختلاف

`Boolean`

الإجراء. إعادة بإمكان كان إذا مما يتحقق

```
public boolean canRedo() {  
    return curPos + 1 < histroy.size();  
}
```

الترجمة:

```
public boolean canRedo() {  
    return curPos + 1 < histroy.size();  
}
```

الإجراءات تنفذ إعادة مع يتعامل تطبيق من جزء أنه ويبدو `Boolean` بلغة مكتوب المقدم الكود ملاحظة:
الموضع أن من التحقق طريق عن `Boolean` الإجراء إعادة تنفذ بإمكان كان إذا مما يتحقق الكود `Boolean`.
يعني فهذا صحيحًا، الشرط كان إذا `histroy.size()` التاريخ قائمة حجم من أقل واحد زائد `curPos` الحالي
الإجراء. إعادة تنفذ يمكن أنه

الخلاصة

`DrawActivity` نظام في مخصص رسم عرض لتنفيذ شاملًا مثالًا به المرتبطة والفئات `DrawActivity` يوفر
مكونات مع والتكامل الرسم، سجل وإدارة اللمس، أحداث مع التعامل ذلك في بما متنوعة، تقنيات المثال هذا عرض
وخوارزمية، مكون لكل فهم خلال من `Boolean` المتزامنة غير والمهام `Boolean` الأجزاء مثل أخرى
وتفاعلية. قوية رسم ميزات لإنشاء الخاصة تطبيقاتك في التقنيات هذه من الاستفادة يمكنك