「　直線の方程式　」

東京情報クリエイター工学院

直線の方程式

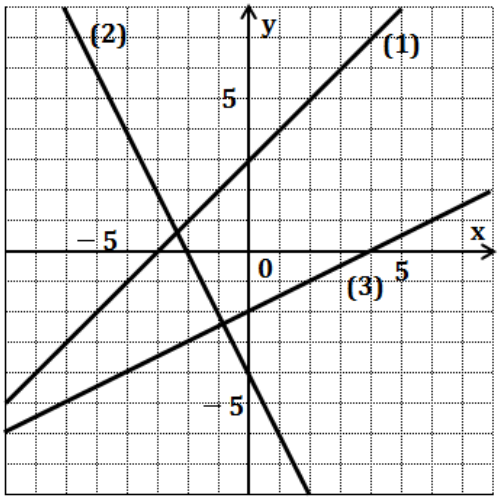
◆直線の方程式

１．傾きと切片から求める

傾きが a　　y 切片が $(0,b)$ の直線を表す式は、「$y=ax+b$ 」

【練習問題 1】

下図のグラフから傾きと切片を読み取り、直線の方程式を求めなさい。



(1) 傾き　　y 切片　　直線を表す方程式

(2) 傾き　　y 切片　　直線を表す方程式

(3) 傾き　　y 切片　　直線を表す方程式

２．傾きと１点から求める

点(x_1,y_1)を通り、傾きaの直線を表す式は、「$y-y_1=a(x-x_1)$」

【例題 1】

点 $A(2,3)$ を通り、傾き 2 の直線の方程式を求めなさい。

上記の公式に代入すると、

$$y-3=2(x-2)\qquad y-3=2x-4\qquad y=2x-1$$

【練習問題 2】

点 A を通り、次の傾きの直線の方程式を求めなさい。

(1) $A(6,8)$ 傾き $-\dfrac{1}{2}$

(2) $A\left(\dfrac{7}{4},\dfrac{9}{4}\right)$ 傾き $-\dfrac{3}{7}$

「　直線の方程式　」

東京情報クリエイター工学院

３．2 点から求める

2 点(x_1, y_1)と(x_2, y_2)を通る直線を表す式は、「$(y - y_2)(x_1 - x_2) = (y_1 - y_2)(x - x_2)$」

【例題 2】

点$(2, 2)$と$(-1, -7)$を通る 1 直線を表す式を求めなさい。

上記の公式に代入すると、

$(y + 7)(2 + 1) = (2 + 7)(x + 1)$　　$3(y + 7) = 9(x + 1)$

$3y + 21 = 9x + 9$　　$3y = 9x - 1$　　$y = 3x - 4$

【練習問題 3】

次の 2 点を通る直線の方程式を求めなさい。

⑴（$-5, 2$）と（$16, -4$）

⑵（$7, -6$）と（$-2, -6$）

４．直線を表す一般式

平面上の直線は、$ax + by + c = 0$ で表される。

⑴平行な直線の式

平行な直線の傾きは　　。逆に、傾きが等しい直線は　　である。

⑵垂直な直線の式

互いに垂直な 2 本の直線の傾きの積は「　　」である。逆に、傾きの積が「　　」である 2 本の直線は互いに「　　」である。

言い換えると、互いに垂直な片方の直線の傾きが「　　」ならもう片方の傾きは、「　　」となる。

直線 $y = ax + b$ と直線 $y = cx + d$ が垂直　⇔　$a \times c = -1$

【例題 3】

直線、$2x + 3y = 12$ に平行で、点$(6, 12)$を通る直線を表す式を求めなさい。

直線、$2x + 3y = 12$　　$3y = -2x + 12$　　$y = -\frac{2}{3}x + 4$

傾きは、$-\frac{2}{3}$　　点$(6, 12)$を通るから、　$y - 12 = -\frac{2}{3}(x - 6)$

従って　$y = -\frac{2}{3}x + 16$

【例題 4】

直線、$2x + 3y = 12$に垂直に交わり、点$(6, 12)$を通る直線を表す式を求めなさい。

直線、$2x + 3y = 12$の傾きは、$-\frac{2}{3}$

これと垂直に交わる直線の傾きを a とすると　　$-\frac{2}{3} \times a = -1$　　$a = \frac{3}{2}$

点$(6, 12)$を通るから、　$y - 12 = \frac{3}{2}(x - 6)$　　従って、　　$y = \frac{3}{2}x + 3$

「　直線の方程式　」

東京情報クリエイター工学院

【練習問題 4】

(1) 直線 $2x-3y+5=0$ に平行で、点 $(2,2)$ を通る直線を表す式を求めなさい。

(2) 直線 $2x+5y+7=0$ に垂直に交わり、点 $(-3,4)$ を通る直線を表す式を求めなさい。

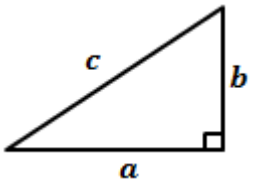
◆ピタゴラスの定理

プログラミングでは、画面上の 2 点間の距離を知らなければならない場合がよくあります。例えば、2 つの物体がぶつかりそうな時や、2 つのキャラクタが接触しそうなときなどがそうです。あるいは、ある距離までキャラクタが近づいてきたら攻撃を開始する場合などもそうでしょう。こんなとき、物体間の距離をすばやく計算できなければなりません。2 点間の距離は、「**ピタゴラスの定理(三平方の定理)**」を使えば簡単に計算することができます。

ピタゴラスの定理(三平方の定理)

「　$a^2+b^2=c^2$　」

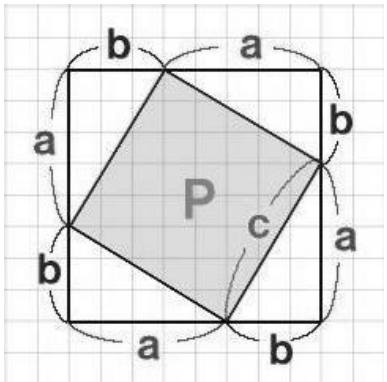
a と b は直角三角形の直角を挟む 2 辺を表し、c は斜辺を表す



ピタゴラスの定理の逆

a 、b 、c を三角形の 3 辺の長さとすると、「　$a^2+b^2=c^2$　」が成り立てばその三角形は、辺 c を斜辺とする直角三角形である

一応、ピタゴラスの定理の証明を記載しておきます。



1 辺の長さが、$(a+b)$の正方形を考えます。

この正方形の面積は、$(a+b)^2$　　… ①

また、その正方形を左図のように、区切っていくとその正方形は、合同な直角三角形 4 つと正方形 1 つに分けられます。

直角三角形 1 つの面積は、$\dfrac{(a\times b)}{2}$　　正方形 P の面積は、c^2

これらの面積の合計は、$c^2+\frac{(a\times b)}{2}\times 4=c^2+2ab$　　… ②

①②より $(a+b)^2=c^2+2ab$　$a^2+2ab+b^2=c^2+2ab$

$a^2+b^2=c^2$　となり、証明できます。

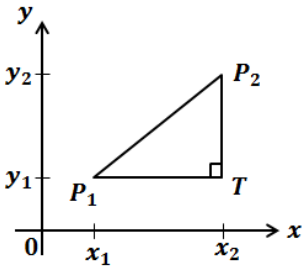
今後学習する「ベクトル演算」で、ピタゴラスの定理が頻繁にでてきます。それに先立ち、ここではまずピタゴラスの定理を使って、画面上の 2 点間の距離を計算する方法を学習しておきましょう。

「　直線の方程式　」

東京情報クリエイター工学院

◆距離の公式

右図に示したように、2 つの点 $P_1(x_1, y_1)$ $P_2(x_2, y_2)$ が与えられたとします。
すると、線分 P_1P_2 を斜辺とする直角三角形を描くことができます。
この三角形の 3 番目の頂点の座標は $T(x_2, y_1)$ です。



図からわかるように、辺 P_1T の長さは、「　　」
辺 P_2T の長さは、「　　」
ピタゴラスの定理を使うと、
$(P_1P_2)^2 = (P_1T)^2 + (P_2T)^2 =$
となります。これは、距離の公式として、まとめることができます。

距離の公式

$P_1(x_1, y_1)$ $P_2(x_2, y_2)$**のとき**「　$P_1P_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$　」

【例題 5】画面上の 2 点間の距離

ある物体は中心が $(25, 80)$ にあり、もう一つの物体は中心が $(55, 40)$ にあります。これら 2 つの物体の中心間の距離を求めなさい。

【解答】

距離の公式より求めることができますね。

$P_1P_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} =$

◆中点の公式

距離の公式に関連する公式として便利なものに、中点の公式があります。画面上の 2 つの物体間の中間の位置が必要になったり、境界円を考えるときに中心の座標を求めたりといろいろに使います。

中点の公式

点 $P_1(x_1, y_1)$ と $P_2(x_2, y_2)$ の中点は

「　$\left(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}\right)$　」で表される

【例題 6】画面上の 2 点間の中点

ある物体は中心が$(25, 80)$ にあり、もうひとつの物体は中心が$(55, 40)$ にあります。2 つの物体の中心間の中点を求めなさい。

【解答】

中点の公式より、

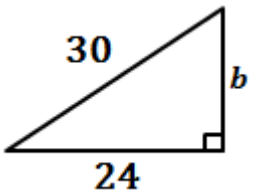
$$\left(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}\right) =$$

「　直線の方程式　」

東京情報クリエイター工学院

【練習問題 5】

１．図に示した図形の辺 b の長さを求めなさい。



２．２点 $A(-3,4)$　$B(2,-1)$ の距離求めなさい。

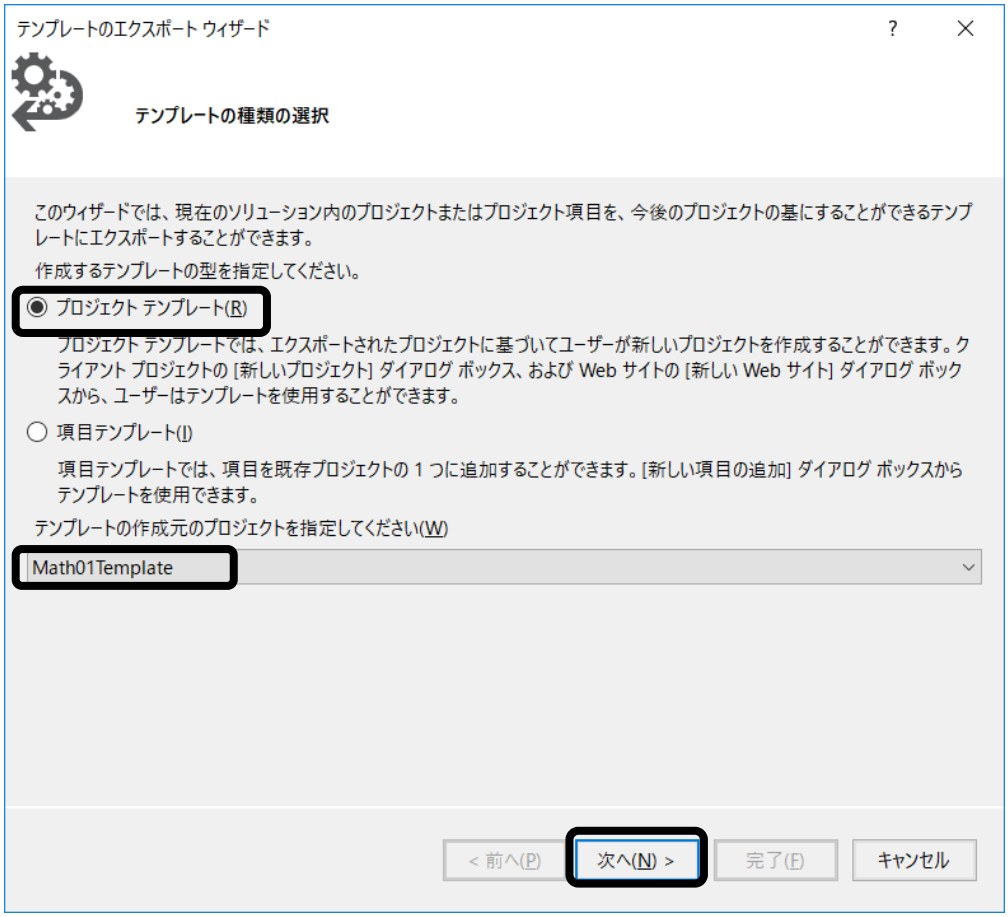
３．２点 $A(30,80)$　$B(150,130)$ の中点を求めなさい。

◆テンプレートの作成

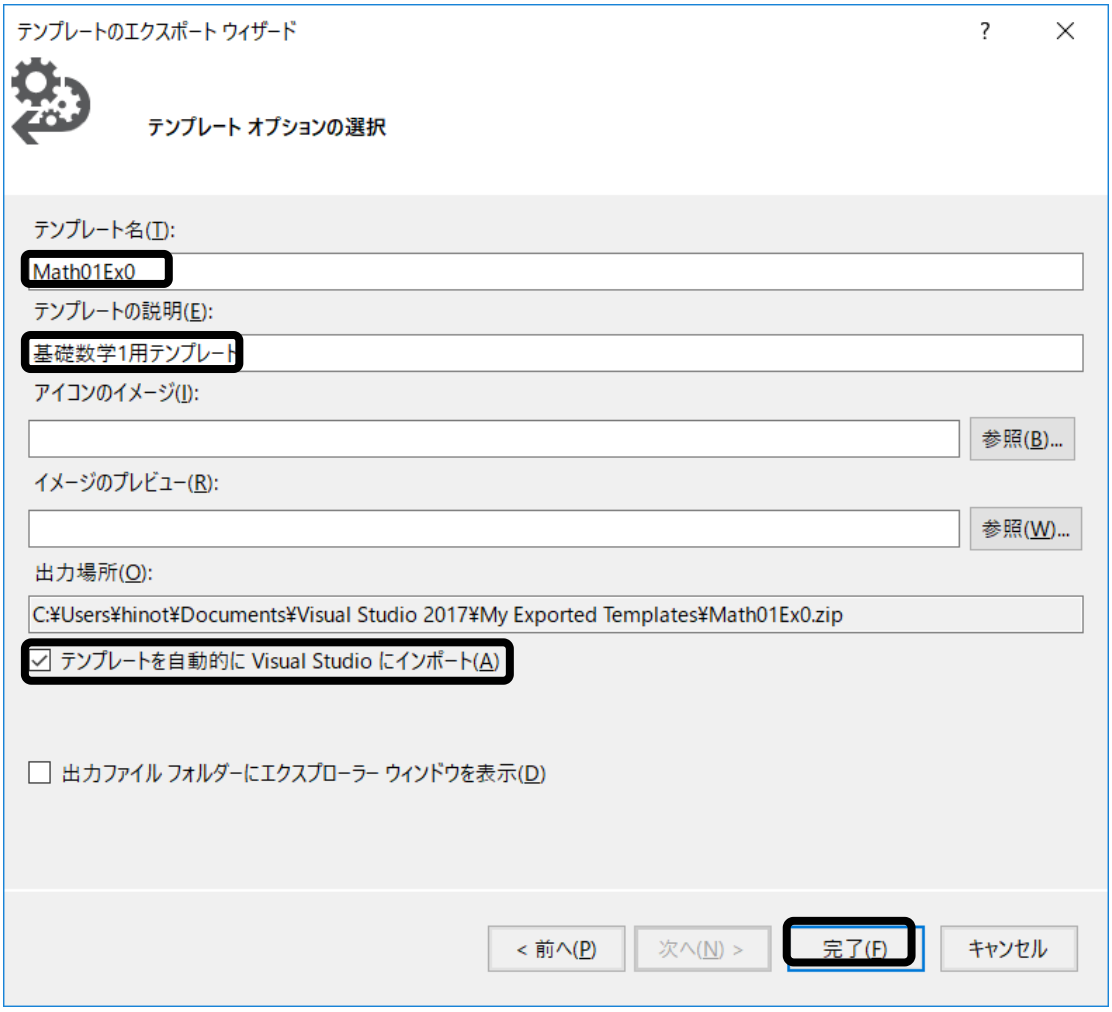
これから作成するプログラムを簡単に作成するために、プロジェクトテンプレートを最初に作成しましょう。

１．テンプレートの作成

①メニュー[プロジェクト]－[テンプレートのエクスポート]をクリックします。
②[プロジェクトテンプレート]が選択されていることを確認します。
③[テンプレートの作成元のプロジェクトを指定してください]に[Math01Template]を選択します。
④次へボタンをクリックします。



⑤テンプレート名を[Math01Ex0]に変更し、テンプレートの説明を「基礎数学１用テンプレート」と記入します。
⑥[テンプレートを自動的に Visual Studio にインポート]にチェックが入っていることを確認します。[出力ファイルフォルダーにエクスプローラウィンドウを表示]のチェックはいりません。
⑦以上を確認したら、完了ボタンをクリックします。



２．使用方法

新規プロジェクト作成時に[Visual C#]を選択すると、作成した[Math01Ex0]が表示されるので、[Math01Ex0]を選択すれば、基礎数学１用プロジェクトが作成されます。

「　直線の方程式　」

東京情報クリエイター工学院

◆演習問題

マウスから入力された 2 点の距離と中点を求めるプログラムを作成しましょう。

１．プログラム作成するための知識

(1)マウス入力・座標関連

①マウスからの入力：List<PointF> p = input.GetPoint(2, "コメント");

②入力された座標の取得：Vector2 p0 = new Vector2(p[0].X, p[0].Y);
Vector2 p1 = new Vector2(p[1].X, p[1].Y);

③座標値：ｘ座標　p0.X　　ｙ座標　p0.Y

④座標の設定：Vector2 pp = new Vector2(100,100)

(2)数式　平方根：(float)Math.Sqrt(d)

(3)表示

①点を描く：Dot(p0,5)　//座標 p0 に点を描く　「5」は点の大きさを指定

②2 点間の線を引く：Line(p0,p1)　//p0 と p1 を線で結ぶ

③文字の表示：Text(p0,"テキスト",12)　//座標 p0 に「テキスト」と表示　「12」は文字の大きさ

２．準備

(1)プロジェクトの追加

①「ソリューションエクスプローラ」の「ソリューション」にマウスポインタを合わせて「右クリック」－「追加」－「新しいプロジェクト」

②「新しいプロジェクトの追加」で、「Visual C#」－「Math01Ex0」をクリックする。
（名前が「Math01Ex01」となっているはず）

(2)「Math01Ex01」をスタートアッププロジェクトに設定

３．プログラムの作成

(1)[MyDrawClass.cs]　変数の定義とマウス入力の処理

テンプレートを使用しているので、マウス入力 1 つになっている。今回は、マウス入力 2 つなので、そこを変更するのを忘れない！

```
public class MyDrawClass:Draws,IDraws
{
    //フィールドは通常と同じで必要に応じて記述
    InputState input;  //マウス入力
    Vector2 p0, p1, mid;  //★★修正　線分の端点と中点
    float d;  //★★追加　線分の距離

    //コンストラクタには初期設定などを記述
    public MyDrawClass()
    {
        input = new InputState();  //入力操作の定義
        input.MouseOn();           //マウス入力を有効にする
    }
```

「　直線の方程式　」

東京情報クリエイター工学院

```
//必要に応じてその他メソッドを追加する
//入力処理と計算処理
public void InputData()
{
    Init(); //属性の初期化（色・太さ・文字高さを既定値にする）
    Clear();//画面をクリアします。(座標軸はクリアしない)
    //マウス入力
    List<PointF> p = input.GetPoint(2, "線分の端点を 2 点入力");  //★★修正
    if (p.Count == 0) return;  //キャンセルされたら以下を実行しない

    //座標値取得
    p0 = new Vector2(p[0].X, p[0].Y);
    //★★以降追加
    p1 = new Vector2(p[1].X, p[1].Y);   //★★追加

    //2 点間の距離
    2 点間の距離(d)を求める式

    //中点
    中点(mid)を求める式

    Draw();
    //バッファ内のレンダリングする。
    //通常のメソッドを作成する際は最後にレンダリングする。
    Render();
}
public void Update()
{          }

public void Draw()
{
    SetPen(Color.Black, 1);//色と太さを変える。
    Line(p0, p1);           //線分を描画
    Text(p0, "p0" + p0,12);
    Text(p1, "p1" + p1,12);
    Text(new Vector2(-200, 200), "距離：" + d,12);  //距離の表示

    SetColor(Color.Red);   //色を変える。
    Dot(mid, 5);              //中点描画
    Text(mid, "中点：" + mid,12);
}
```

(2)[Form1.cs]

テンプレートで入力済。

「　直線の方程式　」

東京情報クリエイター工学院

◆発展問題

１．次の条件にあてはまる１次関数の式を求めなさい。

①傾きが 2 で、$x = 3$ とき、$y = 1$

②傾きが -3 で、点(2, 1)を通る

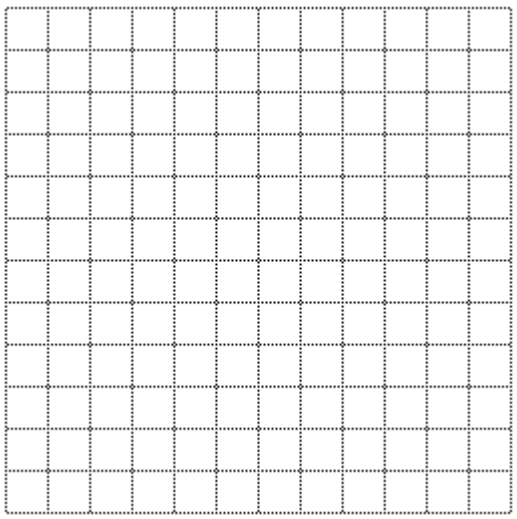
③(1, 3)と(3, 11)を通る

④(2, -1)と(4, -13)を通る

２．点 $(2,1)$ を通り直線 $2x - y + 6 = 0$ に平行な直線と垂直な直線の方程式を求めなさい。

３．次の方程式で表される直線を図示しなさい。

$2x + 3y - 6 = 0$

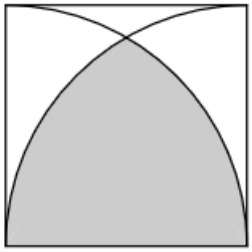


「　直線の方程式　」

東京情報クリエイター工学院

４．$y = ax + 5$ と $y = 2x + b$ の交点が $(2, -3)$ のときの a と b の値を求めなさい。

５．下の図は、一辺 $6cm$ の正方形と半径 $6cm$ のおうぎ形を組み合わせた図形です。影をつけた部分の面積を求めなさい。



[MyDrawClass.cs] プログラム例

```
public class MyDrawClass:Draws,IDraws
{
    //フィールドは通常と同じで必要に応じて記述
    InputState input;  //マウス入力
    Vector2 p0, p1, mid;  //★★修正　線分の端点と中点
    float d;  //★★追加　線分の距離

    //コンストラクタには初期設定などを記述
    public MyDrawClass()
    {
        input = new InputState();  //入力操作の定義
        input.MouseOn();           //マウス入力を有効にする
    }

    //必要に応じてその他メソッドを追加する
    //入力処理と計算処理
    public void InputData()
    {
        Init(); //属性の初期化（色・太さ・文字高さを既定値にする）
        Clear();//画面をクリアします。(座標軸はクリアしない)
        //マウス入力
        List<PointF> p = input.GetPoint(2, "線分の端点を 2 点入力");  //★★修正
        if (p.Count == 0) return;  //キャンセルされたら以下を実行しない

        //座標値取得
        p0 = new Vector2(p[0].X, p[0].Y);
        //★★以降追加
        p1 = new Vector2(p[1].X, p[1].Y);

        //2 点間の距離
        d = (p1.X - p0.X) * (p1.X - p0.X) + (p1.Y - p0.Y) * (p1.Y - p0.Y);
        d = (float)Math.Sqrt(d);

        //中点
        float mx = (p0.X + p1.X) / 2;
        float my = (p0.Y + p1.Y) / 2;
        mid = new Vector2(mx, my);

        Draw();
        //バッファ内のレンダリングする。
        //通常のメソッドを作成する際は最後にレンダリングする。
        Render();
    }
    public void Update()
    {          }

    public void Draw()
    {
        SetPen(Color.Black, 1);//色と太さを変える。
        Line(p0, p1);          //線分を描画
        Text(p0, "p0" + p0,12);
        Text(p1, "p1" + p1,12);
        Text(new Vector2(-200, 200), "距離：" + d,12);  //距離の表示

        SetColor(Color.Red);   //色を変える。
        Dot(mid, 5);              //中点描画
        Text(mid, "中点：" + mid,12);
    }
```