

PSI3502 - Tarefa 5

(Projeto: Lidando com fobias)

1) Projeto Unity

- Cenas:

Mesmo buscando em diversas fontes, o único modelo que se adequou ao projeto foi o da casa abandonada já apresentada. Por esse motivo, optamos por melhorar a iluminação e deixa-la menos sombria.

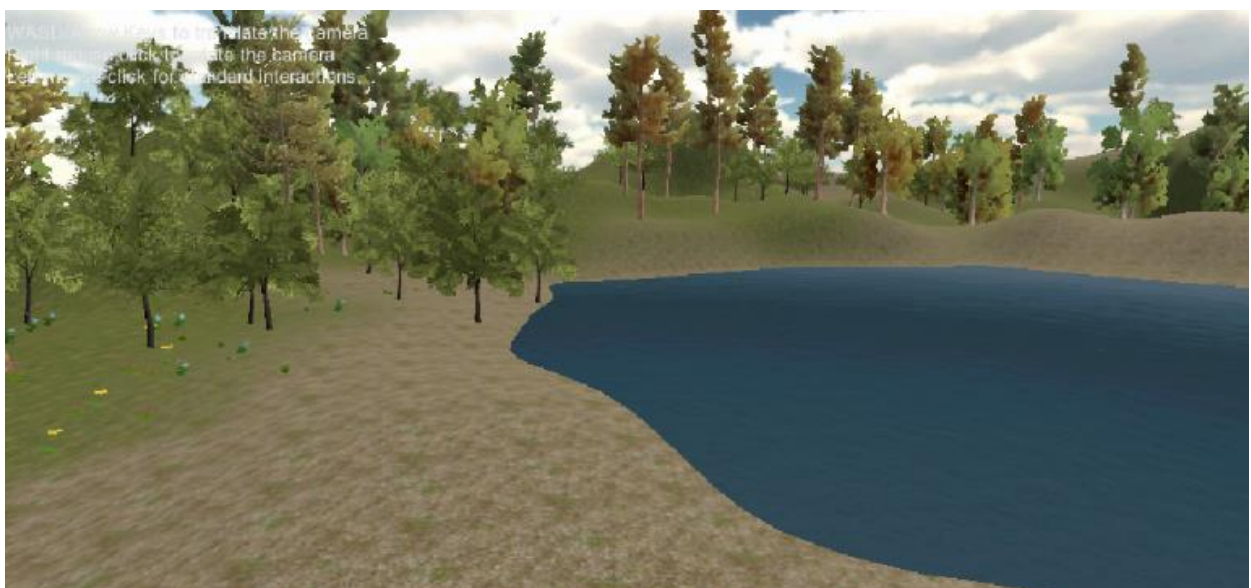


De fato, é bem nítido o contraste entre os ambientes, mas nesse caso, pode ser algo interessante para usuário que é capaz de desvincular rapidamente da sua cabeça o ambiente em que estava exposto as baratas.



Como solicitado, o lugar seguro sofreu melhorias:

- ✓ Foram adicionadas árvores mais realistas, grama, flores...
- ✓ Foi adicionado um Skybox com o céu
- ✓ Foram adicionados água e vento
- ✓ O terreno passou por melhorias



- Prefabs:

Devido a iluminação, os prefabs se tornaram mais visíveis e como pedido, a barata sofreu melhorias, já que não foram encontrados outros modelos gratuitos.



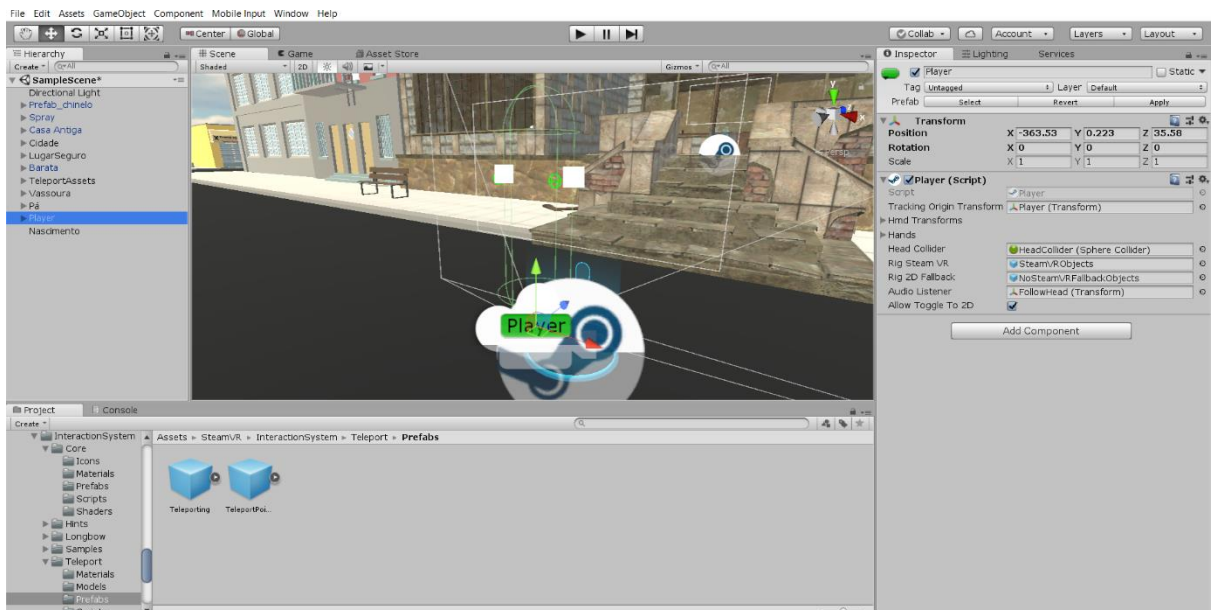
Segue o modelo da barata mais de perto:



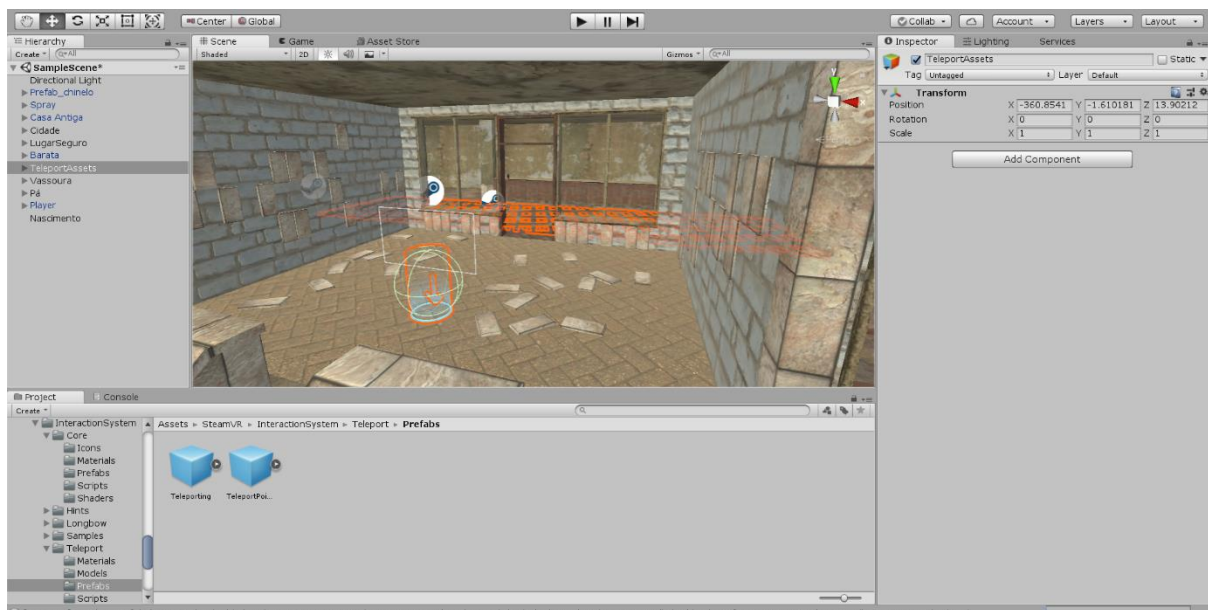
- Métodos de locomoção:

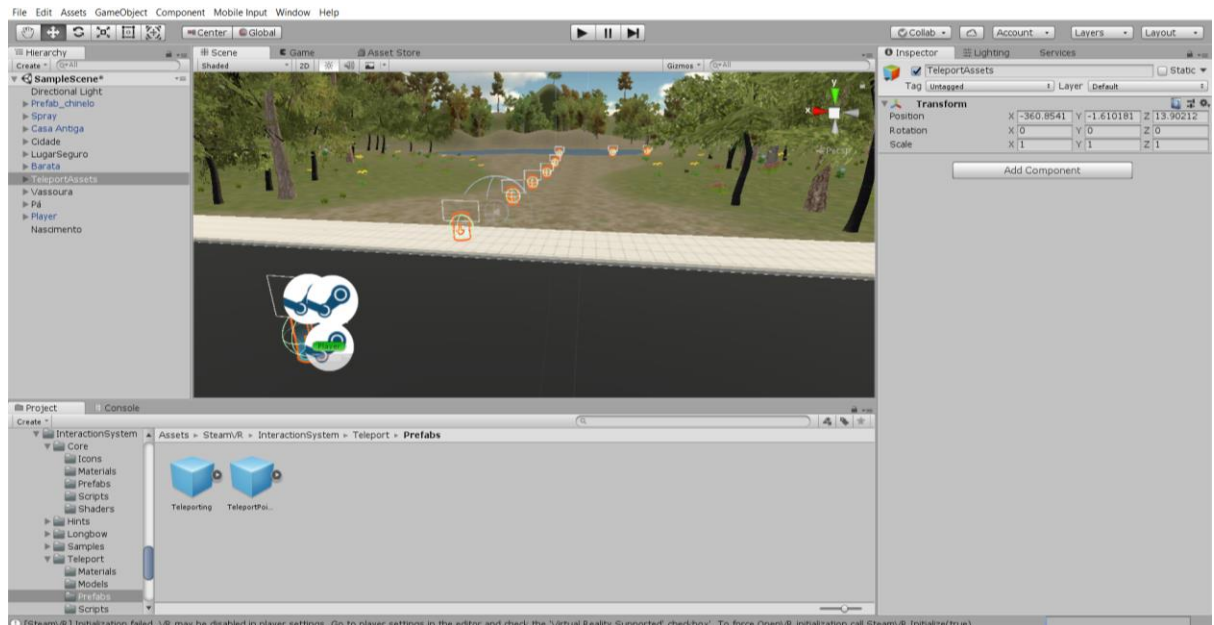
Os métodos de locomoção previstos são:

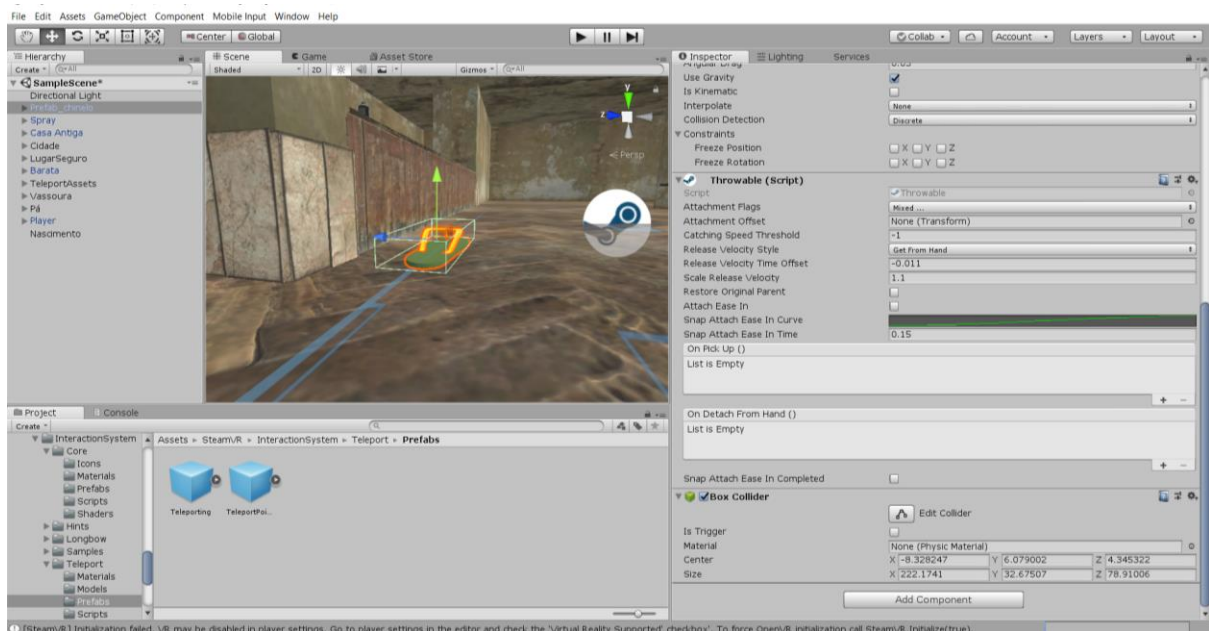
✓ **Andar:** o usuário precisa andar dentro da casa, para analisá-la e interagir com a barata. É utilizado um sensor de rastreamento dos óculos. Já é implementado automaticamente ao usar o Player do SteamVR:



✓ **Teleportar:** é usado ao apertar um botão no joystick para locomover-se para o local de fuga, através da implementação da função teleport, do SteamVR. Também é usado para locomoção na cidade, fora da casa. Seguem os pontos e áreas de teleport:







Códigos

Foram utilizados alguns códigos prontos já da própria SteamVR e alguns prefabs já possuíam códigos próprios, como a movimentação da água e das árvores/flores/grama. Por esse motivo, citarei aqui apenas os códigos criados para o projeto específico.

❖ **Move** (Descreve o movimento aleatório das baratas)

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class move : MonoBehaviour
{
    public Rigidbody rb;
    public float ForceX, TorqueX;
    public float ForceY, TorqueY;
    public float ForceZ, TorqueZ;

    void Start()
    {
        rb = GetComponent<Rigidbody>();
    }

    void FixedUpdate()
    {
        rb.AddForce(ForceX*Random.Range(-10.6f, 10.6f), ForceY*Random.Range(-10.6f, 10.6f), ForceZ*Random.Range(-10.6f, 10.6f), ForceMode.Impulse);
        rb.AddRelativeTorque(TorqueX * Random.Range(-10.6f, 10.6f), TorqueY * Random.Range(-10.6f, 10.6f), TorqueZ * Random.Range(-10.6f, 10.6f), ForceMode.Impulse);
    }
}
```




❖ **Spawner** (Faz as baratas surgirem de um ponto)

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Spawner : MonoBehaviour {
    public GameObject barata_prefab;
    public float time = 5f;
    public int quantidade;
    // Use this for initialization
    void Start () {
        StartCoroutine("Rotina");
    }

    IEnumerator Rotina ()
    {
        for (int i = 0; i < quantidade; i++)
        {
            yield return new WaitForSeconds(time);
            Instantiate(barata_prefab,transform.position,barata_prefab.transform.rotation);
        }
    }
}
```




❖ **Morte** (Baratas atingidas são substituídas por um prefab que cai)

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Morte : MonoBehaviour {
    public GameObject barata_morte;

    void OnTriggerEnter(Collider other)
    {
        if(other.tag == "Arma")
        {
            Instantiate(barata_morte, transform.position,
barata_morte.transform.rotation);
            Destroy(gameObject, 0);
        }
    }
}
```



❖ **Some** (Faz as baratas sumirem depois de mortas)

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Some : MonoBehaviour {
    public float tempo = 5f;
    // Use this for initialization
    void Start () {
        Destroy(gameObject, tempo);
    }
}
```