

ESライブラリ&& ゲームプログラミング

2D編 - 第7回 基本図形でレトロゲームグラフィクス

基本的な図形の描画

- Windowsには、線、四角形、円といった基本的な図形を描画する関数がある
- Direct3Dからも、上記の関数を利用した描画ができる

概要

Windowsには、ビットマップの描画のほかに、線、四角形、円といった基本的な図形を描画する関数が用意されています。これらの関数は、Direct3Dへの描画もできるようになっており、ドロー系ソフトのようなグラフィックを表現することもできます。しかしながら、近年のグラフィックカードは、2D描画のアクセラレータが搭載されていないため、ハードウェアの描画支援は受けられません。また、2D描画中は、3Dアクセラレーションの機能が停止することもあり、1フレーム中に多数の基本図形を描画すると、全体的な処理速度の低下を招いてしまいます。

ESライブラリでの基本図形描画

ESライブラリでは、基本的な図形を描画するための機能が準備されています。まず、描画の前に、Canvasの取得が必要です。Canvasは、画面またはスプライトから取得できるようになっています。取得したCanvasに対し、基本図形の描画関数が実行できます。線や塗りつぶしの描画色は、Paint型の変数に属性を設定し、描画関数に渡すことで自由に設定できます。

```
Canvas    canvas = GraphicsDevice.LockCanvas(); // キャンバスの取得

Paint     paint;

paint.SetColor(Color(255, 0, 0));           // 描画色を赤に設定
canvas.DrawCircle(100, 100, 25, paint);     // 中心座標(100, 100)に半径25の円を描画

paint.SetColor(Color(0, 0, 255));          // 描画色を青に設定
canvas.DrawRect(Rect(50, 50, 300, 300), paint); // 左上(50, 50)と右下(300, 300)を囲む
// 四角形を描画

GraphicsDevice.UnlockCanvas();             // キャンバスの解放
```

基本図形の描画が終わったら、必ずUnlockCanvas関数を実行します。また、基本図形は、3Dの描画ではないため、BeginScene関数とEndScene関数の外側でなければ描画することができません。

基本図形描画関数

Canvasには、以下の基本図形描画関数が備わっています。

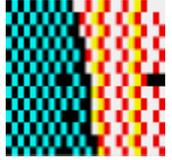
DrawPixel	ピクセルの描画
DrawPoint	点の描画
DrawLine	線の描画
DrawPath	連続した線の描画
DrawBezier	ベジェ曲線の描画
DrawRect	四角形の描画
DrawRoundRect	角の丸い四角形の描画
DrawPolygon	多角形の描画
DrawCircle	円の描画
DrawOval	楕円の描画
DrawArc	弧の描画

色を指定するPaintには、以下の関数を実行して線、塗りつぶし色の設定ができます。

SetColor(色)	境界線、塗りつぶし色の設定
SetLine(色, 幅)	線の色と幅の設定
SetLineColor(色)	線の色の設定
SetLineWidth(幅)	線の幅の設定
SetPaintColor(色)	塗りつぶし色の設定

レトロゲームのグラフィックス

古いゲームでは、メモリが非常に少なかったため、ビットマップのような画像を扱うことが困難でした。そのため、グラフィックスは点を打つ、線を引く、塗りつぶしといった命令を駆使して表示していました。色も8色や16色の原色しか使えません。中間色は、2色または3色を市松状に並べることで、擬似的に作り出すタイリングという技法が編み出され、80年代まで長らく使われていました。



タイリングの例



レトロゲームグラフィックスの例(左: 神戸の街並み 中央: 摩周湖)

課題

基本的な図形を組み合わせて描画してみましょう。

(1)以下のプログラムをDraw関数のGraphicsDevice.EndScene関数の後に追加し、実行結果を確認しましょう。

```
Canvas canvas = GraphicsDevice.LockCanvas(); // キャンバスの取得
Paint paint;

paint.SetColor(Color(0, 182, 255));
canvas.DrawArc(Rect(0, 0, 512, 512), 120.0f, 300.0f, false, paint);

GraphicsDevice.UnlockCanvas(); // キャンバスの解放
```

(2)さらに、以下のプログラムを適切な場所に追加し、実行結果を確認しましょう。

```
paint.SetLineColor(Color_Black);
paint.SetPaintColor(Color_White);
canvas.DrawCircle(256, 304, 192, paint);
```

(3)さらに、以下のプログラムを適切な場所に追加し、実行結果を確認しましょう。

```
canvas.DrawCircle(200, 116, 56, paint);
canvas.DrawCircle(312, 116, 56, paint);
```

(4)さらに、以下のプログラムを適切な場所に追加し、実行結果を確認しましょう。

```
paint.SetPaintColor(Color_Brown);
canvas.DrawRoundRect(Rect(96, 416, 498), 20, 20, paint);
```

(5)想像力を駆使し、図形を組み合わせて描画できるキャラクターを完成させましょう。