ELEVATOR SIMULATION

UNIVERSITÉ DE BOURGOGNE

SOFTWARE ENGENEERING

ATANDA ADEWALE
QIAN ZHILING

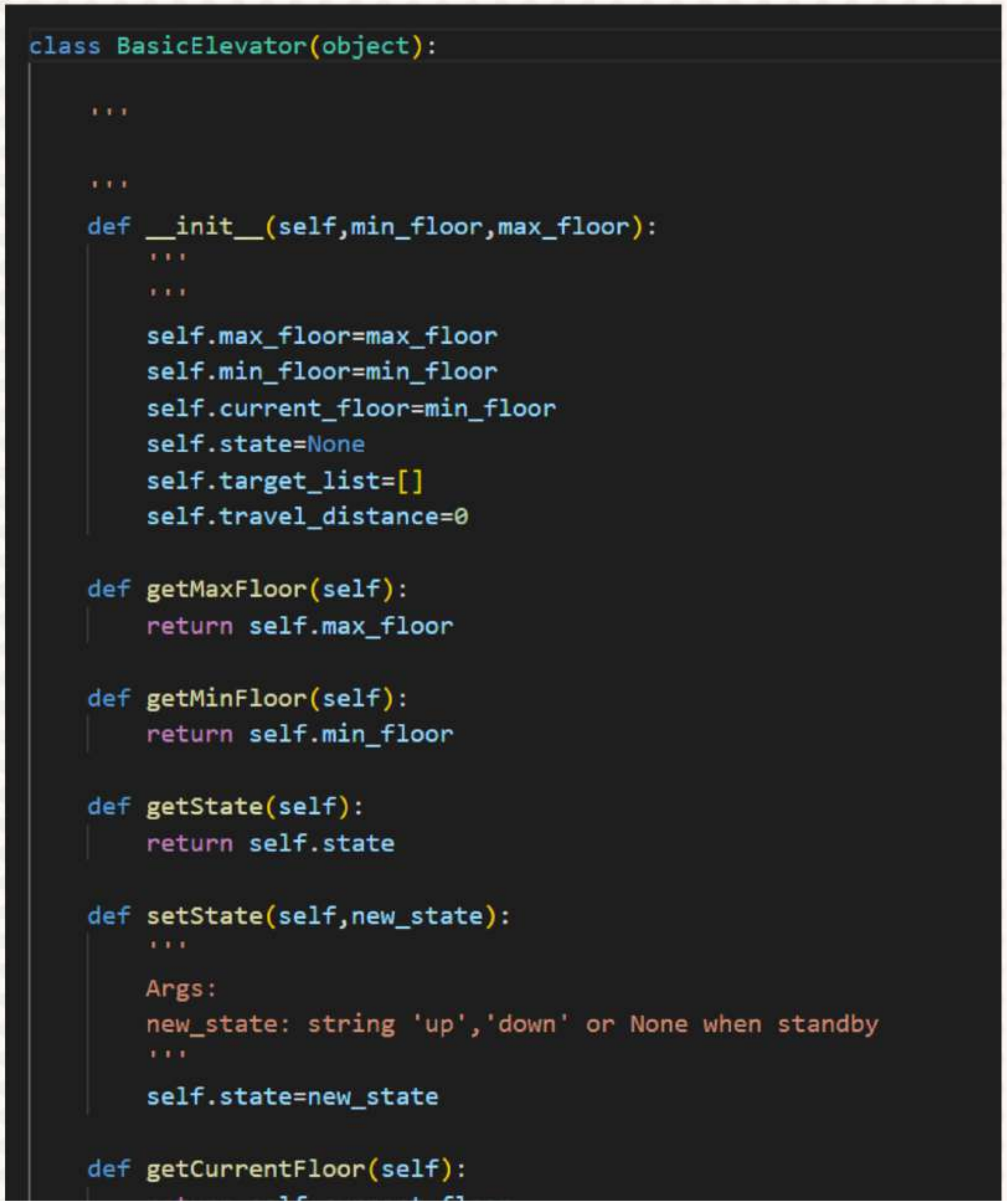
CLASS

THE CAR

An elevator or lift is a cable-assisted, hydraulic cylinder-assisted, or roller-track assisted machine that vertically transports people or freight between floors, levels, or decks of a building, vessel, or other structure.

The car is an elevator component that carries passengers and goods and is the working part of the elevator. The car consists of a car frame and a car body.

- Identify the current position
- Allows up to 8 Passengers
- Carrier



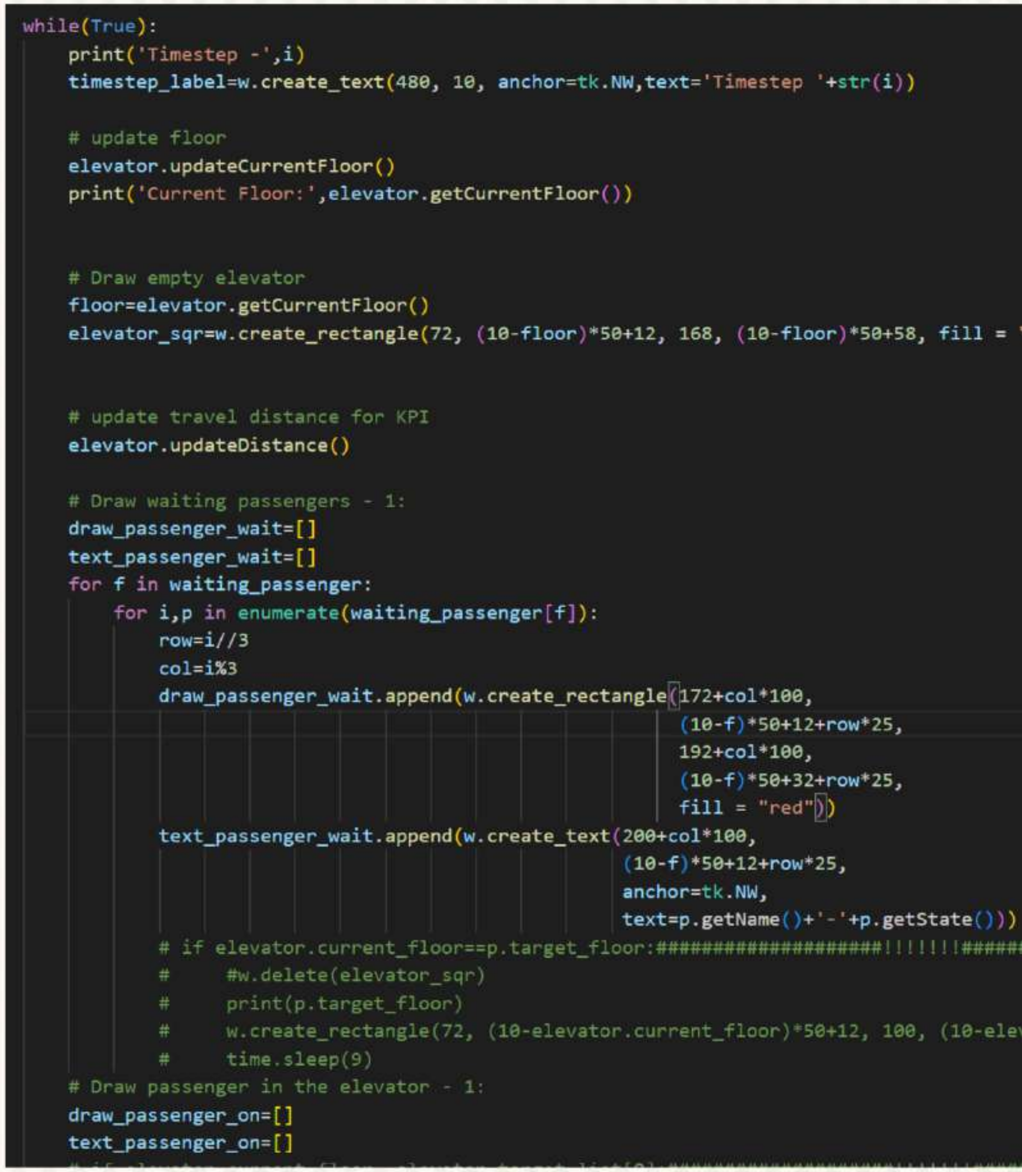
```python
class BasicElevator(object):

    '''

    '''
    def __init__(self,min_floor,max_floor):
        '''
        '''
        self.max_floor=max_floor
        self.min_floor=min_floor
        self.current_floor=min_floor
        self.state=None
        self.target_list=[]
        self.travel_distance=0

    def getMaxFloor(self):
        return self.max_floor

    def getMinFloor(self):
        return self.min_floor

    def getState(self):
        return self.state

    def setState(self,new_state):
        '''
        Args:
        new_state: string 'up','down' or None when standby
        '''
        self.state=new_state

    def getCurrentFloor(self):
```

CLASS

CONTROL PANEL

Controller is a system to control the elevators and the elevator control panel is a cabin that holds all of the components to control the elevator. The elevator control panel holds all the power supply units, orchestrates all the operations, and ensures safety.

- Identify the current position
- Measure Targets Position
- Avoid Starvation



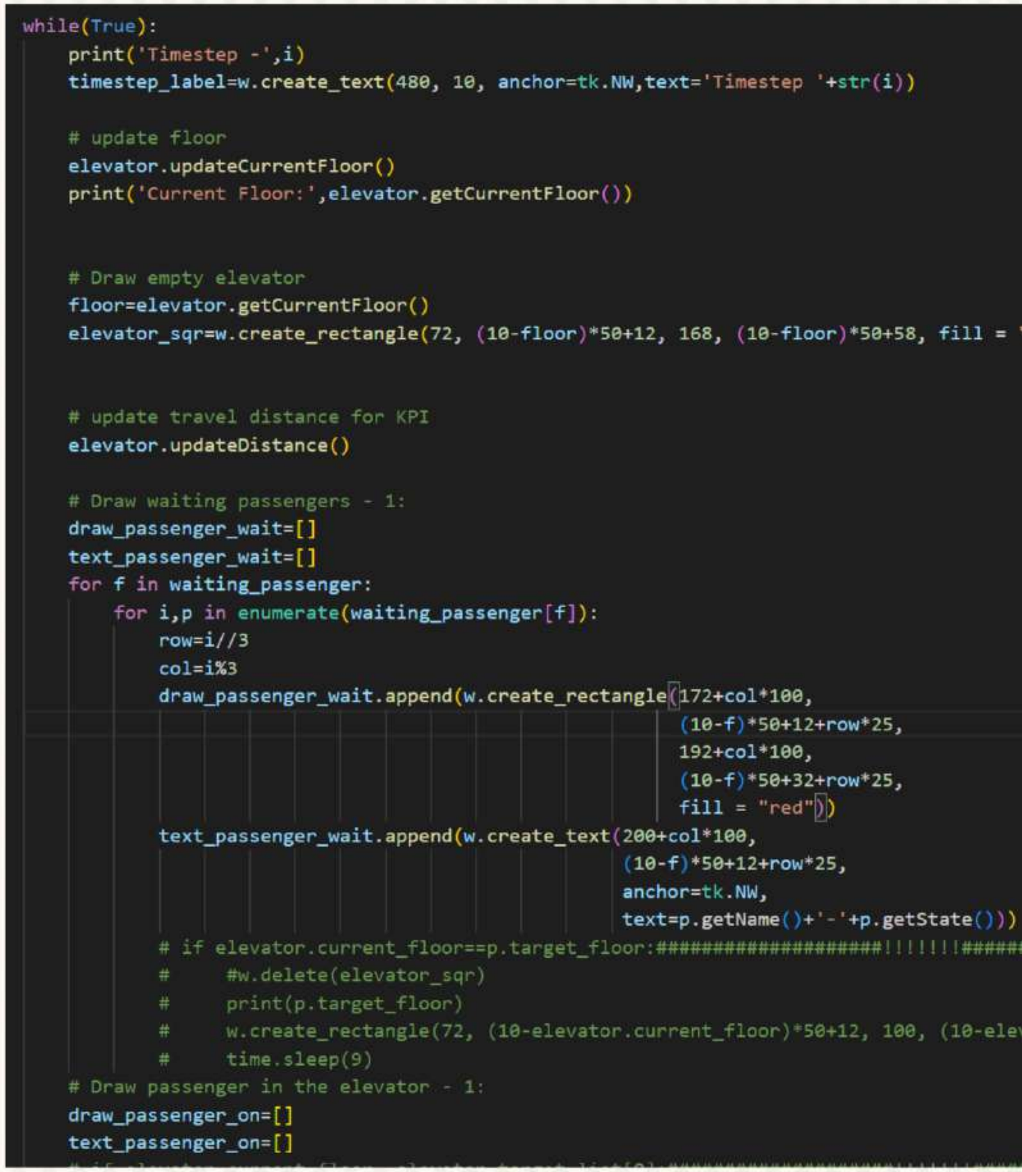
```python
while(True):
    print('Timestep -',i)
    timestep_label=w.create_text(480, 10, anchor=tk.NW,text='Timestep '+str(i))

    # update floor
    elevator.updateCurrentFloor()
    print('Current Floor:',elevator.getCurrentFloor())


    # Draw empty elevator
    floor=elevator.getCurrentFloor()
    elevator_sqr=w.create_rectangle(72, (10-floor)*50+12, 168, (10-floor)*50+58, fill =


    # update travel distance for KPI
    elevator.updateDistance()

    # Draw waiting passengers - 1:
    draw_passenger_wait=[]
    text_passenger_wait=[]
    for f in waiting_passenger:
        for i,p in enumerate(waiting_passenger[f]):
            row=i//3
            col=i%3
            draw_passenger_wait.append(w.create_rectangle(172+col*100,
                                                          (10-f)*50+12+row*25,
                                                          192+col*100,
                                                          (10-f)*50+32+row*25,
                                                          fill = "red"))
            text_passenger_wait.append(w.create_text(200+col*100,
                                                     (10-f)*50+12+row*25,
                                                     anchor=tk.NW,
                                                     text=p.getName()+'-'+p.getState()))
            # if elevator.current_floor==p.target_floor:####################!!!!!!!#####
            #     #w.delete(elevator_sqr)
            #     print(p.target_floor)
            #     w.create_rectangle(72, (10-elevator.current_floor)*50+12, 100, (10-ele
            #     time.sleep(9)
    # Draw passenger in the elevator - 1:
    draw_passenger_on=[]
    text_passenger_on=[]
```

CLASS

PASSENGER

People represented with a small red box who are moving from one floor to another easily without effort.

- Generates and Update targets state
- Defines 'Get on and Drop'
- Calculates time



```python
while(True):
    print('Timestep -',i)
    timestep_label=w.create_text(480, 10, anchor=tk.NW,text='Timestep '+str(i))

    # update floor
    elevator.updateCurrentFloor()
    print('Current Floor:',elevator.getCurrentFloor())


    # Draw empty elevator
    floor=elevator.getCurrentFloor()
    elevator_sqr=w.create_rectangle(72, (10-floor)*50+12, 168, (10-floor)*50+58, fill = '


    # update travel distance for KPI
    elevator.updateDistance()

    # Draw waiting passengers - 1:
    draw_passenger_wait=[]
    text_passenger_wait=[]
    for f in waiting_passenger:
        for i,p in enumerate(waiting_passenger[f]):
            row=i//3
            col=i%3
            draw_passenger_wait.append(w.create_rectangle(172+col*100,
                                                          (10-f)*50+12+row*25,
                                                          192+col*100,
                                                          (10-f)*50+32+row*25,
                                                          fill = "red"))
            text_passenger_wait.append(w.create_text(200+col*100,
                                                     (10-f)*50+12+row*25,
                                                     anchor=tk.NW,
                                                     text=p.getName()+'-'+p.getState()))
            # if elevator.current_floor==p.target_floor:####################!!!!!!!######
            #     #w.delete(elevator_sqr)
            #     print(p.target_floor)
            #     w.create_rectangle(72, (10-elevator.current_floor)*50+12, 100, (10-ele
            #     time.sleep(9)
    # Draw passenger in the elevator - 1:
    draw_passenger_on=[]
    text_passenger_on=[]
```

THANK YOU