

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO

PSI3502 – Realidade Virtual

Tarefa 4

GRUPO: APLICAÇÕES PARA O ENSINO

ALEXANDRE HOPPE INOUE	8988942
BRUNO HARLLEN	9871051
LUCAS SALLABERRY	8038514
LUCAS SPONCHIADO	9373924
RAUL DA SILVA SOUZA	8630540
VITOR AUGUSTO MARTIN	8993100

São Paulo, 29/10/2018

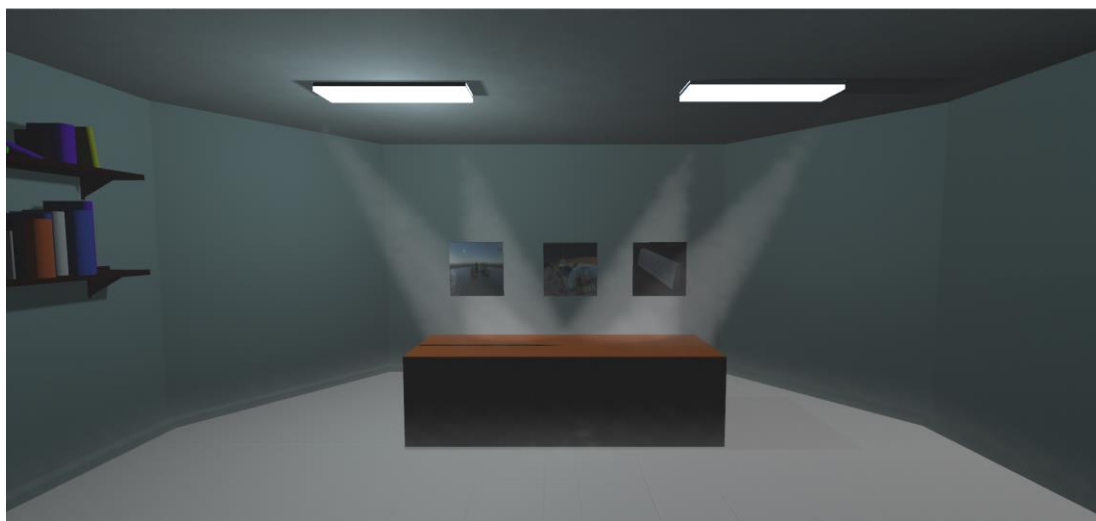
Cenas:

1) Sala Inicial

Esboço 3D:

O aluno é colocado em uma sala simples com uma mesa e alguns objetos de ambiente. O aluno é introduzido a uma rápida explicação de como a cena funciona (provavelmente por uma narrativa) e se dirige à mesa. Para facilitar a orientação do aluno na sala, luzes volumétricas suaves serão lançadas sobre a mesa, criando curiosidade sobre o objeto.

Sobre a mesa estão deitados os ícones de cada um dos desafios. Ao se aproximar, esses ícones se levantam da mesa, flutuando à frente do aluno e girando para que a face da frente esteja sempre voltada para o aluno. Ao colocar a mão sobre um dos desafios, ele irá aumentar e mostrará um vídeo “preview” do desafio em questão. Caso o aluno decida que quer fazer esse desafio, ele deve segurar o ícone que representa o desafio e o colocar sobre a mesa, em uma espécie de socket, e apertar um botão “Iniciar” que estará ao lado.



Prefabs:

A cena conta com um prefab dos ícones. Esses prefabs contém a lógica de animação dos ícones, sendo apenas necessário mudar o vídeo contido em cada um.



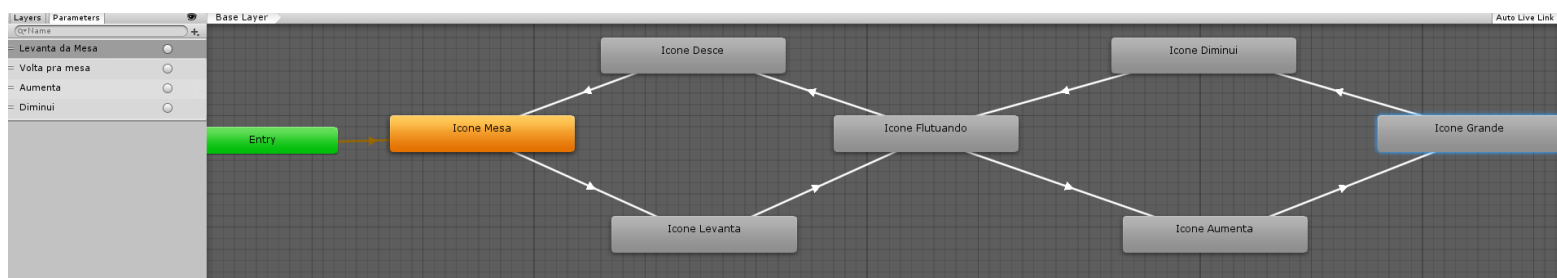
Será também utilizado prefab de luzes volumétricas do plugin Aura - Volumetric Lighting para criar as luzes volumétricas e alguns prefabs prontos do asset Morgue Room PBR para modelar a sala.

Métodos de locomoção:

A Locomoção nesta cena será feita dentro da área 4x4m prevista, visando ser bastante natural.

Interações:

As interações feitas com os ícones seguirão a lógica de animações:



como comentei antes acho que ícones 3D exploram melhor os recursos de VR. O play de video não é ruim, mas miniaturas idealizadas de cada desafio é muito mais lúdico e aproveita melhor a cena-menu

Onde as transições são controladas, até o momento, pelo script:

```

1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.Video;
5
6 public class AtivarAnimacoesIcones : MonoBehaviour {
7     private Animator[] animators;
8     private GameObject player;
9     private OlharParaJogador[] ativarOlhar;
10
11     void Start(){
12
13         //Agrega todos os Icones na cena
14         GameObject[] Icones = GameObject.FindGameObjectsWithTag ("Icône");
15         animators = new Animator[Icones.Length];
16
17         //OlharParaJogador é o script que faz o Icône olhar sempre para o jogador
18         ativarOlhar = new OlharParaJogador[Icones.Length];
19
20         for(int a = 0; a < Icones.Length; a++){
21             animators [a] = Icones [a].GetComponent<Animator> ();
22             ativarOlhar [a] = Icones [a].transform.GetChild(0).gameObject.GetComponent<OlharParaJogador> ();
23         }
24     }
25
26     //jogador entrou na área de ativação
27     void OnTriggerEnter(Collider other)
28     {
29         if(other.gameObject.tag == "Player")
30         {
31             foreach (Animator anim in animators) {
32                 anim.SetTrigger ("Levanta da Mesa");
33             }
34             foreach (OlharParaJogador component in ativarOlhar) {
35                 component.ativar = true;
36             }
37         }
38     }
39
40     //jogador saiu da área de ativação
41     void OnTriggerExit(Collider other)
42     {
43         if(other.gameObject.tag == "Player")
44         {
45             foreach (Animator anim in animators) {
46                 anim.SetTrigger ("Volta pra mesa");
47             }
48             foreach (OlharParaJogador component in ativarOlhar) {
49                 component.ativar = false;
50             }
51         }
52     }

```

Continuação...(chamados pelo evento On Hand Hover Begin/End)

```
53
54 //caso jogador colocar a mão sobre ícone
55 public void AumentaIcone(GameObject objeto){
56     objeto.GetComponent<Animator>().SetTrigger ("Aumenta");
57     objeto.transform.GetChild (0).GetChild (0).GetComponent<VideoPlayer> ().Play ();
58 }
59
60 //caso jogador tirar a mão do ícone
61 public void DiminuiIcone(GameObject objeto){
62     objeto.GetComponent<Animator>().SetTrigger ("Diminui");
63     objeto.transform.GetChild (0).GetChild (0).GetComponent<VideoPlayer> ().Stop ();
64 }
65
66 }
67
```

E a rotação dos Ícones por:

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class OlharParaJogador : MonoBehaviour {
6
7     private GameObject player;
8
9     [HideInInspector]
10    public bool ativar = false;
11
12    void Start(){
13        player = GameObject.FindGameObjectWithTag("Player");
14    }
15
16    void Update(){
17        if (player != null && ativar) {
18            transform.LookAt (player.transform);
19        }
20    }
21
22 }
23
```

Além disso, ainda será necessário criar a lógica de posicionar os ícones no socket para se escolher o desafio e a mudança de cena para esse desafio.

2) Desafio Tiro de canhão

(Para evitar ter que sair do ambiente virtual para realizar contas e resolver o desafio, resolvemos mudar o desafio do foguete para o desafio proposto inicialmente dos

canhões no barco. Só que em vez do aluno ter que calcular as inclinações do canhão, serão abordados conceitos como:

- Maior distância horizontal atingida a 45 graus.
- Um navio que pode ser atingido com um canhão a 30 graus pode ser atingido a 60 graus.

A proposta do desafio mudou um pouco. O objetivo será derrubar os 3 navios inimigos sem acertar as gaivotas presentes no cenário. Um estará bem longe e precisará angular o canhão em 45 graus. O segundo navio só poderá ser atingido atirando a 30 graus (atirando a 60 acertaria um conjunto de gaivotas) e o outro atirando a 60 graus (atirando a 30 acertaria outro conjunto de gaivotas)

Esboço 3D:

A cena inicia com o aluno sendo inserido no seu navio. Ele vai até o canhão destacado para atirar nos navios inimigos. Ao interagir com o canhão ele receberá feedback de quão inclinado o canhão está e poderá inclinar mais ou menos com os seus controles da Oculus e finalmente atirar para tentar acertar os navios.



como entao o aluno aprende os conceitos descritos? tentativa e erro ? Não é uma má ideia, mas nesse caso seria muito bom ele ter mais feedback de cada tentativa para associar ações a efeitos. Tem jogos que o jogador pode dar um replay, mudar de posição, etc. Talvez voces pudessem fazer isso e mostrar alguns dados de maneira visual durante o replay (plano da trajetória, trajetória, velocidade instantânea, etc)

Prefabs:

Por enquanto, a cena possui os seguintes prefabs:

- Do canhão e bola de canhão

- Navios inimigos mais leves e pouco detalhados
- Navio em que o player está embarcado
- Elementos do ambiente como mar, e nuvens.

Ainda vão ser necessários prefabs para a fumaça do tiro de canhão e das gaivotas.

Métodos de locomoção:

A locomoção será com a utilização do Oculus Rift usando a área 4x4, uma vez que o espaço para se explorar no navio é razoavelmente pequeno.

Interações:

a tarefa foi prorrogada, portanto entreguem um pseudo script da interação

O usuário pode interagir com os canhões do navio e serão necessários scripts para inclinar o canhão, atirar o projétil e identificar quando os alvos são atingidos.

3) Construção de Barragem

Esboço 3D:

Prefabs:

Métodos de locomoção:

Interações:

a tarefa foi prorrogada, portanto terminem a descrição dos outros desafios e descrevam pseudo scripts das interações