13장 애니메이션과 게임

이 장의 내용

- 13.1 시간 출력하기
- 13.2 아날로그 시계 만들기
- 13.3 Frog 게임 만들기

13.1 시간 출력하기

시간 출력하기

- time 모듈
  - asctime 함수 : 년, 월, 일, 요일, 시, 분, 초를 하나의 문자열 반환

```
import time
for i in range(10):
    print(time.asctime())
    time.sleep(1)
```

실행결과

```
Tue Jun  1 12:09:25 2021
Tue Jun  1 12:09:26 2021
Tue Jun  1 12:09:27 2021
Tue Jun  1 12:09:28 2021
```

시간 출력하기

- >>> time.localtime()
  - time.struct_time(tm_year=2021, tm_mon=6, tm_mday=1, tm_hour=12, tm_min=11, tm_sec=27, tm_wday=1, tm_yday=152, tm_isdst=0)
- >>> for item in time.locatime():
- print(item)
  - 2021 # year
  - 6 # month
  - 1 # date
  - 12 # hour
  - 10 # min
  - 10 # second
  - 1 # Tuesday
  - 152 # year day
  - 0 # summer time

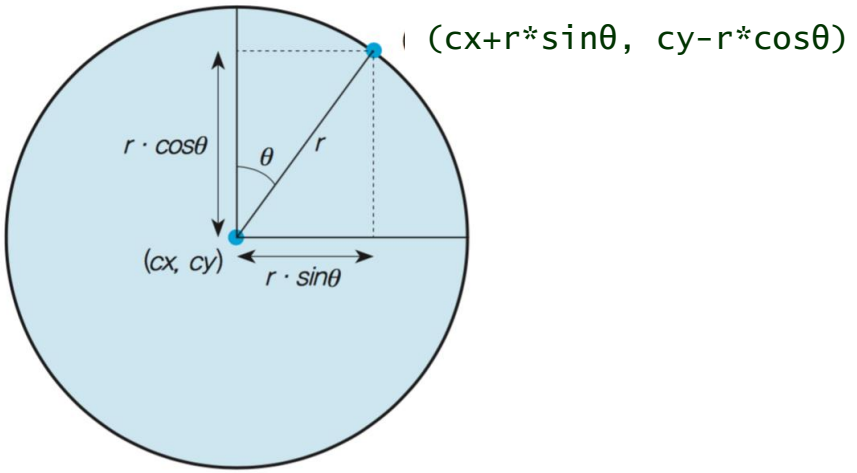
디지털 시계 예제

- canvas.create_text
- 1초마다 화면을 지우고 새롭게 시간을 출력
  - time.sleep(1) , canvas.delete("all")


```
from tkinter import *
import time
tk = Tk()
canvas = Canvas(tk, width=500, height=500)
canvas.pack()
width = 500
height = 500
while True:
    t = time.localtime()
    hour = t[3]
    minute = t[4]
    second = t[5]
    canvas.delete("all")
    myclock = str(hour)+":"+str(minute)+":"+str(second)
    canvas.create_text(250, 250, text=myclock, font=('Arial',25))
    time.sleep(1)
    tk.update()
```

13.2 아날로그 시계 만들기

- 초침 만들기
  - 1초 = 6도(=360/60) , Second = t[5]*6
- math모듈 sin, cos 함수
  - radian = (degree / 360) * 2 * pi
- 초침의 끝 좌표 (sr : 초침의 길이)
  - sx = cx + sr * math.sin(second / 360 * 3.14 * 2)
  - sy = cy - sr * math.cos(second / 360 * 3.14 * 2)
  - Θ 는 시계방향으로 증가



13.2 아날로그 시계 만들기

- 분침, 시침
  - minute = (t[4] + t[5] / 60) * 6 // 1분 = 6도
  - hour = (t[3] + t[4] / 60) * 30 // 1시간 = 30도

  - mx = mr * math.sin(minute / 360 * 3.14 * 2) #Radian
    my = mr * math.cos(minute / 360 * 3.14 * 2)
    canvas.create_line(cx, cy, cx+mx, cy-my, fill='Green', width = 5)

  - hx = hr * math.sin(hour / 360 * 3.14 * 2)
    hy = hr * math.cos(hour / 360 * 3.14 * 2)
    canvas.create_line(cx, cy, cx+hx, cy-hy, fill='Blue', width = 8)

  - canvas.create_line(cx, cy, cx+sx, cy-sy, fill='Red', width = 1)

아날로그 시계 만들기

- 테두리 그리기
  - canvas.create_arc(10, 10, width-10, height-10, extent=359, style = CHORD, width = 2)
- 아날로그 시계 프로그램
  - 프로그램 13.6

13.3 Frog 게임 만들기

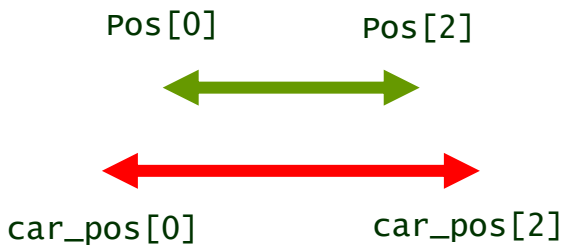
- 개구리 클래스, 자동차 클래스
- 개구리 클래스의 메소드
  - init : Frog 객체 초기화 메소드
  - hit_car : 충돌감지 메소드
  - draw : 화면에 객체를 그리기 위한 메소드
  - move_up, move_ left, move_right : 방향키를 통한 이동 메소드

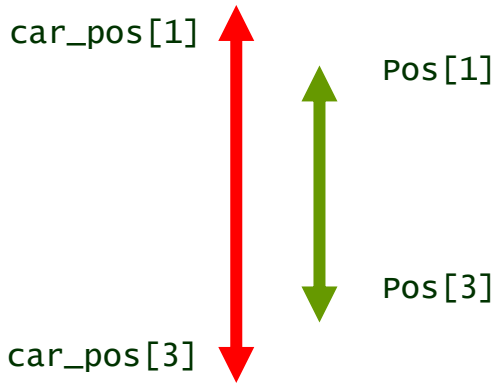
- 방향키 이동
  - self.canvas.bind_all('<KeyPress-Up>', self.move_up)
    self.canvas.bind_all('<KeyPress-Left>', self.move_left)
    self.canvas.bind_all('<KeyPress-Right>', self.move_right)

Frog 게임 만들기

- 충돌감지 (pos : frog)

```
car_pos = self.canvas.coords(self.car1.id)
if pos[2] >= car_pos[0] and pos[0] <= car_pos[2]:     # X
    if pos[3] >= car_pos[1] and pos[1] <= car_pos[3]:     # Y
        return True
return False
```





Frog 게임 만들기

- 자동차 클래스

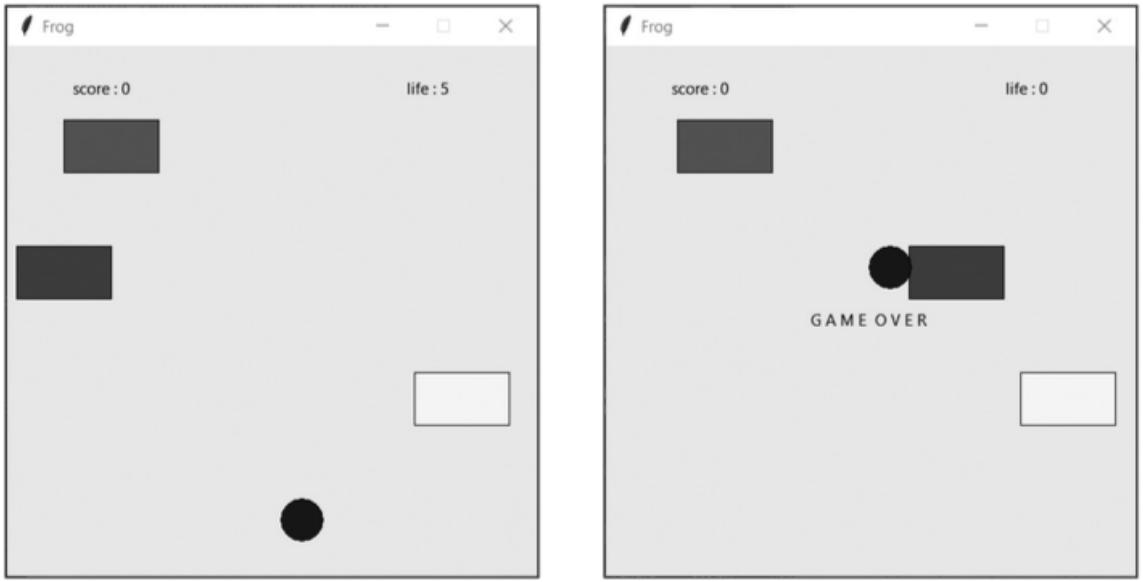
```
car1 = Car(canvas, 10, 60, "red", 2)      # speed and direction
car2 = Car(canvas, 500, 180, "green", -3)
car3 = Car(canvas, 10, 300, "yellow", 1)
```

- 메인루프

```
while True:
    if frog.life >= 0:
        car1.draw()
        car2.draw()
        car3.draw()
        frog.draw()
    #tk.update_idletasks()
    tk.update()
    time.sleep(0.01)
```

Frog 게임 만들기

- 전체 프로그램
  - 프로그램 13.7



Key Point

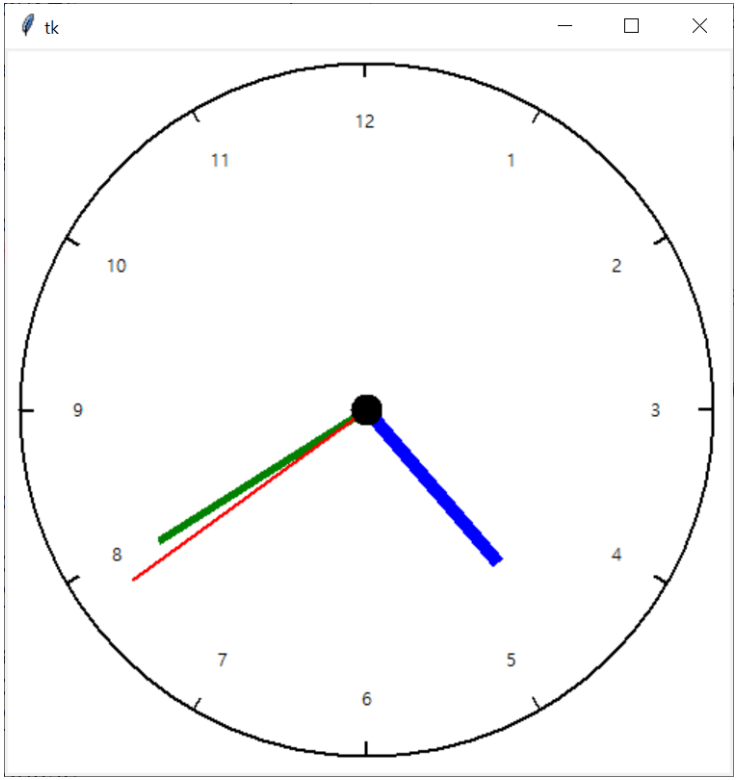
Key Point

- 시계 초침,분침,시침 바늘의 끝좌표는 각 바늘이 회전하는 각도를 라디안 값으로 변환한 후 math.sin 과 math.cos 을 적용하고 바늘의 길이(r)을 곱해준다.

프로그래밍 실습

프로그래밍 실습 1

- 실행결과와 같이 아날로그 시계의 테두리에 5분 단위로 눈금과 시간을 텍스트로 표시하시오.



▶ 프로그래밍 실습 2

- 개구리 게임에 다음 기능을 추가하여 확장하시오.
  <space>바를 눌러서 게임 속도를 2 단계로 조절해 보자.

  - 속도 단계 : normal <-- > fast

```
canvas.bind_all('<space>', change_speed)

def change_speed(evt):
```