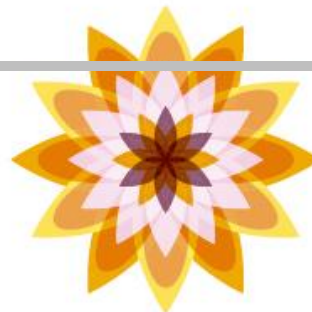
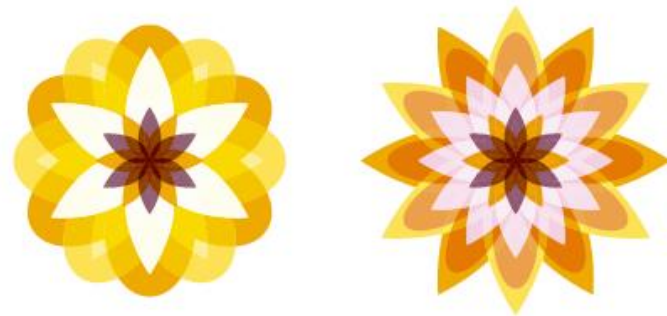


Chapter 11
점수 확인



1. 폰트 고르기

- 아래 두 줄을 프로그램의 셋업에 넣습니다.

```
score = 0  
font = pygame.font.Font(None,20)
```

- 1번째 줄은 점수(score)라는 변수를 만듭니다.
- 2번째 줄은 변수를 표시할 폰트(font)를 만들고요.
- 2번째 줄은 파이게임 모듈에 이어 그 안의 font 모듈을 찾고, font 모듈 안에서 다시 폰트(Font) 클래스를 찾습니다.
- 폰트 클래스는 폰트를 고르기 위해 쓰는 것이 아닙니다.
- "None"이라고 써서 파이게임한테 시스템 기본 폰트를 고르라고 시키는 거지요.
- "20"은 폰트의 크기입니다.

2. Global은 전체에서 찾아

- 점수 변수를 만들고 '점수'라는 글자를 쓸 때 사용할 폰트도 만들었습니다.
- 이제 악당 클래스 안에 score() 함수를 만들겠습니다.
- 악당 클래스 안의 함수들 아래 다음 코드를 추가합니다.

```
class Badguy:  
    (중략)  
    def score(self):  
        global score  
        score += 100
```

2. Global은 전체에서 찾아

- 점수를 악당 클래스에 넣는 이유가 뭐냐고요?
- 점수는 파이터 클래스 안에 있어야 하는 거 아니냐고요?
- 점수는 파이터와 아무 상관없습니다.
- 점수는 악당이 삭제될 때 바뀝니다. 악당이 죽으면 점수가 바뀌지요.
- 2번째 줄의 global score는 뭘까요?
- 함수 안에서 변수를 바꾸면 파이썬은 해당 함수 안에서만 변수를 찾습니다.
- 함수에 self가 들어 있으면 해당 클래스 안에서만 찾지요.
- 하지만 가끔은 클래스나 함수 밖에서 만든 변수를 쓸 필요가 있습니다.
- 위 점수 변수는 악당 클래스 밖에서도 필요합니다.
- 따라서 앞에 global(전역)이라는 단어를 붙였습니다.
- 그러면 파이썬은 점수 변수가 클래스나 함수 밖 프로그램의 어딘가에 또 있다는 걸 알게 되고, 그것을 찾으러 느릿느릿 움직이겠지요.
- global을 쓰지 않으면, 함수 밖은 찾아보지도 않을 테지만요.

2. Global은 전체에서 찾아

- '점수를 `__init__()` 함수 안에서 만들고, 여기 `score()` 함수 안에서 `self. score`를 사용해도 되지 않나?' 생각했나요?
- 아쉽지만 불가능합니다
- `__init__()` 함수 안에 이미 특정 악당을 만들었으니깐요.
- 점수는 전체 게임 안에서 올라가야 하는데, 특정 악당을 죽일 때만 점수가 올라가면 안 되겠지요.
- 그래서 `global`이라고 쓴 거랍니다.
- 마지막 줄은 `score()` 함수가 불러갈 때마다 점수를 100씩 증가시키는 코드입니다.

3. 악당을 지우기 전에 점수 증가

- 미사일이 악당에게 닿을 때마다 score() 함수를 불러와야 하므로 아래같이 미사일과 악당의 충돌을 감지하는 루프 안에서 부릅니다.
- 악당을 삭제하기 전에 score() 함수를 불러야 합니다. 그렇지 않으면 score() 함수를 부를 악당이 없으니까요.

```
i = 0
while i < len(badguys):
    j = 0
    while j < len(missiles):
        if badguys[i].touching(missiles[j]):
            badguys[i].score()
            del badguys[i]
            del missiles[j]
            i -= 1
            break
        j += 1
    i += 1
```

4. 스크린에 글자 표시

- 계산한 점수를 스크린에 나타내려면 다음을 가장 아래 줄 바로 위에 추가해야 합니다.

```
screen.blit(font.render("Score: "+str(score),True,(255,255,255)),(5,5))  
pygame.display.update()
```

- 어떤 것을 화면에 표시한다는 것은, 그것의 이미지를 만드는 것과 같습니다.
- `screen.blit()` 함수를 써서 점수 이미지를 만듭니다.
- `blit()` 함수는 표시할 객체(여기서는 텍스트)와 표시할 위치라는 2개의 인자를 필요로 합니다.
- 표시할 객체는 `font.render()` 함수에 있지요.
- 우리가 앞에서 정한 서체로 점수를 표시할 거예요.

4. 스크린에 글자 표시

- blit() 함수의 첫 번째 인자 font.render() 함수는 3개의 인자를 갖습니다.
- 첫 번째 인자 "Score: "+str(score) 는 표시할 대상입니다.
- 우리는 2개를 표시하려 합니다.
- 첫 번째는 따옴표 안에 들어 있는 텍스트, "Score: "(공백 포함)입니다.
- Score라고 쓴 다음에는 실제 점수 변수의 값을 표시하고 싶습니다.
- 하지만 점수 변수는 숫자입니다.
- 숫자를 표시(render)하려 하면 오류가 나니까 str(score)로 점수 변수의 값을 문자열(string)로 바꿉니다
- + 기호는 2개의 문자열(" Score : "도 문자열입니다.)을 합쳐 하나의 긴 문자열로 만들라는 뜻입니다.

4. 스크린에 글자 표시

- 두 번째 인자 True는 안티 앨리어싱(anti-aliasing)하라는 뜻입니다.
- 안티 앨리어싱은 스크린에 문자가 표시될 때 뾰족한 끝을 둥글리는 기법입니다.
- 세 번째 인자 (255, 255, 255)는 텍스트의 색을 정합니다 지금은 흰색으로 정했습니다.
- blit()함수의 두 번째 인자인 (5, 5)는 텍스트의 위치를 정합니다. (blit() 함수의 첫 번째 인자는 font.render() 함수였습니다.)
- 위치는 가장 위 왼쪽의 좌표를 말합니다.
- 따라서 텍스트의 가장 위 왼쪽의 좌표는 (5, 5)입니다.

4-1. 문자열

- 수학 선생님처럼, 파이썬도 우리가 문자와 숫자를 더하려 하면 싫어합니다.
- 숫자와 숫자의 합은 파이썬도 이해합니다.
- 3+4를 입력하면 7이라고 답하지요.
- 문자와 문자도 더할 수 있습니다.
- "사랑" + "스러운"를 입력하면 사랑스러운'이라고 답합니다.
- 하지만 문자와 숫자를 더하려고 하면 발끈합니다
- 문자(character)를 나열하면, 단어든 의미 없는 글자의 조합이든 문자열(string)로 취급됩니다.
- 문자열은 항상 따옴표 안에 씁니다. 그러면 합칠 수 있습니다
- "qwerty" + "asdf"는 'qwertyasdf'입니다.

4-1. 문자열

- 가끔은 숫자처럼 보이지만 실제로는 문자인 숫자가 필요할 때도 있습니다.
- 이럴 때 `str()`라는 함수를 씁니다. 이 함수는 숫자를 문자열로 바꿔 줍니다.
- `75+75`는 150이지만, `str(75)+str(75)`는 "7575"입니다.
- `score`가 57이면, `str(score) + "75"` 는 5775입니다.
- 75를 따옴표 안에 써서 문자열로 바꾸었습니다.
- 단지 "score"라고 쓴다고 해서 57이 문자열로 바뀌는 것은 아닙니다.
- 이렇게 하면 변수 이름이 문자열로 바뀌게 돼 `score`라는 단어가 표시됩니다.
- `str(score) + 75` 라고 쓰면 에러가 날 것입니다.
- 왜냐하면 숫자에 문자열을 더할 수 없기 때문입니다.
- "Score: " + `str(57)`이라고 입력하면 Score: 57이라고 표시될 것입니다
- 2개의 문자열을 더했기 때문입니다
- "Score: " + "57"이라고 입력하면 마찬가지로 Score: 57이라고 나올 테고요



Thank You
