环境：

Agent找到花蜜并吃掉

目标：

Agent必须通过调整自身姿态（yaw/pitch/speed/position）来使得鸟喙能够进入花蜜中并且不会卡住。

Agent设置：

环境包含一个Agent

Agent奖惩规则：

吃到花蜜+0.01，根据鸟喙与花开口朝向夹角奖励0~+0.02（正对着花开口吃奖励最高）

```
private void OnTriggerStay(Collider collider)
    {
        if (collider.CompareTag("Nectar"))
        {
            Vector3 closePointToBeakTip =
collider.ClosestPoint(BeakTipCenterPosition);

            //表示鸟喙能够吃到花蜜
            if (Vector3.Distance(closePointToBeakTip, BeakTipCenterPosition) <
BeakTipRadius)
            {
                //找到花蜜碰撞体对应的Flower花
                Flower flower = flowerArea.GetFlowerFromNectar(collider);
                //尝试去吃掉0.01f的花蜜。
                float nectarReceived = flower.Feed(.01f);
                NectarObtained += nectarReceived;
                if (trainingMode)
                {

                    float forwardAlignment =
Vector3.Dot(transform.forward.normalized, -
nearestFlower.FlowerUpVector.normalized);
                    //基础奖励0.01f，如果是正对着花进行采食，额外加最多0.02f分。
                    float bonus = .02f * Mathf.Clamp01(forwardAlignment);
                    float baseIncrement = .01f;
                    float increment = baseIncrement + bonus;
                    AddReward(increment);
                }
            }
            //记得更新flower
            if (!nearestFlower.HasNectar)
            {
                UpdateNearestFlower();
            }
        }
    }
```

触碰到边界、墙壁、石头、地面-0.5

```
 private void OnCollisionEnter(Collision collision)
    {
        if (collision.collider.CompareTag("Boundary") && trainingMode)
        {
            //当撞上固体边界：比如树木、地，给智能体负的反馈
            AddReward(-0.5f);

        }
    }
```

行为参数

向量观测空间：观测

```
 public override void CollectObservations(VectorSensor sensor)
    {
        //sensor.AddObservation(观测数据)用于将观测数据添加到智能体感知器，用于训练智能体

        //当最近的花还没有设置出来的时候，要传递进去一个空的10维float数组
        if (nearestFlower == null)
        {
            sensor.AddObservation(new float[10]);
            return;
        }
        //添加：相对于父物体的局部旋转，即相对于小岛的旋转（4）
        //单位四元数是长度为1的四元数，用于表示旋转方向
        Quaternion relativeRotation = transform.localRotation.normalized;

        //添加：指向花的向量(3)
        Vector3 toFlower = nearestFlower.FlowerCenterPosition -
BeakTipCenterPosition;
        //toFlower.Normalize();

        //添加：判断身体是否朝向花开口(+1代表直接在花面前，-1代表在花后面）(1)
        //用向量点乘，A dot B > 0表示朝向相同，表示面朝花。 <0相反。为0则垂直
        float positionAlignment = Vector3.Dot(toFlower.normalized, -
nearestFlower.FlowerUpVector.normalized);

        //添加：判断是否鸟喙朝向花开口(正，则表示鸟喙朝向花开口）(1)
        float beakTipAlignment = Vector3.Dot(beakTip.forward.normalized, -nearestFlower.FlowerUpVector.normalized);

        //添加：鸟喙到花的相对（相对小岛）距离（1）
        float relativeDistance = toFlower.magnitude / FlowerArea.areaDiameter;

        sensor.AddObservation(relativeRotation);
        sensor.AddObservation(toFlower.normalized);
        sensor.AddObservation(positionAlignment);
        sensor.AddObservation(beakTipAlignment);
        sensor.AddObservation(relativeDistance);
        //总共10个观察

    }
```