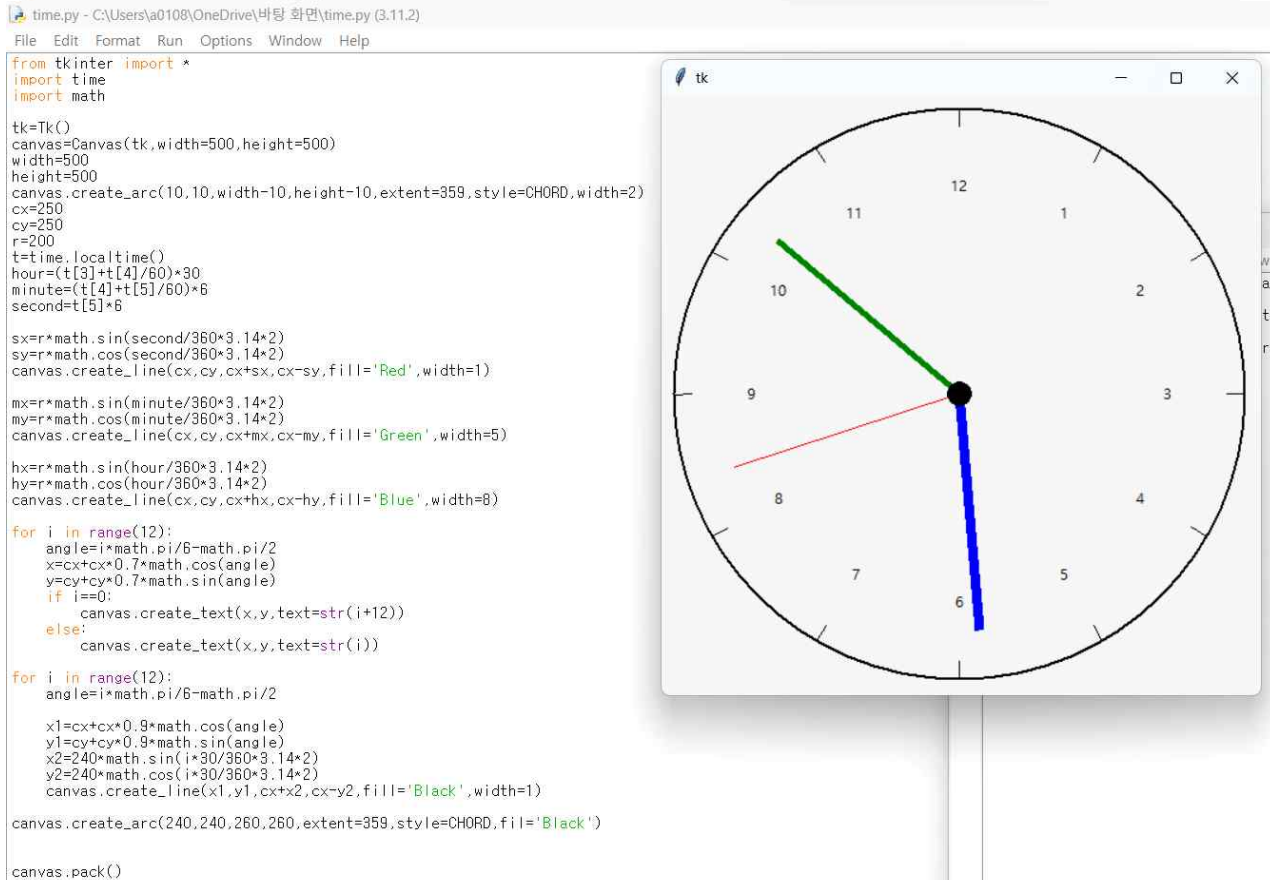Lab09_2312282_임다희

#1 시계

```python
from tkinter import *
import time
import math

tk=Tk()
canvas=Canvas(tk,width=500,height=500)
width=500
height=500
canvas.create_arc(10,10,width-10,height-10,extent=359,style=CHORD,width=2)
cx=250
cy=250
r=200
t=time.localtime()
hour=(t[3]+t[4]/60)*30
minute=(t[4]+t[5]/60)*6
second=t[5]*6

sx=r*math.sin(second/360*3.14*2)
sy=r*math.cos(second/360*3.14*2)
canvas.create_line(cx,cy,cx+sx,cx-sy,fill='Red',width=1)

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

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

for i in range(12):
    angle=i*math.pi/6-math.pi/2
    x=cx+cx*0.7*math.cos(angle)
    y=cy+cy*0.7*math.sin(angle)
    if i==0:
        canvas.create_text(x,y,text=str(i+12))
    else:
        canvas.create_text(x,y,text=str(i))

for i in range(12):
    angle=i*math.pi/6-math.pi/2

    x1=cx+cx*0.9*math.cos(angle)
    y1=cy+cy*0.9*math.sin(angle)
    x2=240*math.sin(i*30/360*3.14*2)
    y2=240*math.cos(i*30/360*3.14*2)
    canvas.create_line(x1,y1,cx+x2,cx-y2,fill='Black',width=1)

canvas.create_arc(240,240,260,260,extent=359,style=CHORD,fil='Black')

canvas.pack()
```

```
from tkinter import *
import time
import math

tk=Tk()
canvas=Canvas(tk,width=500,height=500)
width=500
height=500
canvas.create_arc(10,10,width-10,height-10,extent=359,style=CHORD,width=2)
cx=250
cy=250
r=200
t=time.localtime()
hour=(t[3]+t[4]/60)*30
minute=(t[4]+t[5]/60)*6
second=t[5]*6

sx=r*math.sin(second/360*3.14*2)
sy=r*math.cos(second/360*3.14*2)
canvas.create_line(cx,cy,cx+sx,cx-sy,fill='Red',width=1)

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

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

for i in range(12):
    angle=i*math.pi/6-math.pi/2
    x=cx+cx*0.7*math.cos(angle)
    y=cy+cy*0.7*math.sin(angle)
    if i==0:
        canvas.create_text(x,y,text=str(i+12))
    else:
        canvas.create_text(x,y,text=str(i))

for i in range(12):
    angle=i*math.pi/6-math.pi/2

    x1=cx+cx*0.9*math.cos(angle)
    y1=cy+cy*0.9*math.sin(angle)
    x2=240*math.sin(i*30/360*3.14*2)
    y2=240*math.cos(i*30/360*3.14*2)
    canvas.create_line(x1,y1,cx+x2,cx-y2,fill='Black',width=1)

canvas.create_arc(240,240,260,260,extent=359,style=CHORD,fil='Black')


canvas.pack()
```
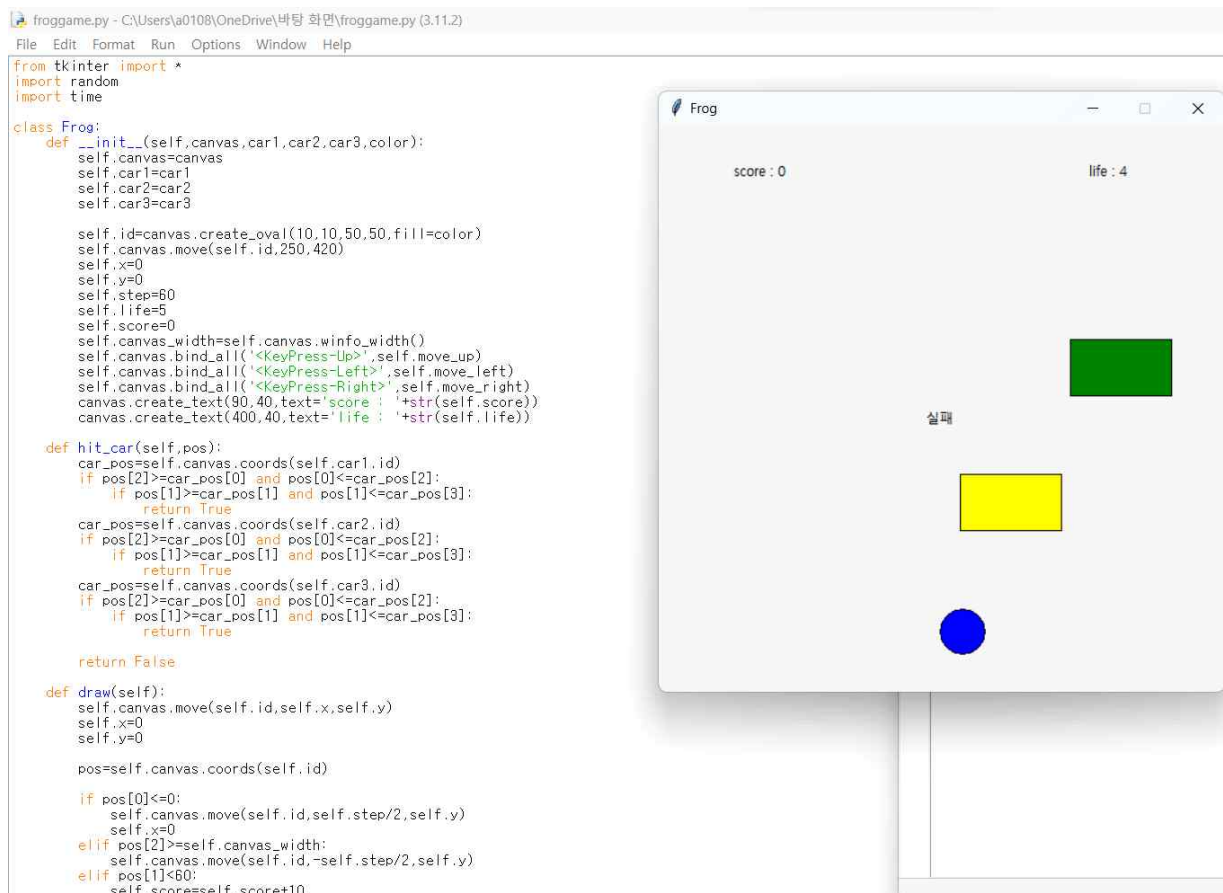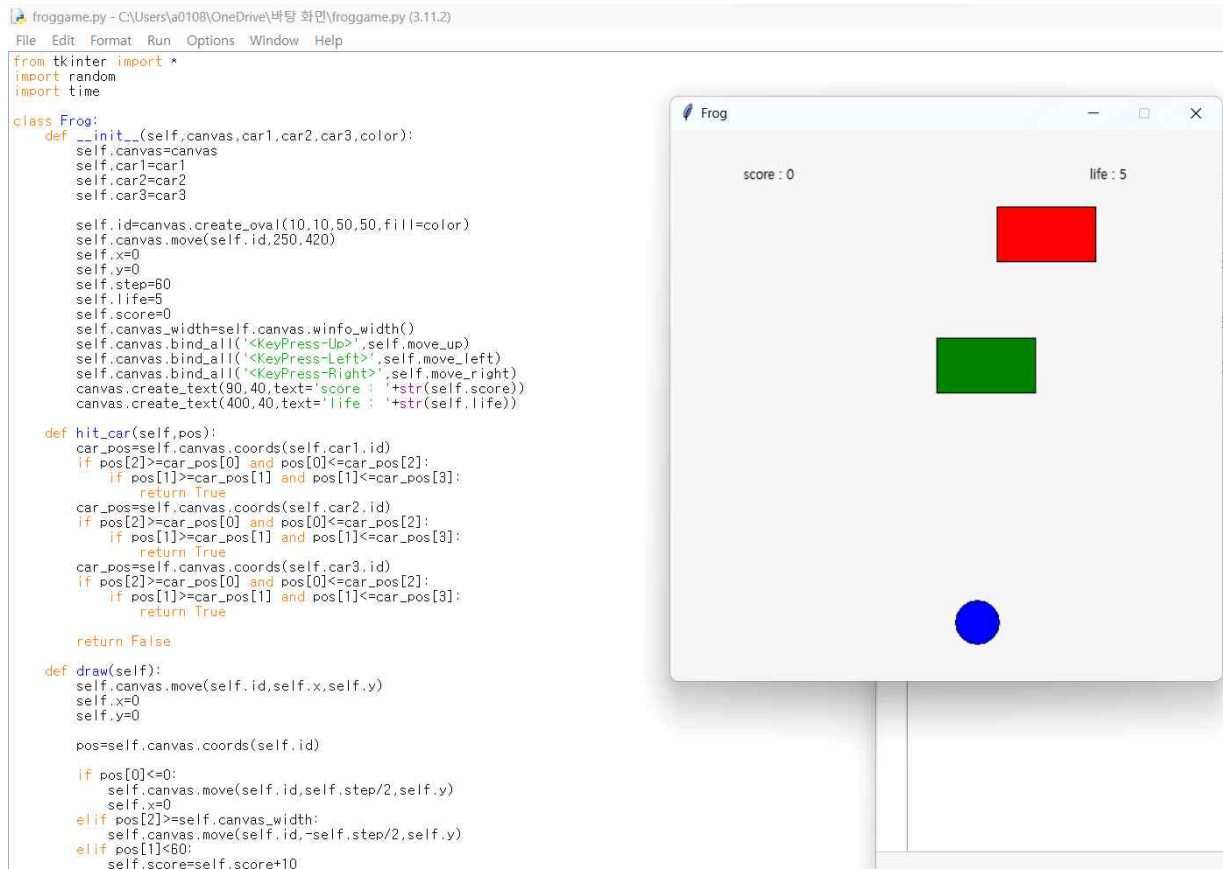
#2. 개구리 게임

```python
from tkinter import *
import random
import time

class Frog:
    def __init__(self,canvas,car1,car2,car3,color):
        self.canvas=canvas
        self.car1=car1
        self.car2=car2
        self.car3=car3

        self.id=canvas.create_oval(10,10,50,50,fill=color)
        self.canvas.move(self.id,250,420)
        self.x=0
        self.y=0
        self.step=60
        self.life=5
        self.score=0
        self.canvas_width=self.canvas.winfo_width()
        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)
        canvas.create_text(90,40,text='score : '+str(self.score))
        canvas.create_text(400,40,text='life : '+str(self.life))

    def hit_car(self,pos):
        car_pos=self.canvas.coords(self.car1.id)
        if pos[2]>=car_pos[0] and pos[0]<=car_pos[2]:
            if pos[1]>=car_pos[1] and pos[1]<=car_pos[3]:
                return True
        car_pos=self.canvas.coords(self.car2.id)
        if pos[2]>=car_pos[0] and pos[0]<=car_pos[2]:
            if pos[1]>=car_pos[1] and pos[1]<=car_pos[3]:
                return True
        car_pos=self.canvas.coords(self.car3.id)
        if pos[2]>=car_pos[0] and pos[0]<=car_pos[2]:
            if pos[1]>=car_pos[1] and pos[1]<=car_pos[3]:
                return True

        return False

    def draw(self):
        self.canvas.move(self.id,self.x,self.y)
        self.x=0
        self.y=0

        pos=self.canvas.coords(self.id)

        if pos[0]<=0:
            self.canvas.move(self.id,self.step/2,self.y)
            self.x=0
        elif pos[2]>=self.canvas_width:
            self.canvas.move(self.id,-self.step/2,self.y)
        elif pos[1]<60:
            self.score=self.score+10
```

```python
from tkinter import *
import random
import time

class Frog:
    def __init__(self,canvas,car1,car2,car3,color):
        self.canvas=canvas
        self.car1=car1
        self.car2=car2
        self.car3=car3

        self.id=canvas.create_oval(10,10,50,50,fill=color)
        self.canvas.move(self.id,250,420)
        self.x=0
        self.y=0
        self.step=60
        self.life=5
        self.score=0
        self.canvas_width=self.canvas.winfo_width()
        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)
        canvas.create_text(90,40,text='score : '+str(self.score))
        canvas.create_text(400,40,text='life : '+str(self.life))

    def hit_car(self,pos):
        car_pos=self.canvas.coords(self.car1.id)
        if pos[2]>=car_pos[0] and pos[0]<=car_pos[2]:
            if pos[1]>=car_pos[1] and pos[1]<=car_pos[3]:
                return True
        car_pos=self.canvas.coords(self.car2.id)
        if pos[2]>=car_pos[0] and pos[0]<=car_pos[2]:
            if pos[1]>=car_pos[1] and pos[1]<=car_pos[3]:
                return True
        car_pos=self.canvas.coords(self.car3.id)
        if pos[2]>=car_pos[0] and pos[0]<=car_pos[2]:
            if pos[1]>=car_pos[1] and pos[1]<=car_pos[3]:
                return True

        return False

    def draw(self):
        self.canvas.move(self.id,self.x,self.y)
        self.x=0
        self.y=0

        pos=self.canvas.coords(self.id)

        if pos[0]<=0:
            self.canvas.move(self.id,self.step/2,self.y)
            self.x=0
        elif pos[2]>=self.canvas_width:
            self.canvas.move(self.id,-self.step/2,self.y)
        elif pos[1]<60:
            self.score=self.score+10
```

```python
from tkinter import *
import random
import time

class Frog:
    def __init__(self,canvas,car1,car2,car3,color):
        self.canvas=canvas
        self.car1=car1
        self.car2=car2
        self.car3=car3

        self.id=canvas.create_oval(10,10,50,50,fill=color)
        self.canvas.move(self.id,250,420)
        self.x=0
        self.y=0
        self.step=60
        self.life=5
        self.score=0
        self.canvas_width=self.canvas.winfo_width()
        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)
        canvas.create_text(90,40,text='score : '+str(self.score))
        canvas.create_text(400,40,text='life : '+str(self.life))

    def hit_car(self,pos):
        car_pos=self.canvas.coords(self.car1.id)
        if pos[2]>=car_pos[0] and pos[0]<=car_pos[2]:
            if pos[1]>=car_pos[1] and pos[1]<=car_pos[3]:
                return True
        car_pos=self.canvas.coords(self.car2.id)
        if pos[2]>=car_pos[0] and pos[0]<=car_pos[2]:
            if pos[1]>=car_pos[1] and pos[1]<=car_pos[3]:
                return True
        car_pos=self.canvas.coords(self.car3.id)
        if pos[2]>=car_pos[0] and pos[0]<=car_pos[2]:
            if pos[1]>=car_pos[1] and pos[1]<=car_pos[3]:
                return True

        return False

    def draw(self):
        self.canvas.move(self.id,self.x,self.y)
        self.x=0
        self.y=0

        pos=self.canvas.coords(self.id)

        if pos[0]<=0:
            self.canvas.move(self.id,self.step/2,self.y)
            self.x=0
        elif pos[2]>=self.canvas_width:
            self.canvas.move(self.id,-self.step/2,self.y)
        elif pos[1]<60:
            self.score=self.score+10
            canvas.create_rectangle(10,10,200,60,outline=tk.cget('bg'),fill=tk.cget('bg'))
            canvas.create_text(90,40,text='score : '+str(self.score))
            self.canvas.move(self.id,250-pos[0],420)

        if self.hit_car(pos)==True:
            self.life=self.life-1
            if self.life<0:
                canvas.create_text(250,260,text= 'G A M E O V E R')
                tk.update()

            else:
                canvas.create_text(250,260,text= '실패',tags=('label'))
                canvas.create_rectangle(300,10,550,60,outline=
```

```python
                    tk.cget('bg'),fill=tk.cget('bg'))
                canvas.create_text(400,40,text='life : '+str(self.life))
                self.canvas.move(self.id,250-pos[0],430-pos[1])

                tk.update_idletasks()
                tk.update()
                time.sleep(3)
                self.canvas.delete('label')

    def move_up(self,evt):
        self.y=-self.step
    def move_left(self,evt):
        self.x=-self.step/2
    def move_right(sel,evt):
        self.x=self.step/2

class Car:
    def __init__(self,canvas,x,y,color,speed):
        self.canvas=canvas
        self.id=canvas.create_rectangle(10,10,100,60,fill=color)
        self.canvas.move(self.id,x,y)
        self.speed=speed
        self.x=speed
        self.y=0
    def draw(self):
        self.canvas.move(self.id,self.x,self.y)
        pos=self.canvas.coords(self.id)
        if  pos[0]<=-100:
            self.canvas.move(self.id,600,0)
        elif  pos[2]>=700:
            self.canvas.move(self.id,-700,0)
tk=Tk()
tk.title('Frog')
tk.resizable(0,0)
tk.wm_attributes('-topmost',1)
canvas=Canvas(tk,width=500,height=500)
canvas.pack()
tk.update()


game_speed=0.01
car1=Car(canvas,10,60,'red',2)
car2=Car(canvas,500,180,'green',-3)
car3=Car(canvas,10,300,'yellow',1)
frog=Frog(canvas,car1,car2,car3,'blue')

def change_speed(evt):
    global game_speed
    if game_speed==0.01:
        game_speed=game_speed/2
    elif game_speed==0.005:
        game_speed=game_speed/2
    elif game_speed==0.0025:
        game_speed=game_speed*4

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


while True:
    if frog.life>=0:
        car1.draw()
        car2.draw()
        car3.draw()
        frog.draw()

    tk.update_idletasks()
    tk.update()
    time.sleep(game_speed)
```