



명물이와 함께 하는
SKKU MAP

2020310792 송 혁
2020313781 이혜연
2020314154 조아영

“어서와! 성대는 처음이지?”





목차



프로젝트 목적



기능 설명



실행 화면 예시



아쉬운 점



프로젝트 목적

- ✓ 코로나 바이러스로 인해 일부 출입구만 개방됨
- ✓ 1학년 학생들은 학교건물이 익숙하지 않은데 출입구 때문에 어려움을 겪을 수 있음
- ✓ 교차 수강으로 명륜 캠퍼스 학생들이 오게 되는데, 길 찾기에 어려움을 겪는 친구들이 많음
- ✓ 짧은 길을 돌아 가는 경우도 빈번함

! 이러한 문제들을 해결하기 위해 최단거리와, 거쳐가는 길을 알려주는 길 안내 프로그램을 제작하기로 결심



프로그램의 기능

```
mapXY={
  '공학실 습동 24동' :{
    'x':690,
    'y':470
  },
  '공학실 습동 28동' :{
    'x':720,
    'y':475
  },
  '대강당' :{
    'x':205,
    'y':560
  },
  '반도체화학과관' :{
    'x':670,
    'y':670
  },
  '복지회관' :{
    'x':220,
    'y':395
  }
}
```

✓ 지도 위 건물의 위치

```
mapLocation = {
  '정문' : {
    'N센터' : 308.7,
    '의학과관' : 319.9,
    '인자서를 승차' : 398.4
  },
  '의학과관' : {
    '정문' : 319.9,
    'N센터' : 215.2,
    '인자서를 승차' : 337.3,
    '지점1' : 54.4
  },
  'N센터' : {
    '정문' : 308.7,
    '의학과관' : 215.2,
    '지점2' : 47.7,
    '인자서를 승차' : 190.5
  }
}
```

✓ 건물과 건물 사이 거리

```
mapturtlePoint = {
  '정문' : [ [ [369, 815], [406, 817], [527, 635], [526, 652] ],
    '의학과관' : [ [369, 815], [406, 817], [527, 635], [526, 652] ],
    '인자서를 승차' : [ [369, 815], [406, 817], [527, 635], [526, 652] ],
  },
  '대강당' : [ [ [307, 631], [327, 605], [406, 617], [369, 815] ],
    'N센터' : [ [ [307, 631], [327, 605], [527, 635], [526, 652] ],
    '인자서를 승차' : [ [ [307, 631], [327, 605], [527, 635], [526, 652] ],
    '지점1' : [ [ [307, 631], [327, 605], [527, 635], [526, 652] ],
  },
  '의학과관' : [ [ [526, 652], [527, 635], [406, 617], [369, 815] ],
    '정문' : [ [ [526, 652], [527, 635], [406, 617], [369, 815] ],
    '인자서를 승차' : [ [ [526, 652], [527, 635], [406, 617], [369, 815] ],
    '지점1' : [ [ [526, 652], [527, 635], [406, 617], [369, 815] ],
  },
  '공학실 습동 24동' : [ [ [526, 652], [527, 635], [406, 617], [369, 815] ],
    '공학실 습동 28동' : [ [ [526, 652], [527, 635], [406, 617], [369, 815] ],
  },
}
```

✓ 지도 위 터틀 경로 표시

```
def btnOnMapCommand(btnName):
    if window.focus_get().__str__()!='':
        if window.focus_get().__str__()=='!!labelframe.combobox2':
            dest_combo.set(btnName)
            depart_combo.focus()
        elif window.focus_get().__str__()=='!!labelframe.combobox':
            depart_combo.set(btnName)
            dest_combo.focus()
        else:
            depart_combo.set(btnName)
            dest_combo.focus()

b1=tkinter.Button(window, text=" ", bg="light green", command=lambda : btnOnMapCommand('후문'))
b2=tkinter.Button(window, text=" ", bg="light green", command=lambda : btnOnMapCommand('신관B'))
b3=tkinter.Button(window, text=" ", bg="light green", command=lambda : btnOnMapCommand('산학협력센터'))
```

✓ 지도 위에 버튼 표시

✓ 버튼을 눌러 실행 되게 하는 함수 제작

```
def dijkstra(graph, start, finish):
    node = start
    # inf 값 0으로 초기화
    short[start] = 0
    while node is not None:
        cost = short[node]
        neighbors = graph[node]
        for n in neighbors.keys():
            new = cost + neighbors[n]
            if short[n] > new:
                short[n] = new
                trace[n] = node
        process.append(node)
        node = find_short_node(short)
    course = []
    current = finish
    while current != start:
        course.append(current)
        current = trace[current]
    course.append(start)
    course.reverse()
    print("==Dijkstra==")
    print(start, "에서", finish, "까지")
    print("최단거리 :", short[finish], "m")
    print("경로 :", course)

    return short[finish], course
```

✓ 최단 거리 출력 함수

✓ Dijkstra 함수



프로그램의 기능

```
# canvas=turtle.getcanvas()
canvas = turtle.Canvas(master = labelimage, width = 822,height=784)

screen=turtle.TurtleScreen(canvas)
screen.bgpic("img/map.gif")
screen.update()
t = turtle.RawTurtle(screen)
canvas.pack()

def movingTurtle(t,depart,dest):
    t.up()
    for v in mapTurtlePoint[depart][dest]:
        t.goto(pointRedirect(v['x'],v['y']))
        print(">",t.pos(),"원래:",v['x'],",",v['y'])
        t.down()
    t.up()
```

✓ 터틀 지도위에 나타내기

```
resultPane= tk.LabelFrame(window, text="경로정보")
resultPane.place(relx=0.8, relwidth=0.2, relheight=0.25)

label4=tk.Label(resultPane, text="거리(m)")
label4.pack()

label5=tk.Label(resultPane, text= " ")
label5.pack()

label6=tk.Label(resultPane, text="경로")
label6.pack()

label7=tk.Label(resultPane, text= " ", wraplength=200)
label7.pack()
```

✓ 결과창

```
findRoutePane= tk.LabelFrame(window, text="출발~도착지 설정")
findRoutePane.place(relx=0.8, relwidth=0.2, relheight=0.2)

label12=tk.Label(findRoutePane,text="출발지")
label12.pack()
values=list(mapXY.keys())
depart_combo=tk.ttk.Combobox(findRoutePane, values=values)
depart_combo.pack()
depart_combo.set("출발지 선택")

label13=tk.Label(findRoutePane,text="도착지")
label13.pack()
values=list(mapXY.keys())
dest_combo=tk.ttk.Combobox(findRoutePane, values=values)
dest_combo.pack()
dest_combo.set("도착지 선택")

def findPath():
    try:
        dpt=depart_combo.get()
        dst=dest_combo.get()

        tmp1=0
        tmp2=0
        for v in mapLocation:
            if(v==dpt):
                tmp1=1
            if(v==dst):
                tmp2=1
            if(tmp1==1 and tmp2==1):
                break
        if(tmp1!=1 or tmp2!=1):
            raise Exception('잘못된 입력입니다!')

        init()
        sht_f,course=dijkstra(mapLocation, dpt, dst)
        label15["text"]=round(sht_f, 1)

        t.clear()
        _course=""
        for i in range(len(course)):
            _course=_course+course[i]
            if i!=len(course)-1:
                movingTurtle(t, course[i], course[i + 1])
                _course = _course+"▶"
            if i%3==2:
                _course=_course+"\n"
        label17["text"]=_course
    except:
        label15["text"] = "잘못된 입력입니다!"
        label17["text"]=" "
```

✓ 출발지 도착지 입력받기



프로그램 시작 화면



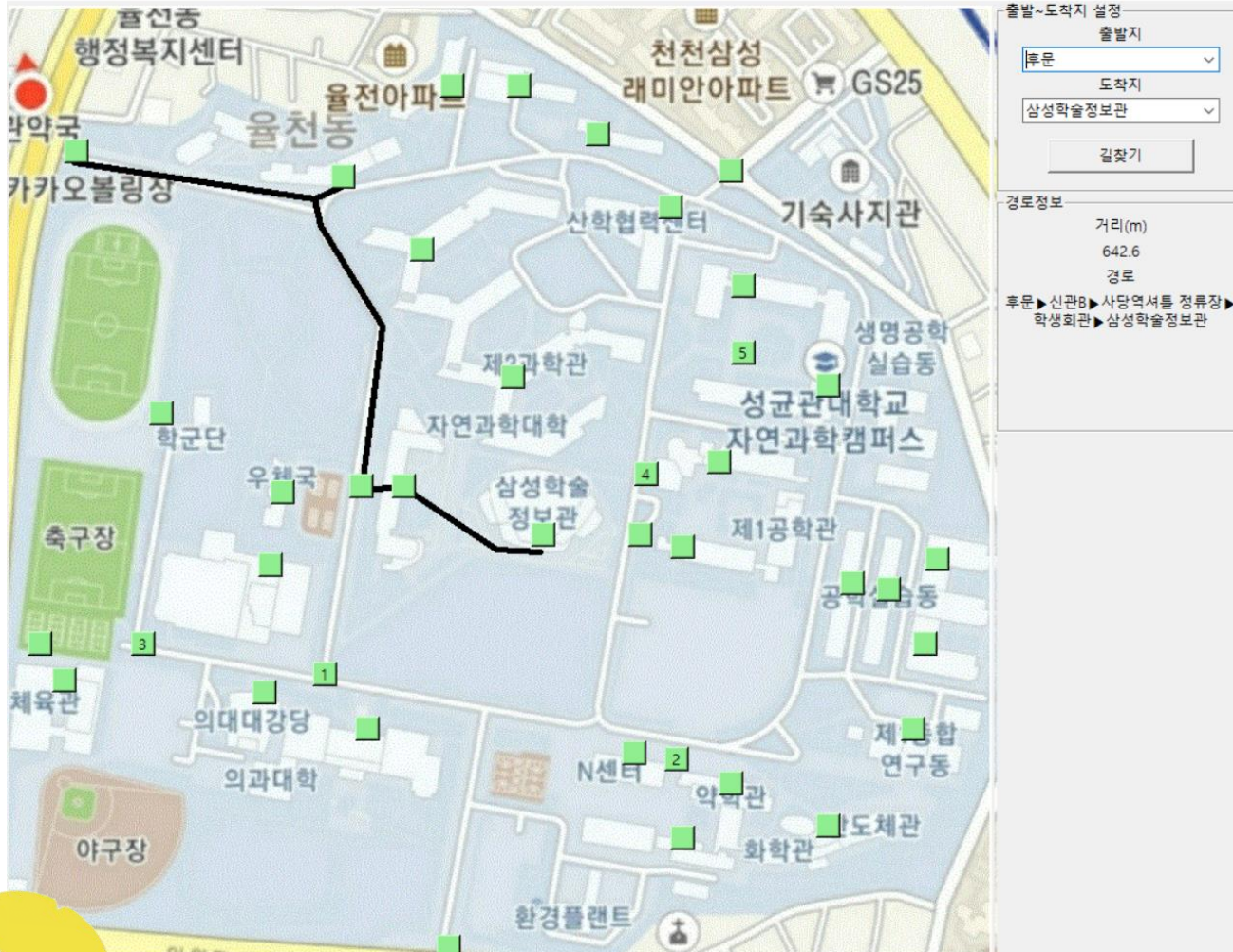
- ✓ 버튼을 눌러 출발지, 도착지 입력 가능
- ✓ 혹은 직접 건물 이름 입력 가능



- ✓ 창을 열어 출발지, 도착지 선택 가능

프로그램 실행 예시

명윤이와 함께하는 SKKU MAP



방법

- ✓ 출발지와 도착지 선택
 - 버튼을 눌러 선택하거나 창을 열어 선택 가능
- ✓ 길찾기 버튼 클릭

결과

- ✓ 최단 거리 & 경로가 경로 정보에 표시
- ✓ 터틀을 통해 눈에 보이는 경로를 지도에 안내



프로젝트에서 아쉬운 점

- ! 건물과 건물을 연결하였기 때문에 대략적인 정보만 전달 가능
- ! 구글 지도를 통해 거리를 측정하여 실제와는 오차가 발생함
- ! 학교 내부의 모든 길을 완벽하게 조사하지 못하여 아쉬움

