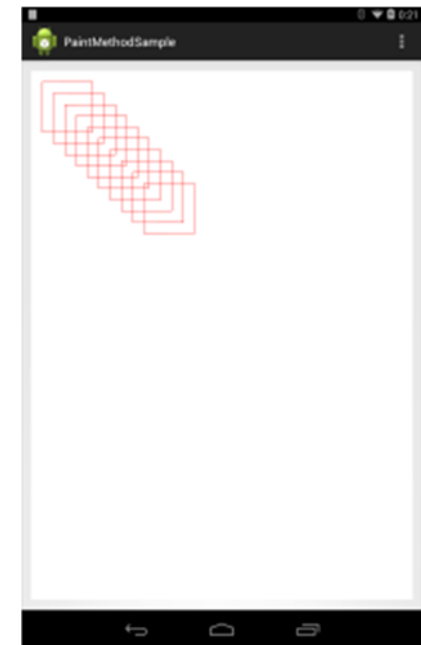


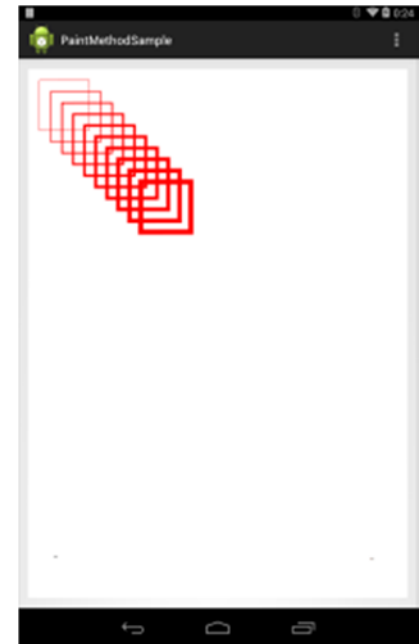
■ 輪郭線の描画

```
/*
 * setStyle(style)
 *
 *      Paint.Style.FILL : 図形の内部を塗りつぶす
 *      Paint.Style.STROKE : 図形の輪郭線のみを描く
 *      Paint.Style.FILL_AND_STROKE : 図形の内部を塗りつぶし輪郭線を描く
 */
private void setStyleSample(Canvas canvas) {
    Paint paint = new Paint();
    paint.setColor(Color.RED);
    paint.setStyle(Paint.Style.STROKE);
    RectF rect = new RectF();
    for(int i = 1; i <= 10; i++) {
        // setメソッド: 位置を再設定しながら描画
        rect.set(i * 22.5f, i * 22.5f, 100 + i * 22.5f, 100 + i * 22.5f);
        canvas.drawRect(rect, paint);
    }
}
```



■ 線分の太さ

```
/*
 * setStrokeWidth(線分の太さ)
 */
private void setStrokeWidthSample(Canvas canvas) {
    Paint paint = new Paint();
    paint.setColor(Color.RED);
    paint.setStyle(Paint.Style.STROKE);
    RectF rect = new RectF();
    for(int i = 1; i <= 10; i++) {
        paint.setStrokeWidth(i);
        rect.set(i * 22.5f, i * 22.5f, 100 + i * 22.5f, 100 + i * 22.5f);
        canvas.drawRect(rect, paint);
    }
}
```



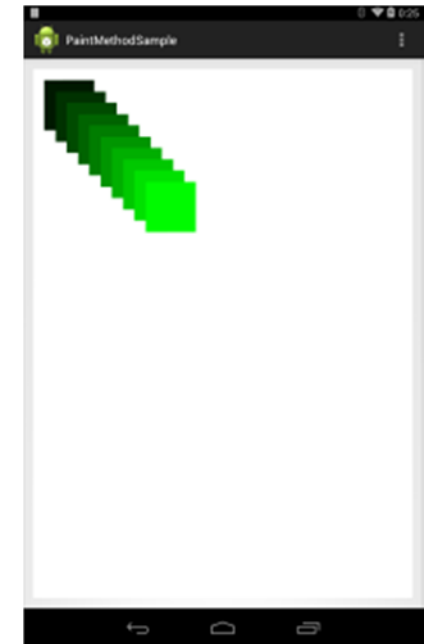
■ 線の先端

```
/*
 * setStrokeCap(先端)
 *
 *      Paint.Cap.SQUARE: 先端を四角描く
 *      Paint.Cap.ROUND: 先端を丸く描く
 *      Paint.Cap.BUTT: 線分の端末を越えて掛かれない
 */
private void setStrokeCapSample(Canvas canvas) {
    Paint paint = new Paint();
    paint.setColor(Color.RED);
    paint.setStyle(Paint.Style.STROKE);
    paint.setStrokeWidth(50f);
    float[] data = new float[] { 50, 50, 150, 150, 150, 150, 250, 50 };
    Paint.Cap[] caps = new Paint.Cap[] {
        Paint.Cap.SQUARE, Paint.Cap.ROUND, Paint.Cap.BUTT
    };
    for(int i = 0; i < caps.length; i++) {
        paint.setStrokeCap(caps[i]);
        canvas.drawLines(data, paint);
        for(int j = 1; j < data.length; j += 2) {
            data[j] += 100;
        }
    }
}
```



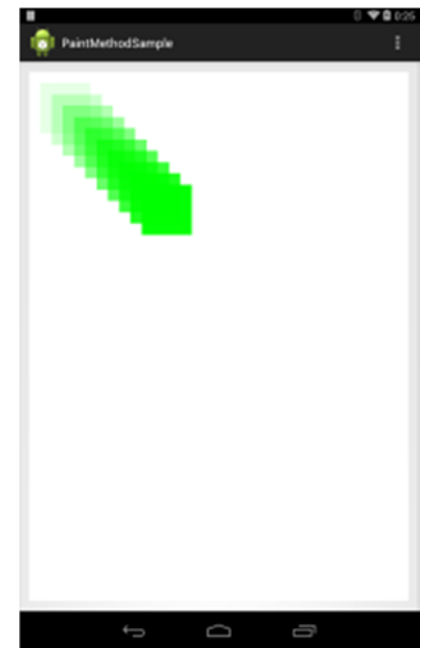
■色の設定

```
/*
 * Color.rgb(アルファ, 赤, 緑, 青)
 *
 *      赤, 緑, 青:    0 から 2 5 5
 */
private void setColorSample(Canvas canvas) {
    Paint paint = new Paint();
    RectF rect = new RectF();
    for(int i = 1; i <= 10; i++) {
        rect.set(i * 22.5f, i * 22.5f, 100 + i * 22.5f, 100 + i * 22.5f);
        paint.setColor(Color.rgb(255, 0, i * 25, 0));
        canvas.drawRect(rect, paint);
    }
}
```



■ 半透明の描画

```
/*
 * Color.setAlpha(アルファ値)
 *
 *      アルファ値:      0 から 2 5 5
 */
private void setAlphaSample(Canvas canvas) {
    Paint paint = new Paint();
    RectF rect = new RectF();
    paint.setColor(Color.GREEN);
    for(int i = 1; i <= 10; i++) {
        rect.set(i * 22.5f, i * 22.5f, 100 + i * 22.5f, 100 + i * 22.5f);
        paint.setAlpha(i * 25);
        canvas.drawRect(rect, paint);
    }
}
```



■テキストの描画

```
/*
 * 影の描画
 *     setShadowLayer(ぼかし, 横, 縦, 色)
 *         ぼかし: 影をぼかす半径を指定。 大きくなるほど影の境界があいまいとなる
 *         横, 縦: ずらす幅
 *         色: 影の色
 * フォントサイズ
 *     setTextSize(float値)
 * フォントのタイプフェイス
 *     setTypeface(Typeface)
 *     Typeface.create(フォントファミリー, フォントスタイル)
 *         フォントファミリー: フォントの名前 あるいは、
 *             DEFAULT, DEFAULT_BOLD, MONOSPACE, SERIF, SANS_SERIF
 *         フォントスタイル: NORMAL, BOLD, ITALIC, BOLD_ITALIC
 */
private void setTypefaceSample(Canvas canvas) {
    Paint paint = new Paint();
    paint.setShadowLayer(5f, 25f, 25f, Color.LTGRAY);
    paint.setTextSize(64);
    paint.setColor(Color.RED);
    paint.setTypeface(Typeface.create(Typeface.SANS_SERIF, Typeface.BOLD));
    canvas.drawText("Welcome!", 30, 100, paint);
    paint.setColor(Color.GREEN);
    paint.setTypeface(Typeface.create(Typeface.SERIF, Typeface.ITALIC));
    canvas.drawText("Welcome!", 30, 260, paint);
}
```

