

4. 그래픽 인터페이스 요소: 올바른 도구 선택하기



순천향대학교 컴퓨터공학과
이 상 정



학습 내용

- 윈도우 관리자(window manager)
 - GUI에 의해 생성된 윈도우를 관리
- GUI 이벤트
 - 운영체제, 윈도우 관리자, 사용자에게 의해 발생
- 윈도우 관리자 프로토콜 속성
 - GUI 윈도우에 연결된 이벤트를 처리하는 메커니즘
 - 프로토콜 이벤트 메시지
- 체크 상자(checkbox) 위젯
 - 켜짐/꺼짐 상태가 전환(토글, toggle)
- 스케일(scale) 위젯
 - 슬라이더를 생성하는 위젯

음악 믹싱 예

□ DJ를 위한 음악 믹싱 프로그램

- LP 앨범을 디지털화 하여 클럽에서 바로 믹싱



음악 믹싱 코드 (1)

```

from tkinter import *
import pygame.mixer

app=Tk()
app.title("Head First Mix")
app.geometry('250x100+200+100')

sound_file="50459_M_RED_Nephlimizer.wav" # 연주할 음악 파일

mixer = pygame.mixer
mixer.init() # 사운드 시스템 시작

def track_start():
    track.play(loops=-1) # play() 함수에 loops 인자 값을 -1로 지정하면
                        # stop()을 호출할 때까지 계속 반복 연주

def track_stop():
    track.stop()

```

음악 믹싱 코드 (2)

```

track=mixer.Sound(sound_file)           # 사운드 파일 로드

start_button = Button(app, command = track_start, text = "Start")
start_button.pack(side = LEFT)

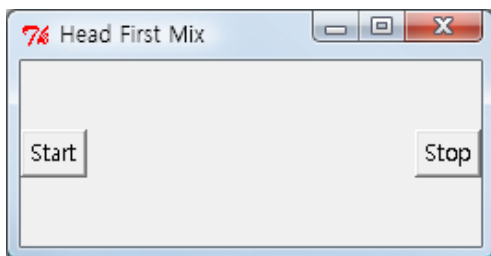
stop_button = Button(app, command = track_stop, text = "Stop")
stop_button.pack(side = RIGHT)

app.mainloop()

```



시험주행



Stop 버튼을 누르지 않고 윈도우 닫기를 하니 음악이 멈추지 않고 계속 연주되네.....

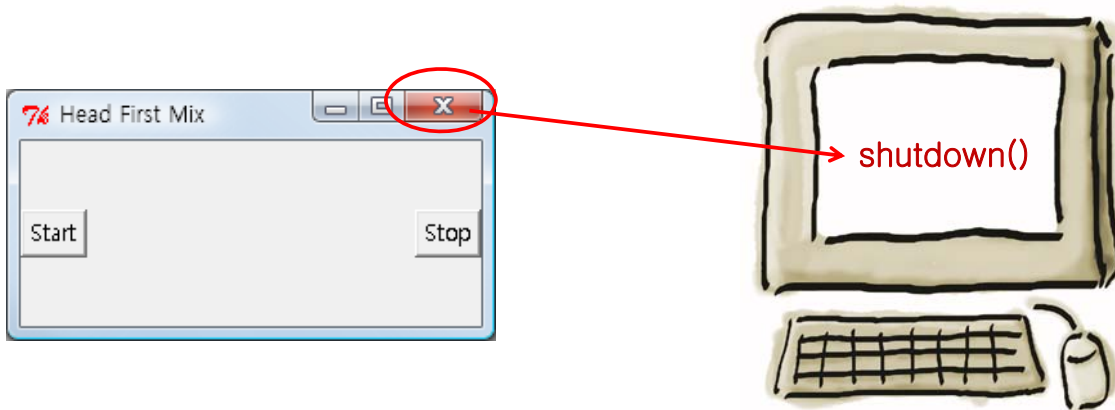


GUI 이벤트

- ❑ GUI 프로그램은 사용자 버튼 클릭과 같은 이벤트 뿐만 아니라 운영체제, 윈도우 관리자가 발생시킨 이벤트도 처리
- ❑ 윈도우 관리자(window manager)는 GUI에 의해 생성된 윈도우를 관리
 - 윈도우 시작, 종료, 활성화 등 윈도우 관리자가 이벤트를 생성
- ❑ 윈도우 관리자 프로토콜 속성
 - GUI 윈도우에 연결된 이벤트를 처리하는 메커니즘
- ❑ 윈도우 관리자 프로토콜 이벤트 메시지
 - WM_TAKE_FOCUS
 - 마우스로 클릭하여 윈도우가 활성화 되었을 때 메인 윈도우로 보내는 메시지
 - WM_SAVE_YOURSELF
 - 운영체제가 종료될 때 메인 윈도우로 보내는 메시지
 - WM_DELETE_WINDOW
 - 닫기 버튼이 클릭 되었을 때 메인 윈도우로 보내는 메시지

윈도우 관리자 제어

- ❑ 이벤트가 tkinter에 도착하기 전에 애플리케이션의 protocol() 메서드를 호출
 - 미리 지정된 기본처리 대신 사용자가 처리할 함수를 지정
 - app.protocol("WM_DELETE_WINDOW", shutdown)



음악 믹싱 코드 : 윈도우 관리자 제어

```

from tkinter import *
import pygame.mixer

app=Tk()
app.title("Head First Mix")
app.geometry('250x100+200+100')

sound_file="50459_M_RED_Nephlimizer.wav"

mixer = pygame.mixer
mixer.init()

def track_start():
    track.play(loops=-1)

def track_stop():
    track.stop()

```

```

def shutdown():
    track.stop()

track=mixer.Sound(sound_file)

start_button = Button(app, command = track_start,
                      text = "Start")
start_button.pack(side = LEFT)

stop_button = Button(app, command = track_stop,
                    text = "Stop")
stop_button.pack(side = RIGHT)

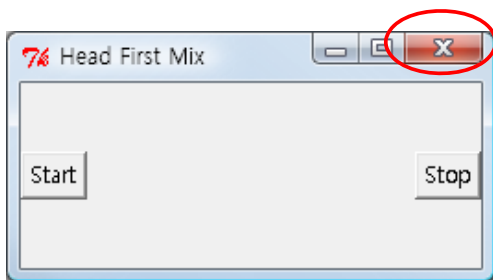
app.protocol("WM_DELETE_WINDOW", shutdown)

app.mainloop()

```



시험주행



달기 버튼을 누르니 음악은
멈추는데 윈도우가 닫히지
않네

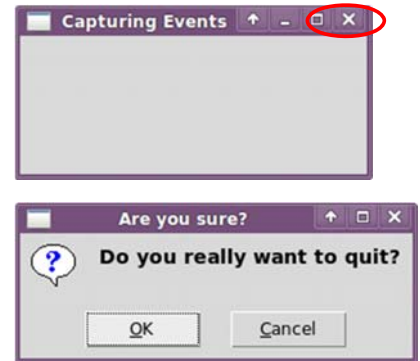


윈도우 관리자 프로토콜

- `protocol()` 메서드는 프로토콜 이벤트를 붙잡아서 작동할 재정의
 - 원래 기본 처리는 작동되지 않음
 - 따라서 사용자 처리에는 원래 기본 처리 동작도 포함되어야 함
- 윈도우 닫기
 - 애플리케이션의 `destroy()` 메서드

```
def shutdown():
    if askokcancel(title='Are you sure?',
                  message='Do you really want to quit?'):
        app.destroy()

app.protocol("WM_DELETE_WINDOW", shutdown)
```



음악 믹싱 코드 : 윈도우 관리자 제어 수정

```
from tkinter import *
import pygame.mixer

app=Tk()
app.title("Head First Mix")
app.geometry('250x100+200+100')

sound_file="50459_M_RED_Nephlimizer.wav"

mixer = pygame.mixer
mixer.init()

def track_start():
    track.play(loops=-1)

def track_stop():
    track.stop()
```

```
def shutdown():
    track.stop()
    app.destroy()

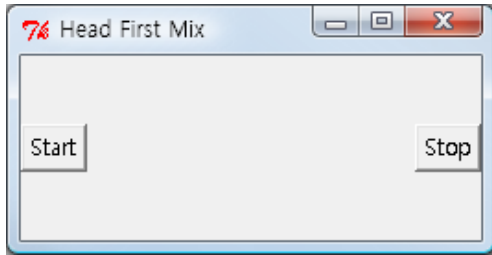
track=mixer.Sound(sound_file)

start_button = Button(app, command = track_start,
                      text = "Start")
start_button.pack(side = LEFT)

stop_button = Button(app, command = track_stop,
                     text = "Stop")
stop_button.pack(side = RIGHT)

app.protocol("WM_DELETE_WINDOW", shutdown)

app.mainloop()
```



체크 상자 위젯 (Checkbox Widget)

- ❑ 두 개의 버튼 대신 하나의 버튼으로 음악 연주 켜짐/꺼짐 상태가 전환(토글, toggle)
- ❑ 체크 상자(checkbox) 위젯
 - 켜짐/꺼짐 상태가 전환(토글, toggle)
 - 체크 상자 위젯 객체 생성: `Checkbutton()`
 - 정수형 제어변수(`IntVar()`) 사용: 체크 1, 비체크 0
 - `get()` 메서드 사용하여 읽기

```
flipper = IntVar()

def flip_it():
    if flipper.get() == 1:
        print("I am on!")
    else:
        print("I am off!")

Checkbutton(app, variable=flipper, command=flip_it,
            text="Flip it?").pack()
```



음악 믹싱 코드 : 체크 박스 버튼

```

from tkinter import *
import pygame.mixer

app=Tk()
app.title("Head First Mix")
app.geometry('250x100+200+100')

sound_file="50459_M_RED_Nephlimizer.wav"

mixer = pygame.mixer
mixer.init()

def track_toggle():
    if track_playing.get() == 1:
        track.play(loops=-1)
    else:
        track.stop()
    
```

```

def shutdown():
    track.stop()
    app.destroy()

track=mixer.Sound(sound_file)

track_playing = IntVar()
track_button = Checkbutton(app, variable = track_playing,
                           command = track_toggle,
                           text = sound_file)

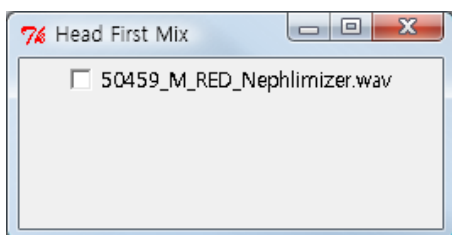
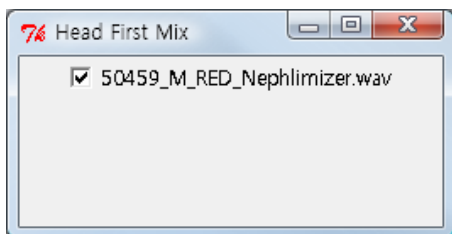
track_button.pack()

app.protocol("WM_DELETE_WINDOW", shutdown)

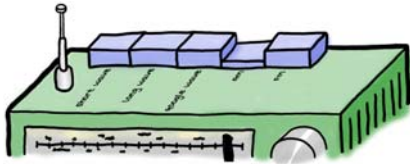
app.mainloop()
    
```



시험주행



□ 음악 믹싱 프로그램에 볼륨 조절 기능 추가



□ 볼륨 조절 주요 특징

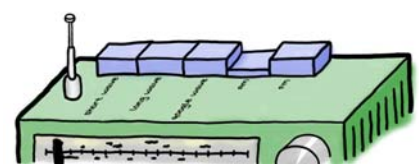
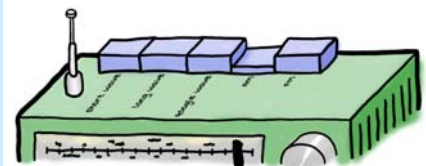
- 최소값에서 최대값까지 눈금자
- 눈금자 간격 일정
- 볼륨 조절 슬라이더는 왼쪽에서 오른쪽 이동
- 슬라이더를 움직이면 눈금자의 현재 위치에 맞게 동적으로 볼륨 조절

□ Sound 객체의 set_volume() 메서드를 사용하여 조절

```
import pygame.mixer

mixer = pygame.mixer
mixer.init()

track=mixer.Sound("50459_M_RED_Nephlimizer.wav")
print("Play it LOUD, man!")
track.play(loops=-1)
track.set_volume(0.9) # 볼륨을 높임
sleep(2)
print("Softly does it ...")
track.set_volume(0.1) # 볼륨을 낮춤
sleep(2)
track.stop()
```



스케일 위젯 (Scale Widget)

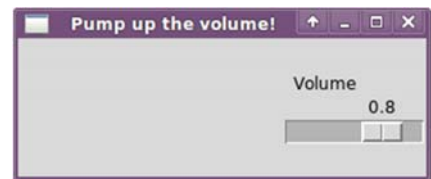
- ❑ tkinter 라이브러리는 **슬라이더**를 위해 스케일 위젯 제공
- ❑ **스케일(scale) 위젯**
 - 슬라이더를 생성하는 위젯
 - 스케일 위젯 객체 생성: `Scale()`
 - 인자로 제어변수, 눈금자 최소/최대값, 해상도(눈금간격), 수평/수직 지정

```

volume = DoubleVar() # 제어변수 생성
volume_scale = Scale(app,
                    variable = volume,
                    from_ = 0.0, # 눈금자 최소값
                    to = 1.0, # 눈금자 최대값
                    resolution = 0.1, # 해상도(눈금간격)
                    command = change_volume,
                    label = "Volume",
                    orient = HORIZONTAL) # 수평

volume_scale.pack()

```



음악 믹싱 코드 : 볼륨 조절(스케일 위젯)

```

from tkinter import *
import pygame.mixer

app=Tk()
app.title("Head First Mix")
sound_file="50459_M_RED_Nephlimizer.wav"
mixer = pygame.mixer
mixer.init()

def track_toggle():
    if track_playing.get() == 1:
        track.play(loops=-1)
    else:
        track.stop()

def change_volume(v):
    # Scale 클래스 내에서 호출 시 인수 포함
    # 스케일 값을 나타내는 인수
    track.set_volume(volume.get())
    # track.set_volume(float(v))

def shutdown():
    track.stop()
    app.destroy()

```

```

track=mixer.Sound(sound_file)

track_playing = IntVar()
track_button = Checkbutton(app, variable = track_playing,
                           command = track_toggle,
                           text = sound_file)

track_button.pack(side=LEFT)

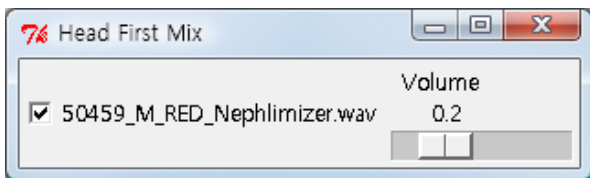
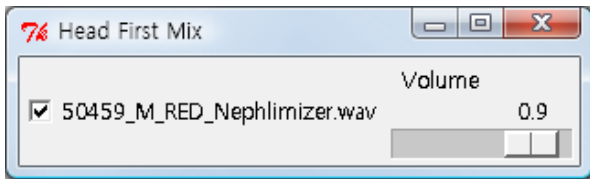
volume=DoubleVar()
volume.set(track.get_volume())
volume_scale=Scale(variable=volume,
                  from_=0.0,
                  to=1.0,
                  resolution=0.1,
                  command=change_volume,
                  label="Volume",
                  orient=HORIZONTAL)

volume_scale.pack(side=RIGHT)

app.protocol("WM_DELETE_WINDOW", shutdown)

app.mainloop()

```



베이스 요소



□ 프로그래밍 도구

- 윈도우 관리자는 GUI에 의해 생성된 윈도우를 관리
- GUI 이벤트는 운영체제, 윈도우 관리자, 사용자에게 의해 발생
- 만약 기본 행위를 재정의한다면 재정의 함수에 원래 수행하던 기본 행위를 포함해야 함
- 체크 상자는 어떤 것이 켜짐 또는 꺼짐 상태를 전환(토글)
- 눈금자/슬라이더로 볼륨 컨트롤을 구현
- 소개되지 않은 자세한 GUI 위젯이나 프로그래밍 기법은 파이썬 tkinter 문서 참조



프로그래밍 도구상자 (2)

□ 파이썬 도구

- `app.destroy()` - tkinter GUI 애플리케이션을 종료
- `DoubleVar()` - `IntVar`, `StringVar`와 비슷하지만 부동소수점 숫자를 저장
- `Checkbutton()` - 체크 상자를 생성하는 tkinter 위젯
- `Scale()` - 슬라이더를 생성하는 tkinter 위젯

과제

1. 최종 음악 믹싱 코드를 작성하고 실행
2. 앞에서 배운 내용을 사용한 임의의 프로그램 작성
 - 프로그램 설명
 - 프로그램 소스
 - 실행 결과