

※ 파이썬 프로젝트 워크북 QR 리스트

No	페이지	Q.R.	URL	내용
0	전체		https://m.site.naver.com/lpsjs	파이썬 프로젝트 코딩 전체.zip
01	1		https://m.site.naver.com/lpmt9	코딩 텍스트
02	5		https://m.site.naver.com/lpmxA	코딩 텍스트
03	9		https://m.site.naver.com/lpmAK	코딩 텍스트 이미지 03. 국가별 수도 맞추기 코딩 자료.zip
04	13		https://m.site.naver.com/lpnc6	코딩 텍스트
05	18		https://m.site.naver.com/lpnk9	코딩 텍스트 이미지, 소리 파일 05. 축구 코딩 자료(이미지 소리).zip

크리스마스카드 그리기

01

•학년: ____ •반: ____ •번: ____ •이름: _____

단 원 (4) 프로그래밍

성취 기준

[12정보04-01] 텍스트 기반 프로그래밍 언어의 개발 환경 및 특성을 이해한다.

[12정보04-04] 표준 입출력과 파일 입출력을 활용한 프로그램을 작성한다.

[12정보04-05] 순차, 선택, 반복 구조를 활용한 프로그램을 작성한다.

[12정보04-08] 함수의 개념을 이해하고 함수를 활용한 프로그램을 작성한다.

개 요

- ① 프로그래밍 언어: 파이썬
- ② 개발 목적: 터틀 모듈을 이용하여 그림을 그릴 수 있다.
- ③ 사용 모듈: turtle
- ④ 기능 및 내용: 터틀을 이용한 크리스마스카드 만들기

프로그래밍

```

1 import turtle as t
2 # 원을 그리는 함수
3 def draw_circle(x, y, color, r):
4     t.penup()
5     t.goto(x, y)
6     t.pendown()
7     t.color(color)
8     t.begin_fill()
9     t.circle(r)
10    t.end_fill()
11 # 삼각형을 그리는 함수
12 def draw_triangle():
13    t.penup()
14    t.goto(-50, 0)
15    t.pendown()
16    t.color('green')
17    t.begin_fill()
18    for i in range(3):
19        t.forward(100)
    
```



코드 보기

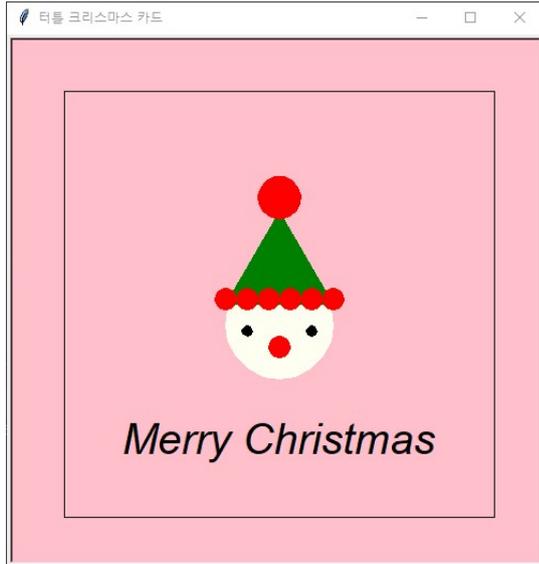
```

20         t.left(120)
21         t.end_fill()
22     # 그래픽 창 설정하기
23     t.setup(500, 500)           ①
24     t.bgcolor('pink')         ②
25     t.title('터틀 크리스마스카드') ③
26     t.speed(0)
27     t.ht()
28     # 테두리 그리기
29     t.penup()
30     t.goto(-200, 200)
31     t.pendown()
32     for i in range(4):
33         t.forward(400)
34         t.right(90)
35     # 얼굴 그리기
36     draw_circle(0, -70, 'ivory', 50)
37     # 삼각형 모자 그리기
38     draw_triangle()
39     # 모자 상단 장식
40     draw_circle(0, 80, 'red', 20)
41     # 모자 하단 장식           ④
42     for i in range(6):
43         draw_circle(-50 + 20 * i, -5, 'red', 10)
44         draw_circle(-30, -30, 'black', 5)       ⑤
45         draw_circle(30, -30, 'black', 5)
46         draw_circle(0, -50, 'red', 10)         ⑥
47     # 메리 크리스마스 글자 쓰기
48     t.penup()
49     t.goto(0, -150)
50     t.color('black')
51     t.write('Merry Christmas', False, 'center', ('나눔바른펜', 30, 'italic'))
52     # 그래픽 작업 완료
53     t.done()

```

- ① 창의 크기를 설정한다.
- ② 창의 배경색을 설정한다.
- ③ 창의 제목 설정한다.
- ④ 반복적으로 작은 원을 그려 모자에 디테일 추가한다.
- ⑤ 눈 그리기, 캐릭터의 표정을 그린다.
- ⑥ 코 그리기

실행 및 결과



자기 평가

평가 항목	😊	😐	😞
1. import ~ as의 역할을 설명할 수 있다.			
2. turtle 모듈을 사용하여 그래픽 작업을 수행하기 위해 필요한 초기 설정은 무엇이며, 어떻게 구현했는지 설명할 수 있다.			
3. draw_circle 함수의 목적과 어떻게 원을 그리는 작업을 수행하는지 설명할 수 있다.			
4. 코드 내에서 for 반복문을 사용한 목적은 무엇인지 설명할 수 있다.			
5. penup()과 pendown() 명령어의 역할을 설명할 수 있다.			

터틀 잡기 게임 만들기

02

•학년: ___ •반: ___ •번: ___ •이름: _____

단 원 (4) 프로그래밍

성취 기준

[12정보04-01] 텍스트 기반 프로그래밍 언어의 개발 환경 및 특성을 이해한다.

[12정보04-04] 표준 입출력과 파일 입출력을 활용한 프로그램을 작성한다.

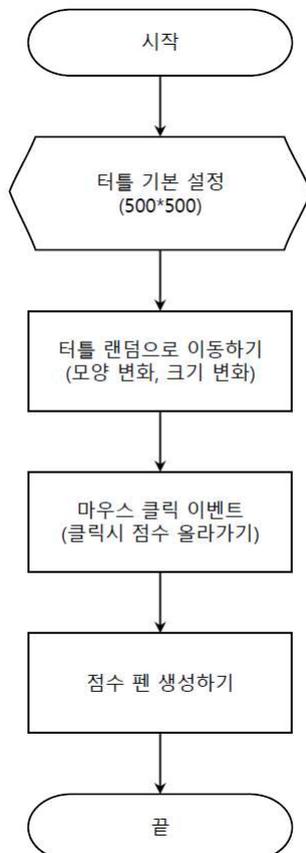
[12정보04-05] 순차, 선택, 반복 구조를 활용한 프로그램을 작성한다.

[12정보04-08] 함수의 개념을 이해하고 함수를 활용한 프로그램을 작성한다.

개 요

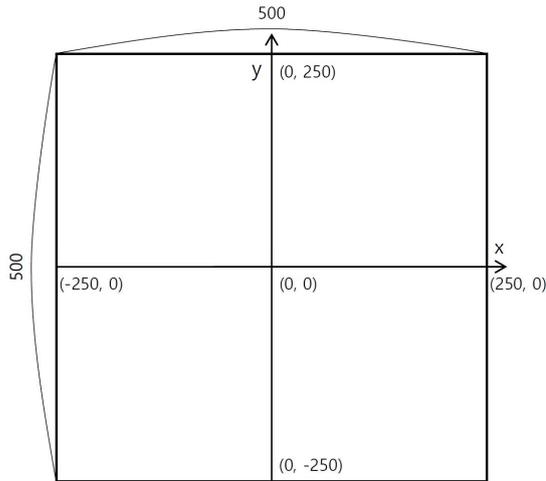
- ① 프로그래밍 언어: 파이썬
- ② 개발 목적: 터틀 모듈을 이용하여 게임을 만들 수 있다.
- ③ 사용 모듈: turtle, random
- ④ 내용 설명: 화면에 나타나는 터틀을 잡으면 점수가 올라가는 게임이다.

순 서 도



프로그래밍 분석

- ① 터틀 좌푯값: 전체를 (500,500)으로 설정하였으므로 좌푯값은 그림과 같다.



- ② turtle.ontimer(함수명, 밀리초), 1초=1000밀리 초
③ turtle.onclick(함수), 클릭한 좌푯값이 전달될 2개의 매개 변수가 필요하다.
→ def catch(x, y)

프로그래밍

```
1 import turtle as t
2 import random
3
4 # 터틀 기본 설정
5 t.setup(500, 500)
6 t.title('터틀 잡기 게임')
7 t.bgcolor('darkblue')
8 t.penup()
9 t.speed(0)
10 t.shape('turtle')
11 t.color('yellow')
12
13 # 클릭 시 점수 올리기
14 def catch(x, y): # 클릭한 좌푯값이 전달될 2개의 매개 변수
15     global score
16     score += 1
17     t.ht() # 클릭해서 캐치가 되면 터틀 숨기기, 두 번 캐치 방지하기 위함이다.
18     pen.clear() # 기존 글자 지우기
19     pen.write(f'score: {score}', False, 'center', ('나눔바른펜', 15))
20
21 # 터틀의 모양 리스트
22 shape_list = ['turtle', 'triangle', 'square', 'arrow', 'circle']
```



코드 보기

- ① 500x500 픽셀 크기의 창을 생성한다.
② 배경색을 어두운 파란색으로 설정한다.
③ 터틀을 클릭했을 때 호출되는 함수로 클릭된 좌표를 매개 변수로 받아 터틀을 숨기고 점수를 1점 증가시킨다. 점수가 증가할 때마다 점수 표시를 업데이트한다.

```

23 score = 0
24
25 def turtle_move(): # 랜덤으로 위치 이동하기
26     t.shape(random.choice(shape_list)) # 터틀의 모양 랜덤 추출하기
27     t.goto(random.randint(-230, 230), random.randint(-230, 230))
28     t.st() # 터틀 보이기
29     for i in range(1, 40): # 터틀 크기 조절하기, 원래 크기 1부터 40까지
30         t.shapesize(i / 40)
31     for i in range(40, 1, -1): # 원래 크기 1부터 1/40까지
32         t.shapesize(i / 40)
33     t.ht() # 터틀 숨기기
34     t.ontimer(turtle_move, 1000) # 1초 간격으로 함수 호출
35
36 # 점수 펜 생성하기
37 pen = t.Turtle() # 새로운 터틀 객체 생성
38 pen.penup()
39 pen.color('white')
40 pen.goto(0, 220)
41 pen.ht()
42 pen.write(f'score: {score}', False, 'center', ('나눔바른펜', 15))
43 #변숫값 출력은 중괄호 안에 표현함.
44
45 t.onclick(catch) #마우스 클릭 이벤트 ④
46 turtle_move()
47
48 t.done() ⑤

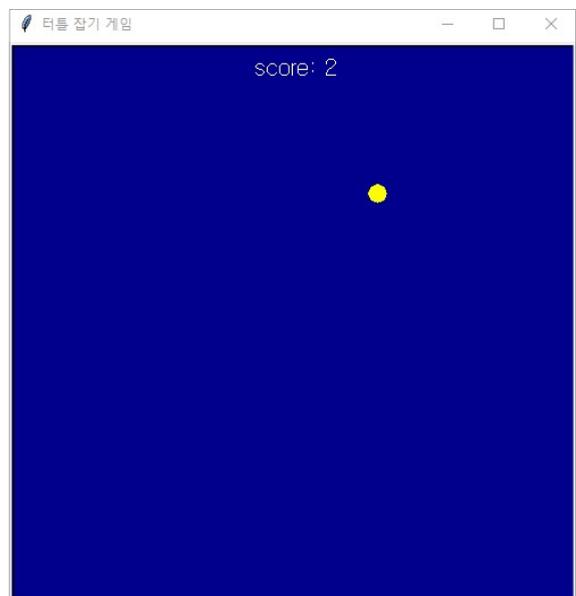
```

④ 마우스를 클릭 이벤트를 catch 함수에 연결하여 사용자가 터틀을 클릭할 때마다 점수가 올라가도록 설정한다.

⑤ 터틀 그래픽 창이 사용자 입력을 계속 기다리게 한다.

실행 및 결과

- ① 메뉴 [Run] → [Run Module] 또는 [F5] 키를 눌러 실행한다.
- ② 터틀 화면에서 무작위로 나오는 터틀 모양을 클릭하면 점수가 올라간다.



자기 평가

평가 항목	☺	☹	☹
1. turtle_move 함수에서 터틀의 위치를 랜덤하게 변경하는 방법을 설명할 수 있다.			
2. 게임의 점수 시스템을 구현하는 방법을 설명할 수 있다.			
3. 프로그래밍에서 발견할 수 있는 반복되는 요소를 찾을 수 있다.			
4. 게임에 추가할 수 있는 다른 터틀 모양이나 색상을 변경할 수 있다.			
5. 현재 게임에 다른 난이도 레벨을 추가하는 방법을 설명할 수 있다.			
6. 터틀을 클릭했을 때 점수가 증가하지 않는 문제가 발생한다면 해결 방법을 설명할 수 있다.			

단 원 (4) 프로그래밍

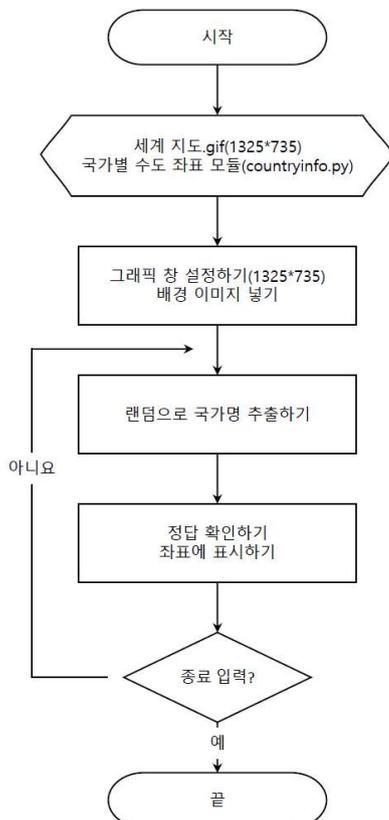
성취 기준

- [12정보04-01] 텍스트 기반 프로그래밍 언어의 개발 환경 및 특성을 이해한다.
- [12정보04-04] 표준 입출력과 파일 입출력을 활용한 프로그램을 작성한다.
- [12정보04-05] 순차, 선택, 반복 구조를 활용한 프로그램을 작성한다.
- [12정보04-08] 함수의 개념을 이해하고 함수를 활용한 프로그램을 작성한다.

개 요

- ① 프로그래밍 언어: 파이썬
- ② 개발 목적: 터틀 모듈을 이용하여 게임을 만들 수 있다.
- ③ 사용 모듈: turtle, random, countryinfo(자체 제공)
- ④ 기능 및 내용: 각 나라의 수도 이름을 물어 정답이면 녹색 글씨로 세계 지도에 표시하고, 오답이면 빨간색으로 지도에 표시한다. "종료"를 입력하면 실행 종료된다.

순 서 도



프로그래밍 분석

- ① 세계 지도 이미지의 크기는 1325*735이다.
- ② country_data.py 파일에는 각 나라 좌표가 있고, 딕셔너리 형태로 "국가", "수도", "좌표"가 저장되어 있다. 또한 수도는 리스트로 좌표는 튜플로 저장되어 있다. 수도를 여러 형태로 불러올 수 있기 때문에 리스트로 저장하여 담아둔다.
(예) 베이징, 북경
- ③ from turtle import 가져올 대상:
→ from을 사용해서 불러오면 "가져올 대상"을 사용할 수 있다.
(예) from countryinfo import country_data
 print(country_data[0]["국가"]) # 국가명(대한민국) 출력
 print(country_data[0]["좌표"]) # 좌표값(427, 65) 출력
- ④ 배경 이미지 넣기와 삭제
→ 넣기: turtle.bgpic("파일명")
→ 삭제: turtle.bgpic("nopic")
- ⑤ 입력 대화 상자
→ 입력값이 문자일 경우: turtle.textinput("제목", "설명")
→ 입력값이 숫자일 경우: turtle.numinput("제목", "설명")
(예) if user_input in cur_quiz["수도"]:
 print("정답")
else:
 print("오답")

프로그래밍

```
1 import turtle as t
2 from countryinfo import country_data # 모듈화한 파일 불러오기
3 import random
4
5 # 그래픽 창 설정하기
6 t.setup(1325, 735) # 맵의 크기와 동일하게 설정한다.
7 t.title('수도 맞추기 게임(종료를 입력하면 프로그램이 종료됩니다.)')
8 t.bgpic('world-map.gif') # 배경 이미지 넣기(gif 파일만 지원한다.)
9
10 # 터틀 속성 설정하기
11 t.penup()
12 t.speed(0) # 스피는 최대로 한다.
13 t.shape('circle')
14 t.shapesize(0.5)
15 t.color('green')
16 t.goto(700, 350) # 초기 터틀 화면에 안보이도록 설정한다.
```



코드 보기

```

17
18 while True:
19     cur_quiz = random.choice(country_data) # 랜덤으로 country_data 1행씩 저장한
        다.
20     user_input = t.textinput('국가별 수도 맞추기', f'{cur_quiz['국가']}'의 수도
        는?')
21
22     if user_input == '종료': # 종료하기
23         break
24
25     elif user_input in cur_quiz['수도']: # 정답 확인하기
26         t.color('green')
27     else:
28         t.color('red')
29
30     t.goto(cur_quiz['좌표'])
31     t.clear()
32
33 # 리스트에서 0번째 인덱스만 가져오고, 터틀 움직임은 false, 가운데 정렬 및 폰
        트 설정하기
34     t.write(f'{cur_quiz['국가']}\n{cur_quiz['수도'][0]}', False, 'center', \ ('
        나눔바른펜', 20, 'bold'))
35
36 # t.done()이 있으면 대화창을 닫으면 배경 창이 계속 유지된다.

```

실행 및 결과

메뉴 [Run] → [Run Module] 또는 [F 5] 키를 눌러 실행한다.



자기 평가

평가 항목	😊	😐	😞
1. countryinfo 모듈을 성공적으로 설치하고 코드에서 어떻게 사용하였는지 설명할 수 있다.			
2. 무한 루프를 왜 사용하였으며, 어떻게 종료되게 설계하였는지 설명할 수 있다.			
3. 사용자의 입력을 처리하기 위해 어떤 조건문을 사용하였는지 설명할 수 있다.			
4. country_data의 데이터 구조는 어떻게 되어 있는지 설명할 수 있다.			

터틀 화살 쏘기 게임 만들기

04

•학년: ____ •반: ____ •번: ____ •이름: _____

단 원 (4) 프로그래밍

성취 기준

[12정보04-01] 텍스트 기반 프로그래밍 언어의 개발 환경 및 특성을 이해한다.

[12정보04-04] 표준 입출력과 파일 입출력을 활용한 프로그램을 작성한다.

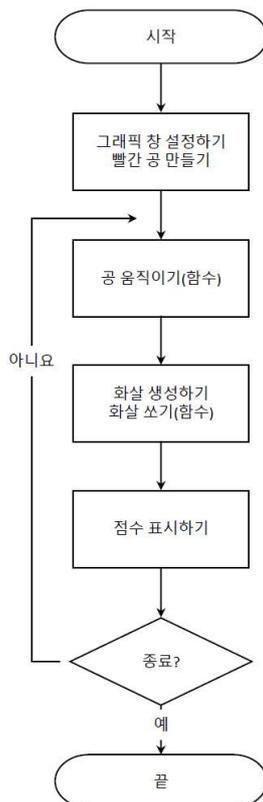
[12정보04-05] 순차, 선택, 반복 구조를 활용한 프로그램을 작성한다.

[12정보04-08] 함수의 개념을 이해하고 함수를 활용한 프로그램을 작성한다.

개 요

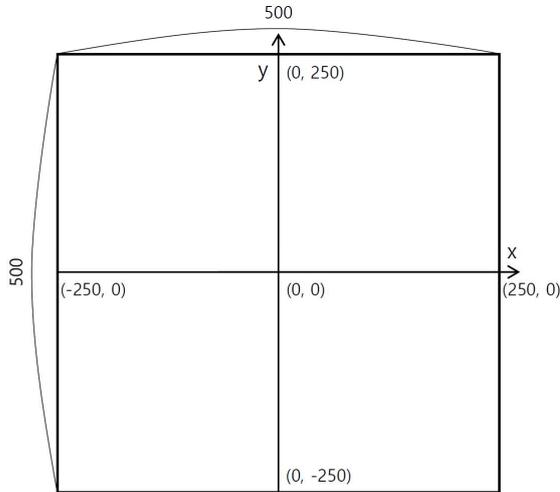
- ① 프로그래밍 언어: 파이썬
- ② 개발 목적: 터틀 모듈을 이용하여 게임을 만들 수 있다.
- ③ 사용 모듈: turtle, random
- ④ 내용 설명: 상단에 빨간색 공이 무작위 속도로 지나가고, 그래픽 창 하단에 터틀 화살이 있다. 마우스를 클릭해서 화살을 쏘아 화살이 빨간색 공에 적중하면 빨간색 공이 노란색으로 변하고 사라진다. 그리고 왼쪽 상단에 점수가 표시된다.

순 서 도

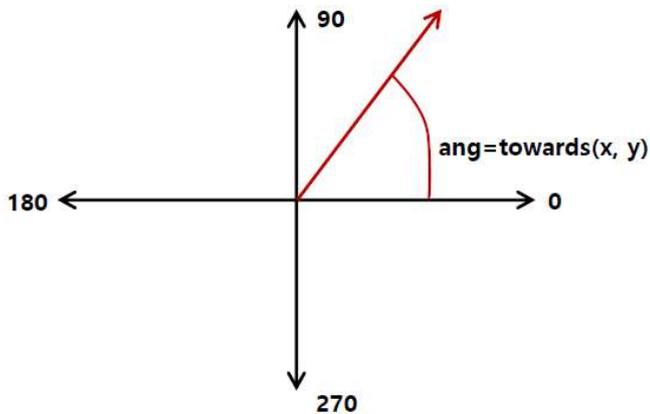


프로그래밍 분석

- ① Hex 색상 코드: RGB 색상을 나타내는 16진수 색상 코드이다.
예) `turtle.color("#00ff00")`
- ② 터틀 방향 설정



- ③ `ontimer(함수명, 밀리초)`: 특정 함수를 시간을 간격을 두고 호출할 때 사용한다.
→ 1초=1000밀리 초
- ④ 방향각 찾기 `setheading()` 함수: `x, y` 좌표값이 있으면 각도를 얻을 수 있다. 현재 터틀 위치에서 `(x, y)`로 향하는 각도를 반환한다.



프로그래밍

```
1 import turtle as t
2 import random
3
4 # 그래픽 창 설정하기
5 t.setup(500, 500)
6 t.bgcolor('#FFAB76') # 배경색 지정하기
7 t.title('터틀 화살 쏘기 게임')
8
9 # 빨간색 공 생성하기
```



코드 보기

```

10 ball = t.Turtle()
11 ball.penup()
12 ball.speed(0)
13 ball.shape('circle')
14 ball.color('red')
15 ball.goto(300, 200) # 시작 x 좌표를 화면 밖으로 설정하기
16 ball.setheading(180) # 공의 방향 설정하기
17
18 # 변수 초기화
19 score = 0
20 ready = True # 화살을 쏘기 위한 준비가 되어 있는지 확인하는 변수이다.
21
22 # 공 움직이기
23 def ball_move():
24     if ball.xcor() > -270: # 터틀의 x 좌표값이 -270보다 크면
25         ball.forward(random.randint(5, 20)) # 5~20 사이로 속도로 변화를 주면서 이동한다.
26     else:
27         ball.goto(300, 200) # 처음 위치로 이동한다.
28     t.ontimer(ball_move, 20) # ball_move 함수를 0.02초 간격으로 실행한다.
29
30 # 화살 생성
31 player = t.Turtle()
32 player.shapesize(1, 2) # 화살의 크기 키우기(너비, 높이), 기본값 1로 한다.
33 player.speed(0) #최고 속도
34 player.penup()
35 player.goto(0, -220)
36 player.setheading(90) # 화살 방향 설정하기
37
38 # 마우스 클릭 시 화살 쏘기
39 t.onscreenclick(fire)
40
41 # 공이 움직이기 시작한다.
42 ball_move()
43
44 # 화살 쏘기
45 def fire(x, y):
46     global score, ready
47
48     if ready: # ready가 없으면 여러 번 클릭하면 여러 발 발사된다.
49         ready = False

```

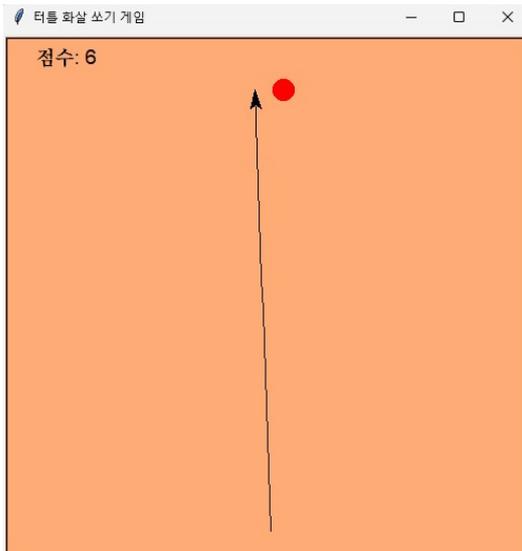
```

50     player.pendown()
51
52     # 클릭한 지점 바라보기
53     ang = player.towards(x, y) # x,y 좌표값이 있으면 각도를 얻을 수 있다.
54     player.setheading(ang)
55
56     # 화살 발사하기
57     while player.ycor() < 200: # y 좌표가 200보다 작으면
58         player.forward(30)    # 간격 30씩 이동한다.
59
60         # 공에 닿았는지 확인하기
61         if player.distance(ball) < 30:
62             ball.color('yellow')
63             score += 1
64             t.clear()
65             t.write(f'점수: {score}')
66             ball.goto(300, 200)
67             ball.color('red')
68             break
69
70     # 화살을 원상으로 복귀하기
71     player.clear()
72     player.penup()
73     player.setheading(90) # 터틀 방향 설정하기
74     player.goto(0, -220) # 원상 복귀 좌표값
75     ready = True
76
77 # 점수 표시하기
78 t.ht()
79 t.penup()
80 t.goto(-220, 220)
81 t.write(f'점수: {score}')

```

실행 및 결과

메뉴 [Run] → [Run Module] 또는 [F5] 키를 눌러 실행한다.



자기 평가

평가 항목	😊	😐	😞
1. 터틀 그래픽 창의 크기와 배경색을 설정하는 방법을 설명할 수 있다.			
2. 터틀 객체를 생성하고, 속성(색상, 모양, 초기 위치 등)을 설정하는 방법을 설명할 수 있다.			
3. 필요한 기능(공 움직이기, 화살 쏘기)을 함수로 정의할 수 있다.			
4. 게임 로직에서 조건문을 사용하여 다양한 상황(예: 화살이 공에 닿았을 때)을 처리할 수 있다.			
5. 화살이 목표물에 도달할 때까지 반복 이동하는 로직을 구현할 수 있다.			

축구 게임 만들기

05

•학년: ___ •반: ___ •번: ___ •이름: _____

단 원 (4) 프로그래밍

성취 기준

[12정보04-01] 텍스트 기반 프로그래밍 언어의 개발 환경 및 특성을 이해한다.

[12정보04-04] 표준 입출력과 파일 입출력을 활용한 프로그램을 작성한다.

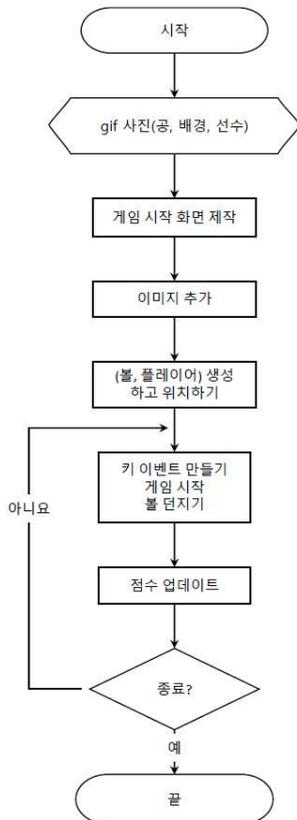
[12정보04-05] 순차, 선택, 반복 구조를 활용한 프로그램을 작성한다.

[12정보04-08] 함수의 개념을 이해하고 함수를 활용한 프로그램을 작성한다.

개 요

- ① 프로그래밍 언어: 파이썬
- ② 개발 목적: 터틀에서 캐릭터 이미지와 소리를 삽입하여 게임을 만들 수 있다.
- ③ 사용 모듈: turtle, random, winsound
- ④ 내용 설명: 방향 키를 조절하여 무작위로 날아오는 공을 골키퍼를 이동시키며 막으면 점수가 올라가고, 못 막으면 상대방 점수가 올라간다.

순서도



프로그래밍 분석

- ① 준비: 그림판을 이용하여 그릴 수 있다.
 - 이미지- ball.gif(공), bg.gif(배경), player.gif(선수)
 - 소리- fall.wav(실패), success.wav(성공)
- ② addshape("파일명"): 이미지 등록, 이미지는 gif만 가능, 회전을 지원하지 않는다. shape 리스트에 이미지를 추가한다.
- ③ t.ontimer(throw_ball,500): 0.5초 간격으로 throw_ball 함수를 실행한다.
- ④ 윈도우 소리 재생 방법: winsound 모듈
 - winsound.PlaySound('파일경로', winsound.SND_ASYNC)
 - wav 파일
 - winsound.SND_ASYNC: 비동기적으로 소리 재생(소리 재생과 다른 작업 동시 수행)

프로그래밍

```
1 import turtle as t
2 import random
3
4 # OS에 따른 사운드 재생 설정
5 import platform
6 if platform.system() == "Windows":
7     import winsound
8 else:
9     import os
10
11 # 키보드 방향 키로 플레이어를 좌우로 이동하기
12 def move_right():
13     if player.xcor() < 100 and ready:
14         player.forward(25)
15
16 def move_left():
17     if player.xcor() > -100 and ready:
18         player.backward(25)
19
20 # 공 던지기
21 def throw_ball():
22     global my_score, com_score, ready
23     ball.setheading(random.randint(80, 100))
24     while ball.ycor() < 240:
25         ball.forward(13)
26     ready = False
27
```



코드 보기

```

28     if ball.distance(player) < 30:
29         my_score += 1
30         play_sound('success.wav')
31     else:
32         com_score += 1
33         play_sound('fail.wav')
34
35     score_update()
36     t.ontimer(game_start, 1000)
37
38 # 사운드 재생
39 def play_sound(sound_file):
40     if platform.system() == "Windows":
41         winsound.PlaySound(sound_file, winsound.SND_ASYNC)
42     elif platform.system() == "Darwin": # macOS
43         os.system(f'afplay {sound_file}&')
44     else: # Linux
45         os.system(f'aplay {sound_file}&')
46
47 # 게임 시작
48 def game_start():
49     global ready
50     ready = True
51     t.clear()
52     ball.goto(0, -270)
53     t.ontimer(throw_ball, random.randint(500, 2000))
54
55 # 변수 초기화
56 ready = False
57 my_score = 0
58 com_score = 0
59
60 # 게임 화면 설정
61 t.setup(380, 600)
62 t.title('터틀 골기퍼')
63 t.bgpic('images/bg.gif')
64 t.penup()
65 t.ht()
66 t.speed(0)
67 t.color('white')
68 t.write('축구 게임', False, 'center', ('나눔바른펜', 30, 'bold'))

```

```

69 t.goto(0, -50)
70 t.write('<< 시작: 스페이스 키 >>', False, 'center', ('나눔바른펜', 20,
    'bold'))
71
72 # 이미지 로드
73 ball_img = 'images/ball.gif'
74 player_img = 'images/player.gif'
75 t.addshape(ball_img)
76 t.addshape(player_img)
77
78 # 플레이어 및 볼 생성
79 player = t.Turtle()
80 player.penup()
81 player.speed(0)
82 player.goto(0, 230)
83 player.shape(player_img)
84
85 ball = t.Turtle()
86 ball.penup()
87 ball.speed(0)
88 ball.goto(0, -270)
89 ball.shape(ball_img)
90
91 # 키 이벤트 설정
92 t.onkeypress(game_start, 'space')
93 t.onkeypress(move_right, 'Right')
94 t.onkeypress(move_left, 'Left')
95 t.listen()
96
97 # 점수 업데이트 함수
98 def score_update():
99     t.clear()
100     t.write(f'{my_score} : {com_score}', False, 'center', ('나눔바른펜', 24,
    'bold'))
101
102 t.done()

```

실행 및 결과

메뉴 [Run] → [Run Module] 또는 [F 5] 키를 눌러 실행한다.



자기 평가

평가 항목	😊	☹️	😞
1. 게임 화면 크기와 배경 이미지를 설정하는 방법을 설명할 수 있다.			
2. 키보드 이벤트를 처리하는 방법을 이해하고 설명할 수 있다.			
3. 플레이어가 이동할 수 있는 범위를 설정하는 데 조건문을 어떻게 사용했는지 설명할 수 있다.			
4. 공을 던지는 데 사용된 반복문의 목적과 작동 방식을 설명할 수 있다.			
5. 게임 코드의 성능이나 기능을 향상시킬 수 있는 방법을 제안할 수 있다.			