Re:ゼロから始めるアプリ開発

— Xamarin ってなぁに？ —

ここあ　著

2019 年 9 月 22 日　 ver 1.0

■免責

本書は情報の提供のみを目的としています。
本書の内容を実行・適用・運用したことで何が起きようとも、それは実行・適用・運用した人自身の責任であり、著者や関係者はいかなる責任も負いません。

■商標

本書に登場するシステム名や製品名は、関係各社の商標または登録商標です。
また本書では、™、®、© などのマークは省略しています。

目次

第1章 Xamarinってなぁに

はい，いきなりサブタイ回収しました．

Xamarin っていうのは，Android なら `Java` もしくは `Kotlin` のコードを，iOS なら `Swift` のコードをラッピングして `C#`を用いて開発することのできるフレームワークです．

こいつを使うことで，各 OS 間でのロジックに関するコードを共通化することができるってわけなんですね．

さらにさらに，(ちょいと制限があるけど)UI すらも共通化することも可能です．

とりあえず実際にアプリをいくつか作っていきながら Xamarin での開発に慣れていきましょうね．

そうそう，今回は Android アプリに関してのみ触れたいと思います．

そうそう，本著では Xamarin の導入方法とかプロジェクトの作成方法は解説しないことにします．

めんどくさいんだもん．

第2章 キッチンタイマー

まず手始めに定番とも言えるキッチンタイマーを作ってみましょう.
キッチンタイマーにはどのような機能が必要でしょうか.

- タイマーをスタートする
- 終了時に知らせる

本当に最低限の要求仕様はこのくらいではないでしょうか.
てことで開発に移っていきましょう！

2.1 レイアウトを実装する

まずは，アプリのレイアウトを決めましょう．プロジェクトを作成した時に activity_main.xml というファイルが作成されると思うので，そちらをいじっていきましょう.
ここはシンプルに TextView と Button でいこうではありませんか.

▼activity_main.xml

```
 1: <?xml version="1.0" encoding="utf-8"?>
 2: <RelativeLayout
 3:  xmlns:android="http://schemas.android.com/apk/res/android"
 4:  xmlns:app="http://schemas.android.com/apk/res-auto"
 5:  xmlns:tools="http://schemas.android.com/tools"
 6:  android:layout_width="match_parent"
 7:  android:layout_height="match_parent">
 8:  <TextView
 9:   android:layout_width="wrap_content"
10:   android:layout_height="wrap_content"
11:   android:id="@+id/remains"
12:   android:textAppearance="?android:attr/textAppearanceLarge"
13:   android:textSize="80sp"
14:   android:text="03:00"
15:   android:layout_centerInParent="true"
16:   />
17:  <Button
18:   android:layout_width="wrap_content"
19:   android:layout_height="wrap_content"
20:   android:id="@+id/start"
21:   android:text="start"
22:   android:layout_below="@id/remains"
23:   android:layout_centerHorizontal="true"/>
24: </RelativeLayout>
```

2.2　プログラムを書いていこう

最低限キッチンタイマー的な動作をするプログラムを書いてみよう．
先ほどの要件定義を満たすために，カウントダウン機構とスタート動作，ビープ音の生成・発音機能を実装していく．

▼MainActivity.cs

```
 1: using System;
 2: using Android.App;
 3: using Android.OS;
 4: using Android.Runtime;
 5: using Android.Support.Design.Widget;
 6: using Android.Support.V7.App;
 7: using Android.Views;
 8: using Android.Widget;
 9: using Android.Media;
10:
11: namespace KitchenTimer {
12:  [Activity(Label = "@string/app_name", Theme = "@style/AppTheme.NoActionBar",›
   › MainLauncher = true)]
13:  public class MainActivity : AppCompatActivity {
14:
15:   private int sec_ = 180;
16:
17:   Handler handler_;
18:   TextView tv_;
19:   AudioTrack audio_;
20:
21:   // Beep音用の変数群
22:   const double amplification_ = 0.4;
23:   const int sampleRate_ = 44100; // [samples / sec]
24:   const short bitRate_ = 16; // [bits / sec]
25:   const short freq_ = 440; // [Hz] = [1 / sec]
26:   const double duration_ = 0.5; // [sec]
27:   short[] audioBuf_;
28:
29:   protected override void OnCreate(Bundle savedInstanceState) {
30:    base.OnCreate(savedInstanceState);
31:    Xamarin.Essentials.Platform.Init(this, savedInstanceState);
32:    SetContentView(Resource.Layout.activity_main);
33:
34:    handler_ = new Handler();
35:
36:    // 時間を表示させるviewの取得
37:    tv_ = FindViewById<TextView>(Resource.Id.remains);
38:
39:    // Buttonのクリック動作を設定
40:    // どうせ保持していても使わないので直接構築
41:    FindViewById<Button>(Resource.Id.start)
42:     .Click += (sender, e) => {
43:      handler_.PostDelayed(() => Action(), 1000);
```

```
44:     };
45:   }
46:
47:   protected override void OnResume() {
48:    base.OnResume();
49:
50:    // [samples / sec] * [sec] = [samples]
51:    int samples = (int)(sampleRate_ * duration_);
52:    audioBuf_ = new short[samples];
53:
54:    // Beep音の生成
55:    for(int point = 0;point < samples;point++) {
56:     // pointの最大値はsamplesと同値．すなわち発音時間でのsample数
57:     // すなわち，point / sampleRate_は時間位置(time / freq)と等価的存在
58:     audioBuf_[point] = (short)((amplification_ * short.MaxValue) *
59:      System.Math.Sin(2.0 * System.Math.PI * freq_ * point / sampleRate_));
60:    }
61:
62:    audio_ = new AudioTrack(Stream.Music,
63:                sampleRate_,
64:                ChannelOut.Mono,
65:                Encoding.Pcm16bit,
66:                audioBuf_.Length * bitRate_ / 8,
67:                AudioTrackMode.Static);
68:    audio_.Write(audioBuf_, 0, audioBuf_.Length);
69:   }
70:
71:   void Action() {
72:    handler_.RemoveCallbacks(Action);
73:    sec_--;
74:
75:    if(sec_ > 0)
76:     handler_.PostDelayed(Action, 1000);
77:    else
78:     Beep();
79:
80:    // この関数が別スレッドで動いているので，
81:    // UIスレッドを明示的に指定
82:    RunOnUiThread(() => {
83:     tv_.Text = (sec_ / 60).ToString("D2") +
84:     ":" +
85:     (sec_ % 60).ToString("D2");
86:    });
87:
88:   }
89:
90:   void Beep() {
91:    audio_.Stop();
92:    audio_.ReloadStaticData();
93:    audio_.Play();
94:   }
95:  }
96: }
```

```
97:
98:
```

Re:ゼロから始めるアプリ開発

Xamarin ってなぁに？

2019 年 9 月 22 日　ver 1.0

著　者　ここあ
印刷所　○○印刷所

(powered by Re:VIEW Starter)