Простой диалог выбора цвета для настроек

Не давно мне понадобилось в своем приложении предоставить пользователю в настройках (**preferences**) выбор цвета для какого либо элемента. Немного поискав я не нашел ничего что работало бы по android 2.1 и выше поэтому быстренько набросал свой диалог выбора цвета.  
За основу был взят пример google [ColorPickerDialog.html](http://developer.android.com/resources/samples/ApiDemos/src/com/example/android/apis/graphics/ColorPickerDialog.html)  
  
Пример google был немного упрощен и вот что получилось в начале

import android.content.Context;

import android.graphics.\*;

import android.view.MotionEvent;

import android.view.View;

public class ColorPickerView extends View {

public interface OnColorChangedListener {

void colorChanged(int color);

}

private final Paint mPaint;

private final Paint mCenterPaint;

private final int[] mColors;

private final OnColorChangedListener colorChangedListener;

ColorPickerView(Context c, int color, OnColorChangedListener l) {

super(c);

colorChangedListener = l;

mColors = new int[]{

0xFFFF0000, 0xFFFF00FF, 0xFF0000FF, 0xFF00FFFF, 0xFF00FF00,

0xFFFFFF00, 0xFFFF0000

};

Shader s = new SweepGradient(0, 0, mColors, null);

mPaint = new Paint(Paint.ANTI\_ALIAS\_FLAG);

mPaint.setShader(s);

mPaint.setStyle(Paint.Style.STROKE);

mPaint.setStrokeWidth(32);

mCenterPaint = new Paint(Paint.ANTI\_ALIAS\_FLAG);

mCenterPaint.setColor(color);

mCenterPaint.setStrokeWidth(5);

}

@Override

protected void onDraw(Canvas canvas) {

float r = CENTER\_X - mPaint.getStrokeWidth() \* 0.5f;

canvas.translate(CENTER\_X, CENTER\_X);

canvas.drawOval(new RectF(-r, -r, r, r), mPaint);

canvas.drawCircle(0, 0, CENTER\_RADIUS, mCenterPaint);

}

@Override

protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {

setMeasuredDimension(CENTER\_X \* 2, CENTER\_Y \* 2);

}

private static final int CENTER\_X = 100;

private static final int CENTER\_Y = 100;

private static final int CENTER\_RADIUS = 32;

private int ave(int s, int d, float p) {

return s + java.lang.Math.round(p \* (d - s));

}

private int interpColor(int colors[], float unit) {

if (unit <= 0) {

return colors[0];

}

if (unit >= 1) {

return colors[colors.length - 1];

}

float p = unit \* (colors.length - 1);

int i = (int) p;

p -= i;

// now p is just the fractional part [0...1) and i is the index

int c0 = colors[i];

int c1 = colors[i + 1];

int a = ave(Color.alpha(c0), Color.alpha(c1), p);

int r = ave(Color.red(c0), Color.red(c1), p);

int g = ave(Color.green(c0), Color.green(c1), p);

int b = ave(Color.blue(c0), Color.blue(c1), p);

return Color.argb(a, r, g, b);

}

private static final float PI = 3.1415926f;

@Override

public boolean onTouchEvent(MotionEvent event) {

float x = event.getX() - CENTER\_X;

float y = event.getY() - CENTER\_Y;

switch (event.getAction()) {

case MotionEvent.ACTION\_MOVE:

float angle = (float) java.lang.Math.atan2(y, x);

// need to turn angle [-PI ... PI] into unit [0....1]

float unit = angle / (2 \* PI);

if (unit < 0) {

unit += 1;

}

mCenterPaint.setColor(interpColor(mColors, unit));

colorChangedListener.colorChanged(mCenterPaint.getColor());

invalidate();

break;

}

return true;

}

}

Далее нам необходимо реализовать свой диалог выбора цвета для этого реализуем ColorPreference наследником от DialogPreference

import android.app.AlertDialog;

import android.content.Context;

import android.content.SharedPreferences;

import android.preference.DialogPreference;

import android.util.AttributeSet;

import android.view.LayoutInflater;

import android.widget.LinearLayout;

import ru.jobdone.cow.R;

import ru.jobdone.cow.activity.game.GameActivity;

public class ColorPreference extends DialogPreference {

public static final String COLOR\_PREFERENCE\_FILE\_NAME = "colorPreference";

private final Context context;

private int color = 0;

public ColorPreference(Context context, AttributeSet attrs){

super(context, attrs);

this.context=context;

}

@Override

protected void onPrepareDialogBuilder(AlertDialog.Builder builder) {

super.onPrepareDialogBuilder(builder);

color = context.getSharedPreferences(COLOR\_PREFERENCE\_FILE\_NAME,0).getInt(GameActivity.WORLD\_COLOR\_PREFERENCE\_PARAM\_NAME,3);

ColorPickerView colorPickerView = new ColorPickerView(context, color, new ColorPickerView.OnColorChangedListener() {

@Override

public void colorChanged(int color) {

ColorPreference.this.color=color;

}

});

LayoutInflater inflater = LayoutInflater.from(context);

LinearLayout v = (LinearLayout)inflater.inflate(R.layout.color\_picker, null, false);

v.addView(colorPickerView);

builder.setView(v);

}

@Override

protected void onDialogClosed(boolean positiveResult) {

SharedPreferences.Editor edit1 = context.getSharedPreferences(COLOR\_PREFERENCE\_FILE\_NAME, 0).edit();

edit1.putInt(GameActivity.WORLD\_COLOR\_PREFERENCE\_PARAM\_NAME,color);

System.out.println(edit1.commit());

super.onDialogClosed(positiveResult);

}

}

Теперь для того чтобы добавить диалог выбора цвета в настройки просто добавляем в preference.xml

<ваш.пакет.ColorPreference android:title="Цвет слов в игре"

android:key="color\_game\_world"

android:summary="Выберите цвет, которым будут отображаться слова"

android:dialogTitle="Выберите цвет"

/>

Так же наобходимо добавить в layout следующий файл color\_picker.xml  
Он необходим чтобы выровнять ColorPickerView по центру диалога

<linearlayout android:gravity="center" android:id="@+id/colorPickerLayout" android:layout\_gravity="center\_horizontal" android:layout\_height="fill\_parent" android:layout\_width="fill\_parent" xmlns:android="http://schemas.android.com/apk/res/android"/>

Для того чтобы прочитать выбранный цвет пишем следующий код

SharedPreferences sharedPreferences = getSharedPreferences(ColorPreference.COLOR\_PREFERENCE\_FILE\_NAME, textDefaultColor);

int color = sharedPreferences.getInt(WORLD\_COLOR\_PREFERENCE\_PARAM\_NAME, textDefaultColor);

Вот и все