

공학적 문제해결 기법

14주차 2차시

움직이는 그림 그리기





14주차 2차시 : 움직이는 그림 그리기

I. 개요

기본적인 애니메이션 만들기

II. 학습 개요

1) 학습 목표

기본적인 애니메이션을 하기 위해 tkinter 모듈을 사용하고, 어떤 키를 눌렀을 때 반응하도록 이벤트 바인딩을 사용한다. 또한 그렸던 도형을 이동한다거나 색상을 바꿔 보고, 움직이는 공도 만들어 본다.

2) 학습 목차(세부 목차)

- 기본적인 애니메이션 만들기
- 객체가 어떤 것에 대해 반응하도록 만들기
- 공을 만들어 움직여 보기



14주차 2차시 : 움직이는 그림 그리기

III. 학습 1 - 기본적인 애니메이션 만들기

화면 위에서 움직이기

- ✓ 움직이는 애니메이션을 만들려면 어떻게 해야 할까요?
- ✓ tkinter 모듈이 애니메이션을 전문으로 하지 않지만 기본적인 것들을 처리 가능
- ✓ 다음 코드는 색이 채워진 삼각형을 만들고, 화면 위에서 움직일 것이다.



```
import time
from tkinter import *
tk = Tk()
canvas = Canvas(tk, width=400, height=200)
canvas.pack()
canvas.create_polygon(10, 10, 10, 60, 50, 35)
for x in range(0, 60):
    canvas.move(1, 5, 0)
    tk.update()
    time.sleep(0.05)
```



14주차 2차시 : 움직이는 그림 그리기

III. 학습 1 - 기본적인 애니메이션 만들기

화면 위에서 움직이기

- ✓ 코드를 살펴보면,
- ✓ `canvas.create_polygon(10, 10, 10, 60, 50, 35)` 코드로 삼각형을 만들고
- ✓ `canvas.move(1, 5, 0)`는 ID가 1 (삼각형의 식별자가)인 객체를 가로로 5픽셀만큼, 세로로 0픽셀만큼 움직임
- ✓ 다시 돌아가려면 `canvas.move(1, -5, 0)` 라고 함수를 호출
- ✓ `tk.update()` 함수는 tkinter가 화면을 다시 그리도록 함
- ✓ `update`를 사용하지 않으면 삼각형은 움직이지 않고, tkinter는 루프가 끝날 때까지 기다리게 됨
- ✓ 이 말은 삼각형이 캔버스 위를 부드럽게 움직이는 것이 아니라, 마지막 위치로 점프하게 되는 것을 보게 됨
- ✓ `time.sleep(0.05)`는 계속 진행하기 전에 0.05초 잠들라고 함



14주차 2차시 : 움직이는 그림 그리기

III. 학습 1 - 기본적인 애니메이션 만들기

대각선 방향으로 삼각형 움직이기

 move(1,5,5)를 호출하도록 수정



```
import time
from tkinter import *
tk = Tk()
canvas = Canvas(tk, width=400, height=400)
canvas.pack()
canvas.create_polygon(10, 10, 10, 60, 50, 35)
for x in range(0, 60):
    canvas.move(1, 5, 5)
    tk.update()
    time.sleep(0.05)
```

 다시 시작 위치로 대각선 방향에 따라 움직이도록 하려면

 canvas.move(1, -5, -5) 사용



14주차 2차시 : 움직이는 그림 그리기

III. 학습 2 - 객체가 어떤 것에 대해 반응하도록 만들기

이벤트 바인딩(event binding)

- ✓ 키보드를 눌렀을 때 삼각형이 반응하도록 만들 수 있다.
- ✓ 이벤트는 프로그램을 실행하는 중에 발생하는 일
- ✓ 마우스를 움직였다거나 키보드를 눌렀다거나 또는 창을 닫았다거나 하는 것
- ✓ 바인딩은 tkinter에 어떤 함수가 특정 이벤트와 연결되었다는 것을 알려주는 것
- ✓ 다시 말해, 이벤트를 처리하도록 tkinter에 의해 자동으로 호출됨
- ✓ ENTER 키를 눌렀을 때 삼각형을 움직이도록 하기 위해서 다음의 함수를 정의



```
def movetriangle(event):
```

```
    canvas.move(1,5,0)
```

- ✓ 이 함수는 단 하나의 매개변수(event)를 받는다.
- ✓ tkinter가 그 이벤트에 대한 정보를 함수에 전달하기 위해 사용



14주차 2차시 : 움직이는 그림 그리기

III. 학습 2 - 객체가 어떤 것에 대해 반응하도록 만들기

이벤트 바인딩(event binding)

- ☑ 캔버스의bind_all 함수를 사용하여tkinter에게이 함수가특정 이벤트에대해사용될것이고알려
줌



```
from tkinter import *  
tk = Tk()  
canvas = Canvas(tk, width=400, height=400)  
canvas.pack()  
canvas.create_polygon(10, 10, 10, 60, 50, 35)  
def movetriangle(event):  
    canvas.move(1, 5, 0)  
    canvas.bind_all('<KeyPress-Return>', movetriangle)
```

- ☑ tkinter가감시해야할 이벤트 <KeyPress-Return> 이라는것
- ☑ 이것은 ENTER 또는 RETURN 키를 누르는것
- ☑ KeyPress 이벤트가 발생할때마다 movetriangle 함수가 호출



14주차 2차시 : 움직이는 그림 그리기

III. 학습 2 - 객체가 어떤 것에 대해 반응하도록 만들기

이벤트 바인딩(event binding)

☒ 화살표키를 눌러서 삼각형의 방향을 바꾸려면 어떻게 해야 할까요?



```
def movetriangle(event):  
    if event.keysym == 'Up':  
        canvas.move(1,0,-3)  
    elif event.keysym == 'Down':  
        canvas.move(1,0,3)  
    elif event.keysym == 'Left':  
        canvas.move(1,-3,0)  
    else:  
        canvas.move(1,3,0)
```

☒ keysym(키 심볼)은 눌러진 실제 키의 값을 가진 문자열

☒ 매개변수 (1,0,-3)에서 1은 캔버스에 그려진 도형의 식별 번호이고, 두 번째, 세 번째 매개변수 인 0과 -3은 각각 x좌표와 y좌표에 더할 값



14주차 2차시 : 움직이는 그림 그리기

III. 학습 2 - 객체가 어떤 것에 대해 반응하도록 만들기

이벤트 바인딩(event binding)

 앞서 배운 코드를 정리하면, 삼각형은 방향키를 누르는 것에 따라 움직이게 됨

```
from tkinter import *
tk = Tk()
canvas = Canvas(tk, width=400, height=400)
canvas.pack()
canvas.create_polygon(10, 10, 10, 60, 50, 35)
def movetriangle(event):
    if event.keysym == 'Up':
        canvas.move(1, 0, -3)
    elif event.keysym == 'Down':
        canvas.move(1, 0, 3)
    elif event.keysym == 'Left':
        canvas.move(1, -3, 0)
    else:
        canvas.move(1, 3, 0)
canvas.bind_all('<KeyPress-Up>', movetriangle)
canvas.bind_all('<KeyPress-Down>', movetriangle)
canvas.bind_all('<KeyPress-Left>', movetriangle)
canvas.bind_all('<KeyPress-Right>', movetriangle)
```



14주차 2차시 : 움직이는 그림 그리기

III. 학습 3 - 공을 만들어 움직여보기

게임 캔버스 생성하기

- ✓ 공을 만들기 위한 캔버스를 생성
- ✓ 쉘에서 새로운 파일 열기, 파일명은 paddleball.py
- ✓ 이전 예제들과 약간 다를 것이다.

```
from tkinter import *  
import random  
import time  
tk = Tk()  
tk.title("Game")  
tk.resizable(0, 0)  
tk.wm_attributes("-topmost", 1)  
canvas = Canvas(tk, width=500, height=400, bd=0,  
highlightthickness=0)  
canvas.pack()  
tk.update()
```

- ✓ tkinter가 그 이벤트에 대한 정보를 함수에 전달하기 위해 사용



14주차 2차시 : 움직이는 그림 그리기

III. 학습 3 - 공을 만들어 움직여보기

게임 캔버스 생성하기

- ☒ 다른 모든 창들 앞에 캔버스를 가진 창이 위치하도록 ("-topmost")하기 위해 위해 `wm_attributes`를 호출
- ☒ 매개변수 `bd=0, highlightthickness=0`은 캔버스 외곽에 둘러싼 외곽선 (border)이 없도록 하는 것
- ☒ 이는 게임 화면을 좀더 좋게 보이도록 하기 위함



14주차 2차시 : 움직이는 그림 그리기

III. 학습 3 - 공을 만들어 움직여보기

공에 대한 클래스 생성하기

- ✓ 첫 번째로, 캔버스와 그릴 공의 색깔에 대한 매개변수들을 받는 Ball이라는 이름의 클래스를 생성

```
class Ball:
    def __init__(self, canvas, color):
        self.canvas = canvas
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color)
        self.canvas.move(self.id, 245, 100)

    def draw(self):
        pass
```

- ✓ 왼쪽상단구석에서 시작하는 (10, 10) 좌표와 우측 하단구석에서 시작하는 (25, 25) 좌표, 원을 채울 색깔의 매개변수 다섯 개로 create_oval 함수를 호출
- ✓ create_oval 함수는 그려진 도형에 대한 식별자를 반환하며, 객체 변수 id에 저장
- ✓ 원을 캔버스의 중앙(245, 100 위치)으로 이동
- ✓ draw 함수는 조금 후에 몇몇 코드가 더 추가될 것임



14주차 2차시 : 움직이는 그림 그리기

III. 학습 3 - 공을 만들어 움직여보기

공에 대한 클래스 생성하기

✓ Ball 클래스를 생성했으니, 이제 이 클래스의 객체를 생성

✓ 생성된 객체가 실제로 작업을 할 것이다.

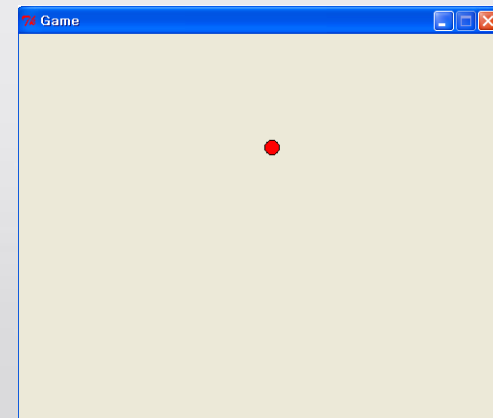
➔ `ball = Ball(canvas, 'red')`

✓ 프로그램을 지금 실행하면 캔버스가 잠깐 나왔다가 사라짐

✓ 창이 즉시 닫히는 것을 막으려면 게임의 메인 루프(main loop)라고 불리는 애니메이션 루프를 추가해야 함

✓ 프로그램의 하단에 다음의 코드를 추가

➔ `while 1:`
`tk.update_idletasks()`
`tk.update()`
`time.sleep(0.01)`





14주차 2차시 : 움직이는 그림 그리기

III. 학습 3 - 공을 만들어 움직여보기

공을 움직이게 만들기

- ✓ 공을 움직이게 하려면 다음과 같이 draw 함수를 수정해야함



```
def draw(self):
```

```
    self.canvas.move(self.id, 0, -1)
```

- ✓ `_init_`은 매개변수 `canvas`를 객체 변수인 `canvas`에 저장하므로 `self.canvas`로 그 변수를 사용
- ✓ `move`에 세 개의 매개변수로 공의 `id`와 숫자 `0, -1`을 전달
- ✓ `0`은 가로로 움직이지 말라는 뜻이고, `-1`은 화면의 위쪽으로 1픽셀 움직
- ✓ 프로그램의 하단에 있는 메인 루프에 공 객체의 `draw` 함수를 호출해야함



```
while 1:
```

```
    ball.draw()
```

```
    tk.update_idletasks()
```

```
    tk.update()
```

```
    time.sleep(0.01)
```




14주차 2차시 : 움직이는 그림 그리기

III. 학습 3 - 공을 만들어 움직여보기

공을 움직이게 만들기

 지금까지 작성한 게임 코드를 한번에 보면

```
74 bounce3.py - C:\WDocuments and Settings\WAdministrator\W바탕 화면\W파이썬W...
File Edit Format Run Options Windows Help

from tkinter import *
import random
import time

tk = Tk()
tk.title("Game")
tk.resizable(0, 0)
tk.wm_attributes("-topmost", 1)
canvas = Canvas(tk, width=500, height=400, bd=0, highlightthickness=0)
canvas.pack()
tk.update()

class Ball:
    def __init__(self, canvas, color):
        self.canvas = canvas
        self.id = canvas.create_oval(10, 10, 25, 25, fill=color)
        self.canvas.move(self.id, 245, 100)

    def draw(self):
        self.canvas.move(self.id, 0, -1)

ball = Ball(canvas, 'red')

while 1:
    ball.draw()
    tk.update_idletasks()
    tk.update()
    time.sleep(0.01)
```



14주차 2차시 : 움직이는 그림 그리기

III. 학습 3 - 공을 만들어 움직여보기

공을 튕기게 만들기

- ✓ 앞서 만든 코드는 공이 화면 상단으로 사라지게 됨
- ✓ 공이 사라지면 안 되니 공을 튕기도록 만들어본다.
- ✓ Ball 클래스의 초기화 함수에 몇 가지 객체 변수를 다음과 같이 저장

```
def __init__(self, canvas, color):  
    self.canvas = canvas  
    self.id = canvas.create_oval(10, 10, 25, 25, fill=color)  
    self.canvas.move(self.id, 245, 100)  
    self.x = 0  
    self.y = -1  
    self.canvas_height = self.canvas.winfo_height()
```

- ✓ 객체 변수 `canvas_height`는 캔버스 함수인 `winfo_height()` 함수를 호출하는데, 이 함수는 캔버스의 현재 높이를 반환



14주차 2차시 : 움직이는 그림 그리기

III. 학습 3 - 공을 만들어 움직여보기

공을 튕기게 만들기

- ✓ draw 함수를 다시 수정
- ✓ 공이 사라지면 안 되니 공을 튕기도록 만들어본다.
- ✓ 공이 사라지면 안 되니 공을 튕기도록 만들어본다.

```
def draw(self):  
    self.canvas.move(self.id, self.x, self.y)  
    pos = self.canvas.coords(self.id)  
    if pos[1] <= 0:  
        self.y = 1  
    if pos[3] >= self.canvas_height:  
        self.y = -1
```

- ✓ 캔버스 함수인 coords를 호출하여 pos라는 변수를 생성
- ✓ 이 함수는 식별 번호로 캔버스에 그려진 현재의 x와 y의 좌표를 반환
- ✓ coords 함수는 네 개의 숫자 리스트로 된 좌표를 반환



14주차 2차시 : 움직이는 그림 그리기

III. 학습 3 - 공을 만들어 움직여보기

공을 튕기게 만들기



```
def movetriangle(event):
```

```
    canvas.move(1,5,0)
```

- ✓ 리스트의 처음 두 개의 숫자(255.0과 109.0)는 원의 왼쪽 상단의 좌표
- ✓ 두 번째 두 개의 숫자(270.0, 124.0)은 우측 하단의 좌표
- ✓ y1 좌표(공의 상단!)가 0과 같거나 작은지를 확인
- ✓ 만약에 이것이 맞는다면, y 객체 변수를 1로 설정 계속 올라가는 움직임을 멈추라고 하는 것
- ✓ y2의 좌표도 같은 이치
- ✓ 이 코드를 실행하면 창을 닫을 때까지 공이 캔버스의 위아래로 튕기게 됨



『 이 콘텐츠는 2014학년도 학부교육 선도대학 육성사업(ACE)에 의하여 개발한 것임 』

