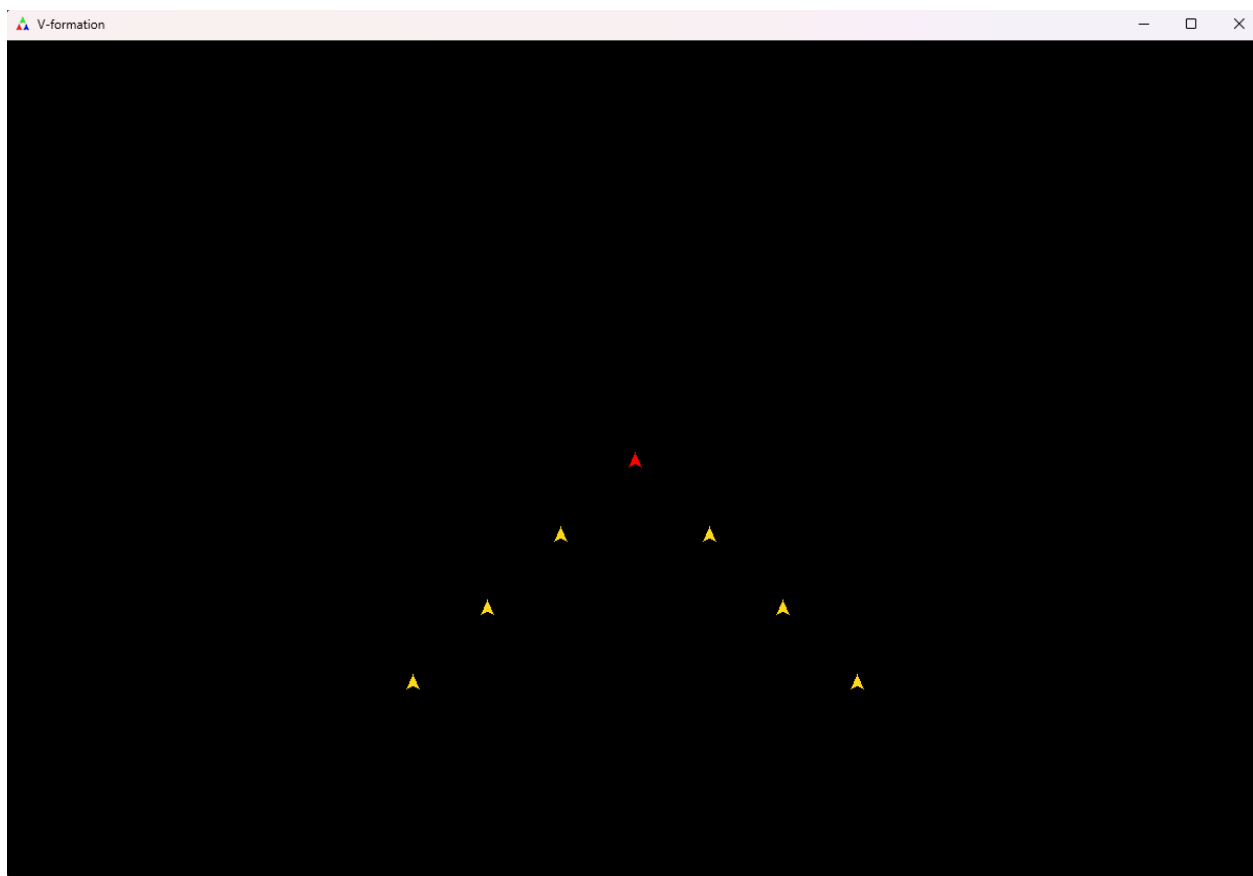




Báo cáo môn Robot Phân tán 5

Type	Assignment
Done	<input type="checkbox"/>
Topic	

Bài tập: mô phỏng điều khiển đội hình chữ V (V-shaped formation) di chuyển theo chuột máy tính.



Trạng thái ban đầu đội hình chữ V

Bài toán này sẽ có:

- Một con dẫn đầu (màu đỏ) làm nhiệm vụ dẫn đường
- Các con lân cận (màu vàng) sẽ bám theo con gần nhất

Các con chim sẽ tạo thành hình chữ V dựa vào thông tin xung quanh

```
class Bird(pg.sprite.Sprite):  
  
    def __init__(self, grid, drawSurf, isFish=False, isLead=False,  
        ...
```

Ở đây có biến isLead để biểu diễn con chim nào là leader.

```
def getcurrentbird(self,maxW,maxH,isLead):  
    if isLead:  
        self.appear.append((maxW//2,maxH//2))  
        return (maxW//2,maxH//2)  
    else:  
        basebird=self.getparentbird(isLead)  
        dx=abs(sin(radians(self.angle))*self.distance)  
        dy=abs(cos(radians(self.angle))*self.distance)  
        tempy=basebird[1]+dy  
        if len(self.appear)%2==1:  
            tempx=basebird[0]-dx  
        else:  
            tempx=basebird[0]+dx  
        self.appear.append((tempx,tempy))  
        return (tempx,tempy)
```

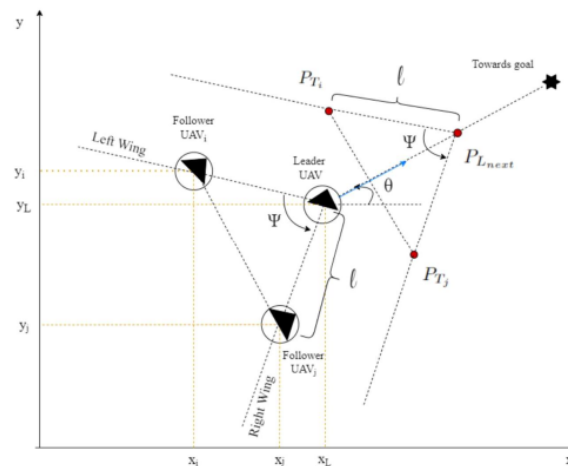
Hàm này có nhiệm vụ hình thành đội hình ban đầu

Quy trình chạy

Bước 1: Lập đội hình chữ V ban đầu

Bước 2: Con dẫn đầu sẽ đi theo chuột

Bước 3: Các con còn lại sẽ đi theo con trước và giữ nguyên hình chữ. Chúng sẽ dùng điểm ảo để đi theo. Hướng đi theo con phía trước.



Hình minh họa điểm ảo (điểm màu đỏ)

***Note:** Khi chuột dừng thì các con cùng dừng lại

Giải thích

Để xác định điểm ảo, em sẽ dựa vào điểm leader và các cha mẹ của các con chim. Em sẽ dùng ma trận chuyển đổi từ tọa độ khung hình sang tọa độ từng con để xác định điểm cho con kế tiếp nối đuôi cũng như tránh va chạm rồi đổi lại sang tọa độ khung vì em làm theo tọa độ khung.

```
def rotate(self, isLead):
    if isLead:
        self.tDistance, self.angle = (pg.mouse.get_pos() - self.pos).polar()
        self.vel.from_polar((self.speed, self.angle))
        self.image = pg.transform.rotozoom(self.orig_image, self.angle, 1)
        self.rect = self.image.get_rect(center=self.rect.center)
    else:
        self.tDistance, self.angle = (self.getvirtualpoint() - self.pos).polar()
        self.vel.from_polar((self.speed, self.angle))
        self.image = pg.transform.rotozoom(self.orig_image, self.angle, 1)
        if self.tDistance < self.bSize*2 :
```

```
self.image = pg.transform.rotozoom(self.orig_image, 1, 1)  
self.rect = self.image.get_rect(center=self.rect.center)
```

Ở đây python hỗ trợ 2 hàm cho việc đó là `as_polar()` và `from_polar()` để chuyển đổi qua lại giữa tọa độ cực và tọa độ Cartesia bởi bản chất nó hoạt động như dùng ma trận chuyển đổi.

Video mô phỏng

chạy file v-formation.mp4