

**V SIINTEC**INTERNATIONAL SYMPOSIUM ON INNOVATION AND TECHNOLOGY  
SIMPÓSIO INTERNACIONAL DE INOVAÇÃO E TECNOLOGIA

CIRCULAR ECONOMY

ECONOMIA CIRCULAR

## **SIMULAÇÕES ROBÓTICAS EM PLATAFORMAS DE DESENVOLVIMENTO DE JOGOS**

Carlos Alberto de Araújo Pereira Filho<sup>1</sup>; Marco Antônio dos Reis<sup>2</sup>;

<sup>1</sup> Centro Universitário SENAI CIMATEC; Salvador/Bahia; carlospereira.179@gmail.com

<sup>2</sup> Centro Universitário SENAI CIMATEC; Salvador/Bahia

**Resumo:** Com o crescimento da robótica principalmente dentro das indústrias, a demanda por novos projetos nessa área segue a mesma tendência. A simulação é uma etapa fundamental do desenvolvimento de qualquer projeto robótico, em projetos onde a visão computacional se faz presente, uma simulação que represente da melhor maneira possível o mundo real e ainda mais importante. Dessa forma, é feita uma comparação entre os simuladores V-REP e Gazebo com a Unreal Engine 4 para verificar as vantagens de utilizar uma ferramenta de desenvolvimento de jogos nas simulações.

**Palavras-Chave:** Jogos; Robótica; Simulação;

## **ROBOTICS SIMULATIONS ON GAME DEVELOPMENT PLATFORMS**

**Abstract:** With the robotics growing in the industries, the demand for new projects in this area follows the same trend. Simulation is a fundamental step in the development of any robotic project, in projects where computer vision is present, a simulation that best represents the real world is even more important. This compares V-REP and Gazebo simulators with Unreal Engine 4 to see the advantages of using a game development tool in simulations.

**Keywords:** Games; Robotics; Simulation;



## 1. INTRODUÇÃO

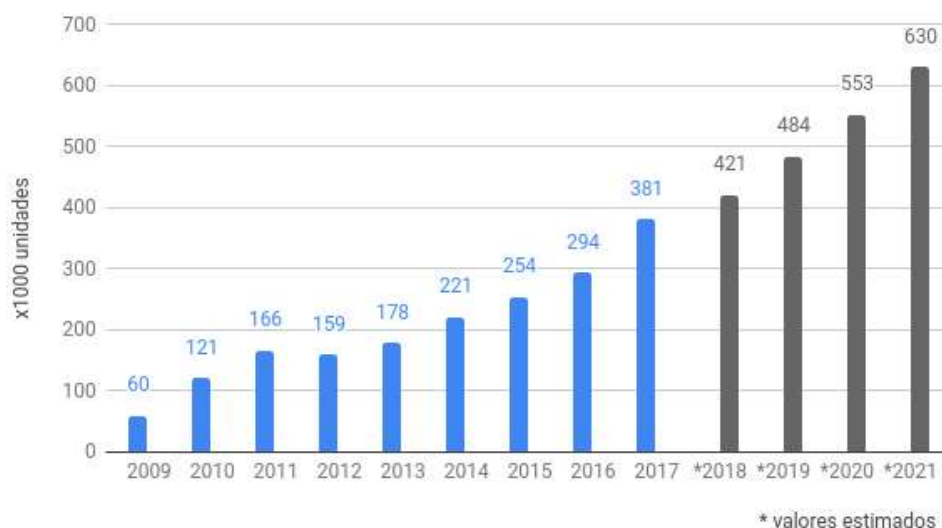
Durante esta última década, o espaço ocupado pela robótica nas indústrias cresceu de maneira significativa[1]. Diversas atividades foram automatizadas utilizando principalmente braços robóticos, ou manipuladores, no lugar de trabalhadores. Esse movimento vem se tornando crucial para uma parcela do desenvolvimento das indústrias de transformação.

O maior exemplo de indústria que adotou largamente a robótica é a automotiva. Diversas atividades nas suas linhas de produção são realizadas exclusivamente por robôs[1]. Em outra grande parte, tem pessoas realizando montagem de componentes, porém sendo auxiliadas por equipamentos robóticos. Nesses casos, os robôs tem o objetivo de melhorar a ergonomia no ambiente de trabalho.

Com o aumento crescente da demanda de dispositivos eletrônicos, nesta indústria também há uma alta implementação de processos automatizados com robôs. Dessa forma, as fábricas conseguem garantir maior qualidade nos seus produtos e redução dos custos, já que robôs conseguem manipular pequenas partes com muita precisão e velocidade.

O Industrial Robot Report publicado pela IFR (*International Federation of Robotics*) em 2018 mostra que o número de robôs industriais comercializados em 2017 foi 381.000 unidades, quase 30% a mais do que o ano anterior. A projeção da IFR para os anos seguintes segue a mesma tendência, uma média de 14% ao ano até 2021.

Figura 1 Robôs industriais comercializados no ano. (\*Adaptado do IFR)



Quando um sistema é complexo, com o seu funcionamento sendo descrito por diversas variáveis, executar a fabricação e/ou montagem desse sistema sem ter sido realizada a sua validação é arriscado. Esse processo de análise prévia com o intuito de verificar o funcionamento é chamado de simulação. A simulação consiste em recriar determinado sistema ou componente em um ambiente computacional, onde são fornecidas entradas e analisadas as saídas.



A simulação traz diversos benefícios para o desenvolvimento do projeto. Esse processo permite, durante o *design*, validar diversas alternativas, proporcionando a tomada das melhores decisões sem a necessidade de comprar componentes ou construir um número desnecessário de protótipos. Nas simulações também é possível realizar testes extremos ao sistema, verificando assim as condições limite de operação de uma maneira completamente segura.

### 1.1. Objetivos

Este trabalho tem como objetivo demonstrar as principais vantagens da utilização de softwares como Unreal Engine 4 na construção de ambientes grandes e bastante detalhados, pontos de grande relevância para o desenvolvimento de simulações de alguns tipos de robôs, em comparação com simuladores bastante populares no meio robótico como Gazebo e V-REP. Assim como essas comparações, é necessário verificar como ou através de quais ferramentas os dados desta simulação são disponibilizados no ROS (*Robot Operating System*) para pacotes de outras funcionalidades do robô.

### 1.2 Justificativa

Com o objetivo de testar o desempenho de componentes e validar funcionalidades, a simulação é uma etapa fundamental do desenvolvimento de um projeto robótico. Essa atividade proporciona um ambiente completamente seguro para os testes e traz economia de recursos, já que ainda não se faz necessária a construção de um protótipo[2].

É importante que o ambiente simulado seja fiel ao qual o robô, ou componente do robô, será submetido. As características dessa fidelidade irão variar com o experimento que será executado durante a simulação, como características físicas de gravidade e arrasto por exemplo ou visuais, para simulações de câmeras. Dessa forma, visto o alto desenvolvimento da indústria de jogos eletrônicos e a fidelidade que os gráficos presentes nesses jogos tem com a realidade, é interessante utilizar essa mesma tecnologia para aperfeiçoar o desenvolvimento da robótica.

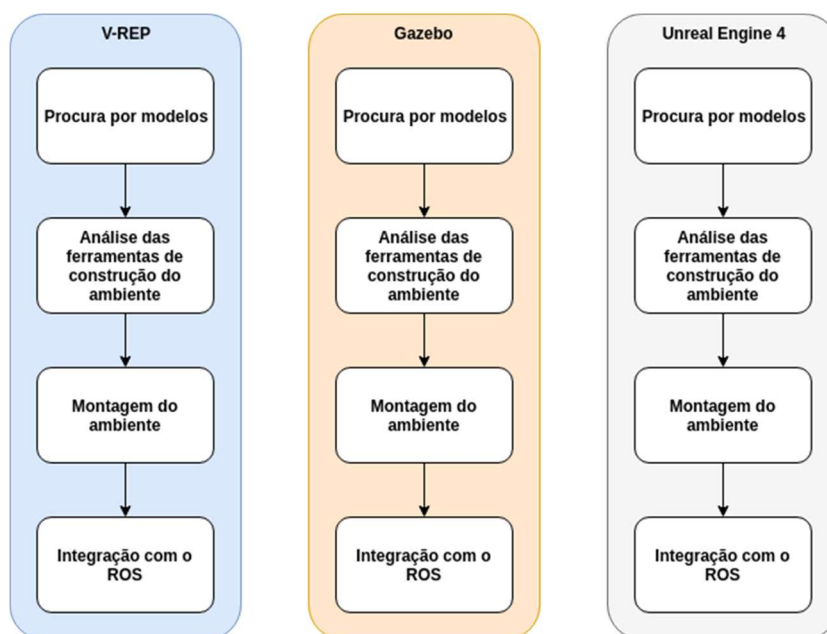
Outro grande incentivo à utilização de plataformas de simulação que trazem uma alta qualidade gráfica é o crescimento da inteligência artificial e utilização de *machine learning* nos projetos. Com esse crescimento, também há a necessidade de mais e maiores *datasets*[3]. Projetos de carros autônomos que utilizam câmeras para localizar obstáculos, por exemplo, necessitam de uma quantidade enorme de imagens de situações reais das ruas. Um ambiente de simulação, com a qualidade gráfica similar a um jogo recém lançado facilita muito a produção desses *datasets*, visto que dessa maneira não é necessário um veículo com todo o aparato de câmeras circulando em diversas condições.



## 2. METODOLOGIA

A fim de verificar as vantagens de utilizar a Unreal Engine 4 para simulações robóticas de detecção de obstáculos e criação de *datasets* deve-se, a princípio, comparar a qualidade gráfica das diferentes plataformas. Dessa forma, são montados ambiente com as mesmas características básicas. Os pontos principais de comparação entre essas simulações são: A quantidade de modelos já integrados à plataforma, a disponibilidade de itens em suportes oficiais e na comunidade e a facilidade de importação de modelos desenvolvidos pelo usuário.

Figura 2 Etapas para cada plataforma



Para a construção do ambiente é preciso buscar fontes de modelos que compõem a cena desejada, itens como objetos de decoração, plantas, construções, veículos e etc. Verifica-se a disponibilidade desses modelos dentro do próprio software e em fóruns da comunidade de usuários. No caso de não estar disponível em nenhuma dessas fontes o objeto necessário, é avaliada a possibilidade de importar para a simulação um componente customizado pelo usuário em softwares de modelagem 3D como o Blender e SolidWorks.

Relevo, cores e texturas são outros pontos importantes para avaliar o realismo da simulação. É feita uma análise, se há e como funcionam, ferramentas nas plataformas para o trabalho com esses recursos.

Depois de finalizado o ambiente, é preciso investigar como os dados adquiridos durante a simulação serão utilizados. Visto que o ROS é um framework largamente utilizado no desenvolvimento de soluções robóticas, esses dados são disponibilizados para uso através dessa solução. Como as plataformas de simulação realizam a integração com o ROS é outro importante ponto de comparação.

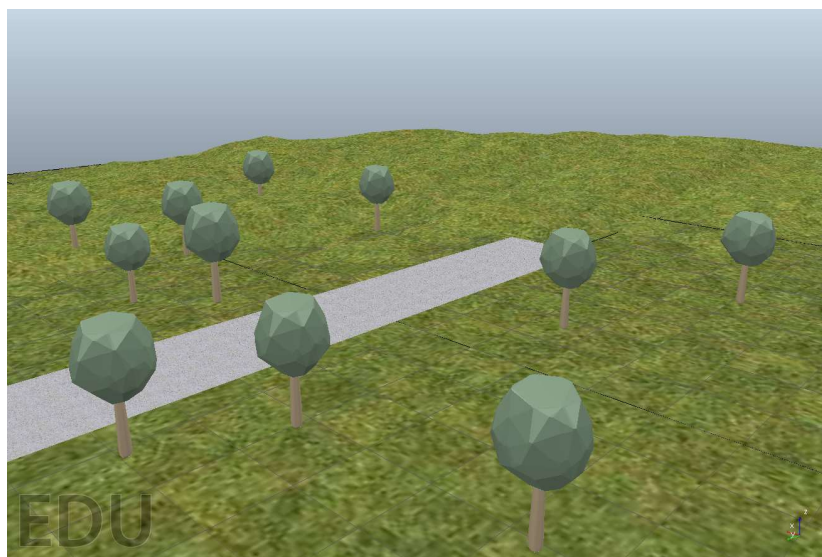


### 3. RESULTADOS E DISCUSSÃO

O ambiente escolhido para a comparação entre os simuladores foi um gramado com árvores e um caminho de concreto. Essa cena foi construída primeiro no V-REP. Nesse simulador há uma grande biblioteca de itens prontos para serem utilizados mas são itens com pouco detalhe. Existe a possibilidade de importar modelos 3D nos formatos OBJ, DXF, STL, Collada e URDF. É recomendado pelos desenvolvedores que na cena os modelos não ultrapassem 20.000 triângulos, pois a simulação perde sua velocidade, por causa disso, as árvores utilizadas são próprias do V-REP.

Há duas formas de modificar o relevo, com a ferramenta *terrain* e importando um *heightfield*. Na ferramenta *terrain* é possível ajustar a amplitude, posição e frequência das elevações. A outra alternativa é importar uma imagem onde as suas cores representam a altura de cada região. No V-REP não é possível criar mapas maiores que 100m x 100m. Nessa simulação foi utilizada a ferramenta *terrain* onde foi importada uma textura de grama.

Figura 3- Mapa criado no V-REP



O V-REP tem na sua API (*application program interface*) o ROS Interface que faz com que o V-REP atue como um nó do ROS, se comunicando por *services*, *publishers* e *subscribers*.

O Gazebo também tem um leque de modelos que estão disponíveis para compor a cena, são modelos mais detalhados do que o V-REP, porém não há árvores. É possível importar objetos nos formatos OBJ e Collada, não havendo limitação de complexidade do modelo.

A maneira de modificar o relevo do ambiente no Gazebo é com um *heightmap*. Foi feito um arquivo SDF com as informações de tamanho do mapa, altura do *heightmap* e o caminho de uma imagem em preto e branco, sem o canal alfa, que representa o perfil do relevo, onde os pontos mais claros têm maior altitude.





Figura 4 Mapa criado no Gazebo



No Gazebo é utilizado um conjunto de pacotes do ROS chamado de `gazebo_ros_pkgs` que proporciona uma interface entre o simulador e o *framework*.

Já integradas naturalmente na Unreal Engine, há apenas formas geométricas simples. Em contrapartida, existem diversos materiais que podem ser aplicados a essas formas ou a objetos importados para o projeto. Nos fóruns da comunidade de usuários são encontrados com facilidade outros modelos para uso na plataforma, as árvores utilizadas nesta simulação foram provenientes de um pacote, com 16 espécies e 4 variações de cada espécie, que estava disponível para download no fórum oficial da desenvolvedora.

Figura 5 Mapa criado na Unreal Engine 4



É possível importar arquivos 3D nos formatos FBX e OBJ e aplicar qualquer material existente na plataforma. Materiais também são facilmente encontrados nos fóruns e no Unreal Engine Marketplace, plataforma onde também há opções pagas e gratuitas de objetos e mapas completos. O relevo do mapa foi feito utilizando a



ferramenta *Landscape* onde é utilizado um “pincel” que eleva o terreno enquanto o usuário o mantém naquela região.

Como não foi desenvolvido para ser um simulador robótico, a Unreal Engine 4 não tem interface nativa com o ROS, mas há na comunidade de usuários *plugins* como o ROSIntegration que dão a possibilidade de transferir informações da Unreal Engine 4 para o ROS através de tópicos e serviços.

#### **4. CONCLUSÃO (ARIAL 12)**

Durante esse trabalho foram feitas comparações entre simuladores de robótica comumente utilizados e a Unreal Engine 4. Com essas comparações é possível constatar as vantagens de se utilizar uma plataforma de desenvolvimento de jogos em simulações onde o nível de detalhe e semelhança com o mundo real é importante.

A Unreal Engine 4 se destaca principalmente pela qualidade gráfica do ambiente, tendo como responsáveis por tamanho realismo a flexibilidade de materiais que podem ser aplicados à formas, pela qualidade da iluminação, sombras e reflexos. Os tamanhos da comunidade de usuários e do suporte por parte da Epic Games, desenvolvedora da plataforma, são grandes auxiliares na utilização dessa solução.

#### **5. REFERÊNCIAS (ARIAL 12)**

[1] International Federation of Robotics. Global industrial robot sales doubled over the past five years. 2018.

[2] Robot simulation software: The value for manufacturers. <https://www.robotics.org/blog-article.cfm/Robot-Simulation-Software-The-Valuefor-Manufacturers/57>. Acessado em: 2019-07-30.

[3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. arXiv preprint arXiv:1711.03938, 2017.