

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**Estudo de caso em aprendizado por reforço
no universo Duckietown**

Cainã Setti Galante

MONOGRAFIA FINAL

MAC 499 — TRABALHO DE
FORMATURA SUPERVISIONADO

Supervisor: Prof. Denis Deratani Mauá
Cossupervisora: Prof^a. Nina S. T. Hirata

São Paulo
2022

*O conteúdo deste trabalho é publicado sob a licença CC BY 4.0
(Creative Commons Attribution 4.0 International License)*

Resumo

Cainã Setti Galante. **Estudo de caso em aprendizado por reforço no universo Duckietown**. Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2022.

Neste trabalho se estuda o uso de um modelo de aprendizado por reforço para a criação e desenvolvimento de veículos completamente autônomos em ambiente urbano. Para isso se utiliza o simulador Duckietown para definir o ambiente de treino, que consiste em um simples circuito no qual o veículo deve percorrer sem cometer infrações. Para a aprendizagem é utilizado o Deep Q-Network (DQN), que combina redes neurais e deep learning com classico método de aprendizagem por reforço Q-learning. Com as punições e recompensas utilizadas para treinar o modelo é observado que nem sempre o veículo aprende o esperado, ao encontrar diferentes formas de trapacear o sistema de pontuação.

Palavras-chave: Inteligência Artificial. Duckietown. Aprendizado de Máquina.

Abstract

Cainã Setti Galante. **Reinforced learning on Duckietown**. Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2022.

This work is the study of the use of a reinforced learning model for the creation and development of autonomous vehicles in the urban environment. It uses the Duckietown simulator to define the training environment, which is made of a simple circuit that the car has to run without making any infractions. To learn, the agent uses the Deep Q-Network algorithm, which is a combination of neural networks and deep learning with the reinforced learning method classic Q-learning. With the punishments and rewards used to train the model, it is observed that the vehicle does not always learn what is expected when it finds different ways to cheat the scoring system.

Keywords: Artificial Intelligence. Duckietown. Machine Learning.

Lista de abreviaturas

DQN	<i>Deep-Q Networks</i>
AI	Inteligência artificial (<i>Artificial intelligence</i>)
RL	Aprendizado por reforço (<i>Reinforcement learning</i>)
GPU	Unidade de processamento gráfico (<i>Graphics processing unit</i>)
CPU	Unidade central de processamento (<i>Central processing unit</i>)
MIT	Instituto de Tecnologia de Massachusetts (<i>Massachusetts Institute of Technology</i>)
ABNT	Associação Brasileira de Normas Técnicas
URL	Localizador Uniforme de Recursos (<i>Uniform Resource Locator</i>)
IME	Instituto de Matemática e Estatística
USP	Universidade de São Paulo

Lista de figuras

1.1	Exemplo de Duckietown (figura retidada de (<i>Guide for Researches - Duckietown 2022</i>)).	3
1.2	Imagem do Gym Duckietown (imagem retidada de (CHEVALIER-BOISVERT <i>et al.</i> , 2018)).	4
3.1	Informações do sistema.	8
4.1	Gráfico da evolução durante sessão de treino.	9

Lista de tabelas

Lista de programas

Sumário

Introdução	1
1 Duckietown	3
1.1 O que é Duckietown?	3
1.2 Gym-Duckietown	4
2 <i>Deep Q-Network</i>	5
2.1 Q-learning	5
3 Metodologia	7
3.1 Critérios de treinamento	7
4 Conclusão	9
 Apêndices	
A Especificações de hardware	11
 Referências	27
Índice remissivo	29

Introdução

“*Can machines think?*” (máquinas conseguem pensar?), uma simples pergunta proposta por Alan M. Turing em seu *paper Computing Machinery and Intelligence* (TURING, 1950) mas com repercussões que iriam alterar o rumo da humanidade para sempre. Com a revolução industrial, os rápidos avanços da ciência permitiram que ideias antes presas na ficção e nos mitos entrassem nas vidas cotidianas das pessoas e, com o passar do tempo, é se criado no subconsciente humano a ideia de um dia ser possível construir objetos capazes de pensar. Essa ideia começa a tomar forma concreta com a publicação do livro *The Wonderful Wizard of Oz* (BAUM e DENSLOW, 1996), no qual um homem feito de lata é capaz de se mover e pensar como uma pessoa normal, e alguns anos depois com estreia a peça “*R.U.R.: Robôs Universais de Rossum*” do escritor tcheco Karel Čapek o termo ‘robô’ é cunhado, o qual permite que a imaginação humana junte às mais diversas concepções de inteligência artificial e criação de máquinas, antes referidas apenas como autômatos.

Essa ideia influenciou autores de ficção científica por todo o mundo, porém apenas em 1950 alguém para para refletir se é realmente possível se construir uma máquina capaz de pensar. Em seu trabalho, o matemático britânico Alan M. Turing propõe um teste, o qual ele se refere como “The Imitation Game” (TURING, 1950) ou em português “O jogo da imitação”, no qual um indivíduo, referido como interrogador, deve identificar através de uma série de perguntas e respostas algum “rótulo” de dois outros indivíduos enquanto eles tentam fazer com que o interrogador erre na identificação. Após a definição do jogo, ele propõe o experimento mental em que uma máquina participa do jogo no lugar de um dos interrogados e questiona se o interrogador irá errar quem é o humano e quem é a máquina na mesma proporção que ele era algo como quem é o homem e quem é a mulher. Essa reformulação transforma a pergunta na questão de se é possível fazer com que uma máquina (computador) e com um dado conjunto de instruções (programa) seja capaz de imitar de forma indistinguível o comportamento humano.

Essa nova pergunta despertou a curiosidade de inúmeros cientistas pelo mundo e levou, não somente, à criação de programas de processamento de linguagem natural, como ELIZA (WEIZENBAUM, 1966), mas também, inúmeras pesquisas deram origem à inteligência artificial e, mais importante para esse trabalho, ao aprendizado de máquina. O aprendizado de máquina, diferente das demais inteligências artificiais, se caracteriza pela criação de sistemas computacionais capazes de aprender e evoluir ao usar de modelos estatísticos para convergir para uma aproximação do modelo matemático capaz de resolver o problema dado, contudo tais capacidades vem através de custosos processos de treinamento além uma grande quantidade de informações sobre o problema a ser resolvido. Assim, seus primeiros sucessos ficaram presos ao laboratório pois seu custos operacionais inviabilizam

o uso prático, como foi o caso do trabalho Leonard Uhr e Charles Vossler (UHR e VOSSLER, 1961), um dos primeiros sucessos em desenvolver um programa capaz de reconhecer letras a partir de imagens, e foi necessário mais alguns anos de evoluções tecnológicas para que seus resultados fossem aplicados no cotidiano.

O aumento do poder computacional torna possível o desenvolvimento das metodologias de aprendizado por reforço que se baseiam em treinar o agente em um ambiente simulado. Em suma, nessas técnicas o agente aprende a maximizar sua pontuação por processos de tentativa e erro, os quais permitem uma exploração sistemática do ambiente simulado o que leva o agente a descobrir a solução do problema uma vez sua pontuação em cada iteração de treino é baseada em quão bem é o seu desempenho em resolver o problema. Contudo, essas técnicas demandam um largo poder computacional, visto que o espaço de ações possíveis para cada estado é infinitamente maior que o conjunto de ações corretas e dessa forma acelerar o treinamento consiste principalmente em desenvolver maneiras de eliminar mais rapidamente caminhos que não são capazes de melhorar o resultado atual sem que haja perda em comparação com fazer uma exploração completa do ambiente e aumentar a paralelização, de forma geral, do processo de treinamento para maximizar a utilização dos recursos computacionais disponíveis.

Neste trabalho é estudado o uso de um modelo de aprendizado por reforço para a criação e desenvolvimento de veículos completamente autônomos em ambiente urbano. Para isso o simulador utilizado para definir e manipular o ambiente é o Duckietown o qual é utilizado na disciplina MAC0318 - Introdução à Programação de Robôs Móveis, com ele é simulado um veículo simples que deve ser capaz de percorrer um circuito na cidade dos patos, que será discutido no capítulo 1. Além disso, o método de aprendizagem por reforço utilizado é o Deep Q-Network (DQN) que combina redes neurais com método Q-learning que será discutido no capítulo 2

Capítulo 1

Duckietown

1.1 O que é Duckietown?

Duckietown é um modelo de ensino de inteligência artificial com foco em veículos autônomos. Esse modelo, cujo desenvolvimento se deu em 2016 pelo MIT para ser usado na disciplina de mesmo nome, é uma simplificação do tráfego urbano de veículos e serve de porta de entrada para as áreas da robótica e da automação (*Duckietown Foundation 2022*). Apesar de tudo isso o Duckietown é bem robusto, como pode ser visto na figura 1.1, ao contar com ruas de mãos duplas, intersecções, calçadas, obstáculos e pedestres além de ser flexível para permitir o aumento gradual da complexidade do problema e a criação de inúmeros cenários a serem estudados e resolvidos.



Figura 1.1: Exemplo de Duckietown (figura retida de de (*Guide for Researches - Duckietown 2022*)).

Ademais, o desenvolvimento do Duckietown teve como um dos principais objetivos que ele fosse acessível com o intuito de que o modelo viesse a ser utilizado por instituições de ensino de todo o planeta. O veículo ou, como batizado, Duckiebot foi projetado para não só ser a plataforma mais simples e barata com a qual é possível se ensinar uma disciplina avançada de autonomia, como também, é completamente open source. Logo, nos anos seguintes o Duckietown se expandiu rapidamente e em 2018 foi fundada a *Duckietown*

Foundation, uma organização sem fins lucrativos, com a missão de tornar a robótica e inteligência artificial acessíveis e inclusivas pelo mundo.

1.2 Gym-Duckietown

Este trabalho utiliza o simulador Gym-Duckietown(CHEVALIER-BOISVERT *et al.*, 2018) para o treino e execução do veículo autônomo. Esse simulador do universo de Duckietown teve seu início em 2018, dentro do ecossistema Gym desenvolvido pela OpenAI(BROCKMAN *et al.*, 2016), como parte do trabalho feito pelo *Mila - Quebec AI Institute* e com o passar do tempo o projeto evoluiu para um simulador de direção autônoma completamente funcional, que pode ser usado para treinar e testar sistemas de aprendizado de máquina, além dos algoritmos clássicos da robótica como árvores de decisão ou veículos de *Braitenberg*. Por último, o Gym-Duckietown possui diversas ferramentas para recriar complicações reais, como distorção de câmera, além de aleatorização de domínio para evitar o *overfit* do modelo ao simulador.



Figura 1.2: Imagem do Gym Duckietown (imagem retidada de (CHEVALIER-BOISVERT *et al.*, 2018)).

Capítulo 2

Deep Q-Network

Deep Q-Network ou DQN é um algoritmo desenvolvido em 2015 pela DeepMind para aprendizado por reforço (Mnih *et al.*, 2015). O DQN é um dos primeiros algoritmos a ter sucesso em adicionar *deep neural network* a métodos de aprendizagem por reforço, visto que, ele é uma variação do *Q-Learning*, um clássico algoritmo de aprendizado por reforço, cujos *Q-Values* são, ao contrário do método clássico que armazena os valores em uma tabela, aproximados por uma *deep network*. As tentativas anteriores de criar um algoritmo que une o *deep learning* e o aprendizado por reforço falharam em virtude dos métodos de *deep learning* a sofrer *overfitting* o que deixa o sistema instável, contudo a equipe de pesquisadores do DeepMind foi capaz de resolver esse problema ao remover a correlação da sequência de observação com a aleatorização dos dados e ao reduzir a correlação com o valor alvo com a atualização deles de forma periódica.

2.1 Q-learning

Q-learning é um modelo de inteligência artificial desenvolvido para se espelhar no processo com o qual os animais aprendem (WATKINS, 1989). Dessa forma ele aprende direto das interações com o ambiente ao invés de um modelo do ambiente, como resultado o agente não precisa de um modelo desenvolvido a priori e se reduz o risco do agente introduzir bias que gerem uma grande perda de resultados quando estiver interagindo com o ambiente real. Além disso, o *Q-learning* utiliza uma política diferente para colher os dados da qual ele tenta otimizar, o que resulta em uma maior eficiência amostral em virtude do reuso mais eficiente dos dados (NGUYEN e LA, 2019).

No entanto, o algoritmo do *Q-learning* depende da criação, preenchimento e armazenamento de uma tabela com os valores aproximados para cada estado-ação do ambiente, chamada de *Q-table*. O uso de tabelas inviabiliza a utilização do algoritmo sem modificações para problemas mais complexos, por causa do armazenamento necessário, ao passo que em ambientes mais simples o algoritmo possui bons resultados, além servir de base para modelos mais complexos de aprendizado por reforço. O *Q-learning* na fase de treino possui uma probabilidade de escolher uma ação aleatória para aumentar a exploração do ambiente, o que reduz as chances de ficar preso em ótimos locais, e constantemente utiliza do resultado das ações para atualizar os valores de $Q(S, A)$ conforme a seguinte equação

de Bellman 2.1 (BELLMAN, 1954), onde S é o estado, A é a ação, α é a taxa de aprendizado, γ é o fator de desconto, R é a recompensa e t é o instante de tempo.

$$Novo Q(S_t, A_t) = Q(S_t, A_t) + \alpha * (R_t + \gamma * \max_a Q(S_{t+1}, a) - Q(S_t, A_t)) \quad (2.1)$$

Capítulo 3

Metodologia

Este estudo de caso se faz com a utilização do *framework* TensorFlow 2.0, uma atualização do TensorFlow original lançada pelo Google em setembro de 2019 (GADDE, 2019). A utilização de uma *framework*, como o TensorFlow, simplifica os processos de criação, de treino e de execução do agente inteligente, o que, por um lado, permite um maior investimento energético em melhorar as definições do ambiente e em desenvolver as métricas utilizadas para recompensar ou punir as ações tomadas pelo agente durante o treino, contudo, por outro lado, passa o controle do treino e desenvolvimento do agente para a *framework*, por conseguinte, os transforma em uma caixa preta cujo o funcionamento só é conhecido pelos desenvolvedores de dada *framework*.

Para realizar o treinamento é necessário um computador rodando sistema Linux, preferencialmente Ubuntu uma vez que é o único sistema operacional oficialmente suportado pelo simulador (PAULL *et al.*, 2022). Dessa forma, este trabalho é desenvolvido na versão 22.04.1 do Ubuntu para processadores de 64-bit, além do uso de GPU GeForce RTX 2060 da Nvidia™ para acelerar o processo ao permitir uma maior paralelização de instâncias de treino e de CPU Intel Core™ i7-8700¹ como visto na figura 3.1. Para o gerenciamento de software usa-se a versão 4.13.0 do Conda (Anaconda 2022) que é capaz de controlar a versão do Python além de criar ambientes independentes para cada projeto, e é utilizado o Python 3.10.4, a versão mais recente de Python quando este trabalho foi iniciado.

3.1 Critérios de treinamento

Como em qualquer método de ensino, para realizar um aprendizado por reforço é necessário alguma métrica com a qual se é possível julgar se o comportamento aprendido é bom ou ruim. Desta forma, para este trabalho utiliza de diferentes parâmetros para calcular a pontuação da rede neural durante o treino, em primeiro lugar, é encerrada prematuramente qualquer instância em que o robô saia da pista, pois sair da pista é visto como uma infração gravíssima. Depois, a rede neural recebe uma bonificação pequena por andar do lado certo da pista, uma média por andar em linha reta e uma grande toda vez que o Duckiebot atinge uma nova casa da pista, com a finalidade de incentivar o robô a se

¹ Para a definição de hardware completa consulte o anexo A

Hardware Model	Gigabyte Technology Co., Ltd. Z390 M GAMING
Memory	16,0 GiB
Processor	Intel® Core™ i7-8700K CPU @ 3.70GHz × 12
Graphics	NVIDIA Corporation TU104 [GeForce RTX 2060]
Disk Capacity	1,5 TB
OS Name	Ubuntu 22.04.1 LTS
OS Type	64-bit
GNOME Version	42.4
Windowing System	X11
Software Updates	>

Figura 3.1: Informações do sistema.

mover pelo circuito definido ao invés de ficar girando dentro das mesmas casas da pista. E por último, existe uma punição pequena quando o robô faz curvas, para promover uma maior estabilidade no movimento, uma média quando o robô anda na contra-mão e uma grande quando o ângulo do robô em relação a pista é maior que 40°, cujo o intuito é inibir que ele fique girando na pista ao invés de seguir o circuito.

Ademais, as definições da rede neural também são de suma importância para permitir um bom aprendizado por parte da inteligência artificial. Com isso em mente, a rede densa possui 3 camadas, com 100, 75 e 50 nos respectivamente, e de entrada ela recebe a distância em relação à faixa central da pista, a qual é definida como 1 sendo o limite direito da pista, -1 o limite esquerdo e 0 a faixa central, e o ângulo que o robô está fazendo com a pista, 0 quando ele está paralelo à pista e ± 90 ao estar perpendicular à pista. E por fim, a rede retorna um número inteiro correspondente a ação que deve ser tomada, com o universo das ações possíveis simplificado para 3 diferentes possibilidades: ou ir em frente ou virar à esquerda ou virar à direita; para acelerar o processo de treinamento, uma vez que ele progredia muito lentamente com as 25 possibilidades iniciais.

Capítulo 4

Conclusão

Aprendizado de máquina, e principalmente aprendizado por reforço, nem sempre é capaz de chegar ao resultado esperado e infelizmente este trabalho é um desses casos. Isso acontece por diversos motivos, mas os principais são uma alta complexidade a qual faz com que o aprendizado precise de muito tempo para convergir e o sistema, mais comumente em aprendizagem por reforço, pode aprender um comportamento diferente do desejado. Este trabalho simplifica o aprendizado ao reduzir o número de ações possíveis, porém não é capaz de gerar regras para recompensar o robô de forma que incentive o modelo a aprender a percorrer o circuito ao invés de ficar girando em círculos ou valer-se de outra forma para maximizar a pontuação.

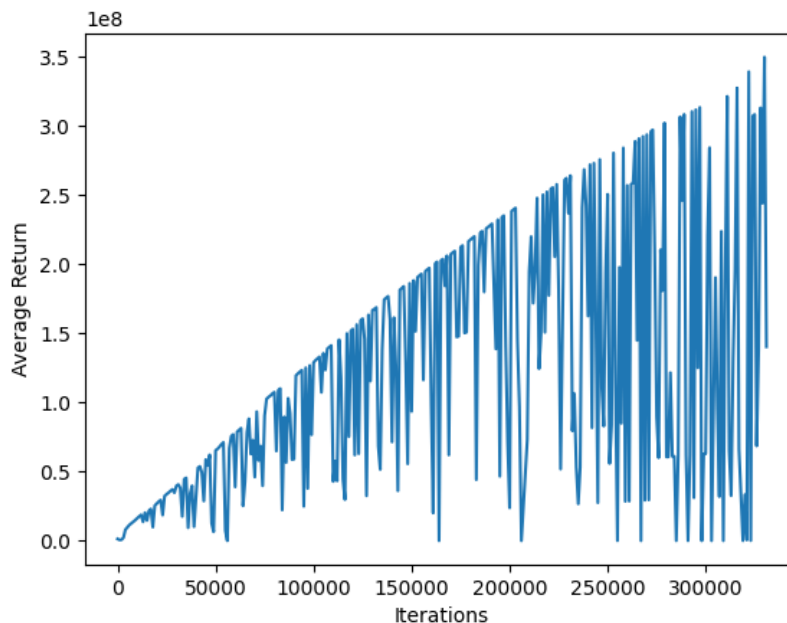


Figura 4.1: *Gráfico da evolução durante sessão de treino.*

Outro grande problema é a grande instabilidade do modelo durante as sessões de treinamento, como pode ser observado na [4.1](#). Tal instabilidade na pontuação não parece

ser proveniente da alta bonificação dada quando o robô avança para uma nova casa da pista mas das formas que ele encontra para tirar proveito do sistema de pontuação, o que por sua vez dificulta analisar a evolução ou regressão do robô. Finalmente, mesmo com as simplificações e mudanças do modelo o processo de aprendizagem ainda leva muito tempo para gerar resultados conclusivos sobre o desempenho do robô, assim faz com que seja necessário um gasto muito grande de tempo para poder determinar se o comportamento visto é originário de uma má pontuação ou do comportamento aleatório aprendido enquanto se explora o ambiente.

Em síntese, para obter resultados mais satisfatórios se faz necessário um maior desenvolvimento do projeto a fim de remover as insuficiências do modelo. Dessa maneira, é fundamental a implementação de novos métodos de otimização sejam capazes de acelerar o processo de aprendizado de forma que se torne possível uma mais rápida avaliação da evolução do treinamento da rede neural e, por conseguinte, a reavaliação dos critérios de pontuação. Por último, também é de suma importância um estudo mais aprofundado no que faz um bom sistema de pontuação para aprendizado por reforço, em virtude de estabelecer bons parâmetros de avaliação da evolução da rede além de permitir a localização com uma maior exatidão da causa do comportamento inesperado aprendido pelo robô.

Anexo A

Especificações de hardware

```

cainotis-z390-m-gaming
  description: Desktop Computer
  product: Z390 M GAMING (Default string)
  vendor: Gigabyte Technology Co., Ltd.
  version: Default string
  serial: Default string
  width: 64 bits
  capabilities: smbios-3.1.1 dmi-3.1.1 smp vsyscall32
  configuration: boot=normal chassis=desktop family=Default string sku=Default string
*-core
  description: Motherboard
  product: Z390 M GAMING-CF
  vendor: Gigabyte Technology Co., Ltd.
  physical id: 0
  version: x.x
  serial: Default string
  slot: Default string
*-firmware
  description: BIOS
  vendor: American Megatrends Inc.
  physical id: 0
  version: F8
  date: 10/15/2019
  size: 64KiB
  capacity: 16MiB
  capabilities: pci upgrade shadowing cdboot bootselect socketedrom edd int13floppy0 floppy1 floppy2 floppy3 floppy4 floppy5 floppy6 floppy7 floppy8 floppy9
*-memory
  description: System Memory
  physical id: 3a
  slot: System board or motherboard
  size: 16GiB
*-bank:0

```

```
description: DIMM DDR4 Synchronous 2133 MHz (0,5 ns)
product: CMK16GX4M2B3000C15
vendor: AMI
physical id: 0
serial: 000000000
slot: ChannelA-DIMM0
size: 8GiB
width: 64 bits
clock: 2133MHz (0.5ns)
*-bank:1
  description: [empty]
  physical id: 1
  slot: ChannelA-DIMM1
*-bank:2
  description: DIMM DDR4 Synchronous 2133 MHz (0,5 ns)
  product: CMK16GX4M2B3000C15
  vendor: AMI
  physical id: 2
  serial: 000000000
  slot: ChannelB-DIMM0
  size: 8GiB
  width: 64 bits
  clock: 2133MHz (0.5ns)
*-bank:3
  description: [empty]
  physical id: 3
  slot: ChannelB-DIMM1
*-cache:0
  description: L1 cache
  physical id: 44
  slot: L1 Cache
  size: 384KiB
  capacity: 384KiB
  capabilities: synchronous internal write-back unified
  configuration: level=1
*-cache:1
  description: L2 cache
  physical id: 45
  slot: L2 Cache
  size: 1536KiB
  capacity: 1536KiB
  capabilities: synchronous internal write-back unified
  configuration: level=2
*-cache:2
  description: L3 cache
  physical id: 46
```

```

slot: L3 Cache
size: 12MiB
capacity: 12MiB
capabilities: synchronous internal write-back unified
configuration: level=3
*-cpu
description: CPU
product: Intel(R) Core(TM) i7-8700K CPU @ 3.70GHz
vendor: Intel Corp.
physical id: 47
bus info: cpu@0
version: 6.158.10
serial: To Be Filled By O.E.M.
slot: U3E1
size: 4317MHz
capacity: 4700MHz
width: 64 bits
clock: 100MHz
capabilities: lm fpu fpu_exception wp vme de pse tsc msr pae mce cx8 apic sep
configuration: cores=6 enabledcores=6 microcode=240 threads=12
*-pci
description: Host bridge
product: 8th Gen Core Processor Host Bridge/DRAM Registers
vendor: Intel Corporation
physical id: 100
bus info: pci@0000:00:00.0
version: 07
width: 32 bits
clock: 33MHz
configuration: driver=skl_uncore
resources: irq:0
*-pci:0
description: PCI bridge
product: 6th-10th Gen Core Processor PCIe Controller (x16)
vendor: Intel Corporation
physical id: 1
bus info: pci@0000:00:01.0
version: 07
width: 32 bits
clock: 33MHz
capabilities: pci pm msi pciexpress normal_decode bus_master cap_list
configuration: driver=pcieport
resources: irq:121 ioport:3000(size=4096) memory:52000000-550fffff ioport:
*-display
description: VGA compatible controller
product: TU104 [GeForce RTX 2060]

```

```

    vendor: NVIDIA Corporation
    physical id: 0
    bus info: pci@0000:01:00.0
    version: a1
    width: 64 bits
    clock: 33MHz
    capabilities: pm msi pciexpress vga_controller bus_master cap_list
    configuration: driver=nvidia latency=0
    resources: irq:142 memory:54000000-54ffffff memory:40000000-4ffffff

*-multimedia
    description: Audio device
    product: TU104 HD Audio Controller
    vendor: NVIDIA Corporation
    physical id: 0.1
    bus info: pci@0000:01:00.1
    logical name: card1
    logical name: /dev/snd/controlC1
    logical name: /dev/snd/hwC1D0
    logical name: /dev/snd/pcmC1D10p
    logical name: /dev/snd/pcmC1D11p
    logical name: /dev/snd/pcmC1D12p
    logical name: /dev/snd/pcmC1D3p
    logical name: /dev/snd/pcmC1D7p
    logical name: /dev/snd/pcmC1D8p
    logical name: /dev/snd/pcmC1D9p
    version: a1
    width: 32 bits
    clock: 33MHz
    capabilities: pm msi pciexpress bus_master cap_list
    configuration: driver=snd_hda_intel latency=0
    resources: irq:17 memory:55080000-55083fff

*-input:0
    product: HDA NVidia HDMI/DP,pcm=3
    physical id: 0
    logical name: input18
    logical name: /dev/input/event10

*-input:1
    product: HDA NVidia HDMI/DP,pcm=7
    physical id: 1
    logical name: input19
    logical name: /dev/input/event11

*-input:2
    product: HDA NVidia HDMI/DP,pcm=8
    physical id: 2
    logical name: input20
    logical name: /dev/input/event12

```



```

*-input:3
    product: HDA NVidia HDMI/DP,pcm=9
    physical id: 3
    logical name: input21
    logical name: /dev/input/event13
*-input:4
    product: HDA NVidia HDMI/DP,pcm=10
    physical id: 4
    logical name: input22
    logical name: /dev/input/event14
*-input:5
    product: HDA NVidia HDMI/DP,pcm=11
    physical id: 5
    logical name: input23
    logical name: /dev/input/event15
*-input:6
    product: HDA NVidia HDMI/DP,pcm=12
    physical id: 6
    logical name: input24
    logical name: /dev/input/event16
*-usb
    description: USB controller
    product: TU104 USB 3.1 Host Controller
    vendor: NVIDIA Corporation
    physical id: 0.2
    bus info: pci@0000:01:00.2
    version: a1
    width: 64 bits
    clock: 33MHz
    capabilities: msi pciexpress pm xhci cap_list
    configuration: driver=xhci_hcd latency=0
    resources: irq:129 memory:50000000-5003ffff memory:50040000-5004ffff
*-usbhost:0
    product: xHCI Host Controller
    vendor: Linux 5.15.0-52-generic xhci-hcd
    physical id: 0
    bus info: usb@3
    logical name: usb3
    version: 5.15
    capabilities: usb-2.00
    configuration: driver=hub slots=2 speed=480Mbit/s
*-usbhost:1
    product: xHCI Host Controller
    vendor: Linux 5.15.0-52-generic xhci-hcd
    physical id: 1
    bus info: usb@4

```

```

        logical name: usb4
        version: 5.15
        capabilities: usb-3.10
        configuration: driver=hub slots=4 speed=10000Mbit/s
*-serial
    description: Serial bus controller
    product: TU104 USB Type-C UCSI Controller
    vendor: NVIDIA Corporation
    physical id: 0.3
    bus info: pci@0000:01:00.3
    version: a1
    width: 32 bits
    clock: 33MHz
    capabilities: msi pciexpress pm bus_master cap_list
    configuration: driver=nvidia-gpu latency=0
    resources: irq:126 memory:55084000-55084fff
*-generic
    description: Signal processing controller
    product: Cannon Lake PCH Thermal Controller
    vendor: Intel Corporation
    physical id: 12
    bus info: pci@0000:00:12.0
    version: 10
    width: 64 bits
    clock: 33MHz
    capabilities: pm msi cap_list
    configuration: driver=intel_pch_thermal latency=0
    resources: irq:16 memory:5533d000-5533dfff
*-usb
    description: USB controller
    product: Cannon Lake PCH USB 3.1 xHCI Host Controller
    vendor: Intel Corporation
    physical id: 14
    bus info: pci@0000:00:14.0
    version: 10
    width: 64 bits
    clock: 33MHz
    capabilities: pm msi xhci bus_master cap_list
    configuration: driver=xhci_hcd latency=0
    resources: irq:128 memory:55320000-5532ffff
*-usbhost:0
    product: xHCI Host Controller
    vendor: Linux 5.15.0-52-generic xhci-hcd
    physical id: 0
    bus info: usb@1
    logical name: usb1

```

```

version: 5.15
capabilities: usb-2.00
configuration: driver=hub slots=16 speed=480Mbit/s
*-usb:0
  description: USB hub
  product: USB Hub 2.0
  vendor: ALCOR
  physical id: 1
  bus info: usb@1:1
  version: 7.02
  capabilities: usb-2.00
  configuration: driver=hub maxpower=100mA slots=4 speed=12Mbit/s
*-usb:0
  description: Mouse
  product: Wacom One by Wacom S Pen
  vendor: Wacom Co.,Ltd.
  physical id: 1
  bus info: usb@1:1.1
  logical name: input12
  logical name: /dev/input/event5
  logical name: /dev/input/mouse1
  version: 1.00
  serial: 0KE00L1003422
  capabilities: usb-2.00 usb
  configuration: driver=usbhid maxpower=498mA speed=12Mbit/s
*-usb:1 UNCLAIMED
  description: Generic USB device
  product: Xbox 360 Wireless Receiver for Windows
  physical id: 2
  bus info: usb@1:1.2
  version: 1.00
  serial: E041DA20
  capabilities: usb-2.00
  configuration: maxpower=260mA speed=12Mbit/s
*-usb:1
  description: Mouse
  product: E-Signal/A-One COUGAR Minos X2 Keyboard
  vendor: E-Signal/A-One
  physical id: 5
  bus info: usb@1:5
  logical name: input3
  logical name: /dev/input/event3
  logical name: /dev/input/mouse0
  logical name: input4
  logical name: /dev/input/event4
  version: 1.11

```

```

        capabilities: usb-2.00 usb
        configuration: driver=usbhid maxpower=100mA speed=12Mbit/s
*-usb:2
    description: Keyboard
    product: COUGAR Vantar COUGAR Vantar System Control
    vendor: COUGAR Vantar
    physical id: 6
    bus info: usb@1:6
    logical name: input11
    logical name: /dev/input/event9
    logical name: input7
    logical name: /dev/input/event6
    logical name: input7::capslock
    logical name: input7::compose
    logical name: input7::kana
    logical name: input7::numlock
    logical name: input7::scrolllock
    logical name: input8
    logical name: /dev/input/event7
    logical name: input9
    logical name: /dev/input/event8
    version: 1.10
    capabilities: usb-1.10 usb
    configuration: driver=usbhid maxpower=500mA speed=1Mbit/s
*-usbhost:1
    product: xHCI Host Controller
    vendor: Linux 5.15.0-52-generic xhci-hcd
    physical id: 1
    bus info: usb@2
    logical name: usb2
    version: 5.15
    capabilities: usb-3.10
    configuration: driver=hub slots=10 speed=10000Mbit/s
*-memory UNCLAIMED
    description: RAM memory
    product: Cannon Lake PCH Shared SRAM
    vendor: Intel Corporation
    physical id: 14.2
    bus info: pci@0000:00:14.2
    version: 10
    width: 64 bits
    clock: 33MHz (30.3ns)
    capabilities: pm cap_list
    configuration: latency=0
    resources: memory:55336000-55337fff memory:5533c000-5533cfff
*-communication

```

```

description: Communication controller
product: Cannon Lake PCH HECI Controller
vendor: Intel Corporation
physical id: 16
bus info: pci@0000:00:16.0
version: 10
width: 64 bits
clock: 33MHz
capabilities: pm msi bus_master cap_list
configuration: driver=mei_me latency=0
resources: irq:140 memory:5533b000-5533bfff

```

*-sata

```

description: SATA controller
product: Cannon Lake PCH SATA AHCI Controller
vendor: Intel Corporation
physical id: 17
bus info: pci@0000:00:17.0
logical name: scsi0
logical name: scsi1
version: 10
width: 32 bits
clock: 66MHz
capabilities: sata msi pm ahci_1.0 bus_master cap_list emulated
configuration: driver=ahci latency=0
resources: irq:127 memory:55334000-55335fff memory:5533a000-5533a0ff ioport:5533a000-5533a00f

```

*-cdrom

```

description: DVD-RAM writer
product: DVD-RAM GH24NSB0
vendor: HL-DT-ST
physical id: 0
bus info: scsi@0:0.0.0
logical name: /dev/cdrom
logical name: /dev/sr0
version: LM01
capabilities: removable audio cd-r cd-rw dvd dvd-r dvd-ram
configuration: ansiversion=5 status=nodisc

```

*-disk

```

description: ATA Disk
product: WDC WD10EZEX-00B
vendor: Western Digital
physical id: 1
bus info: scsi@1:0.0.0
logical name: /dev/sda
version: 1A01
serial: WD-WCC3FEA3S9DD
size: 931GiB (1TB)

```

```

capabilities: gpt-1.00 partitioned partitioned:gpt
configuration: ansiversion=5 guid=922b4323-45ef-449c-9d8b-13588d88
*-volume:0 UNCLAIMED
    description: Windows FAT volume
    vendor: mkfs.fat
    physical id: 1
    bus info: scsi@1:0.0.0,1
    version: FAT32
    serial: 3a19-5553
    size: 510MiB
    capacity: 511MiB
    capabilities: boot fat initialized
    configuration: FATs=2 filesystem=fat name=EFI System Partition
*-volume:1
    description: EXT4 volume
    vendor: Linux
    physical id: 2
    bus info: scsi@1:0.0.0,2
    logical name: /dev/sda2
    logical name: /
    logical name: /var/snap/firefox/common/host-hunspell
    version: 1.0
    serial: 6b9b5a59-c110-468a-a502-50d72f0f0e16
    size: 931GiB
    capabilities: journaled extended_attributes large_files huge_files
    configuration: created=2022-06-08 19:33:05 filesystem=ext4 last
*-pci:1
    description: PCI bridge
    product: Cannon Lake PCH PCI Express Root Port #17
    vendor: Intel Corporation
    physical id: 1b
    bus info: pci@0000:00:1b.0
    version: f0
    width: 32 bits
    clock: 33MHz
    capabilities: pci pciexpress msi pm normal_decode bus_master cap_list
    configuration: driver=pcieport
    resources: irq:122
*-pci:2
    description: PCI bridge
    product: Cannon Lake PCH PCI Express Root Port #21
    vendor: Intel Corporation
    physical id: 1b.4
    bus info: pci@0000:00:1b.4
    version: f0
    width: 32 bits

```

```

clock: 33MHz
capabilities: pci pciexpress msi pm normal_decode bus_master cap_list
configuration: driver=pcieport
resources: irq:123 memory:55200000-552fffff
*-nvme
    description: NVMe device
    product: INTEL SSDPEKNW512G8
    vendor: Intel Corporation
    physical id: 0
    bus info: pci@0000:03:00.0
    logical name: /dev/nvme0
    version: 004C
    serial: PHNH944304ZP512A
    width: 64 bits
    clock: 33MHz
    capabilities: nvme pm msi pciexpress msix nvm_express bus_master cap_list
    configuration: driver=nvme latency=0 nqn=nqn.2019-44.com.intel:nvm-subnqn.2019-44.com:500000000000000000
    resources: irq:16 memory:55200000-55203fff
*-namespace:0
    description: NVMe disk
    physical id: 0
    logical name: hwmon1
*-namespace:1
    description: NVMe disk
    physical id: 2
    logical name: /dev/ng0n1
*-namespace:2
    description: NVMe disk
    physical id: 1
    bus info: nvme@0:1
    logical name: /dev/nvme0n1
    size: 476GiB (512GB)
    capabilities: partitioned partitioned:dos
    configuration: logicalsectorsize=512 sectorsize=512 signature=023d83
*-volume:0
    description: Windows NTFS volume
    physical id: 1
    bus info: nvme@0:1,1
    logical name: /dev/nvme0n1p1
    version: 3.1
    serial: 6c1caca4-4352-1c49-a646-bee9ef73a6d6
    size: 348MiB
    capacity: 350MiB
    capabilities: primary bootable ntfs initialized
    configuration: clustersize=4096 created=2014-12-13 16:01:52 files
*-volume:1

```

```

        description: Windows NTFS volume
        physical id: 2
        bus info: nvme@0:1,2
        logical name: /dev/nvme0n1p2
        version: 3.1
        serial: 4400abf3-c04e-5a4a-aa59-92c151350d33
        size: 476GiB
        capacity: 476GiB
        capabilities: primary ntfs initialized
        configuration: clustersize=4096 created=2014-12-13 16:01:53
*-volume:2
        description: Windows NTFS volume
        physical id: 3
        bus info: nvme@0:1,3
        logical name: /dev/nvme0n1p3
        version: 3.1
        serial: a28d-fa64
        size: 525MiB
        capacity: 552MiB
        capabilities: primary ntfs initialized
        configuration: clustersize=4096 created=2020-07-02 23:35:02
*-pci:3
        description: PCI bridge
        product: Cannon Lake PCH PCI Express Root Port #1
        vendor: Intel Corporation
        physical id: 1c
        bus info: pci@0000:00:1c.0
        version: f0
        width: 32 bits
        clock: 33MHz
        capabilities: pci pciexpress msi pm normal_decode bus_master cap_list
        configuration: driver=pcieport
        resources: irq:124
*-pci:4
        description: PCI bridge
        product: Cannon Lake PCH PCI Express Root Port #9
        vendor: Intel Corporation
        physical id: 1d
        bus info: pci@0000:00:1d.0
        version: f0
        width: 32 bits
        clock: 33MHz
        capabilities: pci pciexpress msi pm normal_decode bus_master cap_list
        configuration: driver=pcieport
        resources: irq:125 ioport:5000(size=4096) memory:50100000-502fffff io
*-isa

```



```
description: ISA bridge
product: Z390 Chipset LPC/eSPI Controller
vendor: Intel Corporation
physical id: 1f
bus info: pci@0000:00:1f.0
version: 10
width: 32 bits
clock: 33MHz
capabilities: isa bus_master
configuration: latency=0
*-pnp00:00
    product: PnP device PNP0c02
    physical id: 0
    capabilities: pnp
    configuration: driver=system
*-pnp00:01
    product: PnP device PNP0501
    physical id: 1
    capabilities: pnp
    configuration: driver=serial
*-pnp00:02
    product: PnP device PNP0c02
    physical id: 2
    capabilities: pnp
    configuration: driver=system
*-pnp00:03
    product: PnP device PNP0b00
    physical id: 3
    capabilities: pnp
    configuration: driver=rtc_cmos
*-pnp00:04
    product: PnP device INT3f0d
    physical id: 4
    capabilities: pnp
    configuration: driver=system
*-pnp00:05
    product: PnP device PNP0c02
    physical id: 5
    capabilities: pnp
    configuration: driver=system
*-pnp00:06
    product: PnP device PNP0c02
    physical id: 6
    capabilities: pnp
    configuration: driver=system
*-pnp00:07
```

```

        product: PnP device PNP0c02
        physical id: 7
        capabilities: pnp
        configuration: driver=system
*-pnp00:08
        product: PnP device PNP0c02
        physical id: 8
        capabilities: pnp
        configuration: driver=system
*-multimedia
        description: Audio device
        product: Cannon Lake PCH cAVS
        vendor: Intel Corporation
        physical id: 1f.3
        bus info: pci@0000:00:1f.3
        logical name: card0
        logical name: /dev/snd/controlC0
        logical name: /dev/snd/hwC0D0
        logical name: /dev/snd/pcmC0D0c
        logical name: /dev/snd/pcmC0D0p
        logical name: /dev/snd/pcmC0D1p
        logical name: /dev/snd/pcmC0D2c
        version: 10
        width: 64 bits
        clock: 33MHz
        capabilities: pm msi bus_master cap_list
        configuration: driver=snd_hda_intel latency=32
        resources: irq:141 memory:55330000-55333fff memory:55100000-551fffff
*-input:0
        product: HDA Intel PCH Front Mic
        physical id: 0
        logical name: input25
        logical name: /dev/input/event17
*-input:1
        product: HDA Intel PCH Rear Mic
        physical id: 1
        logical name: input26
        logical name: /dev/input/event18
*-input:2
        product: HDA Intel PCH Line
        physical id: 2
        logical name: input27
        logical name: /dev/input/event19
*-input:3
        product: HDA Intel PCH Line Out Front
        physical id: 3

```

```

        logical name: input28
        logical name: /dev/input/event20
*-input:4
    product: HDA Intel PCH Line Out Surround
    physical id: 4
    logical name: input29
    logical name: /dev/input/event21
*-input:5
    product: HDA Intel PCH Line Out CLFE
    physical id: 5
    logical name: input30
    logical name: /dev/input/event22
*-input:6
    product: HDA Intel PCH Line Out Side
    physical id: 6
    logical name: input31
    logical name: /dev/input/event23
*-input:7
    product: HDA Intel PCH Front Headphone
    physical id: 7
    logical name: input32
    logical name: /dev/input/event24
*-serial:0
    description: SMBus
    product: Cannon Lake PCH SMBus Controller
    vendor: Intel Corporation
    physical id: 1f.4
    bus info: pci@0000:00:1f.4
    version: 10
    width: 64 bits
    clock: 33MHz
    configuration: driver=i801_smbus latency=0
    resources: irq:16 memory:55338000-553380ff ioport:efa0(size=32)
*-serial:1 UNCLAIMED
    description: Serial bus controller
    product: Cannon Lake PCH SPI Controller
    vendor: Intel Corporation
    physical id: 1f.5
    bus info: pci@0000:00:1f.5
    version: 10
    width: 32 bits
    clock: 33MHz
    configuration: latency=0
    resources: memory:fe010000-fe010fff
*-network
    description: Ethernet interface

```

```

        product: Ethernet Connection (7) I219-V
        vendor: Intel Corporation
        physical id: 1f.6
        bus info: pci@0000:00:1f.6
        logical name: eno1
        version: 10
        serial: b4:2e:99:f2:25:8a
        size: 1Gbit/s
        capacity: 1Gbit/s
        width: 32 bits
        clock: 33MHz
        capabilities: pm msi bus_master cap_list ethernet physical tp 10bt 10
        configuration: autonegotiation=on broadcast=yes driver=e1000e driverv
        resources: irq:131 memory:55300000-5531ffff
*-power UNCLAIMED
    description: To Be Filled By O.E.M.
    product: To Be Filled By O.E.M.
    vendor: To Be Filled By O.E.M.
    physical id: 1
    version: To Be Filled By O.E.M.
    serial: To Be Filled By O.E.M.
    capacity: 32768mWh
*-graphics
    product: EFI VGA
    physical id: 2
    logical name: /dev/fb0
    capabilities: fb
    configuration: depth=32 resolution=1024,768
*-input:0
    product: Sleep Button
    physical id: 3
    logical name: input0
    logical name: /dev/input/event0
    capabilities: platform
*-input:1
    product: Power Button
    physical id: 4
    logical name: input1
    logical name: /dev/input/event1
    capabilities: platform
*-input:2
    product: Power Button
    physical id: 5
    logical name: input2
    logical name: /dev/input/event2
    capabilities: platform

```

Referências

- [Anaconda 2022] Anaconda. URL: <https://www.anaconda.com/> (acesso em 08/11/2022) (citado na pg. 7).
- [BAUM e DENSLOW 1996] L. Frank BAUM e W. W. DENSLOW. *The wonderful wizard of Oz*. New York: Dover Publications, 1996. ISBN: 9780486291161 (citado na pg. 1).
- [BELLMAN 1954] Richard BELLMAN. “The theory of dynamic programming”. en. Em: *Bulletin of the American Mathematical Society* 60.6 (1954), pgs. 503–515. ISSN: 0273-0979, 1088-9485. DOI: [10.1090/S0002-9904-1954-09848-8](https://doi.org/10.1090/S0002-9904-1954-09848-8). URL: <https://www.ams.org/bull/1954-60-06/S0002-9904-1954-09848-8/> (citado na pg. 6).
- [BROCKMAN et al. 2016] Greg BROCKMAN et al. *OpenAI Gym*. 2016. eprint: [arXiv:1606.01540](https://arxiv.org/abs/1606.01540) (citado na pg. 4).
- [CHEVALIER-BOISVERT et al. 2018] Maxime CHEVALIER-BOISVERT, Florian GOLEMO, Yan-jun CAO, Bhairav MEHTA e Liam PAULL. *Duckietown Environments for OpenAI Gym*. <https://github.com/duckietown/gym-duckietown>. 2018 (citado nas pgs. vi, 4).
- [Duckietown Foundation 2022] Duckietown Foundation. 2022. URL: https://www.duckietown.org/about/duckietown-foundation?doing_wp_cron=1667914840.7376658916473388671875 (acesso em 08/11/2022) (citado na pg. 3).
- [GADDE 2019] Goldie GADDE. *Epsilon-Greedy Q-learning*. 30 de set. de 2019. URL: <https://github.com/tensorflow/tensorflow/releases/tag/v2.0.0> (acesso em 08/11/2022) (citado na pg. 7).
- [Guide for Researches - Duckietown 2022] Guide for Researches - Duckietown. 2022. URL: <https://www.duckietown.org/research/guide-for-researchers> (acesso em 08/11/2022) (citado nas pgs. vi, 3).
- [MNIH et al. 2015] Volodymyr MNIH et al. “Human-level control through deep reinforcement learning”. en. Em: *Nature* 518.7540 (fev. de 2015), pgs. 529–533. ISSN: 0028-0836, 1476-4687. DOI: [10.1038/nature14236](https://doi.org/10.1038/nature14236). URL: <http://www.nature.com/articles/nature14236> (citado na pg. 5).

- [NGUYEN e LA 2019] Hai NGUYEN e Hung LA. “Review of deep reinforcement learning for robot manipulation”. Em: *2019 Third IEEE International Conference on Robotic Computing (IRC)*. Naples, Italy: IEEE, fev. de 2019, pgs. 590–595. ISBN: 9781538692455. DOI: [10.1109/IRC.2019.00120](https://doi.org/10.1109/IRC.2019.00120). URL: <https://ieeexplore.ieee.org/document/8675643/> (citado na pg. 5).
- [PAULL et al. 2022] Liam PAULL et al. *Book The AI Driving Olympics*. Inglês. 2 de fev. de 2022. URL: https://docs.duckietown.org/daffy/AIDO/out/cm_sw.html (acesso em 16/11/2022) (citado na pg. 7).
- [TURING 1950] A. M. TURING. “Computing machinery and intelligence”. English. Em: *Mind*. New Series 59.236 (1950), pgs. 433–460. ISSN: 00264423. URL: <http://www.jstor.org/stable/2251299> (citado na pg. 1).
- [UHR e VOSSLER 1961] Leonard UHR e Charles VOSSLER. “A pattern recognition program that generates, evaluates, and adjusts its own operators”. en. Em: *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference on - IRE-AIEE-ACM '61 (Western)*. Los Angeles, California: ACM Press, 1961, pg. 555. DOI: [10.1145/1460690.1460751](https://doi.org/10.1145/1460690.1460751). URL: <http://portal.acm.org/citation.cfm?doid=1460690.1460751> (citado na pg. 2).
- [WATKINS 1989] Chris WATKINS. “Learning from Delayed Rewards”. Tese de dout. Londres, Inglaterra: King’s College, mai. de 1989. URL: https://www.cs.rhul.ac.uk/~chrisw/new_thesis.pdf (acesso em 08/11/2022) (citado na pg. 5).
- [WEIZENBAUM 1966] Joseph WEIZENBAUM. “Eliza a computer program for the study of natural language communication between man and machine”. Em: *Commun. ACM* 9.1 (jan. de 1966), pgs. 36–45. ISSN: 0001-0782. DOI: [10.1145/365153.365168](https://doi.org/10.1145/365153.365168). URL: <http://doi.acm.org/10.1145/365153.365168> (citado na pg. 1).

Índice remissivo

A

Alan M. Turing, [1](#)

B

Bellman, [6](#)

D

DeepMind, [5](#)

Deep Q-Network, [5](#)

Duckietown, [3](#)

G

Google, [7](#)

Gym-Duckietown, [4](#)

K

Karel Čapek, [1](#)

M

MIT, [3](#)

O

OpenAI, [4](#)

Q

Q-learning, [5](#)

Q-table, [5](#)

S

simulador, [4](#)

T

TensorFlow, [7](#)