꽃 자판기

4조

아이디어

```
print('-----------------------------------')
print('----\t장미\t\t국화\t\t히아신스\t----')
print('----\t ■ \t\t □ \t\t   ■ \t----')
print('-----------------------------------')
print('----\t튤립\t\t데이지\t\t카네이션\t----')
print('----\t □ \t\t ■ \t\t   ■ \t----')
print('-----------------------------------')
print('----\t목화\t\t해바라기\t안개꽃\t----')
print('----\t □ \t\t ■ \t\t   ■ \t----')
print('-----------------------------------')
```

```
자판기
C:\Users\choha\AppData\Local\Programs\Python\Python37\pythc
-----------------------------------
----    장미      국화      히아신스  ----
----     ■        □          ■     ----
-----------------------------------
----    튤립      데이지    카네이션  ----
----     □        ■          ■     ----
-----------------------------------
----    목화      해바라기  안개꽃    ----
----     □        ■          ■     ----
-----------------------------------
```

순서

1. 꽃의 정보를 class 로 저장
2. 사용자가 가진 금액 입력
3. 금액 > 0 : 반복문 순회하며 꽃을 구매
4. 금액 <= 0 : 프로그램 종료

자판기 구동 심플 테스트

```
# 자판기 프로그램 작성 👇 ---------------------------------
if not credit:
    print("잔액이 부족합니다.")

'''
# 리스트
flowers = ['rose', '국화']
flowers_qtt = [3, 5]
flowers_price = [500, 500]

# 딕셔너리
flowers = dict()
flowers['rose'] = 3
flowers['국화'] = [3, 500]
'''

while credit:
    cur_flower = input("꽃을 선택해주세요 >>> ")
    for flower in flowers:
        if cur_flower == flower.name:
            if flower.qtt > 0:
                print(flower.name, ' 꽃을 드립니다...')
                credit -= flower.price
print('잔액이 부족합니다.')


# 구성한 화면을 실제 GUI에 반영
pygame.display.flip()

pygame.quit()
```

```python
import datetime as dt   # 유통기한의 표현을 위한 datetime 라이브러리
import pygame           # GUI 사용을 위한 pygame 라이브러리

# game 초기화
pygame.init()

# 전역 변수 선언
# 화면 관련 ----------
# 색상
BLACK  = (  0,   0,   0)
WHITE  = (255, 255, 255)
BLUE   = ( 10,  10, 255)
GREEN  = (183, 240, 177)
GREEN2 = ( 29, 219,  22)
YELLOW = (250, 244, 192)
RED    = (241,  95,  95)
RED2   = (240,   0,   0)

# 글씨
font_lang = pygame.font.SysFont('모리스9', 20, False, False)
font_credit = pygame.font.SysFont('모리스9', 30, True, True)
font_name = pygame.font.SysFont('모리스9', 15, False, False)

# GUI 창
width, height = 500, 750
screen = pygame.display.set_mode((width, height))
pygame.display.set_caption('4조 🌸꽃🌸 자판기')

# 종료 조건
done = False
clock = pygame.time.Clock()
```

GUI 도입

색상 선정

각 항목별
글씨 설정

창 크기 설정

종료 조건

결과물

```python
# 자판기 이름
vending = pygame.image.load('image/title.png').convert()

# 지폐
credit = int(input("투입할 금액을 입력하세요 > > >  "))
credit_str = font_credit.render(str(credit), True, BLUE)


# 버튼 class
class Button:
    def __init__(self, a, b, x, y):
        mouse = pygame.mouse.get_pos()
        click = pygame.mouse.get_pressed(3)

        bx, by, bw, bh = x + 30, y + 125, 60, 25

        # 품절일 때 - 빨간색으로 표시
        if flowers[a][b].qtt <= 0:
            # 마우스 오버 액션
            if bx < mouse[0] < bx + bw and by < mouse[1] < by + bh:
                pygame.draw.rect(screen, RED2, (bx, by, bw, bh), 0)
            else:
                pygame.draw.rect(screen, RED, (bx, by, bw, bh), 0)
        # 수량이 남았을 때 - 초록색으로 표시
        else:
            # 마우스 오버 액션
            if bx < mouse[0] < bx + bw and by < mouse[1] < by + bh:
                pygame.draw.rect(screen, GREEN2, (bx, by, bw, bh), 0)
                # 마우스 클릭 액션
                if click[0]:
                    pygame.time.delay(60)
                    buy_action(a, b)
            else:
                pygame.draw.rect(screen, GREEN, (bx, by, bw, bh), 0)
```

자판기 이름 설명

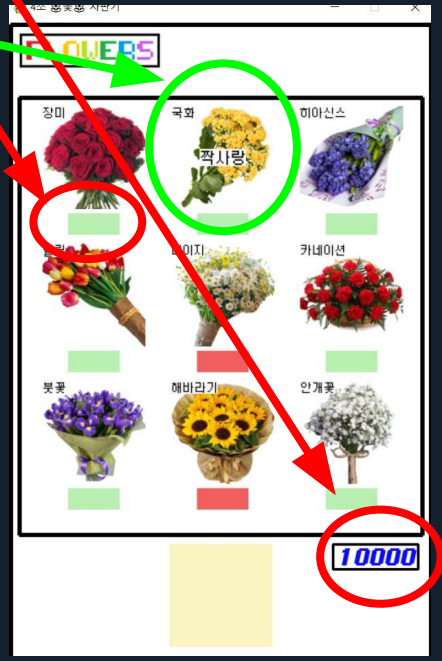
금액 표시

버튼 설정

__init__ : class 시작에 필요한 명령어

마우스가 위에 있을 때, 클릭 시의 각 반응

버튼의 크기

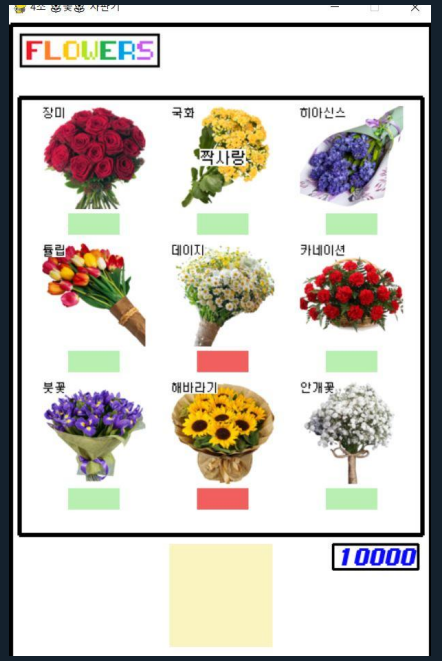


```python
# 꽃 class
class Flower:
    def __init__(self, name, qtt, price, lang, doc, img):
        # 이름
        self.name = name
        # 수량
        self.qtt = qtt
        # 가격
        self.price = price
        # 꽃말
        self.lang = lang
        # 설명
        self.doc = doc
        # 사진
        self.img = img
        # 유통기한 (당일로부터 15일)
        self.exdate = dt.datetime.today() + dt.timedelta(days=15)
```

꽃 class 정의

```python
# 꽃 초기화
# 이곳에서 꽃 정보를 수정해주세요
rose = Flower('장미', 3, 5000, '열렬한 사랑', '', 'rose')
chrysanthemum = Flower('국화', 3, 5000, '짝사랑', '', 'chrysanthemum')
hyacinth = Flower('히아신스', 3, 5000, '겸손한 사랑', '', 'hyacinth')
tulip = Flower('튤립', 3, 5000, '사랑의 고백', '', 'tulip')
daisy = Flower('데이지', 0, 5000, '희망과 평화', '', 'daisy')
carnation = Flower('카네이션', 3, 5000, '영원한 사랑', '', 'carnation')
iris = Flower('붓꽃', 3, 5000, '좋은 소식', '', 'iris')
sun = Flower('해바라기', 0, 5000, '프라이드', '', 'sunflower')
gypsophila = Flower('안개꽃', 3, 5000, '깨끗한 마음', '', 'gypsophila')
```

꽃별로 정보 입력



```python
# 9개의 꽃을 3*3 행렬에 저장
flowers = [[rose, chrysanthemum, hyacinth],
           [tulip, daisy, carnation],
           [iris, sun, gypsophila]]

# 각 꽃의 사진을 객체로 불러와 저장
flower_images = []
for i in range(3):
    tmp = []
    for j in range(3):
        img_str = 'image/' + flowers[i][j].img + '.png'
        tmp.append(pygame.image.load(img_str).convert_alpha())
    flower_images.append(tmp)

fw, fh = 120, 120

# 꽃 이름을 텍스트로 표현
flower_names = []
for i in range(3):
    tmp = []
    for j in range(3):
        tmp.append(font_name.render(flowers[i][j].name, True, BLACK))
    flower_names.append(tmp)

# 꽃말을 텍스트로 표현
flower_texts = []
for i in range(3):
    tmp = []
    for j in range(3):
        tmp.append(font_lang.render(flowers[i][j].lang, True, BLACK))
    flower_texts.append(tmp)


# 꽃 사진 class
class FlowerImg:
    def __init__(self, a, b, x, y):
        # 꽃 사진 띄우기
        screen.blit(flower_images[a][b], (x, y))
```

3 * 3 행렬로 나타내기

꽃 사진 불러오기

꽃 이름,

꽃말,

꽃 사진

나타내기

```python
# 꽃 사진 class
class FlowerImg:
    def __init__(self, a, b, x, y):
        # 꽃 사진 띄우기
        screen.blit(flower_images[a][b], (x, y))
        # Surface 위에 꽃 이름 띄우기
        name_sfc = pygame.Surface(flower_names[a][b].get_size())
        name_sfc.fill(WHITE)
        name_sfc.blit(flower_names[a][b], (0, 0))
        screen.blit(name_sfc, (x, y))

        mouse = pygame.mouse.get_pos()
        # 마우스 오버되면 꽃말 띄우기
        if x < mouse[0] < x + fw and y < mouse[1] < y + fh:
            lang_sfc = pygame.Surface(flower_texts[a][b].get_size())
            lang_sfc.fill(WHITE)
            lang_sfc.blit(flower_texts[a][b], (0, 0))
            screen.blit(lang_sfc,
                        (x + 60 - lang_sfc.get_width() // 2, y + 50))
```

자판기 스크린 위에
꽃 사진 띄우기

마우스가 어떤 꽃 사진
위에 있는지 확인하고,
해당 꽃의 꽃말 띄우기

```python
# Main Game Start
if __name__ == "__main__":
    while not done:
        # 프레임 설정
        clock.tick(10)
        credit_str = font_credit.render(str(credit), True, BLUE)

        # 버튼 눌렀을 때 - 구매 함수
        def buy_action(a, b):
            global credit
            #
            flowers[a][b].qtt -= 1
            #
            credit -= flowers[a][b].price
            # 산 꽃의 이미지 띄우기
            screen.blit(flower_images[a][b], (width // 2 - 60, height - 140))

        # Event Loop
        for event in pygame.event.get():    # 사용자 이벤트 감지
            if event.type == pygame.QUIT:   # 사용자가 닫기 버튼을 누르면
                done = True                 # 종료

        # 기본 화면 구성 ----------------------------------------------------
        screen.fill(WHITE)

        # 자판기 몸체
        screen.blit(vending, (10, 10))
        pygame.draw.rect(screen, BLACK, (5, 5, width - 10, height - 10), 5)
        pygame.draw.rect(screen, BLACK, (15, 90, width - 30, height - 240), 5)
        # 꽃 나오는 곳
        pygame.draw.rect(screen, YELLOW, (width // 2 - 60, height - 140, 120, 120), 0)
        # 지폐 투입구
        credit_window = pygame.draw.rect(screen, BLACK, (380, 610, 100, 30), 3)
        screen.blit(credit_str, (385, 610))
```

자판기 구동과

기본 화면 구성


```python
# 꽃
for i in range(3):
    for j in range(3):
        cx, cy = j * 150 + 42, i * 160 + 100
        # 사진
        FlowerImg(i, j, cx, cy)
        # 버튼
        Button(i, j, cx, cy)
```

금액이 부족한 경우

```
flag = False
if credit <= 0:
    print('잔액이 부족합니다. 금액을 더 넣으시겠습니까?')
    while not flag:
        new_money = input("금액을 입력하거나, 'n' 또는 0을 입력하세요>>>")
        if new_money == 'n' or new_money == '0':
            flag = True
            done = True
        elif new_money.isdigit():
            if int(new_money):
                # 금액을 추가한 경우
                credit += int(new_money)
                flag = True
        else:
            print("다시 입력해주세요!", end=' ')
            flag = False
```

보완점

```
# 0으로 집계 되는 부분,,
credit_str = font_credit.render(str(credit), True, BLUE)
```

- 잔액이 0일 때 화면상 문제

```
# 산 꽃의 이미지 띄우기
screen.blit(flower_images[a][b], (width // 2 - 60, height - 140))
```

- 꽃 선택시 화면 delay 문제

감 사 합 니 다