



타이타닉호 탐사선 실종사건의 해결방안

담당강정 : 3506 강지혁, 3511 임정섭

작품목적

최근 타이타닉호 관광 잠수정이 실종되는 사건이 발생하였다. 현재 소음이 감지된 곳을 기점으로 하여 수색이 이루어지고 있지만, 효과적인 결과를 얻지 못하고 있다. 따라서, 우리 팀은 관광 잠수정이 실종된 장소의 지구과학적 환경을 분석하고 물리학적으로 관광잠수정을 찾기 위한 방법을 고안하고자 한다. 또한, 앞으로 이러한 사건이 발생했을 때, 해결해줄 수 있는 프로그램을 개발하고자 한다.

융합분야

1) 물리학

물리학 분야의 파동과 관련하여 파동의 간섭과 회절, 파동에너지 등 여러 부분을 융합하였다. 또, 이러한 사건이 발생했을 때, 물 속을 효과적으로 이동하기 위해서는 잠수함의 모형이 유선형이고, 부력을 덜 받는 여러 조건들을 고려해주어야 하기 때문에 유체역학 분야와 융합하였다.

2) 지구과학

타이타닉호가 침몰해 있는 곳의 지형적 조건과 난류, 한류, 유속 등 여러 조건을 지구과학적으로 분석하였다.

3) 정보과학

탐색을 하는데 있어서 반복되게 진행하는 것을 코딩을 이용하여 표현하였다.

코드설명

1) **임포트** : vpython과 numpy를 임포트한다.

2) **3D 지형 생성 파라미터 설정** : 'width', 'height', 'depth' 변수를 통해서 가상의 3D 지형의 크기를 정의하고 terrain_height'을 통해서 지형의 기본 높이를 정의한다. 그리고 'object_velocity'를 통해서 움직이는 객체의 속도를 정의한다.

3) **vpthon 씬 생성** : 'canvas' 함수를 사용해서 vpython 씬을 생성한다.

4) **지면과 움직이는 객체 생성** : 'box' 함수를 이용해서 지면을 생성하고 'sphere' 함수를 사용해서 움직이는 객체를 생성한다.

5) **거리 계산 함수 정의** : 'calculate_distance_to_terrain' 함수를 통해 움직이는 객체와 지형 사이의 거리를 계산하도록 정의한다. 이때, 지형 밖에 있을 경우 'None'을 반환하도록 한다.

6) **초기 지형 생성 및 애니메이션 실행** : 'np.random.uniform' 함수를 사용해서 초기 지형 높이를 무작위로 생성하고, 'run_time' 동안 애니메이션을 실행시킨다.

7) **움직이는 객체 업데이트 및 거리 계산 및 출력** : 앞서 만든 함수를 이용해서 객체와 지형 사이의 거리를 계산하고 화면에 출력한다.

```
from vpython import *
import numpy as np

# 3D 지형 생성 파라미터 설정
width, height, depth = 50, 50, 20
terrain_height = 0.5 # 지형의 기본 높이
object_velocity = vector(0.1, 0.05, 0)

scene = canvas(title="Underwater Terrain", width=800, height=600)
scene.background = color.blue

floor = box(pos=vector(0, 0, -terrain_height-0.05), size=vector(50, 50, 0.1), color=color.green)

object_position = vector(0, 0, terrain_height + 0.5)
object_marker = sphere(pos=object_position, radius=0.2, color=color.red)

def calculate_distance_to_terrain(position, terrain_heights):
    x_idx = int(position.x) + width // 2
    y_idx = int(position.y) + height // 2

    if x_idx < 0 or x_idx >= width or y_idx < 0 or y_idx >= height:
        return None

    terrain_height = terrain_heights[x_idx][y_idx]
    distance = position.z - terrain_height

    return distance

# 초기 지형 생성
terrain_heights = np.random.uniform(0, terrain_height, size=(width, height))

# 실행 시간 설정 (초)
run_time = 10

# 현재 시간 초기화
current_time = 0
```

```
while current_time < run_time:
    rate(30)

    object_position += object_velocity

    if object_position.x < -width/2 or object_position.x > width/2 or #
        object_position.y < -height/2 or object_position.y > height/2:
        object_velocity *= -1

    distance_to_terrain = calculate_distance_to_terrain(object_position, terrain_heights)
    if distance_to_terrain is not None:
        output = f"Position: ({object_position.x:.2f}, {object_position.y:.2f}, {object_position.z:.2f}),
        Distance to terrain: {distance_to_terrain:.2f}"
        print(output)

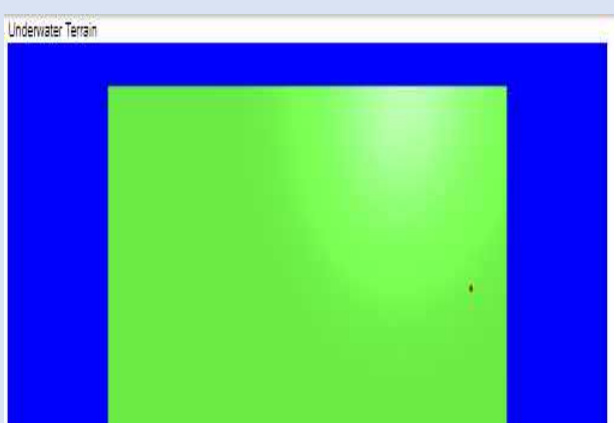
    x_idx = int(object_position.x) + width // 2
    y_idx = int(object_position.y) + height // 2

    if 0 <= x_idx < width and 0 <= y_idx < height:
        object_position.z = terrain_heights[x_idx][y_idx] + 0.5
        object_marker.pos = object_position

    current_time += 1/30 # 30 프레임당 1초씩 증가
```

결과화면

왼쪽 그림은 현재 가상의 잠수함이 움직이는 것을 표현한 시뮬레이션이고 오른쪽 그림은 가상의 잠수함이 주변 지형과의 거리를 계속해서 측정하고 있는 모습이다.



```
Position: (0.10, 0.05, 1.00), Distance to terrain: 0.64
Position: (0.20, 0.10, 0.88), Distance to terrain: 0.60
Position: (0.30, 0.15, 0.88), Distance to terrain: 0.60
Position: (0.40, 0.20, 0.88), Distance to terrain: 0.60
Position: (0.50, 0.25, 0.88), Distance to terrain: 0.60
Position: (0.60, 0.30, 0.88), Distance to terrain: 0.60
Position: (0.70, 0.35, 0.88), Distance to terrain: 0.60
Position: (0.80, 0.40, 0.88), Distance to terrain: 0.60
Position: (0.90, 0.45, 0.88), Distance to terrain: 0.60
Position: (1.00, 0.50, 0.88), Distance to terrain: 0.60
Position: (1.10, 0.55, 0.88), Distance to terrain: 0.78
Position: (1.20, 0.60, 0.67), Distance to terrain: 0.60
Position: (1.30, 0.65, 0.67), Distance to terrain: 0.60
Position: (1.40, 0.70, 0.67), Distance to terrain: 0.60
Position: (1.50, 0.75, 0.67), Distance to terrain: 0.60
Position: (1.60, 0.80, 0.67), Distance to terrain: 0.60
Position: (1.70, 0.85, 0.67), Distance to terrain: 0.60
Position: (1.80, 0.90, 0.67), Distance to terrain: 0.60
```

느낀 점

타이타닉호 탐사선이 실종되어서 안에 타고 있던 여러 연구진들이 실종되어 결국 죽었다는 사실을 알게 된 이후, 이러한 사건을 없애기 위해 코드를 직접 제작하였고, 우리가 만든 프로그램을 이용해 해양 실종사건 같은 문제를 해결할 수 있게 되면 좋겠다.