

ステレオカメラ画像表示 Androidアプリ 最終発表

112947H 高橋昌史

計画発表時の全体構想

FPGAボード

Wi-Fiで通信

Androidアプリ

ステレオカメラ

画像を表示

左

右

計画発表時の全体構想

STEP 1 Android内蔵カメラの画像を表示するアプリの作成

STEP2 カメラ入力のI/Fの設計

STEP3 通信側とシステムの統合

STEP4 アプリ内で表示する画像をステレオカメラにする

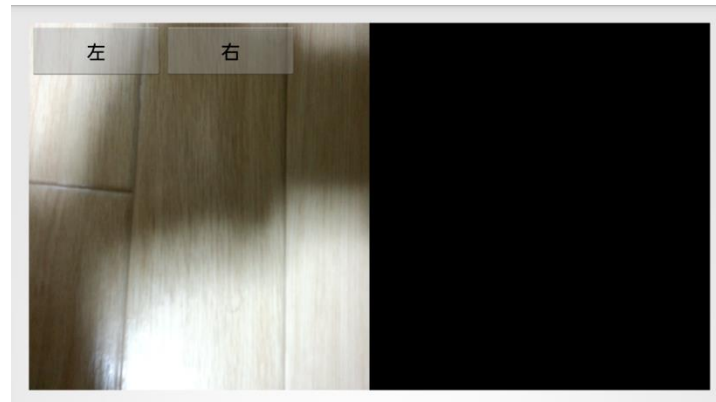
計画に対する到達度と進捗 中間発表段階

STEP1 Android内蔵カメラの画像を表示するアプリの作成



内蔵カメラの1画面のみを表示する試作アプリの作成

↑ 中間発表段階で完成。



計画に対する到達度と進捗 中間発表段階

STEP2 カメラ入力のI/Fの設計



FPGAのメソッドを呼び出した際カメラの動作をするオブジェクトをjavaで作成(FPGAの設計は時間的に難しい)
また、Androidアプリの改良

↑ 最終発表段階で完成

計画に対する到達度と進捗 最終発表段階

STEP3 通信側とシステムの統合

↑ 達成できず

STEP4 アプリ内で表示する画像をステレオカメラに

↑ 10%程度達成か

計画に対する到達度と進捗 最終発表段階

FPGAの設計や通信側との統合ができなかったり、
アプリ内で表示する画像をステレオカメラには
できなかったなので、達成度は50%～55%ほど。

当初計画との比較 中間発表まで

回数	日付	開発計画時	中間発表段階	実際
第4回	10/24	計画発表		計画発表
第5回	10/31	アプリ開発		調査
第6回	11/7	アプリ開発		調査 アプリ開発
第7回	11/14	アプリ開発 動作テスト		調査 アプリ開発
第8回	11/21	FPGA設計		調査
第9回	12/5	FPGA設計		調査 オブジェクトの作成
第10回	12/12	FPGA設計		パワーポイント作成
第11回	12/19	中間発表	中間発表	中間発表

当初計画との比較

中間発表から最終発表まで

回数	日付	開発計画時	中間発表段階	実際
第11回	12/19	中間発表	中間発表	中間発表
第12回	12/26	通信側との統合	オブジェクトの作成 Androidアプリの改良	調査 オブジェクトの作成
第13回	1/9	通信側との統合	オブジェクトの作成 Androidアプリの改良	Androidアプリの改良
第14回	1/16	通信側との統合 動作テスト	オブジェクトの作成 Androidアプリの改良	Androidアプリの改良
第15回	1/27	最終発表	最終発表	最終発表

開発の分量

- ・ファイル数

ButtonActivity.java

Jpeg.java

YUV.java

の3つjavaのファイルと

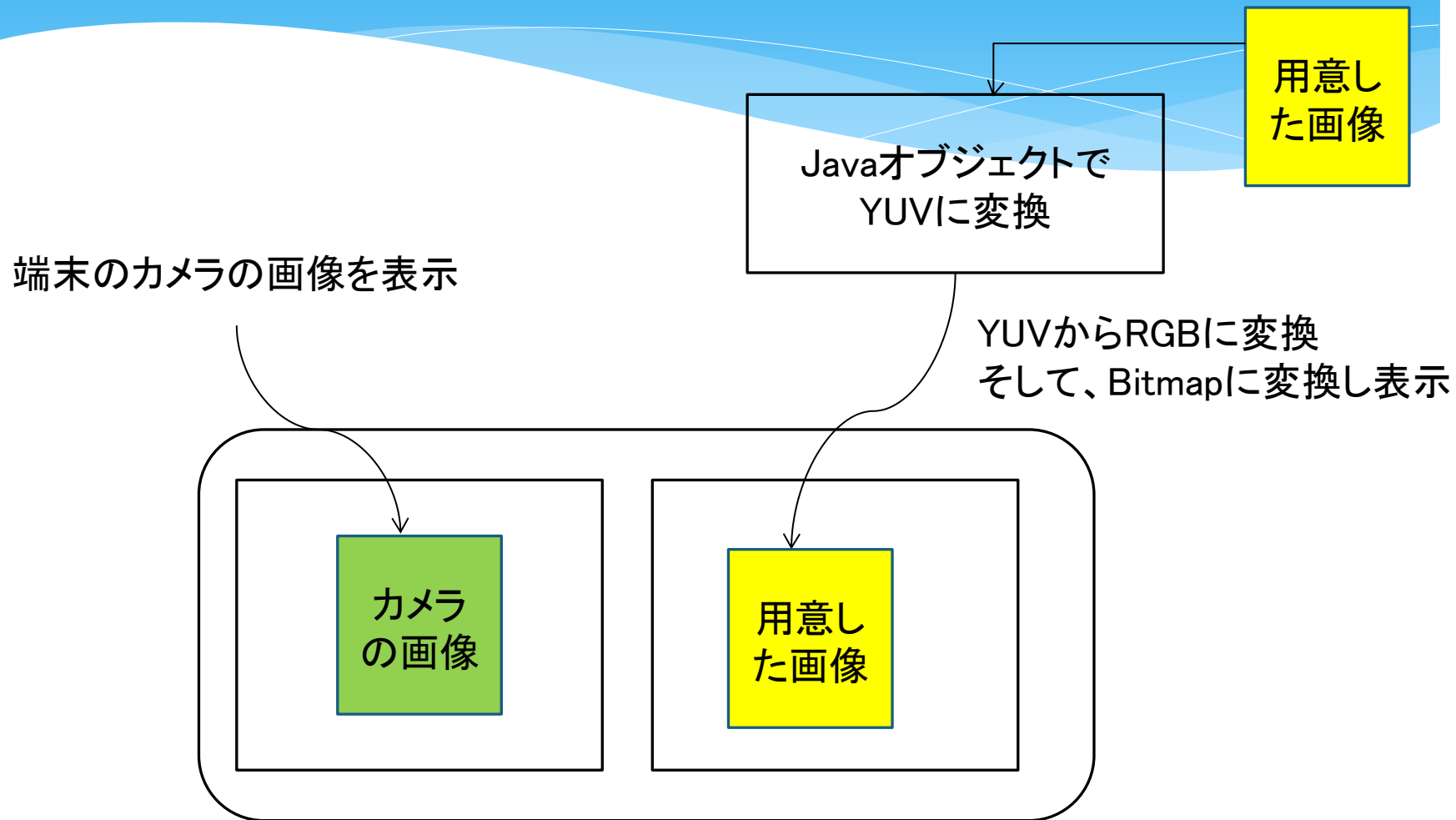
activity_button.xml

のレイアウトファイル

- ・ソースコード行数 302行

- ・文字数 6705文字

アプリの画面



アプリの画面

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
        android:orientation="horizontal"
        android:gravity="center_vertical" >

    <SurfaceView
        android:id="@+id/surface_view"
        android:layout_width="326.4dp"
        android:layout_height="244.8dp" />

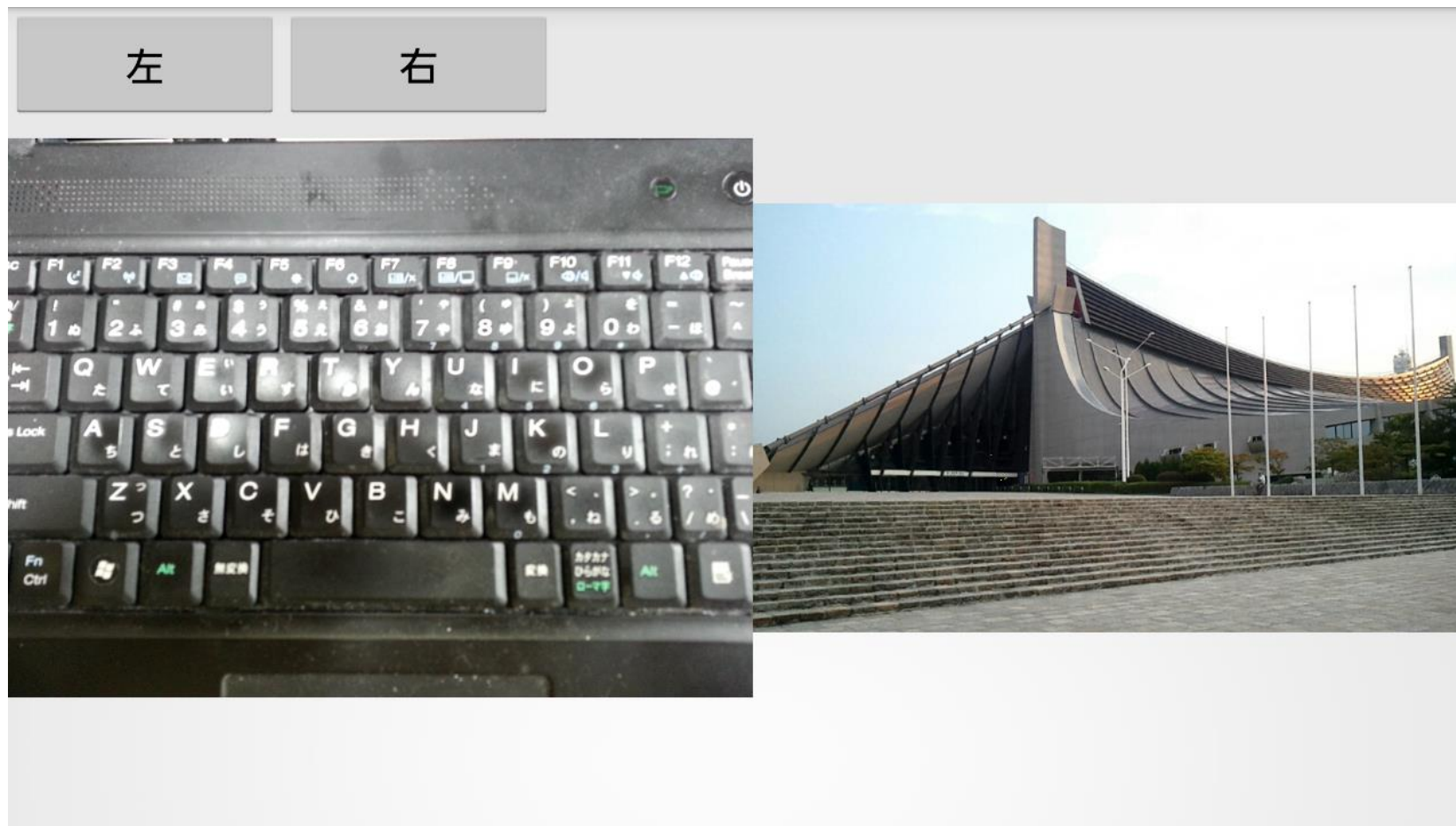
    <ImageView
        android:id="@+id/image_view"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:contentDescription="@string/imagecircle"/>

</LinearLayout>
```

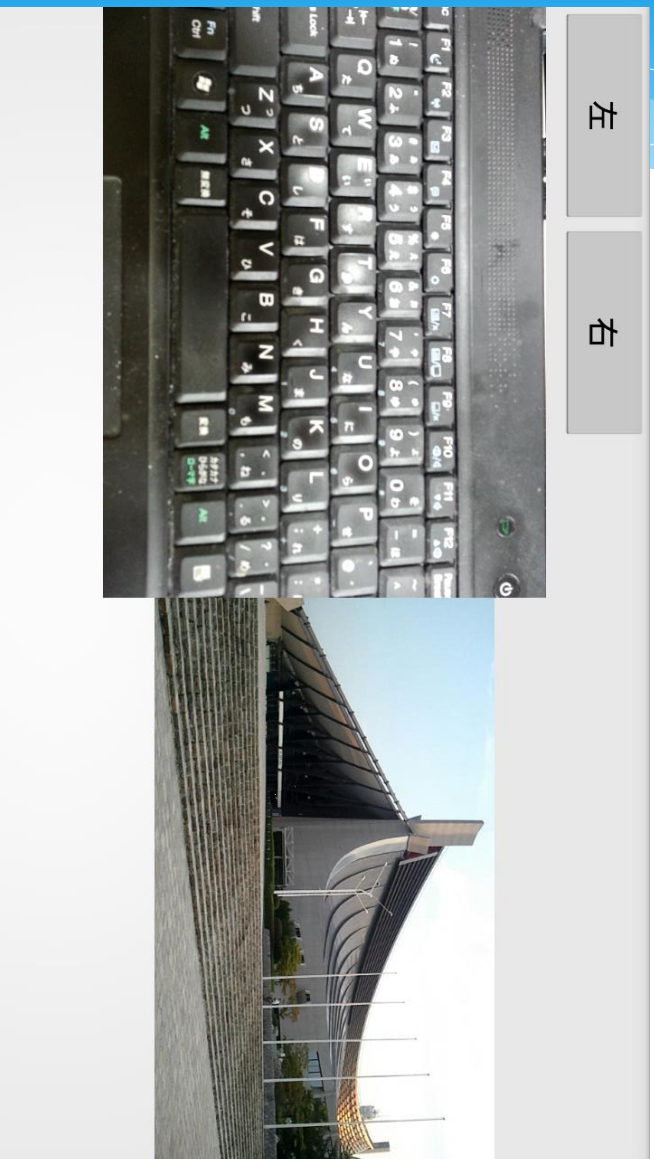
カメラの解像度が 3264×2448 だったため、SurfaceViewを $1/10$ の 326.4×244.8 にし、1:1表示をやめたことによりアスペクト比が正常に。

また、`android:gravity="center_vertical"`によりSurfaceViewとImageViewを中央に表示。

横向き



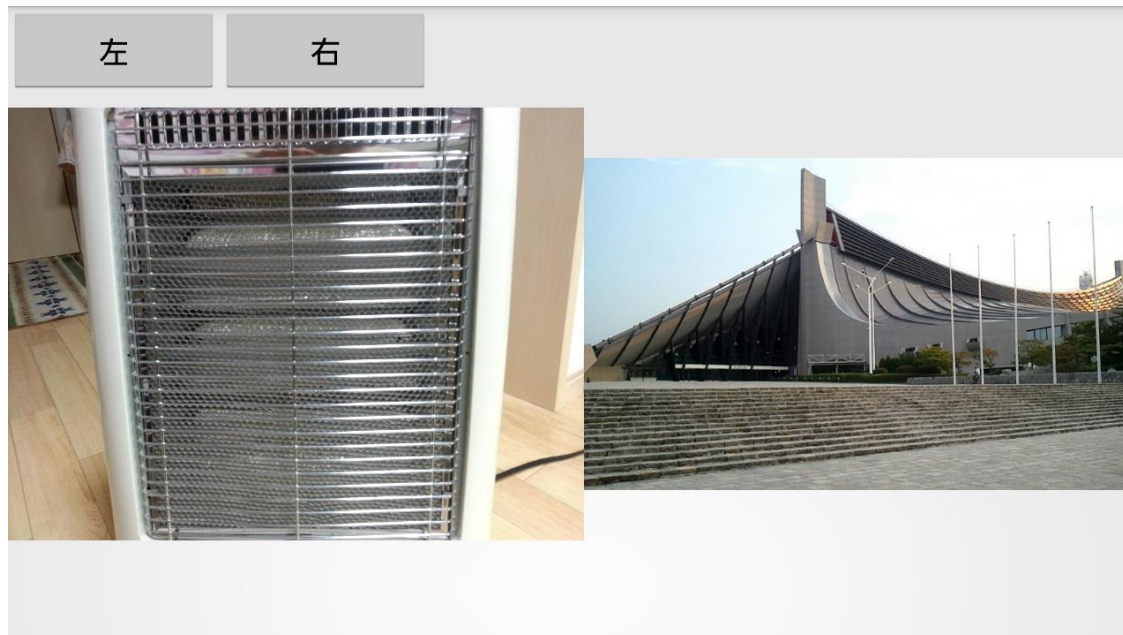
縦向き



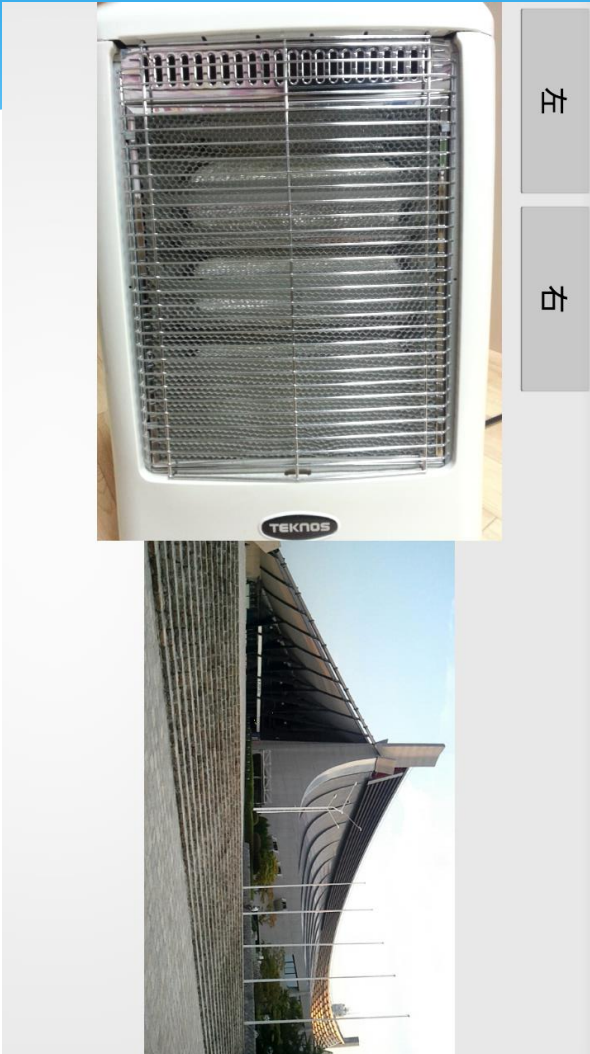
前回



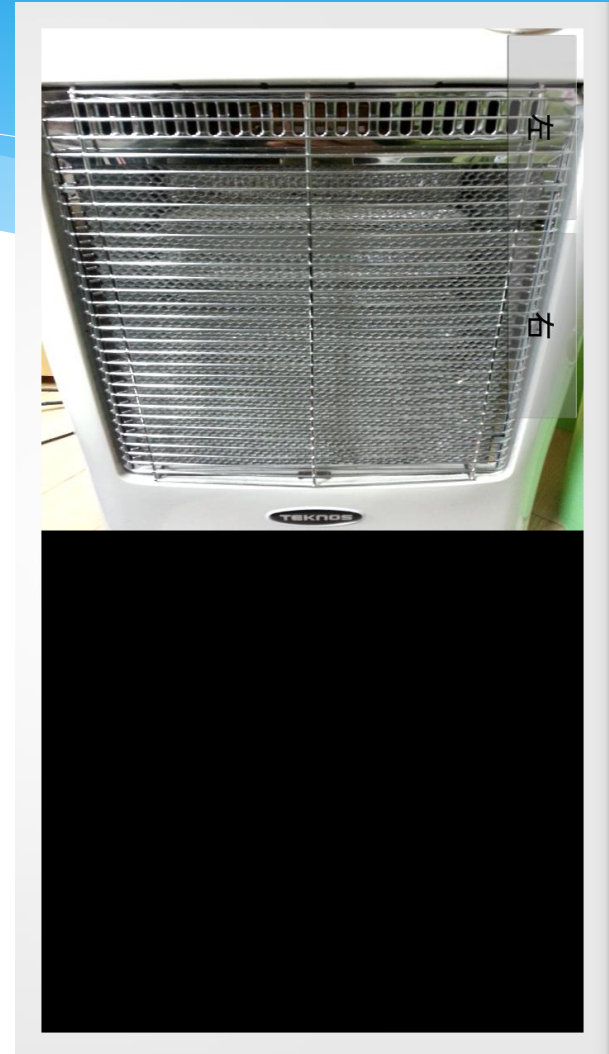
今回



前回



今回



アプリの画面

手動でSurfaceViewの幅を設定しているため、端末によっていちいち変える必要がある。(今回の端末はSC-03E)
surfaceChanged内で変更することで自動的にできる可能性

また、ボタンでの画面変更は実装できなかった

機能仕様 最終版

クラス一覧

アクセス	クラス名	extends(継承)	implements(実装)
public	ButtonActivity	Activity	OnClickListener
public	Jpeg		
public	YUV		

ButtonActivityは主にカメラの表示と画像の表示

JpegクラスはYUVからRGBに変換

YUVクラスはRGBからYUVに変換

機能仕様 最終版

ButtonActivityクラスフィールド一覧

アクセス	型名	フィールド名	機能
private	Camera	camera	カメラ
private	SurfaceView	mySurfaceView	カメラ用サーフェイスビュー
private	SurfaceHolder.Callback	mSurfaceListener	カメラのプレビューを表示するSurfaceViewのリスナー
private	Bitmap	bitmap	読み込む画像
private	Bitmap	b	表示する画像

機能仕様 最終版

ButtonActivityクラスメソッド一覧 (1)

アクセス	型名	メソッド名	機能
public	void	surfaceDestroyed(SurfaceHolder holder)	カメラを停止する パラメータ: SurfaceHolder holder: SurfaceHolderの取得
public	void	surfaceCreated(SurfaceHolder holder)	カメラを起動する ・インターフェイス初期化 ・ボタンリスナー設定 ・リモコンコード設定 パラメータ: SurfaceHolder holder: SurfaceHolderの取得
public	void	surfaceChanged(SurfaceHolder holder, int format, int width, int height)	サイズ変更 パラメータ: SurfaceHolder holder: SurfaceHolderの取得 int format: ピクセルフォーマット int width: 横のサイズ int height: 縦のサイズ
public	void	onClick(View v)	各ボタンのクリック処理 パラメータ: View v: ビュー

機能仕様 最終版

ButtonActivityクラスメソッド一覧 (2)

アクセス	型名	メソッド名	機能
public	void	onCreate(Bundle savedInstanceState)	初期化関数実行 パラメータ: SurfaceHolder holder: SurfaceHolderの取得 Bundle savedInstanceState: インスタンスの状態 Button button1: ボタンの設定 Button button2: ボタンの設定 int width: 横幅 int height: 縦幅 int[] rgb: 変換してきたRGB Jpeg jp: インスタンス変数 ImageView imageView: ImageViewの設定
public	void	showMessage(String msg)	トースト設定 パラメータ: String msg: メッセージ
public	int	widthSet()	横幅セット 戻り値: bitmap.getWidth()
public	int	heightSet()	縦幅セット 戻り値: bitmap.getHeight()
public	int[]	rgbSet()	画像のピクセル値を得る int width: 横幅 int height: 縦幅 int[] rgb: ピクセル値を格納する配列 戻り値: rgb

機能仕様 最終版

Jpegクラスメソッド一覧

アクセス	型名	メソッド名	機能
public	int□	Imageset()	RGBを求める 戻り値:rgb

機能仕様 最終版

YUVクラスメソッド一覧

アクセス	型名	メソッド名	機能
public	int□	YuvPix()	YUVを求める 戻り値:yuv

今後の抱負

- ・これからやろうとすることに対して、自分の知識量や開発経験も考えつつスケジュールを立てなければ予定通り行かない
- ・開発経験が無い言語で、本などで見ただけではイメージが湧かなかったが、実際にソースを色々書いて動かしていくとだんだんと分かってきた
- ・JavaやAndroidプログラミングは全くやったことがなかったので、当初の目的通り以前よりは多少はスキルアップできたと思う
- ・将来仕事でプロジェクトを進める際はスケジュール管理をきちんとしたい

開発端末

今回使用した端末は

- Nexus 7 (Android 4.3)
- Galaxy S III α (Android 4.1.1)