

【実習5】 ブロックの衝突判定と消去

1 ソースコード

(1) テンプレート Jump の①の箇所に Jump050View を入力してください。

(2) 次のアプリケーションを新規作成してください。

Jump040View をコピー&ペーストして、ファイル名を「Jump050View」に変更してください。

プロジェクト名 : StepPro????(年組席) アプリケーション名 : Jump050View

```

/*   年   組   席   名前
 * Jump050View
 *   ボールとブロックとの衝突判定
 *   衝突したときのブロックの消去処理
 */
...

public class Jump050View extends SurfaceView
                        implements SurfaceHolder.Callback
{
    ...

    public Jump050View(Context context)
    {
        ...
    }
    //サーフェイス生成時に呼ばれる
    public void surfaceCreated(SurfaceHolder holder)
    {
        ...
    }
    public void draw(Canvas canvas)
    {
        ...
    }
    //ボールの移動
    private void moveBall()
    {
        ...
        //バーとの衝突判定
        ...

        //ブロックとの衝突判定
        for (int i=0;i<BLOCK.length;i++)
        {
            if (block[i]==0) continue;
            int r1=hitLineRect(
                ballX-ballVX,ballY-ballVY,ballX,ballY,(i%13)*blockWIDTH,
                50+(i/13)*blockHEIGHT,blockWIDTH,blockHEIGHT);
            if (r1>=0)
            {
                if (r1==0 || r1==1) ballVY=-ballVY;//上、下
            }
        }
    }
}

```

```

        if (r1==2 || r1==3) ballVX=-ballVX;//左、右
        block[i]=0;//ブロックの消去
        break;
    }
}
...
}

```

2 解説

数ミリ秒ごとにボール座標にボール速度が加算します。ボールが壁やブロックやバーに衝突すると、ボール速度が変わります。壁やブロックに衝突した時は、そのまま速度が逆になります。

```

//ブロックとの衝突判定
for (int i=0;i<BLOCK.length;i++)
{
    if (block[i]==0) continue;
    int r=hitLineRect(
        ballX-ballVX,ballY-ballVY,ballX,ballY,
        (i%13)*BMP_1_WIDTH,50+(i/13)*BMP_1_HEIGHT,
        BMP_1_WIDTH,BMP_1_HEIGHT);
    if (r>=0)
    {
        if (r==0 || r==1) ballVY=-ballVY;//上、下
        if (r==2 || r==3) ballVX=-ballVX;//左、右
        block[i]=0;//ブロックの消去
        break;
    }
}

```

ブロックすべてを判定しています。ブロックと衝突したら block[i]=0 により、表示しないようにしています。