

# **VISEDU: JOGO DE REALIDADE AUMENTADA DE LETRAS COM CONTEÚDO DINÂMICO**

Aluna: Vivian de Lima Panzenhagen

Orientador: Prof. Dalton Solano dos Reis, M. Sc

# Roteiro

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

# Introdução

- Exploração da Realidade Aumentada (RA)
- RA no ensino pode propiciar um melhor aproveitamento dos conteúdos
- Desenvolvimento de um aplicativo que auxilia na alfabetização

# Objetivos

- Desenvolver um aplicativo de Realidade Aumentada de letras para dispositivos móveis com conteúdo dinâmico
- Utilizar marcadores contendo letras para o reconhecimento da palavra
- Possibilitar a utilização de imagens definidas pelo usuário
- Possibilitar inserir a imagem definida pelo usuário em diferentes níveis de dificuldade

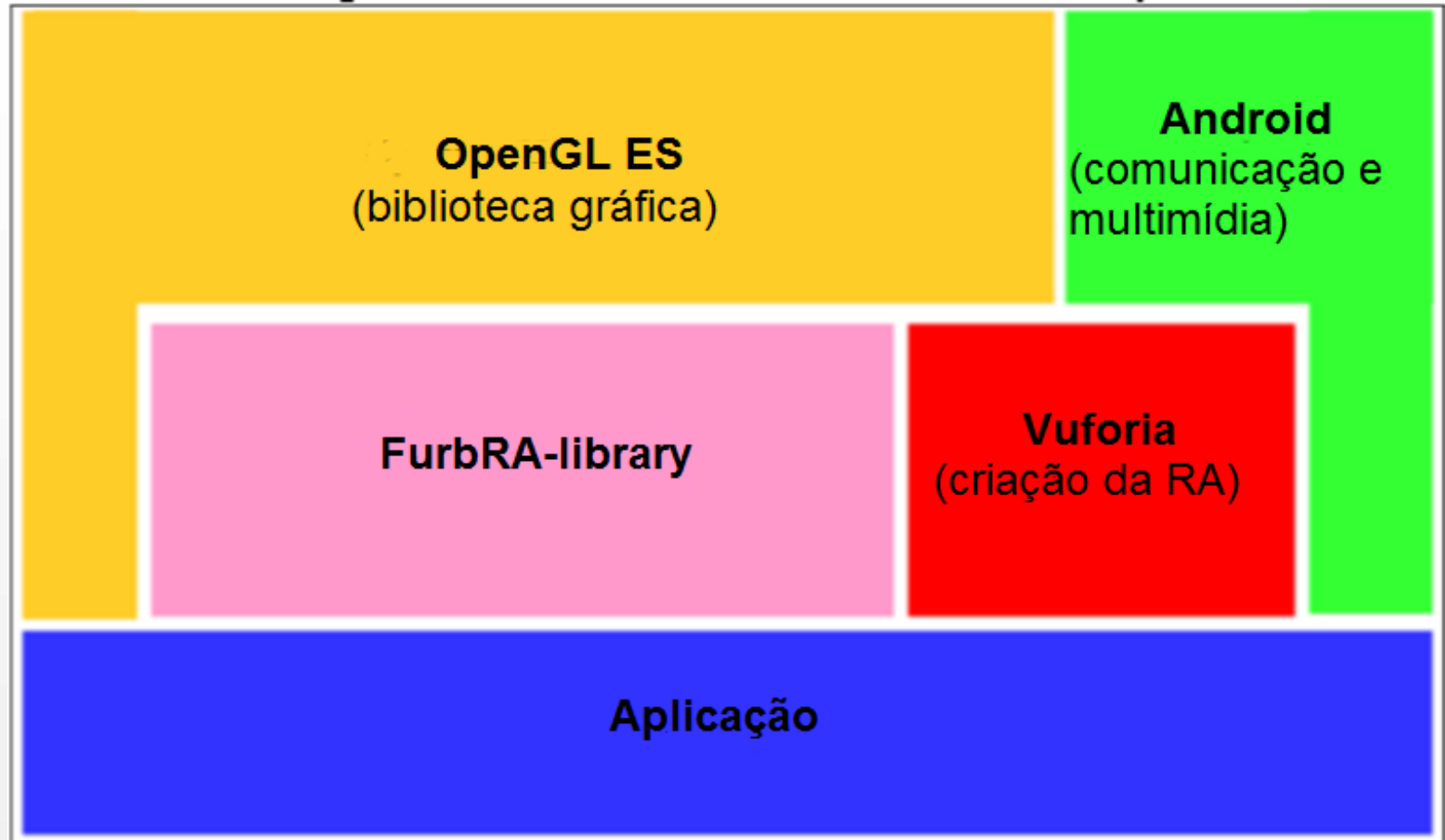
# Fundamentação Teórica

- Realidade Aumentada
- Biblioteca FurbRA-library de Lira (2015)
- Unity
- Vuforia

# Realidade Aumentada

- Propriedades para existência da RA:
  - Combinação do ambiente real com o virtual
  - Interatividade do usuário em tempo real com a aplicação

# Biblioteca FurbRA-library Lira (2015)



# Unity

- Plataforma para desenvolvimento de jogos e experiências interativas
- Multiplataforma
- Disponibilidade de recursos para o desenvolvimento



# Vuforia

- Biblioteca com recursos para implementação da RA
- Integração com o Unity

# Vuforia

- Vuforia
  - Marcadores
    - Pré-cadastrados (base de alvos)
      - Imagens, cilindros, cubos e objetos 3D
    - Cadastrados (disponíveis no próprio ambiente)
      - Frame ou **Palavras**

# Trabalhos Correlatos

## OsmoPlay(2015)



# Trabalhos Correlatos

## Forma Palavras(2015)



# Trabalhos Correlatos

## Monta Palavras(2015)

A word-building puzzle interface. On the left is a colorful sofa. In the center are two empty boxes: a blue one on the left and a pink one on the right. Below the blue box are the letters 'F' and 'A' in pink boxes. Below the pink box are the letters 'S' and 'O' in pink boxes. A question mark is in the bottom right corner.

# Requisitos

- **Requisitos funcionais do aplicativo:**
  - possuir uma tela inicial para seleção de opções
  - disponibilizar uma opção de jogo que contenha imagens em cada nível de dificuldade
  - disponibilizar uma forma de adicionar imagens dinamicamente ao aplicativo

# Requisitos

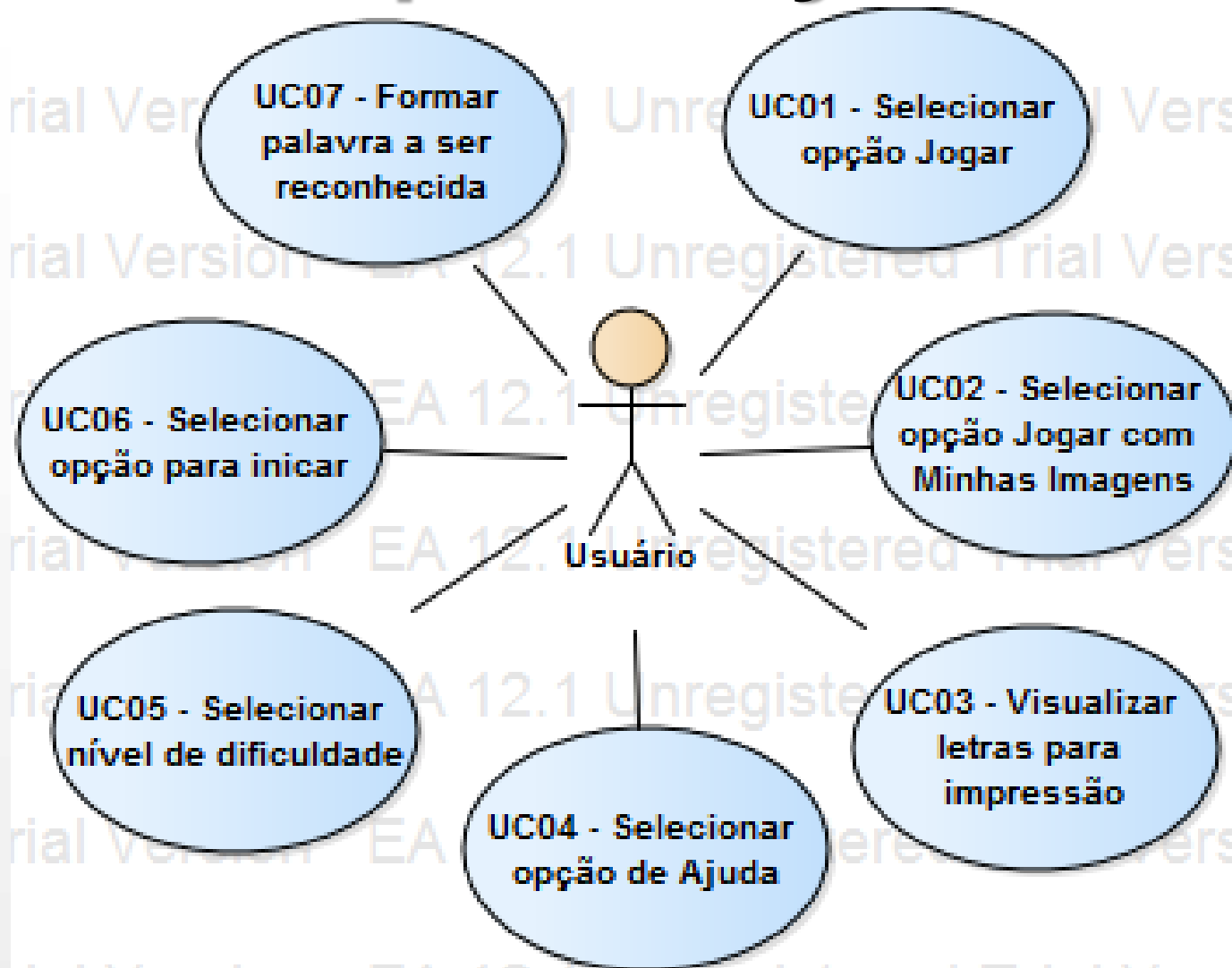
- **Requisitos funcionais do aplicativo:**
  - disponibilizar uma opção de jogo para poder visualizar e imprimir o marcador
  - disponibilizar uma opção de ajuda contendo informações que auxilia o usuário antes de iniciar o jogo
  - disponibilizar diferentes níveis de dificuldade
  - disponibilizar opção de parar o jogo durante o reconhecimento de texto

# Requisitos

- **Requisitos não funcionais do aplicativo**
  - ser disponibilizado para a plataforma Android
  - utilizar o Software Development Kit (SDK) Vuforia para a implementação dos recursos de Realidade Aumenta
  - utilizar a linguagem C# para definir os scripts do aplicativo
  - utilizar o motor de jogos Unity 3D para a implementação da cena gráfica 3D



# Especificação



# Especificação

«MonoBehaviour» BarraVida
+ pontos: float + preenchimento: Image + tempo: float
+ Start(): void + Update(): void

«MonoBehaviour» CarregarImagem
+ arraySprite: Sprite[] - nivel1: List<string> - nivel2: List<string> - nivel3: List<string> - sprites: List<string>
+ retornarImagensNivel(): void + Start(): void

«MonoBehaviour» CarregaProximaCena
+ carregaCena(string): void

«MonoBehaviour» SalvaPontuacao
+ Start(): void

«MonoBehaviour» ConfigCenaNivel
+ btnPlay: Button + nomeNivel: string + txtNomeNivel: Text
+ ajuda(): void + carregarNivel(): void + proximaDica(): void + sairAjuda(): void + seleccioneNivel(int): void + Start()

«MonoBehaviour» ScriptBotoes
+ imprimir(): void + jogarImagensPadrao(): void + jogarMinhasImagens(): void

«MonoBehaviour» OpcoesPause
+ sair(string): void + voltarJogo(): void

«MonoBehaviour» Iniciar Seq Sprites
+ iniciarSequenciaSprite(): void

Classes Text Recognition
--------------------------

# Implementação

- Importação do Vuforia no Unity
- Importação do SDK de reconhecimento de texto disponibilizado pelo Vuforia
- Configuração do arquivo de palavras adicionais

# Implementação

- Arquivo de palavras adicionais

AdditionalWords.lst

1	UTF-8
2	
3	QCAR
4	BALEIA
5	GIRAFÁ
6	CACHORRO
7	PORCO
8	RATO
9	CAVALO
10	COELHO
11	ELEFANTE
12	GATO
13	LEAO
14	PEIXE
15	TIGRE

Inspector

TextRecognition  Static

Tag Untagged Layer Default

Prefab Select Revert Apply

**Transform**

**Text Reco Behaviour (Script)**

The list of words the TextTracker can detect and track.  
The word list is loaded from a binary file and can be extended by a list of custom words.

Word List Vuforia-English-word

Additional Word File AdditionalWords

Additional Words:

Write one word per line. Open compound words can be specified using whitespaces.

The filter list allows to specify subset of words that will be detected and tracked.

Filter Mode WHITE\_LIST

Filter List File AdditionalWords

# Implementação

- Busca de imagens

```
27 public void Start(){
28
29     if (PlayerPrefs.GetString ("jogarMinhasImagens") == "sim") {
30         buscarImagens ();
31     } else if (PlayerPrefs.GetInt ("NivelEscolhido") == 1) {
32         arraySprite = Resources.LoadAll<Sprite> ("Sprites/facil");
33     } else if (PlayerPrefs.GetInt ("NivelEscolhido") == 2) {
34         arraySprite = Resources.LoadAll<Sprite> ("Sprites/medio");
35     } else if (PlayerPrefs.GetInt ("NivelEscolhido") == 3) {
36         arraySprite = Resources.LoadAll<Sprite> ("Sprites/dificil");
37     }
38
39     adicionaSpriteEscolhida ();
40     buscarImagensAleatorias ();
41 }
```

# Implementação

```
118 public void buscarImagens() {
119     //string myDir = "C:\\Users\\Vivian\\Documents";
120     string myDir = "/storage/emulated/0/Download/VISEDU";
121     DirectoryInfo dir = new DirectoryInfo (myDir);
122     nivel1 = new List<string>();
123     nivel2 = new List<string>();
124     nivel3 = new List<string>();
125
126     List<System.IO.FileInfo> info = new List<System.IO.FileInfo> ();
127     info.AddRange (dir.GetFiles ("*.png", System.IO.SearchOption.TopDirectoryOnly));
128     info.AddRange (dir.GetFiles ("*.jpg", System.IO.SearchOption.TopDirectoryOnly));
129     if (info != null) {
130         string[] paths = new string[info.Count];
131         for (int i = 0; i < info.Count; i++) {
132             paths [i] = info [i].FullName.ToString ();
133             string nomeImagem = Path.GetFileNameWithoutExtension (paths [i]);
134
135             if (nomeImagem.Contains ("F_")) {
136                 nivel1.Add(paths[i]);
137             } else if (nomeImagem.Contains ("M_")) {
138                 nivel2.Add(paths[i]);
139             } else if (nomeImagem.Contains ("D_")) {
140                 nivel3.Add(paths[i]);
141             }
142         }
143     }
```

# Implementação

```
142  if (PlayerPrefs.GetInt ("NivelEscolhido") == 1) {
143      sprites = nivel1;
144  } else if (PlayerPrefs.GetInt ("NivelEscolhido") == 2) {
145      sprites = nivel2;
146  } else if (PlayerPrefs.GetInt ("NivelEscolhido") == 3) {
147      sprites = nivel3;
148  }
149
150  arraySprite = new Sprite[sprites.Count];
151
152  for (int i = 0; i < sprites.Count; i++) {
153      byte[] data = File.ReadAllBytes (sprites.ElementAt(i));
154      Texture2D texture = new Texture2D (270, 186, TextureFormat.ARGB32, false);
155      texture.LoadImage (data);
156      texture.name = Path.GetFileNameWithoutExtension (sprites.ElementAt(i));
157      Sprite spt = Sprite.Create (texture, new Rect
158          (0, 0, texture.width, texture.height), Vector2.zero);
159      spt.name = texture.name.Substring(2);
160      Debug.Log ("Sprite Texture = " + spt.name);
161      arraySprite[i] = spt;
162  }
163  }
```

# Implementação

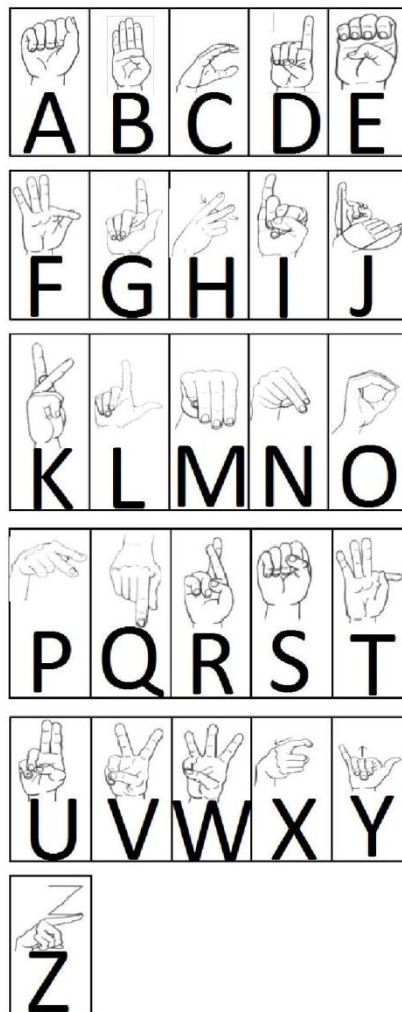
- Identificação da palavra

```
160     public void OnWordDetected(WordResult wordResult) {
161
162         var word = wordResult.Word;
163         var img = GameObject.Find("Imagem").GetComponent<UiImage>();
164
165         if (ContainsWord(word))
166             Debug.LogError("Word was already detected before!");
167
168         if (wordResult.Word.StringValue == img.sprite.name.ToUpper()) {
169             Application.LoadLevel("notaFinal");
170         }
171
172         AddWord(wordResult);
173
174     }
```



# Operacionalidade da Implementação

- Letras utilizadas



# Operacionalidade da Implementação

- Tela inicial



# Operacionalidade da Implementação

- Tela de reconhecimento de texto



# Resultados e Discussões

- Realizado experimento com usuários no laboratório LIFE da FURB
- Turma de Sistemas Multimídia dos cursos de Ciências da Computação e Pedagogia

# Experimento

- Perfis de usuários envolvidos no teste

<b>Sexo</b>	Masculino	8,3%
	Feminino	91,7%
<b>Idade</b>	Entre 16 e 20 anos	8,3%
	Entre 21 e 25 anos	91,7%
<b>Nível de escolaridade</b>	Ensino médio completo	8,3%
	Ensino superior incompleto	91,7%
<b>Frequência que utiliza computador</b>	Frequentemente	100%
<b>Frequência que utiliza dispositivos móveis</b>	Às vezes	8,3%
	Frequentemente	91,7%
<b>Grau de familiaridade com Realidade Aumentada</b>	Conheço, mas nunca utilizei	50%
	Já utilizei	50%

# Experimento

- Respostas do questionário de avaliação

<b>Conseguiu selecionar a opção jogar</b>	Conseguiu	100%
<b>Conseguiu selecionar o nível de dificuldade</b>	Conseguiu	100%
<b>Conseguiu tocar no botão de jogar</b>	Conseguiu	100%
<b>Conseguiu formar a palavra de acordo com a imagem exibida</b>	Conseguiu	100%
<b>Conseguiu posicionar o dispositivo móvel de forma que a palavra seja exibida na área de reconhecimento</b>	Conseguiu	100%
<b>Na tela nota final escolher a opção para continuar</b>	Conseguiu	100%
<b>Repetir o processo para as três imagens do nível</b>	Conseguiu	100%

# Experimento

- Respostas do questionário de usabilidade

<b>Das atividades solicitadas, quantas conseguiu executar</b>	A maior parte	16,7%
	Todas	83,3%
<b>Achou o aplicativo intuitivo e fácil de usar</b>	Sim	100%
<b>Avaliação do aplicativo</b>	Bom	25%
	Muito bom	75%
<b>Acredita que jogos auxiliam no aprendizado</b>	Sim	100%
<b>Acredita que jogos juntamente com dispositivos móveis torna o aprendizado mais atrativo</b>	Sim	100%
<b>Sentiu atraído a aprender a língua de sinais</b>	Não	8,3%
	Pouco atraído	16,7%
	Sim	75%

# Comparação dos trabalhos correlatos

Características/ Trabalhos	OsmoPlay (2014)	Games (2015)	Kids (2015)	Lima (2016)
É um jogo que auxilia na alfabetização	X	X	X	X
Permite a utilização de marcadores	X			X
É um jogo de Realidade Aumentada	X			X
É um trabalho acadêmico				X
Utiliza Vuforia para identificação de palavras				X
Plataforma Suportada	iOS	Android	Android	Android



# Conclusões e Sugestões

- Ferramentas adequadas (Unity com Vuforia)
- Objetivos alcançados
- Facilidade na geração do aplicativo para outras plataformas

# Extensões

- Permitir a utilização de qualquer dispositivo móvel para execução do aplicativo
- Cadastro de usuário, sendo eles professor e aluno
- No perfil do usuário aluno, permitir acesso a evolução dentro do jogo
- Utilização de um maior número de palavras no arquivo de palavras adicionais

# Apresentação prática