

# プログラミング1(ドリトル)

22j1-109

教科書 P184-P185

# プログラミング1(ドリトル)

- プログラミング1(ドリトル)
- S:プログラミングとは何かよく理解でき、自分なりに考えることができた
- A:プログラミングとは何かよく理解できた
- B:プログラミングとは何か理解できた
- C:プログラミングとは何か理解できなかった

ドリトル


# ドリトルとは

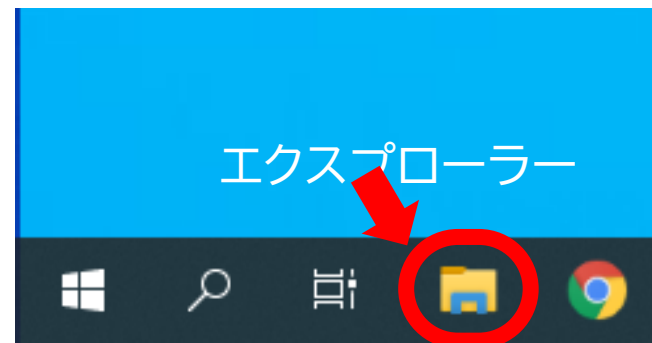
- 教育用プログラミング言語
- 日本語で記述できる
- 半角・全角どちらでも動く

# 宝物拾いゲームの作成

プログラミングとは何かを学ぼう

# ドリトルの起動

1. [エクスプローラー]をクリック
2. [自分のドライブ]を開く
3. [ dolittle.jar]をダブルクリック



# プログラミングとは何かを学ぶ

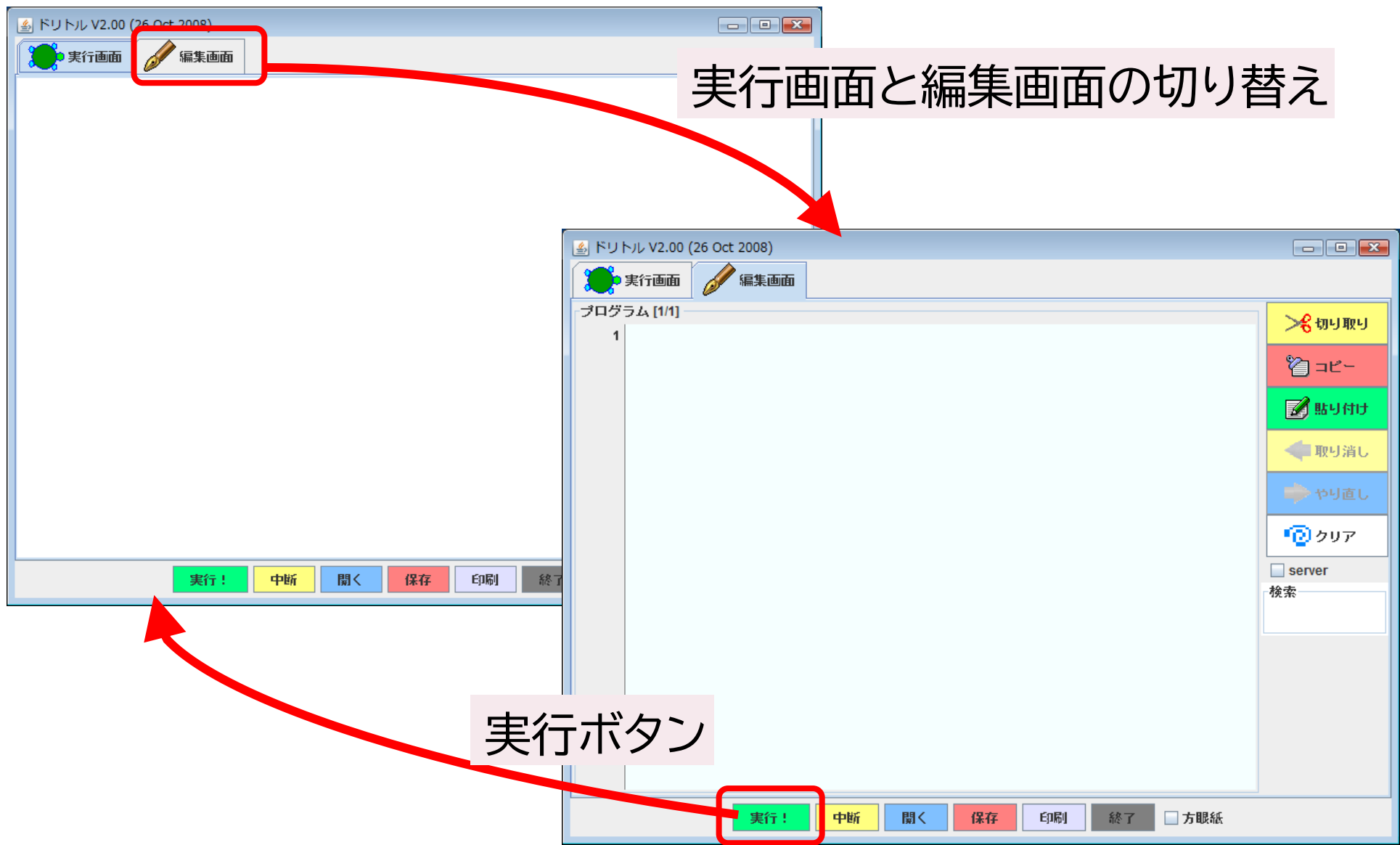
- 今日全員で同じペースが基本
  - 教え合いOK
  - 隣の人の様子も見てね
- 改造も可
  - どうなるのかな？→確かめてみる
  - 次に進む前に元に戻すこと
- エラーが出たら「ヤッタ！」と思うこと
  - エラーを解決する中でわかることがたくさんある

# 実習を通して学ぶこと

- プログラミングとは
- プログラミングの注意点
  
- 常に考えながら実習を進める



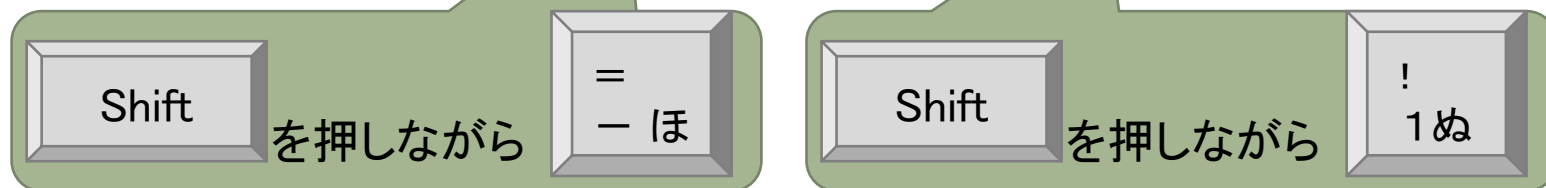
# ドリトルの画面



# 1. キャラクター登場-1

- 編集画面1行目に入力

かめた=タートル！作る。

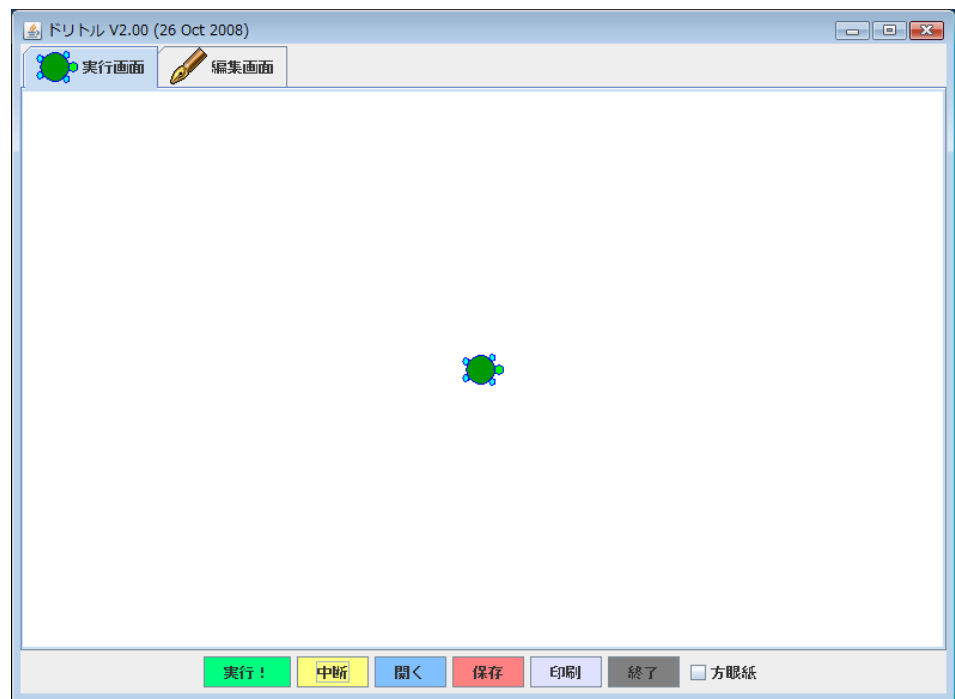


- 実行画面にして実行ボタンをクリック

# 1. キャラクター登場-2

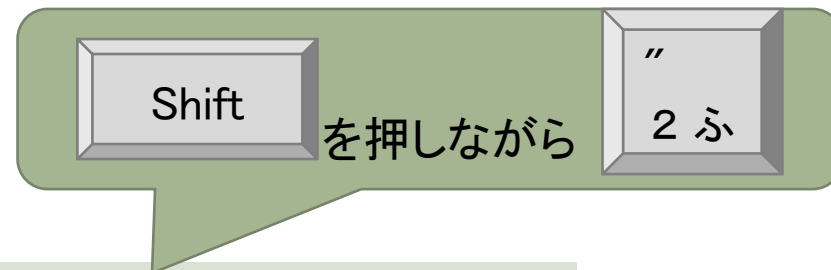
- キャラクター「かめた」が登場

ここまでのプログラム  
かめた=タートル！作る。



## 2. 操作ボタンをつくる-1

- 編集画面2行目に入力



左ボタン=ボタン!“左” 作る。

- 実行画面にして実行ボタンをクリック

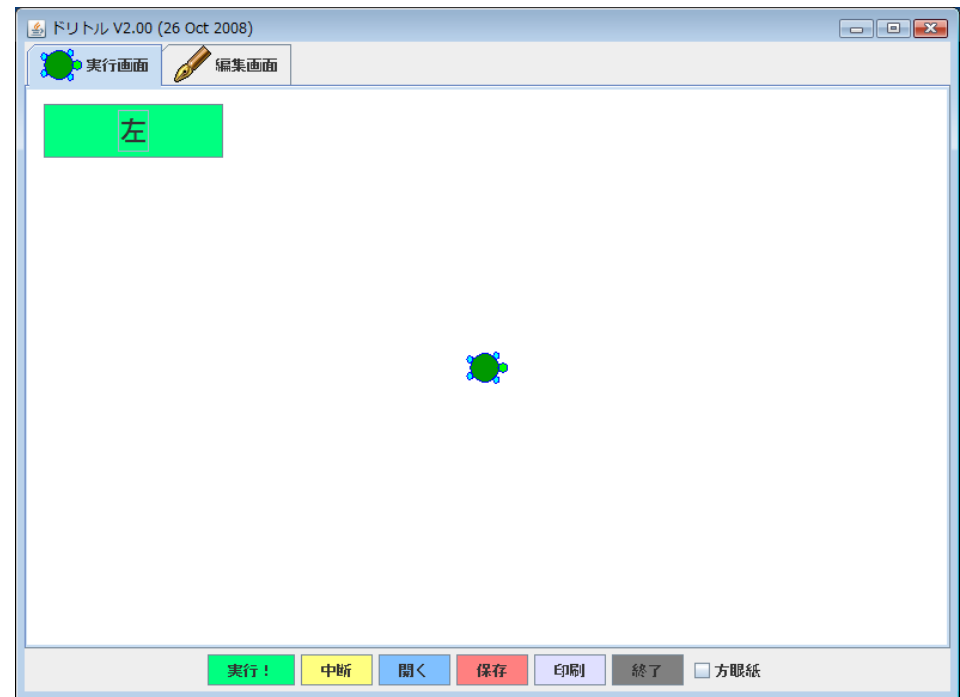
ここまでのプログラム  
かめた=タートル！作る。

## 2. 操作ボタンをつくる-2

- 操作ボタン[左]が表示される
- ボタンをクリックするとどうなる？

- なぜだろう

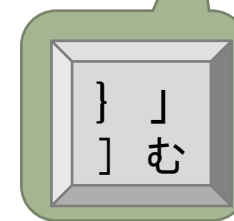
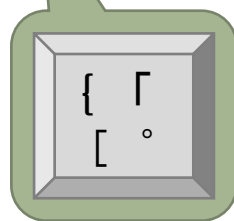
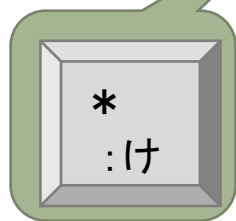
ここまでのプログラム  
かめた=タートル！作る。  
左ボタン=ボタン！“左”作る。



# 3. 動作を定義-1

- 編集画面3行目に入力

左ボタン：動作＝「かめた！30 左回り」。



- 実行画面にして実行ボタンをクリック

ここまでのプログラム  
かめた＝タートル！作る。  
左ボタン＝ボタン！“左”作る。

# 3. 動作を定義-2

- 表示は変わらない
- ボタンをクリックするとどうなる？

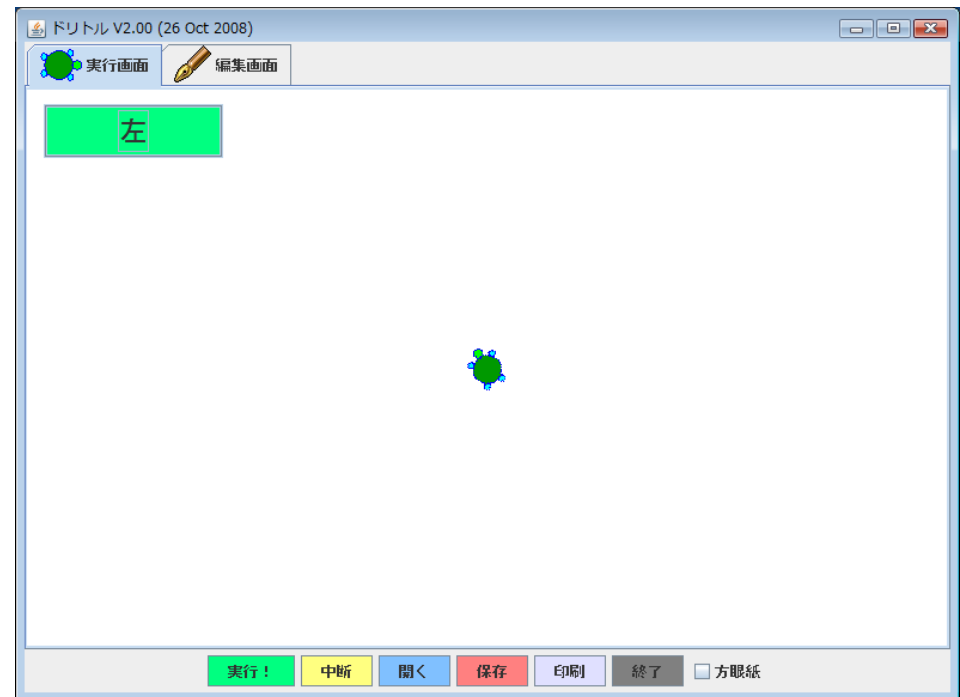
- 右回転のボタンを作ってみよう

ここまでのプログラム

かめた=タートル！作る。

左ボタン=ボタン！“左”作る。

左ボタン:動作=「かめた！30 左回り」。



# 3. 動作の定義-3

- 右ボタンと右ボタンの定義のため  
編集画面4・5行目に入力

(コピーして使うと便利)

右ボタン=ボタン!“右” 作る。  
右ボタン:動作=「かめた!30 右回り」。

- 実行画面にして実行ボタンをクリック

ここまでのプログラム  
かめた=タートル!作る。  
左ボタン=ボタン!“左”作る。  
左ボタン:動作=「かめた!30 左回り」。



# 3. 動作の定義-4

- 左右に回転させることができる

## ここまでのプログラム

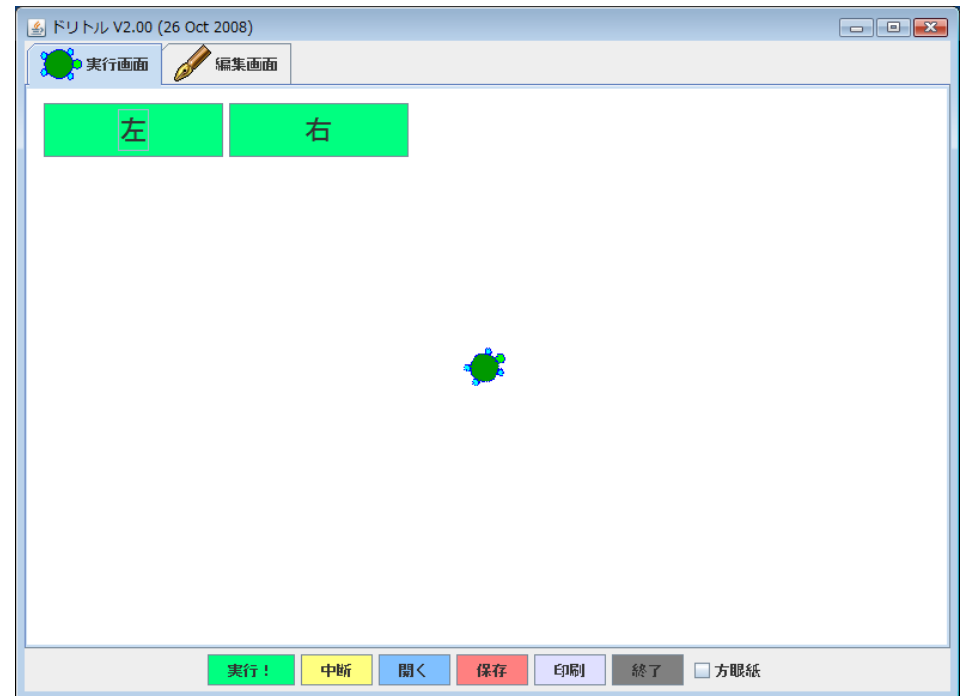
かめた=タートル！作る。

左ボタン=ボタン！“左”作る。

左ボタン:動作=「かめた！30 左回り」。

右ボタン=ボタン！“右”作る。

右ボタン:動作=「かめた！30 右回り」。



## 4. 進め！かめた-1

- かめたを動かすために、編集画面6・7行目に入力

時計＝タイマー！作る。  
時計！「かめた！10 歩く」実行。

- 実行画面にして実行ボタンをクリック

### ここまでのプログラム

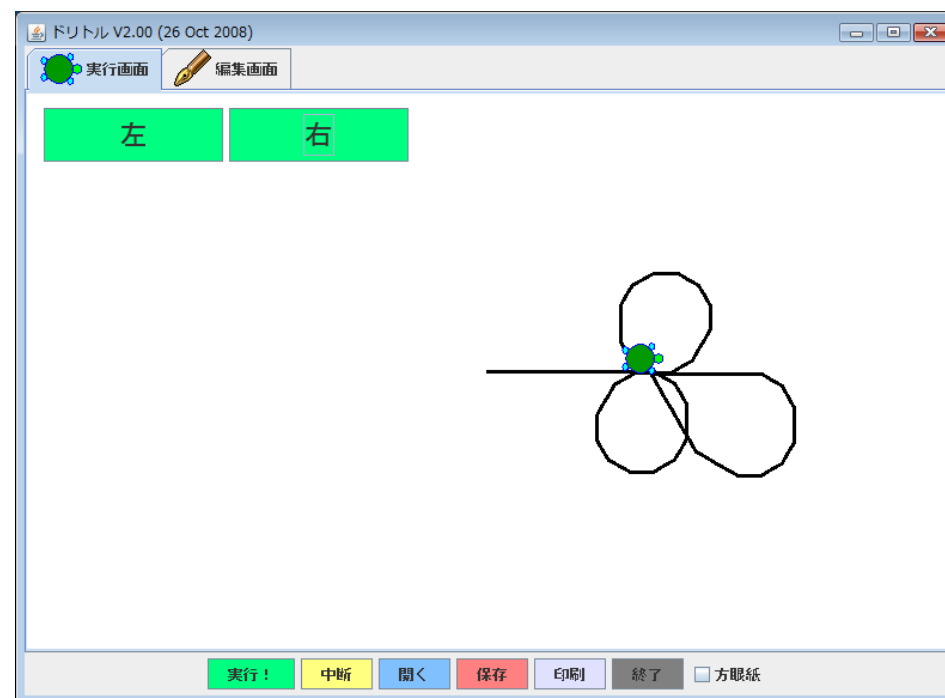
かめた＝タイトル！作る。  
左ボタン＝ボタン！“左”作る。  
左ボタン：動作＝「かめた！30 左回り」。  
右ボタン＝ボタン！“右”作る。  
右ボタン：動作＝「かめた！30 右回り」。

# 4. 進め！かめた-2

- かめたが前進する
- [左]、[右]ボタンでコントロールできる

## ここまでのプログラム

かめた=タートル！作る。  
左ボタン=ボタン！“左”作る。  
左ボタン:動作=「かめた！30 左回り」。  
右ボタン=ボタン！“右”作る。  
右ボタン:動作=「かめた！30 右回り」。  
時計=タイマー！作る。  
時計！「かめた！10 歩く」実行。

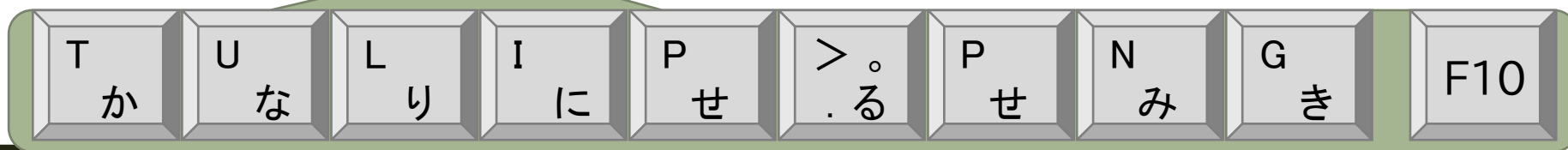


# 5. 宝拾いゲーム化-1

- 編集画面8行目に入力(1行につなげて入力)

タイトル！作る “tulip.png” 変身する  
ペンなし 100 100 位置。

タイトル！作る “tulip.png” 変身する ペンなし 100 100 位置。



- 実行画面にして実行ボタンをクリック

## ここまでのプログラム

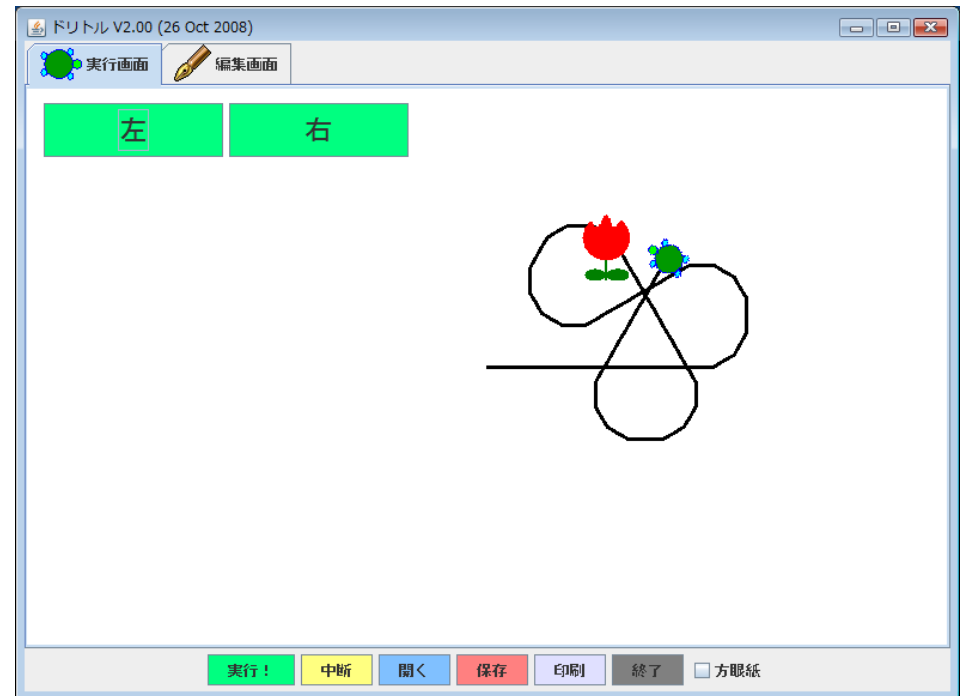
かめた=タイトル！作る。  
左ボタン=ボタン！“左”作る。  
左ボタン:動作=「かめた！30 左回り」。  
右ボタン=ボタン！“右”作る。  
右ボタン:動作=「かめた！30 右回り」。  
時計=タイマー！作る。  
時計！「かめた！10 歩く」実行。

# 5. 宝拾いゲーム化-2

- チューリップが表示される
- かめたを操作してチューリップへ重ねる
- どうなる？
- チューリップを左下に追加

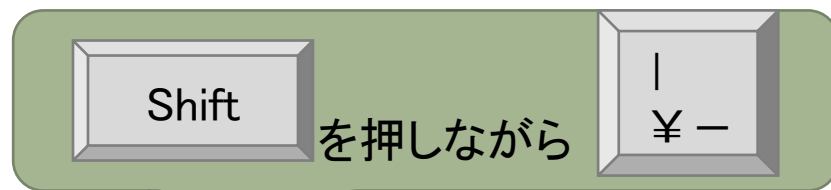
## ここまでのプログラム

かめた=タートル！作る。  
左ボタン=ボタン！“左”作る。  
左ボタン:動作=「かめた！30 左回り」。  
右ボタン=ボタン！“右”作る。  
右ボタン:動作=「かめた！30 右回り」。  
時計=タイマー！作る。  
時計！「かめた！10 歩く」実行。  
タートル！作る “tulip.png” 変身する ペンなし 100 100 位置。  
タートル！作る “tulip.png” 変身する ペンなし 100 -100 位置。  
タートル！作る “tulip.png” 変身する ペンなし -100 100 位置。



# 5. 宝拾いゲーム化-3

- 当たったら消えるように編集画面11行目に入力



かめた：衝突＝「|相手| 相手！消える」。

- 実行画面にして実行ボタンをクリック

## ここまでのプログラム

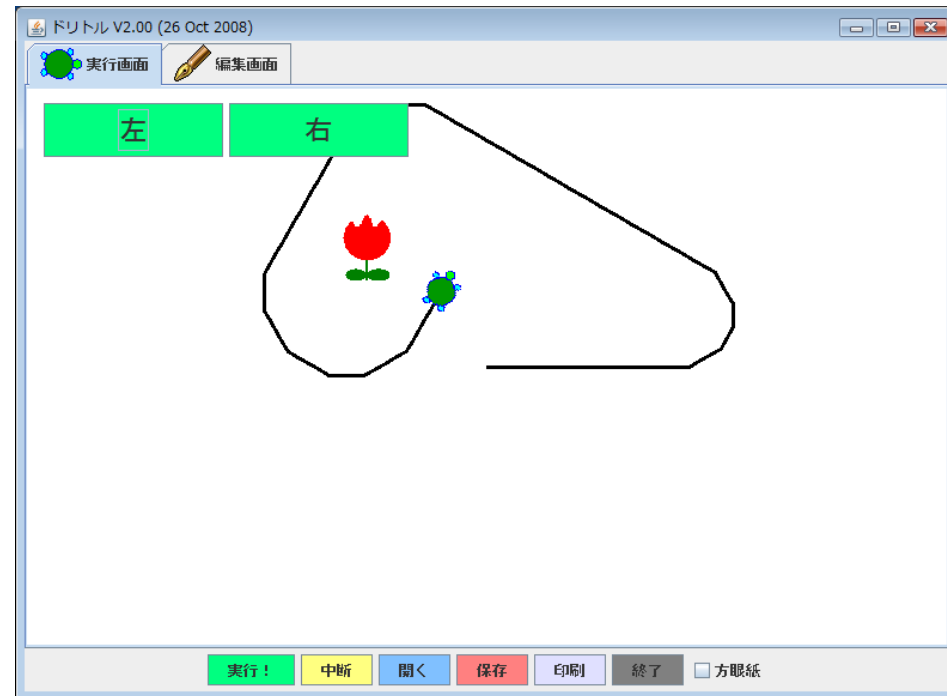
かめた＝タートル！作る。  
左ボタン＝ボタン！“左”作る。  
左ボタン：動作＝「かめた！30 左回り」。  
右ボタン＝ボタン！“右”作る。  
右ボタン：動作＝「かめた！30 右回り」。  
時計＝タイマー！作る。  
時計！「かめた！10 歩く」実行。  
タートル！作る “tulip.png” 変身する ペンなし 100 100 位置。  
タートル！作る “tulip.png” 変身する ペンなし 100 -100 位置。  
タートル！作る “tulip.png” 変身する ペンなし -100 100 位置。

# 5. 宝拾いゲーム化-4

- チューリップが3つ
- 衝突するとチューリップが消える
- 10秒で終わり
- チューリップをふやしてもよい

## ここまでのプログラム

かめた=タートル！作る。  
左ボタン=ボタン！“左”作る。  
左ボタン:動作=「かめた！30 左回り」。  
右ボタン=ボタン！“右”作る。  
右ボタン:動作=「かめた！30 右回り」。  
時計=タイマー！作る。  
時計！「かめた！10 歩く」実行。  
タートル！作る “tulip.png” 変身する ペンなし 100 100 位置。  
タートル！作る “tulip.png” 変身する ペンなし 100 -100 位置。  
タートル！作る “tulip.png” 変身する ペンなし -100 100 位置。  
かめた:衝突=「|相手| 相手！消える」。



# 5. 宝拾いゲーム化-5

- 最後のチュールリップを修正(1行につなげて入力)

タートル！作る “tulip.png” 変身する ペンなし  
(乱数(600)-300) (乱数(400)-200) 位置。

タートル！作る “tulip.png” 変身する ペンなし (乱数(600)-300) (乱数(400)-200) 位置。

- 実行画面にして実行ボタンをクリック

## ここまでのプログラム

かめた=タートル！作る。  
左ボタン=ボタン！“左”作る。  
左ボタン:動作=「かめた！30 左回り」。  
右ボタン=ボタン！“右”作る。  
右ボタン:動作=「かめた！30 右回り」。  
時計=タイマー！作る。  
時計！「かめた！10 歩く」実行。  
タートル！作る “tulip.png” 変身する ペンなし 100 100 位置。  
タートル！作る “tulip.png” 変身する ペンなし 100 -100 位置。  
タートル！作る “tulip.png” 変身する ペンなし -100 100 位置。  
かめた:衝突=「|相手| 相手！消える」。



# 5. 宝拾いゲーム化-6

- 何回か実行するとどうなる？

## ここまでのプログラム

かめた=タートル！作る。

左ボタン=ボタン！“左”作る。

左ボタン:動作=「かめた！30 左回り」。

右ボタン=ボタン！“右”作る。

右ボタン:動作=「かめた！30 右回り」。

時計=タイマー！作る。

時計！「かめた！10 歩く」実行。

タートル！作る“tulip.png”変身する ペンなし 100 100 位置。

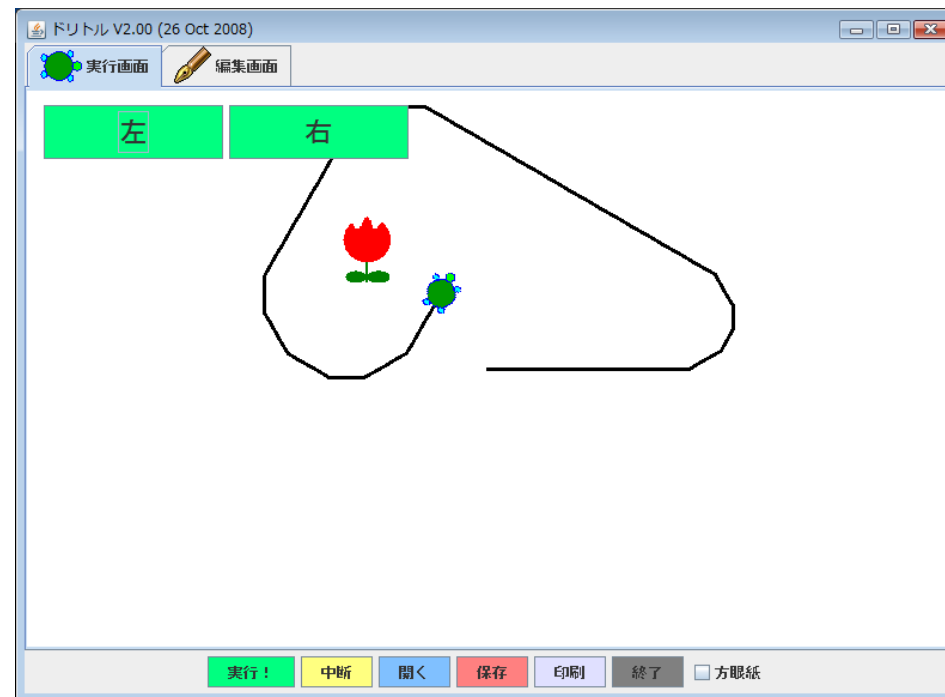
タートル！作る“tulip.png”変身する ペンなし 100 -100 位置。

タートル！作る“tulip.png”変身する ペンなし -100 100 位置。

タートル！作る“tulip.png”変身する ペンなし (乱数(600)-300)

(乱数(400)-200) 位置。

かめた:衝突=「|相手| 相手！消える」。



# 5. 宝拾いゲーム化-7

- 2行目と4行目を修正

左ボタン=ボタン！ “左” “LEFT” 作る。  
右ボタン=ボタン！ “右” “RIGHT” 作る。

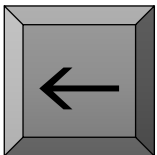
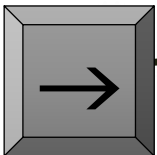
- 実行画面にして実行ボタンをクリック

## ここまでのプログラム

かめた=タイトル！作る。  
左ボタン=ボタン！“左”作る。  
左ボタン:動作=「かめた！30 左回り」。  
右ボタン=ボタン！“右”作る。  
右ボタン:動作=「かめた！30 右回り」。  
時計=タイマー！作る。  
時計！「かめた！10 歩く」実行。  
タイトル！作る “tulip.png” 変身する ペンなし 100 100 位置。  
タイトル！作る “tulip.png” 変身する ペンなし 100 -100 位置。  
タイトル！作る “tulip.png” 変身する ペンなし -100 100 位置。  
タイトル！作る “tulip.png” 変身する ペンなし (乱数(600)-300)  
(乱数(400)-200) 位置。  
かめた:衝突=「|相手| 相手！消える」。

tulip.pngを  
kuno.png  
star.png  
crab.png  
fish.png  
などに変えてみる

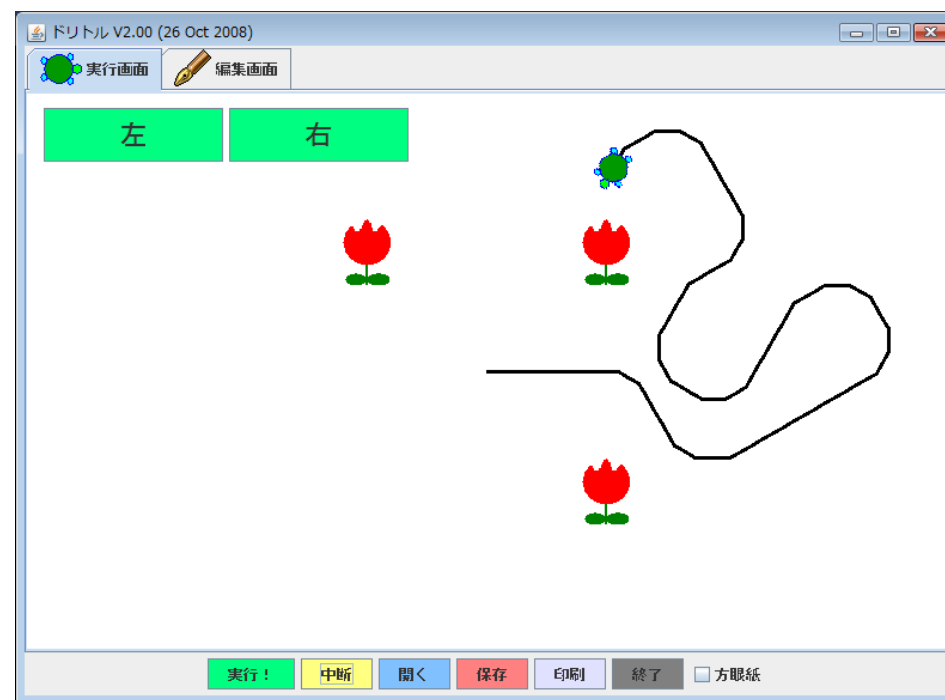
# 5. 宝拾いゲーム化-8

• キーボードの   で操作できる

tulip.pngを  
kuno.png  
star.png  
crab.png  
fish.png  
などに変えてみる

## ここまでのプログラム

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」。  
時計=タイマー！作る。  
時計！「かめた！10 歩く」実行。  
タートル！作る “tulip.png” 変身する ペンなし 100 100 位置。  
タートル！作る “tulip.png” 変身する ペンなし 100 -100 位置。  
タートル！作る “tulip.png” 変身する ペンなし -100 100 位置。  
タートル！作る “tulip.png” 変身する ペンなし (乱数(600)-300)  
(乱数(400)-200) 位置。  
かめた:衝突=「|相手| 相手！消える」。

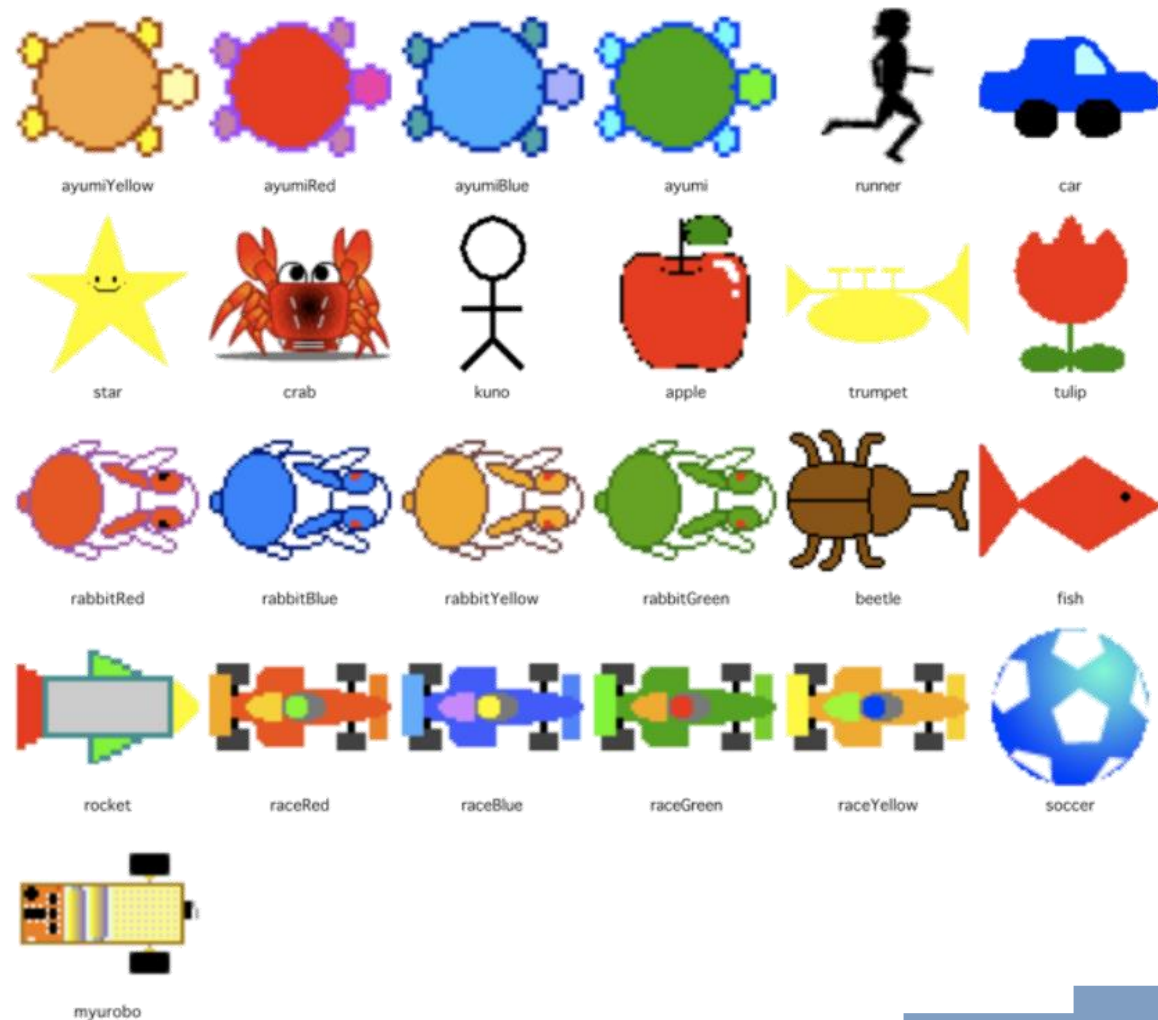


# 5. 宝拾いゲーム化-9

- tulip.pngを
- kuno.png star.png
- crab.png fish.png
- などに変えてみる

## ここまでのプログラム

かめた=タートル！作る。  
左ボタン=ボタン！“左”作る。  
左ボタン:動作=「かめた！30 左回り」。  
右ボタン=ボタン！“右”作る。  
右ボタン:動作=「かめた！30 右回り」。  
時計=タイマー！作る。  
時計！「かめた！10 歩く」実行。  
タートル！作る“tulip.png”変身する ペンなし 100 100 位置。  
タートル！作る“tulip.png”変身する ペンなし 100 -100 位置。  
タートル！作る“tulip.png”変身する ペンなし -100 100 位置。  
タートル！作る“tulip.png”変身する ペンなし (乱数(600)-300)  
(乱数(400)-200) 位置。  
かめた:衝突=「|相手| 相手！消える」。



# 実習を通して学んだこと

- プログラミングとは
- プログラミングの注意点