

A 3D Simulator for a Social and Ludic Robotics Platform

Marcelo Marques da Rocha¹, Esteban Clua², Débora Christina Muchaluat-Saade¹

¹MídiaCom Lab, ²Medialab
Instituto de Computação, Universidade Federal Fluminense

{marcelo_rocha,debor}@midia.com.uff.br, esteban@ic.uff.br

Abstract. *Robots are expected to become ubiquitous devices with the ability to transform people's lives. Socially Assistive Robots (SARs) can provide assistance to humans through stimuli during the interaction process. This work presents a proposal for a 3D simulator for an open source and low cost robotics platform that can be used in educational applications and health therapies. The physical robot has elements of verbal and non-verbal communication and can express emotions through the eyes and mouth, through the colors of the LEDs on its thorax and also through body expressions through poses and movements (dance). The proposed simulator is able to reproduce these capabilities in a similar way to the physical robot, serving as an auxiliary tool in the teaching/learning process and tests of the interaction scripts developed for the robot. The simulated robot can be controlled through the publish/subscribe mechanism of the MQTT protocol, since each of its functionalities was associated with a topic and a message. To serve as a proof of concept for the simulator proposed in this work, a demonstrative video was presented with both robots, the physical and the virtual version (simulated).*

Resumo. *Espera-se que os robôs se tornem dispositivos onipresentes com a capacidade de transformar a vida das pessoas. Robôs Socialmente Assistivos (SARs - Socially Assistive Robots) podem fornecer assistência aos humanos por meio de estímulos durante o processo de interação. Este trabalho apresenta uma proposta de simulador 3D para uma plataforma de robótica social de código aberto e baixo custo que pode ser utilizada em aplicações educacionais e terapias em saúde. O robô físico possui elementos de comunicação verbal e não verbal e pode expressar emoções através dos olhos e da boca, através das cores dos LEDs em seu tórax e também através de expressões corporais por meio de poses e movimentos (danças). O simulador proposto é capaz de reproduzir essas capacidades de forma similar ao robô físico, servindo como ferramenta auxiliar no processo de ensino/aprendizado e testes dos scripts de interação desenvolvidos para o robô. O robô simulado pode ser controlado através do mecanismo publish/subscribe do protocolo MQTT, pois cada uma de suas funcionalidades foi associada a um tópico e a uma mensagem. Para servir de prova de conceito para o simulador proposto neste trabalho, foi apresentado um vídeo demonstrativo com ambos os robôs, na versão física e na versão virtual (simulada).*

1. Introdução

Espera-se que os robôs se tornem dispositivos onipresentes e sejam capazes de transformar a vida cotidiana das pessoas. Segundo [Feil-Seifer and Mataric 2005], além de fornecer algum tipo de assistência física ao ser humano, melhorando suas habilidades físicas,

a robótica é uma tecnologia que pode melhorar suas habilidades cognitivas. Para isso, os robôs sociais assistivos (SARs - *Socially Assistive Robots*) podem usar várias formas de comunicação: comunicação verbal (usando fala e reconhecimento de voz) e comunicação não-verbal (usando expressão de olhar e movimento, por exemplo). Os SARs têm sido usados tanto na área da saúde quanto na educação. Na área da saúde, principalmente em terapias com pacientes com Transtorno do Espectro Autista (TEA), existem estudos [Santos et al. 2021, Rocha et al. 2022b] que investigam a eficácia da aplicação de robôs sociais em terapias com crianças autistas, mostrando resultados positivos na interação da criança com o robô. A proposta de usar robôs no ensino de programação e outras disciplinas relacionadas a STEM¹ tem sido incentivada em todo o mundo [Johal et al. 2018]. Com os avanços das tecnologias robóticas e da inteligência artificial, os robôs tornaram-se mais baratos e acessíveis, tornando-se um instrumento com grande potencial de interação social com os alunos.

O robô FRED [Rocha and Muchaluaat-Saade 2023] é uma plataforma de robótica social *open-source* de baixo custo destinada a aplicações em saúde e educação. O novo robô foi inspirado na plataforma de robótica EVA [Cruz-Sandoval and Favela 2019, Rocha et al. 2022b] e no projeto do robô Otto². Ele tem seu corpo impresso em impressora 3D, utiliza hardware de baixo custo e possui capacidades de interação multimodal podendo se comunicar de maneira verbal e não verbal. Apesar de ser uma ferramenta *open-source* e de baixo custo, nem sempre é possível ter um robô físico completamente montado durante o processo de desenvolvimento e teste de *scripts* de interação. Isso dificulta o ensino e o aprendizado de programação para o robô, assim como a validação dos *scripts* desenvolvidos.

Este trabalho apresenta uma proposta de desenvolvimento de um sistema digital que permite simular as capacidades do robô físico FRED facilitando os testes e a validação dos *scripts* desenvolvidos. Os simuladores são uma solução comum para solucionar problemas relacionados a cenários de teste do mundo real [Perico et al. 2016]. Os simuladores de robôs têm sido utilizados como ferramentas no treinamento educacional com evidências de que os resultados obtidos com o uso do robô físico, durante o treinamento, são semelhantes aos obtidos com o uso do simulador [Kurniawan et al. 2018]. Além de permitirem recriar com facilidade situações semelhantes as reais, estes simuladores também permitem explorar aspectos lúdicos e gamificados do processo de lidar com ensino de programação.

Este trabalho propõem um simulador para uma plataforma de robótica social *open-source*. O simulador pode simular várias capacidades de comunicação da versão física do robô, como: expressão de emoções através dos olhos e da boca; comunicação não verbal através das luzes dos LEDs no tórax do robô; expressão corporal através da utilização de poses e movimentos (dança). O simulador pode ser controlado através do protocolo MQTT (*Message Queuing Telemetry Transport*) e isso permite que o robô simulado possa ser controlado, local ou remotamente, por qualquer aplicação que utilize esse protocolo.

Este trabalho está organizado da seguinte maneira. A Seção 2 discute os trabalhos relacionados. A Seção 3 mostra uma visão geral do robô físico com seus elementos de

¹STEM é o acrônimo formado pelas iniciais das palavras Science (ciência), Technology (tecnologia), Engineering (engenharia) e Mathematics (matemática), em inglês.

²<https://www.ottodiy.com>

hardware e software. A Seção 4 apresenta a proposta do simulador 3D descrevendo toda a etapa de concepção do modelo 3D, o processo de *rigging* e animação, assim como os *tópicos* e *mensagens* utilizadas para o controle do robô via o protocolo MQTT. A Seção 5 conclui este trabalho e apresenta os trabalhos futuros.

2. Trabalhos Relacionados

Vários campos da engenharia usam simuladores e a robótica não é exceção. A utilização do simulador facilita a realização de experimentos sem o risco de danificar o robô e também é conveniente para ensinar as pessoas a programá-los. Simuladores desempenham um papel importante no projeto e desenvolvimento de sistemas robóticos. Através da simulação é possível verificar a viabilidade do projeto e desenvolver experimentos que podem ajudar a prever possíveis problemas. Existem vários tipos de simuladores, alguns são de código aberto e outros são simuladores comerciais. De acordo com [Ayala et al. 2020] os simuladores de código aberto mais usados e bem avaliados são *Webots* e *Gazebo* e entre os simuladores de código fechado, o *V-REP*³ é o mais utilizado.

Na Coreia, a empresa Yujin Robotics⁴ criou os robôs iRobi e iRobiQ, que são robôs educacionais e domésticos. Os autores em [Truong et al. 2011] apresentaram uma proposta de interface para facilitar o desenvolvimento de conteúdo para um simulador baseado nesses robôs, que foi criado considerando o alto custo de robôs educacionais reais. O simulador é voltado para desenvolvedores de conteúdo educacional, como professores.

Em [Ishimura et al. 2003] foi proposto um ambiente de simulação de código aberto capaz de acomodar qualquer tipo de robô. O trabalho fornece o ambiente de simulação como um *test bed* primário para programadores iniciantes do robô AIBO⁵ da Sony, acelerando a curva de aprendizado. Mais um projeto de simulador robótico com proposta de facilitar o aprendizado de programação para a plataforma de robótica Pololu 3PI⁶ foi apresentado em [de Lima et al. 2018]. Em [Carpin et al. 2007] os autores apresentam uma proposta de simulador de código aberto de alta fidelidade que pode ser usado tanto em pesquisa quanto em educação, o USARSim.

Em [Gena et al. 2021] há uma proposta de aplicativo que visa ampliar a funcionalidade da versão simulada do robô NAO [Gouaillier et al. 2009, Pot et al. 2009], agregando ao simulador as características necessárias para o desenvolvimento de interações sociais. Os autores apontam que uma aplicação interessante do simulador seria utilizá-lo como um robô virtual que poderia ser utilizado para dar continuidade às terapias na casa do paciente, sem o uso do robô físico. Em [Rocha et al. 2022a] foi apresentada uma proposta de simulador para a plataforma de robótica *open-source* EVA. A proposta visa facilitar o teste e o desenvolvimento de interações sociais para o robô EVA.

Existem vários simuladores de robôs que foram testados em uma variedade de estudos como [Ayala et al. 2020, Kumar and Reel 2011]. Eles observaram que o uso de simuladores tem dois objetivos principais. O primeiro é auxiliar no processo de prototipagem e construção de robôs, fazendo uso de *frameworks* como a *engine* de física ODE

³Desde Novembro de 2019, o V-REP foi substituído pelo CoppeliaSim versão 4.0. Veja o simulador <https://www.coppeliarobotics.com/>

⁴<http://www.yujinrobot.com/>

⁵<https://us.aibo.com/>

⁶<https://www.pololu.com/category/279/original-3pi-robot>

(*Open Dynamics Engine*). O segundo objetivo é ajudar os usuários de robôs, seja na área industrial, educacional ou de saúde, a aprender a desenvolver programas que possam controlar esses robôs.

Diferentemente das propostas apresentadas nos estudos citados anteriormente, este trabalho apresenta um software simulador de código aberto para a plataforma de robótica social FRED, que também é uma plataforma robótica de código aberto. O simulador é multiplataforma e pode ter versões compiladas para Windows, OSX e Linux, podendo também ser compilado para rodar em navegadores *web* e dispositivos móveis. Diferente da proposta de [Rocha et al. 2022a], que apresenta um simulador 2D para o robô EVA, sem apresentar qualquer tipo de movimentação, não sendo capaz de simular a movimentação da cabeça do robô, o simulador proposto neste trabalho é 3D e pode simular os movimentos e poses do robô físico. Além disso, é possível visualizar o robô em um cenário, como o de uma sala de aula, controlando, inclusive, a luz ambiente.

Embora os simuladores robóticos sejam importantes para os desenvolvedores de robôs tradicionais em termos de segurança, para um robô social, é particularmente importante como o robô é percebido e se suas dicas sociais (*social cues*) são claramente identificadas pelo usuário. O simulador 3D proposto neste trabalho pode funcionar como ferramenta complementar ao robô físico, podendo ser utilizado pelos usuários que não possuem o robô físico disponível e também para o treinamento de técnicos, entusiastas de IHR (Interação Humano-Robô) e profissionais de educação e saúde no desenvolvimento de programas para o robô físico, facilitando a prototipagem rápida de intervenções centradas no usuário. Particularmente na área de educação, o simulador proposto pode ser extremamente útil para ser utilizado por professores em sala de aula ou pelos alunos em suas residências quando não possuem o robô físico disponível.

3. O Robô Físico

O robô FRED [Rocha and Muchaluat-Saade 2023] é uma plataforma de robótica social *open-source* destinada a aplicações em educação e saúde. O corpo do robô é todo impresso em 3D e ele possui dois braços que são articulados, mas não possuem nenhum controle automático. Óculos foram adicionados ao robô para torná-lo mais inclusivo, já que o robô também personifica usuários com problemas visuais. O robô tem cerca de 20 centímetros de altura e, completamente montado, com todos os seus componentes de hardware, pesa cerca de 345 gramas. Seu custo aproximado fica em torno de 200 dólares.

3.1. Componentes de Hardware

O robô é montado com elementos de hardware de baixo custo. Para a construção dos elementos da face do robô (olhos e boca), são utilizadas três matrizes de LEDs 8x8. O robô usa um anel de 16 LEDs RGB preso ao seu tórax. Esse anel de LEDs é utilizado como elemento de comunicação não verbal e possui cores associadas às emoções: verde (alegria), azul (tristeza) e vermelho (raiva). Quatro servomotores são usados para mover as pernas e os pés do robô. O FRED pode reproduzir sons e músicas através de uma caixa de som *bluetooth*. Uma *webcam*, com microfone embutido, é usada para capturar expressões faciais e a voz do usuário. O robô também pode controlar uma lâmpada inteligente XIAOMI, o que dá a ele a capacidade de adicionar efeitos sensoriais de luz durante as interações. O software de controle do robô é executado em um Raspberry Pi 4 e em uma placa Arduino UNO.

3.2. Componentes de Software

A arquitetura de software do robô é dividida em duas camadas. A camada de baixo nível é implementada dentro de uma placa Arduino que se conecta diretamente ao hardware interno do robô: 4 servomotores, 3 matrizes de LED 8x8 e um anel de LEDs RGB. A camada de alto nível é implementada dentro de um Raspberry PI 4. O software de controle do robô, executado nessa camada, usa o *framework* robótico ROS (*Robot Operating System*), que é considerado o *framework* padrão para o desenvolvimento de soluções robóticas [Mukhandi et al. 2019]. Essa camada contém os módulos ROS (nós) que conectam os dispositivos externos ao robô (lâmpada inteligente, microfone, *webcam*) e também se conectam aos serviços em nuvem (IBM-Watson e Google Speech API). A comunicação entre as duas camadas é feita através da porta serial.

3.3. Capacidades do Robô Físico

Apesar de ser uma plataforma robótica de baixo custo, que utiliza elementos simples para sua construção (peças 3D e hardware de baixo custo), ela oferece diversas capacidades de interação multimodal: comunicação verbal e não verbal, reconhecimento de expressões faciais (através de uma *webcam*) e controle de efeitos sensoriais de luz.

- **Fala.** O robô pode se comunicar verbalmente usando os serviços na nuvem de *Text-To-Speech* (TTS) do IBM-Watson ou a API Google Speech. Por meio dos recursos oferecidos por esses serviços, que são gratuitos, o software do robô pode enviar o conteúdo de uma *string* para a nuvem e receber esse conteúdo (falado) em formato de arquivo de áudio.
- **Reconhecimento de Voz.** Usando o serviço na nuvem de *Speech-To-Text* (STT) da API Google Speech, o robô é capaz de reconhecer a fala. Um arquivo de áudio contendo um trecho da fala do usuário, captado pelo robô através do microfone da *webcam*, é enviado ao serviço Google que retorna uma *string* contendo o texto referente à transcrição do áudio.
- **Reconhecimento de Expressões Faciais.** Usando imagens capturadas por uma *webcam*, o robô pode reconhecer as expressões faciais do usuário.
- **Controle de Efeitos Sensoriais de Luz.** Com o objetivo de tornar as sessões interativas mais atrativas, principalmente para as crianças, o robô tem a capacidade de controlar uma lâmpada inteligente sem fio, podendo ligá-la, desligá-la e definir sua cor.
- **Expressões Faciais.** Talvez uma das formas mais importantes de comunicação não-verbal seja através de expressões faciais. Apesar da simplicidade dos componentes usados para representar as expressões faciais do FRED, ele pode demonstrar 7 emoções: neutralidade ou indiferença (*neutral*), satisfação (*pleased*), felicidade (*happy*), tristeza (*sad*), raiva (*angry*), surpresa (*surprised*) e medo (*afraid*).
- **Animação com o Anel de LEDs.** O robô tem um anel com 16 LEDs RGB em seu tórax. Ele pode controlar as cores dos LEDs, bem como realizar algumas

animações com eles. Atualmente, o robô pode realizar 5 animações usando cores. Algumas delas estão associadas a emoções: raiva (vermelho), tristeza (azul) e felicidade (verde). O robô também pode realizar uma animação utilizando a cor branca e uma animação de arco-íris, onde apresenta uma sequência de cores imitando um arco-íris.

- **Expressão Corporal (Movimentos e Poses).** O robô pode movimentar suas pernas e pés usando 4 servomotores. Duas formas de expressão foram pensadas para o robô utilizando esses recursos. A primeira foi o uso de poses. Existem atualmente 5 poses definidas: a pose inicial (pose de equilíbrio), a pose com o pé esquerdo levantado, a pose com o pé direito levantado, a pose na ponta dos pés e a pose com os pés abertos, como se tivesse caído (quebrado). A segunda forma de expressão corporal pensada foi a caminhada e a dança. O robô pode andar para frente e para trás, para a esquerda e para a direita, e também pode dançar de três maneiras diferentes. As capacidades de expressão corporal (movimentos e poses) quando utilizadas em conjunto com os outros tipos de comunicação verbal e não verbal, apresentados anteriormente, podem aumentar consideravelmente a expressividade do robô.

4. O Simulador 3D

4.1. Modelo 3D, *Rigging* e Animação

Como o robô FRED é um projeto de robótica *open-source*, seus criadores disponibilizaram um modelo 3D para ser usado na impressão de suas partes. O modelo foi criado no software Tinkercad⁷ e é a base do modelo para a criação do robô no simulador 3D.

O processo de *rigging* e animação do robô foi feito no software Blender⁸. Os quatro servomotores nas pernas e pés do robô dão a ele 4 graus de liberdade. A Figura 1(a) mostra o posicionamento dos ossos (*bones*) da armadura do robô e o item (b), da mesma figura, mostra o grafo de cena no Blender contendo todos os elementos da cena, incluindo a estrutura hierárquica da armadura de *bones*.

A animação dos movimentos e poses do robô também foi feita no software Blender. Foram criadas animações para 4 poses: pé esquerdo levantado, pé direito levantado, pés para baixo (quebrado/triste) e ponta dos pés. Para cada uma dessas animações, foram criadas ainda as animações de retorno à posição inicial (posição de equilíbrio). Foram criadas quatro movimentações (danças): dança 1, dança 2, *moonwalk* para esquerda e *moonwalk* para a direita. Além disso, foi criada também a animação do robô batendo com o pé direito no chão (*stomping foot*). Essas animações encerram seus ciclos voltando à posição inicial do robô (posição de equilíbrio). A técnica de animação utilizada foi a animação por *keyframes* e para a sua criação foram utilizados 800 *frames*. O modelo gerado (malha 3D, *bones* (*rigging*) e animação) foi exportado no formato *fbx* a fim de ser importado no software Unity 3D.

⁷O Tinkercad é um aplicativo Web gratuito para projetos 3D, eletrônica e codificação.

⁸O Blender é uma suíte de criação 3D gratuita e de código aberto. Ele suporta todo o *pipeline* 3D - modelagem, *rigging*, animação, simulação, renderização, composição e rastreamento de movimento, até mesmo edição de vídeo e criação de jogos.

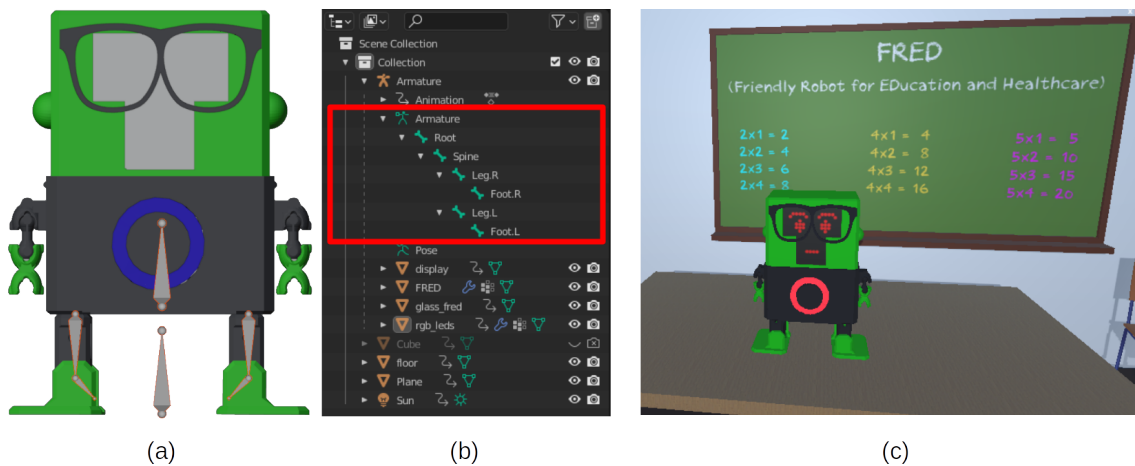


Figura 1. *Rigging* do robô com a armadura de *bones* (a), estrutura hierárquica dos *bones* sendo mostrada no grafo de cena no Blender (b) e o modelo 3D do robô inserido em um cenário de sala de aula dentro do Unity 3D (c).

4.2. Trabalhando no Unity 3D

Após a exportação do modelo no Blender, o arquivo *fbx* foi importado no Unity 3D. O arquivo foi importado como um *prefab* contendo a malha 3D, materiais com as texturas básicas e os 800 *frames* de animação do robô. Como uma das propostas do FRED é sua utilização na área de educação, para compor a cena do simulador, um cenário de sala de aula foi montado. Para isso, uma textura de lousa, com algumas contas (tabuada), foi adicionada ao plano que representa a lousa. A Figura 1(c) mostra a cena com robô dentro desse cenário.

As expressões faciais do robô físico são desenhadas utilizando-se 3 matrizes de LEDs 8x8. Para a representação do *display* do robô, no simulador, foram criadas 8 texturas representando as seguintes expressões: medo (*afraid*), raiva (*angry*), saudação (*greetings*), alegria (*happy*), neutralidade ou indiferença (*neutral*), satisfação (*pleased*), tristeza (*sad*) e surpresa (*surprised*). Além dessas texturas, uma textura com a imagem da lousa também foi criada e para mesa da sala de aula, uma textura de madeira foi utilizada. Essas texturas podem ser vistas na Figura 2(a).

4.3. Controlando o Simulador

Com o objetivo de permitir que o simulador seja controlado facilmente por outras aplicações, todas as capacidades do robô podem ser controladas através do protocolo MQTT⁹. Cada funcionalidade do robô simulado está associada a um *tópico* MQTT e os estados do robô podem ser modificados através da *publicação* de *mensagens* nesses *tópicos*.

Inicialmente, o simulador está configurado para se conectar localmente a um *broker* MQTT na porta 1883, porta *default*. Para o desenvolvimento da aplicação do simulador e para os testes, o *broker* Mosquitto¹⁰ foi utilizado. Após a conexão com o *broker*, o

⁹O protocolo MQTT fornece um método leve de envio de mensagens usando um modelo publish/subscribe.

¹⁰<https://mosquitto.org/>

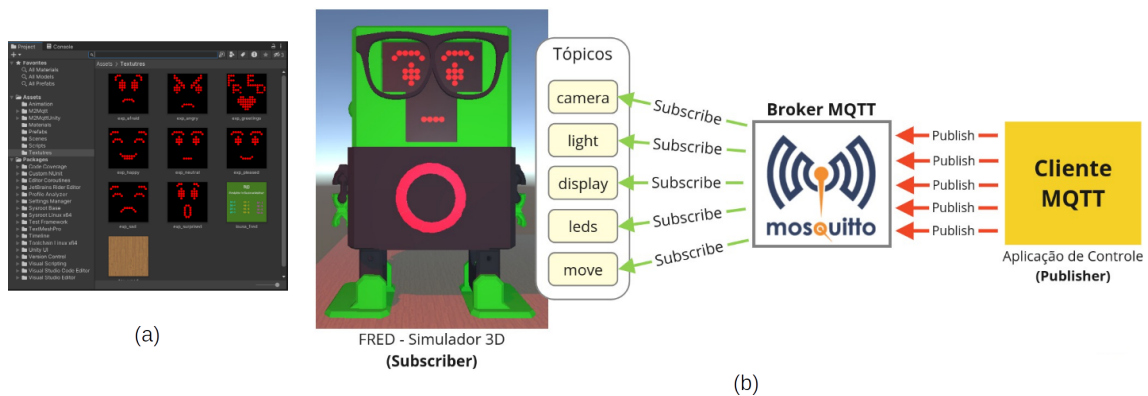


Figura 2. Texturas criadas para a representação das expressões faciais do robô no simulador (a) e o esquema de controle do simulador através do protocolo MQTT (b).

simulador assina (*subscribe*) os *tópicos* referentes às suas funcionalidades e pode receber *mensagens* nesses *tópicos*. Ao todo, estão definidos 5 *tópicos*: *light*, *camera*, *display*, *leds* e *move*. A partir da conexão com o *broker* e da assinatura dos *tópicos*, qualquer aplicação que implemente um cliente MQTT e que se conecte ao *broker*, publicando as *mensagens* adequadas nos *tópicos*, será capaz de controlar o robô virtual. A Figura 2(b) ilustra essa estrutura de comunicação/controle. Cada *tópico* e suas respectivas *mensagens* serão apresentados e descritos a seguir.

4.3.1. Seleção de Câmera

Através do *tópico camera*, é possível selecionar uma das 5 câmeras disponíveis no simulador. A seleção é feita *publicando-se* as *mensagens* 1, 2, 3, 4 ou 5. O modos de câmera são: (1) *distante*, (2) *perspectiva*, (3) *de frente*, (4) *de baixo* e (5) *de cima*. A Figura 3(a) mostra um *take* de uma cena com cada uma das 5 câmeras.

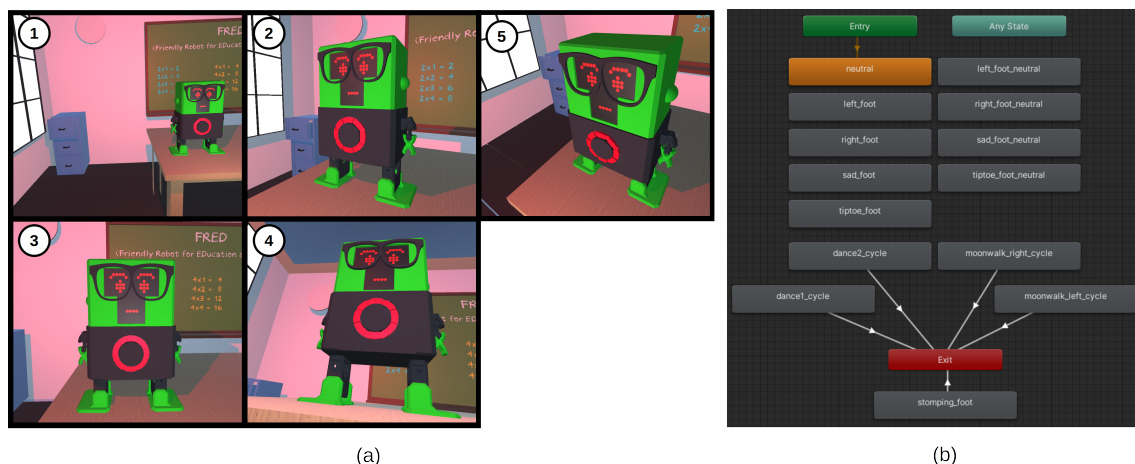


Figura 3. Os 5 modos de câmera disponíveis no software simulador (a) e os *clips* de animação (estados) definidos para o modelo 3D (b).

4.3.2. Controle de Iluminação

Um dos recursos do robô físico é a capacidade de controlar uma lâmpada inteligente. Isso dá ao robô a capacidade de tornar as interações mais atrativas e imersivas, principalmente para crianças. O simulador permite o controle da luz ambiente da cena. Para isso, basta que o cliente MQTT *publique* no *tópico light*. As *mensagens* responsáveis por controlar o estado e as cores da lâmpada são: *on*, *off*, *red*, *green*, *blue*, *cyan*, *white*, *magenta* e *yellow*.

4.3.3. Controle de Expressão Facial

Uma das maneiras mais eficazes de expressar emoções é através da expressão facial. Assim como a versão física do robô, o simulador é capaz de representar expressões faciais através do seu *display*. Como para a representação dos olhos e boca no robô físico são utilizadas 3 matrizes de LEDs 8x8, para o simulador, buscou-se criar uma textura que imitasse esse tipo de componente. Para selecionar a expressão a ser exibida no *display* do robô simulado, deve-se *publicar* no *tópico display*. O modelo 3D do robô pode representar 7 emoções que podem ser selecionadas utilizando-se as seguintes *mensagens*: *happy* (feliz), *angry* (raiva), *sad* (triste), *neutral* (neutra ou indiferente), *pleased* (satisfeito), *afraid* (medo) e *surprised* (surpreso). Além dessas emoções, o *display* do modelo 3D pode exibir uma imagem de saudação (*greetings*). Essas texturas foram apresentadas na Figura 2 (a).

4.3.4. Controle do Anel de LEDs RGB

O robô físico conta com elementos de comunicação verbal e não verbal. O anel de LEDs RGB, em seu tórax, é um elemento de comunicação não verbal. Esse elemento utiliza as cores para demonstrar emoção. Para controlar o estado e as cores do anel de LEDs basta *publicar*, no *tópico leds*, as seguintes *mensagens*: *off* (para desligar os LEDs) e as *strings* que representam as cores, *red*, *green*, *blue*, *cyan*, *white*, *magenta* e *yellow*.

4.3.5. Controle de Poses e Movimentos

Um outro elemento importante de comunicação não verbal é a expressão corporal. O robô físico pode usar o seu corpo para se expressar, utilizando poses e movimentos. Toda a animação do modelo 3D foi feita no software Blender e para sua criação foram utilizados 800 frames. Após a importação do modelo no Unity 3D, essa animação (800 frames), foi subdividida em *clips* menores que representam cada uma das 14 animações do modelo. A Figura 3(b) apresenta os *clips* de animação (estados do robô) definidos no componente *Animator* do Unity 3D.

Com o objetivo de validar a proposta do simulador 3D para a plataforma de robótica social, um vídeo de comparação foi gravado. O vídeo apresenta uma demonstração das funcionalidades do robô físico sendo executada lado a lado com o modelo 3D no simulador proposto neste trabalho. Para o teste, um *script* de controle do robô

foi implementado na linguagem Python utilizando-se a biblioteca *Eclipse Paho*¹¹ MQTT. O vídeo comparativo pode ser assistido neste link¹².

5. Conclusão

Os robôs sociais são dispositivos capazes de transformar a vida das pessoas. Isso é feito não através de algum tipo de assistência física mas através de estímulos durante o processo de interação com o usuário. Espera-se que esses dispositivos se tornem onipresentes com um aumento de seu uso em aplicações educacionais e também em saúde. Várias plataformas robóticas vêm sendo propostas, tanto por empresas privadas, que oferecem modelos de custo elevado e arquiteturas de hardware e software fechadas, quanto na literatura científica, onde existem diversas propostas de plataformas robóticas de código aberto e baixo custo. Apesar de algumas dessas propostas serem *open-source* e de baixo custo, nem sempre é possível ter um robô físico montado para testes e o aprendizado de como programá-lo. Este trabalho apresentou a proposta de um simulador 3D *open-source*, multiplataforma, para uma plataforma de robótica social de baixo custo, o FRED. O simulador é capaz de simular grande parte das funcionalidades do robô físico, como: o controle da luz ambiente, a capacidade de expressar emoções utilizando expressões faciais, o controle do anel de LEDs RGB e o uso de expressões corporais utilizando poses e animações (danças). O simulador proposto pode ser controlado utilizando-se o modelo *publish/subscribe* do protocolo de comunicação MQTT. Através dos *tópicos* e *mensagens*, que definem as funcionalidades do modelo 3D e os comandos de controle, respectivamente, é possível controlar o simulador a partir de qualquer aplicação que implemente um cliente MQTT. Para servir como uma avaliação inicial do proposta do simulador, um vídeo comparativo entre o robô físico e o modelo 3D simulado foi apresentado.

Como trabalhos futuros, pretende-se aumentar o número de animações no modelo simulado, oferecer mais opções de cenários para a interação com o robô 3D, aprimorar as animações do anel de LEDs RGB no tórax do robô, se aproximando das animações dos LEDs reais. Pretende-se também fazer testes de usabilidade do simulador 3D com usuários do robô físico.

Agradecimentos

Os autores agradecem o apoio recebido do Google Research, CAPES, CAPES PRINT, CNPq, INCT-MACC, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) 2015/24144-7 e FAPERJ.

Referências

- Ayala, A., Cruz, F., Campos, D., Rubio, R., Fernandes, B., and Dazeley, R. (2020). A comparison of humanoid robot simulators: A quantitative approach. In *2020 Joint IEEE 10th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)*, pages 1–6. IEEE.
- Carpin, S., Lewis, M., Wang, J., Balakirsky, S., and Scrapper, C. (2007). Usarsim: a robot simulator for research and education. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 1400–1405. IEEE.

¹¹<https://pypi.org/project/paho-mqtt/>

¹²<https://drive.google.com/file/d/1LBIPxCqFTqIROj1NVZhV1uMplWnjVo1V/view?usp=sharing>

- Cruz-Sandoval, D. and Favela, J. (2019). A Conversational Robot to Conduct Therapeutic Interventions for Dementia. *IEEE Pervasive Computing*, 18(2):10–19.
- de Lima, P. V. S., Bezerra, M. H. R., de Sousa Tavares, A. C., Júnior, J. R. F., Teixeira, J. M. X. N., Cajueiro, J. P. C., Melo, G. N., and Henriques, D. B. (2018). Improving early robotics education using a line-following robot simulator. In *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, pages 547–553. IEEE.
- Feil-Seifer, D. and Mataric, M. J. (2005). Defining socially assistive robotics. In *9th International Conference on Rehabilitation Robotics, 2005. ICORR 2005.*, pages 465–468. IEEE.
- Gena, C., Mattutino, C., Maltese, W., Piazza, G., and Rizzello, E. (2021). Nao_prm: an interactive and affective simulator of the nao robot. In *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, pages 727–734. IEEE.
- Gouaillier, D., Hugel, V., Blazevic, P., Kilner, C., Monceaux, J., Lafourcade, P., Marnier, B., Serre, J., and Maisonnier, B. (2009). Mechatronic design of nao humanoid. In *2009 IEEE international conference on robotics and automation*, pages 769–774. IEEE.
- Ishimura, T., Kato, T., Oda, K., and Ohashi, T. (2003). An open robot simulator environment. In *Robot Soccer World Cup*, pages 621–627. Springer.
- Johal, W., Castellano, G., Tanaka, F., and Okita, S. (2018). Robots for learning.
- Kumar, K. and Reel, P. S. (2011). Analysis of contemporary robotics simulators. In *2011 International Conference on Emerging Trends in Electrical and Computer Technology*, pages 661–665. IEEE.
- Kurniawan, O., Lee, N. T. S., Datta, S., Sockalingam, N., and Leong, P. K. (2018). Effectiveness of physical robot versus robot simulator in teaching introductory programming. In *2018 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, pages 486–493. IEEE.
- Mukhandi, M., Portugal, D., Pereira, S., and Couceiro, M. S. (2019). A novel solution for securing robot communications based on the mqtt protocol and ros. In *2019 IEEE/SICE International Symposium on System Integration (SII)*, pages 608–613. IEEE.
- Perico, D. H., Homem, T. P., Almeida, A. C., Silva, I. J., Vilão, C. O., Ferreira, V. N., and Bianchi, R. A. (2016). A robot simulator based on the cross architecture for the development of cognitive robotics. In *2016 XIII Latin American robotics symposium and IV Brazilian robotics symposium (LARS/SBR)*, pages 317–322. IEEE.
- Pot, E., Monceaux, J., Gelin, R., and Maisonnier, B. (2009). Choregraphe: a graphical tool for humanoid robot programming. In *RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication*, pages 46–51. IEEE.
- Rocha, M., Cruz-Sandoval, D., Favela, J., and Muchaluat-Saade, D. C. (2022a). Evasim: a software simulator for the eva open-source robotics platform. In *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 714–721. IEEE.

- Rocha, M. and Muchaluat-Saade, D. C. (2023). Friendly robot for education and healthcare: Fred. In *Proceedings of LIQUE - Life Improvement in Quality by Ubiquitous Experiences Workshop, together with ACM IMX 2023*.
- Rocha, M., Valentim, P., Barreto, F., Mitjans, A., Cruz-Sandoval, D., Favela, J., and Muchaluat-Saade, D. (2022b). Towards enhancing the multimodal interaction of a social robot to assist children with autism in emotion regulation. In *International Conference on Pervasive Computing Technologies for Healthcare*, pages 398–415. Springer.
- Santos, L., Geminiani, A., Schydlo, P., Olivieri, I., Santos-Victor, J., and Pedrocchi, A. (2021). Design of a robotic coach for motor, social and cognitive skills training toward applications with asd children. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 29:1223–1232.
- Truong, L. B., Kang, D.-B., Ji, S.-H., and Jeong, G.-M. (2011). A control algorithm for robot simulator in educational platform. In *ICTC 2011*, pages 735–736. IEEE.