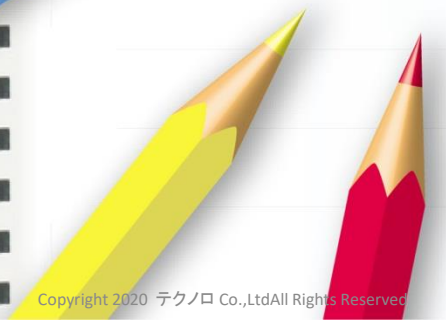
プログラミング教室のテクノロ

Pythonの道

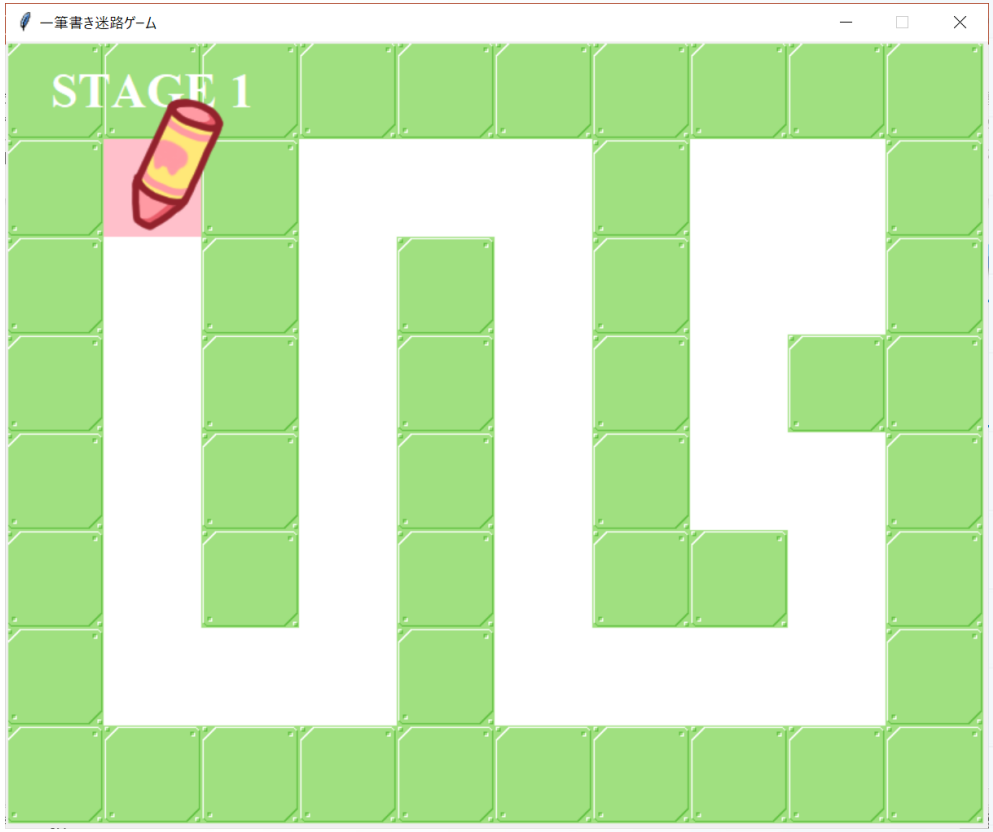


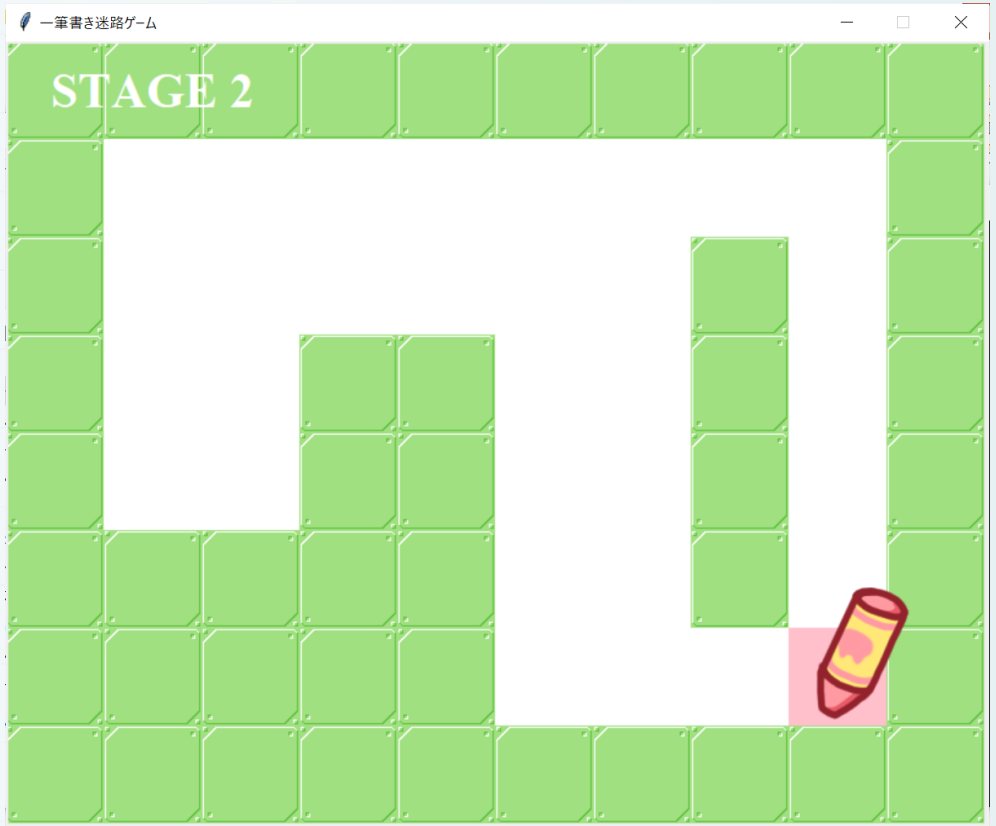
本格的なゲーム開発①(練習)

もくじ

・一筆書き迷路ゲームをつくる

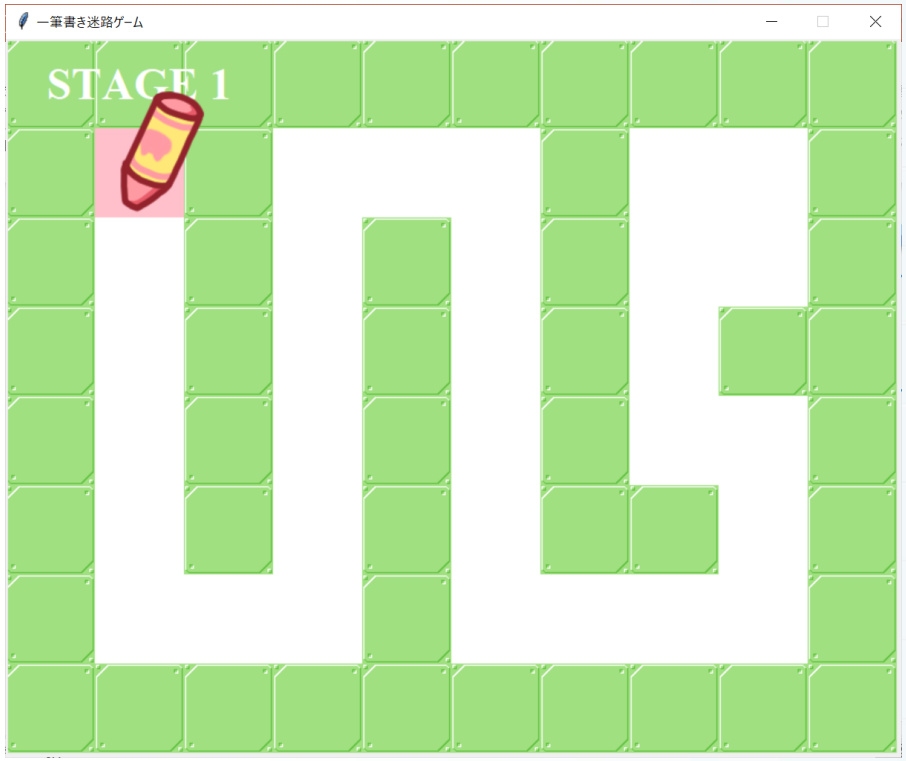
床塗りゲームを発展させたゲームを作る。

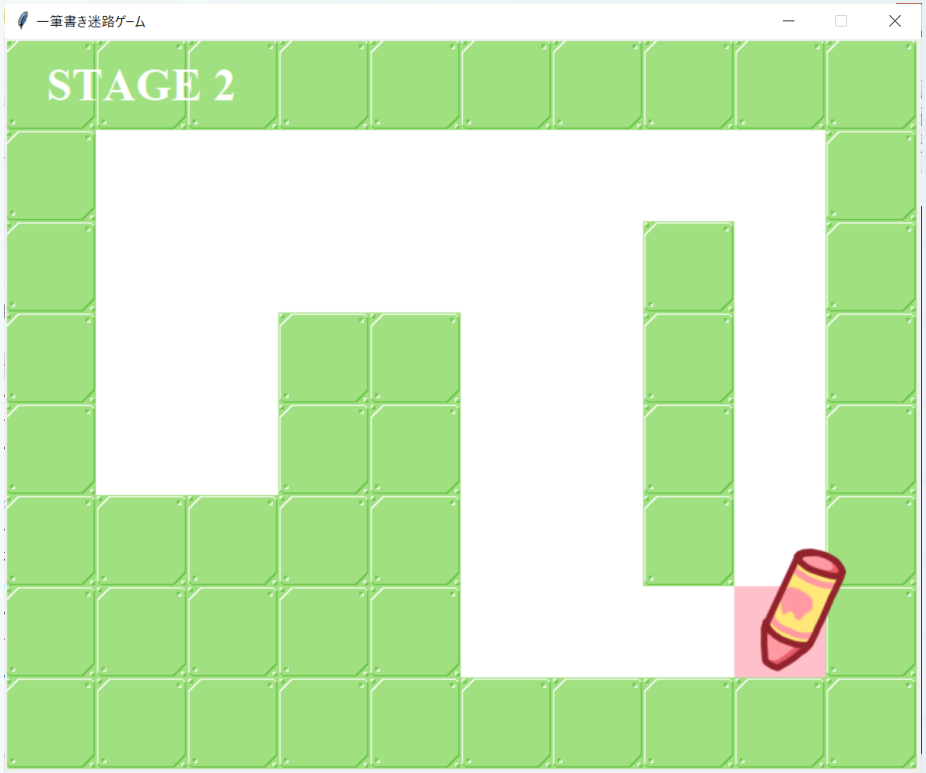




一筆書き迷路ゲームを作る

床塗りゲームを応用させて、全部で5ステージある一筆書きをするゲームを作る。





操作方法

・方向キーでクレヨンの移動

・やり直しは左の「Shiftキー」または「G」キーを押す。

一筆書き迷路ゲームを作る

コーディングにあたって

■stage_data()という関数内で5ステージ分の迷路データを定義する。stage_data()の処理はステージを管理するstageという変数の値によって、各ステージの迷路の形を二次元リストmazeにセットする。

```
def stage_data():
    global ix, iy
    global maze #リスト全体を変更する場合 global宣言が必要
    if stage == 1:
        ix = 1
        iy = 1
        maze = [# 0が床、1が塗ったところ、9が壁
        [9, 9, 9, 9, 9, 9, 9, 9, 9, 9],
        [9, 0, 9, 0, 0, 0, 9, 0, 0, 9],
        [9, 0, 9, 0, 9, 0, 9, 0, 0, 9],
        [9, 0, 9, 0, 9, 0, 9, 0, 9, 9],
        [9, 0, 9, 0, 9, 0, 9, 0, 0, 9],
        [9, 0, 9, 0, 9, 0, 9, 9, 0, 9],
        [9, 0, 0, 0, 9, 0, 0, 0, 0, 9],
        [9, 9, 9, 9, 9, 9, 9, 9, 9, 9]
        ]
```

■ゲームのメイン処理を行うgame_main()関数では、インデックスの値が1時にペンを動かし、迷路を塗りつぶしたかを判定する。インデックスの値が２の時にステージクリアの処理を行う。

一筆書き迷路ゲームを作る

使用する変数

変数名	用途
idx,tmr	ゲーム進行を管理するインデックスとタイマー
stage	ステージ番号
ix,iy	クレヨンの位置
key	押されたキーの値
maze	迷路のデータを入れるリスト

一筆書き迷路ゲームを作る

使用する関数

関数名	用途
key_down(e)	キーが押された時に働く
key_up(e)	キーが離された時に働く
stage_data()	各ステージのデータをセットする
draw_bg()	画面を描く
erase_bg()	画面を消す
move_pen()	クレヨンを動かす
count_tile()	塗っていないマスを数える
game_main()	メイン処理を行う

二次元リストのおさらい

横方向をx、縦方向をyとした場合の添え字の表し方を理解する。

m[y][x]

添え字

x →

y ↓

m[0][0]	m[0][1]	m[0][2]	m[0][3]
m[1][0]	m[1][1]	m[1][2]	m[1][3]
m[2][0]	m[2][1]	m[2][2]	m[2][3]

例えば右下角のm[2][3]に10を代入する場合、

m[2][3] ＝ 10 と記載する。

1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	0	0	1
1	0	1	1	0	0	1	0	0	1
1	0	0	1	0	0	0	0	0	1
1	0	0	1	1	1	1	1	0	1
1	0	0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1	1

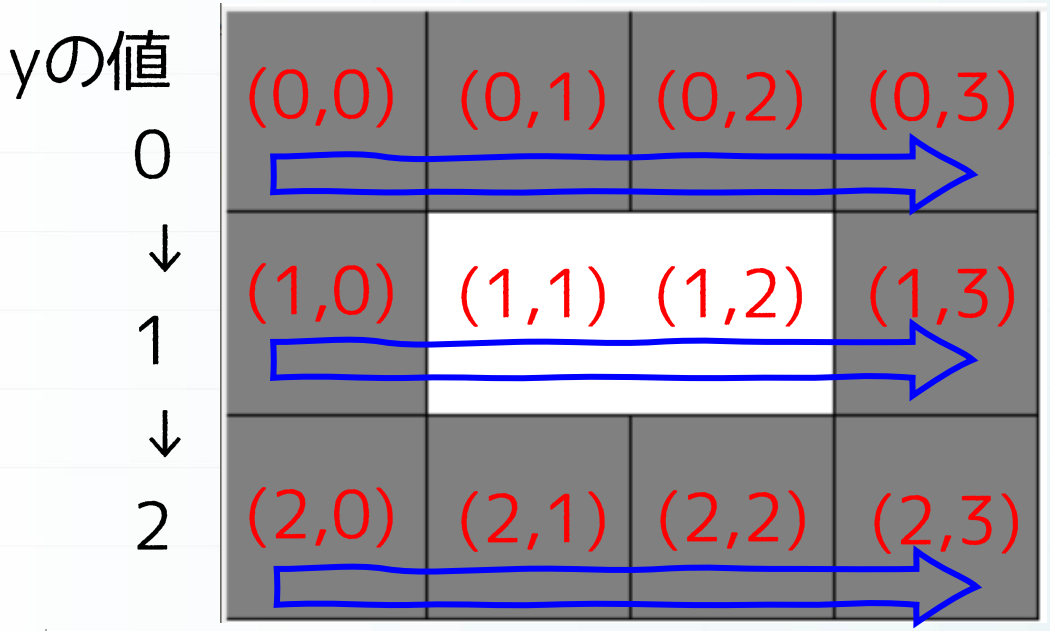
迷路データのおさらい

for文を使って、同じことをする。壁を囲むには二重ループの
for文を使用する。

```
for 変数1 in 変数1の範囲:
    for 変数2 in 変数2の範囲:
        処理のブロック
```

```
for y in range(3):
    for x in range(4):
```

xの値 0 ⇒ 1 ⇒ 2 ⇒ 3



(y座標,x座標)

一筆書き迷路ゲームのコード

```
import tkinter
import tkinter.messagebox ・・・tkinter.messegeboxモジュールの呼出し

idx = 0
tmr = 0
stage = 1
ix = 0
iy = 0
key = 0
```

変数名	用途
idx,tmr	ゲーム進行を管理するインデックスとタイマー
stage	ステージ番号
ix,iy	クレヨンの位置
key	押されたキーの値
maze	迷路のデータを入れるリスト

```
def key_down(e):  ・・・ key_down関数を呼び出した時に実行する関数を定義
    global key
    key = e.keysym  ・・・ 押されたキーのコードを変数「key」に代入

def key_up(e): #キーを離した時に実行する関数の定義
    global key
    key = 0

maze = [[],[],[],[],[],[],[],[]] ・・・ 変数 mazeに空のリストを8つ作成

def stage_data():  ・・・各ステージのデータをセットする関数
    global ix, iy
    global maze #リスト全体を変更する場合 global宣言が必要
    if stage == 1:
        ix = 1
        iy = 1      ・・・クレヨンの最初の座標
```

次のページに続く

一筆書き迷路ゲームのコード

```
    maze = [  # 0が床、1が塗ったところ、9が壁
    [9, 9, 9, 9, 9, 9, 9, 9, 9, 9],
    [9, 0, 9, 0, 0, 0, 9, 0, 0, 9],
    [9, 0, 9, 0, 9, 0, 9, 0, 0, 9],
    [9, 0, 9, 0, 9, 0, 9, 0, 9, 9],
    [9, 0, 9, 0, 9, 0, 9, 0, 0, 9],
    [9, 0, 9, 0, 9, 0, 9, 9, 0, 9],
    [9, 0, 0, 0, 9, 0, 0, 0, 0, 9],
    [9, 9, 9, 9, 9, 9, 9, 9, 9, 9]
    ]
if stage == 2:
    ix = 8
    iy = 6
    maze = [  # 0が床、1が塗ったところ、9が壁
    [9, 9, 9, 9, 9, 9, 9, 9, 9, 9],
    [9, 0, 0, 0, 0, 0, 0, 0, 0, 9],
    [9, 0, 0, 0, 0, 0, 0, 9, 0, 9],
    [9, 0, 0, 9, 9, 0, 0, 9, 0, 9],
    [9, 0, 0, 9, 9, 0, 0, 9, 0, 9],
    [9, 9, 9, 9, 9, 0, 0, 9, 0, 9],
    [9, 9, 9, 9, 9, 0, 0, 0, 0, 9],
    [9, 9, 9, 9, 9, 9, 9, 9, 9, 9]
    ]
if stage == 3:
    ix = 3
    iy = 3
```

二次元リスト[

・・・・リストで迷路を定義

[0][0] , [0][1] , [0][2] , [0][3] , ・・・
[1][0] , [1][1] , [1][2] , [1][3] , ・・・
[2][0] , [2][1] , [2][2] , [2][3] , ・・・
・
・
・
]

次のページに続く

一筆書き迷路ゲームのコード

```
        maze = [  # 0が床、1が塗ったところ、9が壁
        [9, 9, 9, 9, 9, 9, 9, 9, 9, 9],
        [9, 9, 9, 0, 0, 0, 0, 9, 9, 9],
        [9, 9, 0, 0, 0, 0, 0, 0, 9, 9],
        [9, 0, 0, 0, 0, 0, 0, 0, 0, 9],
        [9, 0, 9, 0, 0, 0, 0, 0, 0, 9],
        [9, 0, 0, 0, 0, 0, 0, 0, 9, 9],
        [9, 9, 0, 0, 0, 0, 0, 9, 9, 9],
        [9, 9, 9, 9, 9, 9, 9, 9, 9, 9]
        ]
    if stage == 4:
        ix = 4
        iy = 3
        maze = [  # 0が床、1が塗ったところ、9が壁
        [9, 9, 9, 9, 9, 9, 9, 9, 9, 9],
        [9, 0, 0, 0, 0, 0, 0, 0, 0, 9],
        [9, 0, 0, 0, 9, 0, 0, 0, 0, 9],
        [9, 0, 0, 0, 0, 0, 0, 0, 0, 9],
        [9, 0, 0, 9, 0, 0, 0, 9, 0, 9],
        [9, 0, 0, 0, 0, 0, 0, 9, 0, 9],
        [9, 0, 0, 0, 0, 0, 0, 0, 0, 9],
        [9, 9, 9, 9, 9, 9, 9, 9, 9, 9]
        ]
    if stage == 5:
        ix = 1
        iy = 6
```

次のページに続く

一筆書き迷路ゲームのコード

```
        maze = [# 0が床、1が塗ったところ、9が壁
        [9, 9, 9, 9, 9, 9, 9, 9, 9, 9],
        [9, 0, 0, 0, 0, 0, 0, 0, 0, 9],
        [9, 0, 9, 0, 0, 0, 0, 0, 0, 9],
        [9, 0, 0, 0, 0, 0, 9, 9, 0, 9],
        [9, 0, 0, 0, 0, 9, 9, 9, 0, 9],
        [9, 0, 0, 9, 0, 0, 0, 0, 0, 9],
        [9, 0, 0, 0, 0, 0, 0, 0, 0, 9],
        [9, 9, 9, 9, 9, 9, 9, 9, 9, 9]
        ]
    maze[iy][ix] = 1 ・・・クレヨンが塗ったところを1に変更していく

def draw_bg():  ・・・画面を描く関数を定義
    for y in range(8):
        for x in range(10):
            gx = 80*x ・・・マスを管理する変数
            gy = 80*y ・・・マスを管理する変数
            if maze[y][x] == 0:・・・もし迷路のブロックが床だった場合、白いブロックを配置
                cvs.create_rectangle(gx, gy, gx+80, gy+80, fill="white", width=0, tag="BG")
            if maze[y][x] == 9:・・・もし迷路のブロックが壁だった場合、ウォール画像を配置
                cvs.create_image(gx+40, gy+40, image=wall, tag="BG")
    cvs.create_text(120, 40, text="STAGE "+str(stage), fill="white", font=("Times New Roman",
30, "bold"), tag="BG")  ・・・STAGE〇〇と表示
    gx = 80*ix   ・・・クレヨンの座標
    gy = 80*iy                                          ・・・クレヨンが通った後をピンク色にする処理
    cvs.create_rectangle(gx, gy, gx+80, gy+80, fill="pink", width=0, tag="BG")
    cvs.create_image(gx+60, gy+20, image=pen, tag="PEN")
```

次のページに続く

一筆書き迷路ゲームのコード

```
def erase_bg():  ・・・画面を消す関数を定義
    cvs.delete("BG")  ・・・白とピンクのブロックを削除
    cvs.delete("PEN") ・・・ペンのブロックを削除

def move_pen():  ・・・クレヨンを動かす関数を定義
    global idx, tmr, ix, iy, key
    bx = ix
    by = iy
    if key == "Left" and maze[iy][ix-1] == 0:
        ix = ix-1
    if key == "Right" and maze[iy][ix+1] == 0:
        ix = ix+1
    if key == "Up" and maze[iy-1][ix] == 0:
        iy = iy-1
    if key == "Down" and maze[iy+1][ix] == 0:
        iy = iy+1
    if ix != bx or iy != by:
        maze[iy][ix] = 2
        gx = 80*ix
        gy = 80*iy
        cvs.create_rectangle(gx, gy, gx+80, gy+80, fill="pink", width=0, tag="BG")
        cvs.delete("PEN")
        cvs.create_image(gx+60, gy+20, image=pen, tag="PEN")

    if key == "g" or key == "G" or key == "Shift_L":
        key = 0
        ret = tkinter.messagebox.askyesno("ギブアップ", "やり直しますか？")
```

次のページに続く

一筆書き迷路ゲームのコード

```
        if ret == True:
            stage_data()
            erase_bg()
            draw_bg()

def count_tile():
    cnt = 0
    for y in range(8):
        for x in range(10):
            if maze[y][x] == 0:
                cnt = cnt + 1
    return cnt

def game_main():
    global idx, tmr, stage
    if idx == 0: #初期化
        stage_data()
        draw_bg()
        idx = 1
    if idx == 1: #ペンの移動とクリア判定
        move_pen()
        if count_tile() == 0:
            txt = "STAGE CLEAR"
            if stage == 5:
                txt = "ALL STAGE CLEAR!"
            cvs.create_text(400, 320, text=txt, fill="white", font=("Times New Roman", 40,
"bold"), tag="BG")
            idx = 2
```

次のページに続く

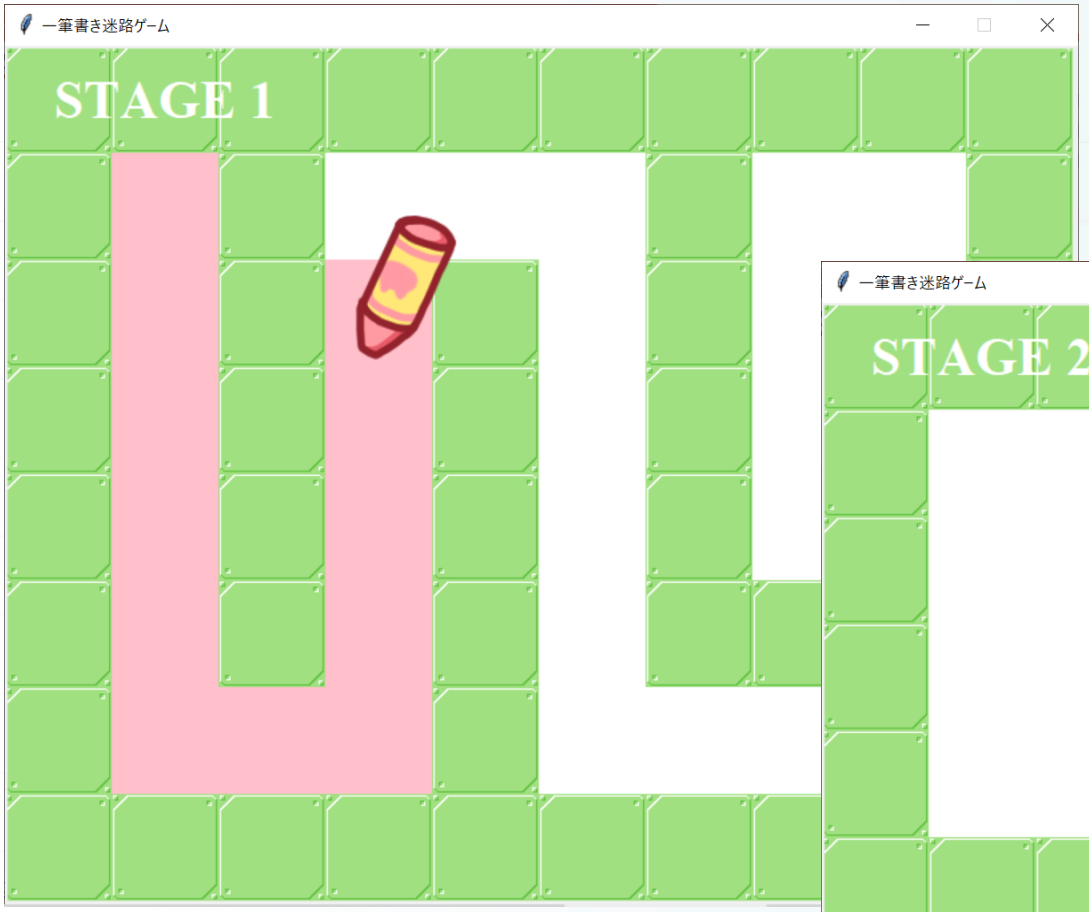
一筆書き迷路ゲームのコード

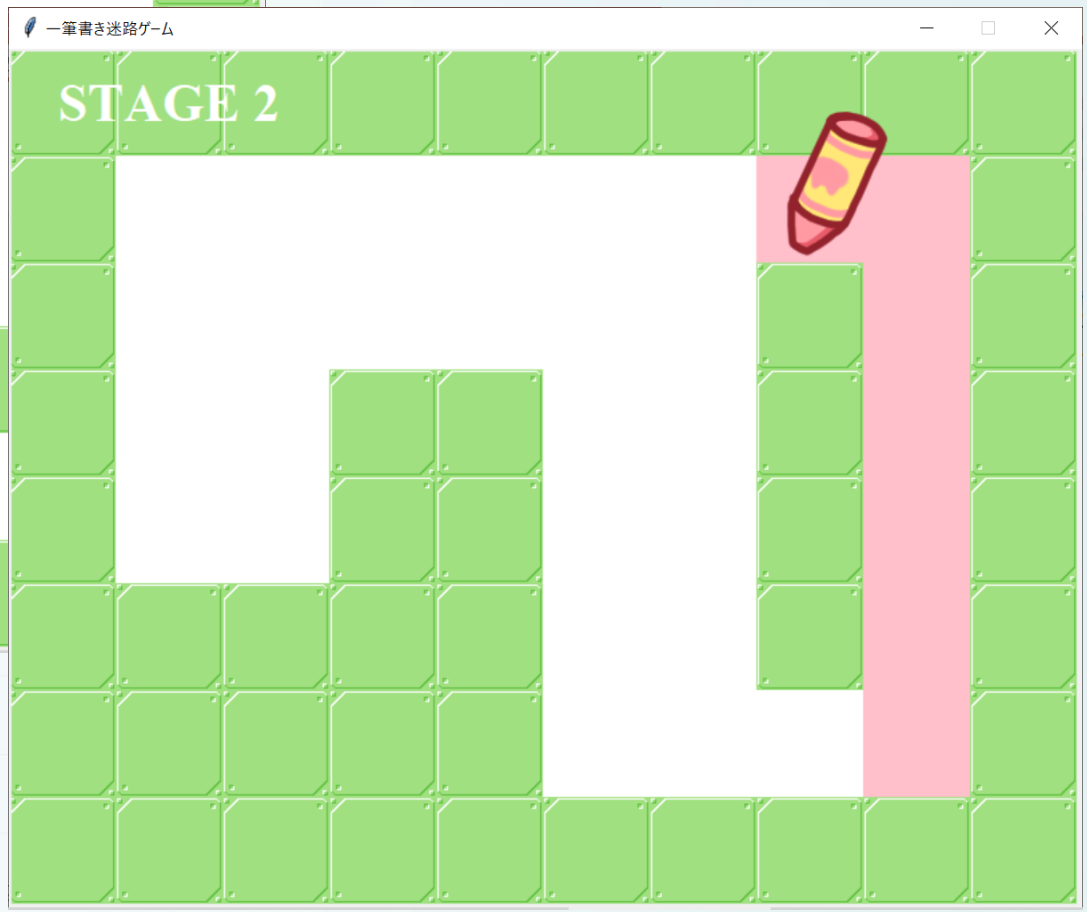
```
            tmr = 0
    if idx == 2: #ステージクリア
        tmr = tmr + 1
        if tmr == 30:
            if stage < 5:
                stage = stage + 1
                stage_data()
                erase_bg()
                draw_bg()
                idx = 1
    root.after(200, game_main)

root = tkinter.Tk()
root.title("一筆書き迷路ゲーム")
root.resizable(False, False)
root.bind("<KeyPress>", key_down)
root.bind("<KeyRelease>", key_up)
cvs = tkinter.Canvas(root, width=800, height=640)
cvs.pack()
pen = tkinter.PhotoImage(file="pen.png")
wall = tkinter.PhotoImage(file="wall.png")
game_main()
root.mainloop()
```

一筆書き迷路ゲームのコード

「Run Module」またはF5キーを押してプログラムを実行する。





復習＆チャレンジ

ここまで習ったことをScratchでもできるかチャレンジしてみよう。
その過程でScratchでできること、Pythonでないとできないことを整理してみよう。

メモ

プログラミング教室のテクノロ

なまえ：