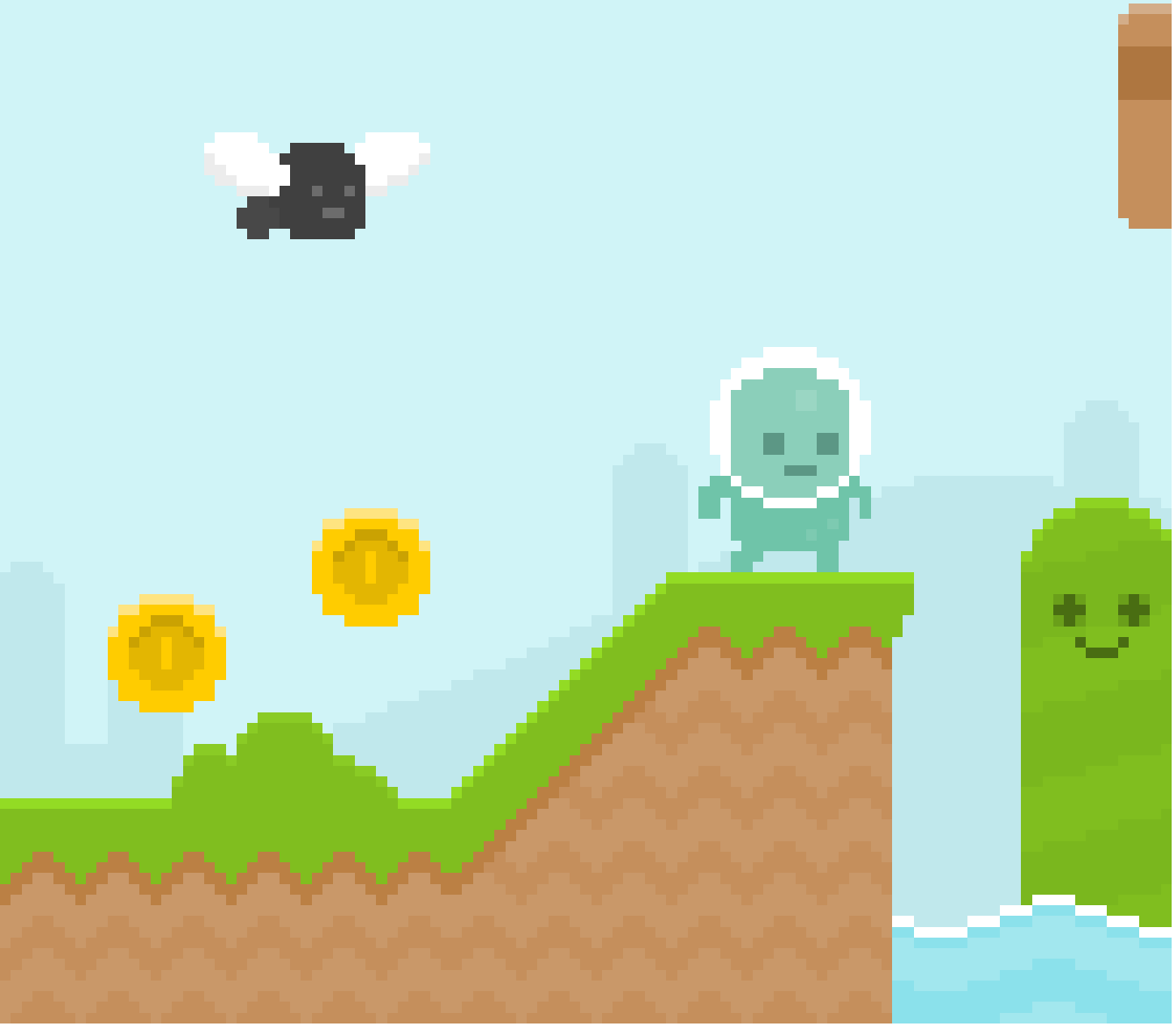

게임을 만들며 배우는 파이썬 4 (with pygame)

- 캐릭터, 점수 관리

오류중학교 코딩 동아리
강사: 여 은 선

확인합시다!

☆ https://github.com/chpink0518/oryuCC.git
소스 코드를 다운로드합니다

☆ infos4.ipynb 파일만 make_game폴더에서 카피해 옵니다
main.py는 3강에서 사용한 것에 더해 사용합니다

☆ infos4.ipynb 파일의 각 셀 단위를 main.py로 옮겨오며 테스트
-> 주의: 순서나 위치가 맞지 않음, 셀을 실행하지 마세요!!

이벤트 루프 수정

☆ infos4.ipynb의 1번째 셀

```
#01. change event loop also
startTime = 0

# player jumps only active mode
if gameActive:
    # do event check here
else:
    if event.type == pygame.KEYDOWN and event.key == pygame.K_SPACE:
        gameActive = True
        snail_rect.left = 800
        startTime = pygame.time.get_ticks()     # re-start time check

if player_rect.colliderect(snail_rect):
    gameActive = False
```

-> 게임 오버 상태에서
스페이스 바를 누르면
다시 게임이 액티브 상태

-> 달팽이 사각형 좌표를
바꾸는 이유는?

버틴 시간을 스코어로

☆ infos4.ipynb의 2번째 셀

```
#02. get time for score
score = 0

def display_score():
    c_time = int((pygame.time.get_ticks() - startTime)/1000)
    score_surf = my_font.render(f'SCORE: {c_time}', False, (64, 64, 64))
    score_rect = score_surf.get_rect(center = (400, 50))
    win.blit(score_surf, score_rect)
    return c_time

score = display_score()
```

-> 함수 정의 후 호출

-> pygame.time.get_ticks()
게임 경과 시간 체크(밀리세컨드: 1000ms == 1초)

-> 게임 오버 후 되돌아 가면 스코어는?

게임 오버 화면을 장식할 서피스 생성

☆ infos4.ipynb의 3번째 셀

```
#03. make surface for game over screen
player_stand = pygame.image.load('graphics/player/player_stand.png').convert_alpha()
player_stand = pygame.transform.rotozoom(player_stand, 0, 2)
player_stand_rect = player_stand.get_rect(center = (400, 200))

game_msg = my_font.render('Press space to run', False, (111, 196, 169))
game_msg_rect = game_msg.get_rect(center = (400, 325))
```

-> pygame.transform.rotozoom(대상, 회전각도, 확대 배수)

생성한 서피스들을 그리자

☆ infos4.ipynb의 4번째 셀

```
#04. blit game over screen
else:
    win.fill((94, 129, 162))
    win.blit(player_stand, player_stand_rect)

    score_msg = my_font.render(f'Your score: {score}', False, (111, 196, 169))
    score_msg_rect = score_msg.get_rect(center = (400, 325))

    win.blit(score_msg, score_msg_rect)
    win.blit(game_msg, game_msg_rect)
```

-> game_msg와 달리 score_msg 서피스를 이곳에서 만드는 이유는?

-> pygame.transform.rotozoom(대상, 회전각도, 확대 배수)
값을 바꿔 테스트해 보자

```
player_stand = pygame.transform.rotozoom(player_stand, 180, 0.5)
```

타이머 이벤트를 만들자

☆ infos4.ipynb의 5번째 셀

```
#05. create a custom timer event
enemy_timer = pygame.USEREVENT
pygame.time.set_timer(enemy_timer, 1500)

# event loop
if event.type == enemy_timer :
    print('enemy')
```

-> 타이머 간격: ms
- 1.5초

-> pygame event 수 : 32개
이중 23개를 pygame이 이미 사용 중
24부터 32까지 유저 이벤트 생성 가능(9개)

-> 이후 이벤트 추가 시 pygame.USEREVENT + 1과 같이 값을 증가해서 사용하면 된다

더 많은 달팽이들

☆ infos4.ipynb의 6번째 셀

```
#14. change snail movement
from random import randint          -> 랜덤한 정수 얻기: randint

# remove snail_rect & collision detection part    -> snail_rect와 충돌탐지 부분 제거
enemy_rect_list = []                              -> rect 정보 저장할 빈 리스트 생성

# event loop - gameActive
if event.type == enemy_timer:
    enemy_rect_list.append(snail_surface.get_rect(bottomright = (randint(900, 1100), 300)))
```

-> 랜덤한 정수를 구하기: randint

-> 서피스는 하나, rect(위치-랜덤)는 여러 개
- 화면을 벗어난 위치에서 x 좌표 생성
- y 좌표는 땅에 붙어서 이동하도록 고정

더 많은 달팽이들 (cont'd)

☆ infos4.ipynb의 6번째 셀

```
# define the obstacle movement function
def enemy_movement(enmy_rct_list):
    if enmy_rct_list:  # list is not null
        for enmy_rct in enmy_rct_list:
            enmy_rct.x -= 5
            win.blit(snail_surface, enmy_rct)
        enmy_rct_list = [enmy for enmy in enmy_rct_list if enmy.right > 0]
    return enmy_rct_list

# comment snail movement block
# obstacle movement - main loop
obstacle_rect_list = obstacle_movement(obstacle_rect_list)
```

-> 화면을 벗어난 rect는 지우기

-> 리스트의 모든 x 좌표 값을 5만큼 뺀 위치에 달팽이를 그린다
-> 화면을 벗어난 rect는 지워준다

파리의 등장

☆ infos4.ipynb의 7번째 셀

```
#07. add different enemies - fly
fly_surf = pygame.image.load('graphics/fly/fly1.png').convert_alpha()

if event.type == enemy_timer:
    if randint(0, 2):
        enemy_rect_list.append(snail_surface.get_rect(bottomright = (randint(900, 1100), 300)))
    else:
        enemy_rect_list.append(fly_surf.get_rect(bottomright = (randint(900, 1100), 180)))
```

-> 파리 서피스 생성

-> 0,1 중 랜덤한 값을 구해 1이면 달팽이, 0이면 파리
- 파리는 공중에 떠 있으므로 y 좌표가 180

파리의 등장 (cont'd)

☆ infos4.ipynb의 7번째 셀

```
def enemy_movement(enmy_rct_list):
    if enmy_rct_list:  # list is not null
        for enmy_rct in enmy_rct_list:
            enmy_rct.x -= 5
            if enmy_rct.bottom == 300: win.blit(snail_surface, enmy_rct)
            else: win.blit(fly_surf, enmy_rct)

        enmy_rct_list = [enmy for enmy in enmy_rct_list if enmy.right > 0]
    return enmy_rct_list
```

-> bottom 값에 따라 어느 서피스를 그릴지 결정

충돌 감지 방법 수정하기

☆ infos4.ipynb의 8번째 셀

```
#08. fix the collision logic

def isnt_Collide(plyr_rect, enmy_rect_list):
    if enmy_rect_list:
        for enmy_rect in enmy_rect_list:
            if plyr_rect.colliderect(enmy_rect): return False
    return True

# main loop
gameActive = isnt_Collide(player_rect, enemy_rect_list)

# game over
enemy_rect_list.clear()
```