

# LISTAR: BIBIOTECA PARA LISTAGEM DE OBJETOS E INTERFACE DE USUÁRIO EM REALIDADE AUMENTADA

Rafael Diegoli, Dalton Solano dos Reis – Orientador

Curso de Bacharel em Ciência da Computação  
Departamento de Sistemas e Computação  
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brazil  
rdiegoli@furb.br, dalton@furb.br

**Resumo:** Este artigo apresenta o processo de desenvolvimento e testes de uma biblioteca que tem por objetivo abstrair a funcionalidade de expor objetos virtuais e a criação de interfaces tangíveis em realidade aumentada. A biblioteca voltada para a plataforma Unity com o uso do Vuforia e desenvolvida em C#, é composta pelo módulo de lista o qual permite a navegação entre objetos virtuais e o módulo de interface que traz alguns componentes tradicionais de tela para o ambiente virtual. Para avaliar seu desempenho foram realizados testes tanto com alunos da computação observando a perspectiva do desenvolvedor, quanto com alunos de ensino médio e fundamental experimentando o aplicativo de testes para avaliar a usabilidade dos componentes. Com os resultados obtidos, a biblioteca se mostrou útil para o desenvolvimento de outras aplicações, bem como seus componentes se mostraram aptos para servirem de interface com o usuário final, cumprindo com o esperado.

**Palavras-chave:** Realidade aumentada. Interface tangível. Unity. Vuforia.

## 1 INTRODUÇÃO

Na Realidade Aumentada (RA) o real e o virtual se mesclam em camadas, melhorando as interações dos usuários com as informações digitais, o que tem levado à expansão da utilização da realidade aumentada em muitos domínios (ZANDOMENEGUI, 2014 p. 264). "A realidade aumentada mistura em tempo real, objetos virtuais tridimensionais gerados por computador com elementos do ambiente físico" (RODRIGUES; SATO; BOTEGA, 2012 p. 1). "[...] Com o mesmo objetivo da RA, as interfaces tangíveis visam contribuir para o processo interativo por meio de objetos físicos reais, toques e marcadores, utilizados como coleção de entradas no sistema" (RODRIGUES; SATO; BOTEGA, 2012 p. 1).

"Atualmente há maior facilidade na adoção da RA, devido à evolução de microprocessadores mais velozes e novas tecnologias de placas de vídeo, que permitem camadas de imagens e o suporte necessário para a formação de imagens em 3D [...]" (ZANDOMENEGUI, 2014 p. 277). A fácil adesão à realidade aumentada por parte dos desenvolvedores e também usuários, permitiu o seu uso em diversas áreas: na publicidade, entretenimento, turismo, vendas imobiliárias, museologia, educação, medicina, fisioterapia, arquitetura, teleoperação de robôs, aviação militar, engenharia, treinamento para manutenção de aeronaves, devendo-se a isto o seu potencial para a simulação (ZANDOMENEGUI, 2014 p. 277).

Alguns aplicativos que empregam essas tecnologias em conjunto, desejam disponibilizar ao usuário a chance de visualizar e manipular diversos objetos virtuais diferentes, optando pela criação de um marcador o qual mantém uma lista com esses objetos. Ou até mesmo possibilitar a expansão da área de interação do usuário para o mundo real através de interfaces tangíveis. Em contraste com interfaces tradicionais limitadas à área da tela do dispositivo utilizado, que por sua vez, podem disputar espaço em tela e desviar o foco dos principais atrativos do aplicativo.

Embora exista o kit de desenvolvimento de software (SDK) Vuforia que em conjunto com a *engine* de jogos Unity facilitam o desenvolvimento de aplicativos com RA, componentes que auxiliem na criação de aplicativos com interfaces tangíveis com RA ainda são escassos. O que pode obrigar os desenvolvedores a reinventarem as mesmas funcionalidades ou retirá-las durante o desenvolvimento da aplicação.

Diante do contexto apresentado, o objetivo deste trabalho é disponibilizar uma biblioteca que abstraia a funcionalidade de expor objetos virtuais em forma de lista e o uso de componentes para construção de interfaces de usuário tangíveis. Os objetivos específicos são:

- permitir ao desenvolvedor expor objetos virtuais além de adicionar e remover itens de forma dinâmica em tempo de execução;
- disponibilizar componentes reutilizáveis para interface de usuário no espaço RA;
- desenvolver os componentes de forma que possam ser utilizados em conjunto ou individualmente.

## 2 FUNDAMENTAÇÃO TEÓRICA

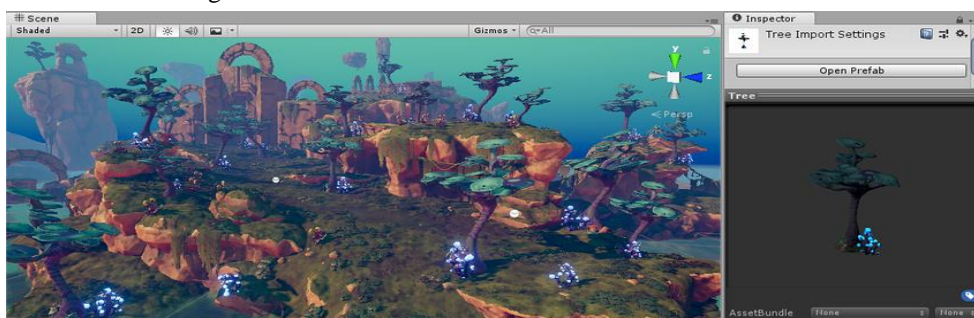
Neste capítulo serão apresentados alguns assuntos que fundamentam a implementação da biblioteca, assim como uma introdução do que representam e como funcionam para melhor compreensão sobre o detalhamento dos componentes desenvolvidos. Serão apresentados os conceitos de *Assets* e *Prefabs* dentro do Unity, bem como uma breve descrição das tecnologias disponíveis para interface de usuário no Unity. Por fim será apresentada neste capítulo uma relação dos trabalhos correlatos que serviram como referência na criação desta biblioteca.

### 2.1 UNITY: RECURSOS REUTILIZÁVEIS E INTERFACES DE USUÁRIO

Segundo Unity (2019a), um *Asset* é a representação de qualquer item que seja possível utilizar em algum jogo ou projeto. Pode vir como um arquivo criado fora do Unity como um modelo 3D, um arquivo de áudio, uma imagem ou qualquer tipo de arquivo que o Unity suporte, assim como alguns tipos de componentes criados no Unity. No manual Unity (2019b) é explicado que o processo de importação lê o arquivo fonte e cria internamente uma representação pronta para uso no projeto, conforme as configurações de importação escolhidas. Desta forma o Unity nunca modifica o arquivo fonte original, havendo alterações nas configurações de importação ou no arquivo fonte, o Unity reimporta o *Asset* para refletir as mudanças. Eles são distribuídos através de pacotes denominados *Asset Packages* que são coleções de arquivos e dados de projetos do Unity ou elementos de projetos, os quais são comprimidos e armazenados em um arquivo. Os *Assets* padrões do Unity bem como os itens na loja de *Assets* estão contidos em *Asset Packages*, conforme é detalhado em Unity (2019c).

De acordo com Unity (2019d), o sistema *Prefab* do Unity permite criar, configurar e armazenar um *GameObject* com seus componentes, propriedades e objetos vinculados como um *Asset* reutilizável. Qualquer alteração realizada em um *Prefab* é automaticamente refletida nas instâncias daquele *Prefab*, permitindo alterar todos os objetos que derivam dele, no entanto ainda é possível alterar as configurações de cada instância para diferenciar as cópias do *Prefab*. Utilizar *Prefabs* também é recomendado quando se deseja instanciar objetos em tempo de execução. A Figura 1 mostra um exemplo da utilização de *Prefabs* (janela do lado direito, a árvore) para replicar o mesmo objeto diversas vezes na cena.

Figura 1 - Cena com diversas instâncias do mesmo *Prefab*



Fonte: Unity (2019d).

A documentação do Unity detalha sobre três tecnologias diferentes para criar interfaces de usuário: *User Interface Elements* (*UIElements*), *Unity User Interface* (*Unity UI*) e *Immediate Mode Graphical User Interface* (*IMGUI*). Conforme a relação em Unity (2019e), *UIElements* ainda em desenvolvimento, é baseada em tecnologias web reconhecidas e suporta estilos, gerenciamento de eventos contextual e dinâmico e persistência de dados. A seguir *IMGUI* é uma ferramenta para interface de usuário direcionada por código, utilizada para ajudar na depuração das aplicações e não é recomendada para interações com o usuário final. Por fim *Unity UI* é uma ferramenta baseada em *GameObjects* a qual utiliza os objetos e seus componentes para agrupar, posicionar e customizar a interface de usuário.

A tecnologia utilizada, *Unity UI*, foi escolhida por ser mais recomendada atualmente para interfaces de aplicativos e será detalhada a seguir. Os principais componentes de interface da *Unity UI* são classificados em componentes Visuais e de Interação. Responsáveis pelos efeitos visuais em geral, os componentes visuais contêm *Text*, *Image*, *Raw Image*, *Mask* e *Effects* Unity (2019f). Enquanto os componentes de interação cujos elementos *Button*, *Toggle*, *Toggle Group*, *Slider*, *Scrollbar*, *Dropdown*, *Input Field* e *Scroll Rect* permitem a interação do usuário via teclado, mouse ou toque, Unity (2019g).

De acordo com Unity (2019h) para o funcionamento de todos os componentes é necessário um objeto *Canvas* e um *EventHandler*. Enquanto o *Canvas* representa a área da tela e agrupa os elementos da interface, o *EventHandler* gerencia os eventos disparados quando houver alguma interação com os componentes. O posicionamento e alinhamento podem ser realizados através da ferramenta *Rect Tool*, a qual auxilia na edição do script *Rect Transform* que controla o espaço que o componente de interface irá ocupar dentro do *Canvas* (Unity, 2019i).

## 2.2 TRABALHOS CORRELATOS

Nesta seção serão apresentados três trabalhos correlatos com aspectos similares à biblioteca desenvolvida. As características do primeiro deles são exibidas no Quadro 1 e trata do aplicativo VisEdu realizado por Silva (2016), em seguida o segundo trabalho é apresentado no Quadro 2 e consiste em uma ferramenta para ensino sobre o sistema Solar. A seção dos correlatos se encerra com o Quadro 3 detalhando sobre o AnimAR, uma ferramenta para animações em realidade aumentada com interface de usuário tangível.

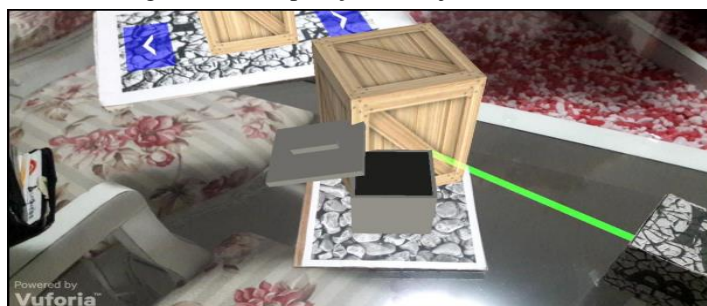
Quadro 1 - VisEdu

Referência	Silva (2016)
Objetivos	O aplicativo busca possibilitar a manipulação de objetos tridimensionais virtuais, através de uma interface tangível utilizando realidade aumentada.
Principais funcionalidades	Disponível para a plataforma Android, o aplicativo permite a visualização do objeto virtual, a movimentação utilizando um marcador de cubo e a alteração de algumas propriedades do objeto selecionado. Pode-se adicionar o objeto no cenário ou removê-lo aproximando-o para o marcador Alvo ou Lixo respectivamente.
Ferramentas de desenvolvimento	O aplicativo foi desenvolvido utilizando o Unity 3D na versão 5.3.5f1 em conjunto com o ambiente integrado de desenvolvimento (IDE) Microsoft Visual Studio Community 2015, utilizando a linguagem C#. Para a Realidade Aumentada foi utilizado o SDK Vuforia na versão 5.5.9 integrado ao Unity, para os marcadores utilizou-se o programa Photoshop CS6.
Resultados e conclusões	Conforme mencionado pelo autor, os objetivos propostos foram atingidos com resultados satisfatórios apesar de se perceber uma infamiliaridade dos usuários com a Interface de Usuário Tangível.

Fonte: elaborado pelo autor.

A Figura 2 mostra um exemplo da manipulação disponível no VisEdu. Pode ser observado ao fundo da imagem o marcador *Fábrica* com os botões azuis e a visualização do objeto, no canto inferior direito é possível ver o marcador de cubo com o feixe de luz verde agindo como um cursor. No centro da imagem o objeto sendo arrastado pelo cursor para o marcador *Lixo* a fim de removê-lo da cena virtual.

Figura 2 - Manipulação do objeto no VisEdu



Fonte: Silva (2016).

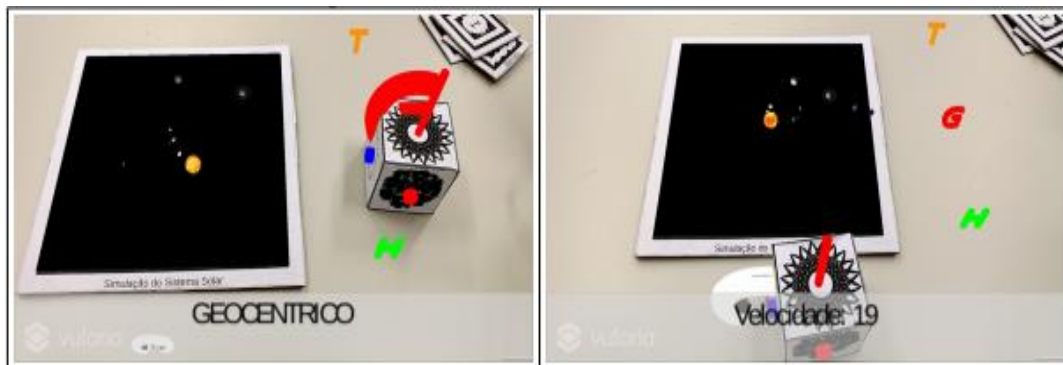
Quadro 2 - Ferramenta para ensino do sistema Solar utilizando Realidade Aumentada

Referência	Schmitz (2017)
Objetivos	A ferramenta tem por objetivo auxiliar no ensino do sistema Solar utilizando realidade aumentada.
Principais funcionalidades	A ferramenta foi dividida em dois módulos os quais compõem o aplicativo de teste. O primeiro chamado de Dissecção do Sistema Solar onde o usuário pode ver os planetas em detalhes, podendo mostrar a estrutura interna dos planetas e informações sobre eles. O segundo denominado Sistema Solar que apresenta todos os oito planetas orbitando o Sol, podendo ser uma das teorias, Heliocêntrica, Geocêntrica ou o sistema Solar segundo Tycho Brahe.
Ferramentas de desenvolvimento	O aplicativo de teste e a ferramenta foram desenvolvidos com o Unity 3D na versão 5.4.1f1 em conjunto com a IDE Microsoft Visual Studio Community 2015, na linguagem C#. Para a Realidade Aumentada foi utilizado o SDK Vuforia na versão 6.2.6, para os marcadores utilizou-se o programa Inkscape e para modelagem dos planetas utilizou-se o Blender na versão 2.76.
Resultados e conclusões	O autor menciona que os usuários sentiram fadiga no braço por segurarem o dispositivo com uma mão após um longo tempo. Outra conclusão é sobre o desempenho regular do aplicativo devido à configuração dos dispositivos. Contudo afirma que o objetivo de disponibilizar uma ferramenta com o intuito de ajudar a ensinar sobre o sistema Solar foi alcançado.

Fonte: elaborado pelo autor.

A Figura 3 demonstra a utilização do módulo Sistema Solar no aplicativo de teste. Com o marcador de cubo agindo como um cursor, na cena do lado esquerdo o cursor é usado para alterar a representação do sistema Solar para a teoria Geocêntrica, enquanto a cena do lado direito mostra o cursor alterando a velocidade do movimento dos planetas de acordo com a rotação do cubo.

Figura 3 - Cenas do módulo Sistema Solar



Fonte: Schmitz (2017).

Quadro 3 - AnimAR

Referência	Reiter (2018)
Objetivos	A ferramenta possui a finalidade de permitir a criação de animações em 3D por meio de uma combinação de Interface de Usuário Tangível e Realidade Aumentada.
Principais funcionalidades	A aplicação da ferramenta disponível para as plataformas Android e iOS, permite a visualização de objetos e cenas virtuais, além de possibilitar copiar os objetos e posicioná-los nas cenas e gravar os movimentos realizados criando-se uma animação que pode ser salva e reproduzida posteriormente.
Ferramentas de desenvolvimento	O aplicativo de teste e a ferramenta foram desenvolvidos com o Unity 3D na versão 2017.4.0f1 em conjunto com a IDE Microsoft Visual Studio Ultimate 2013, utilizando a linguagem C#. Para a Realidade Aumentada foi utilizado o SDK Vuforia na versão 7.0.47, para os marcadores utilizou-se o programa AR Marker Generator e Photoshop CS6.
Resultados e conclusões	Segundo o autor, foi possível obter resultados satisfatórios na utilização do aplicativo embora seu uso em conjunto com o Cardboard tenha se provado insatisfatório.

Fonte: elaborado pelo autor.

A Figura 4 mostra o marcador Seletor no modo de seleção de cenas. Na porção superior da figura é possível observar um objeto virtual no seletor com o número zero, na porção do meio ao pressionar o botão ele altera a cor de fundo. Por fim ao soltar o botão a cena visualizada no marcador Cena é alterada, assim como o número do objeto no marcador Seletor.

Figura 4 - Seleção de cenas do AnimAR



Fonte: Reiter (2018).

### 3 DESCRIÇÃO DA BIBLIOTECA

Este capítulo tem por objetivo destacar os aspectos relacionados ao desenvolvimento da biblioteca e do aplicativo de testes, assim como abordar as técnicas de implementação e como fazer uso dos componentes desenvolvidos. A primeira seção relaciona os componentes que integram a biblioteca ListAR, enquanto a segunda seção explica o desenvolvimento da aplicação de testes utilizando alguns destes componentes.

#### 3.1 BIBLIOTECA LISTAR

Durante o desenvolvimento da biblioteca e dos aplicativos de testes utilizou-se o editor do Unity na versão 2018.1.4f1, enquanto o SDK Vuforia 7.1.34 gerenciava o efeito de realidade aumentada. Os scripts foram escritos na linguagem de programação C# 4.0 dentro da IDE Microsoft Visual Studio Enterprise 2015. Quanto aos marcadores foram utilizadas as imagens do trabalho correlato AnimAR elaboradas por Reiter (2018), para a criação dos marcadores no portal do desenvolvedor do Vuforia e alguns marcadores inclusos nos *Assets* do Vuforia.

A biblioteca ListAR pode ser dividida em dois módulos, o primeiro disponibiliza a funcionalidade de lista propriamente dita enquanto o segundo contém os componentes de interface de usuário para realidade aumentada, embora possuam forte integração, ambos podem ser utilizados separadamente. A seguir serão descritos os aspectos técnicos e utilidades do módulo de lista. Os componentes de interface de usuário, desenvolvidos no módulo de interface, serão detalhados em seguida. Já os diagramas com os componentes dos módulos são apresentados no Apêndice A.

A fim de alcançar a navegação entre objetos e seus materiais, foram criados os scripts `ListAR` e `ListARItem`. Ambos funcionam como listas visto que implementam a interface `IList<T>` disponibilizada pelo Framework .NET. A classe `ListARItem` pode agrupar objetos da classe `Material` pertencente ao Framework do Unity, enquanto a classe `ListAR` agrupa objetos do tipo `ListARItem`. Ambos implementam a interface `IIterableCollection` da biblioteca ListAR, a qual expõe os métodos básicos para a navegação das listas.

A classe `ListARItem` serve como uma estrutura de dados que encapsula um `GameObject` e uma lista interna de materiais, permitindo incluir, excluir e visualizar diferentes cores ou texturas para o mesmo objeto. Desenvolvedores podem criar classes que herdam de `ListARItem` para adicionar ou sobrescrever algumas funcionalidades. É possível vincular métodos aos eventos disponíveis. Dessa forma, pode-se realizar uma ação específica quando são adicionados ou removidos materiais e quando o material do objeto é alterado.

A classe `ListAR` por outro lado opera como um script do Unity, sendo vinculado aos objetos da cena no editor. Ao mesmo passo que a classe `ListARItem` mantém vários materiais, a classe `ListAR` mantém vários objetos do tipo `ListARItem` e disponibiliza eventos para ações específicas quando objetos são adicionados, removidos ou o objeto exibido é alterado. Ela controla o modo de iteração, ou seja, se a navegação deve ser entre os objetos ou os materiais do objeto atual, através das propriedades `IterationType` e `CurrentIterableCollection`. Para utilizar a funcionalidade da lista, deve-se aplicar o script a um objeto da cena e configurar a propriedade `DisplayObj` com um objeto na posição e com o tamanho que deseja aplicar aos itens da lista. No método `Update` do `ListAR`, a posição e rotação do objeto exibido são atualizadas com base no `DisplayObj`. Quando o objeto exibido é alterado, sua escala é reduzida a zero, ficando invisível ao usuário e então o novo objeto aplica a escala do `DisplayObj` ficando visível para o usuário. Os objetos apenas são excluídos da cena quando forem removidos da lista.

O módulo de interface foi criado com a intenção de abstrair o uso dos botões virtuais do SDK Vuforia e trazer componentes tradicionais de tela, semelhantes aos disponibilizados pelo Unity, como `CheckBox`, `Scroll` e `ComboBox` para o contexto da realidade aumentada. Internamente quase todos os componentes desenvolvidos fazem uso dos botões do Vuforia, por isso precisam estar vinculados a um objeto do tipo `ImageTarget` com o devido script de canvas aplicado, conforme será explicado posteriormente. O módulo de interface do ListAR disponibiliza em forma de *Prefabs* do Unity componentes com funcionalidade de: botão, *checkbox*, *scroll/slider* e *combo box*.

O primeiro componente criado e também o mais básico é o `VirtualButton`. Este componente expõe a funcionalidade de um botão virtual do Vuforia adicionando algumas opções para personalização. É possível configurar qual será a imagem de fundo do botão quando estiver pressionado ou em repouso, bem como o seu ícone. Este *Prefab* é utilizado pelo `CheckBox`, `Scroll` e `Combo Box`. Ele pode ser utilizado para interagir com os objetos do módulo de lista, caso a propriedade `Name` do script `VirtualButtonBehaviour` seja configurada com o nome de uma das constantes do enum `CanvasVuforiaAction`, podendo também exercer outras ações utilizando o evento `ExecuteAction`. O Quadro 4 exhibe as ações predefinidas caso o desenvolvedor queira se abstrair da manipulação da lista de objetos enquanto a Figura 5 demonstra a interação através dos botões.



Quadro 4 - Ações predefinidas para interação com a lista virtual

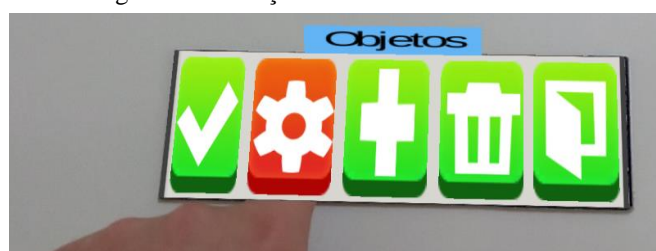
```

8      public enum CanvasVuforiaAction
9      {
10         Nothing,
11         Next,
12         Previous,
13         ChangeType,
14         BackToMenu,
15         AddItems,
16         DeleteItem,
17         SelectItem,
18         SetVisible,
19     }

```

Fonte: elaborado pelo autor.

Figura 5 - Interação com os botões do ListAR



Fonte: elaborado pelo autor.

O próximo Prefab é o VirtualCheckBox, o qual se comporta como um *checkbox*. Mantendo as funcionalidades do VirtualButton ele possui um valor do tipo *bool* que é alterado toda vez que é pressionado. A personalização é semelhante ao do botão, adicionando uma opção para o ícone quando o *checkbox* estiver marcado ou desmarcado e uma propriedade *Text* que define a descrição dele através do *TextMesh* interno. Diferentemente do botão o *checkbox* possui o evento *CheckChanged*, disparado ao trocar de marcado para desmarcado e vice-versa, para disparar ações específicas. O Prefab VirtualCheckBox é amplamente utilizado no Scroll e no Combo Box. A Figura 6 mostra a utilização do VirtualCheckBox para tornar o objeto atual visível ou não.

Figura 6 - Exemplo de utilização do VirtualCheckBox



Fonte: elaborado pelo autor.

O VirtualScroll é um Prefab mais complexo, principalmente pelo fato de instanciar dinamicamente algumas cópias do Prefab VirtualCheckBox. O número exato de cópias é definido pelo desenvolvedor através da propriedade *Elements*, onde também é definido se cada elemento terá uma descrição ao lado ou não. É possível definir uma descrição para as extremidades com as opções *MinTextString*, *MaxTextString*, uma descrição geral do componente com *DescriptionText* e definir se esta descrição será vertical ou não com a propriedade *VerticalText*. O Quadro 5 apresenta como os botões virtuais são instanciados.

Quadro 5 - Trecho do método responsável por instanciar dinamicamente os botões virtuais

```

170     for (int i = 0; i < Elements.Length; i++)
171     {
172         var vbPosition = new Vector3(transform.position.x, transform.position.y, transform.position.z - newScale * (i * Elements.Length));
173
174         var virtualStep = Instantiate(VirtualStepPrefab, vbPosition, transform.rotation, transform);
175         virtualStep.gameObject.name = string.Format("{0}Step{1}", gameObject.name, i);
176         virtualStep.transform.localScale = new Vector3(0.25f, newScale, 0.25f);
177
178         var virtualBtn = virtualStep.GetComponentInChildren<VirtualButtonBehaviour>();
179
180         //Vuforia does not allow setting VirtualButtonName outside editor
181         var fieldNameInfo = virtualBtn.GetType().GetField("mName", BindingFlags.NonPublic | BindingFlags.Instance);
182         fieldNameInfo.SetValue(virtualBtn, virtualStep.gameObject.name);
183
184         virtualBtn.RegisterEventHandler(this);

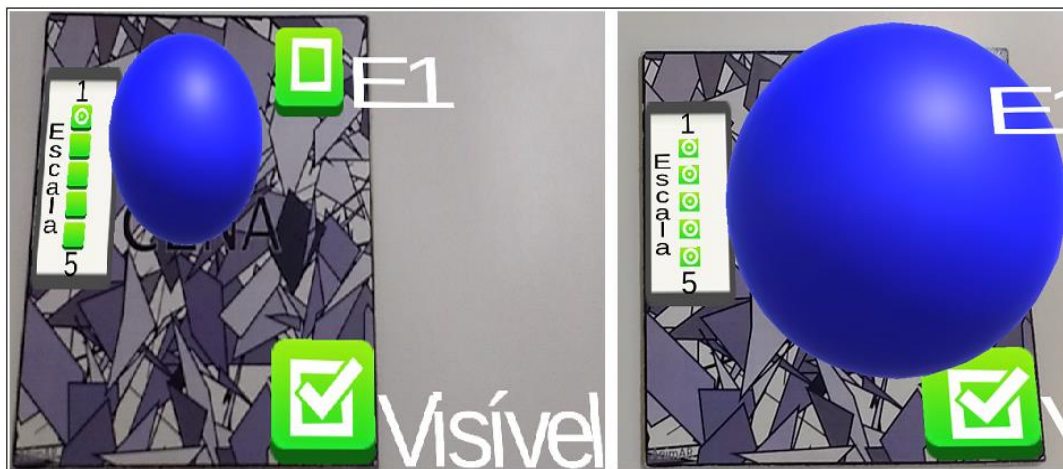
```

Fonte: elaborado pelo autor.

No Quadro 5 é possível observar que na linha 172 é calculada a posição onde o Prefab deve ser criado, enquanto as linhas 174 a 176 instanciam o Prefab na cena e um nome único é atribuído ao objeto. Por último as linhas 178 a 184 configuram o membro privado `mName` utilizado pelo Vuforia para gerenciar os botões virtuais e registram o controlador de eventos para o botão criado.

O componente `VirtualScroll` pode se comportar como um *scroll* (apenas um item selecionado de uma vez) ou como um *slider* (vários itens consecutivos selecionados de uma vez). Este comportamento é definido através da propriedade `BehaviourType`. A interação dele pode ser realizada com o evento `ValueChanged`, disparado quando o valor correspondente à seleção é alterado, vinculando um método a ser executado, ou apenas com a propriedade `Value` que retorna o valor máximo selecionado no Prefab. A Figura 7 traz uma das aplicações deste componente.

Figura 7 - Aplicação do VirtualScroll para alterar a escala do objeto

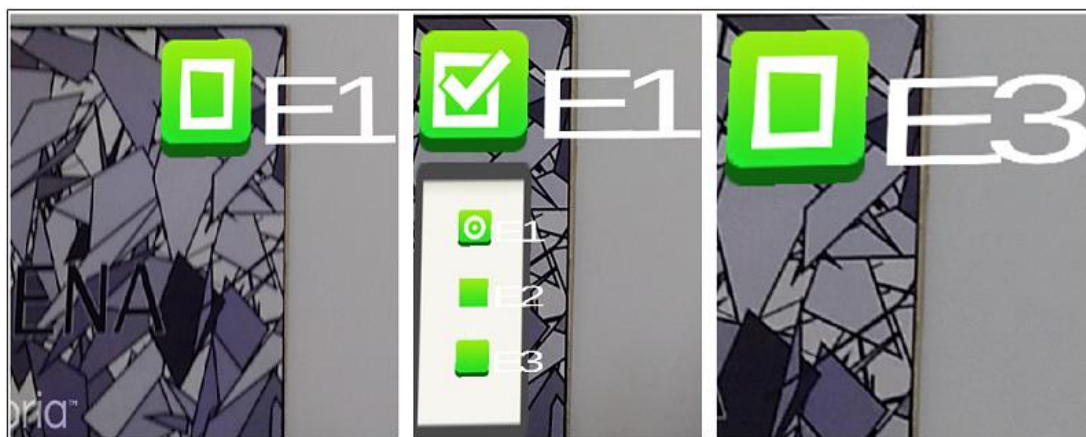


Fonte: elaborado pelo autor.

Na Figura 7 o `VirtualScroll` é utilizado para alterar o tamanho do objeto visualizado, para cada botão selecionado o tamanho aumenta em uma unidade sendo o valor mínimo seu tamanho original. Como o seu `BehaviourType` está com a opção `SliderBehaviour` percebe-se que ao selecionar o tamanho máximo, todos os outros botões aparecem como selecionados. Caso estivesse com a opção `ScrollBehaviour` apenas o botão mais abaixo estaria selecionado.

O `VirtualComboBox` é uma simbiose dos Prefabs `VirtualCheckBox` e `VirtualScroll`, pois a opção selecionada é visualizada no `CheckBox`, que ao ser pressionado exibe a lista de opções num `Scroll`, que por sua vez quando uma das opções é selecionada altera o valor e retorna o estado anterior do `CheckBox` principal. Sua interação pode ser através da propriedade `SelectedIndex` que pode ser utilizada para retornar ou definir qual o índice da opção selecionada dentro do `Scroll` interno. Ou do membro `SelectedValue` que retorna o texto da descrição do `CheckBox` principal. A Figura 8 ilustra a sua utilização na primeira parte está o componente sem qualquer interação, em seguida ao pressionar o `CheckBox` surge o `Scroll` com as opções disponíveis. Selecionando qualquer opção ou pressionando novamente o `CheckBox` a lista de opções deixa de ser exibida, neste caso a última opção foi selecionada e a descrição do `CheckBox` foi atualizada.

Figura 8 - Passos para alterar o item selecionado do VirtualComboBox



Fonte: elaborado pelo autor.

Para o Vuforia reconhecer quando um botão virtual foi pressionado, é necessário adicionar um script ao objeto do marcador que implemente a interface `IVirtualButtonEventHandler` do SDK Vuforia e realize o tratamento. A biblioteca ListAR disponibiliza o script `CanvasVuforiaPlusBehaviour` para este propósito. Quando os Prefabs `VirtualButton`, `VirtualCheckBox` e `VirtualComboBox` são utilizados é necessário adicionar este script e configurar a propriedade `Buttons`, caso contrário os eventos `ExecuteAction` do botão e `CheckChanged` do `CheckBox`, incluindo o `CheckBox` principal do `Combo Box`, não serão disparados. O script `CanvasVuforiaListBehaviour` é uma extensão do `CanvasVuforiaPlusBehaviour` especializada para interações com a lista do ListAR, que permite a navegação através do marcador de cubo e a visualização analítica da lista. Conforme mostra a Figura 9 o marcador Cubo está indicando que pode ser rotacionado para navegar pelos itens da lista, à direita está a visualização analítica da lista de objetos exibindo a descrição dos itens, a qual é atualizada cada vez que o material de um objeto é alterado. Bem como qual é o objeto visualizado no momento com o fundo em um tom mais escuro e sua posição dentro da lista com o *scroll* ao lado, que se ajusta conforme quantidade de itens na lista e o objeto atualmente selecionado.

Figura 9 - Visualização analítica de uma lista do ListAR



Fonte: elaborado pelo autor.

### 3.2 A APLICAÇÃO

Com a intenção de realizar testes de usabilidade da biblioteca ListAR desenvolveu-se o aplicativo dentro do contexto de questões ambientais, mais especificamente sobre o ciclo e a conscientização do uso da água. O aplicativo expõe atividades utilizando tanto os componentes de interface de usuário do Unity quanto os recursos de interface do ListAR. O aplicativo é constituído de três cenas utilizando o *template* 3D do Unity, contando com os recursos do Vuforia para a realidade aumentada, Unity com os componentes de interface de usuário e os módulos de lista e interface do ListAR. A seguir será apresentado como foi desenvolvido o aplicativo apontando quais recursos da biblioteca foram utilizados e na sequência serão detalhadas as telas e atividades propostas pela aplicação.



### 3.2.1 IMPLEMENTAÇÃO

Para realizar o desenvolvimento da aplicação foi criado um script para gerenciar o fluxo de cada cena, estes scripts alteram os componentes presentes conforme as opções do menu principal. Para representar o ciclo da água e a utilização indevida ou consciente da água, foram importadas algumas imagens para o Unity com o formato de *Sprite*. Referente à interface de usuário foram utilizados tanto componentes do *UnityEngine.UI* como da biblioteca *ListAR*, no entanto para enfatizar a comparação entre esses recursos eles são dispostos de forma que o usuário possa avançar nas atividades interagindo apenas com uma interface de cada vez.

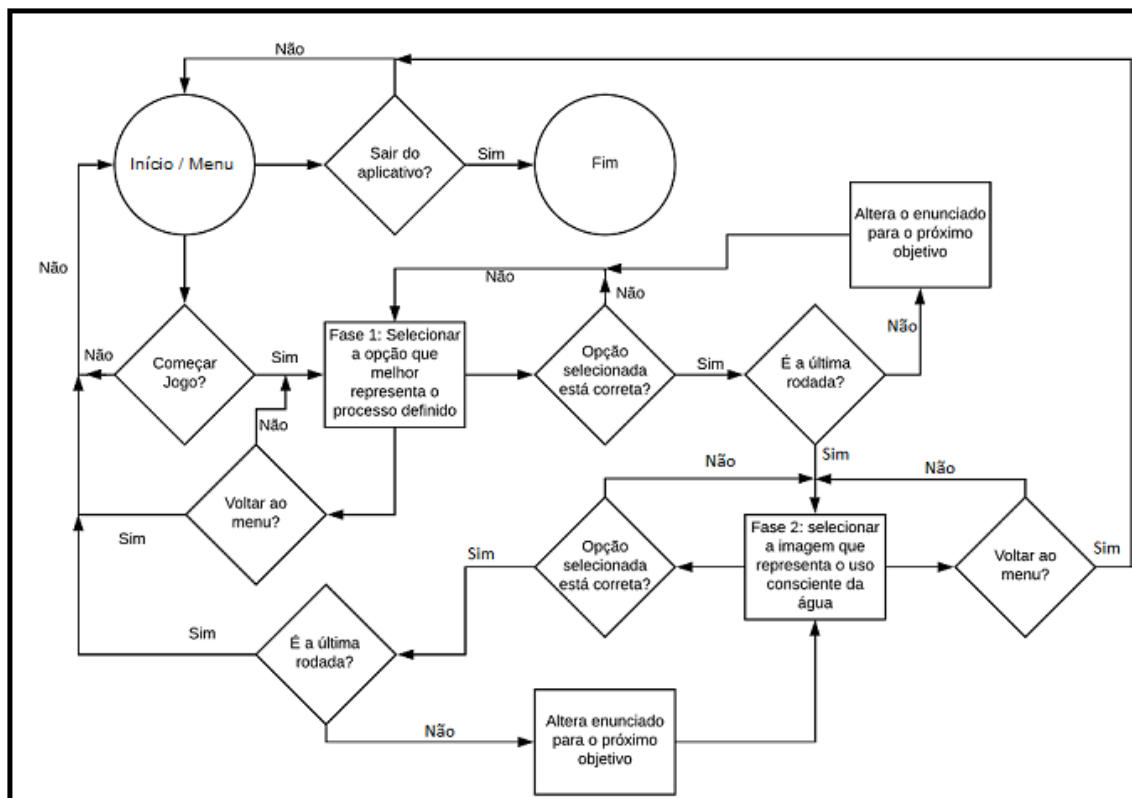
Dentre os *Prefabs* disponibilizados pelo Unity são utilizados o *Button* e *Dropdown*. O *Button* é amplamente utilizado tanto para progredir nas atividades como para voltar ao menu principal, ou ainda sair do aplicativo. Enquanto o *Dropdown* está presente apenas em uma fase como seletor de opções. Existe um objeto do tipo *Text* nas atividades apenas para demonstrar o objetivo atual, não possuindo nenhuma interação direta com o jogador. Ao adicionar qualquer um dos *Prefabs* mencionados, o Unity também adiciona um componente do tipo *Canvas* que serve para o desenho e alinhamento dos objetos subordinados.

A fim de comparar os componentes desenvolvidos e os disponibilizados pelo Unity, foram utilizados os *Prefabs* *VirtualButton*, *VirtualCheckBox* e *VirtualScroll*. O *VirtualButton* equiparado ao *Button* é utilizado amplamente em todas as cenas do aplicativo, inclusive no menu principal. Embora o *Prefab Toggle* do Unity não esteja presente no aplicativo o *VirtualCheckBox* está presente no menu principal operando como *checkbox* tradicional (opção destro ou canhoto) e como *radio button* (opção *Vuforia Tradicional* e *Vuforia Plus*). Por fim, optou-se pelo *VirtualScroll* para ser comparado com o *Dropdown*, pois embora o *VirtualComboBox* também pudesse ser utilizado foi escolhido o *VirtualScroll* por ter uma interação mais simples com o usuário.

### 3.2.2 VISÃO GERAL

O principal objetivo do aplicativo era criar um comparativo entre os componentes de interface de usuário disponibilizados pelo Unity e os da biblioteca *ListAR*. Com esta finalidade a aplicação foi desenvolvida de forma a possibilitar ao jogador escolher qual estilo de interface seria utilizado durante as atividades do jogo. As cenas de cada fase são montadas com os dois tipos de interface coexistindo. Quando o jogo inicia é avaliada a configuração do jogador e desabilitados os objetos de acordo com a opção escolhida. A Figura 10 detalha a sequência das atividades do aplicativo enquanto a Figura 11 mostra como os elementos de interface de usuário foram dispostos na cena.

Figura 10 - Fluxograma do processo da aplicação de testes



Fonte: elaborado pelo autor.

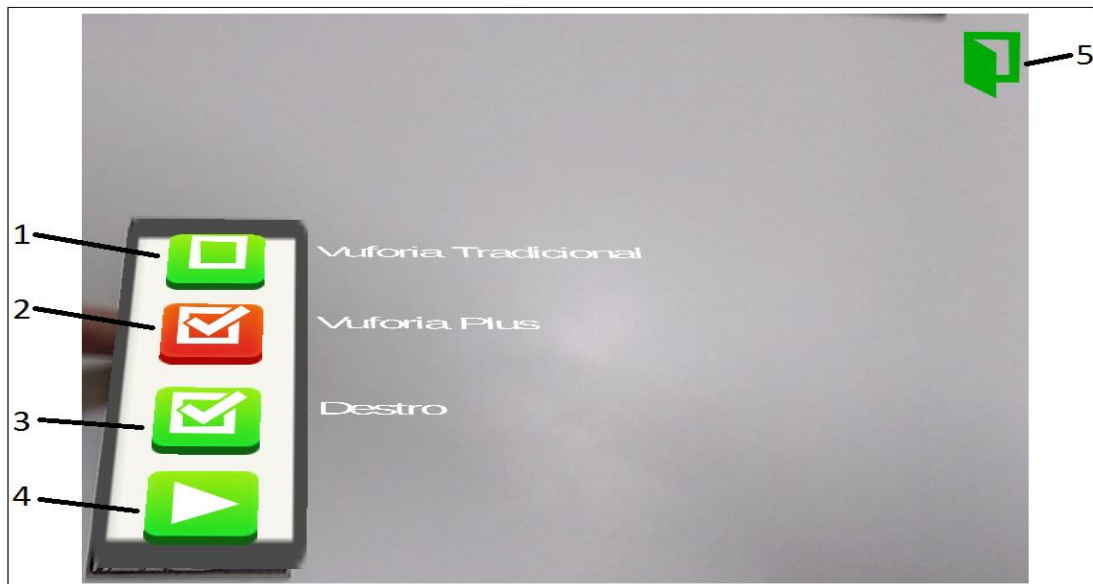
Figura 11 - Grafo de cena da segunda fase do jogo



Fonte: elaborado pelo autor.

Ao iniciar a aplicação o usuário deve apontar a câmera do dispositivo para o marcador Seletor para visualizar o menu principal. Ele é composto pelas opções Vuforia Tradicional e Vuforia Plus, que definem se a interação do usuário será por meio dos componentes do Unity ou do ListAR, respectivamente. Existe a opção para indicar se o jogador é destro ou canhoto, alterando a orientação dos componentes de tela do Unity no modo Vuforia Tradicional e da área que reconhece a rotação do marcador *cubo* no modo Vuforia Plus. Por fim, o botão que inicia as atividades do jogo localizado mais abaixo, conforme Figura 12 e em seguida o Quadro 6 demonstram o menu.

Figura 12 - Menu do aplicativo



Fonte: elaborado pelo autor.

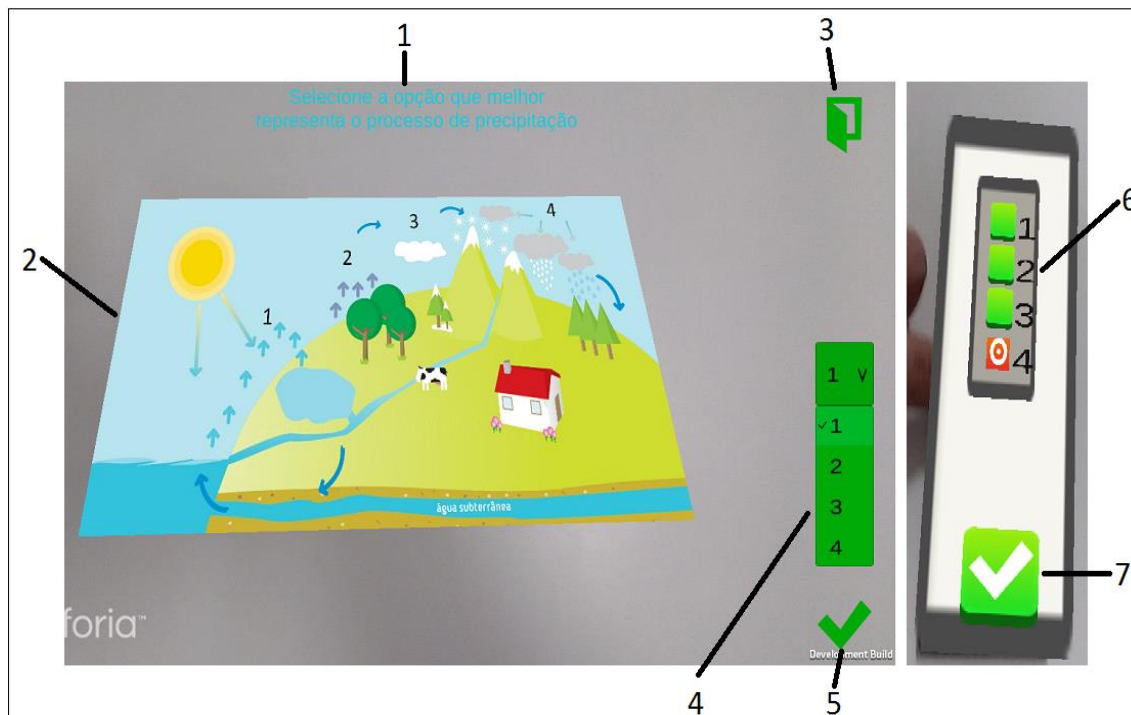
Quadro 6 - Descrição dos elementos no menu do aplicativo

Identificador	Descrição
1	Quando selecionado indica que o jogo utilizará os componentes de tela do Unity.
2	Quando selecionado indica que o jogo utilizará os componentes de tela do ListAR.
3	Indica qual é a mão dominante do jogador e ajusta alguns elementos para melhor ergonomia.
4	Botão para iniciar o jogo.
5	Botão para sair do aplicativo.

Fonte: elaborado pelo autor.

A primeira atividade do jogo é composta pela imagem do ciclo da água visível através do marcador Cena, do texto na parte superior da tela explicando o objetivo atual dentro da atividade e do seletor de opções com o botão de confirmar, na tela ou no marcador Seletor. Para avançar nesta fase o jogador deve ler o enunciado do texto, observar a imagem do ciclo da água e marcar qual opção melhor representa o processo descrito no enunciado. Caso o jogador acerte o enunciado irá mudar, e após quatro rodadas o jogo irá para a próxima fase. A Figura 13 e o Quadro 7 demonstram a primeira fase do jogo.

Figura 13 - Cena da primeira fase do jogo



Fonte: elaborado pelo autor.

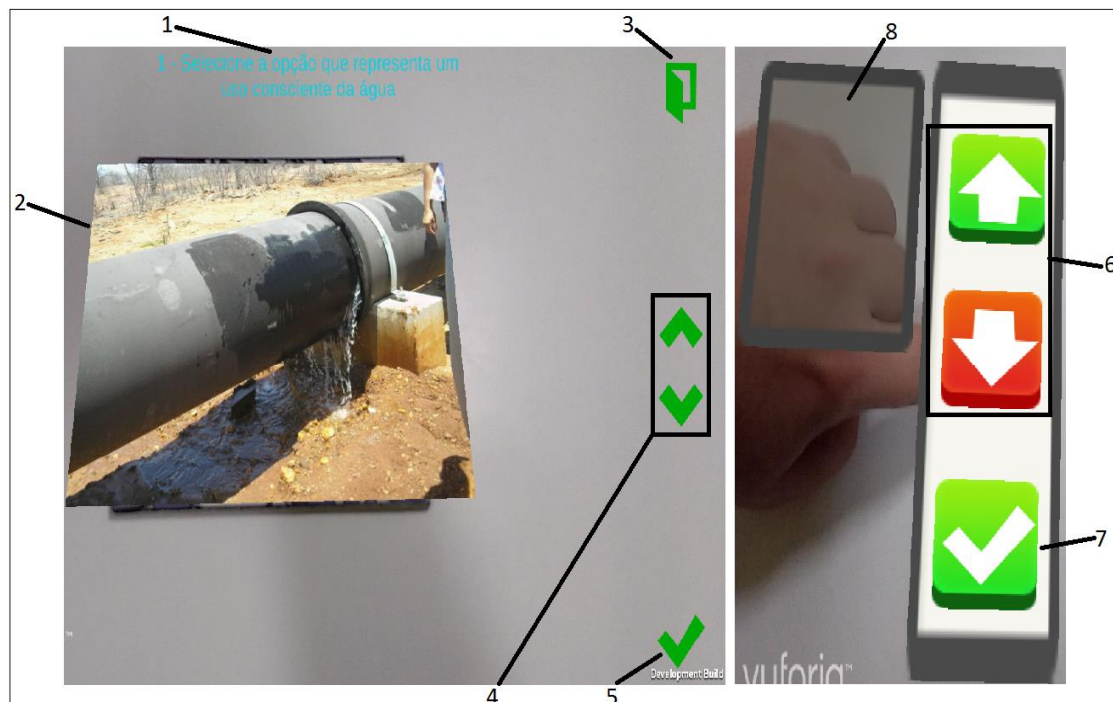
Quadro 7 - Descrição dos elementos na primeira fase do jogo

Identificador	Descrição
1	Enunciado indicando qual o objetivo atual que o jogador deve cumprir.
2	Imagem do ciclo da água para o jogador consultar.
3	Botão para voltar ao menu principal.
4	Componente para selecionar as opções no modo Vuforia Tradicional.
5	Botão para confirmar a resposta no modo Vuforia Tradicional.
6	Componente para selecionar as opções no modo Vuforia Plus.
7	Botão para confirmar a resposta no modo Vuforia Plus.

Fonte: elaborado pelo autor.

Em sequência, a segunda fase trata sobre a conscientização do uso da água. Apontando a câmera para o marcador Cena surgirá a imagem de uma situação cotidiana que pode ser um bom ou mau uso da água. Para avançar deve-se selecionar a imagem que representa o uso consciente da água conforme descrito pelo enunciado na parte superior da tela. Esta atividade possui três rodadas no total. Utilizando os botões de setas é possível trocar a imagem exibida no marcador enquanto o botão confirma se é a imagem correta ou não, seja em tela ou no marcador Seletor. Nesta fase para o modo Vuforia Plus foi adicionada a navegação através do marcador cubo, ou seja, posicionando o cubo na área sombreada ao lado do marcador Seletor é possível girar o cubo para navegar rapidamente entre as imagens. Cada vez que o jogador acertar o enunciado será alterado e na lista de imagens serão adicionadas duas imagens com o uso incorreto da água e uma com o uso correto, após as três rodadas caso o jogador pressione o botão de confirmar o aplicativo retornará para o menu principal. A Figura 14 e o Quadro 8 detalham a segunda fase.

Figura 14 - Cena da segunda fase do jogo



Fonte: elaborado pelo autor.

Quadro 8 - Descrição dos elementos na segunda fase do jogo

Identificador	Descrição
1	Enunciado indicando qual o objetivo atual que o jogador deve cumprir.
2	Imagem do uso cotidiano da água para o jogador consultar.
3	Botão para voltar ao menu principal.
4	Botões para navegar pelas imagens no modo Vuforia Tradicional.
5	Botão para confirmar a resposta no modo Vuforia Tradicional.
6	Botões para navegar pelas imagens no modo Vuforia Plus.
7	Botão para confirmar a resposta no modo Vuforia Plus.
8	Espaço para ativar a navegação através do marcador de cubo no modo Vuforia Plus.

Fonte: elaborado pelo autor.

## 4 RESULTADOS

Este capítulo apresenta um comparativo da biblioteca desenvolvida com os trabalhos correlatos, e algumas observações dos testes realizados durante o desenvolvimento da biblioteca e do aplicativo. Assim como os resultados obtidos nas experiências com usuários finais e desenvolvedores. A primeira seção aponta os aspectos relevantes dos correlatos e as principais observações sobre o desenvolvimento dos módulos de lista, de interface e suas utilizações na aplicação de teste. Por fim, a segunda seção destaca os resultados obtidos com a aplicação dos testes, na perspectiva dos usuários finais ao utilizarem o aplicativo. E na visão dos desenvolvedores ao realizarem as tarefas propostas utilizando os componentes da biblioteca.

### 4.1 COMPARATIVO COM CORRELATOS E TESTES DURANTE DESENVOLVIMENTO

Os trabalhos apresentam características em comum com o trabalho desenvolvido, por exemplo, todos os correlatos utilizam interfaces tangíveis com a realidade aumentada. Apenas o trabalho de Schmitz (2017) não possui nenhum mecanismo de seleção de objetos virtuais, no entanto é o único a disponibilizar uma biblioteca reutilizável no desenvolvimento de outras aplicações. Tanto o VisEdu quanto o AnimAR implementaram uma lista fixa de possíveis objetos que podem ser visualizados pelo seletor. Para atender esta funcionalidade ambos precisaram desenvolver o mesmo tratamento de alterar o objeto visualizado quando um botão virtual é pressionado, utilizando diretamente os componentes do Vuforia.

Com base no exposto anteriormente, foi possível se basear nos seletores de objetos do VisEdu e do AnimAR, na interface de usuário tangível utilizada por todos e nos módulos reaproveitáveis do trabalho de Schmitz (2017), para definir as funcionalidades que a biblioteca desenvolvida deveria atender. No trabalho de Schmitz (2017), foi observado o modo como os scripts e componentes foram separados em módulos a fim de se obter uma melhor reusabilidade deles.



A partir do VisEdu surgiu a ideia de criar uma lista genérica e dinâmica para objetos virtuais, enquanto da disposição, customização e funcionalidade dos botões no AnimAR partiu a ideia de disponibilizar componentes de interface de usuário virtuais que poderiam ser reutilizados em outros projetos.

Para validar o protótipo do módulo de lista foi desenvolvido um aplicativo com botões tradicionais do Unity, responsáveis por adicionar e remover objetos e alternar a iteração entre objetos ou materiais. Após incluir alguns protótipos do módulo de interface no lugar da interface tradicional, ficou evidente um problema no reconhecimento de quando um botão era pressionado e que muitas vezes o objeto da lista permanecia num lugar fixo da tela ao invés de acompanhar a posição do marcador. Questionou-se sobre o uso do *Extend Tracker* nas configurações do Vuforia ser o responsável pelas situações encontradas. Foi visto que o *Extend Tracker* por padrão vem desligado então testou-se com ele ligado porém foi observado que os componentes de tela virtuais não permaneciam na posição designada. Após testes foi visto que alterando a configuração *World Center Mode*, da câmera do Vuforia, de *FIRST\_TARGET* para *CAMERA* e com ajustes no método *Update* do script *ListAR* ambas as situações foram resolvidas.

Durante a implementação de alguns componentes do módulo de interface percebeu-se uma dificuldade ao organizá-los no editor e testar a visualização na versão do aplicativo. Os componentes que sofrem mais com esta situação são aqueles que possuem atributos de texto ou instanciam diversos botões virtuais, pois não é possível observar a disposição dos objetos na cena enquanto as propriedades são alteradas no editor. Os *Prefabs VirtualScroll* e *VirtualComboBox* possuem uma limitação percebida durante o desenvolvimento do aplicativo de testes. Acredita-se que por criarem diversos botões virtuais em tempo de execução, a implementação do Vuforia não reconhece esses botões após ser inicializada. Desta forma caso exista uma cena utilizando o Vuforia e esses *Prefabs* sejam utilizados numa cena subsequente, os botões criados por código embora visíveis não interagem quando são pressionados. Para concluir a aplicação de testes foi necessária uma solução paliativa, ao carregar a cena da primeira fase do jogo o script responsável por gerenciar os objetivos da fase reinicializava a instância do Vuforia. O Quadro 9 contém o código utilizado para contornar esta limitação.

Quadro 9 - Solução encontrada para utilizar componente com botões virtuais

```
101 private void OnEnable()  
102 {  
103     SceneManager.sceneLoaded += OnSceneLoaded;  
104 }  
105  
106 private void OnDisable()  
107 {  
108     SceneManager.sceneLoaded -= OnSceneLoaded;  
109 }  
110  
111 private void OnSceneLoaded(Scene arg0, LoadSceneMode arg1)  
112 {  
113     if (arg0.name == TestUserConfigManager.TEST_1_SCENE)  
114     {  
115         //For some reason the Virtual Scroll/Slider and ComboBox only work on the the scene where Vuforia is initialized  
116         Vuforia.VuforiaRuntime.Instance.Deinit();  
117         Vuforia.VuforiaRuntime.Instance.InitVuforia();  
118     }  
119 }
```

Fonte: elaborado pelo autor.

## 4.2 TESTES DE USABILIDADE

Para melhor avaliar a usabilidade da biblioteca foram propostas algumas atividades para dois bolsistas do curso de Ciência da Computação do professor Dalton Solano dos Reis, utilizando o pacote de *Assets* e seguindo as instruções junto com o formulário disponíveis no Bitbucket (DIEGOLI, 2019a). Os alunos realizaram as atividades com as versões 2018.2 e 2019.1 do Unity e relataram utilizar frequentemente a ferramenta. Conforme os resultados obtidos (DIEGOLI, 2019b), todas as atividades foram concluídas com o mínimo de ajuda externa. A usabilidade tanto do módulo de lista quanto de interface foram consideradas medianas, porém concordam na facilidade de usar os *Assets* e que podem ser utilizados no desenvolvimento de outros aplicativos. A maior dificuldade reportada foi na utilização do Vuforia por ser a primeira vez deles criando uma cena com esta tecnologia. Também foi sugerido organizar os *Assets* em pastas separadas para não se misturarem com os outros componentes existentes. Atendendo a sugestão recebida o *Asset Package* que antes continha todos os componentes *ListAR*, em pastas conflitantes com outros projetos, foi dividido em dois pacotes. O primeiro possui as informações dos marcadores, enquanto o segundo contém os componentes *ListAR* dispostos em uma pasta específica para evitar conflitos com novos projetos.

De forma semelhante foi realizado um teste de usabilidade nos dias 14 e 16 de maio de 2019 com alunos do ensino fundamental e médio de duas instituições diferentes. Em pares, os alunos recebiam uma breve explicação sobre o aplicativo, com os tablets realizavam os objetivos do jogo e ao final respondiam o questionário disponibilizado (DIEGOLI, 2019c). No primeiro dia de testes os alunos do ensino médio, de uma instituição de ensino pública da cidade de Balneário Camboriú, apontaram que nunca haviam experimentado um aplicativo com realidade aumentada. Notou-se por parte deles certa dificuldade ao utilizar a interface tangível e grande interesse na tecnologia ao realizarem as fases do jogo. No segundo dia de testes os alunos do ensino fundamental, de uma instituição de ensino privada do município de Ascurra, responderam que alguns já utilizaram aplicações de realidade aumentada, estes se mostraram em geral mais adaptados com a realidade aumentada e não apresentaram muitas dificuldades no uso da interface tangível. Fotos dos dias das atividades são mostradas no Apêndice B.

Para realizar os testes foi utilizado um questionário eletrônico. A primeira seção deste questionário era sobre o perfil de usuário, para saber principalmente o grau de familiaridade dos alunos com a realidade aumentada. A Tabela 1 apresenta as respostas obtidas no perfil de usuário, somando os alunos das duas instituições mencionadas e um aluno bolsista, assim totalizando 20 respostas.

Como é possível observar na Tabela 1 a faixa etária da maioria dos envolvidos permaneceu entre 12 e 18 anos, sendo um pouco mais da metade o público feminino. A pequena variância na escolaridade se deve ao aluno bolsista do curso de Ciência da Computação mencionado. Quase todos afirmaram utilizar frequentemente dispositivos móveis, no entanto a maior parte apontou não ter conhecimento sobre realidade aumentada.

Tabela 1 - Perfil dos usuários

Idade?	10% 12 anos 20% 13 anos 10% 14 anos 10% 16 anos 20% 17 anos 20% 18 anos 5% 19 anos 5% 20 anos
Sexo?	55% feminino 45% masculino
Nível de escolaridade?	45% ensino fundamental incompleto 45% ensino médio incompleto 5% ensino médio completo 5% ensino superior incompleto
Utiliza dispositivos móveis com qual frequência?	85% frequentemente 10% às vezes 5% nunca utilizou
Já utilizou aplicações com Realidade Aumentada?	55% não 45% sim

Fonte: elaborado pelo autor.

O questionário prosseguia com um passo a passo para completar o jogo, esta seção abordava informações sobre como configurar o menu para o modo de jogo desejado, definir se o jogador era destro ou canhoto e como avançar em cada fase. Nesta etapa, conforme os resultados (DIEGOLI, 2019d), ao jogar no modo Vuforia Plus apenas uma pessoa não conseguiu concluir a segunda fase e três pessoas sinalizaram que não conseguiram utilizar a navegação através do marcador de cubo. A parte referente ao modo Vuforia Tradicional foi modificada depois dos testes com o aluno bolsista, por isso algumas perguntas possuem 19 respostas ao invés do total de 20. Na questão sobre selecionar o modo Tradicional no menu das 19 respostas 21,1% marcaram que não jogaram este modo, das 20 respostas na pergunta sobre a primeira fase 25% não conseguiu concluir os objetivos, enquanto na segunda fase das 19 respostas 73,7% afirmaram que terminaram todos os objetivos jogando no modo Tradicional.

Para finalizar o questionário a seção de avaliação da usabilidade perguntava aos jogadores quantas tarefas foram concluídas e como foi a interação com cada componente desenvolvido, detalhando nas questões onde os componentes foram utilizados no aplicativo. A Tabela 2 apresenta os resultados obtidos da avaliação da usabilidade, incluindo a resposta enviada pelo aluno bolsista.

Tabela 2 - Respostas da avaliação da usabilidade

Você conseguiu concluir todos os objetivos com facilidade?	40% todos 30% a maior parte 20% metade 5% a menor parte 5% nenhum
Quantas tarefas você conclui sem nenhum auxílio externo?	25% todas 15% a maior parte 20% metade 20% a menor parte 20% nenhuma
Qual modo você gostou mais de jogar?	70% Vuforia Plus 30% Vuforia Tradicional
Como você classifica a usabilidade dos Botões Virtuais?	60% excelente 25% boa 5% regular 5% ruim 5% péssima
Como você classifica a usabilidade do Slider Virtual?	55% excelente 30% boa 10% regular 5% ruim
Como você classifica a usabilidade da navegação pelo Cubo?	55% excelente 25% boa 15% regular 5% ruim
Como você classifica a experiência geral de usar os componentes do ListAR?	70% excelente 25% boa 5% regular

Fonte: elaborado pelo autor.

Conforme a Tabela 2 mostra, a avaliação de usabilidade geral dos componentes foi positiva, contudo a navegação com o marcador de Cubo foi a que recebeu as piores notas comparando com os botões e o *slider*. Enquanto alguns alunos enfrentavam dificuldades para concluir as atividades outros precisaram apenas de uma breve explicação de como interagir com a interface e como avançar na fase, por isso houve respostas variadas com relação à pergunta sobre auxílio externo. A maioria preferiu jogar no modo com interface tangível provavelmente por ser o primeiro contato de muitos deles com realidade aumentada. Já alguns, no entanto, favoreceram o modo com interface tradicional possivelmente por não se adaptarem muito com a interface alternativa.

## 5 CONCLUSÕES

Diante dos resultados apresentados conclui-se que a biblioteca ListAR atingiu os objetivos pretendidos, tanto os testes com os desenvolvedores quanto os testes com usuários finais obtiveram retornos positivos. As sugestões feitas pelos alunos bolsistas do curso de Ciência da Computação que realizaram as atividades propostas apontam que a biblioteca pode ser utilizada, mesmo que ainda existam algumas melhorias a se fazer para atingir os requisitos necessários para publicação na loja de *Assets* do Unity. O aplicativo desenvolvido também se mostrou relevante, pois além de exaltar os efeitos positivos da tecnologia bem aplicada na educação mostrou também que os componentes da interface tangível são equiparáveis aos da interface tradicional, proporcionando ainda uma experiência de usuário diferenciada.

Embora os resultados gerais sejam positivos e os objetivos tenham sido alcançados, existem diversos aspectos ainda a melhorar. A interação com o marcador de Cubo recebeu boas avaliações mas no entanto foi insatisfatória, pois durante os testes os alunos eram livres para optarem como navegar pelas imagens da lista na segunda fase do jogo, e percebeu-se que apenas uma dupla preferiu utilizar o cubo ao invés dos botões. A proposta do Cubo de ser uma alternativa para navegar rapidamente pelos itens da lista não foi alcançada totalmente devida sua instabilidade ao ser acionado pelo *Box Collider* da área especificada. Por fim, muitos aspectos internos da biblioteca, como a relação entre os scripts e os componentes, e a configuração deles na cena do aplicativo não estão muito sugestivos, podendo causar certa dificuldade ou confusão aos desenvolvedores que pretendem utilizar alguns componentes específicos da biblioteca. As possíveis extensões averiguadas durante o desenvolvimento deste projeto são:

- reformular a interação do marcador de cubo a fim de estabilizar sua ativação e melhorar sua usabilidade;
- investigar uma possível solução para evitar a reinicialização do Vuforia ou abstrair-la nos componentes

- afetados;
- c) criação de scripts de editor na tentativa de melhorar a experiência do desenvolvedor ao configurar os componentes na cena do aplicativo;
- d) inclusão de novos componentes de tela para o módulo de interface como, por exemplo um selecionador de data e hora;
- e) aprimoramento dos componentes de tela existentes no módulo de interface, por exemplo ao manter pressionado um botão repetir a ação vinculada a cada fração de tempo;
- f) adição de novos tipos de interação para o módulo de lista, por exemplo permitir visualizar diversas animações diferentes para um mesmo item.

## REFERÊNCIAS

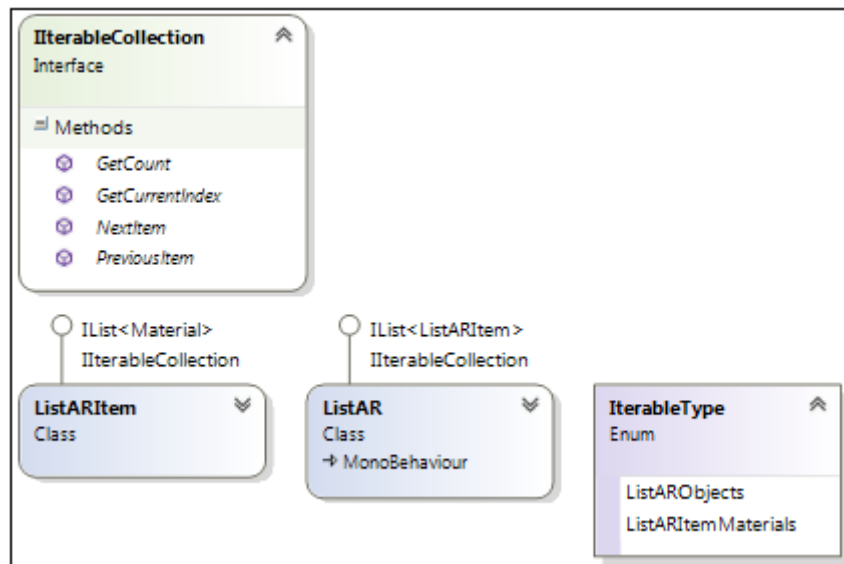
- DIEGOLI, Rafael. **ListAR**. Santa Catarina, 2019. 2019a. Disponível em: <[https://bitbucket.org/Rafael\\_Diegoli/tcc-listar-bkp/src/master/RafaelDiegoli\\_Questiona%CC%81rioDev.pdf](https://bitbucket.org/Rafael_Diegoli/tcc-listar-bkp/src/master/RafaelDiegoli_Questiona%CC%81rioDev.pdf)>. Acesso em: 08 junho 2019.
- DIEGOLI, Rafael. **ListAR**. Santa Catarina, 2019. 2019b. Disponível em: <[https://bitbucket.org/Rafael\\_Diegoli/tcc-listar-bkp/src/master/RafaelDiegoli\\_Questiona%CC%81rioDevRespostas.pdf](https://bitbucket.org/Rafael_Diegoli/tcc-listar-bkp/src/master/RafaelDiegoli_Questiona%CC%81rioDevRespostas.pdf)>. Acesso em: 08 junho 2019.
- DIEGOLI, Rafael. **ListAR**. Santa Catarina, 2019. 2019c. Disponível em: <[https://bitbucket.org/Rafael\\_Diegoli/tcc-listar-bkp/src/master/RafaelDiegoli\\_Questiona%CC%81rioDev.pdf](https://bitbucket.org/Rafael_Diegoli/tcc-listar-bkp/src/master/RafaelDiegoli_Questiona%CC%81rioDev.pdf)>. Acesso em: 08 junho 2019.
- DIEGOLI, Rafael. **ListAR**. Santa Catarina, 2019. 2019d. Disponível em: <[https://bitbucket.org/Rafael\\_Diegoli/tcc-listar-bkp/src/master/RafaelDiegoli\\_Questiona%CC%81rioRespostas.pdf](https://bitbucket.org/Rafael_Diegoli/tcc-listar-bkp/src/master/RafaelDiegoli_Questiona%CC%81rioRespostas.pdf)>. Acesso em: 08 junho 2019.
- REITER, Ricardo F. **AnimAR: Desenvolvimento de uma ferramenta para criação de animações com Realidade Aumentada e Interface Tangível**. 2018. 80 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- RODRIGUES, Fábio; SATO, Fernando; BOTEAGA, Leonardo. **Integrando Interface Tangível com Técnicas de Realidade Aumentada para Ampliar a Experiência Interativa do Usuário**. 2012. 6 f. Artigo - Centro de Interação Humano-Computador, Centro Universitário Eurípides de Marília, Marília.
- SCHMITZ, Evandro M. **Desenvolvimento de uma ferramenta para auxiliar no Ensino do Sistema Solar utilizando Realidade Aumentada**. 2017. 94 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- SILVA, Antônio M. da. **VISEDU: Interface de Usuário Tangível utilizando Realidade Aumentada e Unity**. 2016. 75 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.
- UNITY. **Unity – Manual: Asset Workflow**. [S.l.], [2019?]. 2019a. Disponível em: <<https://docs.unity3d.com/Manual/AssetWorkflow.html>>. Acesso em: 01 junho 2019.
- UNITY. **Unity – Manual: Common types of Assets**. [S.l.], [2019?]. 2019b. Disponível em: <<https://docs.unity3d.com/Manual/AssetTypes.html>>. Acesso em: 01 junho 2019.
- UNITY. **Unity – Manual: Asset packages**. [S.l.], [2019?]. 2019c. Disponível em: <<https://docs.unity3d.com/Manual/AssetPackages.html>>. Acesso em: 01 junho 2019.
- UNITY. **Unity – Manual: Prefabs**. [S.l.], [2019?]. 2019d. Disponível em: <<https://docs.unity3d.com/Manual/Prefabs.html>>. Acesso em: 01 junho 2019.
- UNITY. **Unity – Manual: UI**. [S.l.], [2019?]. 2019e. Disponível em: <<https://docs.unity3d.com/Manual/UIToolkits.html>>. Acesso em: 01 junho 2019.
- UNITY. **Unity – Manual: Visual Components**. [S.l.], [2019?]. 2019f. Disponível em: <<https://docs.unity3d.com/Manual/UIVisualComponents.html>>. Acesso em: 01 junho 2019.
- UNITY. **Unity – Manual: Interaction Components**. [S.l.], [2019?]. 2019g. Disponível em: <<https://docs.unity3d.com/Manual/UIInteractionComponents.html>>. Acesso em: 01 junho 2019.
- UNITY. **Unity – Manual: Canvas**. [S.l.], [2019?]. 2019h. Disponível em: <<https://docs.unity3d.com/Manual/UICanvas.html>>. Acesso em: 01 junho 2019.
- UNITY. **Unity – Manual: Basic Layout**. [S.l.], [2019?]. 2019i. Disponível em: <<https://docs.unity3d.com/Manual/UIBasicLayout.html>>. Acesso em: 01 junho 2019.
- ZANDOMENEGUI, Ana L. A. O. **Conceitos e Práticas em Ambiente Virtual de Aprendizagem Inclusivo**. São Paulo: Pimenta Cultural, 2014.



## APÊNDICE A – DIAGRAMAS DE ESPECIFICAÇÃO

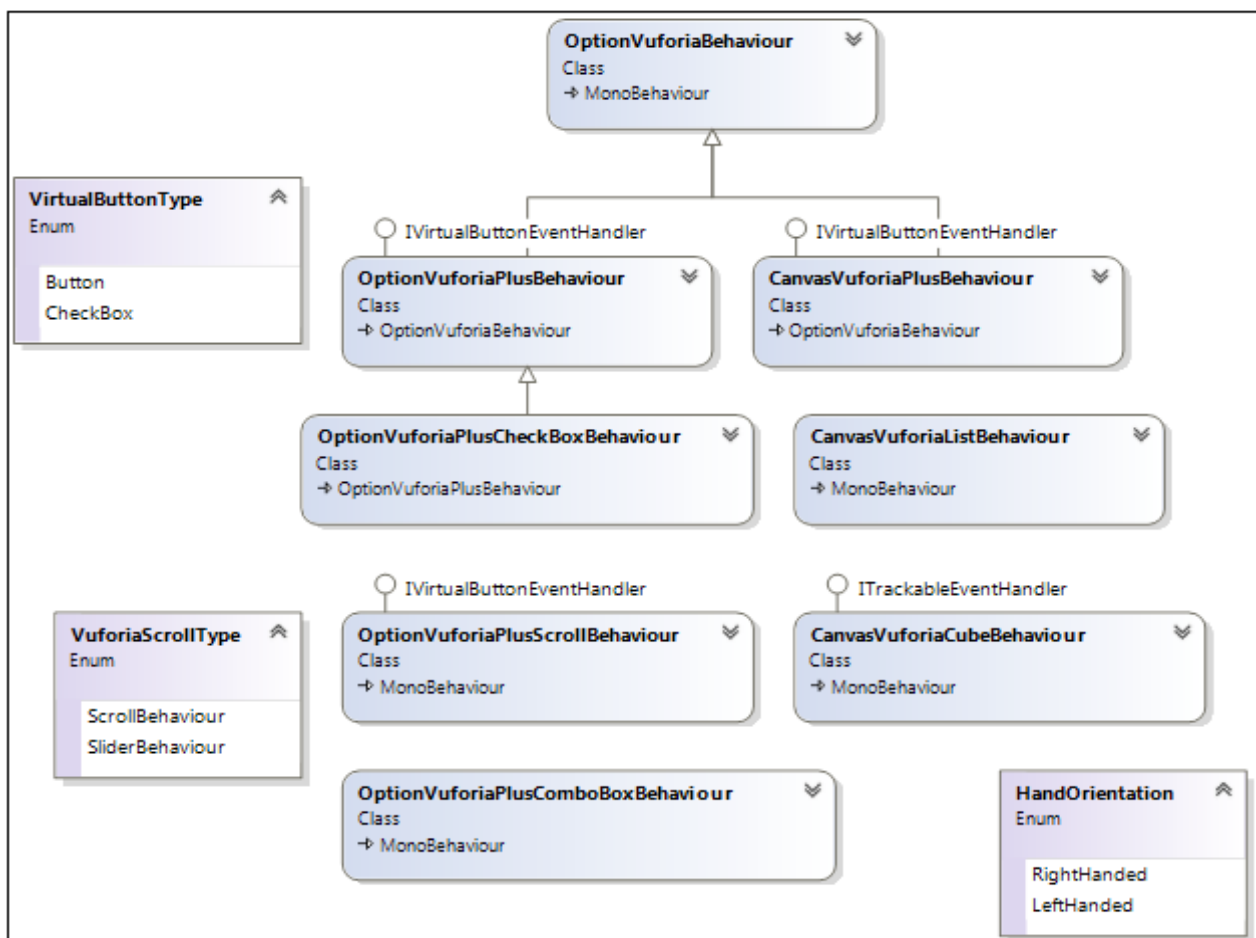
Neste apêndice são apresentados os diagramas de classes contendo os scripts e componentes mais relevantes para a biblioteca. O diagrama referente ao módulo de lista está representado na Figura 15 enquanto a Figura 16 exibe o diagrama de classes do módulo de interface.

Figura 15 - Diagrama de classes do módulo de lista



Fonte: elaborado pelo autor.

Figura 16 - Diagrama de classes do módulo de interface

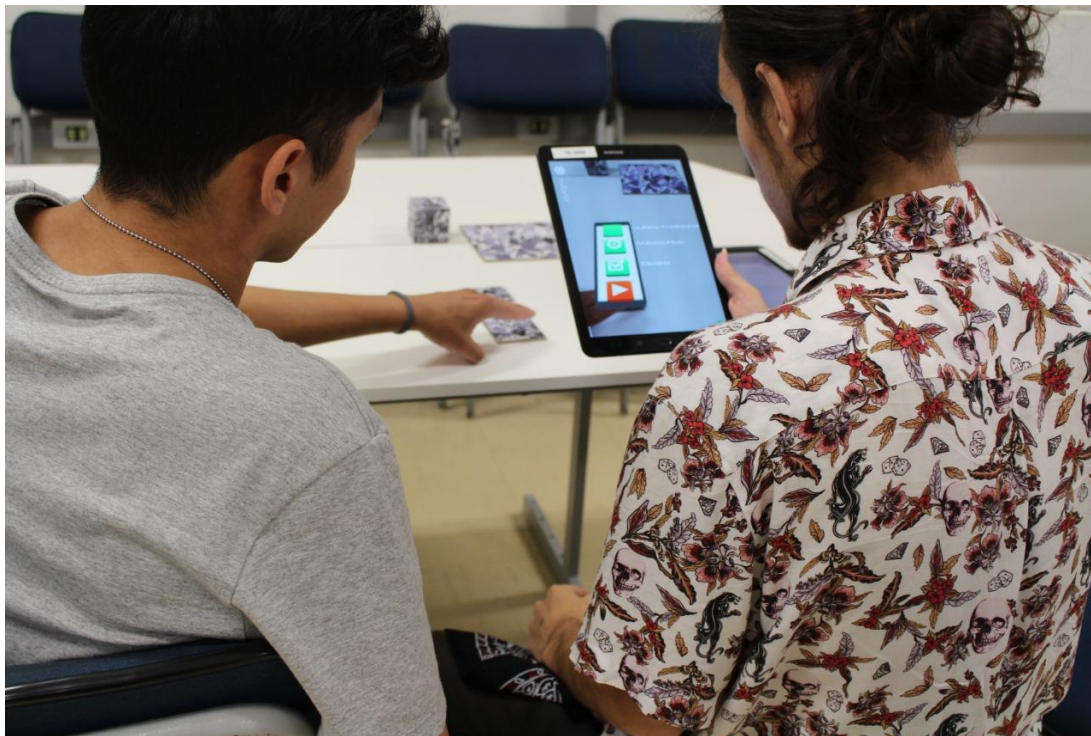


Fonte: elaborado pelo autor.

## APÊNDICE B – TESTE DO APLICATIVO DOS DIAS 14 E 16 DE MAIO DE 2019

Neste apêndice são apresentadas algumas fotos dos testes realizados nos dias 14 e 16 de Maio de 2019 com alunos de Balneário Camboriú e Ascurra no laboratório LIFE na Universidade Regional de Blumenau. A Figura 17 mostra dois alunos no menu principal apertando o botão para iniciar o jogo, a Figura 18 apresenta outros alunos realizando a primeira fase do jogo, por fim na Figura 19 é possível observar a visualização da segunda fase inclusive com o marcador de Cubo.

Figura 17 - Alunos na tela do menu iniciando o jogo



Fonte: elaborado pelo autor.

Figura 18 - Alunos alternando as opções na primeira fase do jogo



Fonte: elaborado pelo autor.

Figura 19 - Alunos navegando pelas imagens na segunda fase do jogo



Fonte: elaborado pelo autor.