

2019年3月25日

株式会社インプレスR&D

<https://nextpublishing.jp/>

田中賢一郎氏の待望の新刊！

『ゲームを作りながら楽しく学べる C#テスト駆動開発』発行

テスト駆動開発の基礎をマスターしよう

インプレスグループで電子出版事業を手がける株式会社インプレスR&Dは、『ゲームを作りながら楽しく学べる C#テスト駆動開発』（著者：田中 賢一郎）を発行いたします。

『ゲームを作りながら楽しく学べるC#テスト駆動開発』

<https://nextpublishing.jp/isbn/9784844396949>



著者：田中 賢一郎

小売希望価格：電子書籍版 2600 円(税別)／印刷書籍版 3200 円(税別)

電子書籍版フォーマット：EPUB3／Kindle Format8

印刷書籍版仕様：B5 判／モノクロ／本文 264 ページ

ISBN：978-4-8443-9694-9

発行：インプレス R&D

<< 発行主旨・内容紹介 >>

本書はゲーム作りを通してテスト駆動開発の基本を学ぶための解説書です。

著者が、最初にテスト駆動開発に触れたのは2003年頃でした。著者も最初は「機能を実装するのにテストをつくるなんて無駄な作業だなあ」と懐疑的でしたが、実際に試してみると、その効果が絶大であることを痛感しました。ソースコードの修正もテストがあるので安心して行えます。

機能を追加することで副作用が起きても、テストがあるのですぐに発見できます。あたかも、「セーフティネットの上でいろいろな技を安心して試すことができる」そんな感覚でした。最初のプロジェクトを終えたころにはすっかりテスト駆動開発の信者になってしまいました。

(本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。)

第1章 XAML 入門

1-1 XAMLとコントロール

XAMLは、ユーザーインタフェースを記述するためのマークアップ言語です。ただ、これだけの説明ではピンとこないかもしれません。具体例をみてみましょう。ボタンを2つ並べるサンプルを以下に示します。

```
<Grid>
  <StackPanel>
    <Button Content="Yes"/>
    <Button Content="No"/>
  </StackPanel>
</Grid>
```

HTMLなどのマークアップ言語とおなじように<と>で文節の意味を与えてゆきながら、ユーザーインタフェースを作成していきます。両側の構成要素（ボタン、テキスト、入力フィールド、パネル等）はコントロールとして実装します。それらコントロールを生成するためには以下のように記述します。

```
<コントロール> 子要素 </コントロール>
```

<コントロール>が開始で、</コントロール>が終了の印です。そのコントロールに含まれる要素を子要素として間に挟みます。タグを使って文節の意味を与えるという考え方はHTMLと似ています。例えば、テキストを表示する場合は以下ようになります。

```
<TextBlock>Hello World</TextBlock>
```

ボタンを表示する場合は以下の通りです。

```
<Button>Click Me</Button>
```

付属の情報を指定する場合は、コントロールのプロパティを使用します。

```
<コントロール プロパティ名="プロパティ値" > ... </コントロール>
```

例えばフォントサイズはテキストに関する情報です。TextBlockのフォントサイズを変更する場合は、以下のようにプロパティを使って指定します。

```
<TextBlock FontSize="50" >Click Me</TextBlock>
```

どのようなプロパティを指定できるかはコントロールに依存します。例えば、画像を表示するImageコントロールには、ファイルを指定するSourceプロパティがあります。TextBlockコントロールにはSourceプロパティはありません。逆に、TextBlockコントロールにはFontSizeプロパティがありますが、Imageコントロールにはありません。コントロールの役目を考えれば利用できるプロパティが異なることは納得できると思います。

UWPで使用できるコントロールの一覧は以下のURLから参照できます。
<https://docs.microsoft.com/en-us/uwp/api/windows.ui.xaml.controls>
<https://docs.microsoft.com/en-us/windows/uwp/design/controls-and-patterns/index>

コントロールは入れ子にすることができます。例えばフォントサイズ50のテキストを子要素に持つ、背景が青、前景が白のボタンを作成するには以下のように記述します。

```
<Button Background="Blue" Foreground="White">
  <TextBlock FontSize="50" >Click Me</TextBlock>
</Button>
```



また、全てのコントロールはx:Nameというプロパティを指定することで一意な名前を付けることができます。

```
<Button x:Name="MyButton" Background="Blue" Foreground="White">
  <TextBlock x:Name="MyButtonText" FontSize="50" >Click Me</TextBlock>
</Button>
```

このx:NameプロパティはC#のプログラムと連携するときに重宝します。

単に説明を説きよりも、手を動かした方が理解できると思うので、さっそくプロジェクトを作成してみましょう。[ファイル] → [新規作成] → [プロジェクト] から新しいプロジェクトダイアログを表示し、[Visual C#] → [Windowsユニバーサル] → [空白のアプリ (ユニバーサルWindows)]

第2章 棒人間ジャンプゲーム

2-1 プロジェクトの作成

Visual Studioを起動し、[ファイル] → [新規作成] → [プロジェクト] で「新しいプロジェクト」ダイアログを起動します。Visual C#の「空白のアプリ (ユニバーサルWindows)」を選択し、プロジェクト名をTDD_JumperとしてOKボタンを押します。

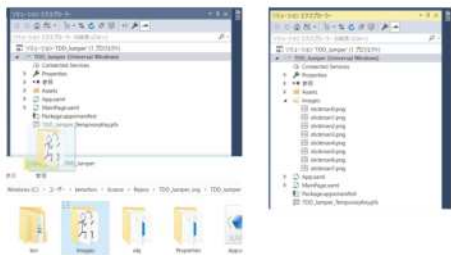
※サンプルはTDD_Jumper1、TDD_Jumper2、……とプロジェクト名を変更しています。



ターゲットバージョンと最小バージョンを選択するダイアログが表示された場合、OKボタンを押して進みます。

棒人間の画像を含むフォルダ (フォルダ名: Images) をエクスプローラーからVisual Studioのソリューションエクスプローラーにドラッグ&ドロップして、プロジェクトにファイルを追加します。この際、ソリューションエクスプローラーでフォーカスがある場所にフォルダが追加されるので、

TDD_Jumper (Universal Windows) がフォーカスされている状態でドロップ操作を行ってください。



第3章 テスト駆動開発入門

3-1 ビューとモデルの分離を意識しないアプリの作成

TDD_ThermoConverter という名前のプロジェクトを [Visual C#] → [Windows ユニバーサル] → [空白のアプリ (ユニバーサル Windows)] で作成します。



MainPage にスライダーと摂氏と華氏を表示する TextBox を配置します。

● TDD_ThermoConverter / MainPage.xaml

```
<Page
  xmlns:local="using:TDD_ThermoConverter"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d"
  Background="{ThemeResource ApplicationPageBackgroundThemeBrush}"
  <Grid>
    <StackPanel Width="500">
      <Slider Minimum="0" Maximum="100"
        ValueChanged="Slider_ValueChanged"/>
      <StackPanel Orientation="Horizontal">
        <TextBox Text="摂氏:"/></StackPanel>
    </StackPanel>
  </Grid>
```

```
<TextBox x:Name="Celsius"/></TextBox>
</StackPanel>
<StackPanel Orientation="Horizontal">
  <TextBox Text="華氏:"/></TextBox>
  <TextBox x:Name="Fahrenheit"/></TextBox>
</StackPanel>
</StackPanel>
</Grid>
</Page>
```

● TDD_ThermoConverter / MainPage.xaml.cs

```
using Windows.UI.Xaml.Controls;
using Windows.UI.Xaml.Controls.Primitives;

namespace TDD_ThermoConverter
{
  public sealed partial class MainPage : Page
  {
    public MainPage()
    {
      this.InitializeComponent();
    }

    private void Slider_ValueChanged(object sender,
      RangeBaseValueChangedEventArgs e)
    {
      Celsius.Text = e.NewValue.ToString();
      Fahrenheit.Text = (e.NewValue * 1.8 + 32).ToString();
    }
  }
}
```

これでアプリは完成しました。スライダーを動かすと摂氏と華氏の値が変化します。

第4章 テスト駆動開発実践

4-2 基本テスト

「テストを作成してから実装を進める」というテスト駆動開発を実践していきます。最初は回りくどい、面倒と感じるかもしれませんが、このリズムを感じ取っていただければと思います。

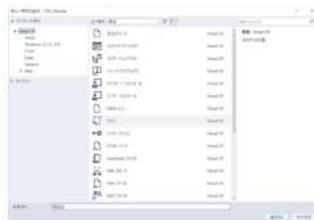
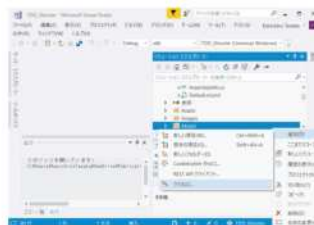
4-2-1 Ship の表示

自機が持つべきプロパティを列挙してみます。描画して表示するためには少なくともこの程度のプロパティは必要になるでしょう。この時点では完成系をイメージできていなくても構いません。必要に応じて都度変更できるのがテスト駆動開発の良いところです。

- ・イメージ：画像
- ・X：X座標
- ・Y：Y座標
- ・Width：画像の幅
- ・Height：画像の高さ

まずは ship.cs をテストするコードを記述します。TDD_Shooter.Tests にある UnitTest.cs を ShipTest.cs にファイル名を変更します。

自機 ship の場所を設定するテストから実装してみましょう。TDD_Shooter プロジェクトの Model フォルダに Ship.cs を追加します。ソリューションエクスプローラーで Model フォルダを右クリックし、[追加] → [クラス] を選択し、新しい項目の追加ダイアログで、クラスを選択し、名前に Ship.cs と記述し、追加ボタンを押下します。



<<目次>>

第1章 XAML 入門

第2章 棒人間ジャンプゲーム

第3章 テスト駆動開発入門

第4章 テスト駆動開発実践

<<著者紹介>>

田中 賢一郎

慶應義塾大学理工学部修了。キャノン株式会社に入社し、デジタル放送局の起ち上げに従事。その間に単独でデータ放送ブラウザを実装し、マイクロソフト(U.S.)へソースライセンスし、Media Center TV チームの開発者としてマイクロソフトへ。MS では Windows、Xbox、Office 365 などの開発・マネージ・サポートに携わる。2016 年に中小企業診断士登録後、セカンドキャリアは IT 教育に携わると決め、IT 系の専門学校で 1 年間現場経験を積んだ後、2017 年春に Future Coders (<http://future-coders.net>)を設立。「プログラミング教育を通して一人ひとりの可能性をひろげる」という理念のもと、英語と数学に重点をおいた実践的なプログラミングスクールの運営を目指す。趣味はジャズピアノ演奏。

<<販売ストア>>

電子書籍:

Amazon Kindle ストア、楽天 kobo イブックスストア、Apple Books、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

印刷書籍:

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※ 各ストアでの販売は準備が整いしだい開始されます。

※ 全国の一般書店からもご注文いただけます。

【株式会社インプレス R&D】 <https://nextpublishing.jp/>

株式会社インプレス R&D (本社：東京都千代田区、代表取締役社長：井芹昌信) は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らが、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム(またはメソッド)の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド(POD)による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少数数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

【インプレスグループ】 <https://www.impressholdings.com/>

株式会社インプレスホールディングス(本社：東京都千代田区、代表取締役：唐島夏生、証券コード：東証1部9479)を持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「旅・鉄道」「学術・理工学」を主要テーマに専門性の高いメディア&サービスおよびソリューション事業を展開しています。さらに、コンテンツビジネスのプラットフォーム開発・運営も手がけています。

【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター

TEL 03-6837-4820

電子メール: np-info@impress.co.jp