

# 2Dアクションゲーム を作ろう

# 1.地上を作成しよう

地上で止まるので、画面の外に出ません。

(1)ソリューションエクスプローラでCMapをクリック

(2)データが1のブロックを作成するプログラムを追加する

```
for (; s < CGame::rangeXE + TIPX*2; s += TIPX*2) {  
    for (int i = (s / (TIPX*2)) * HSIZE; i < (s / (TIPX*2)) * HSIZE + HSIZE; i++) {  
        switch (data[i]) {  
            case 1:  
                b = new CTile();  
                b->setPosition(i,HSIZE);  
                b->setTexture(tile,1,1,16,16);  
                gettm()->add(b);  
                break;  
            case 9:  
                b = new CTile();  
                b->setPosition(i, HSIZE);  
                b->setColor(1.0f, 1.0f, 1.0f, 0.0f);  
                gettm()->add(b);  
                break;  
            }  
        }  
    }  
}
```

## 2.ブロックを作成しよう

ジャンプして乗れます。

(1)1.の続きに、データ2のブロックを追加するプログラムを追加する

```
case 1:
    b = new CTile();
    b->setPosition(i, HSIZE);
    b->setTexture(tile, 1, 1, 16, 16);
    gettm()->add(b);
    break;
case 2:
    b = new CTile();
    b->setPosition(i,HSIZE);
    b->setTexture(tile,33,1,48,16);
    gettm()->add(b);
    break;
case 9:
    b = new CTile();
    b->setPosition(i, HSIZE);
    b->setColor(1.0f, 1.0f, 1.0f, 0.0f);
    gettm()->add(b);
```

## 3.敵を出そう

敵の上にジャンプして乗ると撃退できます。敵に当たっても死にません。

(1)2.の続きに、データ3のブロックを追加するプログラムを追加する

```
case 2:
    b = new CTile();
    b->setPosition(i, HSIZE);
    b->setTexture(tile, 33, 1, 48, 16);
    gettm()->add(b);
    break;
case 3:
    b = new CKuribo();
    b->setPosition(i,HSIZE);
    b->setTexture(enemy,1,17,16,32);
    gettm()->addCollision(b);
    break;
case 9:
```

## 4.別の敵を出そう

敵の上にジャンプし、さらにジャンプすることで敵を攻撃することができます。

(1)3.の続きに、データ4のブロックを追加するプログラムを追加する

```
case 3:
    b = new CKuribo();
    b->setPosition(i, HSIZE);
    b->setTexture(enemy, 1, 17, 16, 32);
    gettm()->addCollision(b);
    break;
case 4:
    b = new CNokonoko();
    b->setPosition(i,HSIZE);
    b->setTexture(enemy,97,9,112,32);
    gettm()->addCollision(b);
    break;
case 9:
```

## 5.アイテムボックスを作ろう

アイテムボックスに下からジャンプすると、キノコが出現します。キノコに接触すると巨大化します。

(1)4.の続きに、データ5のブロックを追加するプログラムを追加する

```
case 4:
    ~
    break;
case 5:
    b = new CTile();
    b->setPosition(i,HSIZE);
    b->setColor(1.0f,1.0f,1.0f,0.0f);
    gettm()->add(b);
    k = new CAkakinoko();
    k->setPosition(i,HSIZE);
    k->setTexture(items,1,16,16,32);
    gettm()->addCollision(k);
    b = new CHatena();
    b->setPosition(i,HSIZE);
    b->setTexture(tile,385,1,400,16);
    gettm()->addCollision(b);
    break;
case 9:
```

## 6.敵に当たる処理

敵に当たると以下ようになります。

1. 巨大な時は、普通の大きさに戻ります
2. 普通の大きさの時は、死にます。

(1)ソリューションエクスプローラでCPlayerをクリックする

(2)void onCollision(const CBase &b) の中にある以下の命令を有効にする(4か所)

```
//          speed.y = PLAYER_SPEED_J;  
//          status = Stop;
```

の先頭の//を削除する

```
speed.y = PLAYER_SPEED_J;  
status = Stop;
```

# 7. マップを編集してみよう

プログラムの1430行目からはステージのデータです。データの値を変更することで、ステージの内容が変更されます。

データを追加するとマップは長くなります。  
データとマップ要素の関係は次のとおりです。  
1: 地上、2: ブロック、3: 敵、4: 敵、5: アイテムボックス

```
1, 1, 9, 9, 9, 9, 9, 9, 9, 9, //これ以前変更しない
1, 1, 0, 0, 0, 0, 0, 0, 0, 0, //10
1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 0, 0, 0, 6, 0, 0, 0, 0,
1, 1, 0, 0, 0, 0, 0, 0, 0, 0,
1, 1, 0, 4, 0, 2, 0, 0, 0, 0,
1, 1, 0, 0, 0, 5, 0, 0, 0, 0,
1, 1, 0, 0, 0, 2, 0, 0, 0, 0,
1, 1, 0, 0, 0, 0, 0, 0, 0, 0, //20
```

