

AppInventor を用いたスマホ（android 端末）アプリの開発

(I) 準備

- (1) Google Account を取る。
- (2) 最新 Java をインストール
- (3) (MIT)から AppInventor をダウンロードしてインストール

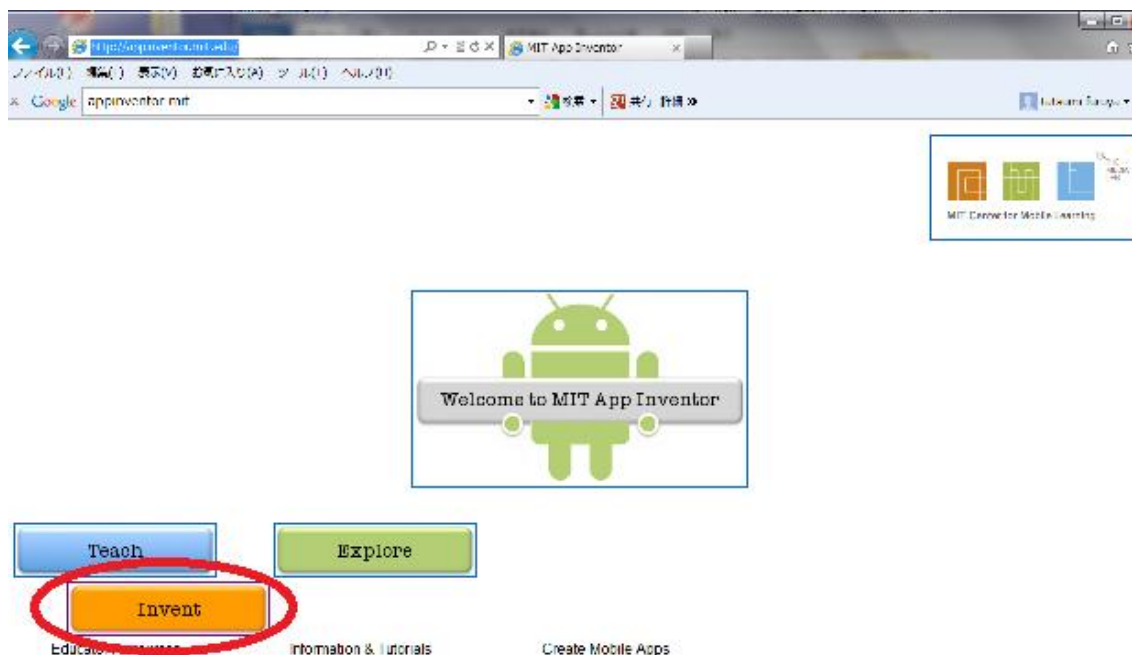
(II) アプリ開発の手順

アプリ開発は3ステップで行われる

準備

<http://appinventor.mit.edu/> (google で mit appinventor でトップ表示)

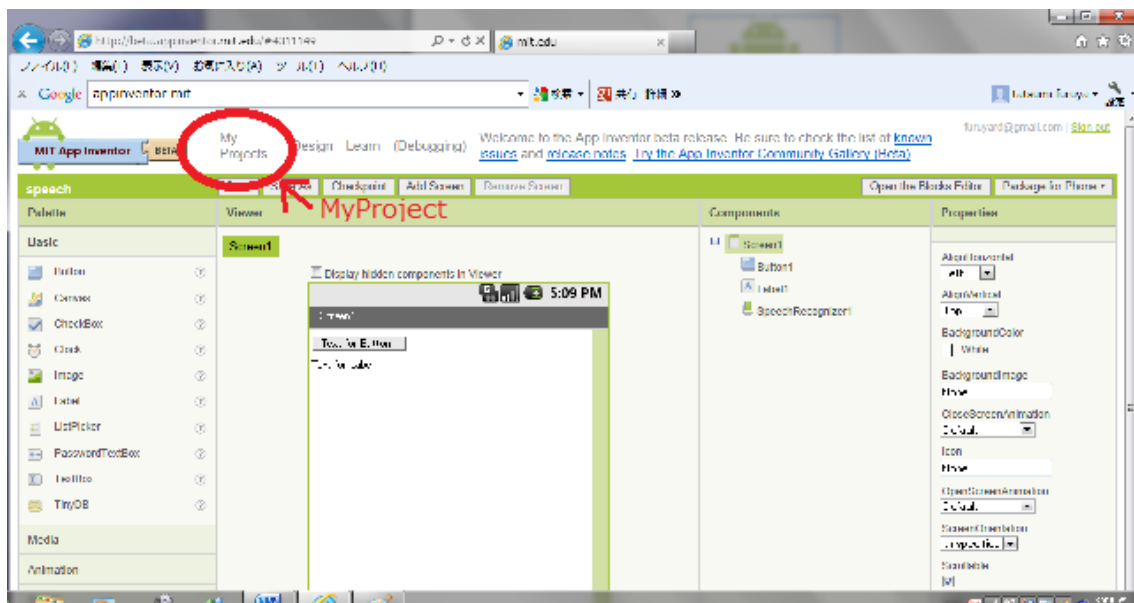
へ行き「Invent」ボタンを押す。



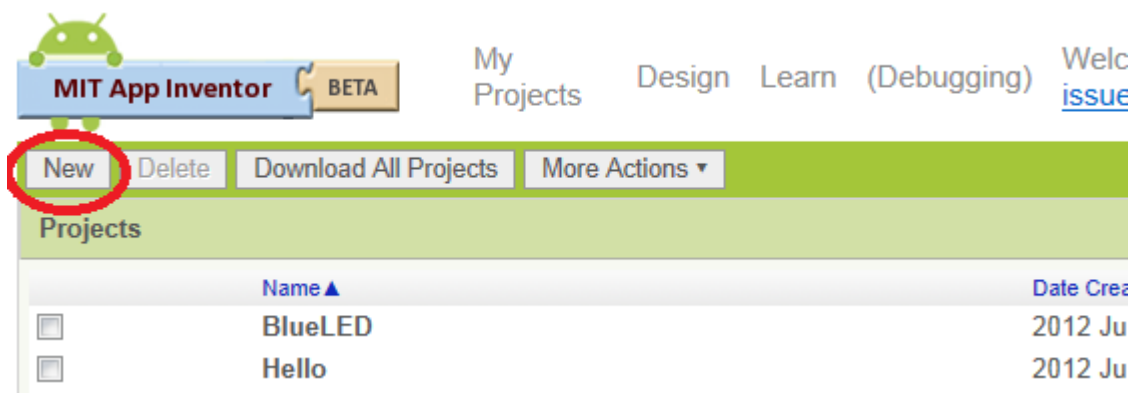
開発ステップ1：スマホの画面設計

(どこに図を表示するか、どこにボタンを配置するかなど)

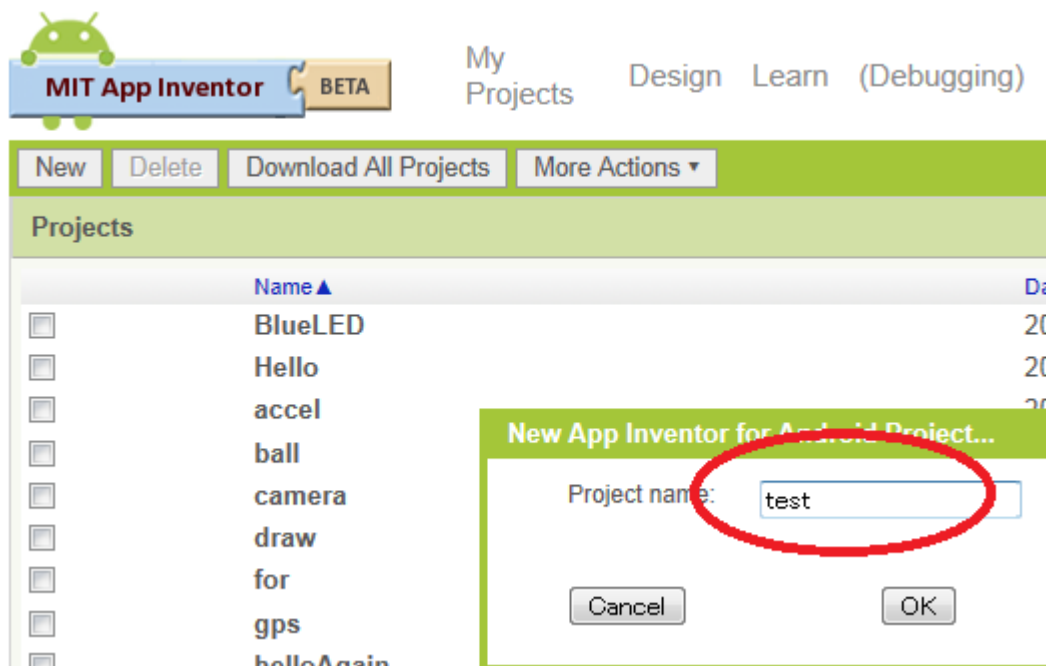
次の画面が出るので、「MyProject」をクリック



次の画面に変わり「New」というボタンが現れるのでクリック



プロジェクト（アプリ）名を決めてくれ、と言ってくるので記入

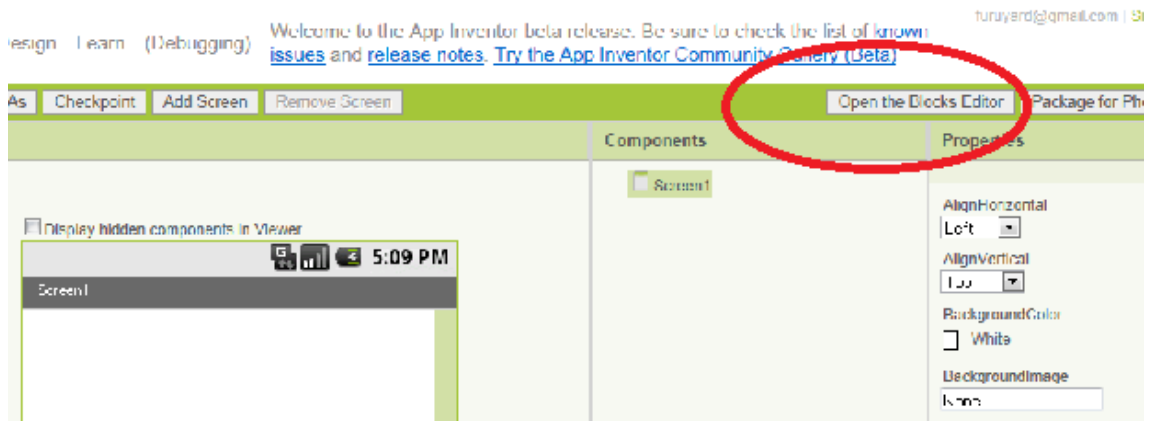


ステップ 1 : 画面設計

次の画面設計画面が現れる



設計が終わったら「Open the Blocks Editor」をクリックしてプログラミング画面に移る

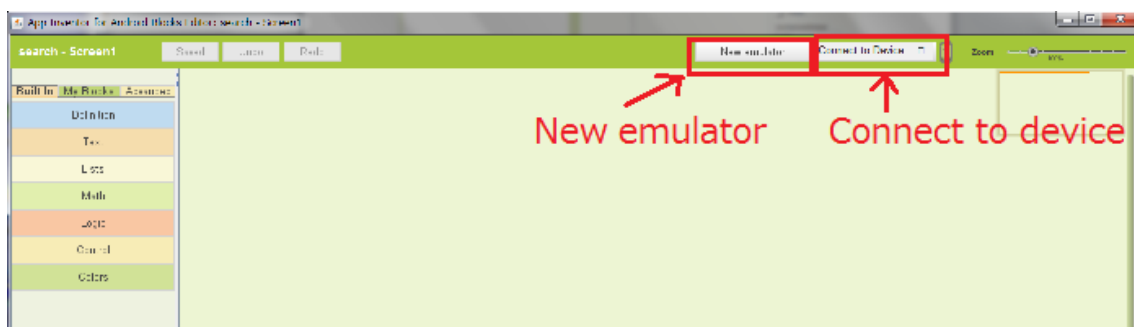


この時、ファイルを開くか聞いてくるので、「開く」を指示。

ステップ2：プログラム作成



作成が終わったら「New emulator」をクリックし、テスト画面に移る



ステップ3：エミュレータでテスト

次の画面が現れるので、「OK」をクリック



しばらくしてスマホ画面が青くなったら、「Connect to Device」で「5554」（これが左のエミュレータ）を選ぶ。これで、作成したアプリが動き出す。

(III) サンプルプログラム集

本章の目次

(1)基本系(Basic)

ボタン、テキストボックス、レーベル、を使った例。

B：ボタンを押したらシリーズ

B1：文字列がでる1 (Label を使う) (注：Label は出力表示専用)

B2：文字列がでる2 (TextBox を使う) (注：TextBox は入出力可)

B3：計算をする (TxexBox に書いた2つの数字を加算)

(2)メディア系 Media

M1：画像がでる1

M2：画像がでる2

M3：音が出る

(3)センサー系 Sensor

S1：タッチパネルで絵を描く

S2：GPS 情報を表示

S3：加速度センサー応用1

S4：加速度センサー応用2

(4) 外注系 Outsourcing

O1 : 写真を撮る

O2 : 音声認識

(5) プログラミングの基本技

繰り返し、変数、条件分岐

P1 : for range による繰り返し 1

P2 : 繰り返し 2 (変数の使い方にも注目)

P3 : 条件分岐 1 (数当て)

P4 : 条件分岐 2 (文字列当て)

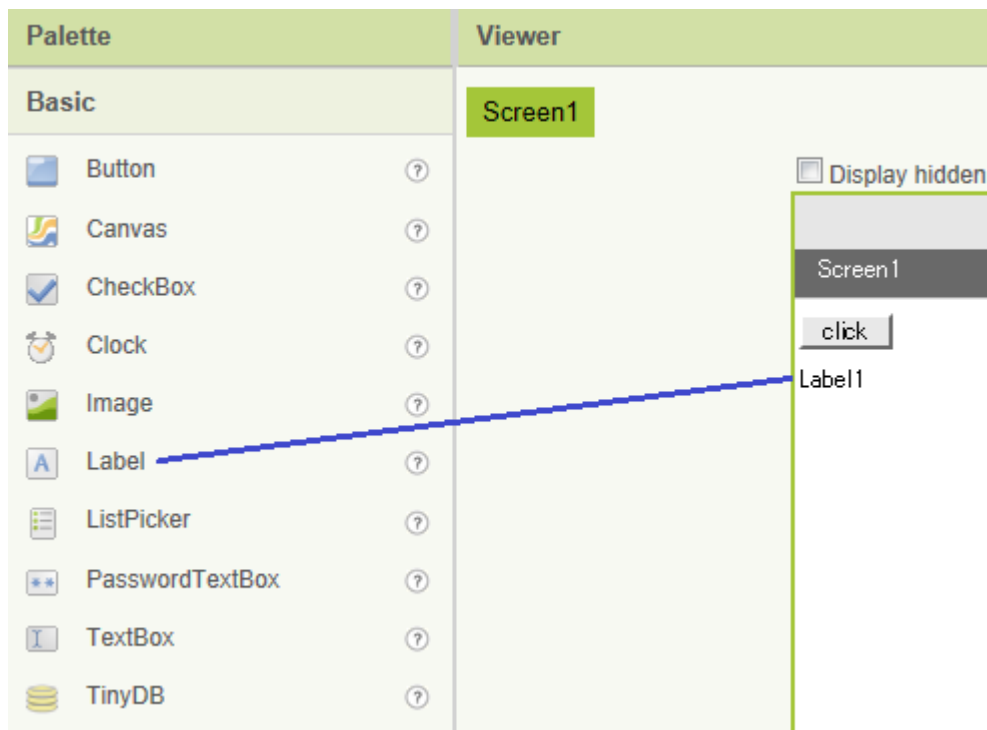
P5 : パラパラ漫画を作る

(1) 基本系(Basic)

(B1) ボタンを押すと、Label に hello と表示される。

画面設計

パレットから Button と Label を選び、Label の初期文字を消しておく。



プログラム

MyBlocks から次のブロックを持ってきてつなぐ。

Button1.Click は Button1 から、Label1.Text は Label1 から。

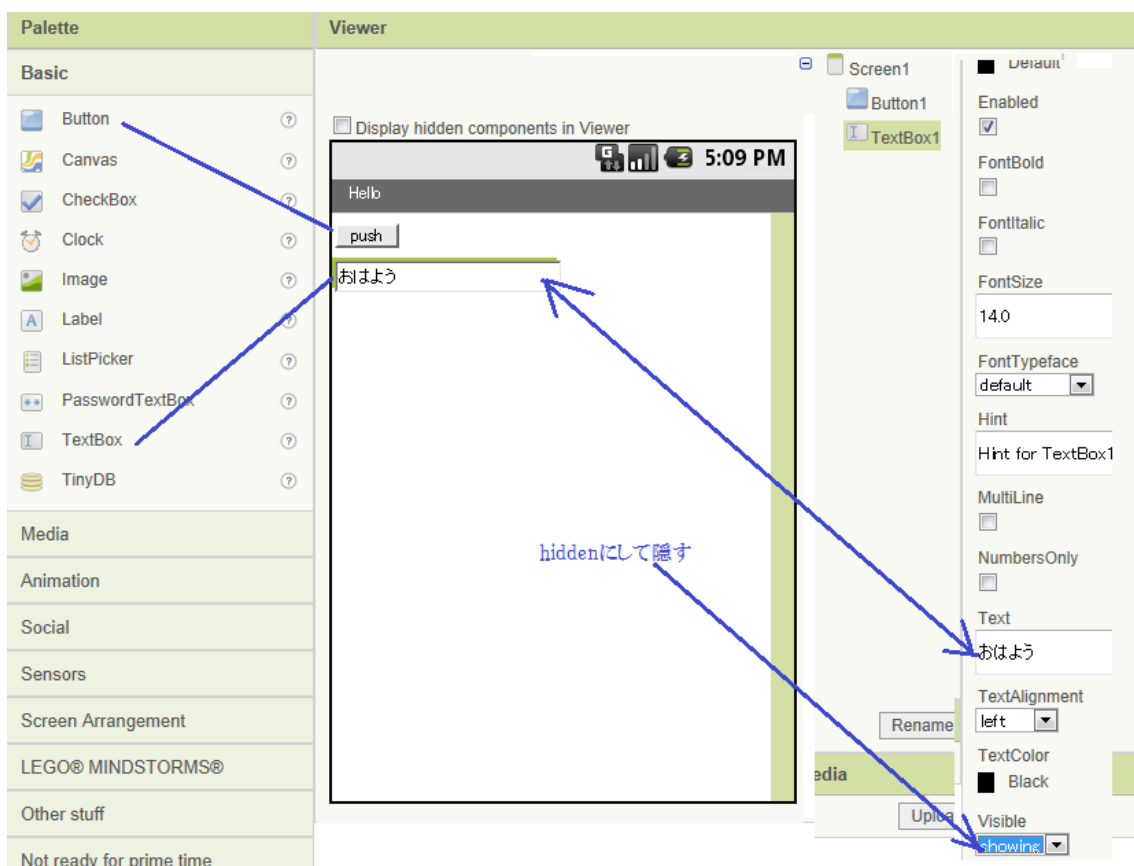
Built-In から text を持ってきてテキストを hello に変える。



(B2)ボタンを押すと、TextBoxに「おはよう」と表示される

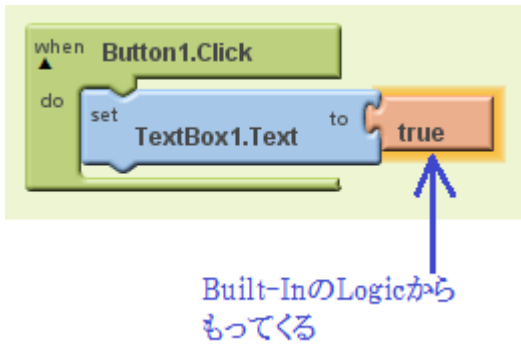
画面設計

TextBoxをもってきて、「おはよう」と記入し、showingになっているVisibleをhiddenにして表示を隠す。

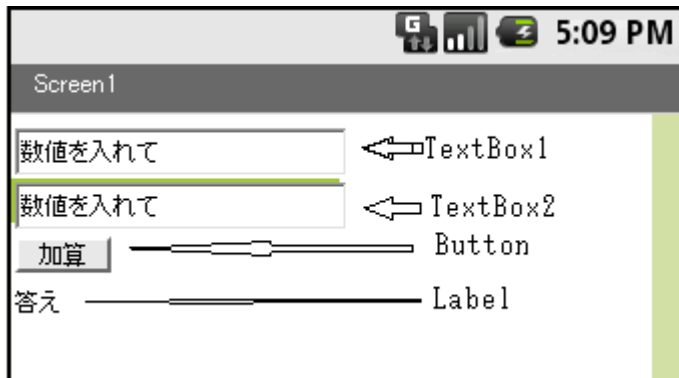


プログラム

ボタンを押すと表示が hidden から showing に変わる。

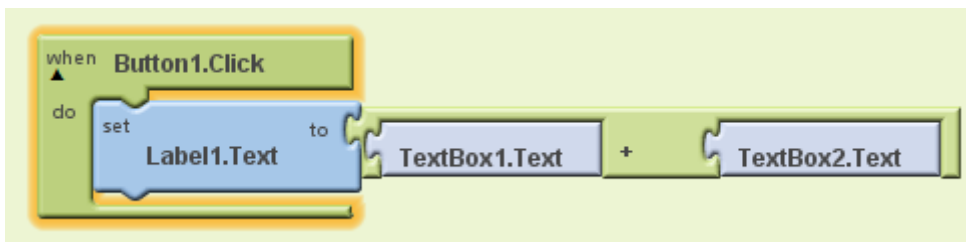


(B3)TextBox に書き込んだ 2つの数字を足す
画面設計



プログラム

加算 (+) のブロックは Math からもってくる

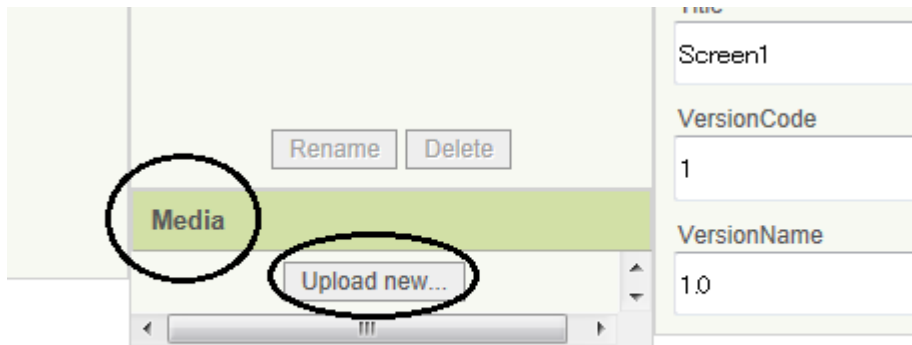


(2)メディア系 Media

(M1)ボタンを押したら画像を表示する方法

準備として、画像をアップロードしておく。

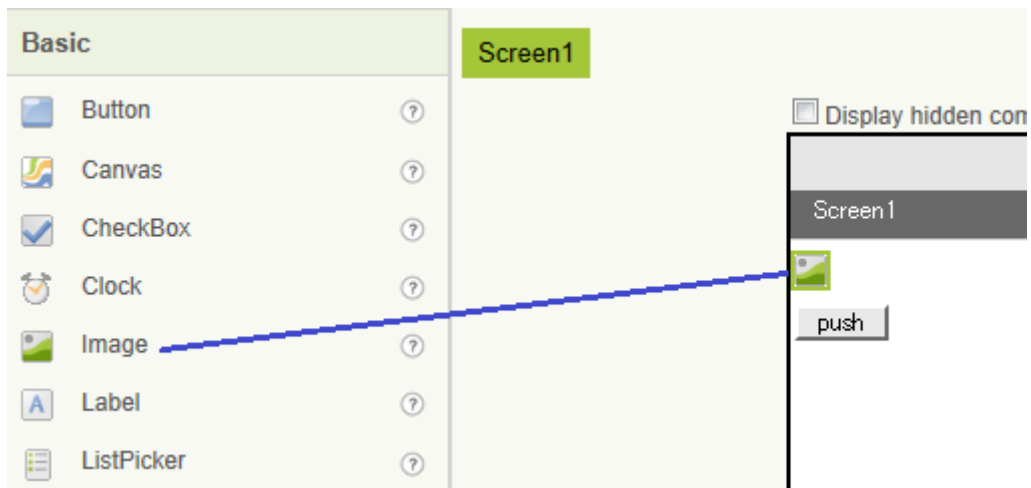
右下の Upload で画像をアップしておく。



画面設計

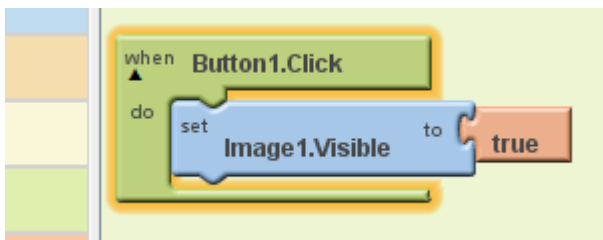


画面設計の際、Image を画面に持ってきておき、Visible を hidden にして画像を隠しておく。



プログラム

隠しておいた画像を表示する。

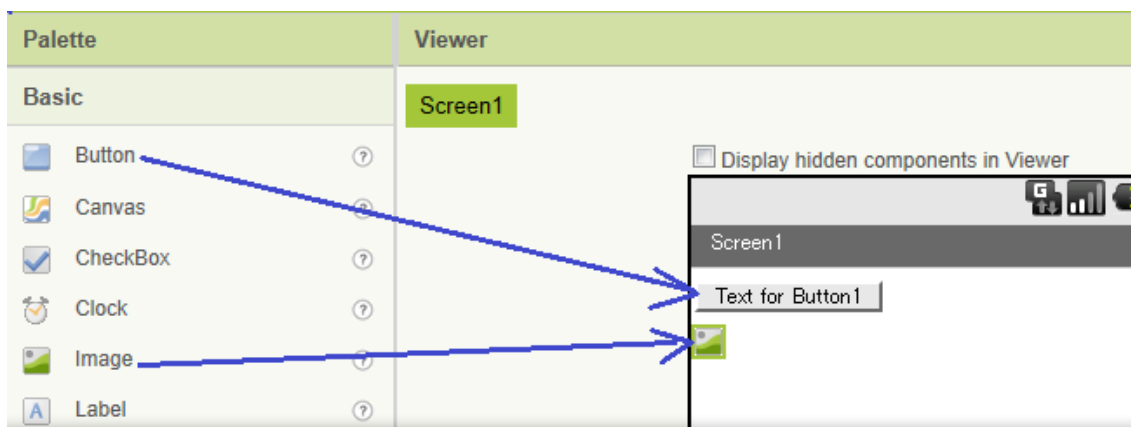


(M2)画像を表示 2

画像のファイル名をプログラム中で作って、画像を表示する。

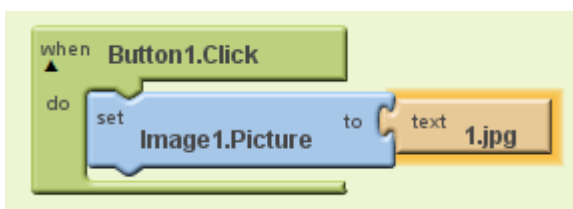
予め「1.jpg」という画像ファイルをアップロードしておく。

画面設計



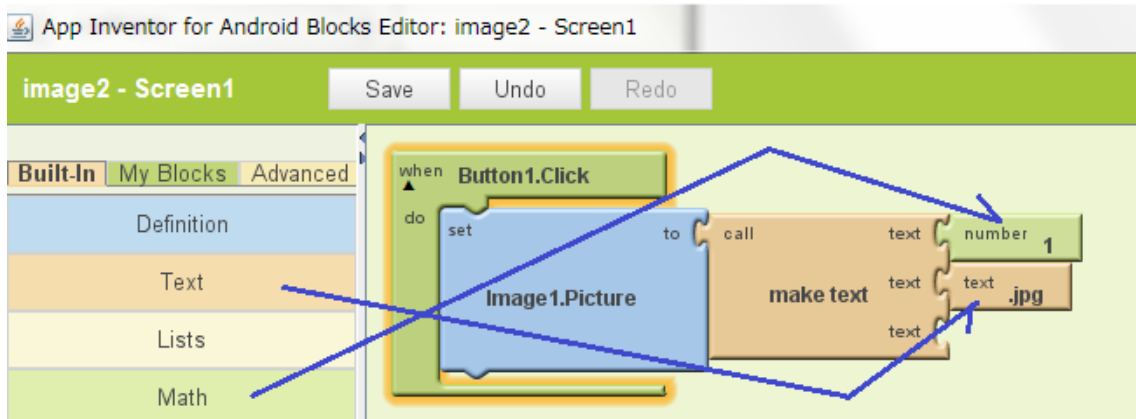
プログラム 1

簡単な方法



プログラム 2

make text でファイル名を作る。この手法は、やがてパラパラ漫画で使用。



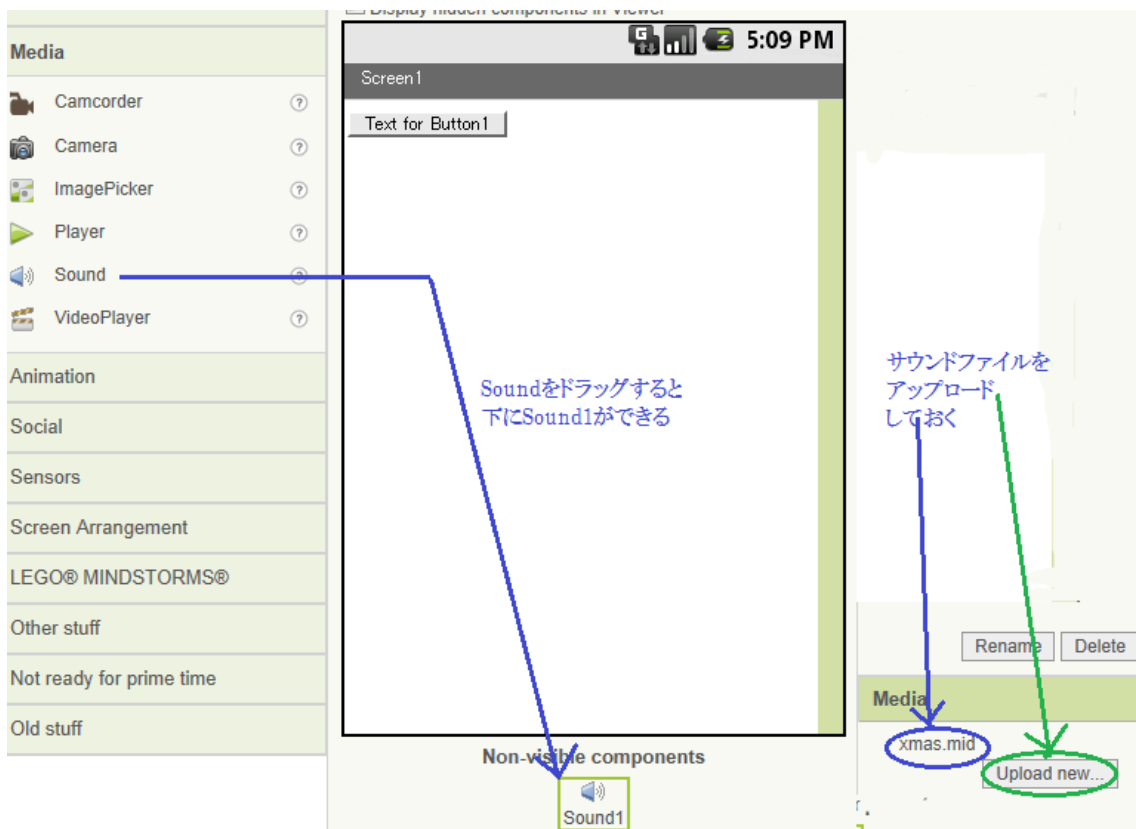
maketext で画像ファイル名を作っている。

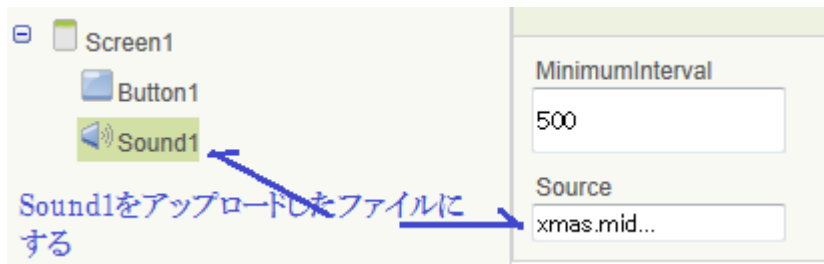
(M3)音を出す

ボタンを押すと音楽が鳴る。

準備として、音楽ファイルをアップロードしておく。

画面設計





プログラム



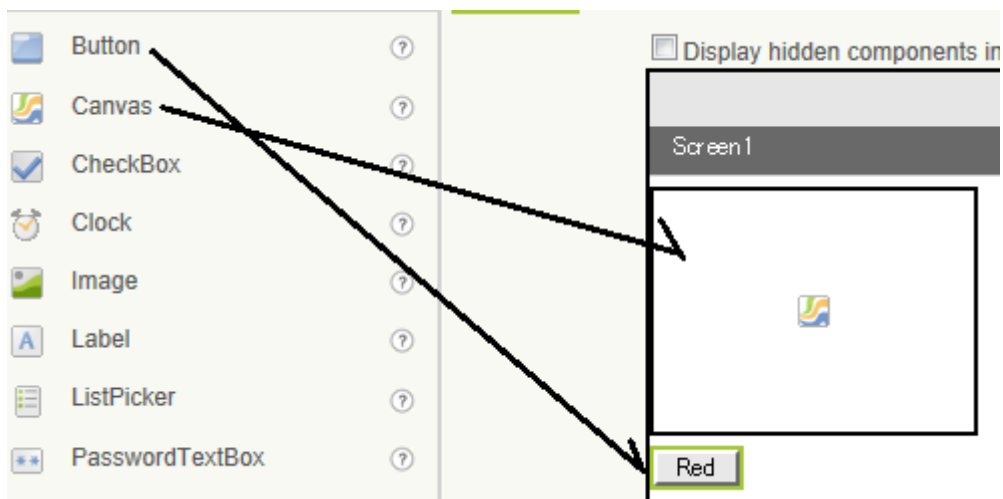
(3)センサー系 Sensor

(S1) 画面上で、絵を描く

画面設計

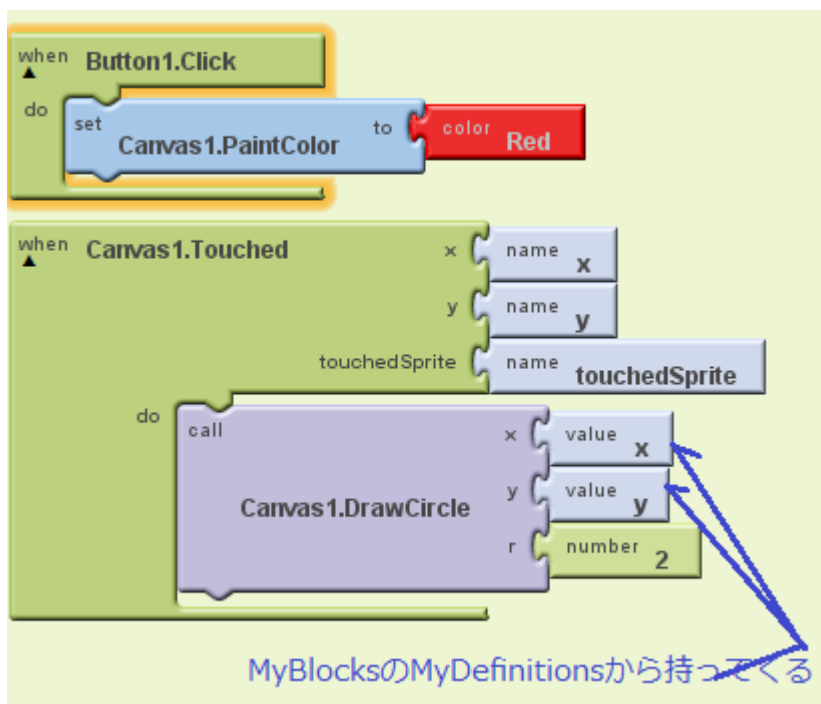
ボタン：ペイントの色を決める

キャンパス：画面



プログラム 1

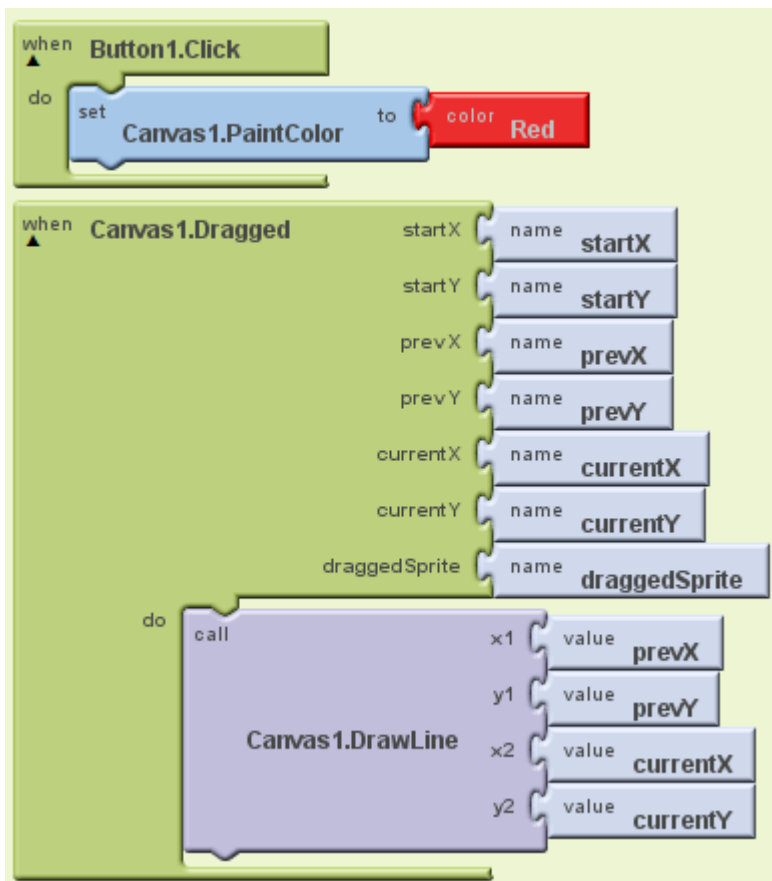
画面に指を触れると、そこに丸が描かれる。



[number 2]ブロックは丸のサイズ。

プログラム2

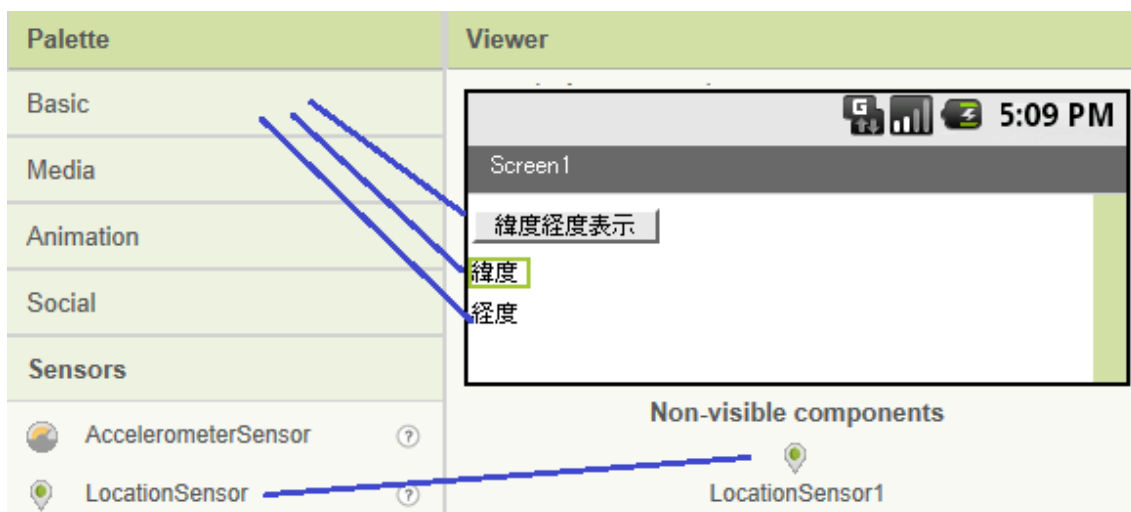
画面上で指を動かすと線が描ける。



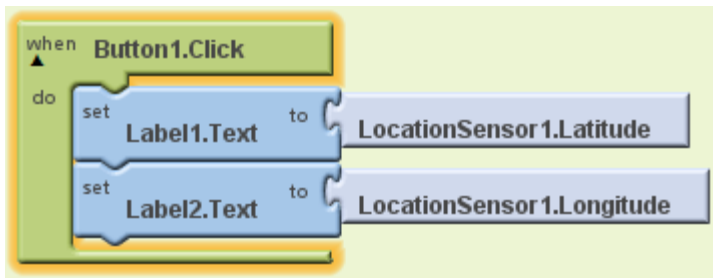
(S2)GPS で緯度経度を表示（野外での動作未確認、室内では 0.0 が表示される）
 ボタンを押すと、Label に緯度経度が表示される。

画面設計

LocationSensor を設計画面に持ってくると、設計画面下にマークがでる。



プログラム



Latitude は緯度、Longitude が経度。

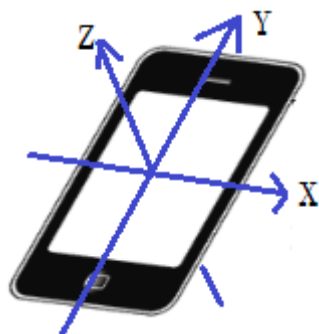
(S3)加速度センサー応用 1

一定時間ごとに、スマホの傾き (x、y、z 方向) を表示
(クロックタイマーを使用)

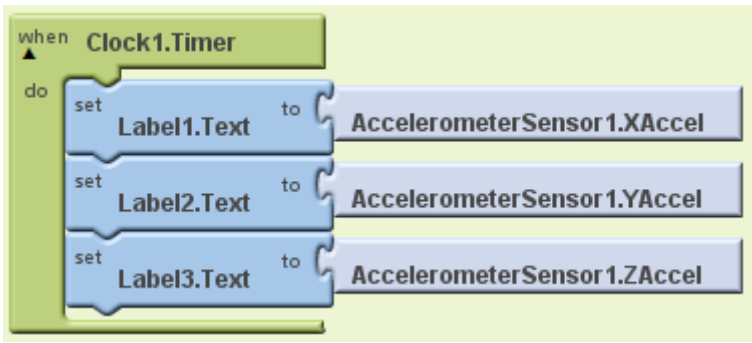
画面設計



傾きの軸は次の通りです。



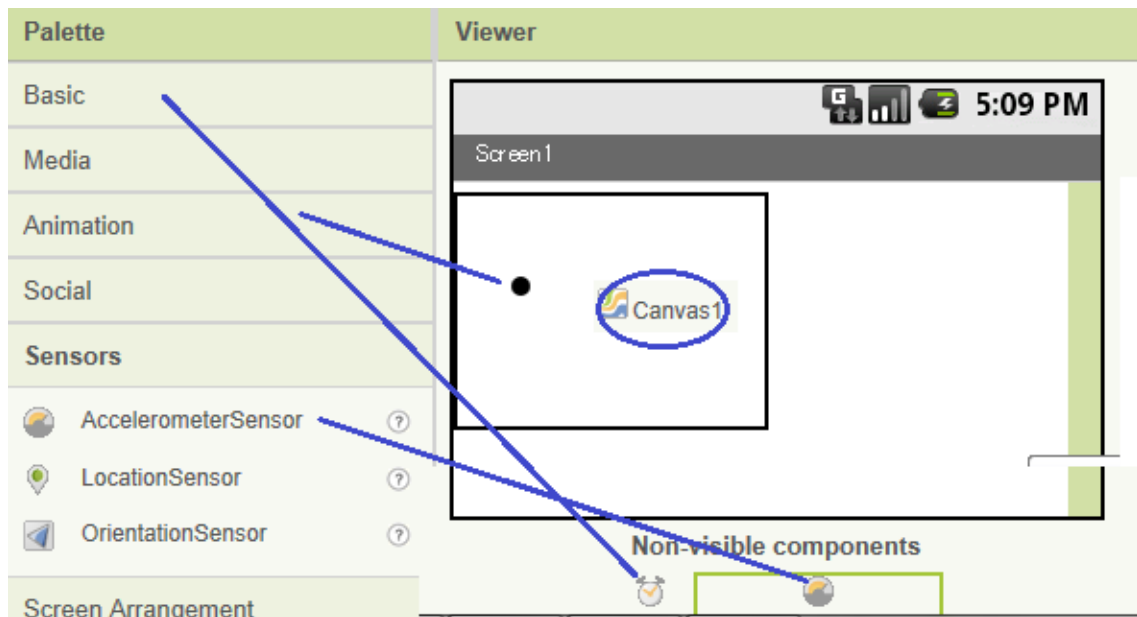
プログラム



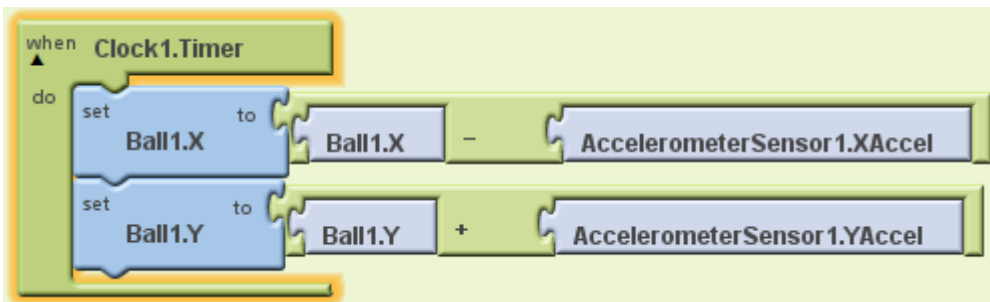
(S4)加速度センサー応用 2

スマホの傾きに反応し、画面上のボールが動く。

Animation から ball をもってくる



プログラム

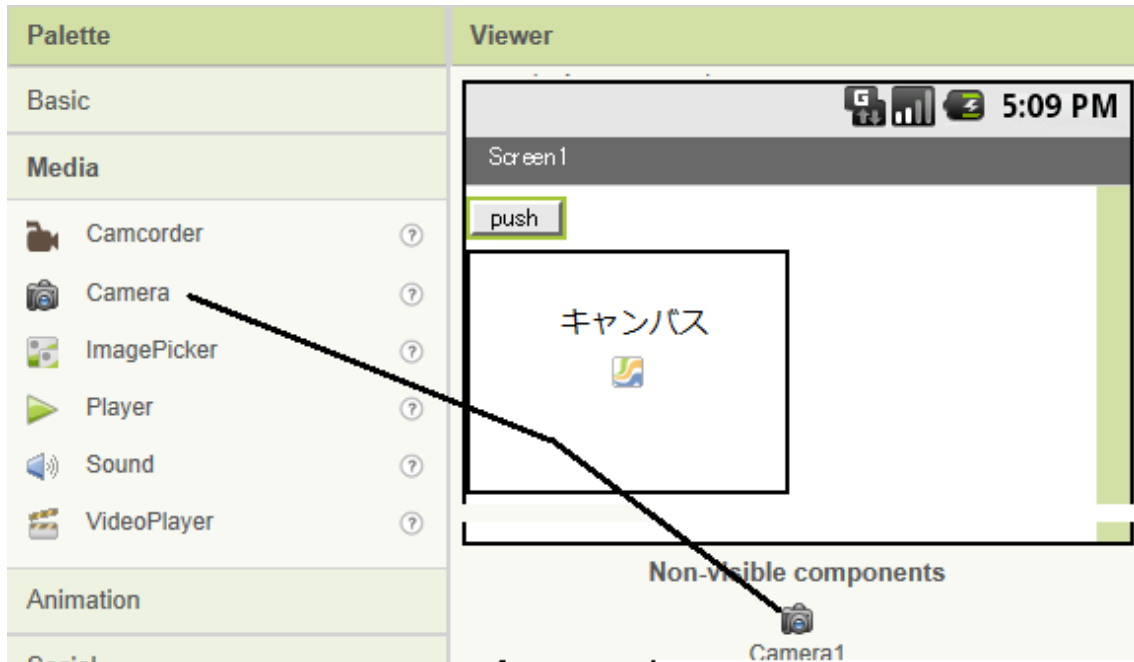


(4) 外注系 Outsourcing

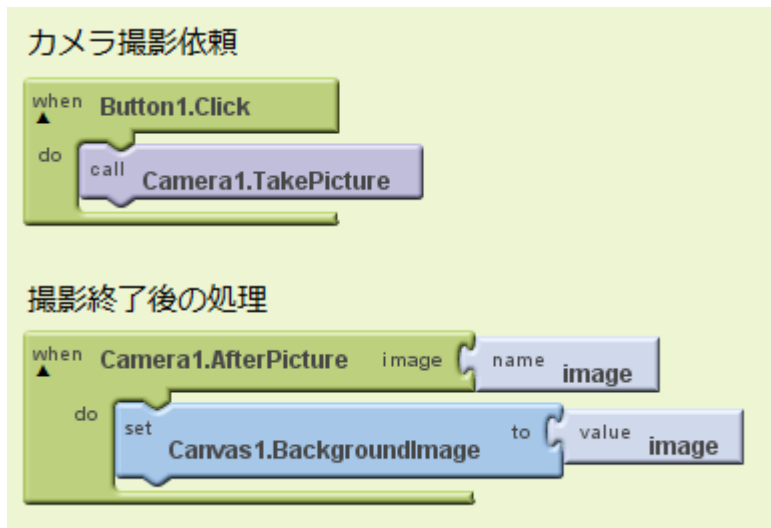
次の2つは、スマホが持っている機能をお借りするので、(1)発注と、(2)納品、に対応する命令が必要です。

(O1)写真撮影

画面設計



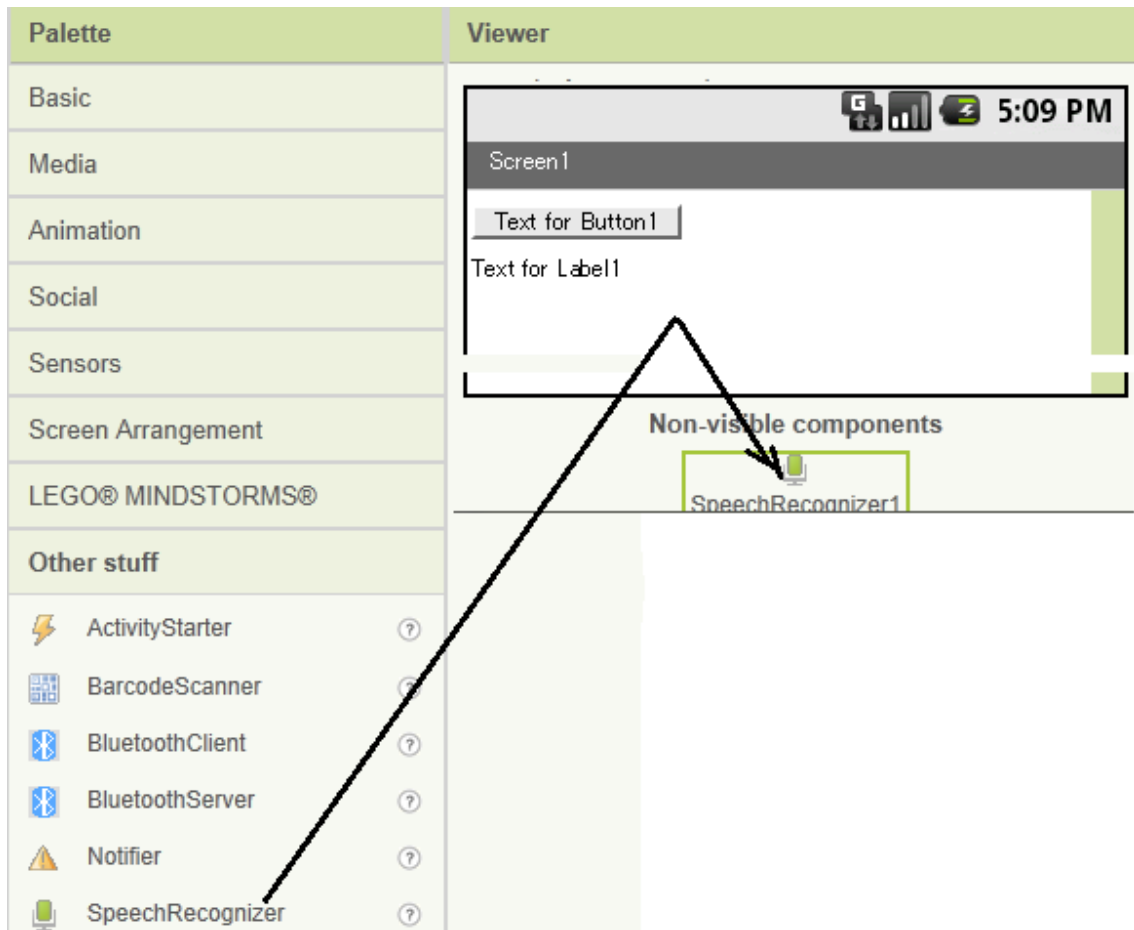
プログラム



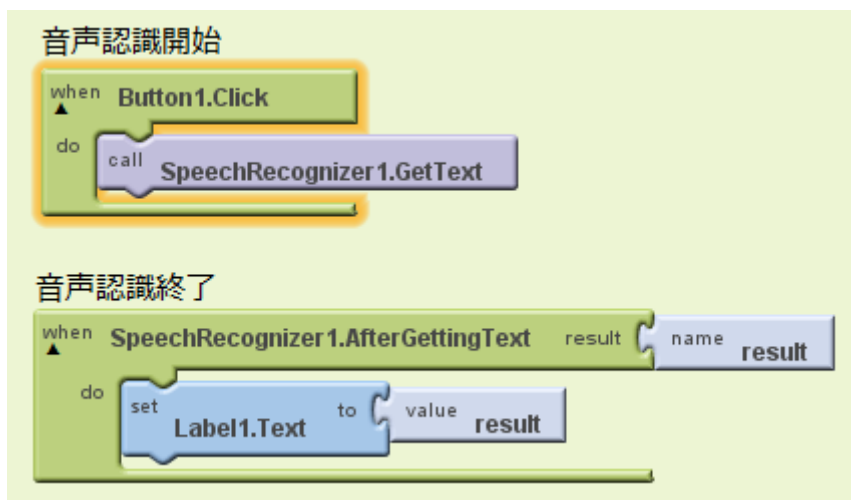
ボタンを押すと、カメラ画像が表示され、シャッターを押すと `image` に貼り付く。

(キャンバスからはみ出て、画面全体に表示されるが、いいのか?)

(O2)音声認識



プログラム



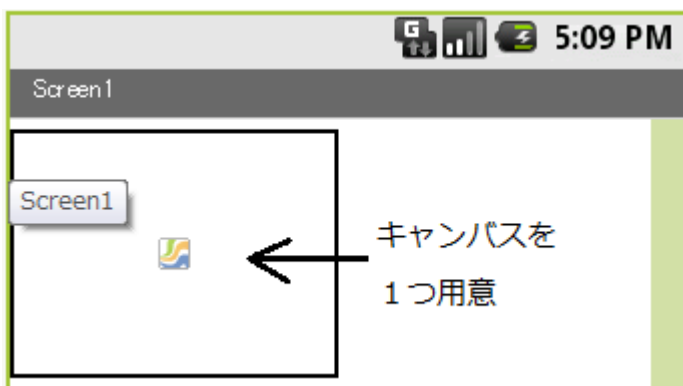
(4) プログラミングの基本技

繰り返して、繰り返し回数が分かっている時は「for range ブロック」を、分かっている時は「while ブロック」を使います。

(P1)繰り返し 1

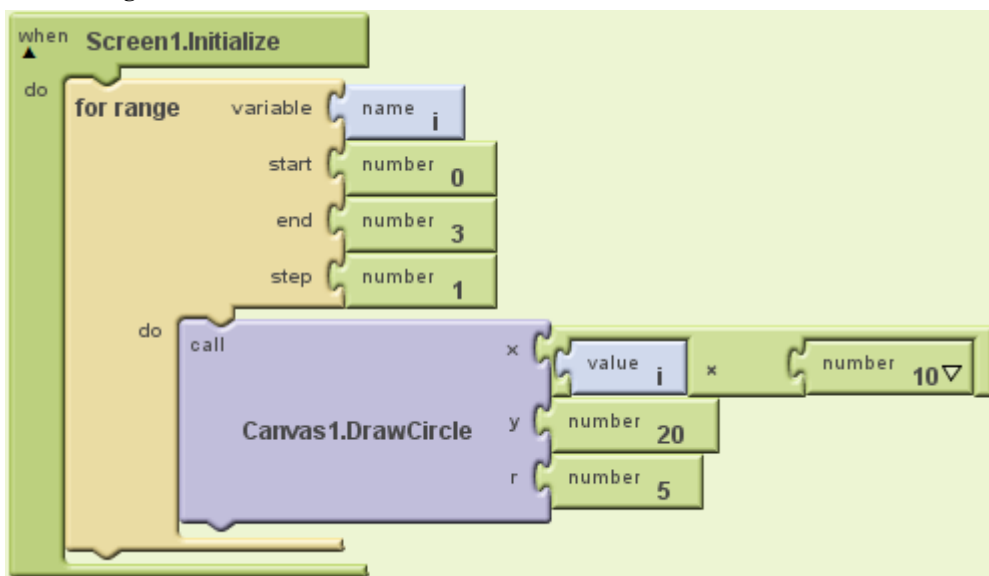
画面 (キャンバス) に 4 つ丸を描く

画面設計



プログラム

「for range」は、Built-In の Control からもってくる

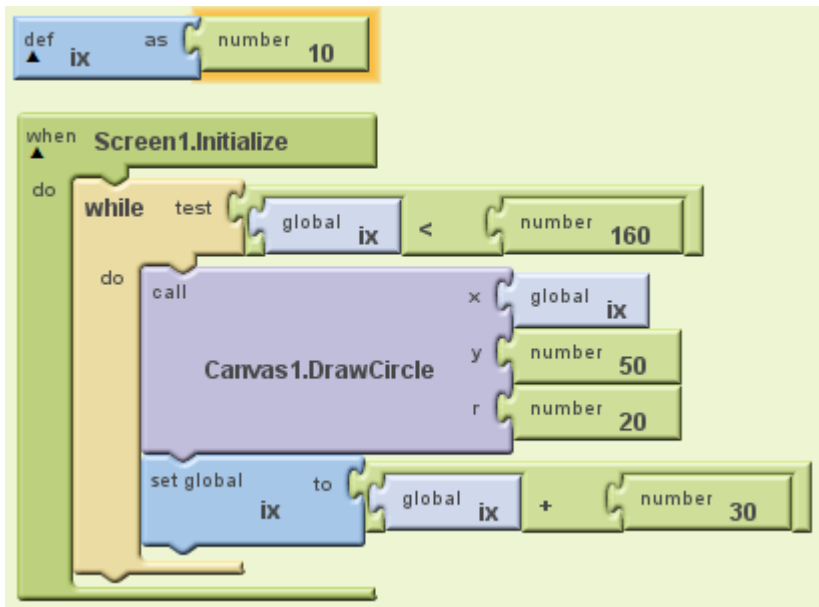


(P2)繰り返し 2 (変数の使い方にも注目)

while ブロックを用いて、x 方向が 160 ピクセルに達するまで円を描く。

画面設計は for と同じキャンバス 1 つ

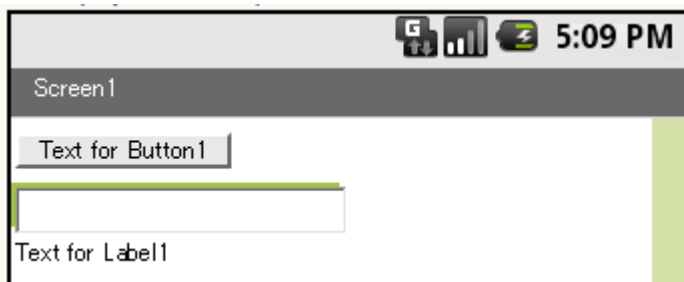
プログラム



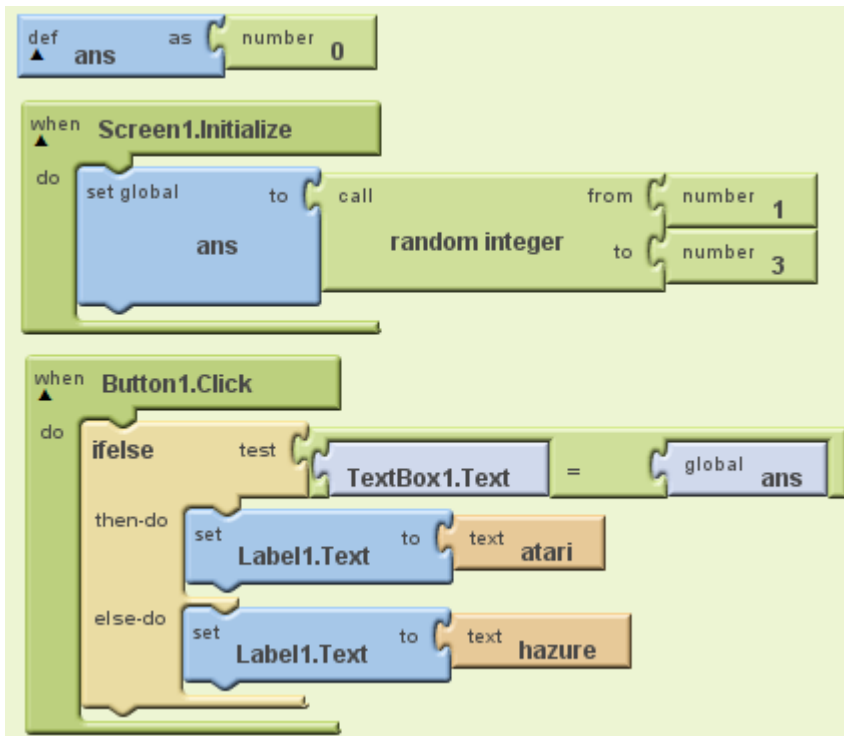
(P3)条件分岐 1

数当て：乱数で1～3の整数を作り、それを当てる。

画面設計

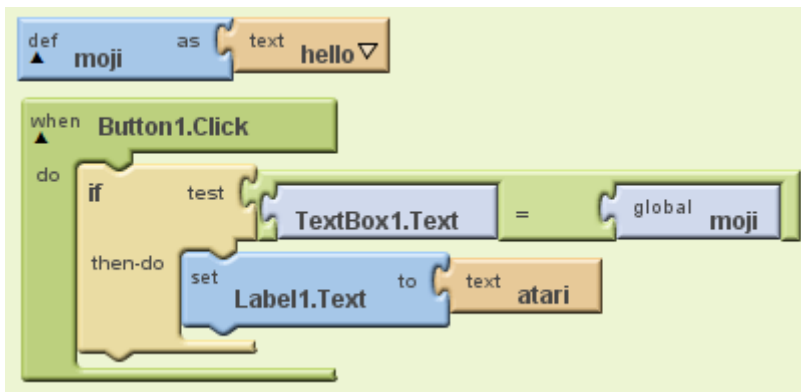


プログラム



(P4)条件分岐 2

文字列当て（変数に文字列を代入できるかのテスト）
プログラム

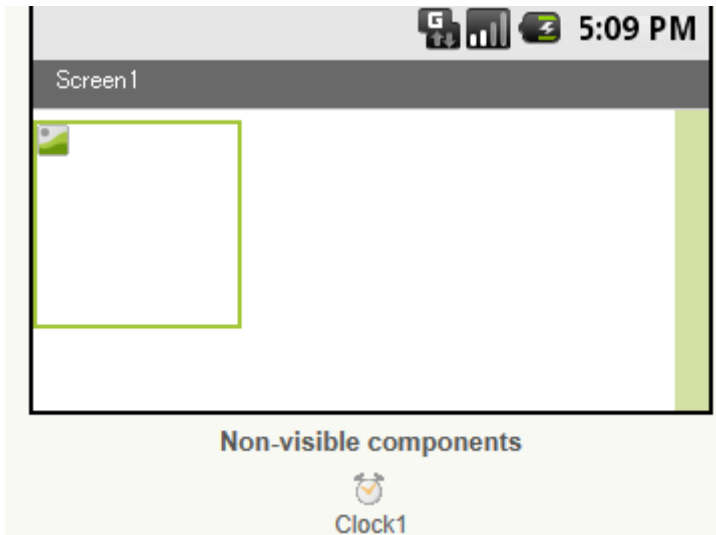


(P5)パラパラ漫画

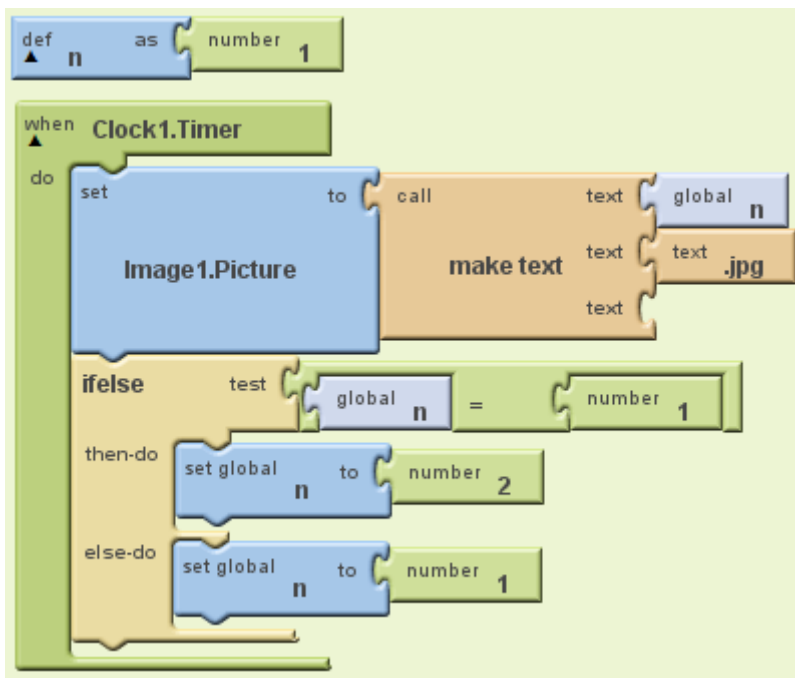
2つの画像（1.jpg と 2.jpg）を一定時間ごとに表示。

画面設計

設計画面にタイマーと image を用意。



プログラム



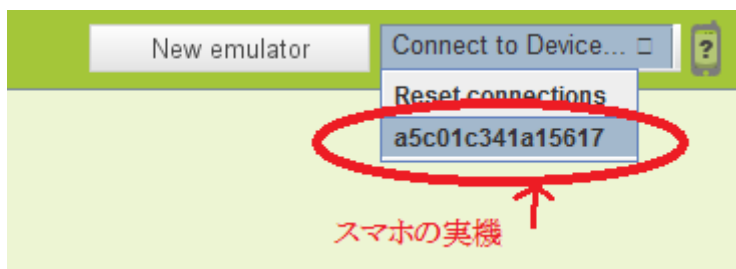
(付録) 実機でのテスト

スマホ (android 端末) の USB ドライバーをメーカーから入手。機種によっては、ADB_USB ドライバーが必要 (sharp)。MAC ならドライバー不要かも。

実機では、「設定」→「アプリケーション設定」→「アプリケーションの管理」で、「提供元不明のアプリ」にチェック「」を入れる。

さらに、「アプリケーション設定」→「開発」に行き、「USB デバッグ」にを入れ、「スリープモードにしない」にを入れる。

そうしてスマホと PC を USB 接続すると「connect to Device」に実機が表示されるので、そこをクリック。



(この時は、emulator を起動しない)