Busan science high school

2023 Ocean ICT Festival
2023 BOIF

C 12

QR 코드 영역
QR 삽입 후
테두리 삭제

Youtube 영상 QR

기름유출사고를 대비한 쉐코아크의 가장 효율적인 동선 시각화 및 예측

아크, 기름을 없애자
2409문서준, 2411양우진, 2417하민석

작품 제작 동기

저희 팀은 우연히 인터넷 기사에서 기름을 회수하는 로봇인 쉐코 아크를 접하게 되었습니다. 기름유출사고의 경우에는 조기에 빠르게 기름을 제거하는 것이 중요하기 때문에 저희 팀은 기름을 빠르게 수거할 수 있도록 쉐코아크의 가장 효율적인 동선을 시각화하고 예측하고자 하였습니다.

융합분야

쉐코 아크란 무엇일까?

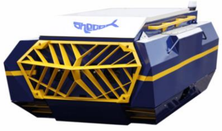
✓ 소형 기름 유출사고에서 사용할 수 있는 소형 방제 로봇

✓ 50kg대의 무게로 100L의 기름을 흡수하여 제거할 수 있음

✓ 기존의 기름 제거장비들은 대형사고에 최적화되어있어 소형사고에 사용하기에 비효율적이지만 쉐코아크는 소형사고에 최적화된 기름제거 로봇임

✓ 기존의 방제작업에 사용한 흡착포의 소각으로 발생하는 환경오염 문제도 해결할 수 있음

✓ 기존의 오일펜스가 늦게 펴져서 기름이 더 확산되는 문제를 해결함



1. 제품이송

<쉐코아크의 사용 과정>

2. 제품투입

3. 기름회수

4. 기름배출

코드설명

```
import pygame
import sys
import numpy as np
from pygame.locals import *
import time
pygame.init()
pygame.font.init()
```

1.프로그램에 필요한 리소스 불러옴

```
height = 600
width = 800

SURF = pygame.display.set_mode((width, height))

WHITE = (255, 255, 255)
RED = (255, 51, 51)
BLUE = (51, 51, 255)
BLACK = (0, 0, 0)
GRAY = (120, 120, 120)
YELLOW = (206, 191, 1)
e= (float(input("해류의 방향을 입력해주세요(도 단위로): ")))
e = np.deg2rad(e)
g = (float(input("표층 해류의 속도를 입력해주세요(m/s): ")))
```

2.프로그램에 필요한 상수를 정의하고 필요한 값들을 입력받음

```
class Vec2:
    def __init__(self, x, y):
        self.x = x-np.cos(e)*g
        self.y = y-np.sin(e)*g


    def __add__(self, other):
        return Vec2(self.x + other.x, self.y + other.y)

    def __mul__(self, num):
        return Vec2(num * self.x, num * self.y)

    def __neg__(self):
        return Vec2(-self.x, -self.y)

    def __sub__(self, other):
        return Vec2(self.x - other.x, self.y - other.y)

    def __repr__(self):
        return f"<{self.x}, {self.y}>"

    def __abs__(self):
        return (self.x ** 2 + self.y ** 2) ** 0.5

    def dot(self, other):
        return self.x * other.x + self.y * other.y

    def unit(self):
        return self * (1 / abs(self))

    def st(self):
        return self.x + 400, 300 - self.y


def sigmoid(x):
    return 20 * (1 / (1 + np.exp(-x * 100000 + 3)))
```

3. 벡터의 연산을 정의해주는 코드이다

<결과>

```
해류의 방향을 입력해주세요(도 단위로): 30
표층 해류의 속도를 입력해주세요(m/s): 2
기름이 퍼지는 속도를 대략적으로 입력하시오[cm/s] : 3

해당 시뮬레이션은 기름이 퍼지는 모습을 시각화하는 시뮬레이션입니다.


빨간점은 무작위로 퍼져나가는 기름

회색점은 쉐코아크의 가장 효율적인 동선

파란점은 본 선박의 위치를 나타낸 것입니다
```

```
class Field_arrow:
    objlist = []

    def __init__(self, pos):
        self.pos = pos
        self.field = Vec2(0, 0)
        self.objlist.append(self)

    def update(self):
        self.field = Vec2(0, 0)
        for p in Particle.objlist:
            r = self.pos - p.pos
            if not abs(r):
                return
            self.field += Vec2.unit(r) * p.charge * abs(r) ** -2
        if not abs(self.field):
            return
        self.field = Vec2.unit(self.field) * sigmoid(abs(self.field))

    def show(self):
        pygame.draw.aaline(SURF, RED, Vec2.st(self.pos), Vec2.st(self.pos + self.field * 0.8), 2)
        pygame.draw.aaline(SURF, BLUE, Vec2.st(self.pos + self.field * 0.8), Vec2.st(self.pos + self.field), 2)


def make_field():
    for x in range(-400, 400, 17):
        for y in range(-300, 300, 17):
            Field_arrow(Vec2(x, y))

make_field()
```

4.벡터장을 정의하고 생성해주는 코드이다

```
class Particle:
    objlist = []

    def __init__(self, charge, mass, pos, vel):
        self.charge = charge

        if charge > 0:
            self.color = RED
        elif charge < 0:
            self.color = BLUE
        else:
            self.color = GRAY
        self.m = mass
        self.radius = 5
        self.pos = pos
        self.vel = vel
        self.acc = Vec2(0, 0)

        self.objlist.append(self)

    def show(self):
        pygame.draw.circle(
            SURF,
            self.color,
            (int(self.pos.x + width / 2), int(height / 2 - self.pos.y)),
            self.radius,
        )
```

5.장 속에서 움직이는 입자를 정의해주는 코드이다

```
class 등속직선운동:
    def __init__(self, mass, velocity, charge):
        pygame.display.set_caption('등속직선운동')
        self.charge = charge
        self.velocity = velocity
        self.mass = mass
        print('')
        time.sleep(1)
        print('해당 시뮬레이션은 기름이 퍼지는 모습을 시각화하는 시뮬레이션입니다.')
        print('')
        print('')
        print('빨간점은 무작위로 퍼져나가는 기름')
        print('')
        print('회색점은 쉐코아크의 가장 효율적인 동선')
        print('')
        print('파란점은 본 선박의 위치를 나타낸 것입니다')
        time.sleep(3)

        for _ in range(30):
            angle = np.random.uniform(0, 2 * np.pi)
            initial_velocity = Vec2(50,0).rot(angle) + velocity*np.cos(e)*g, np.sin(angle) + velocity*np.sin(e)*g)
            Particle(charge, 1, Vec2(0, 0), initial_velocity)

        Particle(0, 1, Vec2(0, 0), Vec2(-np.cos(e)*g, -np.sin(e)*g))
        Particle(-1, 1, Vec2(0, 0), Vec2(+np.cos(e)*g, +np.sin(e)*g))

        while True:
            for event in pygame.event.get():
                if event.type == QUIT:
                    pygame.quit()
                    sys.exit()
            SURF.fill(WHITE)
            for g in Field_arrow.objlist:
                g.update()
                g.show()
            for particle in Particle.objlist:
                particle.pos = particle.pos + particle.vel
                particle.show()

            pygame.display.update()
```

6,움직이는 기름의 운동을 정의해주는 코드이다

```
v = float(input("기름이 퍼지는 속도를 대략적으로 입력하시오[cm/s] : "))
v = (1/10) * v
c = 3
m=1
등속직선운동(m, v, c)
```

7.결과를 출력해주는 코드이다

예상되는 기대효과

1. 기름을 빠르게 제거하여 기름이 더 넓은 범위로 확산되는 것을 막을 수 있다.
2. 기름을 제거하는데 드는 비용과 인력을 줄일 수 있다.