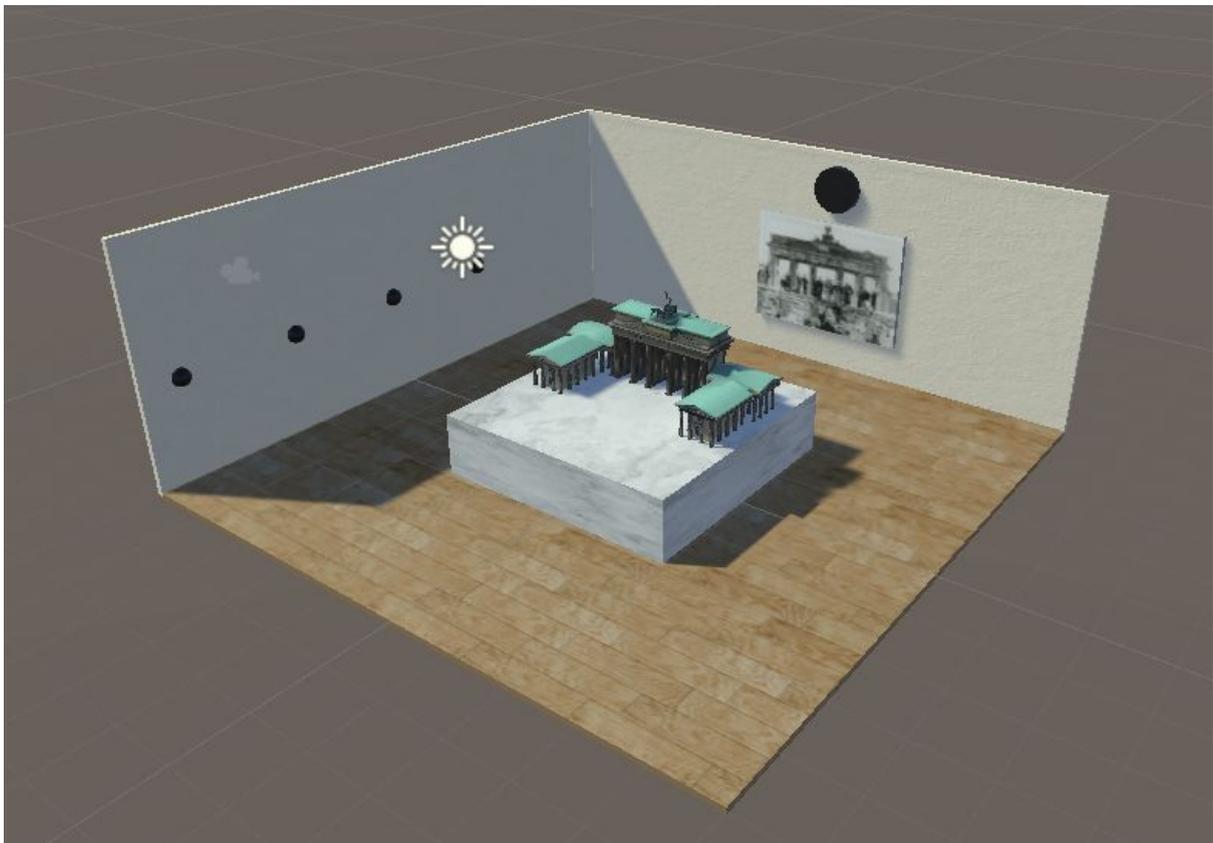


## Relatório 4

- Projeto Unity:
  - Cenas: esboço 3D concluído -> descrever princípios estético-visuais utilizados e mostrar screenshots



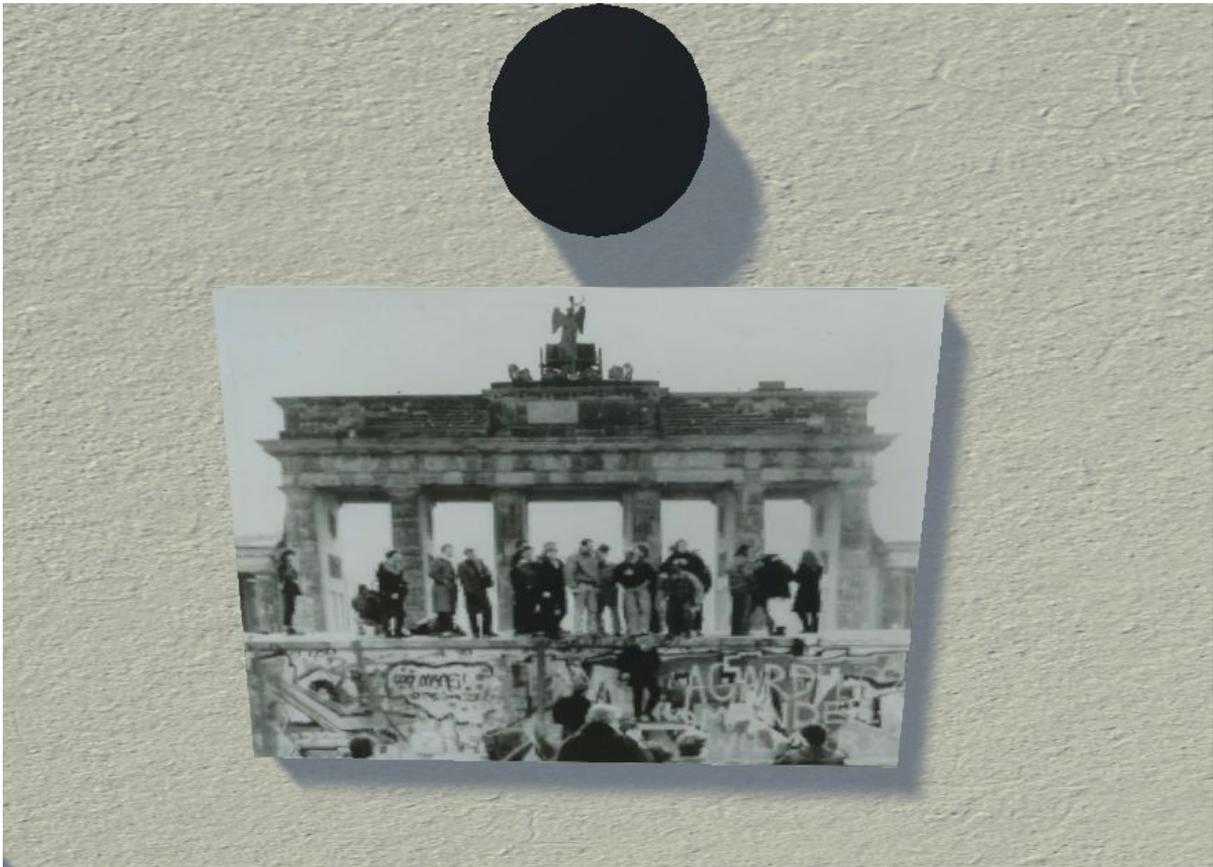
- Prefabs: substituir todos os 'placeholders' (objetos de teste) por modelos (podem ser preliminares ainda) -> mostrar screenshots

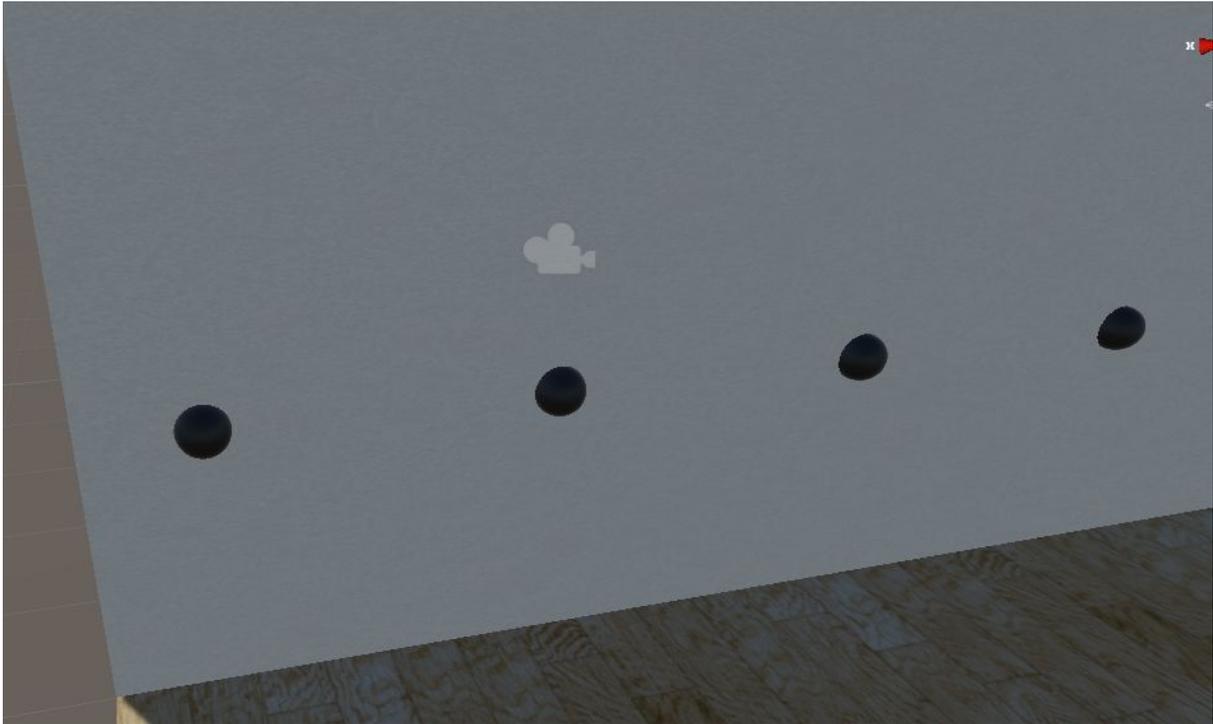
Pessoal, o trabalho precisa ganhar ritmo. É muita gente no grupo e a coisa não está andando. Tem um grupo (o de Fobias-baratas) com apenas 3 pessoas, sem veteranos. Se o ritmo continuar assim vou remanejar 2 veteranos para esse grupo. Caso algum queira mudar, avise. Vocês deveriam, no mínimo ter levantado uma infinidade de fatos, imagens e, de preferência, modelos.

Na apresentação da última aula, o prof. Marcelo demonstrou conhecimento e interesse pelo tema. Pelo que vi existem muitos fatos relacionados ao muro que podem ser ilustrados por modelos e interações. Sugiro fortemente e o prof. espera que o trabalho de vocês - ainda mais considerando a quantidade de membros - incorpore na forma de interações na linha do tempo muitos desses fatos relacionados ao muro, incluindo alguns numa distância absoluta maior. Como sugeri um dos membros do grupo, vocês podem adotar uma licença poética e criar uma forma visual interessante de localizar esses pontos.

Como a principal feature do sistema de vocês é uma interação passiva ou básica (andar e pegar objetos) então esperamos bastante conteúdo e uma diversidade de objetos aparecendo ao longo desta linha do tempo. O aspecto mais marcante é justamente o mecanismo da linha do tempo, portanto esperamos bastante dele. Não ficará legal apenas mover o ponteiro e alguns objetos mudam. É interessante criar uma forma de mostrar ao usuário o que mudou, tanto na cena como no relógio. Imagino girar o ponteiro e ver algo piscando quando uma mudança ocorre e na cena, os novos objetos aparecem com uma representação diferente, até o usuário chegar mais perto, por exemplo. Enfim pessoal, a ideia do trabalho é boa, merece um tratamento melhor.

A entrega da tarefa 4 foi prorrogada até a aula que vem, portanto entreguem novamente e observem tudo o que foi pedido na tarefa. Vocês tem mais tempo para seguir as sugestões acima.





- Protótipos dos métodos de locomoção previstos: se utilizar apenas Steam VR, definir detalhes (pontos de teleport no mapa, etc) -> iniciar testes em sala de aula e entregar screenshots

Pretende-se utilizar para locomoção o teleport do SteamVR de forma a se posicionar em locais que possa interagir e visualizar cada um dos elementos do cenário, aliado com o movimento em área proporcionado pelo Vive. Quando na perspectiva do portão pretende-se uma exploração ainda mais livre.

- Planejamento:
  - Para cada interação prevista, listar prefabs e scripts do SteamVR a serem utilizados ou descrever pseudocódigo dos scripts.

Mudar Cenário: Quando a transição é ativada o cenário é alterado para a época desejada, o método aplicada será simplesmente ativar e desativar os modelos de cada época. Alguns efeitos serão adicionados.

```
OnAction2(){  
    model1.SetActive(false);  
    model2.SetActive(true);  
    model3.SetActive(false);  
    model4.SetActive(false);
```

```
}
```

Exploração do modelo do Brandenburg: Quando a transição é ativada o usuário será transportado de forma a ter impressão de ter diminuído de tamanho para explorar o modelo do portão de Brandenburg como se fosse tamanho real. Será feito com uma transição de cena, para uma cena com todos os modelos de tamanho maior e o player na plataforma do portão.

Interação com os artefatos: Pretende-se poder pegar os objetos de forma a inspecioná-los da forma que o usuário quiser. Será utilizado como base o seguinte script.

```
void Update()
```

```
{
```

```
    if (controller == null)
```

```
    {
```

```
        Debug.Log("Controller not initialized");
```

```
        return;
```

```
    }
```

```
    var device = SteamVR_Controller.Input((int)trackedObj.index);
```

```
    if (controller.GetPressDown(triggerButton))
```

```
    {
```

```
        PickupObj();
```

```
    }
```

```
    if (controller.GetPressUp(triggerButton))
```

```
    {
```

```
        DropObj();
```

```
    }
```

```
}
```

```
void OnTriggerStay(Collider other)
```

```
{
```

```
    if (other.CompareTag("Pickupable"))
```

```
    {
```

```
        obj = other.gameObject;
```

```
    }
```

```
}
```

parece que esse script faz exatamente o que um interactable do SteamVR faz. Basta usar o que já está pronto e otimizado.

```
void OnTriggerExit(Collider other)
{
    obj = null;
}
void PickUpObj()
{
    if (obj != null)
    {
        fJoint.connectedBody = obj.GetComponent<Rigidbody>();
    }
    else
    {
        fJoint.connectedBody = null;
    }
}
void DropObj()
{
    if (fJoint.connectedBody != null)
    {
        fJoint.connectedBody = null;
    }
}
```