

ESTUDO DE TECNOLOGIAS ASSISTIVAS PARA DEFICIENTES VISUAIS APLICADAS AO JOGO FURBOT

Caroline Batistel, Dalton Solano Reis – Orientador

Curso de Bacharel em Ciência da Computação
Departamento de Sistemas e Computação
Universidade Regional de Blumenau (FURB) – Blumenau, SC – Brasil

cbatistel@furb.br, dalton@furb.br

Resumo: Este artigo apresenta o desenvolvimento de um protótipo para o jogo Furbot que visa criar acessibilidade para pessoas cegas ou com baixo grau de visão. O protótipo desenvolvido utiliza o RT-Voice Pro, uma ferramenta text-to-speech, para situar os usuários, e uma série de alterações, como limitar o espaço visível em tela, e novas funções para nivelar o grau de dificuldade entre jogadores com visão normal e com limites de uso. O desenvolvimento foi realizado em Unity, utilizando como base a versão 3.0 do jogo Furbot. Foram realizados testes com cinco usuários, sendo um deles com baixa visão e entre estes usuários alguns já haviam tido contato com o Furbot em versões anteriores. No geral os usuários testados aprovaram o protótipo desenvolvido e indicaram as novas funções como úteis.

Palavras-chave: Deficientes visuais. Acessibilidade. Text-to-speech. RT-Voice Pro. Unity.

1 INTRODUÇÃO

A inclusão social tem assumido grande importância nos dias de hoje, e com a popularização de ferramentas computacionais e da Internet, também é necessário focar na inclusão digital. Segundo dados do Censo demográfico de 2010, existem 6,5 milhões de pessoas no Brasil que possuem deficiências visuais, sendo 582 mil cegas e 6 milhões com baixa visão (IBGE, 2011).

Muitos métodos foram criados para auxiliar pessoas não-videntes (que não enxergam) em seu cotidiano, entre eles se encontram as tecnologias assistivas. Tais tecnologias, podem ser descritas como uma série de equipamentos, estratégias, práticas e serviços concebidos para minimizar problemas funcionais enfrentados por pessoas portadoras de necessidades especiais (COOK; POLGAR, 2014).

Grande parte dos dados passados diariamente às pessoas, seja qual for o ambiente, se dá através de imagens e apelos visuais, o que cria barreiras para pessoas cegas ou de baixa visão (NUNES; MACHADO; VAZIN, 2011). Para proporcionar a essas pessoas o devido acesso aos conteúdos visuais, foi desenvolvida uma tecnologia assistiva chamada audiodescrição. Sendo um recurso amplamente utilizado, a audiodescrição tenta traduzir em palavras todo o conteúdo que pessoas não-videntes não tem acesso por sua condição. Em meios digitais a audiodescrição se adaptou a leitores de tela, que permitem a pessoas não-videntes a navegação na internet, e a utilização com autonomia de celulares e computadores. Além da audiodescrição, existem diversas outras tecnologias assistivas direcionadas a pessoas não-videntes, como os aplicativos que utilizam reconhecimento de imagem, o sistema de escrita em braille, entre outros.

Destacando-se os aspectos de inclusão social e digital, chega-se à plataforma Furbot, criada na Universidade Regional de Blumenau (FURB) em um projeto que, segundo Mattos *et. al.* (2019a, p. 1), “busca promover a inclusão digital cidadã por meio de oficinas de programação que permitam o desenvolvimento de aptidões em pensamento computacional [...]”.

Tendo em vista este cenário, este trabalho se propôs a estudar como tecnologias assistivas para pessoas cegas e de baixa visão podem ser aplicadas ao jogo Furbot, e criar um protótipo do jogo com acessibilidade para tais pessoas. O objetivo deste trabalho é construir uma variação do jogo com suporte à recursos de acessibilidade para pessoas cegas ou com baixo nível de visão. Os objetivos específicos são:

- a) disponibilizar um módulo de audiodescrição integrado ao Furbot, que permita a pessoa se localizar, sem interferir na forma de encontrar a solução das atividades apresentadas;
- b) criar estratégias para auxiliar os usuários a se localizarem e atravessarem uma fase de teste.

2 FUNDAMENTAÇÃO TEÓRICA

Esta seção traz informações sobre os aspectos da fundamentação teórica utilizados para a construção do projeto. São introduzidos conceitos sobre acessibilidade digital, tecnologias assistivas e audiodescrição e sobre o jogo Furbot.

2.1 ACESSIBILIDADE DIGITAL

O conceito de acessibilidade nasceu ligado a assuntos físicos, como facilidade de acesso a locais, e começou a ser difundido no mundo por volta de 1981, declarado pela Organização das Nações Unidas (ONU) como o “Ano Internacional dos Portadores de Deficiência”. A acessibilidade está associada a permitir acesso a todo e qualquer espaço, físico ou não, deixando viável a entrada de diferentes tipos de pessoas, com ou sem necessidades especiais aos locais que elas frequentam, garantindo-lhes qualidade de vida (Lei nº. 10.098/2000) (BRASIL, 2000).

Para Granollers (2004), acessibilidade digital significa propiciar a flexibilidade para adaptações decorrentes das necessidades de cada usuário segundo suas preferências e/ou limitações. Segundo Dias (2007), a acessibilidade digital é a propriedade de um produto que permite atender pessoas, sendo compatível com tecnologias assistivas. Portanto, um software é tido como acessível quando qualquer pessoa, portadora de deficiência ou não, consegue executar as mesmas funções, e alcança os mesmos resultados ao utilizá-lo.

Passerino *et. al.* (2007) afirmam que a acessibilidade digital só pode ser proporcionada mediante uma combinação entre hardware e software, oferecendo os mecanismos físicos para superar barreiras de percepção e o acesso a funções e informações. Os autores acreditam que é possível promover a inclusão digital através de três grandes áreas, sendo elas as tecnologias assistivas, incluir portadores de necessidades especiais na concepção dos softwares e dar mais acesso à internet para estas pessoas, flexibilizando a informação e as interações.

2.2 TECNOLOGIAS ASSISTIVAS – AUDIODESCRIÇÃO

O termo tecnologia assistiva foi proposto por Sasaki (1997), como “coisa que assiste ou coisa que ajuda, que auxilia”. Em 16 de novembro de 2006 foi instituído no Brasil, pela Portaria nº 142 (BRASIL, 2006), o Comitê de Ajudas Técnicas (CAT) o qual apresentou o conceito de Tecnologia Assistiva como uma área do conhecimento que engloba produtos, recursos, metodologias, estratégias, práticas e serviços. A apresentação de tal conceito visa “promover a autonomia e independência de pessoas portadoras de deficiência, gerando assim qualidade de vida e inclusão social” (BRASIL, 2009).

Para Cook e Polgar (2014), as tecnologias assistivas podem ser expostas como uma série de equipamentos, estratégias, práticas e serviços concebidos para tornarem mínimos os problemas funcionais enfrentados por pessoas portadoras de necessidades especiais. Nunes *et. al.* (2011) explicam que grande parte dos dados passados diariamente às pessoas, seja qual for o ambiente, se dá através de imagens e apelos visuais, o que cria barreiras para pessoas cegas ou de baixa visão, por este motivo tecnologias assistivas para cegos são muito necessárias para disponibilizar acesso à informação.

Neste ponto, a audiodescrição segundo Franco *et. al.* (2010), permite que pessoas cegas tenham o acesso aos conteúdos visuais em qualquer tipo de mídia, por se tratar de uma tradução em palavras de toda informação relevante para o entendimento de uma mensagem apresentada de forma visual. Diferindo de outras tecnologias assistivas, a audiodescrição não é um meio que possa ser obtido isoladamente, sendo utilizado apenas quando o usuário desejar, tratando-se de um recurso disponibilizado junto com os produtos.

Em meios digitais a audiodescrição geralmente é percebida através de leitores de tela, ferramentas também conhecidas como *text-to-speech* (TTS) que traduzem texto em fala. Existem diversas interfaces de programação (APIs) que possibilitam criação de conteúdo digital integrado com TTS. Um exemplo dessas APIs é o RTVoice Pro, desenvolvido pela Crosstales (2020) que utiliza vozes TTS já integradas no sistema ou plataforma para pronunciar qualquer texto podendo ser utilizado em tempo de execução.

Segundo a Crosstales (2020), o RTVoice conta com compatibilidade em várias plataformas, integração com ferramentas de TTS online, configurações de entonação, velocidade e volume de fala fornecendo ao usuário uma maior flexibilidade para atender suas necessidades e preferências. A ferramenta também permite alterações como sussurros, implementação de sentimento ou sensação, como alegria, tristeza, tédio, entre outras.

2.3 JOGOS EDUCACIONAIS - FURBOT

Para Savi e Ulbricht (2008), jogos desenvolvidos para fins educacionais podem ser denominados como jogos educativos ou educacionais, jogos de aprendizagem ou ainda jogos sérios. Segundo Gros (2003), jogos digitais são uma das principais formas de acesso ao mundo da tecnologia para crianças e jovens, afinal geralmente o primeiro contato com equipamentos eletrônicos é por meio de um vídeo game.

Prieto *et. al.* (2005) destacam que para serem utilizados como métodos educacionais, qualquer software seja ele jogo ou não, deve conter objetivos pedagógicos e a sua utilização deve estar num contexto e situação de ensino baseados em uma metodologia que oriente o processo, utilizando-se da interação, motivação e descoberta para facilitar o aprendizado de um determinado conteúdo.

Entre os diversos jogos educacionais existentes, Mattos *et. al.* (2019a) descrevem o Furbot como um projeto que atua no desenvolvimento cognitivo infantil através de atividades de programação de computadores, utilizando jogos

de estratégia para facilitar o aprendizado da programação e desenvolver o raciocínio lógico, e a capacidade de resolução de problemas, colocando o pensamento computacional em ação.

A ferramenta apresenta um robô denominado Furbot como personagem principal da história o qual possui missões diferenciadas durante cada cena do jogo. O jogo possui um enredo para contextualizar o jogador. O enredo conta que a Terra foi invadida por alienígenas sendo que estes deixaram pistas no planeta. A missão do robô é salvar o planeta da invasão alienígena, coletando as pistas deixadas por eles pelo caminho. (MATTOS *et. al.*, 2019b, p. 1260)

Segundo Mattos *et. al.* (2019b), o Furbot em sua versão Unity consiste em um jogo programável que inicia a partir da história introdutória que tem como personagens a Professora Sam que cria o Furbot, um robô que os alunos precisam programar para salvar o mundo do ataque dos malvados “buggiens”.

Durante as fases, Mattos *et. al.* (2019b) explica que o jogador deve programar o robô para deixar sua posição inicial no mapa e atingir a última posição do caminho, onde há uma marca deixada por um alienígena. A drone S-223 é uma ajudante que fornece algumas dicas durante todo o jogo. Conforme apresentado por Mattos *et. al.* (2019c), o jogo possui um cenário (Figura 1) composto inicialmente por um robô (o Furbot), um drone (a S-223) e um caminho a ser percorrido. Em cada uma das fases, o jogador precisa programar o robô para percorrer o caminho, desviando de obstáculos (como árvores, arbustos e pedras) e capturar os objetivos do mapa (como tesouros, marcas deixadas pelos alienígenas, vidas, energias, entre outros).

Figura 1 - Fase inicial do jogo Furbot



Fonte: Mattos *et al.* (2019c).

No mapa apresentado na Figura 1, conforme Mattos *et. al.* (2019c), o robô deve sair de onde se encontra e atingir a última posição do caminho localizado na direita do mapa, no qual possui uma marca deixada por um alienígena. Ainda, o jogador deve cuidar para andar somente por cima do caminho, pois se andar pelo gramado ele perderá mais energia, podendo não conseguir completar a fase. A S-223 é a ajudante do Furbot e fornece dicas para ele durante todo o jogo.

No exemplo da Figura 1, tem-se no lado direito o editor de texto com um exemplo de código-fonte produzido pelo jogador. Na parte inferior, tem-se a S-223 que apresenta a mensagem de erro conforme o compilador identifica. Segundo Mattos *et al.* (2019b), são apresentados erros tanto léxicos quanto sintáticos identificados pelo compilador. Caso o código-fonte esteja correto, ao ser executado pela ferramenta o Furbot começa a se movimentar no cenário. Quando o Furbot atingir o objetivo final da fase, que neste caso é coletar a pista deixada pelo alienígena, a fase é encerrada, mostra-se a pontuação adquirida e o jogador é encaminhado para a próxima fase. Todas as fases são constituídas por cenários que contém objetivos a serem desviados e coletados e devem ser programados para que o robô realize a ação (MATTOS *et al.*; 2019c).

Além de sua versão desktop o jogo foi desenvolvido para outras plataformas, a Figura 2 apresenta uma visão da versão para dispositivos móveis onde é possível identificar o painel de comandos (a) que permite que a criança construa os comandos necessários para movimentar o Furbot através do cenário da fase (b) e o painel (c) onde o “código fonte” é apresentado e a partir do qual é possível acompanhar o efeito da execução de cada passo do programa construído (uma espécie de depurador).

Figura 2 - Furbot em sua versão para dispositivos móveis



Fonte: Mattos *et. al.* (2019c).

2.4 TRABALHOS CORRELATOS

Nesta seção são apresentados três trabalhos correlatos que possuem características semelhantes a proposta deste trabalho. O Quadro 1 descreve o trabalho de Kraemer (2017), que desenvolveu uma aplicação multiplataforma para dispositivos móveis para garantir a acessibilidade de deficientes visuais a um jogo de cartas. No Quadro 2 o trabalho de Costa (2013) detalha a criação de um jogo de tiros em primeira pessoa (*first-person shooter* - FPS) para deficientes visuais, utilizando som 3D e sistemas hápticos. No Quadro 3 o trabalho de Sobral *et. al.* (2017) detalha o desenvolvimento de um jogo digital no estilo *role playing game* (RPG) que auxilia no ensino de conceitos da área de Língua Portuguesa e Matemática, para crianças e adolescentes com deficiências visuais.

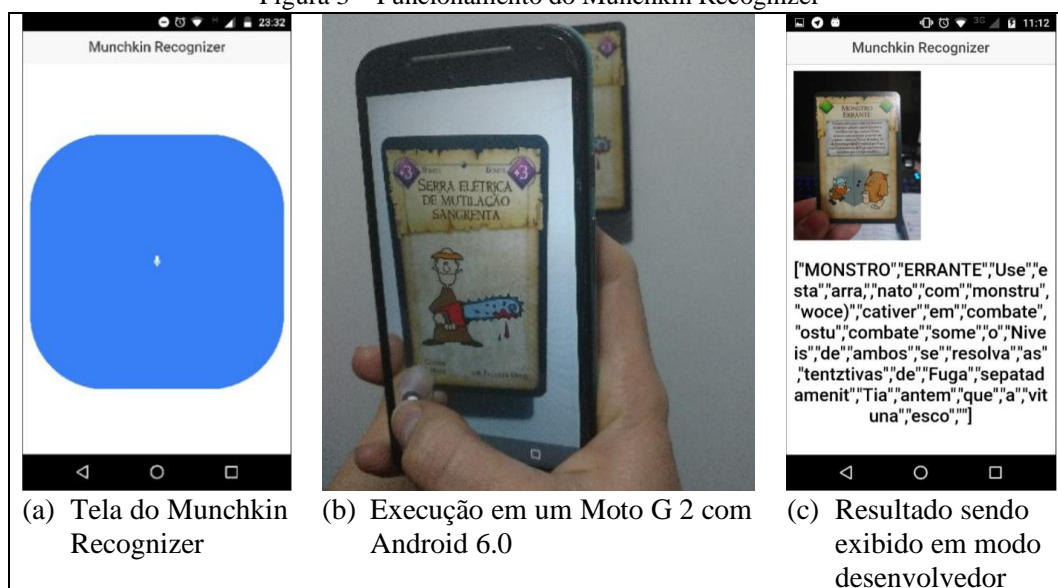
Quadro 1 – Tecnologia assistiva: Tornando jogo de mesa acessível para cegos com auxílio de aplicativo móvel de reconhecimento de imagem

Referência	Kraemer (2017)
Objetivos	Criar uma aplicação móvel multiplataforma para permitir que pessoas cegas possam participar de uma partida do jogo de cartas Munchkin.
Principais funcionalidades	Conforme a Figura 3 (a) o usuário permite o uso do microfone e se comunica com o aplicativo através de comandos de voz, após ter a permissão concedida, o usuário pode tirar uma foto da carta que deseja reconhecer, como demonstrado na Figura 3 (b). Após ter tirado a foto, deve ser feita a confirmação de que a foto que foi tirada será realmente utilizada para o reconhecimento, caso queira o usuário pode optar por tirar uma nova foto. Depois do usuário realizar a confirmação, o sistema irá reconhecer a carta e sintetizar por voz a descrição e informações que foram interpretadas. Estas informações são apresentadas na Figura 3 (c) através do modo desenvolvedor do aplicativo. Depois de realizar este processo, o usuário pode escolher ouvir novamente a descrição da carta ou reconhecer outra carta.
Ferramentas de desenvolvimento	Ambiente de desenvolvimento Visual Studio Code, framework Ionic, Google Cloud Vision API, HTML5, CSS, Javascript e AngularJS.
Resultados e conclusões	Nos testes individuais, o aplicativo demonstrou os melhores resultados em fotos com boa iluminação e utilizando uma rede wi-fi para o envio da foto para a API externa. Nos testes em grupo, ao aumentar os níveis de ruído do ambiente, a aplicação passou a distorcer comandos e eram necessárias novas tentativas. A utilização de fones de ouvido com microfone amenizou o problema do ruído.

Fonte: elaborado pela autora.

Elaborado por Kraemer (2017) o projeto apresenta o desenvolvimento de uma ferramenta de tecnologia assistiva para garantir acessibilidade de pessoas cegas e de baixa visão ao jogo de cartas Munchkin, utilizando um aplicativo multiplataforma para dispositivos móveis. O autor explica que é necessário fornecer ao aplicativo a permissão de acesso a câmera para que o Munchkin Recognizer funcione.

Figura 3 – Funcionamento do Munchkin Recognizer



Fonte: Kraemer (2017).

O autor identificou algumas possibilidades de melhoria, como tornar a ferramenta off-line, implementar o reconhecimento em tempo real, eliminando a necessidade de tirar uma foto da carta. Bem como eliminar a necessidade de um botão para realizar os comandos de voz, e adaptar a ideia para outros tipos de jogos de mesa.

Quadro 2 – Blind Counter-Strike: um jogo FPS para deficientes visuais

Referência	Costa (2013)
Objetivos	Criar um jogo estilo <i>first person shooter</i> (FPS) acessível para pessoas cegas, utilizando técnicas para eliminar a necessidade de recursos visuais.
Principais funcionalidades	O jogo conta com um menu sintetizado por voz que narra cada opção ao usuário, bem como, localiza o mesmo sobre em qual menu ele se encontra (menu inicial, tela de pause etc.). Além destas, Costa (2013) afirma que existem mais narrações no jogo com a intenção de contextualizar o jogador, como as informações sobre o objetivo do jogo, como jogar, quantidade de vida e de munição durante as fases, posição e direção do personagem no mapa, e uma lista com o significado de cada efeito sonoro do jogo. Existem respostas hápticas para localizar o jogador quanto a estar sendo atingido, estar alinhado a um inimigo entre outras situações e som 3D para ajudar o jogador a se localizar. A jogabilidade do Blind Counter-Strike, se dá através de um controle de Xbox360 (Microsoft), utilização de fones de ouvido para ter acesso ao som 3D em que o jogo se baseia e dos acessórios desenvolvidos por Costa (2013). Tais acessórios são mostrados na Figura 4, sendo eles: uma cinta com vibradores que indica quando o jogador está recebendo tiros e um simulador de bengalas, que consistem em duas munhequeiras também com vibradores.
Ferramentas de desenvolvimento	Foi desenvolvido para a plataforma Windows 7 utilizando a <i>engine</i> gráfica XNA juntamente com a linguagem de programação C#, utilizando o ambiente Visual Studio. A comunicação com os hardwares se deu através do microcontrolador Arduino.
Resultados e conclusões	Na fase de testes o jogo se saiu bem, porém Costa (2013) identificou que no início do jogo muitas informações eram dadas em um curto período dificultando a assimilação do jogador, as sintetizações por voz de informações triviais foram provadas importantes, porém estas devem ser dosadas corretamente. As respostas hápticas foram pouco mencionadas nos testes, porém no que foi mencionado estas não tiveram um resultado tão bom por conta de explicações confusas e do excesso de informações ao início do jogo.

Fonte: elaborado pela autora.

Segundo Costa (2013), o objetivo do jogo Blind Counter-Strike é permitir que o jogador passe por cinco fases sozinho, encontrando e matando inimigos em diferentes níveis de dificuldade. O jogador conta com duas armas, uma mais fraca, porém com mais munição disponível e outra mais forte, mas sem tanta munição. Mesmo tendo sido criado para pessoas cegas ou de baixa visão, Costa (2013) desenvolveu uma parte gráfica para o jogo, principalmente para questões de depuração durante o desenvolvimento.

Figura 4 - Cinta e simulador de bengalas dessentidos para o Blind Counter-Strike



Fonte: Costa (2013).

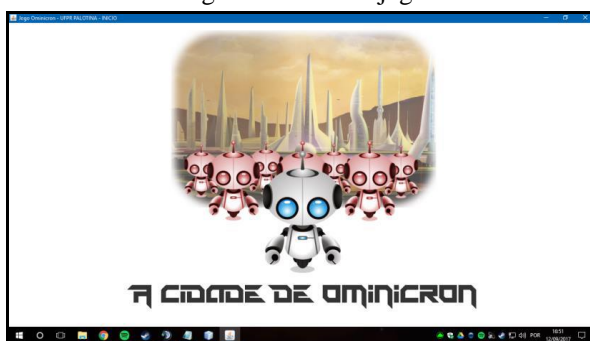
Quadro 3 - A utilização de *role playing* games digitais como ferramenta complementar no processo de aprendizagem de crianças deficientes visuais

Referência	Sobral <i>et. al.</i> (2017)
Objetivos	Entender como ocorre o processo de interação de deficientes visuais com sistemas computacionais na aprendizagem e identificar recursos de entretenimento para esses indivíduos, e com base no conhecimento adquirido criar um jogo.
Principais funcionalidades	O jogo é um áudio game no formato <i>role playing</i> game (RPG), também conhecido como jogo de interpretação, em que o jogador assume o papel do protagonista e passa por uma história sendo que a dificuldade aumenta gradativamente junto com o nível do jogador.
Ferramentas de desenvolvimento	Foi utilizada a linguagem de programação Java, outras ferramentas não foram identificadas no artigo.
Resultados e conclusões	Os autores verificaram que os alunos com deficiência visual se interessaram pelo jogo, porém foram relatadas dificuldades de compreensão na narração da história por conta da velocidade da fala.

Fonte: elaborado pela autora.

Sobral *et. al.* (2017) indica que a princípio foi feita uma coleta de dados, através de entrevistas na Secretaria da Educação e em instituições de atendimento a deficientes visuais. Partindo destas informações foi desenvolvido o áudio game A Cidade de Ominicron. O jogo ocorre inteiramente na tela apresentada na Figura 5, e toda a ambientação ocorre através do som. A interface desenvolvida por Sobral *et. al.* (2017) é baseada no uso de efeitos sonoros e narrações instrutivas, que levam os jogadores a fazerem suas escolhas, caso os jogadores errem muitas vezes eles recebem dicas de como prosseguir.

Figura 5 - Tela do jogo



Fonte: Sobral *et. al.* (2017).

3 DESCRIÇÃO DO PROTÓTIPO

Esta seção apresenta os detalhes de especificação e implementação do protótipo, para isso são apresentadas três seções. Primeiramente é apresentada a visão geral do protótipo, de forma a ambientar o leitor quanto ao funcionamento das novas funcionalidades implementadas. A seguir são apresentados aspectos da implementação do protótipo. Finalmente é realizada uma análise comparativa com a versão anterior do jogo.

3.1 VISÃO GERAL

O protótipo tem como objetivo viabilizar a acessibilidade para pessoas não videntes no jogo Furbot em sua versão Desktop (Windows / MacOS / Linux), para isso, foi criado um novo modo de jogo, chamado “modo labirinto”, que utiliza descrição de tela como forma de audiodescrição, e um cenário escuro, limitando o campo de visão para usuários videntes. As modificações realizadas, como a diminuição da visibilidade de cenário, foram introduzidas com a

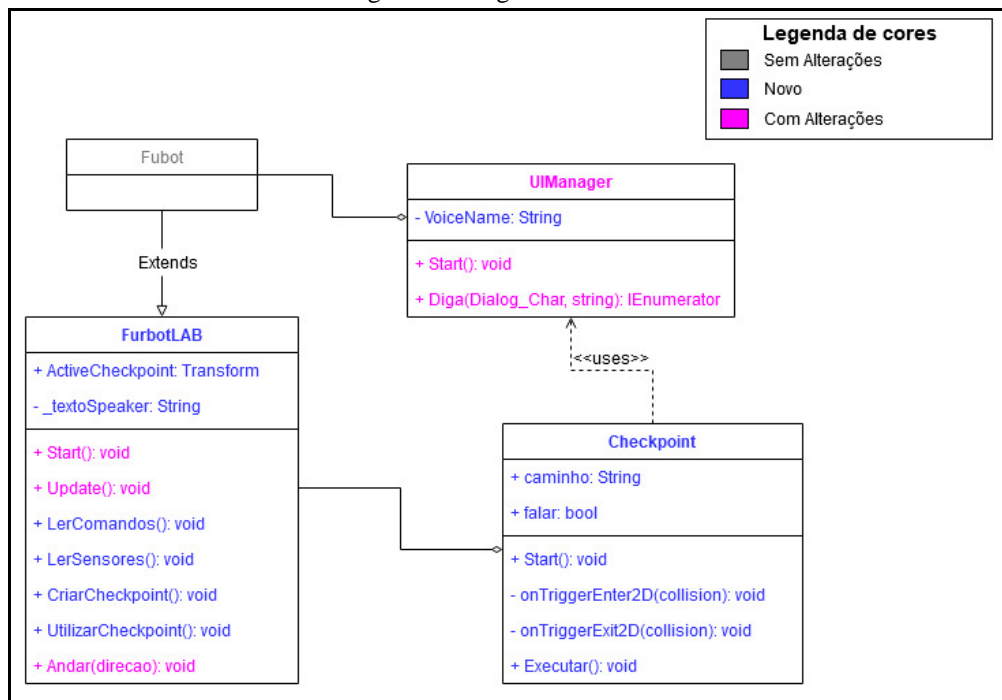
intenção de nivelar a dificuldade entre todos os tipos de usuários. Para a implementação do protótipo, foi utilizada a versão 3.0 do Furbot como base.

O desenvolvimento do protótipo seguiu os principais Requisitos Funcionais (RF) e Requisitos Não Funcionais (RNF) destacados a seguir:

- permitir ao usuário se localizar no ambiente sem depender de recursos visuais (RF);
- permitir ao usuário realizar as atividades sem se beneficiar da descrição do ambiente (RF);
- permitir que o usuário configure a voz do TTS conforme suas preferências (RF);
- permitir que o usuário faça uso de pontos de controle no decorrer do jogo para evitar que fique preso em *dead-ends* (RF)
- criar um ambiente que permita que usuários videntes e não videntes joguem com o mesmo nível de dificuldade (RNF);
- criar uma forma de orientar o usuário de um ponto de controle ao outro (RNF);
- utilizar audiodescrição para ambientar o usuário (RNF);
- utilizar o ambiente de desenvolvimento Unity 2D e linguagem C# (RNF).

O Furbot como um todo recebeu modificações para possibilitar a utilização do *asset* RTVoice, sendo que toda a plataforma recebeu o recurso de TTS desde o menu principal até a fase de teste. Uma série de adaptações foram realizadas dentro da fase para que a acessibilidade fosse possível. De acordo com o diagrama de classes apresentado na Figura 6, foi criada a classe `FurbotLAB` estendendo da classe `Furbot` já existente no projeto. Dentro dela ocorreram modificações nos métodos `Start` e `Update`, para abrigar novas funcionalidades, sendo elas os checkpoints, a leitura de sensores, a leitura de comandos disponíveis, a movimentação pelo teclado e a leitura de status do Furbot.

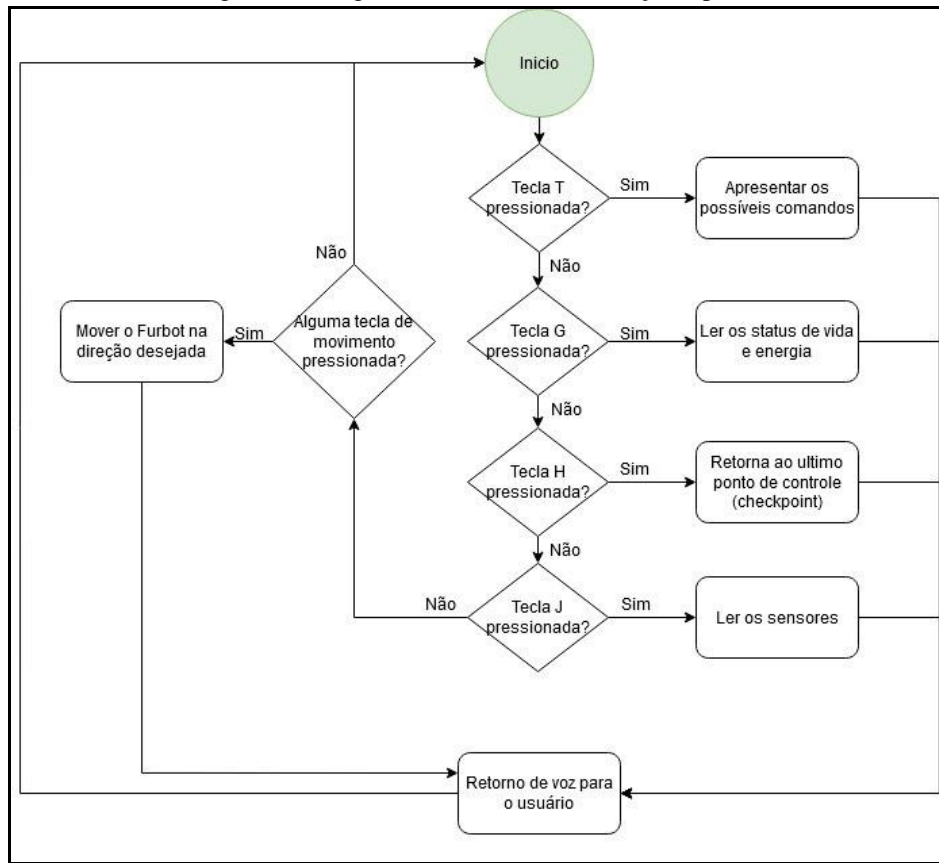
Figura 6 - Diagrama de classes



Fonte: elaborado pela autora.

A lógica do método `Update` da classe `FurbotLAB` foi um dois mais modificados, tendo como base o método `Update` da classe `Furbot` (a Figura 7 ilustra o processo modificado). O método aguarda alguma das teclas de comando ser pressionada, para então fazer alguma ação determinada por aquela tecla. A tecla `T` é utilizada caso o jogador queira ouvir todos os comandos disponíveis. Nesse caso é informado que para mover o Furbot podem ser utilizadas as setas direcionais, ou a combinação `W, A, S, D`, bem como a tecla `H` para utilizar o "portal" para o último checkpoint visitado. Já tecla `G` é usada para fazer a leitura dos sensores e informar o jogador do que se encontra em torno do Furbot. E a tecla `J` é usada para executar o diagnóstico, que vai informar ao jogador a quantidade de energia e vidas restantes. Quando um checkpoint é utilizado, o Furbot retorna totalmente ao estado salvo, exceto pelo código final, ou seja, a vida e energia do Furbot voltam ao estado que foi salvo no checkpoint, independentemente do jogador ter encontrado vidas extras ou energia.

Figura 7 – Diagrama de atividades da função Update



Fonte: elaborado pela autora.

3.2 IMPLEMENTAÇÃO

Esta seção traz os aspectos da implementação do protótipo, apresentando as novas funcionalidades integradas na fase de testes do protótipo e a aplicação do TTS.

A primeira grande alteração foi a aplicação do TTS ao Furbot, para isso foi utilizado o *asset* RTVoice Pro disponibilizado pela Crosstales. O *asset* foi incluído no menu inicial do jogo, sem necessidade de ser replicado em outras cenas. Nos fontes foi necessário a utilização do `Crosstales.RTVoice` onde o `Speaker` foi utilizado, para o correto funcionamento do TTS.

A drone S-223 foi utilizada para reproduzir as falas, sendo mantida como uma auxiliar para o Furbot, causando uma pequena alteração no método `Diga` da classe `UIManager`, que recebeu o trecho de código referente ao `Speaker` do `RTVoice` conforme o código no Quadro 4, através do método `Speaker.Speak()` o *asset* transmite o texto para o TTS nativo do dispositivo, informando também a voz que deve ser utilizada, e dando um retorno praticamente em tempo real (Crosstales, 2020). Neste caso foram utilizados 3 parâmetros opcionais do método `Speak()`, que controlam se o retorno em áudio deve ser dado automaticamente ou não, a velocidade de fala e a tonalidade do TTS.

Quadro 4 – Método Diga()

```
using Crosstales.RTVoice;
.
.
.
public IEnumerator Diga(Dialog_Char personagem, string texto)
{
    //debug.SetActive(true);
    ifCodigo.readOnly = true;
    dialogPanel.MostrarTexto(personagem, texto);
    Speaker.Speak(texto, null, Speaker.VoiceForName(PlayerPrefs.GetString("VoiceName")),
    true, PlayerPrefs.GetFloat("Rate"), PlayerPrefs.GetFloat("Pitch"));
    yield return new WaitForSeconds(3.0f);
    debug.SetActive(false);
#if !UNITY_ANDROID && !UNITY_IOS
    ifCodigo.readOnly = false;
#endif
}
```

Fonte: elaborado pela autora.

Para realizar a adaptação do Furbot foram necessárias algumas modificações no funcionamento inicial do jogo. No protótipo o jogo não ocorre mais em dois momentos, a programação ocorre ao mesmo tempo que a execução, e as linhas de código são produzidas de forma mais automática, não dependendo do jogador. Além disto, foram realizadas alterações visuais no Furbot, escurecendo todo o cenário e permitindo que jogadores videntes tenham acesso através da visão a mesma área que os jogadores não videntes terão ao utilizar os sensores.

Foram criados dois botões no menu principal do jogo, permitindo que o jogador acesse o modo labirinto, e as configurações de usuário. Tais configurações alteram a voz, velocidade de fala e entonação utilizadas pelo TTS. As configurações ficam salvas em `PlayerPrefs` possibilitando que o jogo seja fechado e reaberto sem perder as seleções do usuário. Ao entrar nas configurações são carregadas as vozes em português disponíveis no dispositivo em um Dropdown, conforme o código exibido no Quadro 5.

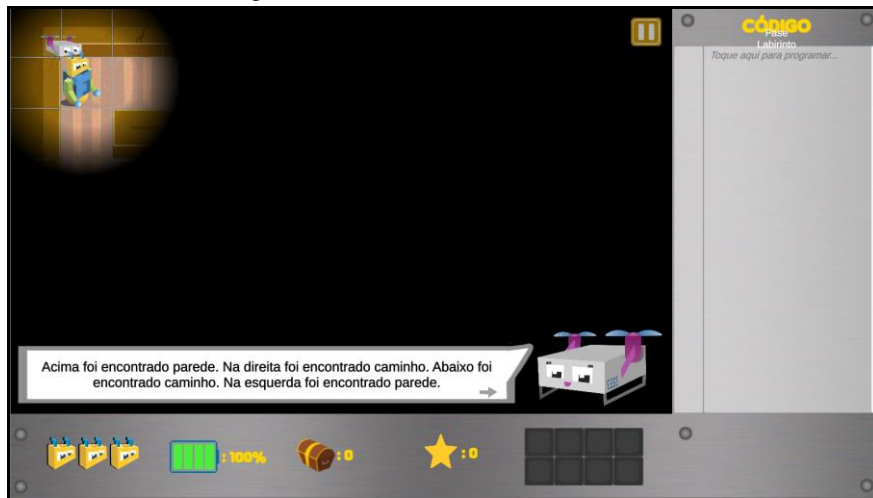
Quadro 5 - Função que carrega as vozes disponíveis para a seleção do usuário

```
void LoadVoices()
{
    voicesContainer.ClearOptions();
    if(Speaker.isTTSAvailable && Speaker.areVoicesReady && voicesContainer
    != null)
    {
        foreach (Crosstales.RTVoice.Model.Voice v in Speaker.Voices)
        {
            Dropdown.OptionData data = new Dropdown.OptionData();
            data.text = v.Name;
            voicesContainer.options.Add(data);
        }
    }
    else if(voicesContainer == null)
    {
        throw new System.Exception("Sem vozes disponíveis");
    }
}
```

Fonte: elaborado pela autora.

Dentro da fase foram acrescentadas novas funcionalidades ao Furbot. O método `Andar` recebeu um retorno sonoro, dizendo ao jogador para que direção ele andou. Além disso, houve alteração para que os movimentos executados pelo Furbot sejam em tempo de execução. Para possibilitar que um usuário não vidente se localize quanto a obstáculos, como paredes, foi criada a função `LerSensores`, que busca as informações dos sensores do Furbot e as traz ao jogador de forma sonora, conforme apresentando na Figura 8.

Figura 8 - Retorno da leitura de sensores



Fonte: elaborado pela autora.

A função `LerComandos` dá ao jogador todas as teclas e comandos disponíveis, informando como utilizá-los, como um breve tutorial. A função de leitura de status retorna ao jogador a quantidade de vidas disponível e a quantidade de energia que o robô ainda possui (Figura 9).

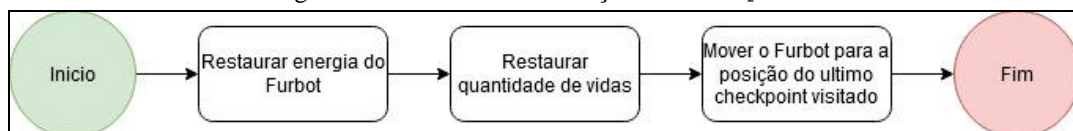
Figura 9 - Retorno da leitura de status



Fonte: elaborado pela autora.

Para garantir os avanços durante a fase foi criado um sistema de checkpoints, ou seja, o jogador atinge pontos específicos do mapa, aos quais ele pode retornar caso se perca no labirinto. Ao entrar em um checkpoint o jogador recebe as coordenadas para o próximo. Essas informações são passadas apenas na primeira vez que o jogador chega ao checkpoint. Caso o jogador se perca, ele pode optar por voltar ao último checkpoint visitado, e neste processo ele também restaura os status do robô salvos no último checkpoint, ignorando caso o jogador tenha encontrado uma vida ou energia antes de se perder. O processo de utilização do checkpoint é ilustrado pela Figura 10.

Figura 10 - Processo de utilização do checkpoint



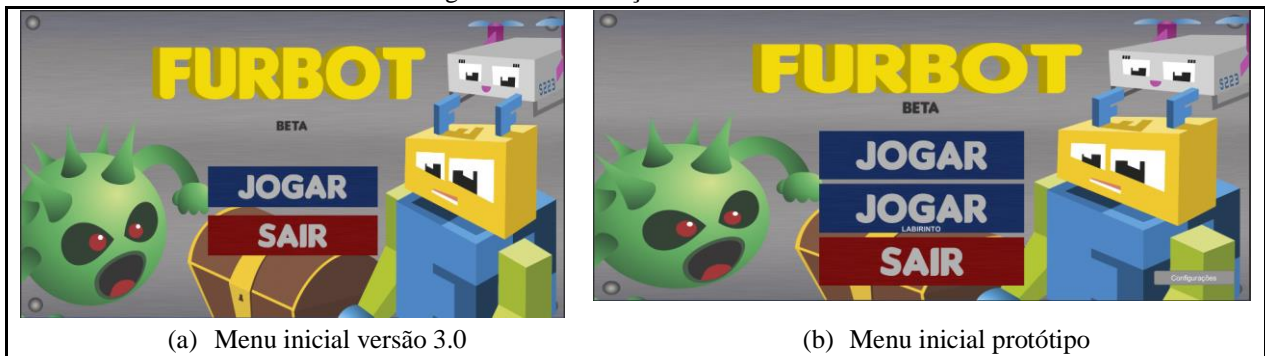
Fonte: elaborado pela autora.

3.3 ANÁLISE COMPARATIVA

Nesta seção são abordadas as diferenças entre a versão 3.0 do Furbot e o protótipo criado no decorrer deste trabalho. A primeira grande diferença entre as duas versões está na remoção da música de fundo que havia na versão 3.0 para dar lugar ao TTS. O menu inicial recebeu alterações já mencionadas, ganhando dois novos botões e a navegação

através das setas direcionais do teclado e da tecla Enter. As diferenças do menu inicial do protótipo e da versão 3.0 são exibidas na Figura 11.

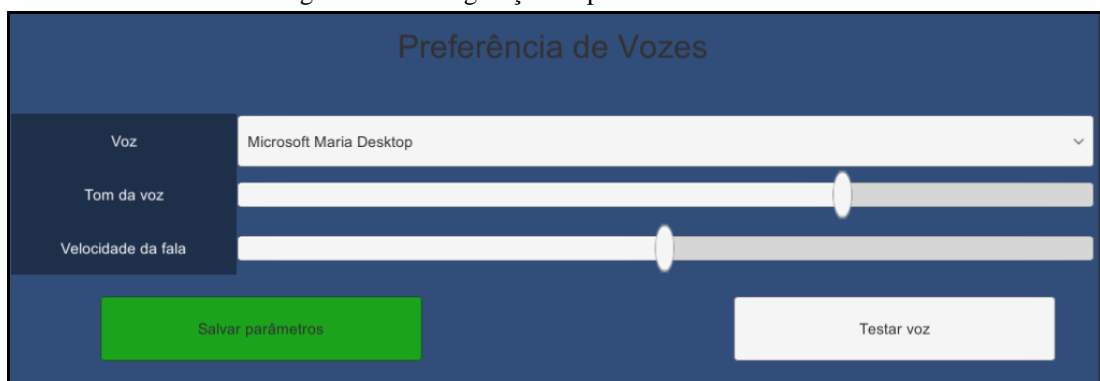
Figura 11 - Diferenças do menu inicial



Fonte: elaborado pela autora.

Quando um modo de jogo é escolhido no protótipo e não existe uma configuração de voz salva, o jogo remete o usuário a uma tela de parametrização de voz (Figura 12). Nessa tela, é possível definir o tipo da voz dentre as já existentes no dispositivo do usuário, a velocidade de fala e a entonação da voz. É possível perceber que nessa tela os tipos de vozes estão limitados ao idioma português brasileiro (pt-BR), não existindo a possibilidade de utilizar um idioma estrangeiro, salvo caso do usuário não tenha nenhuma voz em português instalada em seu computador. Neste caso a biblioteca RTVoice utiliza a primeira voz disponível para a fala, geralmente no idioma inglês. Após salvas, as preferências de voz podem ser alteradas a partir do menu inicial do Furbot no botão *Configurações*.

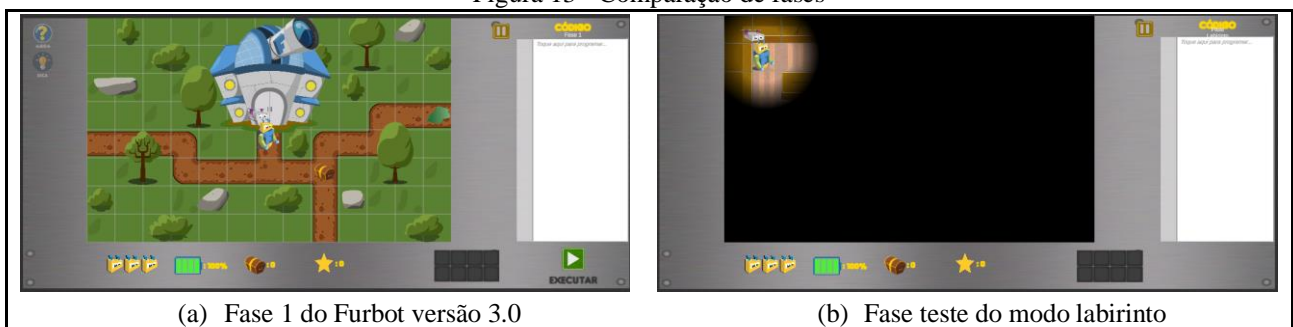
Figura 12 - Configuração de preferências do usuário



Fonte: elaborado pela autora.

A fase criada no protótipo tem um cenário escuro muito diferente do cenário apresentado na fase 1 da versão 3.0, conforme a Figura 13. No protótipo o jogador deve encontrar a pista que o leva a finalizar a fase seguindo as orientações dadas pela assistente S-223 ao longo do percurso. Essas orientações são dadas conforme o jogador atinge os checkpoints no labirinto.

Figura 13 - Comparação de fases

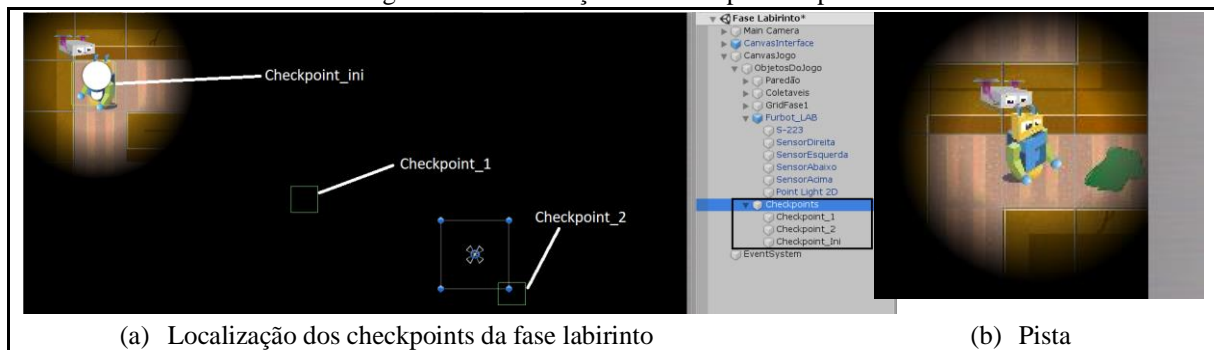


Fonte: elaborado pela autora.

Na fase de teste do protótipo existem três checkpoints (representados na Figura 14 (a) por legendas que não existem no jogo em si) ao todo, sendo um deles o ponto de partida da fase. Caso o jogador saia de um checkpoint do meio da fase e retorne a um checkpoint anterior, o progresso do jogador é salvo no checkpoint anterior novamente, pois o jogador perdeu energia e aumentou o tamanho do código final do Furbot. O modo labirinto acaba quando o

jogador encontrar a pista que o Furbot está procurando, que pode ser vista na Figura 14 (b), ou quando a energia ou vidas do Furbot se esgotarem.

Figura 14 - Localização dos checkpoints e pista



Fonte: elaborado pela autora.

Toda ação executada no modo labirinto tem um retorno em áudio, caso o jogador bata em uma parede, a assistente S-223 irá informar que o caminho está incorreto (Figura 15). Caso o jogador se movimenta, ela irá informar que o Furbot andou na direção informada. Porém para diminuir o processo este retorno é apenas pelo TTS, não envolvendo a caixa de diálogo.

Figura 15 - Diálogo de assistência S-223



Fonte: elaborado pela autora.

4 RESULTADOS

Para avaliar o projeto, foi realizada uma reunião em maio de 2020 com as especialistas Fernanda e Leia (ANDRADE, PACHECO, 2020) do Centro Municipal de Educação Alternativa (CEMEA) de Blumenau. Durante a entrevista foi feita uma demonstração do protótipo desenvolvido para as especialistas, as quais fizeram algumas sugestões de melhorias. A versão utilizada para demonstração na reunião possuía a possibilidade de criação de checkpoints pelo jogador, não tinha checkpoints fixos e nenhum tipo de orientação sobre os caminhos a serem seguidos. Além disso não permitia a possibilidade de configurações da voz do TTS. Estas modificações foram implementadas na *release* seguinte. Além das modificações feitas, a especialista Fernanda sugeriu o uso de som 3D próximo a objetivos ou obstáculos, e que acabou não sendo implementado.

Após as alterações o protótipo foi testado por cinco usuários, destes um usuário realizou o teste vendado e um usuário possui baixa visão. Para a coleta de informações destes testes foi aplicado um formulário on-line via Google Forms. Do total de usuários testados, todos utilizam o computador com frequência e costumam jogar no computador, 60% já havia tido algum contato com a plataforma Furbot, e apenas um deles estava familiarizado com leitores de tela.

Nenhum dos usuários teve dificuldades em navegar pelo menu inicial. Na tela de configuração do TTS o usuário que testou vendado relatou que precisou de ajuda com as configurações, pois a tela não possui um guia em TTS. Durante a fase teste do jogo, nenhum usuário relatou dificuldades com as novas funções. Porém um usuário identificou que ocorreram algumas falhas ao informar os status do Furbot, que estava com 72% de energia e ao ler os sensores foi lido 76%. Apenas o usuário que testou vendado precisou de auxílio para interagir com a plataforma, e todos os usuários classificaram as instruções dadas durante a fase como de fácil compreensão. No geral, os usuários se sentiram bem motivados a concluir a fase e acharam o sistema de checkpoints de fácil uso.

Todos os usuários conseguiram passar pela fase de teste sem ajuda e sem dificuldades, para testes mais conclusivos sobre o nivelamento de dificuldade, seria necessário realizar mais testes com usuários não videntes. Um usuário classificou a aplicação do TTS como ruim, pois ao pular falas ou executar ações muito rápido, as falas se sobrepõem, o que gera confusão. Algumas sugestões de melhorias foram apontadas pelos usuários, como encerrar a fala do TTS em execução ao rodar a próxima evitando que as falas se sobreponham. Ajustar a quantidade de textos por caixa de diálogo ou criar uma forma de elas serem roláveis, e ajustar o cálculo das posições, pois dependendo a resolução selecionada ao andar o Furbot acaba se perdendo, encontrando paredes onde não existem dentre outras situações semelhantes.

5 CONCLUSÕES

Apesar do pequeno número de avaliações, considera-se que o protótipo cumpriu seus objetivos, mostrando que é possível aplicar um leitor de tela ao jogo Furbot em sua versão desktop para torná-la acessível a pessoas não videntes, sem perder a essência da plataforma. Contudo foram encontradas algumas limitações na implementação, tais como: no menu inicial, caso os botões percam o foco, não é possível controlar o menu pelas setas do teclado. As caixas de diálogo ficam muito pequenas para textos maiores, e não foi encontrado uma forma de apresentar os diálogos em caixas separadas que saem em sequência, pois a caixa de diálogo se trata de uma co-rotina. O jogo deve ser rodado em resoluções específicas (1360x768 e 1600x1024) ou o jogo perde as referências das posições dos elementos como trilha, obstáculos e coletáveis. Esta limitação está presente na versão 3.0 do Furbot.

Diferente do trabalho de Sobral *et. al.* (2017), o protótipo desenvolvido permite que o usuário acompanhe mais facilmente as instruções e as falas, pois permite que o usuário configure parâmetros da voz utilizada conforme suas preferências. Da mesma forma que Costa (2013), existiram algumas dificuldades na dosagem de informações, com alguns diálogos ficando muito extensos, e um pouco repetitivos. Diferindo do trabalho de Kraemer (2017), não foi implementada nenhuma forma de reconhecimento de voz, sendo que todas as ações são realizadas através do teclado.

Analisando a estratégia de orientar o usuário quanto ao caminho até o próximo *checkpoint* orientada pela especialista Fernanda (PACHECO, 2020), foi constatado que esta estratégia diverge do objetivo inicial do Furbot, indicando o caminho a ser seguido pelo jogador e não estimulando o pensamento lógico e a navegação. Durante a fase de testes foram encontrados alguns pontos de melhoria, como melhorar a tela de configuração da voz possibilitando que um usuário não vidente configure o TTS sozinho, estender o uso do TTS aos dispositivos móveis e web, aplicar o som 3D sugerido pela especialista Fernanda (PACHECO, 2020), melhorar a quantidade de diálogos para orientar melhor os usuários, criar um sistema de *checkpoints* e orientações de direção dinâmico, entre outras melhorias.

REFERÊNCIAS

- ANDRADE, L. P. de; PACHECO, F. J. **Entrevista sobre demonstração do protótipo**. Entrevistador: Caroline Batistel. Blumenau. 2020. Entrevista feita em reunião pelo Hangouts – não publicada.
- BRASIL. Lei nº 10.098, de 19 de dezembro de 2000. Lei de promoção a acessibilidade, garantindo direitos as pessoas com deficiência. **Diário Oficial da União**: seção 1, Brasília, DF, p. 2, 20 dez. 2000.
- BRASIL Portaria nº 142, de 16 de novembro de 2006. Institui o Comitê de Ajudas Técnicas (CAT). **Diário Oficial da União**: seção 2, Brasília, DF, p. 3, 16 nov. 2006.
- BRASIL. Subsecretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência. Comitê de Ajudas Técnicas. **Tecnologia Assistiva** – Brasília: CORDE, 2009. 138 p.
- COOK, A. M.; POLGAR, J. M. **Assistive Technologies: Principles and Practice**. 4 ed. St. Louis, Missouri: Elsevier Health Sciences, 2014.
- COSTA, D. **Blind Counter-Strike: Um jogo de FPS para deficientes visuais**. 2013. 74 f. Monografia (Bacharelado em Ciência da Computação) - Curso de Ciência da Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre.
- CROSSTALES. **RT-Voice: Basic Tutorial**. 2017. Disponível em: <https://www.youtube.com/watch?v=OJyVgCmX3wU>. Acesso em: 10 abr. 2020.
- CROSSTALES. **RT-Voice PRO: Hearing is understanding - Documentation**. 2020. Disponível em: <https://www.crosstales.com/media/data/assets/rtvoice/RTVoice-doc.pdf>. Acesso em: 21 jun. 2020.
- DIAS, C. **Usabilidade na WEB**. Rio de Janeiro: Alta Books, 2007.
- FRANCO, E. P. C.; SILVA, M. C. C. da. Audiodescrição: Breve Passeio Histórico. In MOTTA, L. M. V. M.; FILHO, P. R. (org.). **Audiodescrição. Transformando Imagens em Palavras**, São Paulo: Secretaria de Estado dos Direitos da Pessoa com Deficiência, 2010, p. 23-42.
- GRANOLLERS, T. **MPlu+a Una metodologia que integra la ingeniería del software, la interacción persona-ordenador y la accesibilidad en el contexto de equipos de desarrollo multidisciplinares**. 2004. 77 f. Tese de doutorado - Ciência da Computação, especialidade em Interação Humano-Computador, Universidade de Lérida, Lérida.

GROS, B. The impact of digital games in education. **First Monday**, v. 8, n. 7, jul. 2003.

IBGE – Instituto Brasileiro de Geografia e Estatística. **Censo demográfico 2010**: Características da população e dos domicílios – Resultados do Universo. Rio de Janeiro. 2011.

KRAEMER, R. G. **Tecnologia Assistiva: Tornando Jogo de Mesa Acessível para Cegos com Auxílio de Aplicativo Móvel de Reconhecimento de Imagem**. 2017. 63 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) - Curso de Ciência da Computação, Universidade Regional de Blumenau, Blumenau

MATTOS, M.; SANTOS, B. F. F.; TRIDAPALLI, J. G.; ZUCCO, F.; WUO, A. FURBOT - Desenvolvimento cognitivo infantil através de atividades de programação de computadores. In: SEMINÁRIO DE EXTENSÃO UNIVERSITÁRIA DA REGIÃO SUL, 37 ed., 2019, Florianópolis. **Anais...** Florianópolis: UFSC, 2019a. p. 112

MATTOS, M.; KOHLER, L. P. A.; ZUCCO, F. D.; FRONZA, L.; BIZON, A.; UGARTE, H.; SANTOS, B. F. F.; GIOVANELLA, G. C. Ambiente de programação para a introdução da lógica de programação **Anais do Workshop de Informática na Escola** [S.l.], ISSN 2316-6541, p. 1259-1264, nov. 2019b.

MATTOS M.; KOHLER L. P. de A.; ZUCCO F.; WUO A.; SANTOS B.; TRIDAPALLI J. G.; SILVEIRA H. U. C. da; FRONZA L.; GIOVANELLA G. C.; LARGURA L.; MELO J.; KOHLS A.; WESSLING R. **Furbot Móvel**: um jogo para o ensino do pensamento computacional. In: VIII Congresso Brasileiro de Informática na Educação (CBIE 2019), 8., 2019, Petrópolis. Anais dos Workshops do VIII Congresso Brasileiro de Informática na Educação: WCBIE, 2019c. p. 1294-1301.

NUNES E. V.; MACHADO F. O.; VANZIN T. Audiodescrição como tecnologia assistiva para o acesso ao conhecimento por pessoas cegas. In: ULBRICHT V. R.; VANZIN T.; VILLAROUÇO V. (org.). **Ambiente virtual de aprendizagem inclusivo**, Florianópolis: Padion, 2011, p. 191-232.

PASSERINO, L. M.; MONTARDO, S. P. Inclusão social via acessibilidade digital: proposta de inclusão digital para Pessoas com Necessidades Especiais. **E-Compós**, v. 8, n 11, abril 2007

PRIETO, L. M.; TREVISAN, M. C. B.; DANESI, M. I.; FALKEMBACH G. A. M.; Uso das Tecnologias Digitais em Atividades Didáticas nas Séries Iniciais. **Renote: revista novas tecnologias na educação**, Porto Alegre, v. 3, n. 1, p.1-11, maio 2005.

SASSAKI, R. K. **Inclusão: construindo uma sociedade para todos**. Rio de Janeiro: WVA, 1997.

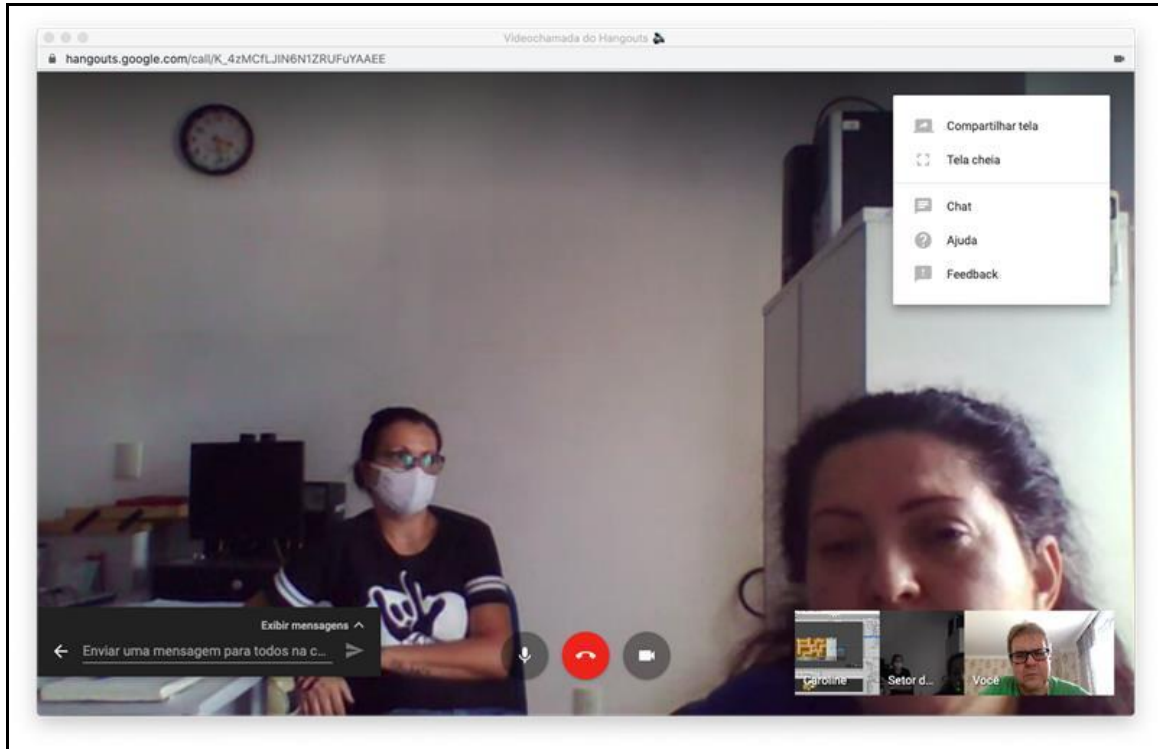
SAVI, R.; ULBRICHT, V. R. Jogos digitais educacionais: benefícios e desafios. **Renote: Revista novas tecnologias na educação**, Porto Alegre, v. 6, n. 2, 2008.

SOBRAL, F. V.; UMERES, L. F.; SCHANOSKI, W.; BARTELMERBS, C. R.; ASSIS, M. V. O. de. A Utilização de Role Playing Games Digitais como Ferramenta Complementar no Processo de Aprendizagem de Crianças Deficientes Visuais. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 6 ed., 2017, Recife. **Anais...** Recife: UFRPE, 2017. p. 635 – 644.

APÊNDICE A – REUNIÃO DE DEMONSTRAÇÃO

Neste apêndice é apresentada uma captura de tela da reunião com as especialistas do Centro Municipal de Educação Alternativa (CEMEA) de Blumenau, que assistiram uma demonstração do protótipo e deram suas considerações referente as novas funcionalidades e a aplicação do TTS. A entrevista foi realizada no dia 12 de maio de 2020.

Figura 16 - Especialistas do CEMEA



Fonte: arquivo pessoal.