

Pygame 미로찾기 게임 만들기

1. 모듈 import 하기

```
import pygame #pygame 모듈  
import random #random 모듈
```

2. pygame 초기화 하기

```
pygame.init() #pygame 모듈 사용 전 반드시 초기화
```

3. FPS(Frames Per Second) 변수 값 설정

```
FPS = 60
```

```
FramePerSec = pygame.time.Clock()
```

4. 색상 변수 설정

WHITE = (255,255,255)

BLACK = (0,0,0)

BLUE = (0,0,255)

RED = (255,0,0)

5. 폰트 설정

font = pygame.font.SysFont(None, 50)

6. 화면 설정

GameDisplay = pygame.display.set_mode((510,510))

pygame.display.set_caption("Pygame Example")

7. 게임 초기값 설정

pos = [0, 0] #시작 맵 좌표

start = 18 #시작 화면 좌표

distance = 25 #맵 한 칸 당 거리

run = True #게임 실행 변수

finish = False #출구 찾기 변수

8. 게임 내 맵 블록 생성

```
block = []  
for i in range(20):  
    L = []  
    for j in range(20):  
        L.append(random.randrange(0,2))  
    block.append(L)
```

```
block[0][0] = 0
```

```
block[-1][-1] = 0
```

#사용자 주변 블록 없애기 함수

```
def destroy_block(pos, block) :
```

```
    dirs = [[0,1], [1,0], [0,-1], [-1,0]]
```

```
    for dir in dirs :
```

```
        try:
```

```
            x = pos[0]+dir[0]
```

```
            y = pos[1]+dir[1]
```

```
            if x != -1 and y != -1 : #블록 없앨 때 예외처리
```

```
                block[x][y] = 0
```

```
        except:
```

```
            pass
```

9. 게임 루프 만들기

```
while run : #게임 실행 중
    #이벤트 처리
    for event in pygame.event.get() :
        #화면 오른쪽 상단 닫기 창을 누를 때
        if event.type == pygame.QUIT :
            run = false
```

#키보드 눌렀을 때

```
elif event.type == pygame.KEYDOWN :  
    if event.key == pygame.K_LEFT:  
        if pos[1]-1 > -1 and block[pos[0]][pos[1]-1] != 1 :  
            pos = [pos[0], pos[1]-1]  
    elif event.key == pygame.K_RIGHT :  
        if pos[1]+1 < 20 and block[pos[0]][pos[1]+1] != 1 :  
            pos = [pos[0], pos[1]+1]  
    elif event.key == pygame.K_UP :  
        if pos[0]-1 > -1 and block[pos[0]-1][pos[1]] != 1 :  
            pos = [pos[0]-1, pos[1]]  
    elif event.key == pygame.K_DOWN :  
        if pos[0]+1 < 20 and block[pos[0]+1][pos[1]] != 1 :  
            pos = [pos[0]+1, pos[1]]
```

10. 게임 루프 만들기(이어서)

GameDisplay.fill(WHITE) #화면 전체를 하얗게 초기화

#이차원 맵 그리기

for i in range(5, 510, 25) :

 #가로선 그리기

 pygame.draw.line(GameDisplay, BLACK, [5,i], [505,i], 2)

 #세로선 그리기

 pygame.draw.line(GameDisplay, BLACK, [i,5], [i,505], 2)

#플레이어 그리기

pygame.draw.circle(GameDisplay, BLUE, [start+pos[1]*distance,
start+pos[0]*distance], 10)

#블록 그리기

```
for i in range(20):
```

```
    for j in range(20):
```

```
        if block[i][j] == 1 :
```

```
            pygame.draw.circle(GameDisplay, BLACK,  
                                [start+j*distance, start+i*distance], 10)
```

#출구 찾았을 경우 게임 진행 종료

```
if finish :
```

```
    pygame.draw.rect(GameDisplay, WHITE, [100, 100, 290, 40])
```

```
    game_text = font.render("Congratulations!", True, RED)
```

```
    GameDisplay.blit(game_text, [100,100])
```

11. 화면 업데이트

```
pygame.display.update()
```

```
FramePerSec.tick(FPS)
```

12. 게임 종료 처리

```
#게임 종료
```

```
pygame.quit()
```

추가 구현 문제.. 뒷장

※ 사용자 주변의 블록을 없앨 때 d 키를 누르면 바로 주변 블록이 없어집니다. 조금 더 게임처럼 보이게 하기 위해 폭탄을 설치하여 2.5초 후에 폭탄이 터지면서 주변 블록을 없애게 하고 싶습니다. 폭탄이 설치되고 2.5초 후에 터지기 전까지는 폭탄이 커졌다 작아졌다 하도록 해봅시다. 일종의 애니메이션 효과를 넣는거죠. 한 번 구현해 봅시다.

1. 2.5초 후에 폭탄이 터져야 합니다. time 모듈을 사용
2. d 키를 눌렀을 때 폭탄을 설치합니다. 폭탄 설치 함수 만들기
3. 폭탄의 설치 시간과 설치 좌표가 필요합니다. 폭탄 설치 시간, 좌표 변수 만들기
4. 폭탄의 터지기 전까지 커졌다 작아졌다 하는 효과를 넣기 위한 애니메이션 타임 변수가 필요합니다. 애니메이션 타임 변수 만들기
5. 폭탄이 커졌다 작아졌다 하는 시간은 0.5초 정도로 설정합니다.
6. 폭탄이 터질 땐 폭탄이 크게 부풀어오르도록 해봅시다.

정답 코드

#타임 모듈 사용

```
import time
```

#폭탄 관련 변수 만들기

```
anim_time = 0 #폭탄 애니메이션 시간
```

```
bomb_size = 10 #폭탄 크기
```

```
bomb_put_time = 0 #폭탄 설치 시간
```

```
bomb_pos = [-1,-1] #폭탄 좌표 초기화
```

뒷장 계속..

#폭탄 설치 함수 만들기

```
def put_bomb(pos) :  
    global bomb_pos  
    global bomb_put_time  
    if bomb_pos[0] == -1 :  
        bomb_pos = pos  
        bomb_put_time = time.time()  
        anim_time = bomb_put_time
```

#d 키를 눌렀을 때 폭탄 설치 함수 호출

```
elif event.key == pygame.K_d :  
    #destroy_block(pos, block); //기존 코드 주석 처리  
    put_bomb(pos)
```

뒷장 계속...

#폭탄 그리기

```
if finish == False and bomb_pos[0] > -1 : #게임이 진행 중이고 폭탄이 설치된 경우
    if cur_time - bomb_put_time > 2.5 : #폭탄이 설치되고 2초가 지나면 주변 블록 제거
        destroy_block(bomb_pos, block)
    elif cur_time - bomb_put_time > 2 : #폭탄이 터지기 전 부풀어오르는 효과
        pygame.draw.circle(GameDisplay, RED, [start+bomb_pos[1]*distance,
                                                start+bomb_pos[0]*distance], bomb_size*3)
    else :
        if cur_time - anim_time > 0.5 : #0.5초 마다 폭탄의 크기를 조절
            if bomb_size == 10 :
                bomb_size = 11
            else :
                bomb_size = 10
            anim_time = time.time()
        pygame.draw.circle(GameDisplay, RED, [start+bomb_pos[1]*distance,
                                                start+bomb_pos[0]*distance], bomb_size)
```

뒷장 계속...

```
if pos[0] == 19 and pos[1] == 19 :  
    finish = True
```

```
cur_time = time.time() #폭탄 애니메이션 효과를 위한 시간 저장
```

```
#화면 초기화
```

```
GameDisplay.fill(WHITE)
```

뒷장 계속...

#블록 제거 함수에 폭탄 변수 초기화 추가

```
def destroy_block(pos, block) :  
    global bomb_pos  
    dirs = [[0,1], [1,0], [0,-1], [-1,0]]  
    for dir in dirs :  
        try:  
            x = pos[0]+dir[0]  
            y = pos[1]+dir[1]  
            if x != -1 and y != -1 :  
                block[x][y] = 0  
                bomb_pos = [-1,-1]  
        except:  
            pass
```

추가 구현 문제 뒷장..

※ 폭탄을 설치하고 터질 때 플레이어가 폭탄 범위 안에 있으면 게임이 종료되도록 구현해 봅시다.

1. 폭탄이 터져서 주변 블록을 제거할 때 플레이어 위치도 확인
2. 플레이어가 범위 안에 있으면 게임 종료 처리
3. 게임 종료 시 메시지를 다르게 출력

※ 폭탄을 최대 5회까지만 설치가능 하도록 구현해 봅시다.

1. 폭탄 설치 횟수를 저장할 변수 만들기
2. 폭탄 설치 시 마다 변수값을 감소시키고 폭탄 설치 함수에서 조건문으로 처리
3. 남은 폭탄 개수를 화면에 표시해 봅시다.

※ 스페이스키를 누르면 게임을 재시작 하도록 구현해 봅시다.

1. 맵 구성과 변수를 초기화하는 함수를 만들어 봅시다.
2. 스페이스키를 눌렀을 때 1에서 만든 함수를 호출해 봅시다.

```
import pygame
import time
import random
```

```
#초기화
pygame.init()
```

```
#FPS 설정
FPS = 60
FramePerSec = pygame.time.Clock()
```

```
#색상 설정
WHITE = (255,255,255)
BLACK = (0,0,0)
BLUE = (0,0,255)
SKY = (150,200,255)
RED = (255,0,0)
```

#폰트 설정

```
font = pygame.font.SysFont(None, 50)
```

#화면 설정

```
GameDisplay = pygame.display.set_mode((510,510))
```

```
pygame.display.set_caption("Pygame Example")
```

#게임 초기값 설정

```
pos = [0,0]
```

```
start = 18
```

```
distance = 25
```

```
run = True
```

```
finish = False
```

```
text_pos = [100, 100]
```

```
text = "Congratulations!"
```

```
anim_time = time.time()
```

```
bomb_size = 10  
bomb_put_time = 0  
bomb_pos = [-1,-1]  
bomb_put_count = 5
```

```
#블록 생성  
block = []  
for i in range(20):  
    L = []  
    for j in range(20):  
        L.append(random.randrange(0,2))  
    block.append(L)
```

```
block[0][0] = 0  
block[-1][-1] = 0
```

```
def restart() :  
    global pos
```

```
global start  
global distance  
global run  
global finish  
global text_pos  
global text
```

```
global anim_time  
global bomb_size  
global bomb_put_time  
global bomb_pos  
global bomb_put_count
```

```
pos = [0,0]  
start = 18  
distance = 25  
run = True  
finish = False
```

```
text_pos = [100, 100]  
text = "Congratulations!"
```

```
anim_time = 0  
bomb_size = 10  
bomb_put_time = 0  
bomb_pos = [-1,-1]  
bomb_put_count = 5
```

```
global block
```

```
for i in range(20):  
    for j in range(20):  
        block[i][j] = random.randrange(0,2)
```

```
block[0][0] = 0  
block[-1][-1] = 0
```

```
def destroy_block(pos, block) :  
    global bomb_pos  
    dirs = [[0,1], [1,0], [0,-1], [-1,0]]  
    for dir in dirs :  
        try:  
            x = pos[0]+dir[0]  
            y = pos[1]+dir[1]  
            if x != -1 and y != -1 :  
                block[x][y] = 0  
            bomb_pos = [-1,-1]  
        except:  
            pass
```

```
def put_bomb(pos) :  
    global bomb_pos  
    global bomb_put_time  
    global bomb_put_count
```

```
if bomb_pos[0] == -1 and bomb_put_count > 0:
    bomb_pos = pos
    bomb_put_time = time.time()
    anim_time = bomb_put_time
    bomb_put_count -= 1
```

#게임 루프

while run :

#키보드 이벤트 처리

for event in pygame.event.get() :

if event.type == pygame.QUIT :

run = False

elif event.type == pygame.KEYDOWN :

if event.key == pygame.K_LEFT:

if pos[1]-1 > -1 and block[pos[0]][pos[1]-1] != 1 :

pos = [pos[0], pos[1]-1]

elif event.key == pygame.K_RIGHT :

if pos[1]+1 < 20 and block[pos[0]][pos[1]+1] != 1 :


```

        pos = [pos[0], pos[1]+1]
    elif event.key == pygame.K_UP :
        if pos[0]-1 > -1 and block[pos[0]-1][pos[1]] != 1 :
            pos = [pos[0]-1, pos[1]]
    elif event.key == pygame.K_DOWN :
        if pos[0]+1 < 20 and block[pos[0]+1][pos[1]] != 1 :
            pos = [pos[0]+1, pos[1]]
    elif event.key == pygame.K_d :
        #destroy_block(pos, block);
        put_bomb(pos)
    elif event.key == pygame.K_SPACE :
        restart()

if pos[0] == 19 and pos[1] == 19 :
    finish = True

cur_time = time.time()

```

#화면 초기화

GameDisplay.fill(WHITE)

#화면 그리기

#맵 그리기

for i in range(5, 510,25):

 pygame.draw.line(GameDisplay, BLACK, [5,i], [505,i], 2)

for i in range(5, 510,25):

 pygame.draw.line(GameDisplay, BLACK, [i,5], [i,505], 2)

#사용자 그리기

pygame.draw.circle(GameDisplay, BLUE, [start+pos[1]*distance,
start+pos[0]*distance], 10)

#블록 그리기

for i in range(20):

 for j in range(20):

```
if block[i][j] == 1 :  
    pygame.draw.circle(GameDisplay, BLACK, [start+j*distance,  
                                              start+i*distance], 10)
```

#폭탄 그리기

```
if finish == False and bomb_pos[0] > -1 :  
    if cur_time - bomb_put_time > 2.5 :  
        destroy_block(bomb_pos, block)  
    elif cur_time - bomb_put_time > 2 :  
        pygame.draw.circle(GameDisplay, RED, [start+bomb_pos[1]*distance,  
                                              start+bomb_pos[0]*distance], bomb_size*3)  
else :  
    if cur_time - anim_time > 0.5 :  
        if bomb_size == 10 :  
            bomb_size = 11  
        else :  
            bomb_size = 10  
    anim_time = time.time()
```

```
pygame.draw.circle(GameDisplay, RED, [start+bomb_pos[1]*distance,  
start+bomb_pos[0]*distance], bomb_size)
```

```
#폭탄 남은 횟수 표시
```

```
game_text = font.render(str(bomb_put_count), True, RED)  
GameDisplay.blit(game_text, [9+19*distance, 5+19*distance])
```

```
if finish :
```

```
    pygame.draw.rect(GameDisplay, WHITE, [text_pos[0], text_pos[1], 290, 40])  
    game_text = font.render(text, True, RED)  
    GameDisplay.blit(game_text, text_pos)
```

```
pygame.display.update()  
FramePerSec.tick(FPS)
```

```
#게임 종료
```

```
pygame.quit()
```