



# Be a Musician!

組員: 謝奇容、黃微淳



# 目錄

CONTENTS

- 01 創作動機
- 02 程式說明
- 03 功能介紹
- 04 程式碼
- 05 團隊分工

# 01

## 創作動機

PART ONE



# 創作動機

---



近年來，市面上充斥許多形形色色的音樂節奏遊戲，獨特的遊玩模式與關卡制度，讓許多玩家沉迷其中。但是在那些遊戲過程中，並不會真的學到樂理或節奏的知識，我們希望能透過這款App，讓遊玩的玩家能學習到更多樂理知識，也讓管樂團、管弦樂團的團員們，能利用空閒時間，練習自己較不熟的段落，節省練習樂器的時間。

# 02

## 程式說明

### PART TWO



## 未來性

現在行動裝置的普及，再加上疫情影響，對於樂團團員來說，無法大家聚在一起練習，就能透過這個**App**在家加強自己不熟的段落，這樣疫情結束後，能很快上手，開始練習。其餘普通玩家也能利用這款**App**，在家學習樂理、節奏，加強自己的音樂能力，引發學習音樂的熱忱。

## 創意性

將使用管樂團及管弦樂團的曲目，讓玩家能成為樂團中的一員，跟著樂團一起演奏樂曲。

## 可行性

我們想使用圖片按鈕來當成點擊鍵，樂譜部分使用圖片，中間指揮則是使用影片撥放。

# 03

## 功能介紹

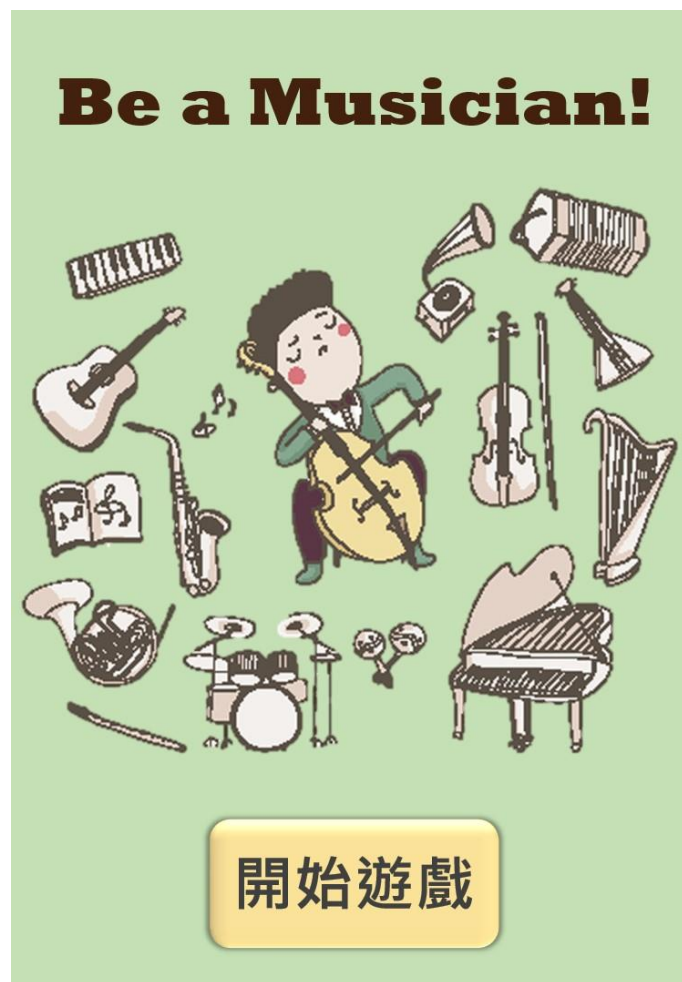
PART THREE



# 功能介紹

首頁

選擇介面



## ● 選擇樂器





# 功能介紹

首頁

選擇介面

**Be a Musician!**



開始遊戲

**Be a Musician!**

樂器



曲目



紅蓮華



難易度



簡單

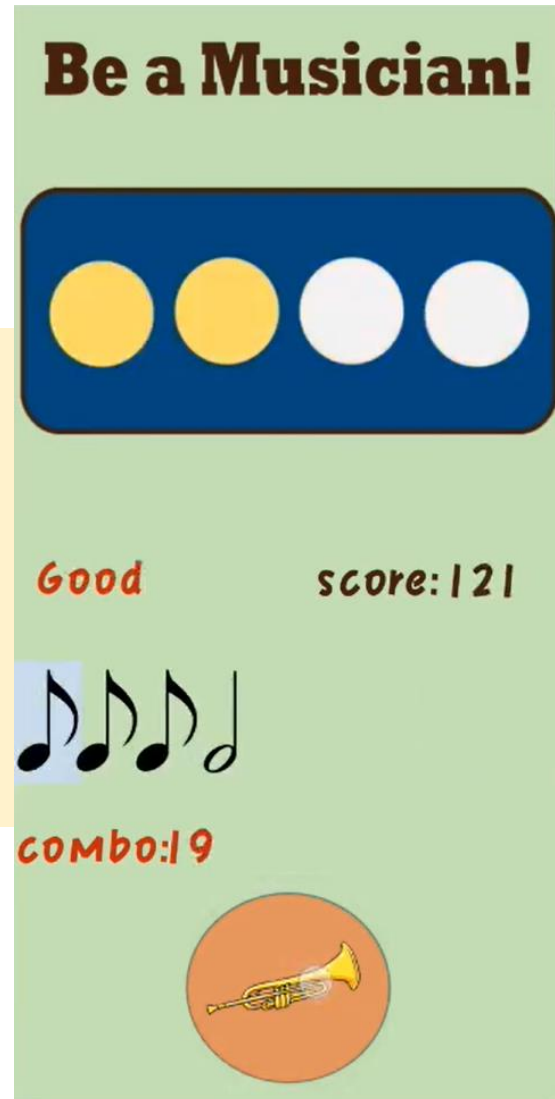


**Start**



- 選擇樂器
- 選擇曲目
- 選擇難易度

## 遊戲畫面



- 動態樂譜
- 上方為節拍器
- 玩家跟著音樂按按鈕打擊
- 判斷玩家打擊時機好壞顯示分數

## 未來可增加功能



- 放入指揮影片
- 樂理小教室
- 多人連線演奏
- 抽取特殊樂器

# 04

## 程式碼

### PART FOUR



```
private Handler mHandler = new Handler();  
private Handler mHandler2 = new Handler();
```

利用Handler控制update( )

```
double[][] music2TrumpetNotes = {music2TrumpetHardNotes,music2TrumpetNormalNotes};  
double[][] music2TrumpetSounds = {music2TrumpetHardNotesSound,music2TrumpetNormalSounds};  
double[][] music2BaritoneNotes = {music2BaritoneHardNotes};  
double[][] getMusic2BaritoneSounds = {};  
double[][][] music2Notes = {music2TrumpetNotes,music2BaritoneNotes};  
double[][][] music2Sounds = {music2TrumpetSounds,getMusic2BaritoneSounds};  
double[][][][] allMusicNotes = {music2Notes};  
double[][][][] allMusicSounds = {music2Sounds};  
int[] allSounds = {R.raw.trumpet2};  
double sec;
```

```
mHandler.post(runnable);
```

啟動runnable( )

```
mHandler2.postDelayed(runnable2, delayMillis: 46500);
```

```
private Runnable runnable = new Runnable() {  
    @Override  
    public void run() {
```

在Runnable裡面加入想要執行的事情

```
        try {  
            songPosition = (mediaPlayer.getCurrentPosition());           //音樂撥放到哪裡  
            songPositionInBeats = songPosition / (secPerBeat*1000);  
            int s = (int) ((songPositionInBeats+1)%4);  
            mImageView13.setImageResource(beatpicture[s]);  
            if(i<allMusicNotes[music][instrument][rank].length)  
            {  
                double m = allMusicNotes[music][instrument][rank][i]*secPerBeat*1000;  
                if(m-songPosition<150)                                   //若在音符後150沒有點擊，判定為miss  
                {  
                    mTextView1.setText("Miss");  
                    i+=1;  
                    combo=0;  
                    score-=4;  
                    miss+=1;  
                    if(score<0)  
                    {  
                        score=0;  
                    }  
                    mTextView2.setText(Integer.toString(score));  
                    mTextView3.setText("");  
                }  
            }  
        }  
    }  
}
```

```
if(combo>maxcombo)
```



```
if(i<allMusicNotes[music][instrument][rank].length)
{
    double m = allMusicNotes[music][instrument][rank][i]*secPerBeat*1000;
    if(m-songPosition<150) //若在音符後150沒有點擊，判定為miss
    {
        mTextView1.setText("Miss");
        i+=1;
        combo=0;
        score-=4;
        miss+=1;
        if(score<0)
        {
            score=0;
        }
        mTextView2.setText(Integer.toString(score));
        mTextView3.setText("");
    }
}
if(combo>maxcombo)
{
    maxcombo = combo; //計算最大連擊數
}

mHandler.postDelayed(runnable, delayMillis: 1);
}
catch (Exception e)
```

每一毫秒執行一次runnable()

```

private Runnable runnable = new Runnable() {
    @Override
    public void run() {
        try {
            songPosition = (mediaPlayer.getCurrentPosition());           //音樂撥放到哪裡
            songPositionInBeats = songPosition / (secPerBeat*1000);
            int s = (int) ((songPositionInBeats+1)%4);
            mImageView13.setImageResource(beatspicture[s]);
            if(i<allMusicNotes[music][instrument][rank].length)
            {
                double m = allMusicNotes[music][instrument][rank][i]*secPerBeat*1000;
                if(m-songPosition<150)                                     //若在音符後150沒有點擊，判定為miss
                {
                    mTextView1.setText("Miss");
                    i+=1;
                    combo=0;
                    score-=4;
                    miss+=1;
                    if(score<0)
                    {
                        score=0;
                    }
                    mTextView2.setText(Integer.toString(score));
                    mTextView3.setText("");
                }
            }
        }
    }
}
if(combo>maxcombo)

```

## 節拍器



```
private Runnable runnable = new Runnable() {  
    @Override  
    public void run() {
```

```
        try {  
            songPosition = (mediaPlayer.getCurrentPosition());           //音樂撥放到哪裡  
            songPositionInBeats = songPosition / (secPerBeat*1000);  
            int s = (int) ((songPositionInBeats+1)%4);  
            mImageView13.setImageResource(beatspicture[s]);  
            if(i<allMusicNotes[music][instrument][rank].length)  
            {  
                double m = allMusicNotes[music][instrument][rank][i]*secPerBeat*1000;  
                if(m-songPosition<150)                                     //若在音符後150沒有點擊，判定為miss  
                {  
                    mTextView1.setText("Miss");  
                    i+=1;  
                    combo=0;  
                    score-=4;  
                    miss+=1;  
                    if(score<0)  
                    {  
                        score=0;  
                    }  
                    mTextView2.setText(Integer.toString(score));  
                    mTextView3.setText("");  
                }  
            }  
        }  
    }  
}
```

利用Runnable( )  
完成按鈕點擊判斷

```
private Runnable runnable2 = new Runnable() {  
    @Override  
    public void run() {
```

```
        try {
```

```
            ImageView[] image = {mImageView, mImageView1, mImageView2, mImageView3, mImageView4, mImageView5, mImageView6, mImageVi
```

```
            image[0].setImageResource(mImageArray[music2TrumpetNormalImg[x]]);
```

```
            if (y > music2TrumpetNormalImg.length) {
```

```
                mHandler2.removeCallbacks( r: this); //停止
```

```
            } else {
```

```
                mHandler2.postDelayed( r: this, delayMillis: 300); //delay 0.5拍
```

```
                if (x < image.length) //九張圖片還沒滿前
```

```
                {
```

```
                    for (y = x; y > 0; y--) {
```

```
                        z = x;
```

```
                        image[y].setImageResource(mImageArray[music2TrumpetNormalImg[z - y]]);
```

```
                        z--;
```

```
                    }
```

```
                } else {
```

```
                    for (y = 1; y < 9; y++) {
```

```
                        image[y].setImageResource(mImageArray[music2TrumpetNormalImg[x - y]]); //圖片全部往左移一格
```

```
                    }
```

```
                }
```

```
                x++;
```

```
            }
```

```
        }
```

```
    }
```

利用Runnable( )、Handler( )做音符動畫

## 利用SoundPool( ) 播放按鈕音效

```
SoundPool snd = new SoundPool( maxStreams: 10, AudioManager.STREAM_MUSIC, srcQuality: 5);  
int sound = snd.load( context: PlaySong.this, allSounds[instrument], priority: 1);
```

```
mHandler.post(runnable); //啟動runnable()  
mHandler2.postDelayed(runnable2, delayMillis: 46500);
```

```
mImageButton.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        if(i<allMusicNotes[music][instrument][rank].length) //若音符還沒打擊完  
        {  
            double m = allMusicNotes[music][instrument][rank][i]*secPerBeat*1000;  
            if(m-songPosition > 400 && m-songPosition < 500) //m和songPosition大概誤差550左右，在音符的前50~後50時間內點擊，判定為Great  
            {  
                snd.play(sound, leftVolume: 1, rightVolume: 1, priority: 1, loop: 0, (float) allMusicSounds[music][instrument][rank][i]);  
                mTextView1.setText("Great");  
                combo+=1;  
                score+=5;  
                great+=1;  
                mTextView2.setText(Integer.toString(score));  
                mTextView3.setText(Integer.toString(combo));  
                i+=1;  
            }  
        }  
    }  
});
```

```

double[][] music2TrumpetSounds = {music2TrumpetHardNotesSound,music2TrumpetNormalSounds};
double[][] music2BaritoneNotes = {music2BaritoneHardNotes};
double[][] getMusic2BaritoneSounds = {};
double[][][] music2Notes = {music2TrumpetNotes,music2BaritoneNotes};
double[][][] music2Sounds = {music2TrumpetSounds,getMusic2BaritoneSounds};
double[][][][] allMusicNotes = {music2Notes};
double[][][][] allMusicSounds = {music2Sounds};
int[] allSounds = {R.raw.trumpet2};
double sec;
MediaPlayer mediaPlayer; //放音樂
TextView mTextView1,mTextView2,mTextView3,mTextView4; //顯示打擊等級,分數,combo數
ImageButton mImageButton; //打擊按鈕
ImageView mImageView,mImageView1,mImageView2,mImageView3,mImageView4,mImageView5,mImageView6,mImageView7,mImageView8,mIm
String[] musicLink = {"https://firebasestorage.googleapis.com/v0/b/finalproject10902.appspot.com/o/Colors%20of%20the%20v"}

```

## 從firebase讀取歌曲音樂

```

int instrument;
int rank;
int music;

Bundle bundleToResult = new Bundle();

```

# PART FOUR



## 05

## 團隊分工



# 團隊分工



謝奇容

蒐集資料、寫程式、報告

黃微淳



蒐集資料、寫程式、介面設計、ppt製作



謝謝大家~