

코드쌤: 지혜야, 슬기야! 지난 시간에는 우리가 직접 그린 그림이나 다운받은 이미지 파일을 blit 마법으로 게임 화면에 “짠!” 하고 나타나게 했었지?

지혜: 네! 제 아기 용이 화면 중앙에서 멋지게 등장했어요! 파일 경로 때문에 잠깐 애먹었지만, 이제 자신 있다고요! 🐉✨

슬기: 맞아요! 이미지 위에 직접 그림도 그릴 수 있어서 신기했어요! 근데 선생님, 게임에 점수나 메시지 같은 글자도 쓰고 싶을 때 있잖아요. “게임 시작!” 이나 “100점!” 이렇게요. 그건 어떻게 해요?

코드쌤: 오, 아주 중요한 질문이야, 슬기! 게임에서 글자는 정말 중요한 역할을 하지. 점수판, 게임 제목, 캐릭터 대사 등등! 오늘은 바로 그 글자를 우리 게임 화면에 예쁘게 써넣는 방법을 배울 거란다. 마치 간판을 다는 것처럼!

(코드 조각 1: 폰트 객체 생성 및 텍스트 렌더링 - 있는 코드 그대로!)

```
import pygame

pygame.init()

#창 크기 설정
WINDOW_WIDTH = 800
WINDOW_HEIGHT = 600

#윈도우 surface(표면) 생성, 창 제목 설정
window_surface = pygame.display.set_mode((WINDOW_WIDTH, WINDOW_HEIGHT))
pygame.display.set_caption("글자를 써보자! 폰트 마법!")

#배경색상 설정 (이번엔 검은색으로!)
window_surface.fill((0, 0, 0)) # BLACK

#hero 이미지를 로드 및 위치 설정 (복습!)
# (이 부분은 hero.png 파일이 'data' 폴더 안에 있다고 가정해요!)
hero_image = pygame.image.load("data/hero.png") # 파일이 없으면 여기서 에러가 날 수 있어요!
hero_rect = hero_image.get_rect()
hero_rect.center = (WINDOW_WIDTH/2, WINDOW_HEIGHT/2)

# --- 여기가 오늘의 핵심! 글자 쓰기! ---

# 1. 폰트 객체 만들기: "어떤 글씨체로, 얼마나 크게 쓸까?"
# pygame.font.SysFont("글꼴이름", 크기)
# "Arial"은 윈도우나 맥에 기본으로 있는 글꼴 중 하나야.
```

```

system_font = pygame.font.SysFont("Arial", 50) # Arial 글꼴, 크기 50

# 2. 텍스트 렌더링: "이 글자를, 이런 색으로, 이렇게 그려줘!"
# 폰트객체.render("쓸 내용", 안티앨리어싱여부, 글자색, 배경색=None)
# 안티앨리어싱(True)은 글자를 부드럽게 보이게 해주는 마법!
# 글자색은 "green", 글자 배경색은 "blue"로 해볼까?
system_text = system_font.render("Test Text Render", True, "green", "blue") # 원래 코드의 텍스트로!
# print(type(system_text)) # 이것도 <class 'pygame.Surface'>! 글자도 이미지처럼 하나의 '표면'이 되는 거야

# 3. 텍스트 위치 정하기: 이 글자 이미지를 어디에 놓을까? (이미지처럼!)
system_text_rect = system_text.get_rect()
system_text_rect.topleft = (20, 20) # 화면 왼쪽 위에서 조금 떨어진 곳에!

#메인 게임 루프 (이 부분은 원래 코드의 게임 루프를 참고할게요!)
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    #hero 이미지 blit
    window_surface.blit(hero_image, hero_rect)

    #text blit
    window_surface.blit(system_text, system_text_rect)

    #가로선 그리기
    pygame.draw.line(window_surface, (255, 255, 255), (0, 77), (WINDOW_WIDTH, 77), 5)

    pygame.display.update()

pygame.quit()

```

지해: pygame.font.SysFont("Arial", 50)요? SysFont는 '시스템 폰트'라는 뜻인가요? "Arial"은 글씨체 이름이고, 50은 글자 크기고요?

코드쌤: 정확해, 지해! pygame.font.SysFont()는 우리 컴퓨터에 이미 설치되어 있는 기본 글꼴(시스템 폰트) 중에서 "Arial"이라는 이름의 글꼴을 사용해서, 크기 50짜리 글자를 쓸 준비를 하는 거야. 마치 우리가 글씨 연습장에 어떤 펜과 어떤 크기로 쓸지 정하는 것과 같지.

슬기: 그럼 `system_font.render("Test Text Render", True, "green", "blue")`는요? `render`는 ‘그리다’, ‘표현하다’ 이런 뜻인데... “Test Text Render” 라는 글자를 초록색으로, 배경은 파란색으로 그려달라는 건가요? `True`는 뭐예요?

코드샴: 오, 슬기 영어 실력도 좋은데! 맞아! `render()` 메소드는 우리가 정한 폰트(`system_font`)를 사용해서, 주어진 문자열(“Test Text Render”)을 실제 그림(`Surface`)으로 만들어주는 역할을 해.

- 첫 번째 인자는 “**쓸 내용**” (문자열).
- 두 번째 인자 `True`는 “**안티앨리어싱(Anti-aliasing) 사용 여부**”야. 이건 글자 가장자리를 부드럽게 처리해서 계단 현상 없이 예쁘게 보이도록 하는 마법 같은 기술이란단. `False`로 하면 좀 더 딱딱하게 보일 수 있어.
- 세 번째 인자는 “**글자색**” (여기서는 “green”).
- 네 번째 인자는 “**글자 배경색**” (여기서는 “blue”). 만약 배경색을 투명하게 하고 싶으면 이 부분을 생략하거나 `None`으로 하면 돼.

지혜: 와! 그럼 `system_text`도 `hero_image`처럼 하나의 이미지(`Surface`)가 되는 거네요? 그래서 `system_text.get_rect()`로 위치 잡는 사각형 정보를 얻고, `system_text_rect.topleft`로 위치를 정하는 거군요!

코드샴: 완벽해! 글자도 일단 이미지처럼 만들어진 다음, 우리가 이미지를 다뤘던 방식 그대로 화면에 `blit` 해주면 되는 거야. 게임 루프 안에서 `window_surface.blit(system_text, system_text_rect)` 코드가 바로 그 역할을 하고 있지!

코드샴: 자, 그런데 시스템 폰트는 종류가 한정적일 수 있어. 만약 우리가 다운받은 아주 특별하고 예쁜 글씨체 파일을 쓰고 싶다면 어떡할까? 예를 들어 게임 분위기에 딱 맞는 손글씨 폰트나, 고딕 느낌의 웅장한 폰트 말이야. 그럴 때 **TTF(TrueType Font) 파일**을 사용한다!

슬기: TTF 파일이요? 그건 뭐예요? 확장자가 `.ttf`로 끝나는 파일인가요?

코드샴: 맞아! TTF 파일은 글꼴의 모양 정보를 담고 있는 파일이야. 인터넷에서 무료 또는 유료로 다양한 TTF 폰트 파일들을 다운받을 수 있어. 만약 우리가 ‘잘난체’라는 아주 멋진 무료 폰트(`Jalnan.ttf`)를 다운받아서, 아까 `hero.png`를 넣어둔 `data` 폴더처럼, 이번엔 `fonts`라는 새 폴더를 만들고 그 안에 `Jalnan.ttf` 파일을 넣었다고 상상해 보자.

프로젝트 폴더 구조 (상상):

내_게임_프로젝트/

```
|—— 나의_게임코드.py
|—— data/
|   |—— hero.png
|—— fonts/           (폰트 파일들을 모아둘 'fonts' 라는 이름의 하위 폴더)
    |—— Jalnan.ttf    (다운받은 TTF 폰트 파일!)
```

코드샴: 이렇게 `Jalnan.ttf` 파일이 `fonts` 폴더 안에 있다면, `system_font`를 만들었던 것처럼 `pygame.font.Font()` 함수를 사용해서 새로운 폰트 객체를 만들 수 있어.

예시 코드 (이건 지금 코드에는 없지만, 이렇게 할 수 있다는 설명이에요!):

```
# 만약 Jalnan.ttf 파일이 fonts 폴더에 있다면...
# custom_font = pygame.font.Font("fonts/Jalnan.ttf", 60)
# custom_text = custom_font.render("레벨 업!", True, "white")
# custom_text_rect = custom_text.get_rect()
# custom_text_rect.center = (WINDOW_WIDTH // 2, WINDOW_HEIGHT - 50)
# 그리고 게임 루프에서 window_surface.blit(custom_text, custom_text_rect) 해주면 되겠지?
```

지혜: 와! pygame.font.SysFont() 대신 pygame.font.Font()를 쓰고, 첫 번째 인자로 글꼴 이름 대신에 TTF 파일 경로를 직접 넣어주는 거네요!

코드쌤: 정확해! pygame.font.Font()는 우리가 제공한 TTF 파일에서 직접 글꼴 정보를 읽어와서 폰트 객체를 만들어줘. 그래서 훨씬 다양하고 개성 있는 글씨체를 게임에 사용할 수 있게 되는 거지! 단, 이 경우에도 파일 경로가 아주 중요해. 아까 hero.png가 data 폴더에 정확히 있어야 했던 것처럼, Jalnan.ttf 파일도 fonts 폴더 안에 정확히 있어야 하고, 코드에도 fonts/Jalnan.ttf라고 올바르게 적어줘야 컴퓨터가 찾을 수 있단다. 만약 파일을 못 찾으면 어떻게 될까?

슬기: FileNotFoundError! (또는 비슷한 폰트 관련 에러) 이제 알아요! 파일이 없으면 프로그램이 멈추거나 빨간 글씨로 에러 메시지가 나오겠죠!

코드쌤: 맞아! 그래서 파일을 사용할 때는 항상 파일 이름과 경로를 꼼꼼하게 확인하는 습관이 중요해. 지금 우리가 보고 있는 코드에서는 hero_image = pygame.image.load("data/hero.png") 이 부분이 만약 data 폴더에 hero.png 파일이 없으면 에러가 발생할 수 있는 부분이지.

(원래 코드의 게임 루프 부분을 보며)

지혜: 게임 루프 안에서 window_surface.blit(hero_image, hero_rect)로 히어로 이미지를 그리고, window_surface.blit(system_text, system_text_rect)로 아까 만든 “Test Text Render” 글자를 그리는 거군요! 그리고 pygame.draw.line으로 선도 그리고요!

슬기: blit 할 때도 이미지랑 똑같이 window_surface.blit(텍스트이미지, 텍스트위치) 이렇게 하는 거군요! 진짜 글자도 이미지처럼 다루는 거네요! 신기하다!

코드쌤: 맞아! Pygame에서 글자를 다루는 핵심은 1) 폰트 객체 만들기 (SysFont 또는 Font), 2) 폰트 객체로 텍스트를 이미지(Surface)로 렌더링하기, 3) 그 이미지를 화면에 blit하기 이 세 단계란다. 이제 너희는 게임에 점수도 표시하고, 멋진 게임 제목도 달고, 캐릭터들의 대사도 보여줄 수 있게 된 거야!

지혜: 선생님, 그럼 한글 폰트도 TTF 파일만 있으면 쓸 수 있는 거예요?

코드쌤: 물론이지! 한글을 지원하는 TTF 폰트 파일(NanumGothic.ttf 같은)을 구해서 pygame.font.Font()에 경로를 지정해주면 깨끗한 한글을 게임 화면에 쓸 수 있단다. 단, SysFont로 한글을 쓰려고 하면 시스템에 따라 잘 안 나올 수도 있어서, 한글을 쓸 때는 TTF 파일을 사용하는 것이 훨씬 안정적이야.

슬기: 우와! 오늘 폰트 마법 완전 마스터한 것 같아요! 이제 제 게임에 막 “몬스터 등장!” “필살기 발사!” 이런 글자도 크게 써넣을 수 있겠어요! 너무 신나요!

코드쌤: 하하, 너희들의 열정에 선생님도 기분이 좋아지는구나! 오늘 배운 내용으로 여러 가지 글씨체와 색깔, 크기로 다양한 메시지를 화면에 표현해보렴. 너희들의 게임이 더욱 풍부하고 재미있어질 거야!