マニュアルに戻る。

シューティングゲーム

今まで学んだことを応用して、このActivityではシューティングゲームを作る。シューティングゲームの作り方を理解しておくと、さまざまなゲームに発展させて楽しむことができる。

ただし、いくつか難易度の高い点がある。解説をよく読んで、プログラムと見比べながら理解してほしい。プログラムは、ステップ1からステップ6までに分かれている。順に入力しながら動作を確認してほしい。全体のプログラムはステップ6の節で見ることができる。

作成するゲーム

画面の下部には主役のカメがいる。ボタンで左右に操作する。画面の上部には敵がいる。主役のカメからロケットを発射して攻撃することができる。敵に命中すると、画面下の得点欄に得点が加算されて行く。制限時間内にすべての敵を破壊することができればクリアとなる。

主役を作る（ステップ1）

最初に、主役のオブジェクトを画面に置くことにする。このゲームでは、画面の下にいる「かめた」が、上から降りてくる敵を攻撃する。まず、かめたを作り、ボタンで左右に動かせるようにしてみよう。

次のプログラムでは、「かめた」という名前のタートルオブジェクトを生成し、「90 左回り」で上向きにし、「ペンなし」で線を描かないように設定した後、「0 150 位置」で画面の下の中央付近に置いている。続いて、「左」と「右」という名前のボタンオブジェクトを作り、かめたの下に置いた後、押されたときにかめたの位置をx方向に20または-20だけ動かすように設定している。ボタンにはLEFT RIGHTを定義しているため、キーボードの左右の矢印キー（「←」、「→」）で操作することもできる。

プログラムを実行すると、画面にタートルとボタンが表示され、ボタンを押すことでタートルを左右に動かすことができる。

```
// 主役と左右の動き（ステップ1）
かめた＝タートル！作る 90 左回り ペンなし 0 -150 位置。
左=ボタン！"左" "LEFT" 作る -200 -180 位置。
左:動作＝「かめた！-20 0 移動する」。
右=ボタン！"右" "RIGHT" 作る 50 -180 位置。
右:動作＝「かめた！20 0 移動する」。
```

弾を発射する（ステップ2）

弾を発射するには、どのようなプログラムを書けばよいだろうか。弾は、「かめた」がいる位置から出て行く必要がある。そして、敵に向かって進んでいく。

ここでは、「かめた」の分身を作り、弾として前進させることにした。次のプログラムでは、「かめた」に「発射」というメソッドを定義し、その中でタイマーオブジェクトを作り、自分を20歩ずつ前進させ

るようにしている。タイマーの実行間隔は標準（0.1秒間隔）である。続いて、「発射」という名前のボタンオブジェクトを作り、左右のボタンの間に置いた後、押されたときに「かめた」を「作る」で複製し、55歩前進させた後、“rocket.gif” 変身する」でロケットの姿にして、先ほど作った「発射」を実行させている。ボタンには**UP**を定義しているため、キーボードの上向きの矢印キー「↑」で操作することもでき る。

ここで、最初に55歩前進している理由は、後から「衝突」を定義したときに、「かめた」と弾が衝突してお互いが消えてしまう事故を防ぐためである。タートルの変身前の姿は半径が20程度であるため、55歩離れてしまえば衝突は発生しない。

かめたの「発射」メソッドでは、タイマーオブジェクトを作った後、続けてブロックを与えて実行している。このように、メッセージの**カスケード**を使うことで、プログラムを簡潔に記述している。

```
// 発射ボタン（ステップ2）
かめた：発射＝「タイマー！作る「自分！20 歩く」実行」。
発射=ボタン！"発射" "UP" 作る -50 -180 位置 100 45 大きさ。
発射:動作＝「かめた！作る 55 歩く "rocket.gif" 変身する 発射」。
```

敵たちを作る（ステップ3）

画面の上部に、敵を配置する。敵は横一列に並ぶようにした。後で敵をまとめて動かしたり、敵の残っている数を管理したりするために、敵を配列で管理する。

次のプログラムでは、最初に「敵たち」という名前の配列を作っている。続いて、「敵」という名前のタートルオブジェクトを作り、“ayumiAka.gif” 変身する」で赤いタートルに変身させた後、「ペンなし」で線を描かないようにして、「90 右回り」で下を向かせている。そして次の行で、（-300, 200）の位置に移動させた後、配列「敵たち」に格納している。続く4行では、敵を「作る」で複製した後、それぞれを画面に配置してから配列「敵たち」に格納している。

最後の4行では、プログラムを簡潔に書くために、「敵」オブジェクトの複製と画面への配置、配列への格納を1行ずつ記述した。もしプログラムが分かりづらければ、次のように、2行ずつに分けて記述することもできる。

```
// 1行で書いた例
敵たち！（敵！作る -200 200 位置）書く。
 // 2行で書いた例
敵2＝敵！作る -200 200 位置。
敵たち！（敵2）書く。
```

```
// 敵たちの生成（ステップ3）
敵たち＝配列！作る。
敵＝タートル！作る "ayumiAka.gif" 変身する ペンなし 90 右回り。
敵たち！（敵！-300 200 位置）書く。
敵たち！（敵！作る -200 200 位置）書く。
敵たち！（敵！作る -100 200 位置）書く。
敵たち！（敵！作る 0 200 位置）書く。
敵たち！（敵！作る 100 200 位置）書く。
```

敵たちの移動（ステップ4）

敵は、1列に横に並んだまま、少しずつ右または左に動き、画面の端まで動くと少し下に進んで、先ほどとは逆の向きに横に移動するようにした。

次のプログラムでは、最初に「時計」という名前のタイマーを作り、実行間隔を1秒に設定している。続く3行では、タイマーを使って敵を移動する。「時計」という1つのタイマーオブジェクトに続けて「実行」を送ることで、「ひとつの実行が終ったら次を実行」という形で、**タイマーの逐次実行**を実現している。

最後の3行では、プログラムを簡潔に書くために、タイマーの実行と配列内の各オブジェクトの実行を1行に記述した。もしプログラムが分かりづらければ、次のように、2行ずつに分けて記述することもできる。ここで「右移動」はタイマーによって実行されるブロックなので、全体を括弧（「…」）で囲む必要がある。

```
// 1行で書いた例
時計!6 回数「敵たち！「｜敵|敵！30 0 移動する」それぞれ実行」実行。
 // 2行で書いた例
右移動＝「敵たち！「｜敵|敵！30 0 移動する」それぞれ実行」。
時計!6 回数（右移動）実行。
```

```
// 敵たちの移動（ステップ4）
時計＝タイマー！作る 1秒 間隔 。
時計!6 回数「敵たち！「｜敵|敵！30 0 移動する」それぞれ実行」実行。
時計!1 回数「敵たち！「｜敵|敵！0 -30 移動する」それぞれ実行」実行。
時計!6 回数「敵たち！「｜敵|敵！-30 0 移動する」それぞれ実行」実行。
```

衝突の定義（ステップ5）

弾が当たったときに、敵と弾が消えるようにする。敵が消えるときは、敵を管理するための配列「敵たち」からも削除する。

次のプログラムでは、最初に「かめた」に「衝突」を定義して、何かにぶつかった場合に自分を消すようにしている。実際には弾がこの衝突を実行するが、「かめた」に定義することによって、「かめた」を複製して作られた弾にも自動的に定義されることになる。

続く3行では、配列「敵たち」に含まれるすべての敵オブジェクトに衝突を定義している。敵は何かにぶつかった場合に、自分を消してから、配列「敵たち」から自分を削除する。

```
// 衝突定義（ステップ5）
かめた：衝突＝「自分！消える」。
敵：衝突＝「自分！消える。敵たち！（自分）消す」。
```

終了判定（ステップ6）

ゲームのプログラムでは、最後に定められた条件をクリアしたかどうかの判定が行われる。このゲームでは、時間内にすべての敵を消したかどうかを判定することにした。

次のプログラムでは、制限時間を15秒にするために、「制限時間」という変数を作り15を代入している。続いて、残り時間を表示するために、「カウントダウン」という名前の**フィールド**を作っている。

続いて、「終了時計」という名前のタイマーオブジェクトを作り、1秒間隔で「制限時間」だけ動くようにした。

続く6行は、「終了時計」の実行内容であり、1秒間隔で「制限時間」の秒数だけ実行される。何回目の実行かは、先頭のパラメータ□n□に入る。フィールド「カウントダウン」には、「制限時間 - n□という式で、残り時間を表示するようにした。続いて、配列「敵たち」の要素数を調べ、値がゼロであれば、ラベルで「おめでとう！！！」というメッセージを表示する。そして、終了時計を中断してカウントダウンを止める。

このプログラムでは、制限時間内にクリアしたときにメッセージを表示している。もしクリアできなかったときに別のメッセージを表示したい場合は、「最後に実行」で終了時計の終了を待って、そのときの「敵たち」の要素数を調べればよい。

```
// 終了判定（ステップ6）
制限時間＝15。
カウントダウン=フィールド！作る 150 0 移動する。
終了時計＝タイマー！作る 1秒 間隔（制限時間）時間。
終了時計□□□n|カウントダウン！（制限時間 - n□書く。
  「（敵たち！要素数?）＝＝0」！なら「
    ラベル！"おめでとう！！！"作る -100 200 位置。
    終了時計！中断
  」実行
」実行。
```

以上でシューティングゲームは完成である。

最後にステップ1からステップ6までの全体のプログラムを掲載しておく。

```
// 主役と左右の動き（ステップ1）
かめた＝タートル！作る 90 左回り ペンなし 0 -150 位置。
左=ボタン！"左" "LEFT" 作る -200 -180 位置。
左:動作＝「かめた！-20 0 移動する」。
右=ボタン！"右" "RIGHT" 作る 50 -180 位置。
右:動作＝「かめた！20 0 移動する」。

// 発射ボタン（ステップ2）
かめた：発射＝「タイマー！作る「自分！20 歩く」実行」。
発射=ボタン！"発射" "UP" 作る -50 -180 位置 100 45 大きさ。
発射:動作＝「かめた！作る 55 歩く "rocket.gif" 変身する 発射」。

// 敵たちの生成（ステップ3）
敵たち＝配列！作る。
敵＝タートル！作る "ayumiAka.gif" 変身する ペンなし 90 右回り。
敵たち！（敵！-300 200 位置）書く。
敵たち！（敵！作る -200 200 位置）書く。
敵たち！（敵！作る -100 200 位置）書く。
敵たち！（敵！作る 0 200 位置）書く。
敵たち！（敵！作る 100 200 位置）書く。

// 敵たちの移動（ステップ4）
時計＝タイマー！作る 1秒 間隔。
時計!6 回数「敵たち！「｜敵|敵！30 0 移動する」それぞれ実行」実行。
時計!1 回数「敵たち！「｜敵|敵！0 -30 移動する」それぞれ実行」実行。
時計!6 回数「敵たち！「｜敵|敵！-30 0 移動する」それぞれ実行」実行。
```

```
// 衝突定義（ステップ5）
かめた：衝突＝「自分！消える」。
敵：衝突＝「自分！消える。敵たち！（自分）消す」。

// 終了判定（ステップ6）
制限時間＝15。
カウントダウン=フィールド！作る 150 0 移動する。
終了時計＝タイマー！作る 1秒 間隔（制限時間）時間。
終了時計□□□n|カウントダウン！（制限時間-n□書く。
  「（敵たち！要素数?）＝＝0」！なら「
    ラベル！"おめでとう！！！"作る -100 200 位置。
    終了時計！中断
  」実行
」実行。
```