

VIRTUALIZAÇÃO DO AMBIENTE DE TRABALHO PARA TREINAMENTOS DE REDES NEURAIS

JOAB TAVARES FAGUDES¹; EDUARDO VARGAS ZUMMACH²

FELIPE DIAS LOPES³; FELIPE MACHADO⁴

MARCELO LEMOS ROSSI⁵

¹Universidade Federal de Pelotas – (joabtavaresf02@gmail.com)

²Universidade Federal de Pelotas – (eduardo.zummach@hotmail.com)

³Universidade Federal de Pelotas – (felopes_7@hotmail.com)

⁴Universidade Federal de Pelotas – (felyppemachado@gmail.com)

⁵Universidade Federal de Pelotas – (marcelo.rossi@ufpel.edu.br)

1. INTRODUÇÃO

Cada vez mais a tecnologia tem ficado mais acessível e aprender e se envolver com ela começa a fazer parte do cotidiano. Essa aproximação e envolvimento também tem atraído deficientes visuais ao aprendizado de eletrônica, um exemplo disso é o primeiro caso de uma aluna deficiente visual a defender uma tese de mestrado no curso de Engenharia Elétrica pela Universidade Tecnológica Federal do Paraná, (GARCIA, 2016), e exemplos assim tem levado ao desenvolvimento de diversos trabalhos como, o desenvolvimento de blocos de circuitos t, (THE AWESOME FOUNDATION, 2020), Redefinição de Símbolos utilizando a eletrônica, (CASS, 2019). Trabalhos estes que contem intenções de criar recursos para tornar mais casos como o da estudante viáveis e mais comuns.

Esta pesquisa utiliza-se da Visão computacional, um recurso que é uma área da engenharia que pode auxiliar na identificação de objetos, faces, materiais, formas e traduzi-los para uma linguagem computacional, onde ocorreria uma “tradução” para descrições via áudio, formação de imagens, moldes impressos, detecção e descrição do tipo de objeto.

A visão computacional necessita de um treinamento para o reconhecimento do formato das portas lógicas e neste processo de treinamento é importante que este seja realizado em um ambiente real pois, por se tratar de visão podem sofrer interferências de sombras luzes e outros. Devido as situações atuais da pandemia de COVID-19, e a suspensão das atividades laboratoriais na Universidade Federal de Pelotas, conforme o Memorando Circular (0903009 – SEI_UFPel), estas ações não seriam viáveis. Afim de contornar esse impasse e uma maneira de realizar esses treinamentos, ainda conseguindo as variações de uma situação real de laboratório, como, sombras, iluminações, artefatos, foi decido pela criação de um ambiente virtual simulado, utilizando o motor Unity 3D (UNITY, 2020).

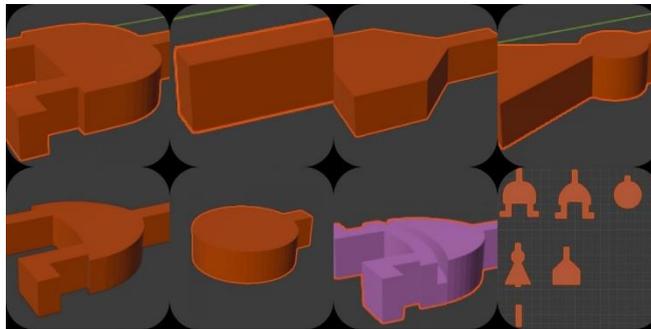
A escolha do Unity3D se deve ao fato de que os jogos de vídeo game estão cada vez mais realistas, principalmente com a utilização de tecnologias como RayTracing (NVIDIA, 2020), este recurso é muito utilizado para trazer realismo aos jogos, devido a seu traçado de raios de luz e seus cálculos de sombra e iluminação em renderização de imagens tridimensionais. Isso torna motores de jogos, como a Unity3D, muito utilizados para treinamentos de redes neurais e para simulação de ambientes reais como no projeto “Training a performant object detection ML model on synthetic data using Unity Perception tools”, (JHANG et al, 2020)

Utilizando a Unity3D, foi criado um ambiente virtual para simular situações reais, e ainda permitir, de forma fácil a manipulação de iluminação, sombras, artefatos, posições das peças. Isso permitiu a criação de um grande banco de dados, contando com inúmeras situações e variações, que foram registradas em imagens e formaram um dataset para o treinamento de uma Rede Neural. Estas imagens extraídas do ambiente virtual, também passaram por processos de segmentação de imagem, utilizando Python (PYTHON, 2020) e OpenCV (OPENCV, 2020). Para a criação de peças modeladas em 3D foi utilizada a ferramenta Blender (BLENDER, 2020)

2. METODOLOGIA

Inicialmente, foi realizado o processo de modelagem das peças em 3D utilizando a ferramenta Blender em sua versão 2.8. As Figuras 1- 8 apresentam o resultado da modelagem das peças feitas na ferramenta Blender.

Figura 1 – Peças Importadas o Blender e Extrudadas em 3D (BLENDER).



Após a realização desse processo de modelagem 3D, as peças foram exportadas no formato .FBX, para melhor importação no motor da Unity 3D. Logo após a importação das peças para o motor, foi criado uma base de apoio para as peças, utilizando um modelo de posicionamento de jogos de tabuleiros, que não permitem a sobreposição de peças e determinam áreas em que podem ser colocadas as peças, podendo apenas rotacionar as peças nas “bairas” pré-estabelecidas.

Com o cenário e as peças prontas, ainda faltava as variações de iluminação e de posições das peças, afim de resolver essas pendencias foi utilizado o *cinemachine*, que se trata de um Package do motor, aonde é possível alterar a posição de uma camera, junto a essa ferramenta foram adicionadas a *Câmera Dolly*, e um *Slider*, que permitem escolher a posição da câmera por uma barra de rolagem entre 0 e 100%. Foi aplicado variações de um Spot de Luz para realizar modificações em sombras e iluminações. Para facilitar o acesso o ambiente virtual foi disponibilizado de forma online por meio do WebGL (MOZILLA,2020).

Figura 10 – Exemplo da Barra de Rolagem de posição da câmera e posicionamento das peças.



Futuramente esperamos que este simulador e seus recursos sirvam para treinar, outros modelos além da YOLOv3 e outros algoritmos de segmentação de imagem para reconhecimento de formas, como uma maneira de obter melhores resultados e melhores dados, podendo até posteriormente ser utilizado outras peças, símbolos, figuras e imagens para treinar e identificar de acordo com a necessidade dos estudantes.

5. REFERÊNCIAS BIBLIOGRÁFICAS

Home of the Blender project - Free and Open 3D Creation Software, **BLENDER**. **blender.org** 2020. Disponível em: <<https://www.blender.org/>> ; Acessado em 23 de set. 2020.

GARCIA, Vera. **Paranaense será primeira deficiente visual do brasil a defender tese de mestrado em Engenharia Elétrica**. DeficienteCiente, 26 de jan. de 2016. Disponível em: <<https://www.deficienteciente.com.br/paranaense-sera-primeira-deficiente-visual-do-brasil-a-defender-tese-de-mestrado-em-engenharia-eletrica.html>> Acessado em 23 de set. de 2020.

REDMON, J.; FARHADI, A. **YOLOv3: An Incremental Improvement**. arXiv, [S.l.], 2018.

Unity: Plataforma de desenvolvimento em tempo real do Unity. **UNITY TECHNOLOGIES**, 2020. Disponível em: <<https://unity.com/pt>>. Acessado em 23 de set. 2020.

OPENCV. Equipe OpenCV, c2020. Disponível em: <<https://opencv.org/>> Acessado em 23 de set. 2020.

COCO COMMON OBJECTS IN CONTEXT. **COCO**, c2015. Disponível em: <<https://cocodataset.org/#home>>. Acessado em 23 de Set. De 2020.

Alexander Mordvintsev & Abid K. Revision, **ORIENTED FAST and ROTATED BRIED, ORB**, c2013. Disponível em: <https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_feature2d/py_orb/py_orb.html>. Acessado em 23 de Set. De 2020.

RTX.IT'S ON. **NVIDIA Corporation**, c2020. Disponível em: <<https://www.nvidia.com/pt-br/geforce/rtx/>>. Acessado em 23 de Set. de 2020.

JHANG et Al. Training a performant object detection ML model on synthetic data using Unity Perception tools. **Unity3D** c2020. Disponível em: <<https://blogs.unity3d.com/pt/2020/09/17/training-a-performant-object-detection-ml-model-on-synthetic-data-using-unity-perception-tools/>>. Acessado em 23 de Set. De 2020.

Teaching Electronics to Visually Impaired Students. **THE AWESOME FOUNDATION**, c2020. Disponível em: <<https://www.awesomefoundation.org/pt/projects/50011-teaching-electronics-to-visually-impaired-students>>. Acessado em 23 de Set. De 2020.

Electronic schematics for blind makers, **Cass**, c2019. Disponível em: <<https://ieeexplore.ieee.org/document/8747303>>. Acessado em 23 de Set. De 2020.

Tecnologia Web para desenvolvedores, WebGL. **Mozilla**, 2020. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/API/WebGL_API>. Acessado em 23 de Set. De 2020.