

VISEDU: INTERFACE DE USUÁRIO TANGÍVEL UTILIZANDO REALIDADE AUMENTADA

Aluno: Antônio Marco da Silva

Orientador: Dalton Solano dos
Reis, M. Sc.

Roteiro

- Introdução
- Objetivos
- Fundamentação Teórica
- Trabalhos Correlatos
- Requisitos
- Especificação
- Implementação
- Operacionalidade da implementação
- Resultados e Discussões
- Conclusões e Extensão

Introdução

- Exploração da Interface de usuário tangível
- Estudo de criação de uma aplicação com uma interface tangível utilizando Realidade aumentada

Objetivos

- Desenvolver uma interface de usuário tangível utilizando realidade aumentada
- Desenvolver funcionalidades para objetos virtuais interagir entre si utilizando marcadores

Fundamentação Teórica

- Interface de usuário tangível
- Variações da interface de usuário tangível:
 - *TableTop* Tangível
 - *Ambient Display*
 - Interface de usuário encorpada
 - Realidade aumentada Tangível



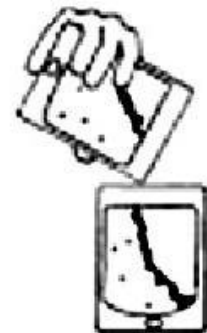
A) Realidade Aumentada Tangível



B) TableTop Tangível



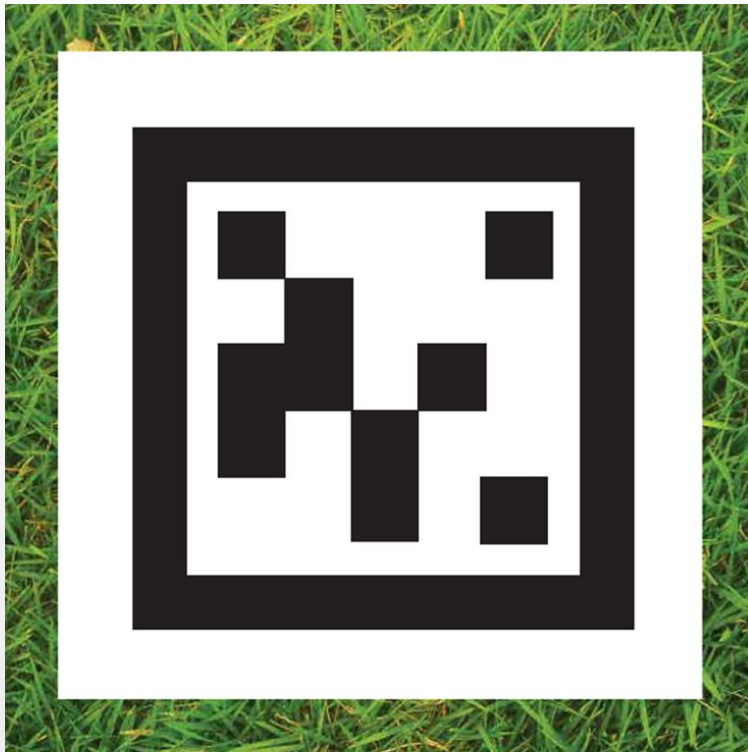
C) Ambient Display



D) Interface de Usuário Encorpada

Fundamentação Teórica

- Realidade Aumentada
 - Marcadores



Fundamentação Teórica

- Unity
 - Plataforma para desenvolvimento de jogos e aplicativos
 - Multiplataforma
 - Fácil aprendizagem
 - Disponibilidade dos recursos para o desenvolvimento do trabalho

Fundamentação Teórica

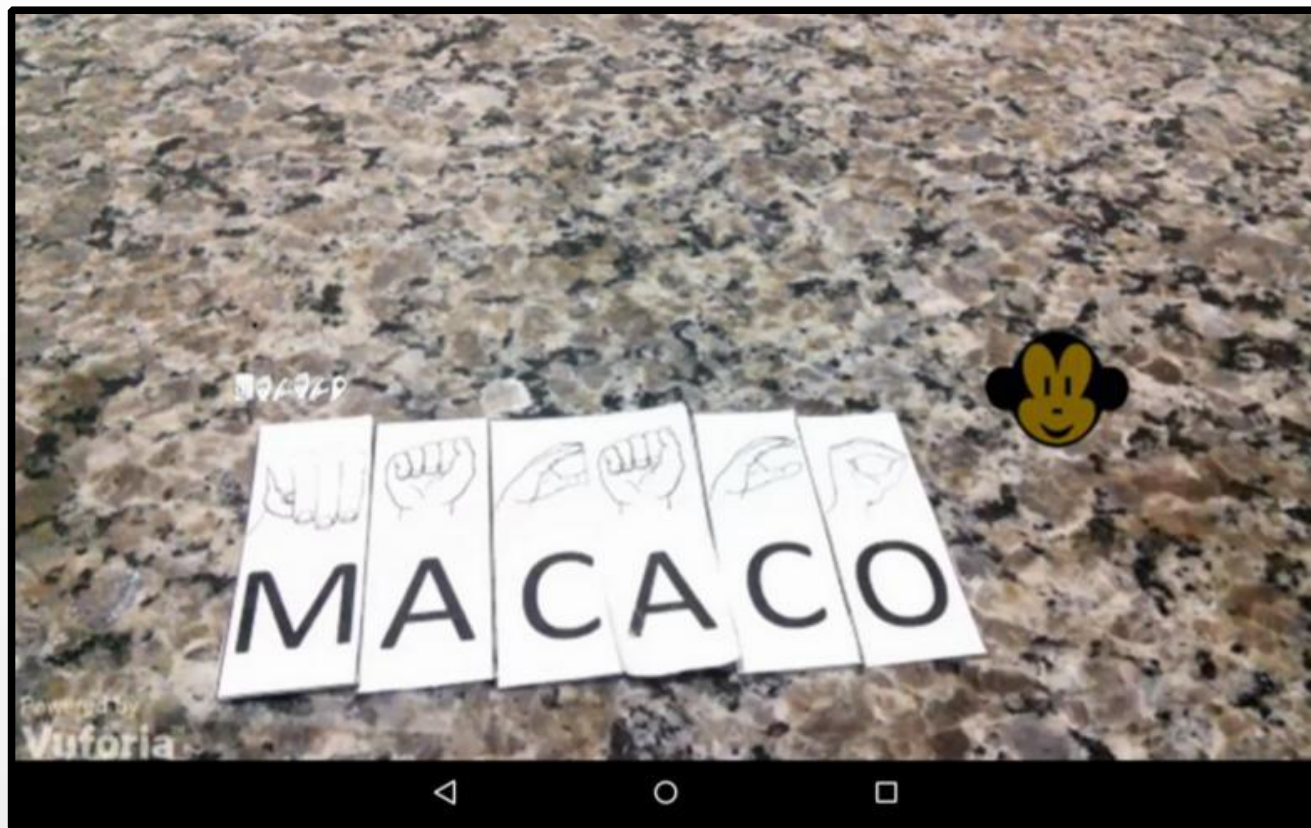
- Vuforia
 - Biblioteca para desenvolvimento de aplicações em Realidade aumentada
 - Possui integração com Unity

Fundamentação Teórica

- Vuforia
- Marcadores
 - Imagens
 - Cilindros
 - Cubos
 - Objetos Tridimensionais
- Extended Tracking

Trabalhos Correlatos

Biblioteca para desenvolvimento de jogos multijogadores
de
Realidade Aumentada para Android (2015)



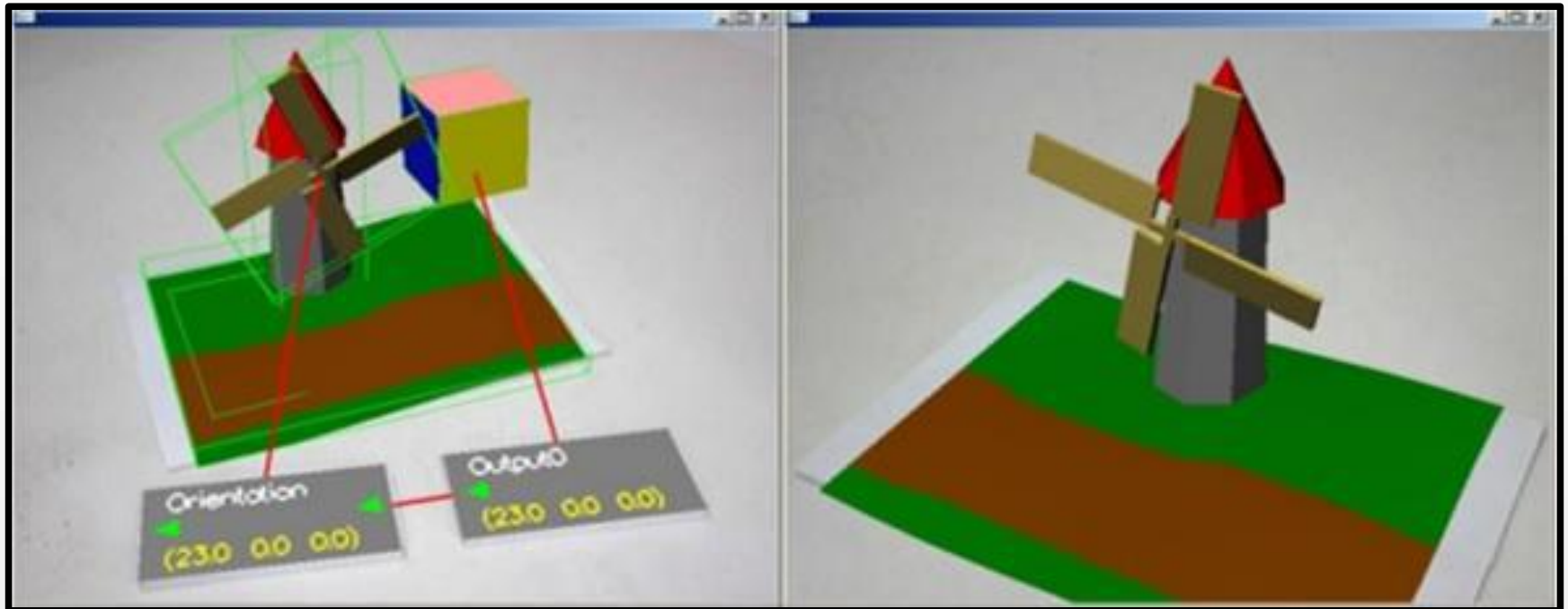
Trabalhos Correlatos

Integrando Interface Tangível com Técnicas de Realidade Aumentada para Ampliar a Experiência Interativa do Usuário (2012)



Trabalhos Correlatos

Immersive Authoring of Tangible Augmented Reality Applications (2004)



Trabalhos Correlatos

Características	Aplicativo desenvolvido	Lira(2015)	Rodrigues, Sato e Botega (2012)	Lee et al. (2004)
Possui interface de usuário tangível	Sim		Sim	Sim
Criação de RA baseada em marcadores	Sim	Sim	Sim	Sim
Permite geração de objetos 3D virtuais	Sim	Sim	Sim	Sim
Detecção de múltiplos marcadores	Sim	Sim	Sim	Sim
Utiliza o SDK Vuforia	Sim	Sim		
Interação ente objetos virtuais	Sim			Sim

Requisitos

- Requisitos funcionais do aplicativo
 - marcador para gerar a biblioteca de objetos virtuais
 - cubo como marcador para selecionar e movimentar objetos virtuais
 - marcador para inserir objetos virtuais no ambiente virtual
 - botões virtuais para selecionar o objeto virtual

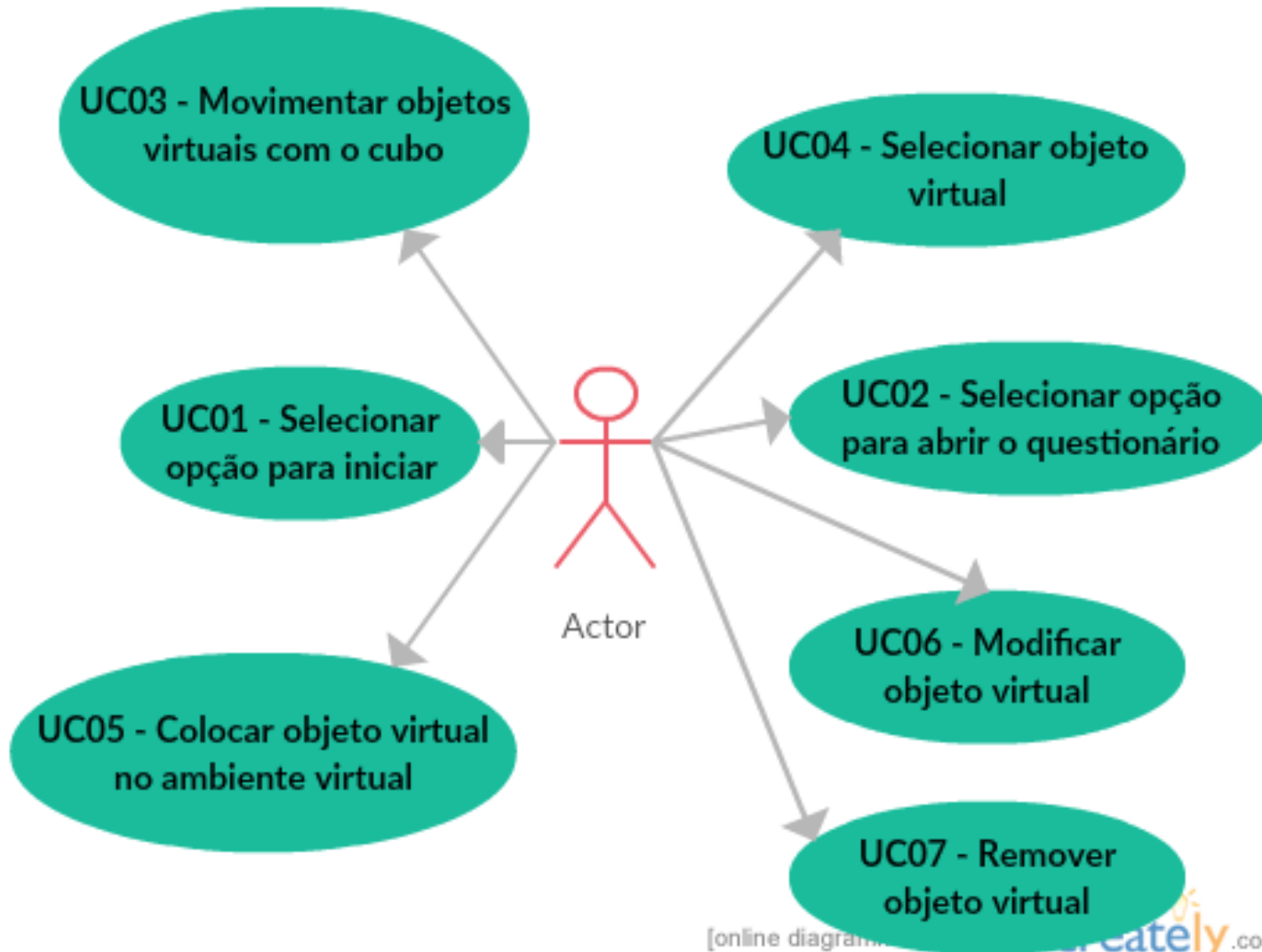
Requisitos

- Requisitos funcionais do aplicativo
 - marcador para selecionar um atributo do objeto virtual
 - marcador para modificar o valor de um atributo do objeto virtual

Requisitos

- Requisitos não funcionais do aplicativo
 - ser disponibilizado para a plataforma Android
 - utilizar a SDK Vuforia para a implementação dos recursos de RA
 - utilizar a linguagem C# para definir os scripts do aplicativo
 - utilizar o motor de jogos Unity para a implementação do espaço 3D

Especificação



[online diagram created with Creately.com]

Especificação

```
<<MonoBehavior>>  
CuboManager  
- void : getObjectByLine()  
- void : Start()  
- void : Update()
```

```
<<MonoBehavior>>  
TrashManager  
- void : Start()  
- void : Update()  
- void : OnTriggerEnter(Collider)  
- void : OnTriggerStay(Collider)  
- void : OnTriggerExit(Collider)
```

```
<<MonoBehavior>>  
SceneMarkerManager  
- void : Start()  
- void : Update()  
- IEnumerator : waitseconds(int)  
- void : OnTriggerEnter(Collider)  
- void : OnTriggerStay(Collider)  
- void : OnTriggerExit(Collider)
```

```
<<Static>>  
SceneControl  
+ Vector3 : Bezier2(Vector3, Vector3, Vector3, float)  
+ GameObject : getObjectFactory()  
+ GameObject : getFactoryPlate()  
+ bool : isChildFactoryPlate(GameObject)  
+ bool : isChildObjectFactory(GameObject)  
+ Vector3 : PositionRelativeBottom(Vector3, Vector3)  
+ Vector3 : ScaleToFit(GameObject, Vector3)  
+ void : ButtonCoolDown(GameObject[], float, float)  
+ bool : isZeroVector(Vector3)
```

```
<<MonoBehavior>>  
TitleScreenManager  
+ void : openURL()  
+ void : loadScene(int scene)  
+ void switchActive(GameObject panel)
```

```
<<MonoBehavior>>  
MainScreenManager  
+ void : Start()  
+ void : Update()
```

```
<<MonoBehaviour>>  
<<IVirtualButtonEventHandler>>  
ObjectFactory  
- void : Start()  
- void : setObject(int)  
- void : Update()  
+ void : OnButtonPressed (VirtualButtonAbstractBehaviour)  
+ void : OnButtonReleased(VirtualButtonAbstractBehaviour)
```

```
<<MonoBehaviour>>  
Modifiers  
+ void : setValues()  
+ void : resetModPosition()  
- void : updateMotor()  
- void : updateObject()  
...
```

Especificação

```
<<MonoBehaviour>>  
<<ITrackableEventHandler>>  
<<IVirtualButtonEventHandler>>  
SettingManager
```

```
+ void : UpdateDescription()  
+ void : activateCanvas()  
+ void : OnButtonPressed(VirtualButtonAbstractBehaviour)  
+ void : OnButtonReleased(VirtualButtonAbstractBehaviour)  
+ void : OnTrackableStateChanged(TrackableBehaviour.Status, TrackableBehaviour.Status)  
+ bool : getActiveCanvas()  
+ GameObject : getSelectedObject()  
+ int : getActiveMod()  
+ float : getActualValue()  
+ float : getModValue()  
+ void : setActualValue(float)  
+ void : setMotorValue(float)  
+ float : getMotorValue()  
- void : getObjectByLine()  
- void : setLinePosition()  
- float : cooldownTime  
- void : updateTitle()  
+ void : setMotorValue(float value)  
+ float : getMotorValue()  
- void : getObjectByLine()  
- void : setLinePosition()  
- float : cooldownTime  
- void : updateTitle()
```

```
<<MonoBehaviour>>  
<<ITrackableEventHandler>>  
<<IVirtualButtonEventHandler>>  
ModifierManager
```

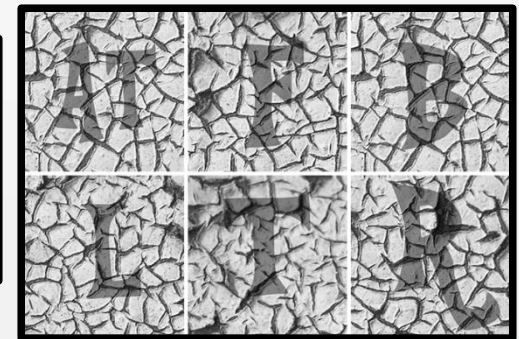
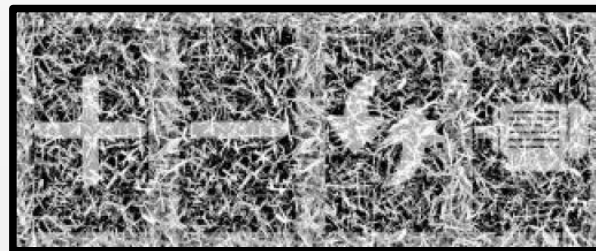
```
+ int : activeModifier()  
+ void : OnTrackableStateChanged(TrackableBehaviour.Status, TrackableBehaviour.Status)  
+ void : OnButtonPressed(VirtualButtonAbstractBehaviour)  
+ void : OnButtonReleased(VirtualButtonAbstractBehaviour)  
+ void : OnTriggerEnter(Collider)  
- void : Start ()  
- void : Update()  
- void : checkSettings()  
- void : setLine()  
- void : UpdateTitle()  
- void : UpdateDesc()  
- void : UpdateMotor()  
- void : updateValue(float)  
- void : activateCanvas()  
- void : changeValue()
```

0..1

1..*

Implementação

- Cadastros de imagens de alvo no Vuforia

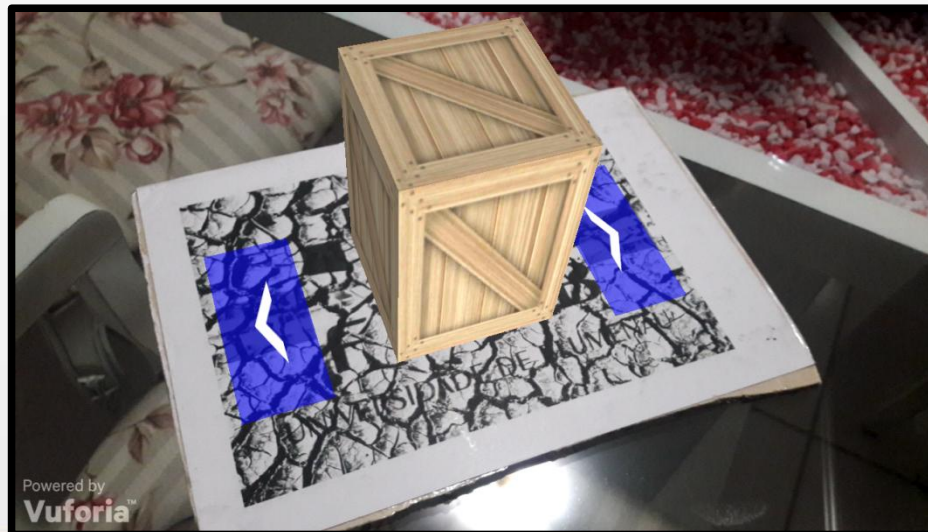
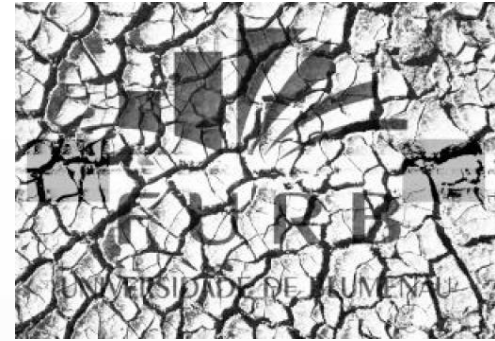


Implementação

- Importação do Vuforia e a base de alvos no Unity
- Câmera para visualizar o ambiente e para detectar os marcadores
 - ARCamera

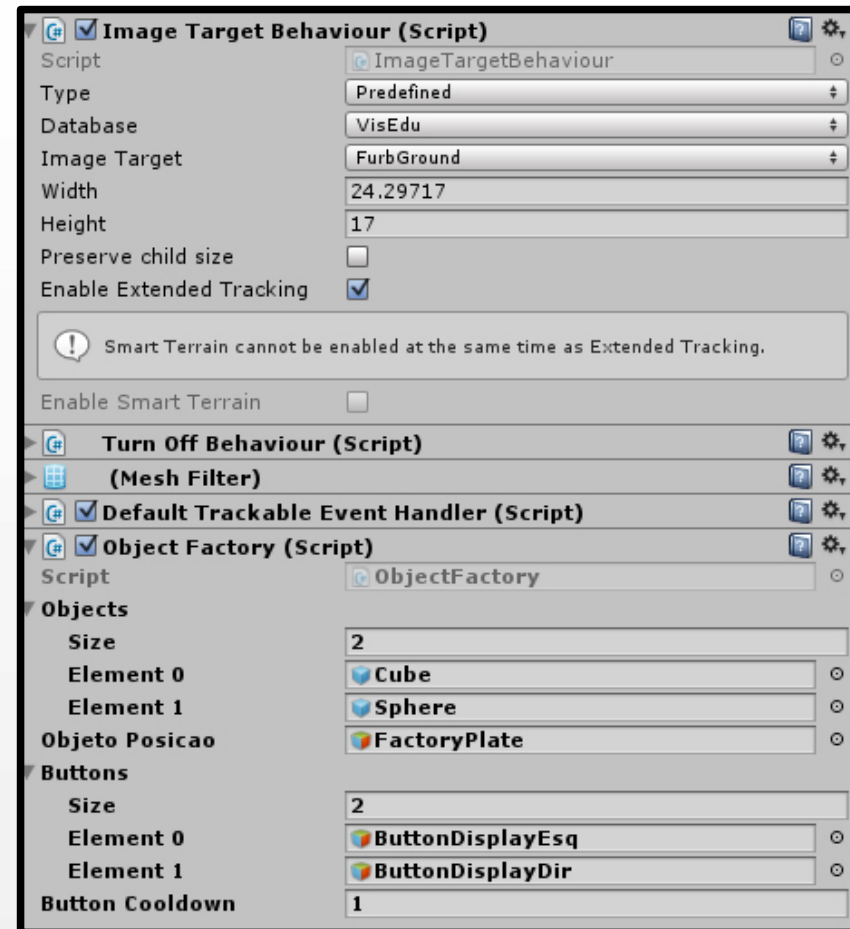
Implementação

- Fábrica de objetos
 - Cria objetos virtuais
 - Possui botões virtuais para trocar de objetos
 - Altera a escala do objeto



Implementação

- Fábrica de Objetos
 - Lista de objetos via Unity
 - Tempo de espera de botões configurado via Unity
 - Extended Tracking



Implementação

. Criação de objetos

...

```
39     if (ObjetoSelecionado != null)
40         Destroy(ObjetoSelecionado);
41     ObjetoSelecionado = Instantiate(objects[value], new Vector3(0, 0, 0),
Quaternion.identity) as GameObject;
42     if (ObjetoSelecionado.GetComponent<Collider>() == null)
43     {
44         BoxCollider bc =
ObjetoSelecionado.AddComponent<BoxCollider>() as BoxCollider;
```

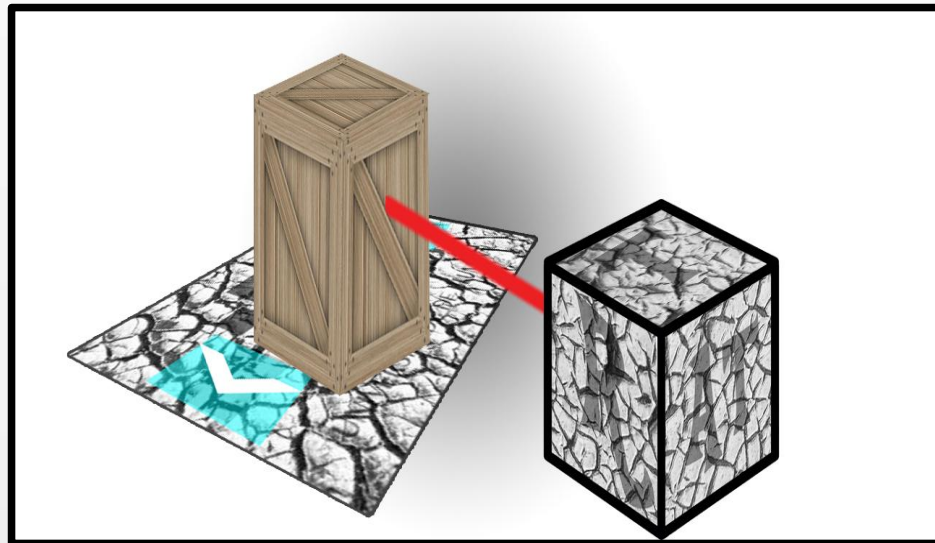
...

```
50     ObjetoSelecionado.transform.SetParent(ObjetoPosicao.transform);
51     ObjetoSelecionado.transform.localScale =
SceneControl.ScaleToFit(ObjetoSelecionado, new Vector3(0.5f,0.5f,0.5f));
52     ObjetoSelecionado.transform.position =
SceneControl.PositionRelativeBottom(ObjetoSelecionado.GetComponent<Rend
erer>().bounds.size, ObjetoPosicao.transform.position);
```

...

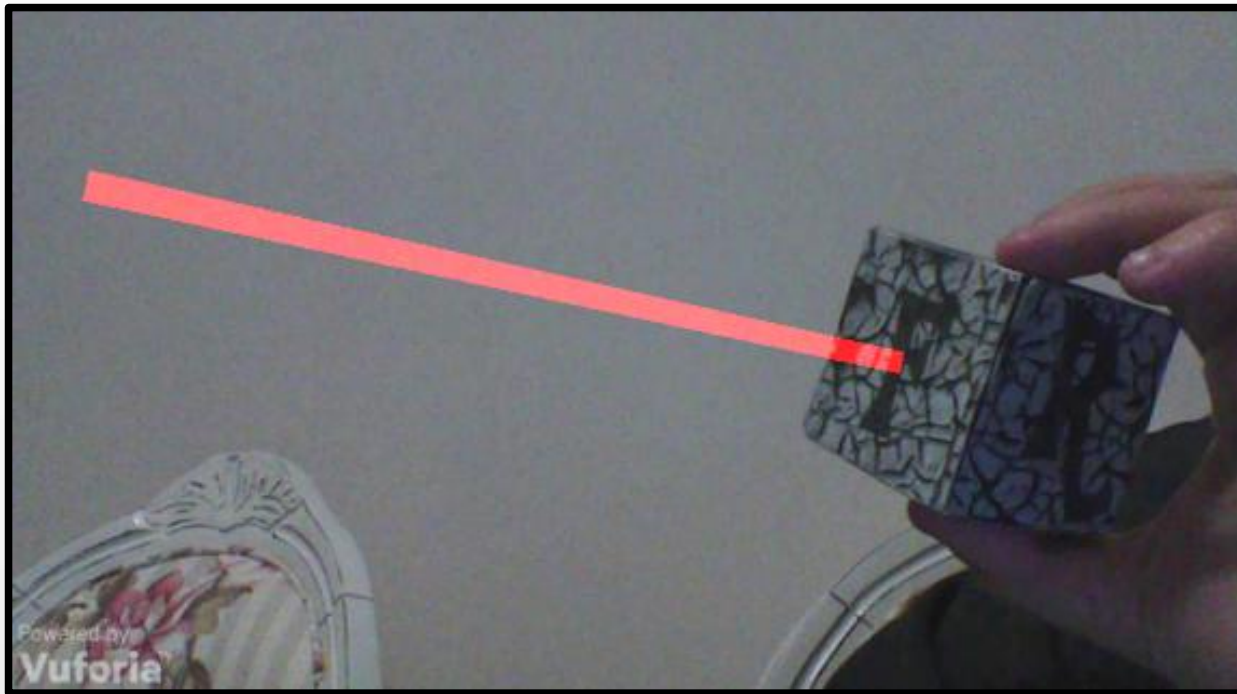
Implementação

- Cubo
 - Movimenta objetos da Fábrica para o ambiente virtual
 - Distancia da linha configurada via Unity



Implementação

- Cubo
 - Possui uma linha virtual para agarrar objetos virtuais



Implementação

. Cubo

- Verifica objetos na distancia definida
- Muda a cor da linha seguindo a coloração de semáforos

...

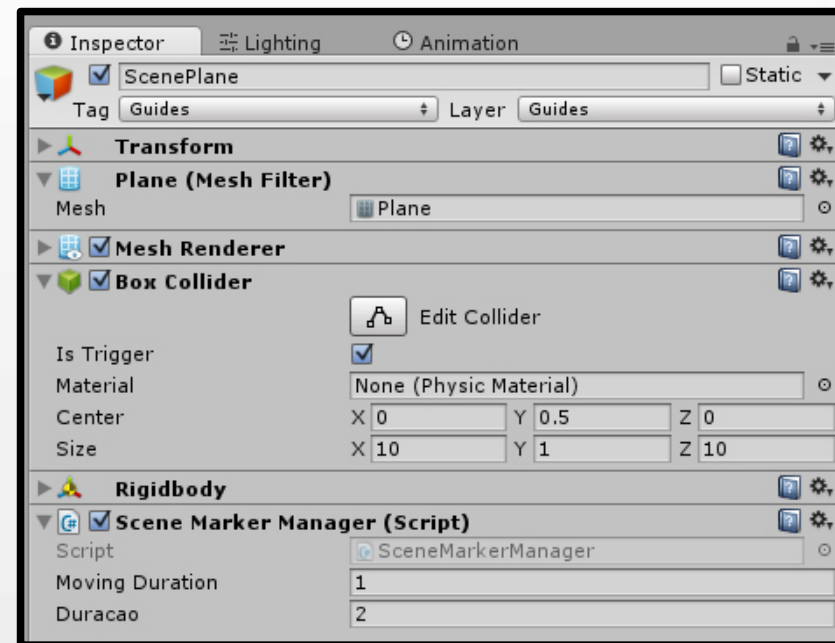
```
28  if (objetoSegurado == null) {
29      if (Physics.Raycast(ray, out hitInfo, distancia + 2, maskLayer)) {
30          if (hitInfo.distance > distancia) {
31              timer = 0;
32              gameObject.GetComponent<Renderer>().material.color = Color.red;
33              line.SetColors(Color.red, Color.red);
34          } else {
35              timer += Time.deltaTime;
36              line.SetColors(Color.yellow, Color.yellow);
37              gameObject.GetComponent<Renderer>().material.color = Color.yellow;
```

...

Implementação



- Alvo
 - Posiciona os objetos virtuais agarrados pelo Cubo no ambiente
 - Tempo de movimentação e de duração configuráveis via Unity



Implementação

- Alvo

- Utiliza o tratamento de colisão para obter os objetos virtuais

...

```
54 void OnTriggerStay(Collider colliderInfo) {
55     if (!SceneControl.isChildFactoryPlate(colliderInfo.gameObject) &&
        !SceneControl.isChildObjectFactory(colliderInfo.gameObject) &&
        colliderInfo.gameObject.tag!="GuidesConnector" &&
        colliderInfo.gameObject.tag!="Guides") {
56         GetComponent<Renderer>().material.color = Color.yellow;
57         if (timer >= duracao || objetoMovido != null) {
58             objetoMovido = colliderInfo.gameObject;
59             objetoMovido.transform.SetParent(null);
```

...

```
67 void OnTriggerExit(Collider colliderInfo) {
68     GetComponent<Renderer>().material.color = Color.red;
69     timer = 0;
```

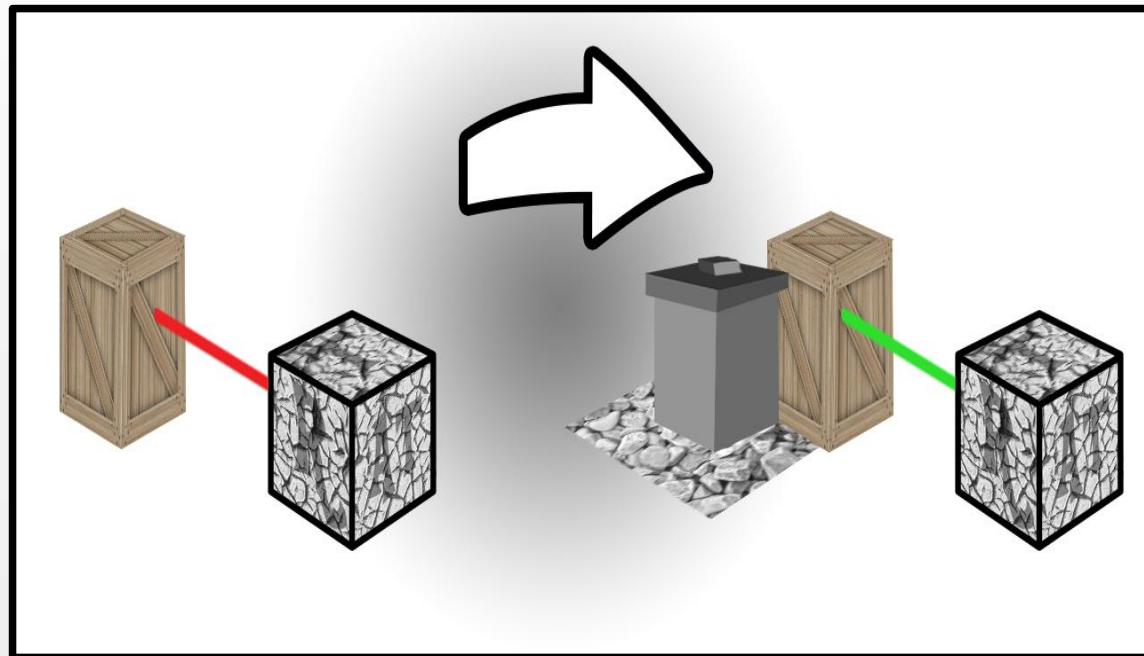
...

Implementação



- Lixo

- Destrói objetos virtuais agarrados pelo Cubo
- Possui animação para definir o progresso



Implementação

• Lixo

- Possui tratamento de colisão semelhante ao Alvo
- Tempo configurável via Unity

...

```
33 void OnTriggerEnter(Collider colliderInfo) {
34     if (colliderInfo.gameObject.tag.Equals("Guides") ||
        colliderInfo.gameObject.tag.Equals("GuidesConnector"))
35         return;
36     if (colliderInfo.gameObject.transform.parent != SceneControl.getFactoryPlate() &&
        colliderInfo.gameObject.transform.parent != SceneControl.getObjectFactory()) {
37         timer = 0;
38         anim.SetBool("opening", true);
```

...

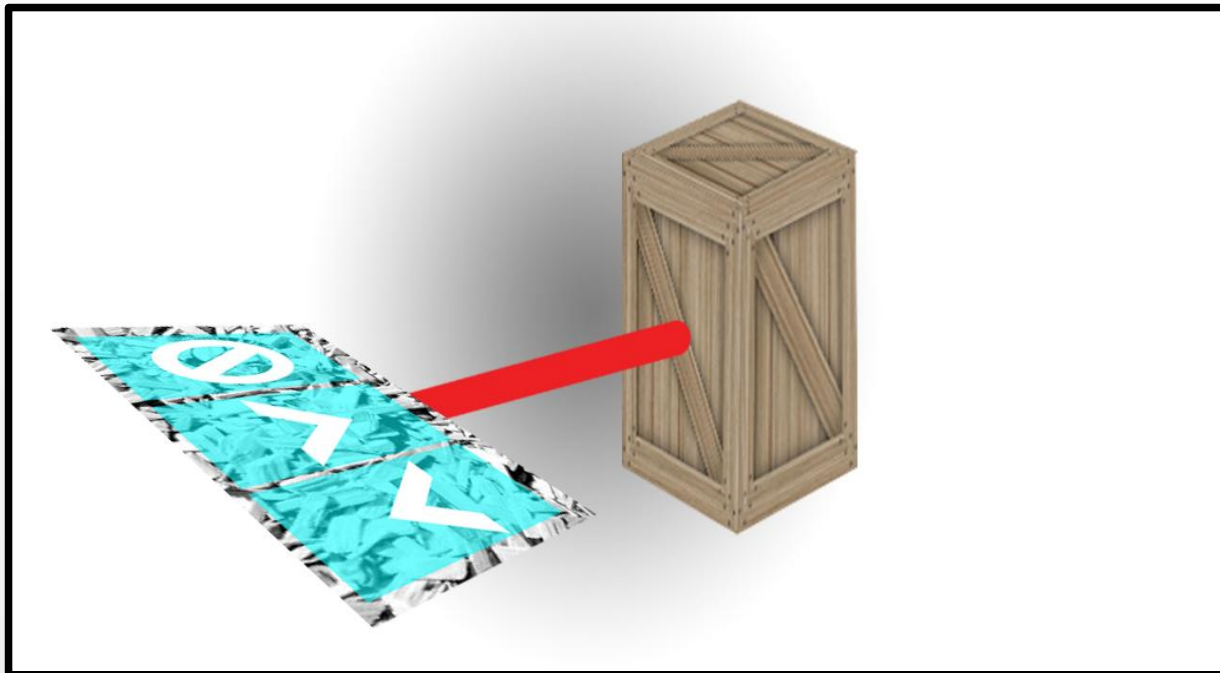
```
57 void OnTriggerExit(Collider colliderInfo) {
58     if (colliderInfo.gameObject.transform.parent != SceneControl.getFactoryPlate()
59     && colliderInfo.gameObject.transform.parent !=
        SceneControl.getObjectFactory()){
60         anim.SetBool("opening", false);
```

Implementação



• Opção

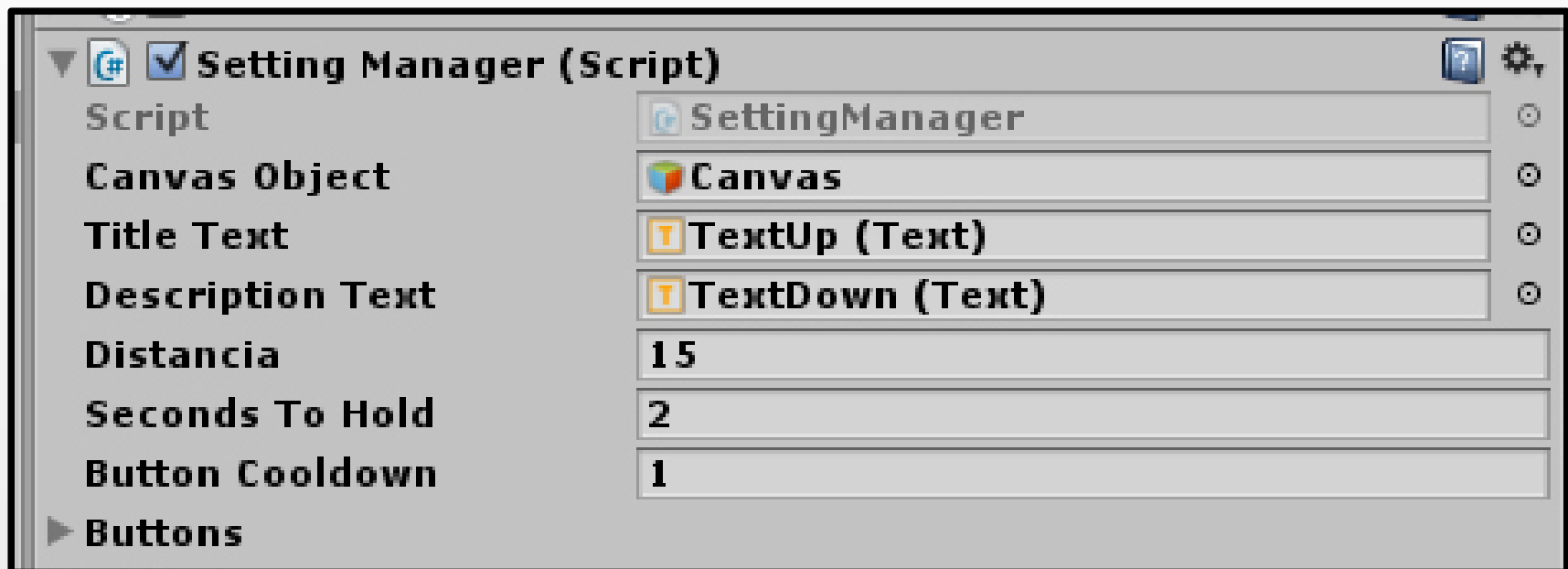
- Possui uma linha virtual para conectar a objetos virtuais
- Seleciona um atributo a ser modificado pelo Modificador



Implementação

• Opção

- Valores da distancia da linha virtual, tempo para conectar ao objeto virtual e tempo de espera configurados via Unity



Implementação

• Opção

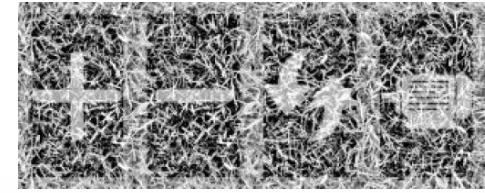
- Método para conectar a objetos semelhante ao Cubo
- Possui botões virtuais para trocar o atributo a ser alterado
- Possui um botão virtual para cancelar a conexão com o objeto virtual

...

```
235 private void getObjectByLine() {
236     ray = new Ray(posicaoLineStart, transform.forward);
237     RaycastHit hitInfo;
238     if (objetoSelecionado == null) {
239         if (Physics.Raycast(ray, out hitInfo, distancia + 2, maskLayer)) {
240             if (SceneControl.isChildObjectFactory(hitInfo.collider.gameObject)) {
241                 if (hitInfo.distance > distancia) {
242                     timer = 0;
243                     line.SetColors(Color.red, Color.red);
```

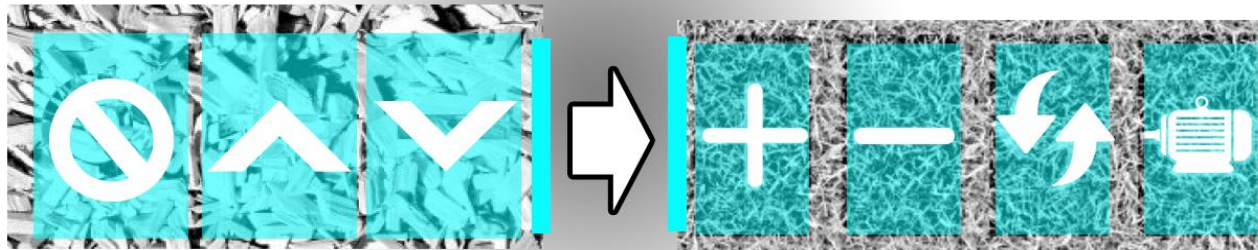
...

Implementação

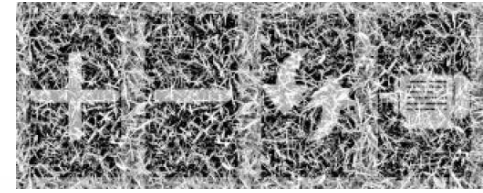


• Modificador

- Possui uma linha virtual para conectar com o marcador Opção
- Modifica um atributo do objeto virtual selecionado pelo marcador Opção

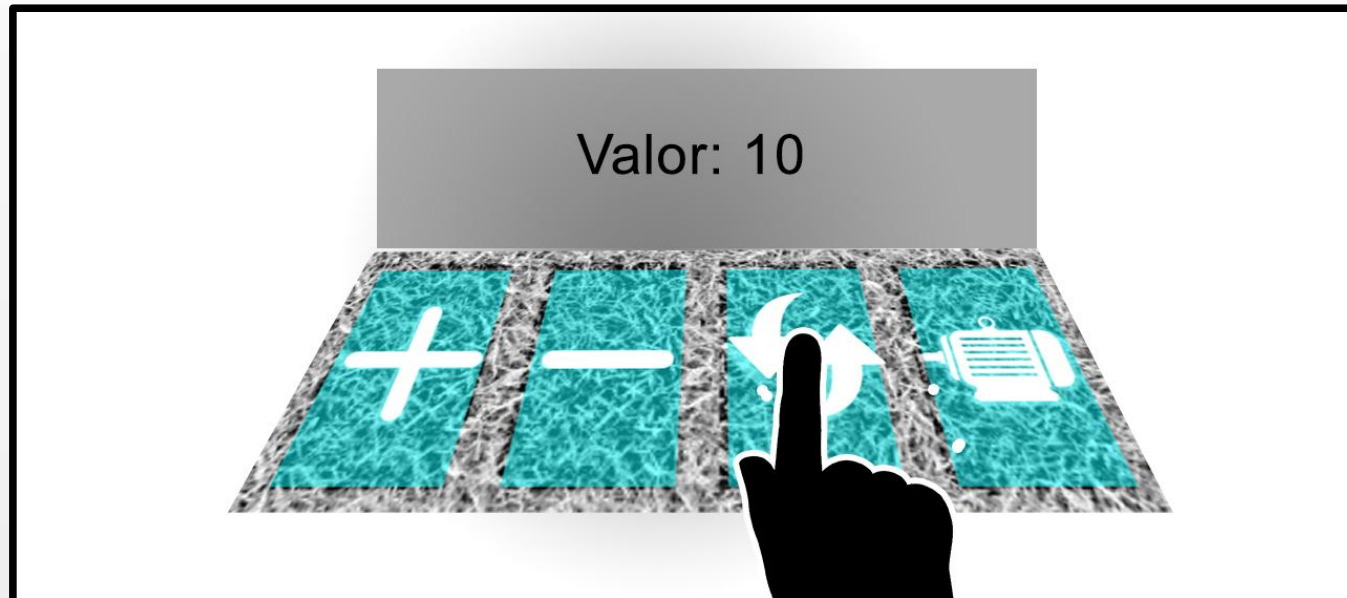


Implementação



• Modificador

- Possui botões virtuais para alterar os valores dos atributos do objeto virtual selecionado
- Possui um botão virtual para alterar o valor do Modificador
- Possui um botão virtual com a funcionalidade Motor



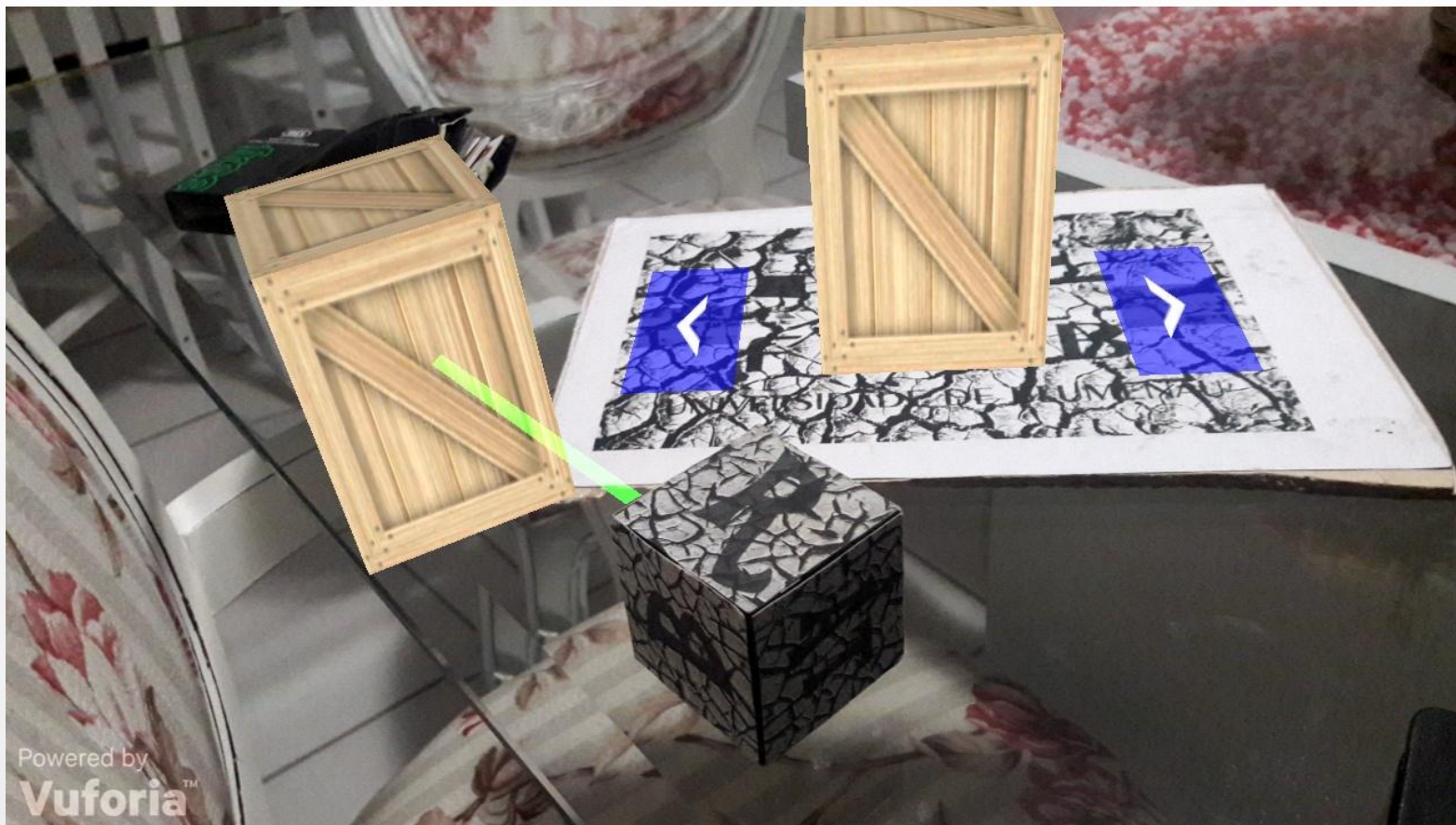
Operacionalidade da Implementação

- Tela Inicial



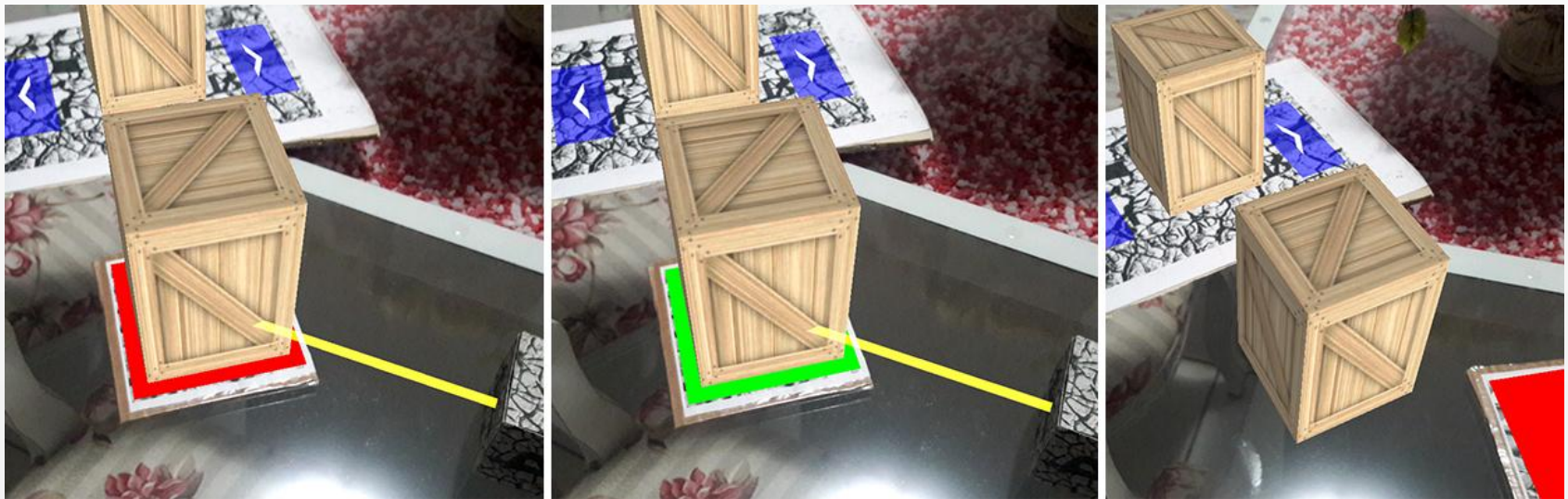
Operacionalidade da Implementação

- Cubo agarrando objeto virtual



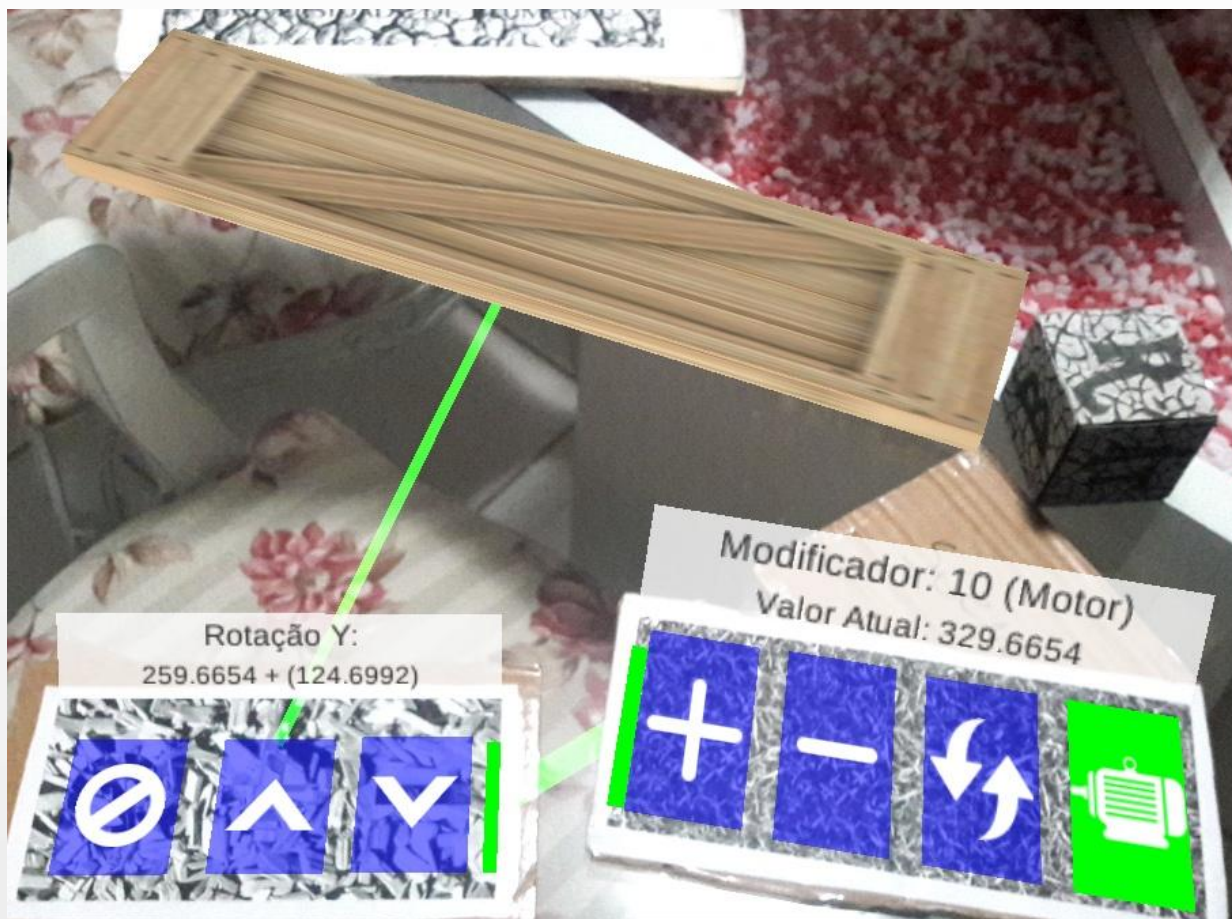
Operacionalidade da Implementação

- Funcionalidade do marcador Alvo



Operacionalidade da Implementação

- Demonstração da alteração de atributos



Resultados e Discussões

• Teste realizado por questionário

Perfil de Usuário	
Sexo	80% masculino 20% feminino
Idade	60% entre 21 a 25 anos 20% entre 26 a 30 anos 20% entre 16 a 20 anos
Nível de escolaridade	100% ensino superior incompleto
Você utiliza o computador com qual frequência	100% frequentemente
Você utiliza dispositivos móveis com qual frequência	80% frequentemente 20% às vezes
Qual o seu grau de familiaridade com Realidade aumentada	80% já utilizei 20% conheço, mas nunca utilizei

Resultados e Discussões

• Teste realizado por questionário

Questionário de usabilidade	
Das atividades solicitadas, quantas atividades você conseguiu executar sem auxílio?	60% todas 40% a maior parte
De modo geral, você achou o protótipo intuitivo e fácil de usar?	80% sim 20% não
Qual a sua avaliação da aplicação?	60% bom 40% muito bom

Resultados e Discussões

- Dificuldade em agarrar o objeto virtual
- O aplicativo pode perder a localização se a câmera mover muito rápido
- Dificuldade na orientação do Cubo
- Aumentado a sensibilidade dos botões
- Aumentado a linha virtual do cubo

Conclusões e Sugestões

- Ferramentas adequadas (Unity com Vuforia)
- Objetivos principais alcançados
- Facilidade em criar um ambiente aumentado
- Facilidade na geração do aplicativo para outras plataformas

Extensões

- Adicionar uma lista no marcador Opções
- Criar marcadores com condição lógica
- Criar atributos no marcador Opção, como animações ou cores
- Criar mais tipos de objetos virtuais como, por exemplo, sons e luzes

Extensões

- .Permitir que outros usuários possam visualizar o ambiente virtual aumentado via Wifi ou Bluetooth
- .Criar uma funcionalidade que permite associar objetos como parentes de outros objetos virtuais
- .Adicionar compatibilidade com a SDK Cardboard ou outros óculos de realidade virtual aumentada

Apresentação Prática

Fim dos Slides