智慧型手機應用程式開發
期末專案

魔法陣產生器 with AR

組別:第二十五組

組員:資工二 4104056004 李家駿

日期:2017/06/25

目錄:

一、 前言

某天在網路上看到了有人在介紹一款 APP，這款 APP 叫「Mahougen-MagicCircleGenerator(魔法陣產生器)」，可讓使用者畫出魔法陣。這挺讓人興奮的，特別是對於我這種沒有美術天分的人，別說是魔法陣，畫個圓就夠困難了。

實際上 APP 的表現也沒有讓我失望，稍微控制一下的確可以畫出非常對稱精美的魔法陣。但總覺得少了一種「實際」的感覺，只在手機 APP 上顯示總是與現實有點差距。在這個想法的驅使下，催生出了這款 APP—魔法陣產生器 AR。

二、 功能簡介

- 讓使用者在畫面上畫出軌跡，app 會自動將軌跡轉成魔法陣
- AR 功能:將圖片或卡片當成目標(ImageTarget)，把魔法陣投影在上面
- 可將圖片透過社群軟體與好友分享

三、 介面

1. 主介面

RESET:清除畫面上的魔法陣

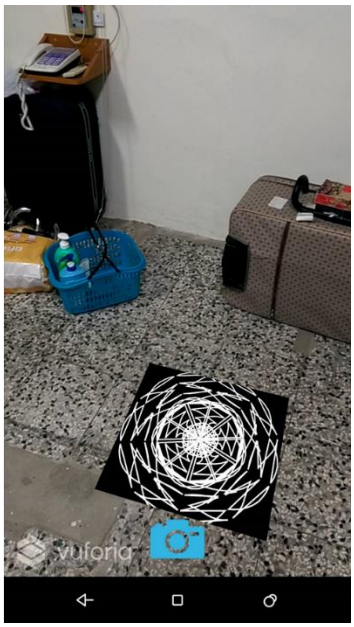
SHARE:將魔法陣透過社群軟體分享給朋友

2.AR 介面

1. 將目標物對準鏡頭:

2. 按下相機圖示:

召喚!

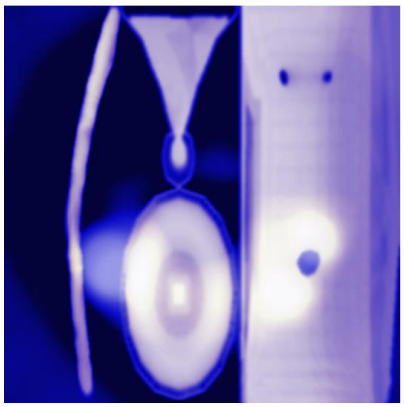


四、 關鍵技術

AR 可分為兩種:Marker 與 Markerless，Marker 就是一個目標物，可以讓鏡頭辨識出手機與目標物的距離。這個專案所用到的是有 Marker 的 AR，vuforia 的 AR SDK 使用的就是 Marker 的技術。vuforia 套件提供許多模式的 AR 技術，這次使用到的技術是 User define target，可將使用者拍攝的圖片作為目標物，顯示出 AR。

流程示意圖↓

圖片



虛擬object

五、 製作過程

製作過程主要分成兩個階段:1.魔法陣產生器 UI，2.AR UI。以下將詳述兩個階段的製作過程。

1. 魔法陣產生器 UI:

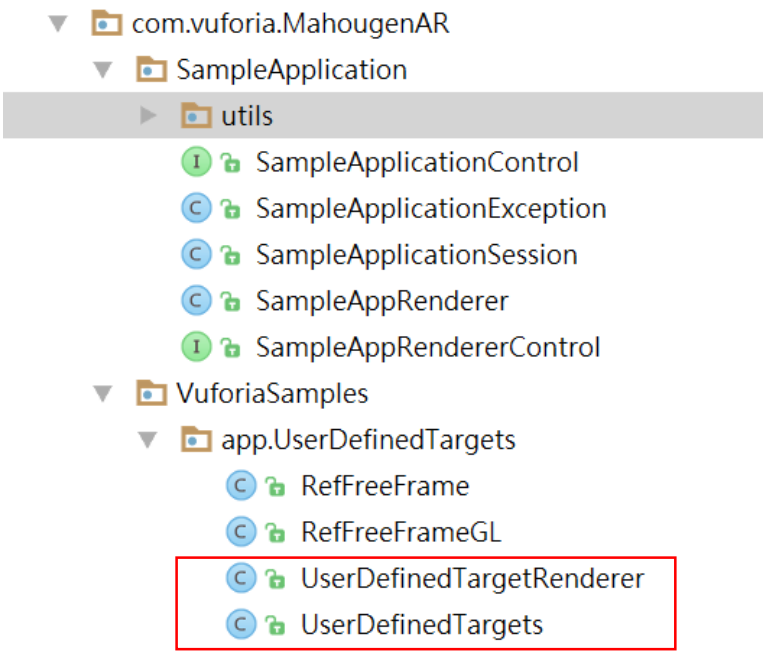
首先是將原本的魔法陣產生器 APPUI 實現出來，根據 https://github.com/pistatium/mahougen 裡的程式碼加以改編。整個 UI 中最重要的部分就是中間的 MahougenView，是利用 java 的 Path 類別紀錄使用者畫出的軌跡，再進行一些數學運算，轉換成魔法陣的形狀。而後再利用 java 的 Canvas 與 Paint class 將路徑畫上顏色，產生圖檔。

與原本 APP 不同的是，加上了利用 menu 改變顏色的功能，增加了魔法陣的多樣性。

2. AR UI:

AR UI 的部分是改寫自 Vuforia 官方的 sample project。主要改寫的部分是 com.vuforia.samples/VuforiaSamples/UserDefinedTargets 中的兩個 class:

UserDefinedTarget.class 與 UserDefinedTargetRenderer.class。



UserDefinedTargetRenderer.class:

```
56      static final float kObjectScale = 0.f;
57      static final float kObjectSize = 400.f;
58      //private Teapot mTeapot;
59      private Mahougen mTeapot;
```

↑kObjectScale 是虛擬物體與 target 的距離，kObjectSize 是我自己宣告的變數，定義物體的大小。在這裡也將原本 Teapot 型態的 mTeapot 換成用

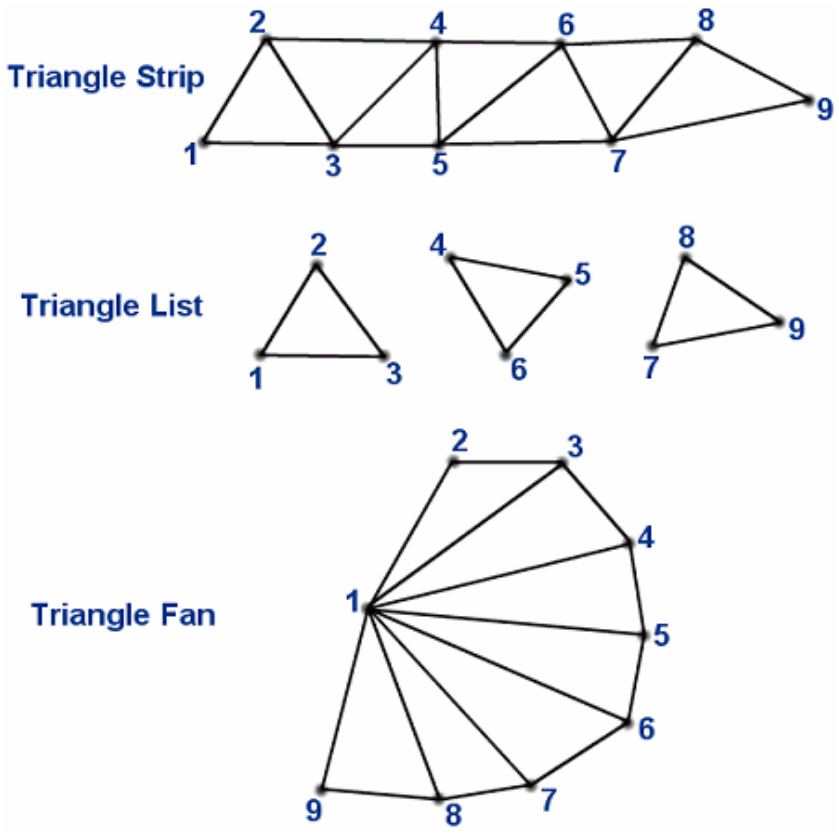
Mahougen 型態宣告。

```
157             Matrix.scaleM(modelViewMatrix, 0, kObjectSize,kObjectSize,
158                 kObjectScale);
```

↑這段程式碼會定義物體的長寬及高度，第三四個參數便是物體的長寬，最後一個參數即是物體的高度。

```
178     GLES20.glDrawArrays(GLES20.GL_TRIANGLE_FAN, 0, mTeapot.getNumObjectIndex());
```

↑這段程式碼會將虛擬物體畫出，GL_TRIANGLE_FAN 參數代表會將物體以三角形畫出，0 是起始的位置，最後一個參數是點的數量。



↑

各種參數的繪製順序(圖片來源:

http://hungyanbin.blogspot.tw/2015/04/android.html

)

UserDefinedTarget.class:

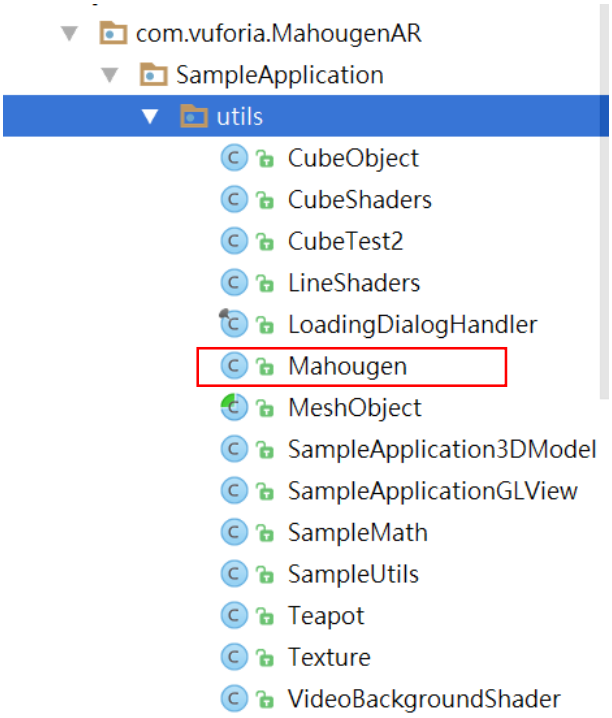
```
171     private void loadTextures()
172     {
173         //mTextures.add(Texture.loadTextureFromApk("TextureTeapotBlue.png",
174         //   getAssets()));
175         /**First get the image name from the intent.
176          * */
177         Intent intent = getIntent();
178         //sdcard path
179         File sdFile = android.os.Environment.getExternalStorageDirectory();
180         //path of image
181         String path = sdFile.getPath()
182                 + File.separator + "mahougens/"+intent.getStringExtra("imageName")
183         //Trying to turn the image into texture and add it into mTextures.
184         try{
185 
186             FileInputStream in = new FileInputStream(path);
187             BufferedInputStream buf = new BufferedInputStream(in);
188             byte[] bMapArray= new byte[buf.available()];
189             buf.read(bMapArray);
190             Bitmap bMap = BitmapFactory
191                     .decodeByteArray(bMapArray, 0, bMapArray.length);
192             mTextures.add(loadTextureFromBitmap(bMap));
```

↑修改的部分是加入了 intent 接取魔法陣圖片的名稱。原本的程式碼是會將 assets 中的圖片轉成 Texture 並加入 mTexture 中，但我們的 APP 是從 sd 卡中讀取圖片，所以加入了讀檔的功能，並將.png 圖片轉成 bitmap 形式，再將 bitmap 轉成 Texture，加入 mTexture 中。

```
200     public static Texture loadTextureFromBitmap(Bitmap bitMap)
201     {
202         int[] data = new int[bitMap.getWidth() * bitMap.getHeight()];
203         bitMap.getPixels(data, 0, bitMap.getWidth(), 0, 0, bitMap.getWidth(), bitM
204         // Convert:
205         byte[] dataBytes = new byte[bitMap.getWidth() * bitMap.getHeight() * 4];
206 
207         for (int p = 0; p < bitMap.getWidth() * bitMap.getHeight(); ++p)
208         {
209             int colour = data[p];
210             dataBytes[p * 4] = (byte)(colour >>> 16); // R
211             dataBytes[p * 4 + 1] = (byte)(colour >>> 8); // G
212             dataBytes[p * 4 + 2] = (byte) colour; // B
213             dataBytes[p * 4 + 3] = (byte)(colour >>> 24); // A
214         }
215 
216         Texture texture = new Texture();
217         texture.mWidth = bitMap.getWidth();
218         texture.mHeight = bitMap.getHeight();
219         texture.mChannels = 4;
220         texture.mData = ByteBuffer.wrap(dataBytes);
221         return texture;
```

↑Bitmap to Texture

我也在 util 裡面加入了自己寫的 Mahougen.class，用於描述魔法陣圖片的形狀。



3. 整合:

將兩個 UI 整合起來，並讓 APP 預先載入魔法陣產生器 UI，必須修改 sample 中的 AndroidManifest，如下:

```
 </activity>
<activity android:name="mahougenar.Mahougen_Drawing_Activity"
    android:theme="@style/Theme.AppCompat"
    android:screenOrientation="portrait" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

加入這段即可讓 APP 預先載入魔法陣產生器 UI activity.

六、 程式碼

1. com.vuforia.MahougenAR

A. Mahougen.java

```
package com.vuforia.MahougenAR.SampleApplication.utils;

import java.nio.Buffer;

public class Mahougen extends MeshObject{
    private Buffer mVertBuff;
    private Buffer mTexCoordBuff;
    private Buffer mNormBuff;
    private Buffer mIndBuff;

    private int indicesNumber = 0;
    private int verticesNumber = 0;

    public Mahougen()
    {
        setVerts();
        setTexCoords();
        setNorms();
        setIndices();
    }
```

```
  private void setIndices(){
      short[] indice = new short[]{ 0, 1, 2, 3};
      mIndBuff = fillBuffer(indice);
      indicesNumber = indice.length;
  }
  private void setVerts(){
      //double[] verts = new double[] {};

      double[] verts = new double[] {
              -0.5f, -0.5f, 0.0f, //bottom-left corner
              0.5f, -0.5f, 0.0f, //bottom-right corner
              0.5f,  0.5f, 0.0f, //top-right corner
              -0.5f,  0.5f, 0.0f  //top-left corner
      };
      mVertBuff = fillBuffer(verts);
      verticesNumber = verts.length / 3;
  }
```

```
private void setNorms(){
    double[] norms = new double []{
            0.0f, 0.0f, 1.0f, //normal at bottom-left corner
            0.0f, 0.0f, 1.0f, //normal at bottom-right corner
            0.0f, 0.0f, 1.0f, //normal at top-right corner
            0.0f, 0.0f, 1.0f  //normal at top-left corner
    };
    mNormBuff = fillBuffer(norms);
}
private void setTexCoords(){
    double[] cords = new double[]{
            0.0f, 0.0f, //tex-coords at bottom-left corner
            1.0f, 0.0f, //tex-coords at bottom-right corner
            1.0f, 1.0f, //tex-coords at top-right corner
            0.0f, 1.0f  //tex-coords at top-left corner
    };
    mTexCoordBuff = fillBuffer(cords);

}
public int getNumObjectIndex()
```

```
public int getNumObjectIndex()
{
    return indicesNumber;
}


@Override
public int getNumObjectVertex() { return verticesNumber; }


@Override
public Buffer getBuffer(BUFFER_TYPE bufferType)
{
    Buffer result = null;
    switch (bufferType)
    {
        case BUFFER_TYPE_VERTEX:
            result = mVertBuff;
            break;
        case BUFFER_TYPE_TEXTURE_COORD:
            result = mTexCoordBuff;
            break;
```

```
            case BUFFER_TYPE_TEXTURE_COORD:
                result = mTexCoordBuff;
                break;
            case BUFFER_TYPE_NORMALS:
                result = mNormBuff;
                break;
            case BUFFER_TYPE_INDICES:
                result = mIndBuff;
            default:
                break;

        }

        return result;
    }
```

B. UserDefineTargetRenderer.java

製作過程中已有解釋修改的部分，其餘皆與 Vuforia sample 中的程式碼相同，故不重複貼上。

C. UserDefineTarget.java

不重複貼上，理由同上。

2. Mahougenar

A. Mahougen_Drawing_Activity.java

Variables:

```
/**Variables declaration*/
private MahougenView mahougenView; //The Mahougen view
private SeekBar sbMP;              //The MP seekBar
private TextView tvMP;             //The TextView of MP
private ArrayList<String> images = new ArrayList<~>(); //The namelist of image
private String imageName;
ArrayAdapter<String> imageAdapter;
@Override
```

onCreate():

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_mahougen__drawing_);

    // ask permission to read and write sdcard
    askPermissions();

    //Create the Mahougen dir
    CreateMahougenDir();
    //update image list
    updateImageList();

    // find the views
    mahougenView = (MahougenView)findViewById(R.id.mahougenView);
    sbMP = (SeekBar)findViewById(R.id.seekBarMP);
    tvMP=(TextView)findViewById(R.id.textView);

    //set the textView tMP
    tvMP.setText("MP:10");
```

seekBar listener:

```
sbMP.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser
        //Set the MP.
        //The MP is the vertexCount in MahougenView.
        tvMP.setText("MP:"+(progress+3));
        mahougenView.changeMP(progress+3);
        //re-initialize the mahougenView
        mahougenView.clear();
        mahougenView=(MahougenView)findViewById(R.id.mahougenView);
    }

    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {

    }

    @Override
    public void onStopTrackingTouch(SeekBar seekBar) {

    }
});
```

askPermissions():

```
@TargetApi(Build.VERSION_CODES.M)
protected void askPermissions() {
    /**ask the permission for R/W sdcard.*/
    String[] permissions = {
            "android.permission.READ_EXTERNAL_STORAGE",
            "android.permission.WRITE_EXTERNAL_STORAGE"
    };
    int requestCode = 200;
    requestPermissions(permissions, requestCode);
}
```

CreateMahougenDir():

```
public void CreateMahougenDir()
{
    File sdFile = android.os.Environment.getExternalStorageDirectory();
    String path = sdFile.getPath() + File.separator + "mahougens"; //The
    File dirFile = new File(path);
    if(!dirFile.exists())//如果資料夾不存在
        dirFile.mkdir();//建立資料夾
}
```

OnResetClick(View v):

```
public void OnResetClick(View v)
{
    /**reset the mahougenView*/
    //clear
    mahougenView.clear();
    //show the clear Toast
    Toast.makeText(this,"Clear", Toast.LENGTH_LONG)
            .show();
}
```

Save():

```
public File Save()
{
    /**Save the mahougen image into the sdcard.
     * If there is no the directory of "Mahougens",create one.
     * This app will get the mahougen images from the directory,so
     * you can put your own images(e.g. .png or .jpg ) into the directory and
     * (This should be a bug,but I want to keep it as a hidden function XD.)
     * */
    try{
        File sdFile = android.os.Environment.getExternalStorageDirectory();
        String path = sdFile.getPath() + File.separator + "mahougens"; //The p

        imageName= System.currentTimeMillis()+".png"; //Use the time as the fi
        File file = new File(path, imageName);
        OutputStream stream = new FileOutputStream(file);
        mahougenView.saveBitmap(stream); //Put the bitmap into file stream.
        stream.close();

        //Show Toast
        Toast.makeText(this,"save success", Toast.LENGTH_LONG)
```

```
            //Show Toast
            Toast.makeText(this,"save success", Toast.LENGTH_LONG)
            .show();
            return file;
        }catch(Exception e){
            //Show error
            Toast.makeText(this,"save failed", Toast.LENGTH_LONG)
            .show();
            e.printStackTrace();
        }
        return null;
    }
```

OnShareClick(View v):

```
public void OnShareClick(View v)
{
    /**Share the mahougen through other SNS.*/
    // get file directory.
    final File pictureFile = Save();
    // invoke an intent with ACTION_SEND
    final Intent shareIntent = new Intent(Intent.ACTION_SEND);
    //Set the image type as png
    shareIntent.setType("image/png");
    shareIntent.putExtra(Intent.EXTRA_STREAM, Uri.fromFile(pictureFile));
    startActivity(Intent.createChooser(shareIntent,"share"));
}
```

OnSummonClick(View v):

```
final String[] chioce = new String[]{"用這張!", "從資料夾選取"}; //"Choose this" or

AlertDialog showChoice
        = new AlertDialog.Builder(Mahougen_Drawing_Activity.this)
        .setTitle("選擇魔法陣來源")
        .setItems(chioce, (dialogInterface, i) -> {
                if (chioce[i] == "用這張!") {
                    //Use the mahougen you draw
                    Save();//save mahougen
                    showTutorial();
                } else if (chioce[i] == "從資料夾選取") {
                    //Use the mahougen(or other images) from the dir
                    AlertDialog imageChioce
                            = new AlertDialog.Builder(Mahougen_Drawing_Activity.t
                            .setAdapter(imageAdapter, (dialogInterface, i) -> {
                                    //get the image name from the list
                                    imageName = imageAdapter.getItem(i).toString(
                                    showTutorial();
                            })
                            .show();
                }
```

showTutorial():

```
public void showTutorial()
{
    /**Showing the tutorial of the generating.*/
    AlertDialog showTheTutorial
            = new AlertDialog.Builder(Mahougen_Drawing_Activity.this)
            .setTitle("即將生成魔法陣")
            .setMessage("生成魔法陣時，請依照以下步驟:\n" +
                    "1.找一張辨識度高的相片或卡片(悠遊卡)，做為目標物\n" +
                    "2.將相機畫面對準對焦至目標物，盡量填滿整個相機畫面\n" +
                    "3.按下正下方的相機圖示，魔法陣將會生成\n" +
                    "4.成為大魔法師吧!\n")
            .setPositiveButton("生成", (dialogInterface, i) -> {
                    //Choose the generate
                    Toast.makeText(Mahougen_Drawing_Activity.this
                            , "即將生成，請稍等...", Toast.LENGTH_LONG).show();
                    //Create the intent and sent the image name to the UserDefine
                    Intent intent = new Intent();
                    intent.setClassName(getPackageName()
                            , getPackageName() +
                                    ".app.UserDefinedTargets.UserDefinedTargets")
                    intent.putExtra("imageName",imageName);
                    //go to the AR ui
```

```
                    startActivity(intent);
            })
            .setNegativeButton("取消", (dialogInterface, i) -> {
                    //Choose to cancel
                    Toast.makeText(Mahougen_Drawing_Activity.this,
                            "取消", Toast.LENGTH_LONG).show();
            })
            .show();
```

onCreateOptionsMenu(Menu menu):

```
public boolean onCreateOptionsMenu(Menu menu)
{
    /** select to change background color or line color*/
    //The manu of background color
    SubMenu subMenuBackground
            = menu.addSubMenu(Menu.NONE, Menu.FIRST, Menu.NONE, "改變背景顏色");
    subMenuBackground.add(Menu.NONE, Menu.FIRST+1, Menu.NONE,"黑色");
    subMenuBackground.add(Menu.NONE, Menu.FIRST+2, Menu.NONE,"白色");
    subMenuBackground.add(Menu.NONE, Menu.FIRST+3, Menu.NONE,"灰色");
    subMenuBackground.add(Menu.NONE, Menu.FIRST+4, Menu.NONE,"藍色");
    subMenuBackground.add(Menu.NONE, Menu.FIRST+5, Menu.NONE,"紅色");

    //The manu of line color
    SubMenu subMenuLine
            = menu.addSubMenu(Menu.NONE+1, Menu.FIRST+6, Menu.NONE, "改變線條顏色");
    subMenuLine.add(Menu.NONE+1, Menu.FIRST+7, Menu.NONE,"黑色");
    subMenuLine.add(Menu.NONE+1, Menu.FIRST+8, Menu.NONE,"白色");
    subMenuLine.add(Menu.NONE+1, Menu.FIRST+9, Menu.NONE,"灰色");
    subMenuLine.add(Menu.NONE+1, Menu.FIRST+10, Menu.NONE,"藍色");
    subMenuLine.add(Menu.NONE+1, Menu.FIRST+11, Menu.NONE,"紅色");

    return super.onCreateOptionsMenu(menu);
```

onOptionsItemSelected(MenuItem item):

```
@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    /**If user choose the color in the menu ,change it.*/
        switch(item.getItemId()){
            case Menu.FIRST+1:
                mahougenView.changeBackgroundColor("BLACK");
                break;
            case Menu.FIRST+2:
                mahougenView.changeBackgroundColor("WHITE");
                break;
            case Menu.FIRST+3:
                mahougenView.changeBackgroundColor("GRAY");
                break;
            case Menu.FIRST+4:
                mahougenView.changeBackgroundColor("BLUE");
                break;
            case Menu.FIRST+5:
                mahougenView.changeBackgroundColor("RED");
                break;
```

```
                    mahougenView.changeLineColor("BLACK");
                    break;
                case Menu.FIRST+8:
                    mahougenView.changeLineColor("WHITE");
                    break;
                case Menu.FIRST+9:
                    mahougenView.changeLineColor("GRAY");
                    break;
                case Menu.FIRST+10:
                    mahougenView.changeLineColor("BLUE");
                    break;
                case Menu.FIRST+11:
                    mahougenView.changeLineColor("RED");
                    break;
            }
            if(item.getItemId()!= Menu.FIRST&&item.getItemId()!= Menu.FIRST+6)
                //If the chosen item is not the "Change background" or "Change l
                Toast.makeText(this,"顏色已修改", Toast.LENGTH_LONG).show();
        return super.onOptionsItemSelected(item);
    }
```

updateImageList():

```
public void updateImageList() {
    /** load the song list form sdcard*/
    File home = new File(Environment
            .getExternalStorageDirectory().getPath()+"/mahougens/");
    /*check if there is any file*/
    if (home.listFiles( new FileExtensionFilter()).length > 0) {
        for (File file : home.listFiles( new FileExtensionFilter())) {
            //add png file to image list
            images.add(file.getName());
        }

        /*put the image list into ListView*/
        imageAdapter = new ArrayAdapter<String>
                (this,R.layout.support_simple_spinner_dropdown_item,images);

    }
}
```

class FileExtensionFilter:

```
public class FileExtensionFilter implements FilenameFilter {
    /** check if file is end with ".png" or ".PNG"*/
    public boolean accept(File dir, String name) {
        return (name.endsWith(".png")
                || name.endsWith(".PNG")
                ||name.endsWith(".jpg")
                ||name.endsWith(".JPG"));
    }
}
```

B. MahougenView.java

Variables:

```
/**Variables declaration*/
private Paint mPaint = null;
private Bitmap mBitmap = null;
private Vector center=new Vector(0.0,0.0);
private int vertexCount=10; //=MP
ArrayList<Path> pathArray;
```

Constructor:

```
public MahougenView(Context context , AttributeSet attrs)
{
    super(context,attrs);
    this.setBackgroundColor(Color.BLACK);
    mPaint = new Paint();
    mPaint.setStyle(Paint.Style.STROKE);
    mPaint.setColor(Color.WHITE);
    mPaint.setStrokeWidth(10);
    mPaint.setAntiAlias(true);
    pathArray=new ArrayList<Path>();
    for(int i=0;i<vertexCount;i++) //initialize the path array by the number of ve
    {
        pathArray.add(new Path());
    }

}
```

onWindowFocusChanged(boolean hasWindowFocus):

```
public void onWindowFocusChanged(boolean hasWindowFocus){
    super.onWindowFocusChanged(hasWindowFocus);
    /**set the center */
    this.center
            = new Vector(
            (this.getRight() - this.getLeft())/2
            ,(this.getBottom()-this.getTop())/2);


}
```

onTouchEvent(MotionEvent event):

```
public boolean onTouchEvent(MotionEvent event)
{
    /**Calculate the path while touching*/
    Vector current = new Vector(event.getX(),event.getY());
    //System.out.println("current.x="+current.x+"current.y="+current.y);
    //System.out.println("center.x="+center.x+"center.y="+center.y);
    Vector direction = current.minus(center);
    double r = direction.size();
    double alpha =(2.0* Math.PI/this.vertexCount);
    double theta = direction.angle();

    for(int i=0;i<this.vertexCount;i++)
    {
        if(i*alpha>theta){
            theta -= (i-1)*alpha;
            break;
        }
    }
```

```
switch (event.getAction())
{
    case MotionEvent.ACTION_DOWN:
        int i=0;
        for(Path p : pathArray) {
            Vector target
                    = this.center
                    .plus(Vector.ofAngle(theta + i * alpha).times(r));
            if(vertexCount%2==0&&i%2==0)
                target
                        = this.center
                        .plus(Vector.ofAngle(-theta+(i+1)*alpha).times(r));
            p.moveTo((float)target.x,(float)target.y);
            i++;
        }
        break;
```

```
    case MotionEvent.ACTION_MOVE:
    case MotionEvent.ACTION_UP:

        i=0;
        for(Path p : pathArray) {
            Vector target
                    = this.center
                    .plus(Vector.ofAngle(theta + i * alpha).times(r));
            if(vertexCount%2==0&&i%2==0)
                target
                        = this.center
                        .plus(Vector.ofAngle(-theta+(i+1)*alpha).times(r));
            p.lineTo((float)target.x,(float)target.y);
          //  System.out.println("x="+target.x+",y="+target.y);
            i++;
        }
        break;

}
invalidate();
return true;
```

changeMP(int MP):

```
public void changeMP(int MP)
{
    /**Changing the MP means change the number of lines.*/
    vertexCount=MP;
    // clean the canvas
    clear();
    // initialize the path array again
    pathArray=new ArrayList<Path>();
    for(int i=0;i<vertexCount;i++)
    {
        pathArray.add(new Path());
    }
}
```

onDraw(Canvas canvas):

```
public void onDraw(Canvas canvas)
{
    // draw the paths
    super.onDraw(canvas);
      for(Path p:pathArray) {
          canvas.drawPath(p, mPaint);
      }
}
```

saveBitmap(OutputStream stream):

```
public void saveBitmap(OutputStream stream)
{
    /**Trying to save the mahougen in the file stream*/
    try {
        // get the mahougen
        mBitmap=getBitmapFromView(this);
        // put bitmap into file stream
        mBitmap.compress(Bitmap.CompressFormat.PNG, 100, stream);
    }
    catch(Exception e)
    {
        // output the error
        System.out.println("compress error");
        e.printStackTrace();
    }
}
```

clear():

```
public void clear()
{
    /**clean the paths*/
    for(Path p:this.pathArray)
    {
        p.reset();
    }
    invalidate();

}
```

changeBackgroundColor(String color):

```
public void changeBackgroundColor(String color)
{
    /**Changes the background color*/
    try {
        //First turn the string into id.
        //And than set the background color.
        this.setBackgroundColor(Color.class.getField(color).getInt(null));
    }
    catch(Exception e)
    {
        //output the error
        System.out.println("change background color error.");
        e.printStackTrace();
    }
}
```

changeLineColor(String color):

```
public void changeLineColor(String color)
{
    /**Changes the line color*/
    try {
        //First turn the string into id.
        //And than set the line color.
        mPaint.setColor(Color.class.getField(color).getInt(null));
    }
    catch(Exception e)
    {
        System.out.println("change line color error.");
        e.printStackTrace();
    }
}
```

getBitmapFromView(View view):

```
public static Bitmap getBitmapFromView(View view) {
    /**Turn the view into bitmap.
     * This code is copy from the below website:
     * https://stackoverflow.com/questions/2801116/converting-a-view-to-bitma
     */
    //Define a bitmap with the same size as the view
    Bitmap returnedBitmap = Bitmap.createBitmap(view.getWidth(), view.getHeig
    //Bind a canvas to it
    Canvas canvas = new Canvas(returnedBitmap);
    //Get the view's background
    Drawable bgDrawable =view.getBackground();
    if (bgDrawable!=null)
        //has background drawable, then draw it on the canvas
        bgDrawable.draw(canvas);
    else
        //does not have background drawable, then draw white background on th
        canvas.drawColor(Color.WHITE);
    // draw the view on the canvas
    view.draw(canvas);
    //return the bitmap
    return returnedBitmap;
```

C. Vector.java

```
public class Vector{

    double x,y;
    public Vector(double x,double y){
        this.x=x;
        this.y=y;
    }

    public Vector plus(Vector v) { return new Vector(this.x+v.x,this.y+v.y); }

    public Vector minus(Vector v) { return new Vector(this.x-v.x,this.y-v.y); }

    public Vector times(double r) { return new Vector(this.x*r,this.y*r); }

    public double size() { return Math.sqrt((this.x*this.x+this.y*this.y)); }

    public double angle() { return Math.atan2(this.x,this.y); }

    public static Vector ofAngle(double theta)
    {
        return new Vector(Math.cos(theta), Math.sin(theta));
    }
```

七、 未來展望

希望可以改進物件不穩定的情形，並增加特效，或將 AR 改進成 Makerless 的版本，可不受限地點使用。

八、 參考文獻

https://developer.vuforia.com/support --Vuforia 相關參考資料
https://github.com/pistatium/mahougen --pistatium/mahougen
https://developer.vuforia.com/forum/faq/android-how-do-i-add-textures-my-model
-- bitmap to Texture
https://stackoverflow.com/questions/2801116/converting-a-view-to-bitmap-without-displaying-it-in-android
--getBitmapFromView

九、 Github 連結

此專案 source code 同步公布於 github 上，以下為連結:
https://github.com/pandatomlcc/MahougenAR-V2