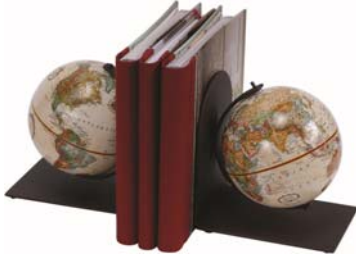


1. 그래픽 사용자 인터페이스 구현하기: 매끈하게 보이기 (7장)



순천향대학교 컴퓨터공학과
이 상 정

순천향대학교 컴퓨터공학과

1

GUI 설계기법

학습 내용

- 지금까지 텍스트 기반의 콘솔(console) 입출력
- 그래픽 사용자 인터페이스
 - GUI, Graphical User Interface
 - 위젯 (widget)
 - GUI 인터페이스 요소
 - 윈도우, 버튼, 레이블, 체크박스
 - Tkinter
 - Tcl/Tk 기반 표준 GUI 라이브러리
- 파이게임(pygame)
 - 써드파티 라이브러리(third-party library)

순천향대학교 컴퓨터공학과

2

1. 그래픽 사용자 인터페이스 구현하기

TV 게임 쇼 예

□ 헤드 퍼스트 TVN의 게임 쇼 프로그램

- 진행자가 게임 참가자에게 질문한 후 정답인 경우 1, 오답인 경우 2를 입력
- 누른 키에 따라 적절한 **사운드 효과** 발생
- 프로그램은 정답과 오답의 수를 기억
- 진행자가 0을 누르면 퀴즈가 끝나고, 프로그램은 질문과 정답, 오답의 수를 출력



TV 게임 쇼 예 의사코드(pseudo code)

```

number_asked = 0    # 질문, 정답, 오답 갯 수 변수 초기화
number_correct = 0
number_wrong = 0

화면에 정답이면 1, 오답이면 2, 끝내려면 0을 누르라고 요청
while 입력 값이 0이 아님
    if 입력 값이 1이면    # 정답이면
        number_asked를 1 증가시킴
        number_correct를 1 증가시킴
        사운드 효과
    if 입력 값이 2이면    # 오답이면
        number_asked를 1 증가시킴
        number_wrong를 1 증가시킴
        사운드 효과
    화면에 정답이면 1, 오답이면 2, 끝내려면 0을 누르라고 요청

화면에 number_asked, number_correct, number_wrong 값 출력
    
```

파이게임(pygame) 사운드 효과

□ 사운드 효과 지원을 위해 파이게임 설치

- 파이게임은 써드파티 라이브러리(third-party library)
- <http://www.pygame.org>
- 프롬프트 창에서 다음 명령 실행하여 설치
C:\W> python -m pip install pygame

```
C:\Users\김민우> python -m pip install pygame
Collecting pygame
  Downloading pygame-1.9.3-cp36-cp36m-win_amd64.whl (4.2MB)
    100% |#####| 4.2MB 325kB/s
Installing collected packages: pygame
Successfully installed pygame-1.9.3
```

□ 파이게임을 설치 후 다음 사운드 효과 예 실행

- 사운드 파일은 강의 페이지에서 **sound.zip** 파일 다운받아 압축 풀어서 사용
- 프로그램 코드와 사운드 파일은 같은 폴더에 저장

사운드 효과 예

```
import pygame.mixer          # mixer 모듈 임포트, 사운드를 메모리에 적재하고 연주

sounds = pygame.mixer        # pygame.mixer 모듈 참조
sounds.init()                # 사운드 시스템 초기화

def wait_finish(channel):     # 함수 정의
    while channel.get_busy(): # channel의 get_busy() 메서드가 거짓이 될 때까지 반복
        pass                 # 아무 일도 하지 않는 동작

s = sounds.Sound("heartbeat.wav") # 연주할 사운드 파일 읽고 Sound 객체 생성
wait_finish(s.play())           # s.play(), 사운드 연주하고 선택된 채널 객체 리턴

s2 = sounds.Sound("buzz.wav")   # 연주할 사운드 파일 읽음
wait_finish(s2.play())          # 사운드 연주

s3 = sounds.Sound("ohno.wav")   # 연주할 사운드 파일 읽음
wait_finish(s3.play())          # 사운드 연주

s4 = sounds.Sound("carhorn.wav") # 연주할 사운드 파일 읽음
wait_finish(s4.play())          # 사운드 연주
```

TV 게임쇼 코드 (1)

```

import pygame.mixer

def wait_finish(channel):
    while channel.get_busy():
        pass

sounds = pygame.mixer
sounds.init()

correct_s = sounds.Sound("correct.wav")
wrong_s = sounds.Sound("wrong.wav")

prompt = "Press 1 for Correct, 2 for Wrong, or 0 to Quit: "

number_asked = 0
number_correct = 0
number_wrong = 0

```

TV 게임쇼 코드 (2)

```

choice = input(prompt)
while choice != '0':
    if choice == '1':
        number_asked = number_asked + 1
        number_correct = number_correct + 1
        wait_finish(correct_s.play())
    if choice == '2':
        number_asked = number_asked + 1
        number_wrong = number_wrong + 1
        wait_finish(wrong_s.play())
    choice = input(prompt)

print("You asked " + str(number_asked) + "questions.")
print(str(number_correct) + " were correctly answered.")
print(str(number_wrong) + " were answered incorrectly.")

```



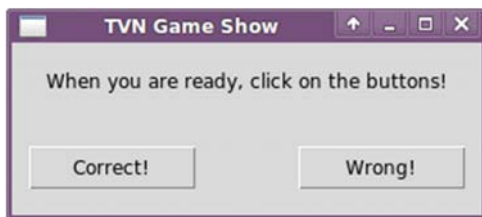
```

Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.2 (default, Sep 4 2011, 09:51:08) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
>>> Press 1 for Correct, 2 for Wrong, or 0 to Quit: 1
Press 1 for Correct, 2 for Wrong, or 0 to Quit: 1
Press 1 for Correct, 2 for Wrong, or 0 to Quit: 1
Press 1 for Correct, 2 for Wrong, or 0 to Quit: 2
Press 1 for Correct, 2 for Wrong, or 0 to Quit: 2
Press 1 for Correct, 2 for Wrong, or 0 to Quit: 1
Press 1 for Correct, 2 for Wrong, or 0 to Quit: 1
Press 1 for Correct, 2 for Wrong, or 0 to Quit: 1
Press 1 for Correct, 2 for Wrong, or 0 to Quit: 1
Press 1 for Correct, 2 for Wrong, or 0 to Quit: 1
Press 1 for Correct, 2 for Wrong, or 0 to Quit: 2
Press 1 for Correct, 2 for Wrong, or 0 to Quit: 1
Press 1 for Correct, 2 for Wrong, or 0 to Quit: 0
You asked 12questions.
9 were correctly answered.
3 were answered incorrectly.
>>> |
Ln: 21 Col: 4

```



□ TV 게임 쇼 프로그램을 GUI로 설계



□ 그래픽 사용자 인터페이스(Graphical User Interface, GUI)

- 파이썬에 tkinter 표준 GUI 라이브러리가 설치
 - Tcl/Tk 기반 표준 GUI 라이브러리
- GUI 는 이벤트 루프(event loop) 기반 동작

GUI 이벤트 루프 (Event Loop)

- GUI 이벤트 루프는 특정 이벤트가 발생하면 관련 코드 호출하여 실행
 - 이벤트(event)는 버튼 등을 눌렀을 때 발생
 - 버튼 등과 같은 GUI 구성 요소를 위젯(widget)이라고 함
 - Tkinter는 while 루프를 사용하지 않고 `mainloop()` 메소드를 호출하여 이벤트 루프 시작

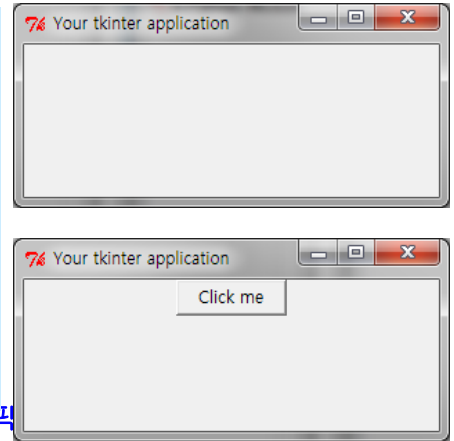
```

from tkinter import * # tkinter 모듈을 임포트

app = Tk() # tkinter 애플리케이션 윈도우 생성
app.title("Your tkinter application") # 윈도우 이름 설정
app.geometry('300x110+200+100') # 윈도우 좌표와 크기를 설정

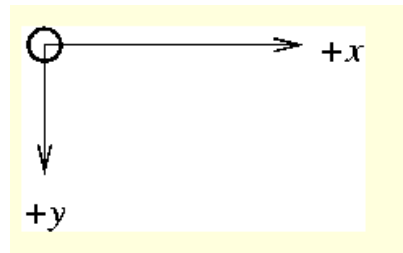
b1 = Button(app, text="Click me", width=10) # 버튼을 윈도우에 추가
b1.pack() # 새로 생성된 버튼을 윈도우에 연결

app.mainloop() # tkinter 이벤트 루프를 시작
    
```



GUI 좌표 시스템 (coordinate system)

- 화면에 표시되는 좌표 시스템
 - 기본 단위는 픽셀(pixel)
 - 화면 위 왼쪽(top left)이 원점 (0,0)



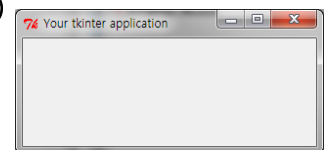
- 윈도우 좌표와 크기

- `app.geometry('300x110+200+100')`
 - width=300, height=110, xoffset=200, yoffset=100

(200,100)

300

110



- 버튼 너비

- `b1 = Button(app, text="Click me", width=10)`
 - 버튼의 너비 크기가 텍스트 10 글자
 - 텍스트가 아니고 이미지가 삽입되면 단위가 글자가 아니고 픽셀

pack()메소드

□ pack() 메소드

- 위젯을 윈도우에 연결 표시
- side 인자를 사용하여 위젯의 위치를 조정
- padx, pady 인자를 사용하여 위젯 주위에 패딩(여백)을 표시

□ 예

- pack (side = 'left')
- pack (side = 'right')
- pack (side = 'top')
- pack (side = 'bottom')
- pack(padx = 10, pady = 10)
=> 4 방향으로 10픽셀씩 패딩 추가



TV 게임쇼 코드: GUI 요소 표시

```

from tkinter import *

app = Tk()
app.title("TVN Game Show")
app.geometry('300x110+200+100')

b1 = Button(app, text = "Correct!", width = 10)
b1.pack(side = 'left', padx = 10, pady = 10)

b2 = Button(app, text = "Wrong!", width = 10)
b2.pack(side = 'right', padx = 10, pady = 10)

app.mainloop()

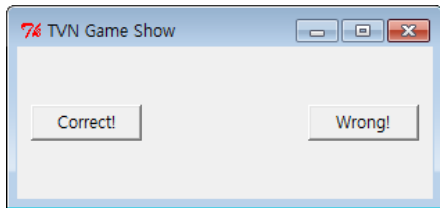
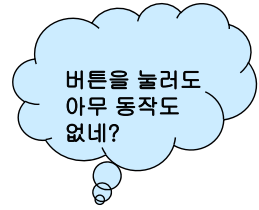
```



```

Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.2 (default, Sep 4 2011, 09:51:08) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
|
Ln: 5 Col: 0

```



1. 그래픽 사용자 인터페이스

버튼 이벤트 코드

- 버튼을 눌렀을 때 실행할 함수
- **command 인자**로 실행할 함수 이름을 지정
- 예

```
b = Button(app, text = "Click on me!", width=15, command=button_click)
```

The composite image illustrates the execution of a button click event. On the left, a window titled 'Click on me' contains a button with the text 'Click on me!'. A red arrow points from this button to the corresponding line in the Python script: `b = Button(app, text = "Click on me!", width = 15, command = button_click)`. Another red arrow points from the `button_click` function definition in the script to the terminal output: `I've just been clicked!`. The Python script window shows the following code:

```

from tkinter import *

app = Tk()
app.title("Click on me")
app.geometry('300x100+200+100')

def button_click():
    print("I've just been clicked!")

b = Button(app, text = "Click on me!", width = 15, command = button_click)
b.pack(padx = 10, pady = 10)

app.mainloop()

```

The terminal window shows the output of the script: `I've just been clicked!`. The bottom right corner of the terminal window contains the text '사용자 인터페이스 구현하기' (Implementing the User Interface).

TV 게임쇼 코드: GUI 프로그램 (1)

```

from tkinter import *
import pygame.mixer

sounds = pygame.mixer
sounds.init()
correct_s = sounds.Sound("correct.wav")
wrong_s = sounds.Sound("wrong.wav")

number_correct = 0
number_wrong = 0

def play_correct_sound():
    global number_correct          # 광역 변수(global variable)
    number_correct = number_correct + 1
    correct_s.play()

def play_wrong_sound():
    global number_wrong           # 광역 변수(global variable)
    number_wrong = number_wrong + 1
    wrong_s.play()

```

순천향

스 구현하기

TV 게임쇼 코드: GUI 프로그램 (2)

```

app = Tk()
app.title("TVN Game Show")
app.geometry('300x110+200+100')

b1 = Button(app, text = "Correct!", width = 10, command = play_correct_sound)
b1.pack(side = 'left', padx = 10, pady = 10)

b2 = Button(app, text = "Wrong!", width = 10, command = play_wrong_sound)
b2.pack(side = 'right', padx = 10, pady = 10)

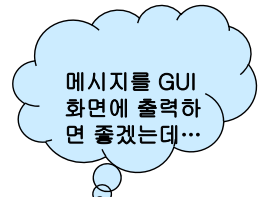
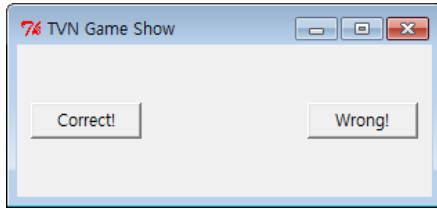
app.mainloop()

print(str(number_correct) + " were correctly answered.")
print(str(number_wrong) + " were answered incorrectly.")

```



시험주행



```
Python Shell
File Edit Shell Debug Options Windows Help
Python 3.2.2 (default, Sep 4 2011, 09:51:08) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>> ===== RESTART =====
>>>
10 were correctly answered.
3 were answered incorrectly.
>>> |
```

1. 그래픽 사용자 인터페이스 구현하기

파이썬 위젯

위젯 이름	설명
버튼 (button)	사용자 응답이나 옵션을 표시하는 위젯
레이블 (label)	윈도우 안에 스트링 메시지를 출력하는 위젯
텍스트 상자 (text box)	텍스트를 입력하는 위젯
메뉴 (menu)	윈도우 꼭대기에 붙어 있는 일련의 명령 옵션들
드롭다운 리스트 (dropdown list)	목록에서 하나의 항목을 선택할 수 있도록 하는 위젯
콤보상자 (combobox)	드롭다운 리스트와 텍스트 상자가 결합되어 있는 위젯
대화상자 (dialogbox)	사용자에게 추가적인 정보를 입력하도록 요청하는 별도의 윈도우

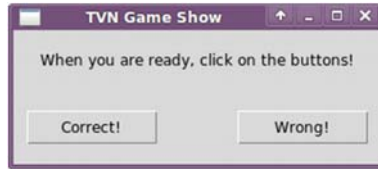
1. 그래픽 사용자 인터페이스 구현하기

레이블 위젯

GUI 윈도우에 레이블 추가

- 예

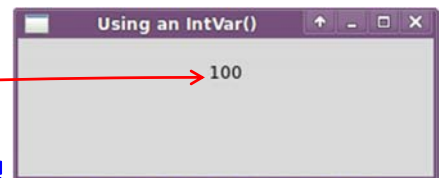
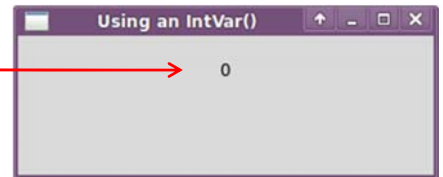
```
l = Label(app, text='When you are ready, click on the buttons!', height=3)
l.pack()
```



```
num_good = IntVar() # 위젯변수 IntVar 생성
num_good.set(0) # StringVar, IntVar, DoubleVar, BooleanVar

l1 = Label(app, textvariable = numgood) # IntVar를 레이블에 연결
l1.pack()
.....

num_good.set(100)
```



TV 게임쇼 코드: 레이블 추가(1)

```
from tkinter import *
import pygame.mixer

def play_correct_sound():
    num_good.set(num_good.get() + 1) # get 메소드로 IntVar 값 읽음
    correct_s.play()

def play_wrong_sound():
    num_bad.set(num_bad.get() + 1)
    wrong_s.play()

app = Tk()
app.title("TVN Game Show")
app.geometry('300x110+200+100')

sounds = pygame.mixer
sounds.init()

correct_s = sounds.Sound("correct.wav")
wrong_s = sounds.Sound("wrong.wav")
```

```

num_good = IntVar()
num_good.set(0)
num_bad = IntVar()
num_bad.set(0)

lab = Label(app, text='When you are ready, click on the buttons!', height = 3)
# 레이블 높이는 3줄

lab.pack()

lab1 = Label(app, textvariable = num_good)
lab1.pack(side = 'left')

lab2 = Label(app, textvariable = num_bad)
lab2.pack(side = 'right')

b1 = Button(app, text = "Correct!", width = 10, command = play_correct_sound)
b1.pack(side = 'left', padx = 10, pady = 10)

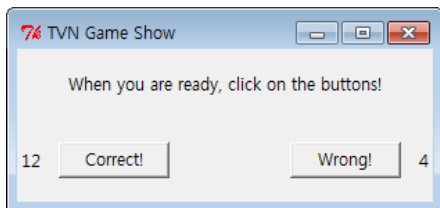
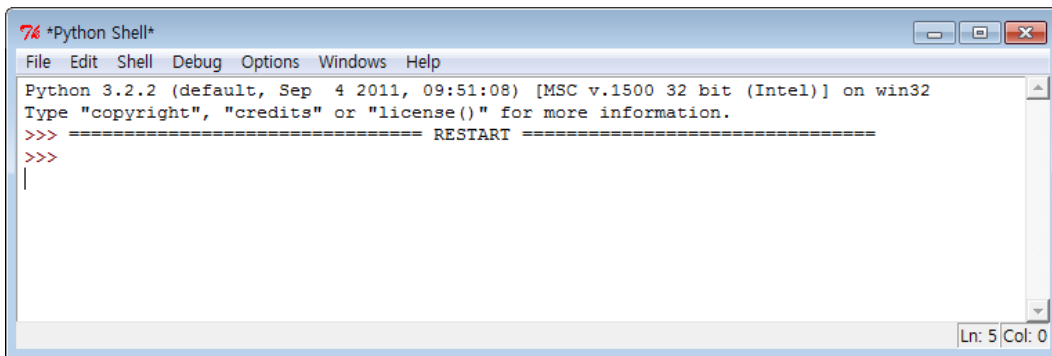
b2 = Button(app, text = "Wrong!", width = 10, command = play_wrong_sound)
b2.pack(side = 'right', padx = 10, pady = 10)

app.mainloop()

```



시험주행





프로그래밍 도구상자 (1)

□ 프로그래밍 도구

- 써드파티 라이브러리 사용하기
- 사운드 라이브러리 연주하기
- 이벤트 루프 - 이벤트가 발생하면 해당 이벤트 처리기 실행
- 이벤트 처리기 - 특정 이벤트가 발생하면 실행되는 코드
- GUI - 그래픽 사용자 인터페이스
- 위젯 - GUI 인터페이스 요소



프로그래밍 도구상자 (2)

□ 파이썬 도구

- 파이게임 - 사운드를 지원하는 전문가 수준의 게임 라이브러리
- pass - 아무 것도 하지 않는 코드
- tkinter - GUI를 생성하기 위한 표준 라이브러리
- Tk() - 빈 GUI 애플리케이션 윈도우
- Button() - tkinter 버튼 위젯
- Label() - tkinter 레이블 위젯
- IntVar() - tkinter 정수형 위젯 변수

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