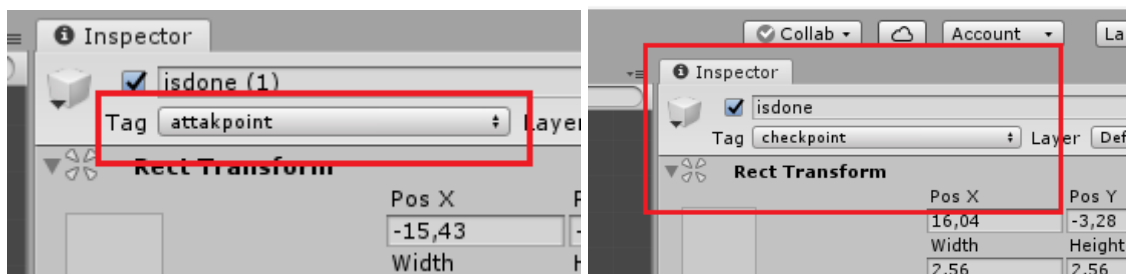


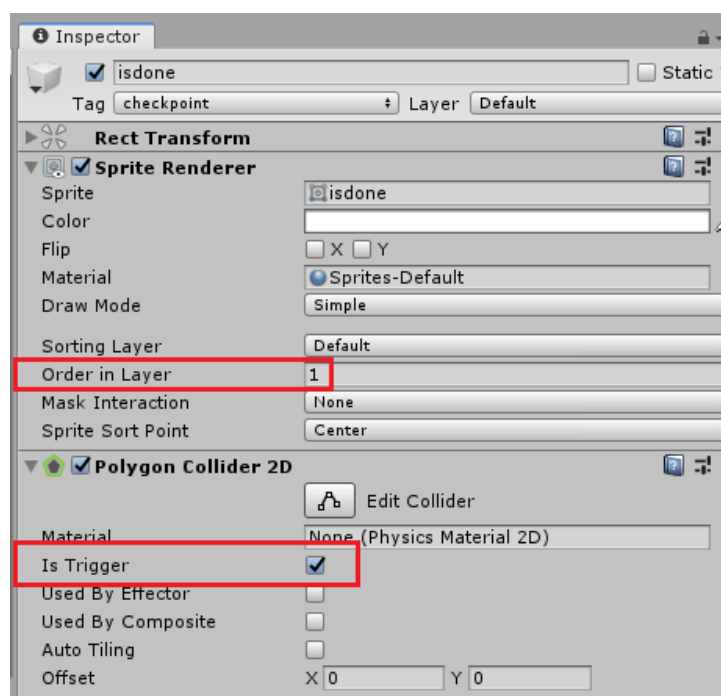
Crie um novo projeto 2D, insira uma imagem de fundo, um asset para o player, uma abelha para ser o enemy e duas bandeiras para marcar duas triggers.



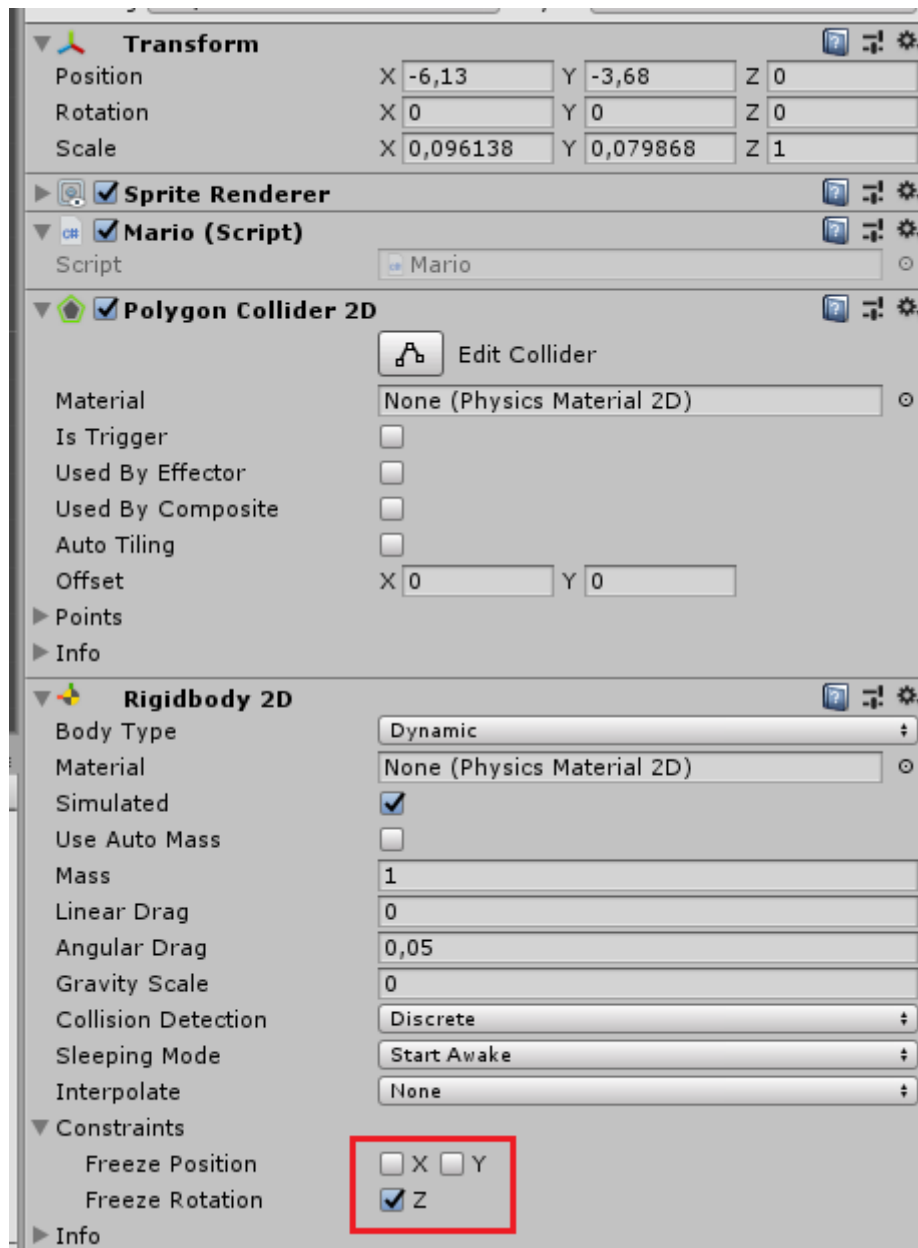
Coloque duas tags, uma em cada bandeira.



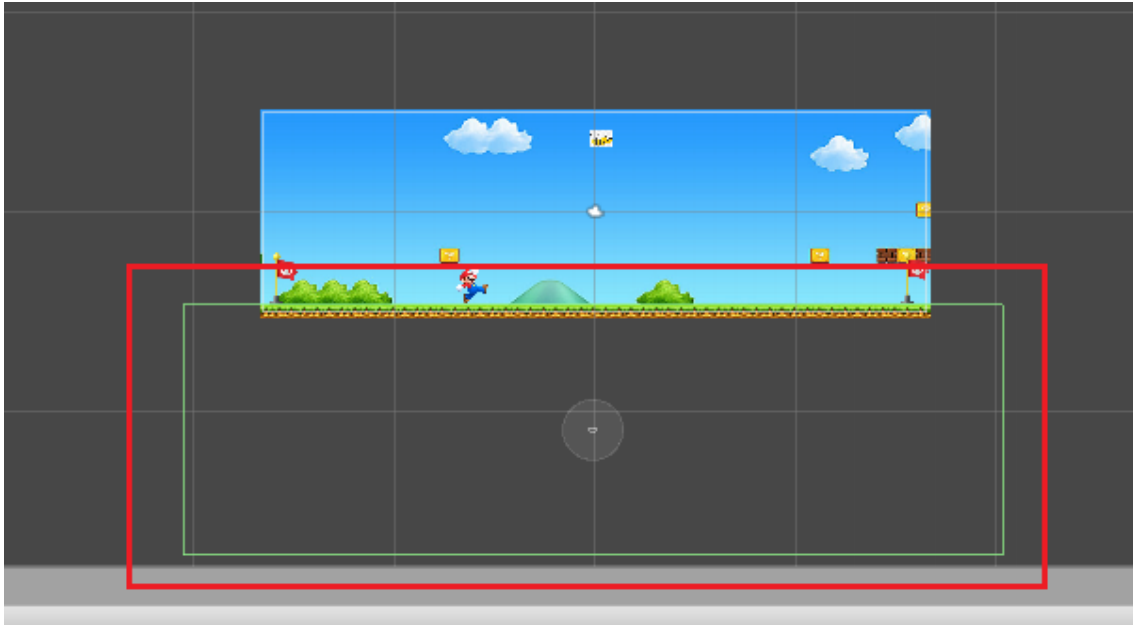
Adicione um Polygon Collider 2D para cada uma das bandeirinhas e ajuste a Order in Layer para 1 e marque a opção Is trigger



Insira um Rigid Body 2D e um Polygon Collider 2D no player, coloque a Order In Layer como 1



Crie um chão transparente, insira um Rigid Body 2D e um Box Collider 2D, congele os três eixos do campo Constraints do Rigid Body 2d e coloque na Order in Layer 1



Agora crie o script para a movimentação do player

```
public class Mario : MonoBehaviour
{
    public static bool attak = false;
    public static float posMarioX, posMarioY;

    void Update()
    {
        posMarioX = transform.position.x;
        posMarioY = transform.position.y;

        if(Input.GetAxis("Horizontal") > 0)
        {
            transform.Translate(Vector3.right * 0.05f);
        }

        if (Input.GetAxis("Horizontal") < 0)
        {
            transform.Translate(Vector3.left * 0.05f);
        }
    }
}
```

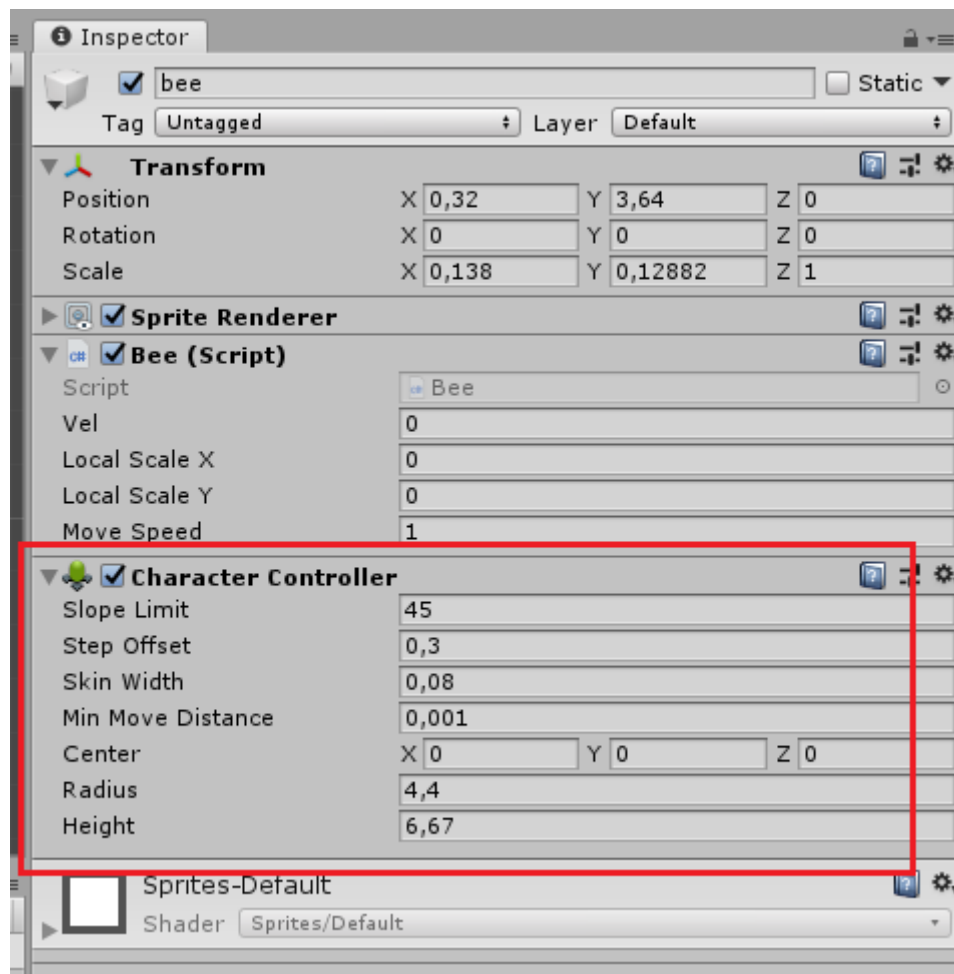
Não esqueça de executar e adicionar o script ao player

Agora vamos criar o código que vai controlar as colisões entre o player e as bandeirinhas

```
private void OnTriggerEnter2D(Collider2D collision)
{
    if(collision.gameObject.tag == "attakpoint")
    {
        attak = true;
    }

    if (collision.gameObject.tag == "checkpoint")
    {
        attak = false;
    }
}
```

Agora insira o inimigo na cena do jogo com os seguintes elementos



Crie o script para o inimigo

```
public class Bee : MonoBehaviour
{
    public float vel, localScaleX, localScaleY;
    CharacterController _controller;
    Transform target, me;
    GameObject Player;

    public float _moveSpeed = 1f;

    void Start()
    {
        vel = 0.03f;
        localScaleX = transform.localScale.x;
        localScaleY = transform.localScale.y;

        Player = GameObject.FindWithTag("Player");
        me = this.transform;
        target = Player.transform;
        _controller = GetComponent<CharacterController>();
    }
}
```

Continue a escrita do script do inimigo

```
// Update is called once per frame
void Update()
{
    if (!Mario.attak)
    {
        transform.Translate(Vector3.right * vel);

        if(transform.position.y < 3f)
        {
            transform.Translate(Vector2.up * 0.02f);
            if(vel < 0)
            {
                transform.localScale = new Vector2(-localScaleX, localScaleY);
            }
            else
            {
                transform.localScale = new Vector2(localScaleX, localScaleY);
            }
        }

        if (transform.position.x > 17.4f)
        {
            vel = -0.03f;
            transform.localScale = new Vector2(-localScaleX, localScaleY);
        }

        if (transform.position.x < -17.4f)
        {
            vel = 0.03f;
            transform.localScale = new Vector2(localScaleX, localScaleY);
        }
    }
    else
```

Ainda no mesmo script

```
}  
else  
{  
    Vector3 direction = target.position - transform.position;  
    direction = direction.normalized;  
    Vector3 velocity = direction * _moveSpeed;  
    _controller.Move(velocity * Time.deltaTime);  
  
    if (direction.x > 0f)  
    {  
        transform.localScale = new Vector2(localScaleX, localScaleY);  
    }  
    else  
    {  
        transform.localScale = new Vector2(-localScaleX, localScaleY);  
    }  
}  
}
```

Coloque a tag Player no seu player e teste seu jogo, veja se quando você toca na bandeira com a tag attackpoint o inimigo começa a perseguir o player.

Assim que o player toca na bandeira com a tag checkpoint o inimigo deve sair de perto e para a perseguição.