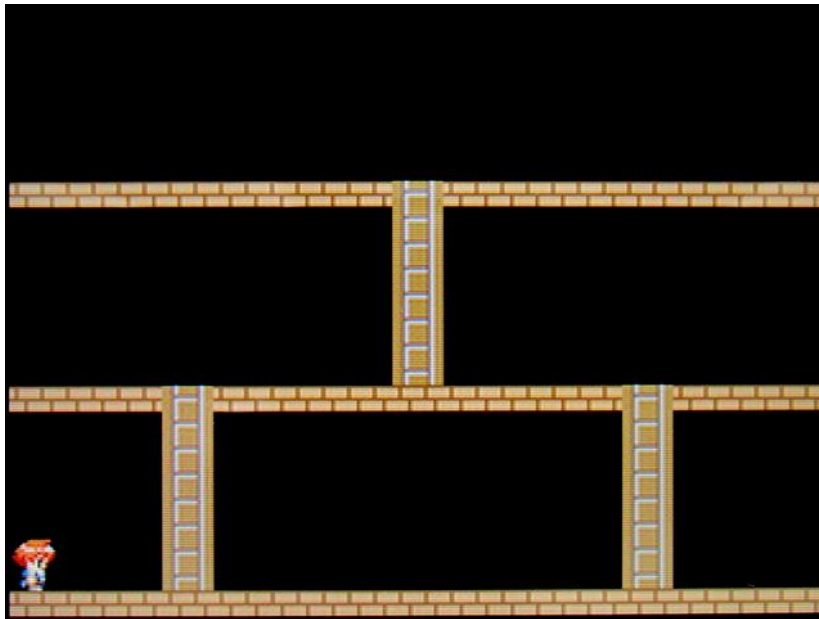


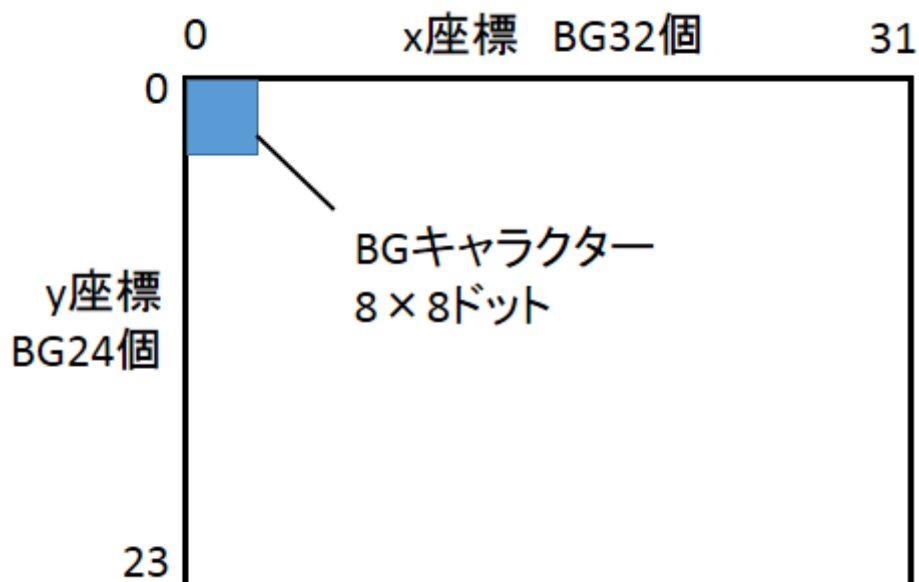
プチコンでアクションゲームを作る(2)

●背景を作る

これまでは真っ暗な画面でキャラクターを動かしていました。
背景(ステージ)を作って、それに沿ってキャラクターを動かしてみましょう。



プチコンの背景面(BG面)の座標は、以下のようにになっています。



BG面には、背景用のBGキャラクター(8×8ドット)を、横32個×縦24個並べられます。
(実際はBG面全体は、横64個×縦64個のBGキャラクターが並べられる大きなものです。
別紙の資料を参照)

まずは一番下の1階の床を表示してみます。

プログラムのタイトルを「ACT3」に変更して、初期設定のプログラムの下に、ステージを表示するプログラムを追加します。

```

1 '*** ACT3 ***
2
3 ACLS
4
5 'サイショ
6 X=0
7 Y=0
8 SPSET 0,64,2,0,0,2
9 SPANIM 0,4,10
10 SPOFS 0,X,Y
11
12 ' --- ステージ ---
13
14 BGPAGE 0
15
16 ' 1カ
17 FOR BX=0 TO 31
18 BGPUT 1,BX,23,677,8,0,0
19 NEXT BX
20
21 ' --- アルク ---
22 @ARUKU

```

タイトルを「ACT3」に変更

背景の描画を上画面に

画面の一番下に、床のBGキャラクターを左から右へ、くり返し表示して並べる。

プログラムを実行してみましよう。

画面の一番下に、床ができます。



プログラムの説明です。

まず、BGPAGE (ビージーページ) 命令で、背景(BG)面のどちらを使うかを指定します。

```
BGPAGE      0
            使う画面
```

使う画面は、0=上画面、1=下画面です。

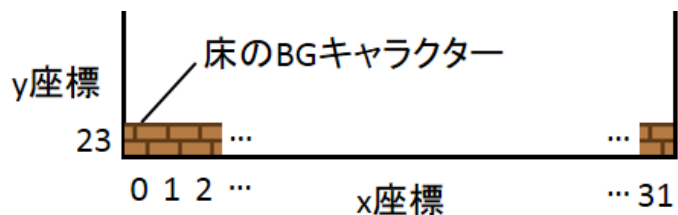
次に、FOR~NEXT のくり返しループで、BGPUT (ビージープット) 命令で背景キャラクターを並べています。

```
BGPUT      1      ,BX      ,23      ,677      ,8      ,0      ,0
            使うBG面  x座標  y座標  BG      パレット  横反転  縦反転
            0=手前  (0~63) (0~63) キャラ  番号    1=反転  1=反転
            1=奥      番号
```

床のBGキャラクターは677番を使っています。



y座標は23のまま、BX(x座標)を0~31まで変えてくり返すので、画面の一番下で、横一列に床のBGキャラクターが並べられます。



同じように、2階の床、3階の床を作ってみましょう。

```
12 ' --- ステージ ---
13
14 BGPAGE 0
15
16 ' 1カ
17 FOR BX=0 TO 31
18 BGPUT 1, BX, 23, 677, 8, 0, 0
19 BGPUT 1, BX, 15, 677, 8, 0, 0
20 BGPUT 1, BX, 7, 677, 8, 0, 0
21 NEXT BX
22
23 ' --- アルク ---
24 @ARUKU
```

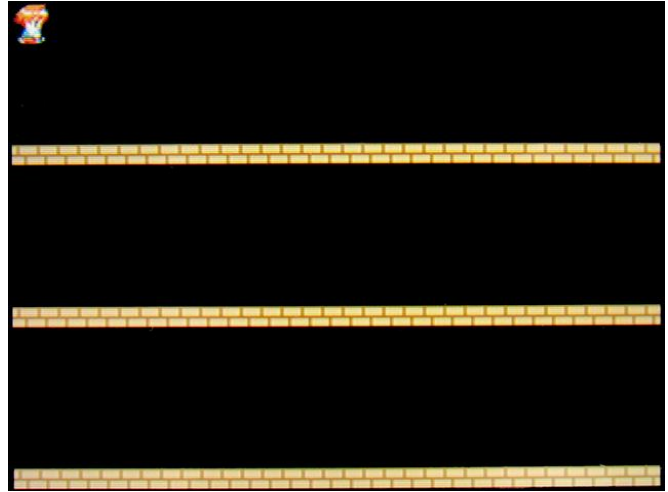
ループの中に、2階の床、3階の床を表示するBGPUT命令を追加

1 階～3 階の床が表示されます。

3 階 y=7

2 階 y=15

1 階 y=23



次に、それぞれの階の間を移動するために、はしごを表示してみましょう。

はしごの BG キャラクターは 28 番と 29 番を使います。



はしごは、1 階と 2 階の間に 2 本、2 階と 3 階の間に 1 本表示することになります。

```

12 ' --- ステージ ---
13
14 BGPAGE 0
15
16 ' 1F
17 FOR BX=0 TO 31
18 BGPUT 1, BX, 23, 677, 8, 0, 0
19 BGPUT 1, BX, 15, 677, 8, 0, 0
20 BGPUT 1, BX, 7, 677, 8, 0, 0
21 NEXT BX
22
23 ' ハシゴ 1F-2F
24 FOR BY=15 TO 22
25 BGPUT 1, 6, BY, 28, 1, 0, 0
26 BGPUT 1, 7, BY, 29, 1, 0, 0
27 BGPUT 1, 24, BY, 28, 1, 0, 0
28 BGPUT 1, 25, BY, 29, 1, 0, 0
29 NEXT BY
30

```

y 座標でループして、1 階と 2 階の間のはしごを 2 本表示

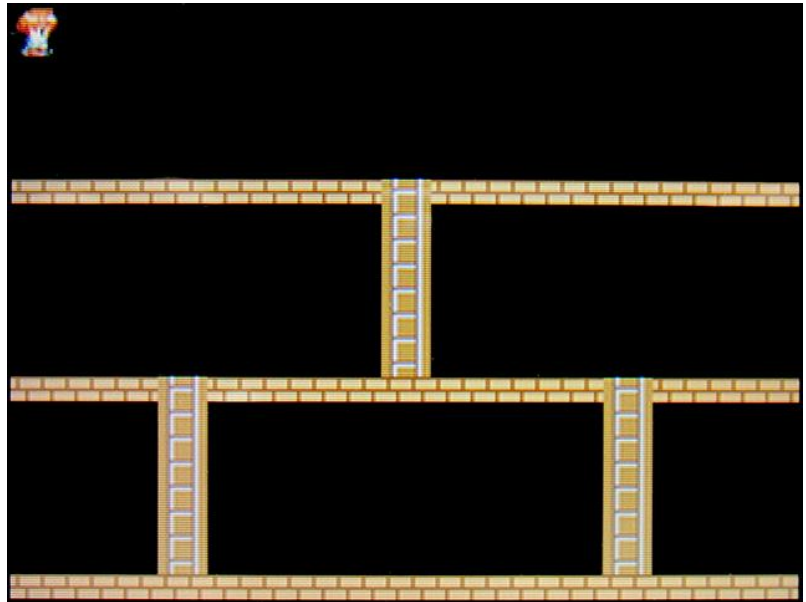
```

31 'ハシゴ' 2F-3F
32 FOR BY=7 TO 14
33 BGPOT 1, 15, BY, 28, 1, 0, 0
34 BGPOT 1, 16, BY, 29, 1, 0, 0
35 NEXT BY
36
37 ' --- フルク ---
38 @ARUKU

```

y 座標でループして、2 階と 3 階の間のはしごを 1 本表示

実行すると、はしごが表示されます。



最後に、キャラクターが空中に浮いてスタートするのは変なので、1 階からスタートするようにしましょう。

```

1 '*** ACT3 ***
2
3 ACLS
4
5 'サイヨ
6 X=0
7 Y=168
8 SPSET 0, 64, 2, 0, 0, 2
9 SPANIM 0, 4, 10
10 SPOFS 0, X, Y

```

●背景にそってキャラクターを左右に動かす

背景はできましたが、今のままだとキャラクターが背景とは関係なく、上下左右に自由に動かせてしまいます。

床の上を歩いたり、はしごを上り降りしたりするようにしましょう。

★キャラクターを動かすプログラムを改造

まず、プログラム先頭のタイトルを「ACT4」に変えましょう。

```
1 ' *** ACT4 ***
2
```

そして、キャラクターが歩くプログラムでいろいろ処理が増えるので、書き換えます。

```
37 ' --- アルク ---
38 @ARUKU
39
40 ' シュウシキ キー
41 B=BUTTON( )
42 IF B==1 AND Y>0 THEN GOSUB @UE
43 IF B==2 AND Y<175 THEN GOSUB @SHITA
44 IF B==4 AND X>0 THEN GOSUB @HIDARI
45 IF B==8 AND X<239 THEN GOSUB @MIGI
```

(→続きは次のページへ)

最後に、以下のプログラムを追加します。

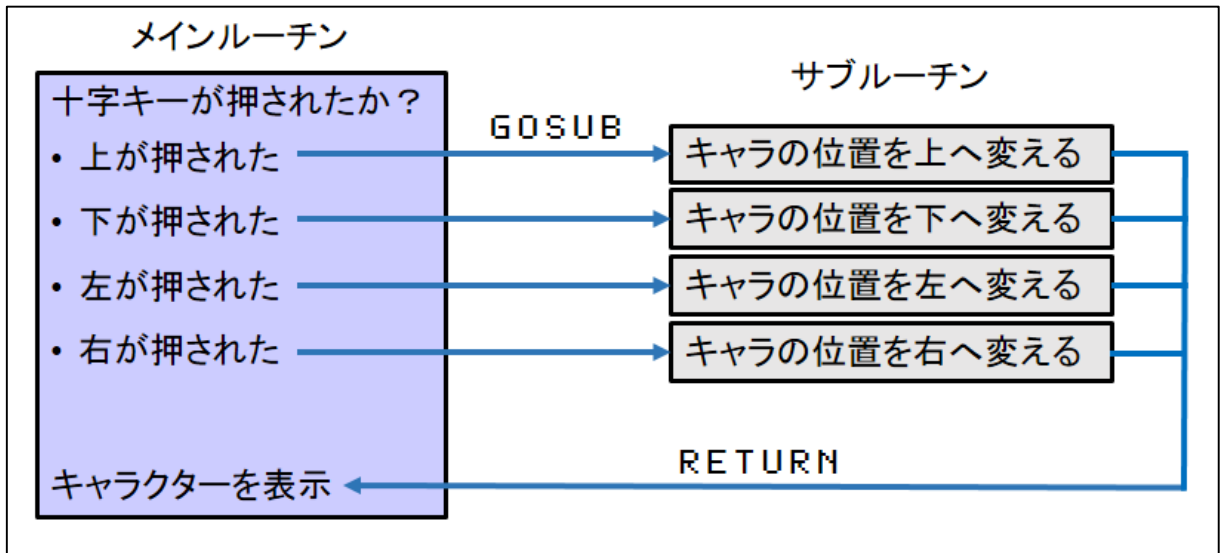
```
56 GOTO @ARUKU
57
58 ' --- ウィ ---
59 @UE
60
61 Y=Y-2
62 S=76
63 RETURN
64
65 ' --- シタ ---
66 @SHITA
67
68 Y=Y+2
69 S=68
70 RETURN
71
72 ' --- ヒタリ ---
73 @HIDARI
74
75 X=X-2
76 S=72
77 RETURN
78
79 ' --- ミキ ---
80 @MIGI
81
82 X=X+2
83 S=64
84 RETURN
```

新しい命令が出てきて、プログラムが長くなりました。

実行してみると、前と同じようにキャラクターが上下左右に動きます。

★サブルーチン

新しく **GOSUB** 命令、**RETURN** 命令が増えました。
これらはサブルーチンを使うための命令です。



今回の場合、十字キーが押されたかどうかをメインのプログラム(メインルーチン)で判断して、キーが押されていれば、**GOSUB** 命令で、上下左右それぞれの処理をするサブルーチンへジャンプしています。

サブルーチンからは、**RETURN** 命令で、メインのプログラムへ戻っています。

これまでは**IF**命令の後ろにいろいろな処理を書いていたのですが、このようにサブルーチンに分けることで、サブルーチンの中でいろいろなプログラムが書けます。

GOSUB 命令、**RETURN** 命令の文法は、以下のとおりです。

```

GOSUB           @UE
                  ジャンプ先の
                  サブルーチンのラベル
  
```

サブルーチンへジャンプします。

```

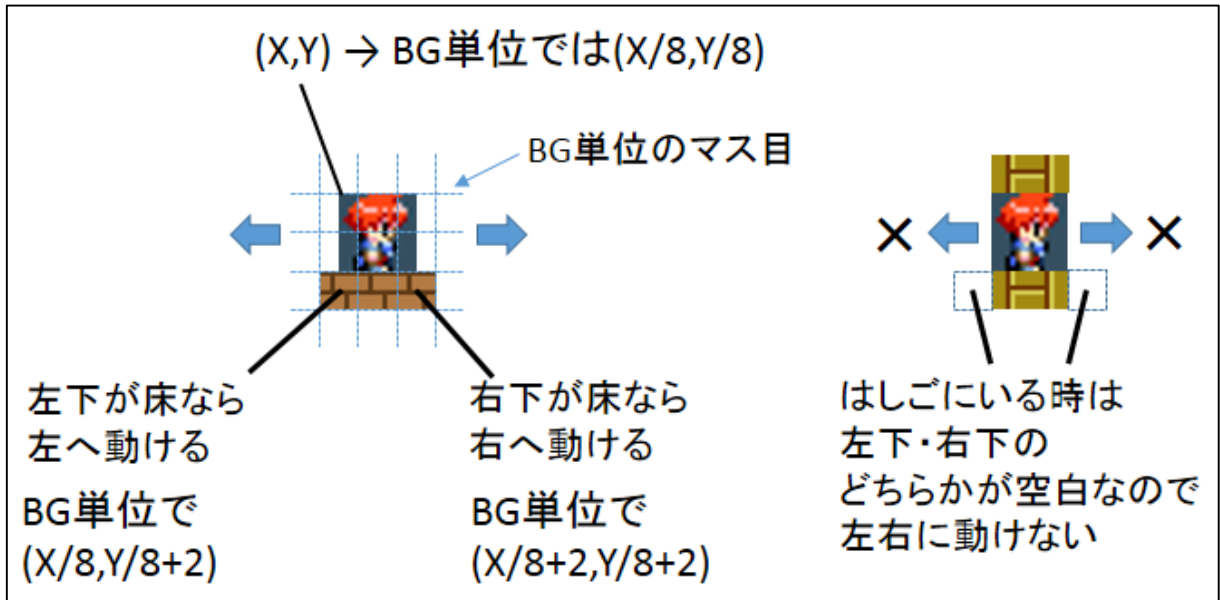
RETURN
  なし
  
```

メインルーチンへ戻ります。

★左右へ動ける条件

さて、準備ができたので、背景にそってキャラクターを動かすプログラムを作っていきます。
まず、床の上に立って左右に動かすことを考えましょう。

キャラクターと背景がどんな条件だったら、左右に動けるでしょうか？



つまり、左下が床なら(あるいは空白でないなら)、左へ動けます。

同じように、右下が床なら(あるいは空白でないなら)、右へ動けます。

背景にどんな BG キャラクターがあるか調べるには、**BGREAD**(ビージーリード)命令を使います。

BGREAD (1	, BX	, BY)	, C	, P	, H	, V
	使う BG 面	x 座標	y 座標	BG 番号	パレット	横反転	縦反転
	0=手前	(0~63)	(0~63)	読み出	読み出	読み出	読み出
	1=奥			し変数	し変数	し	し

読み出したい BG 面の x 座標、y 座標を指定すると、そこにある BG キャラクターの番号がわかります。何も無ければ番号は「0」になります。

BGREAD 命令を使って、左下・右下の背景が空白で無ければ、キャラクターが左右に動くプログラムにしてみましょう。

サブルーチンの「ヒダリ」「ミギ」を書き換えます。

```
72 ' --- ヒダリ ---
```

```
73 @HIDARI
```

```
74
```

```
75 BX=X/8
```

```
76 BY=Y/8
```

```
77 BGREAD(1, BX, BY+2), C, P, H, V
```

```
78 IF C=0 THEN RETURN
```

```
79 X=X-2
```

```
80 S=72
```

```
81 RETURN
```

```
82
```

```
83 ' --- ミギ ---
```

```
84 @MIGI
```

```
85
```

```
86 BX=X/8
```

```
87 BY=Y/8
```

```
88 BGREAD(1, BX+2, BY+2), C, P, H, V
```

```
89 IF C=0 THEN RETURN
```

```
90 X=X+2
```

```
91 S=64
```

```
92 RETURN
```

左下の BG キャラクターを読み取って
もし空白なら移動しないで戻る

右下の BG キャラクターを読み取って
もし空白なら移動しないで戻る

プログラムを実行してみましょう。

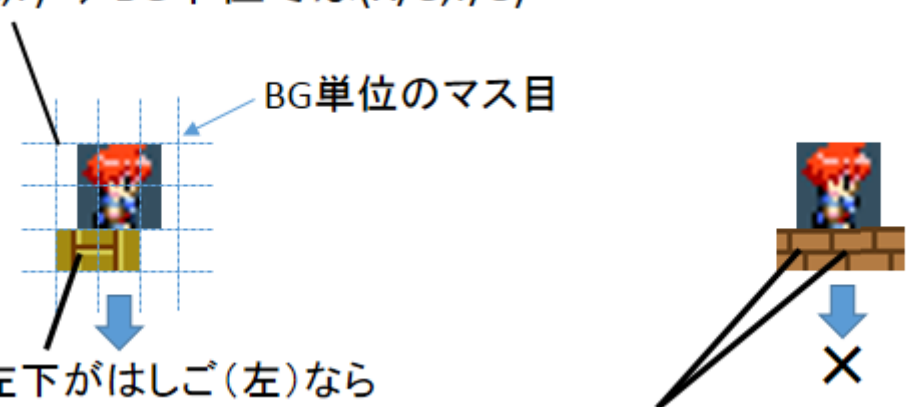
上下には自由に動けますが、左右は床の上か、はしごの上にはないと動けなくなります。

●背景にそってキャラクターを上下に動かす**★下へ動ける条件**

まず、下へ動ける条件を考えましょう。

下へ動くのは、はしごを下る時ですね。一方、床があったら下へは行けません。

(X,Y) → BG単位では(X/8,Y/8)



BG単位のマス目

左下がはしご(左)なら
下へ動ける
BG単位で
(X/8,Y/8+2)

左下・右下が床の時は
下へ動けない

★右下も同じ条件

右下がはしご(右)なら
下へ動ける
BG単位で
(X/8+2,Y/8+2)

下へ動かすサブルーチンを書き換えましょう。

```
65 ' --- シタ ---
```

```
66 @SHITA
```

```
67
```

左下・右下の BG キャラクターを読み取って
もしはしごでなければ、移動しないで戻る

```
68 BX=X/8
```

```
69 BY=Y/8
```

```
70 BGREAD(1, BX, BY+2), C1, P, H, V
```

```
71 BGREAD(1, BX+2, BY+2), C2, P, H, V
```

```
72 IF C1!=28 AND C2!=29 THEN R
```

```
ETURN
```

```
73 Y=Y+2
```

```
74 S=68
```

```
75 RETURN
```

プログラムを実行してみましょう。

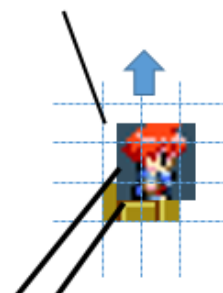
上へは自由に動けますが、はしごの所でしか下へ移動できません。

★上へ動ける条件

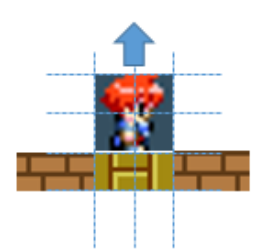
今度は上へ動ける条件を考えてみましょう。

上へ動けるのは、はしごを上る時ですね。キャラクターとはしごの重なり具合を考えます。

$(X,Y) \rightarrow$ BG単位では $(X/8,Y/8)$



このままだと...




左1つ下、または左2つ下が
はしご(左)なら、上へ動ける
BG単位で
 $(X/8,Y/8+1)$ 、 $(X/8,Y/8+2)$


※左2つ下だけ見ると
はしごの一番下でのぼれない

★右下も同じ条件

BG単位で
 $(X/8+2,Y/8+1)$
 $(X/8+2,Y/8+2)$
が、はしご(右)なら
上へ動ける

はしごの一番上でも
上へ移動してしまうので
空中へ行ってしまふ





はしごの一番上に立った時は
BGのマス目にぴったり合うので
Yが8で割り切れるはず。
この時は、それ以上のぼらない。

上へ動かすサブルーチンを書き換えます。
条件が多いので、ちょっと複雑なプログラムになります。

```

58 ' --- UI ---
59 @UE
60
61 BX=X/8
62 BY=Y/8
63 BGREAD(1, BX, BY+1), C1, P, H, V
64 BGREAD(1, BX, BY+2), C2, P, H, V
65 BGREAD(1, BX+2, BY+1), C3, P, H, V
66 BGREAD(1, BX+2, BY+2), C4, P, H, V
67 IF C1!=28 AND C2!=28 AND C
3!=29 AND C4!=29 THEN RETURN
68 IF ((C1==0 AND C2==28) OR (C
3==0 AND C4==29)) AND Y%8==0 TH
EN RETURN
69 Y=Y+2
70 S=76
71 RETURN

```

左下のBGを
読み取り

右下のBGを
読み取り

はしごが無か
ったら戻る

はしごの一番上にいる状態で
Y座標が8で割り切れたら、戻る
(それ以上はのぼらない)

新しく「%」という計算が出てきました。これはわり算の「あまり」を出す計算です。
「Y%8」で、「Yを8でわったあまり」が出ます。

プログラムを実行してみましょう。背景
の床やはしごにそって、キャラクター
が動きます。

