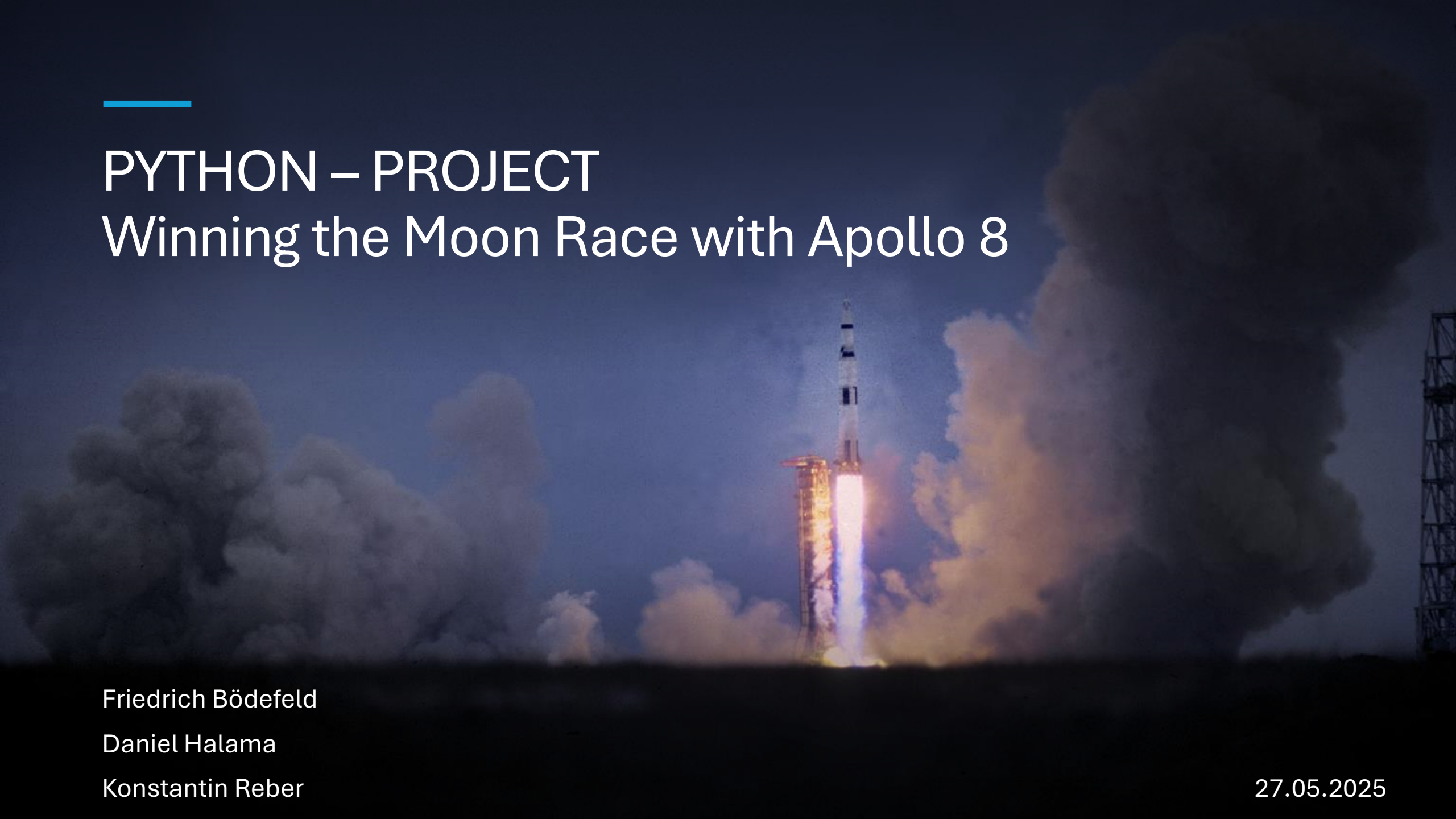

PYTHON – PROJECT

Winning the Moon Race with Apollo 8



Friedrich Bödefeld

Daniel Halama

Konstantin Reber

27.05.2025

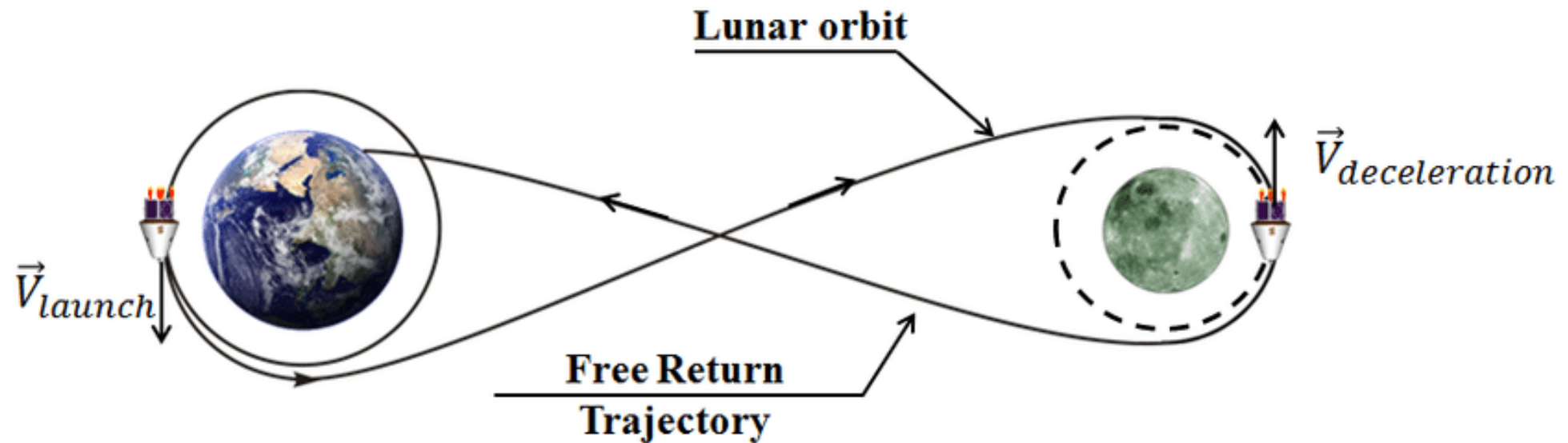
A historical color photograph of John F. Kennedy speaking at a wooden podium. To his left is a large American flag. The podium features the Seal of the President of the United States. In the foreground, several men in suits are seated at a table, looking towards the speaker. The background shows a large, tiered stadium filled with a crowd of people under a clear sky.

INTRODUCTION

“We choose to go to the Moon in this decade and do the other things, not because they are easy, but because they are hard.” - JFK

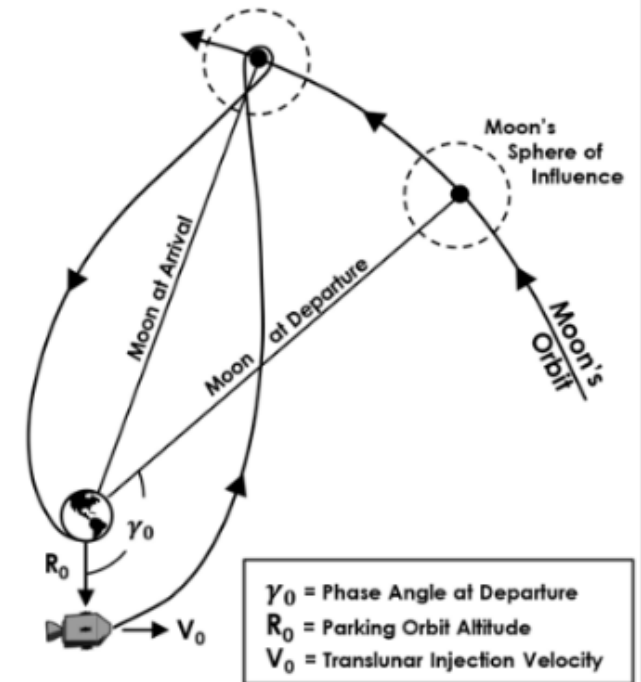
TASK

- Showcase the „Free Return Trajectory“ around the moon



DIFFICULTIES

- Apollo 's 8 „Free Return Trajectory“



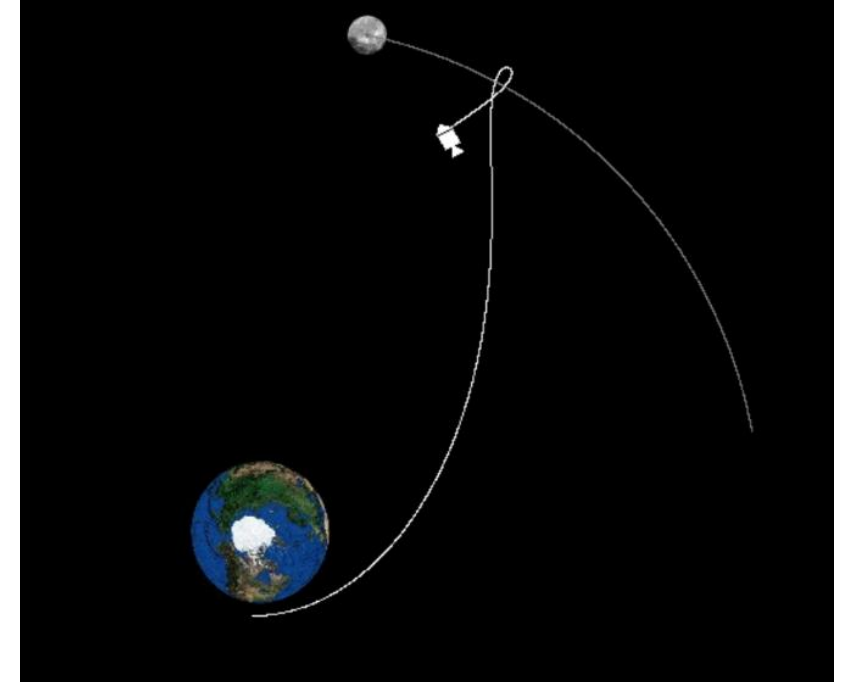
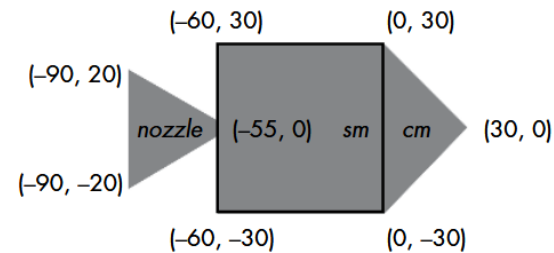
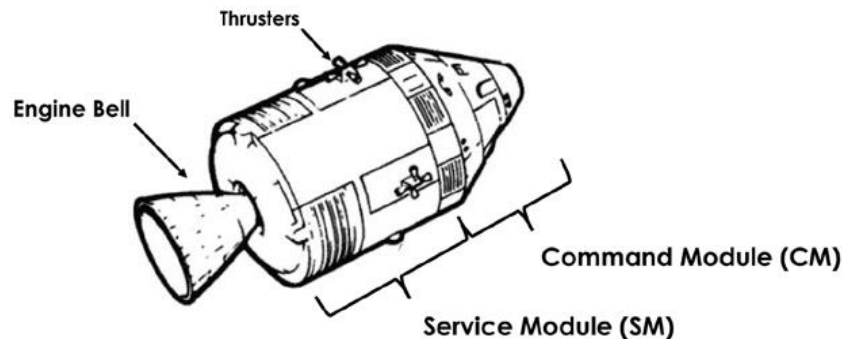


DIFFICULTIES

- Three Body Problem

APPROACH

- Drawing board module – Turtle
- Gravitation influence
- Terrestrial bodies
- Space ship
- Calculation of movement



CODE

- Turtle Import an initialization of variables

```
1  from turtle import Shape, Screen, Turtle, Vec2D as Vec
2
3  G = 8
4  NUM_LOOPS = 4100
5  Ro_X = 0
6  Ro_Y = -85
7  Vo_X = 485
8  Vo_Y = 0
```

CODE

- Gravitational System Class

```
10 class GravSys():
11     def __init__(self):
12         self.bodies = []
13         self.t = 0
14         self.dt = 0.001
15
16     def sim_loop(self):
17         for _ in range(NUM_LOOPS):
18             self.t += self.dt
19             for body in self.bodies:
20                 body.step()
```


CODE

- Initialization of Terrestrial bodies
 - Force and vector

```
24 class Body(Turtle):
25     def __init__(self, mass, start_loc, vel, gravsys, shape):
26         super().__init__(shape=shape)
27         self.gravsys = gravsys
28         self.penup()
29         self.mass = mass
30         self.setpos(start_loc)
31         self.vel = vel
32         gravsys.bodies.append(self)
```

```
35     def acc(self):
36         a = Vec(0, 0)
37         for body in self.gravsys.bodies:
38             if body != self:
39                 r = body.pos() - self.pos()
40                 a += (G * body.mass / abs(r)**3) * r
41         return a
```

CODE

- Movement calculation

```
def step(self):  
    """Calculate position, orientation, and velocity of a body."""  
    dt = self.gravsys.dt  
    a = self.acc()  
    self.vel = self.vel + dt * a  
    self.setpos(self.pos() + dt * self.vel)  
  
    rotate_factor = 0  
    if self.gravsys.bodies.index(self) == 2: # Index 2 = CSM.  
        rotate_factor = 0.0006  
        self.setheading((self.heading() - rotate_factor * self.xcor()))  
    if self.xcor() < -20 and self.gravsys.bodies.index(self) == 2:  
        self.shape('arrow')  
        self.shapesize(0.5)  
        self.setheading(105)
```

CODE

- Setup screen and Attributes of Earth and Moon

```
def main():  
    screen = Screen()  
    screen.setup(width=1.0, height=1.0) # For fullscreen.  
    screen.bgcolor('black')  
    screen.title("Apollo 8 Free Return Simulation")  
    gravsys = GravSys()
```

```
image_earth = 'earth_100x100.gif'  
screen.register_shape(image_earth)  
earth = Body(1000000, (0, -25), Vec(0, -2.5), gravsys, image_earth)  
earth.pencolor('white')  
earth.getscreen().tracer(n=0, delay=0)  
image_moon = 'moon_27x27.gif'  
screen.register_shape(image_moon)  
moon = Body(32000, (344, 42), Vec(-27, 147), gravsys, image_moon)  
moon.pencolor('gray')
```


CODE

- Space Ship

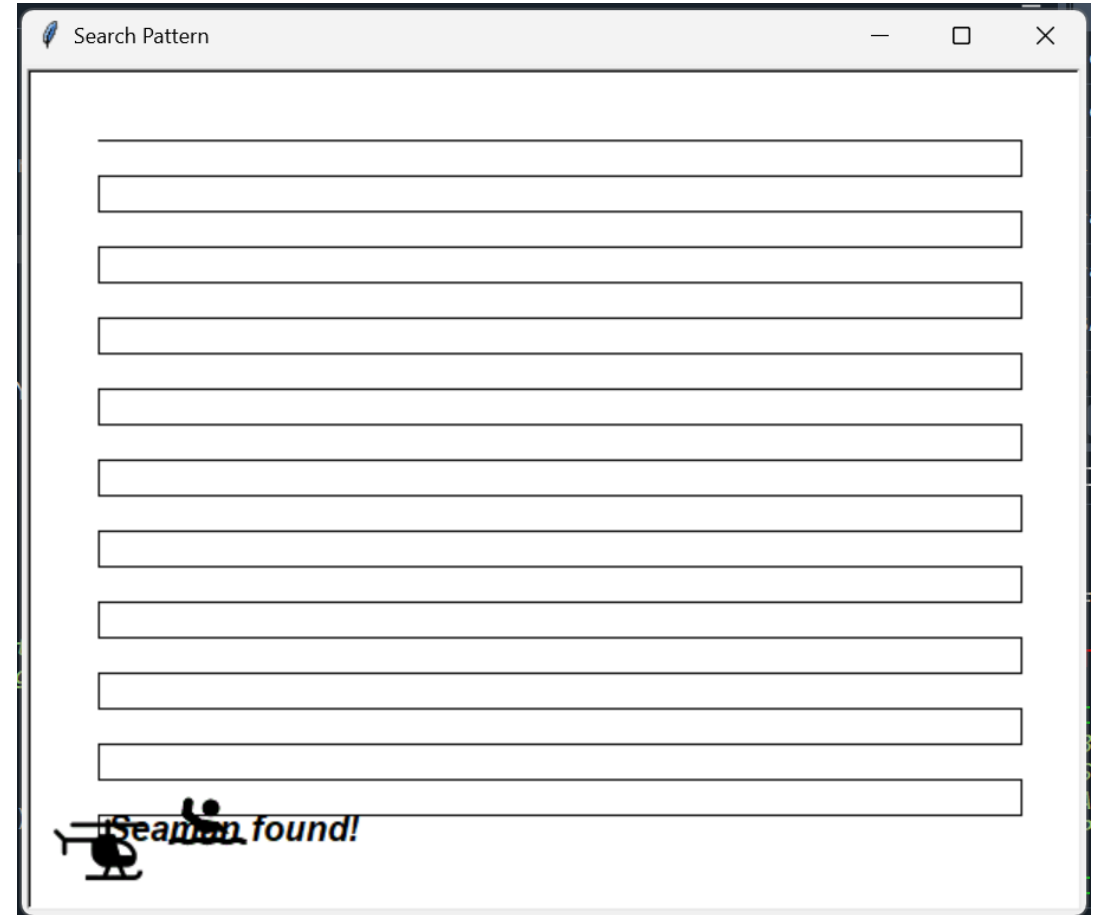
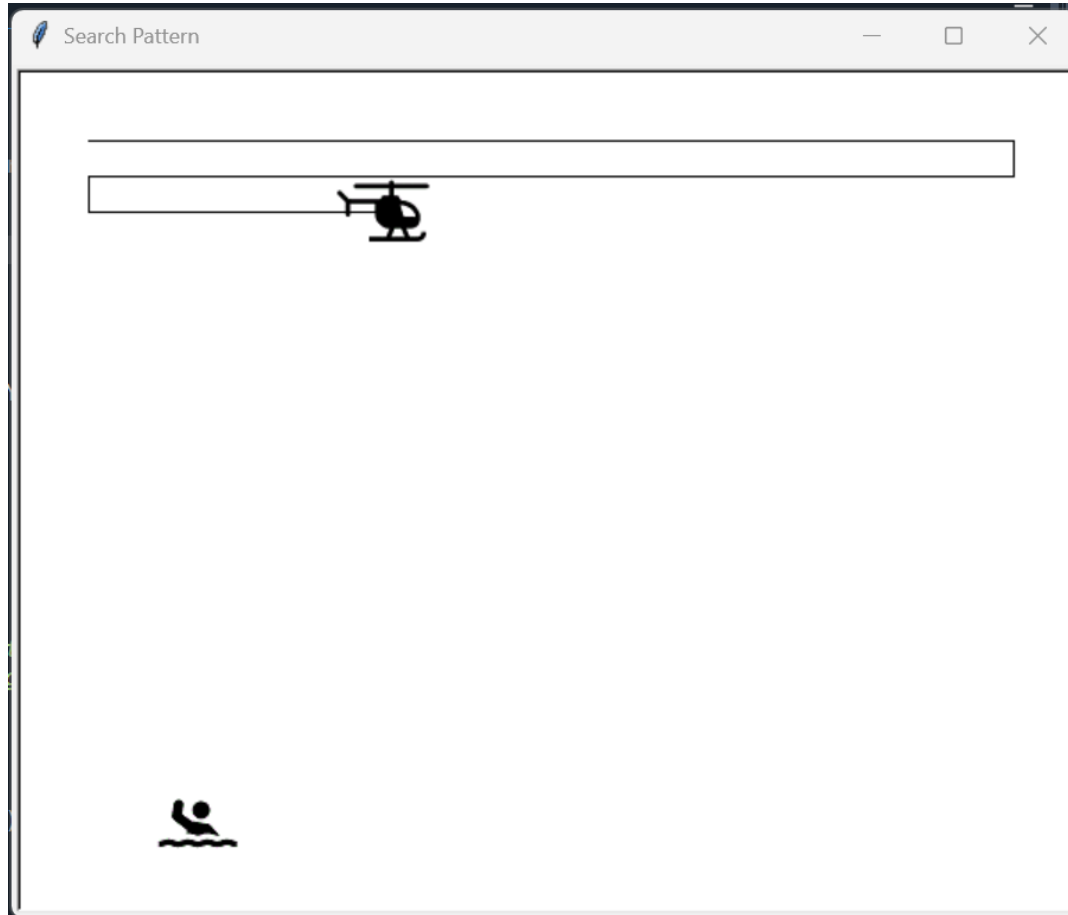
```
csm = Shape('compound')
cm = ((0, 30), (0, -30), (30, 0))
csm.addComponent(cm, 'white', 'white')
sm = ((-60, 30), (0, 30), (0, -30), (-60, -30))
csm.addComponent(sm, 'white', 'black')
nozzle = ((-55, 0), (-90, 20), (-90, -20))
csm.addComponent(nozzle, 'white', 'white')
screen.register_shape('csm', csm)

ship = Body(1, (Ro_X, Ro_Y), Vec(Vo_X, Vo_Y), gravsys, 'csm')
ship.shapesize(0.2)
ship.color('white')
ship.getscreen().tracer(1, 0)
ship.setheading(90)
gravsys.sim_loop()
```

ADDITIONAL TASK



ADDITIONAL TASK - HELICOPTER



THANK YOU FOR THE ATTENTION

