

宝物拾いゲーム

これまでのActivityでは、タートルを移動させることによるグラフィックスと、マウスでクリックしてプログラムを対話的に操作するためのボタン、タイマーによる繰り返し実行などを学んだ。このActivityでは、これまでに学んだことを使って、簡単なゲームプログラムを作る。

作成するゲーム

作成するゲームは宝物拾いゲームである。ボタンでタートルを操作して、宝物を拾い集めていく。



画面にはカメとトンボの姿をしたタートルが表示されており、上に2個のボタンが並んでいる。タートルはタイマーオブジェクトにより自動的に前進する。トンボは乱数により、実行するたびに画面の異なる位置に表示される。ユーザーは、タートルの向きを左右に変える2つのボタンを操作して、タートルを効率よく動かしてトンボのおもちゃを拾う。

タートルを操作する（ステップ1）

まず、画面にタートルを作る。ここでは「かめた」という名前にした。次に、画面の上に2つのボタンを作る。「左」ボタンを押すと、かめたはその場で左に30度回転する。「右」ボタンを押すと、かめたはその場で右に30度回転する。

ボタンの作成時に、表示するラベルの後にキーを表す文字列（[ショートカットキー^{1\)}](#)）を指定すると、キーボードからボタンを操作できる。ここではLEFTとRIGHTを指定することで、ボタンを押すのと同じ動作を左右の矢印キーで行えるようにした。



```
// タートルを操作する（ステップ1）
かめた = タートル！作る。
左ボタン = ボタン！"左" "LEFT" 作る。
左ボタン：動作 = 「かめた！30 左回り」。
右ボタン = ボタン！"右" "RIGHT" 作る。
右ボタン：動作 = 「かめた！30 右回り」。
```

タートルを前進させる（ステップ2）

次に、タートルを前進させる。次のプログラムでは、「時計」という名前のタイマーオブジェクトを作り、かめたを10歩ずつ前進させる動作を200回繰り返す。繰り返す間隔は、標準の0.1秒である。よって、20秒間動作することになる（0.1秒×200回）。



```
// タートルを前進させる（ステップ2）
時計 = タイマー！作る 200 回数。
時計！「かめた！10 歩く」実行。
```

宝物を画面に置く（ステップ3）

次に、トンボの姿をしたタートルオブジェクトを画面に置く。これらのオブジェクトは、画面に置いた後は操作しないため、特に名前を付ける必要はない。タートルの姿は**変身する**で変更することができる。姿はJPEG, GIF, PNGなどの形式の画像ファイルで指定することが可能であり、あらかじめいくつかの画像ファイルが用意されている²⁾

画面に置く位置は、乱数で指定している。横の位置を-299~300の幅600の範囲に、縦の位置を-149~150の高さ300の範囲にするために、それぞれ1~600の乱数と1~300の乱数を生成し、その値から300と150を引いて利用している。乱数や引き算などの数式は、括弧(...)で囲んで記述する。



```
// 宝物を画面に置く（ステップ3）
宝 = タートル！作る "tonbo.gif" 変身する ペンなし。
宝！（乱数（600）-300）（乱数（300）-150）位置。
宝！作る（乱数（600）-300）（乱数（300）-150）位置。
宝！作る（乱数（600）-300）（乱数（300）-150）位置。
```

宝物を拾う（ステップ4）

前のプログラムでは、かめたがトンボに重なっても何も起きず、かめたはトンボを通り抜けてしまった。ここでは、かめたがトンボに衝突したときに、トンボの姿を消すことで拾ったことを表現する。次のプログラムでは、かめたに衝突メソッドを定義し、衝突したときに相手に**消える**命令を送っている。



```
// 宝物を拾う（ステップ4）
かめた：衝突 = 「 | 相手 | 相手！消える」。
```

最後にステップ1からステップ4までの全体のプログラムを掲載しておく。

```
// タートルを操作する（ステップ1）
かめた = タートル！作る。
左ボタン = ボタン！"左" "LEFT" 作る。
左ボタン：動作 = 「かめた！30 左回り」。
右ボタン = ボタン！"右" "RIGHT" 作る。
右ボタン：動作 = 「かめた！30 右回り」。

// タートルを前進させる（ステップ2）
時計 = タイマー！作る 200 回数。
時計！「かめた！10 歩く」実行。

// 宝物を画面に置く（ステップ3）
宝 = タートル！作る "tonbo.gif" 変身する ペンなし。
宝！（乱数（600）-300）（乱数（300）-150）位置。
宝！作る（乱数（600）-300）（乱数（300）-150）位置。
宝！作る（乱数（600）-300）（乱数（300）-150）位置。

// 宝物を拾う（ステップ4）
かめた：衝突 = 「 | 相手 | 相手！消える」。
```

¹⁾

詳しくは[ch_reference](#) を参照。

2)

ローカル版では、ドリトルのフォルダに画像ファイルを置くことで、独自の画像ファイルを使うことも可能である。

From:

<https://dolittle.eplang.jp/> - プログラミング言語「ドリトル」

Permanent link:

https://dolittle.eplang.jp/ch_driveLast update: **2018/02/09 11:03**