

---

This is the **published version** of the bachelor thesis:

Morales Soto, Rubén; Pedret Ferré, Carles, dir. Tic tac toe amb intel·ligència artificial. 2023. 8 pag. (Grau en Gestió de Ciutats Intel·ligents i Sostenibles)

---

This version is available at <https://ddd.uab.cat/record/282838>

under the terms of the  license

---

This is the **published version** of the bachelor thesis:

Morales Soto, Rubén; Pedret Ferré, Carles, dir. Tic tac toe amb intel·ligència artificial. 2023. 8 pag. (Grau en Gestió de Ciutats Intel·ligents i Sostenibles)

---

This version is available at <https://ddd.uab.cat/record/282838>

under the terms of the  license

# Tic tac toe amb intel·ligència artificial

Rubén Morales Soto

**Resum**—El tres en ratlla és el joc seleccionat per poder aplicar els coneixements i tecnologies apreses al llarg del grau. Aquest projecte parteix d'un anterior en el qual es va desenvolupar un robot capaç de moure les peces del joc amb moviments provocats de forma manual. Després del projecte inicial, queda dotar d'intel·ligència al robot. Per fer-ho, ha de saber reconèixer la situació de les peces al tauler a través d'una càmera, calcular la millor jugada i fer els moviments. Aquest projecte es centra en la part de reconeixement de les peces del tauler a través d'eines de processament de vídeo. Es deixen les altres parts per futurs projectes.

**Paraules clau**—Intel·ligència artificial, tres en ratlla, robòtica, ciutats intel·ligents, visió per computador, reconeixement d'imatges, Node-RED, Python, JavaScript, bases de dades

**Resumen**—El tres en raya es el juego seleccionado para poder aplicar los conocimientos y tecnologías aprendidas a lo largo del grado. Este proyecto parte de uno anterior en el cual se se había desarrollado un robot capaz de mover las piezas del juego con movimientos provocados de forma manual. Después del proyecto inicial, queda dotar de inteligencia al robot. Para hacerlo, ha de saber reconocer la situación de las piezas del tablero a través de una cámara, calcular la mejor jugada y hacer los movimientos. Este proyecto se centra en la parte de reconocimiento de las piezas del tablero a través de herramientas de procesamiento de vídeo. Se dejan las otras partes para futuros proyectos.

**Paraules clau**—Inteligencia artificial, tres en raya, robótica, ciudades inteligentes, visión por computador, reconocimiento de imágenes, Node-RED, Python, JavaScript, bases de datos

**Abstract**—Tic tac toe is the selected game to apply the knowledge and technologies learned through the degree. This project comes from a previous one in which was developed a robot able to move game pieces with manual movements. After the initial project, it remains to provide intelligence to the robot. To do this, it must know how to recognize the situation of the pieces of the board through a camera, calculate the best move and make the moves. This project focuses on the part of recognizing the pieces of the board through video processing tools. The other parts are left for future projects.

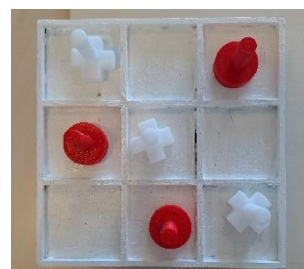
**Index Terms**—Artificial intelligence, tic tac toe, robotics, smart cities, computer vision, image recognition, Node-RED, Python, JavaScript, databases

## 1 INTRODUCCIÓ

AQUEST projecte és la continuació d'un altre començat el curs 2021-2022 per altres estudiants del grau en gestió de ciutats intel·ligents i sostenibles a la Universitat Autònoma de Barcelona, en el qual es va dur a terme la fabricació i programació d'un robot mitjançant impressió 3D [1] per jugar al joc conegut com Tic Tac Toe o tres en ratlla. L'objectiu principal és, seguint aquesta línia de continuïtat relacionada amb el projecte, centrar-se bàsicament en la visió per computador per tal de dotar de visió al robot i que d'aquesta forma pugui conèixer la posició de les peces del tauler. Cal convertir el vídeo que procedirà d'una webcam en una matriu de dades codificades (com a "zeros" o "uns", per exemple), que representaran la posició de les peces del joc. En projectes futurs es desenvoluparà l'algoritme de presa de decisions i en funció de la matriu de dades codificades proporcionada per aquest projecte, un

algoritme calcularà la millor jugada i farà la implementació als moviments del robot, de tal forma que aquest pugui jugar de forma autònoma al tres en ratlla.

Emmarcat en l'àmbit de les ciutats intel·ligents, les competències adquirides en aquest projecte són directament exportables a diverses aplicacions dins d'una ciutat. Per exemple, un sistema de visió per computador pot ser capaç de detectar quantes places de pàrquing queden lliures a un aparcament i així regular l'aforament o activar un LED quan el pàrquing estigui ple, de tal forma que la resta d'usuaris ho puguin saber d'avançat. Un altre cas es pot



Imatges 1 i 2: Robot i tauler.  
Font pròpia

- E-mail de contacte: [ruben.morales@autonoma.cat](mailto:ruben.morales@autonoma.cat)
- Treball tutoritzat per: Carles Pedret Ferré (Departament de Telecomunicació i Enginyeria de Sistemes)
- Curs 2022/23

trobar a un vehicle autònom o a diversos sistemes d'ajuda a la conducció en els quals, mitjançant una webcam, es detecta si hi ha algun vehicle o obstacle pròxim, per tal de facilitar la sortida d'un estacionament a un vehicle.

### 1.1 Objectius

Els objectius d'aquest treball són:

- **Objectiu 1:** Estudi de l'estat de l'art.

Ens proposem fer un estudi de les diferents eines disponibles de visió per computador, analitzar-les i seleccionar les més adequades per fer la tasca en qüestió.

- **Objectiu 2:** Estudi d'elements hardware i software necessaris.

Ens proposem analitzar els elements necessaris, com web cams, i sistemes per monitoritzar de forma esquemàtica amb un tauler digital, la situació de les peces al tauler.

- **Objectiu 3:** Desenvolupar l'escenari de proves i programar.

Ens proposem desenvolupar l'escenari de funcionament i programar.

- **Objectiu 4:** Realització de tests.

Ens proposem dissenyar un conjunt de tests per provar el hardware i software seleccionats i determinar si funcionen correctament.

- **Objectiu 5:** Redactar la memòria i preparar la presentació.

### 1.2 Planificació temporal

Tots aquests objectius es volen assolir mitjançant la següent planificació temporal:

	Febrer	Març	Abril	Maig	Juny
<b>Objectiu 1</b>					
<b>Objectiu 2</b>					
<b>Objectiu 3</b>					
<b>Objectiu 4</b>					
<b>Objectiu 5</b>					

### 1.3 Motivació

Hi ha una motivació especial personal relacionada amb els àmbits de la robòtica, sensors i intel·ligència artificial. Un altre objectiu és reflexar el coneixement obtingut mitjançant diverses assignatures sobre l'àmbit de la programació i la intel·ligència artificial. Finalment, es pretén aprofundir més en el funcionament de llibreries que ajudaran a assolir l'objectiu mencionat prèviament.

### 1.4 Organització del document

L'organització del document està estructurada de la següent forma:

1. Introducció
  - 1.1. Objectius
  - 1.2. Planificació temporal
  - 1.3. Motivació
  - 1.4. Organització del document
2. Estudi de l'estat de l'art
  - 2.1. Intel·ligència artificial
  - 2.2. Visió per computador

2.3. Eines de visió per computador

2.4. Node-RED

2.5. PostgreSQL

3. Desenvolupament i programació

3.1. Diagrama de treball

3.2. Instal·lacions necessàries

3.3. Divisió en imatges individuals

3.4. Anàlisi de les imatges individuals

3.5. Connexió amb la base de dades

3.6. Dashboard a Node-RED

4. Resultats

4.1. Mètode d'avaluació

4.2. Resultats obtinguts

4.2.1. Resultat 1

4.2.2. Resultat 2

4.2.3. Resultat 3

4.3. Discussió

5. Conclusió i línies de continuació

6. Agraïments

7. Referències

## 2 ESTUDI DE L'ESTAT DE L'ART

### 2.1 Intel·ligència artificial

El concepte d'intel·ligència artificial (IA) actualment és molt present a la nostra societat, però cal definir què és realment i en quin punt es troba actualment.

La IA és la capacitat que tenen les màquines d'imitar la forma en que els éssers humans aprenen i raonen. Això s'aconsegueix gràcies a que les màquines aprenen a fer tasques i millorar-les mitjançant la informació que tenen, de forma similar a com ho faria un humà. Té com a objectiu replicar a les màquines els processos cognitius. L'objectiu principal des de que es va començar a investigar aquest camp, finalitzant la II Guerra Mundial, és sintetitzar, emular i automatitzar qualsevol tasca que impliqui la intervenció de la intel·ligència [2]. La quantitat de branques que té la IA actualment és molt extensa, alguns exemples són [3]:

- Machine Learning (ML)

Es fonamenta en l'entrenament d'algoritmes (models) mitjançant dades obtingudes prèviament. A mesura que els models es van entrenant, milloren la seva precisió i aconsegueixen arribar a optimitzar processos.

- Deep Learning (DL)

Branca del ML que es caracteritza per construir i entrenar xarxes neuronals amb múltiples capes amb la finalitat d'endregar, classificar i trobar anomalies a les dades.

- Neural Networks (NN)

Algoritmes de ML i models computacionals dissenyats per simular el comportament de les neurones al cervell humà. S'entrenen amb un conjunt de dades que utilitzen per predir la resposta a una consulta. Si es produeixen errors, les NN ajusten el seu procés i el repeteixen fins que l'error disminueixi.

- Computer Vision (CV) [6]

Permet a un ordinador extraure informació rellevant a partir d'imatges, vídeos i altres entrades visuals, per tal d'oferir mesures o recomanacions en funció de la

informació rebuda mitjançant les entrades. Funciona de la mateixa forma que la visió humana, amb la diferència de que aquesta ha estat molt més entrenada al llarg de tota la vida, i això permet detectar més ràpidament objectes, la seva distància, si s'està movent o qualsevol error al visualitzar una imatge.

- Natural Language Processing (NLP)

S'enfoca en la capacitat de les màquines per entendre paraules verbals o escrites en llenguatge humà.

- Natural Language Generation (NLG)

Fa al·lusió a la capacitat d'una màquina de generar llenguatge natural, de tal forma que pugui donar una resposta coherent a un humà. S'aconsegueix gràcies a la capacitat de comprensió de la màquina i la seva capacitat d'expressar-ho en paraules.

- Chatbot

Utilitza un conjunt de regles per establir una conversa amb un humà mitjançant una interfície. Fan us de NLP (per processar el missatge que l'humà envia) i NLG (per generar la resposta, la qual depèn del missatge enviat). Un exemple és ChatGPT [11].

- Virtual Digital Assistants

Versió sofisticada dels Chatbots que poden organitzar, emmagatzemar i generar informació en funció de la posició de l'usuari. No només responen consultes escrites, sinó també de forma verbal (mitjançant reconeixement de veu). Un exemple és Siri, d'Apple [12].

- Recommender Systems

Sistemes de filtratge d'informació que poden predir les preferències d'un usuari i oferir, per exemple, noves recomanacions (productes, articles, cançons, etcètera) en funció del comportament passat (compres anteriors, pel·lícules vistes, àlbums escoltats, ...).

- Predictive Analysis

Utilitza una combinació de tècniques pròpies de la ciència de dades, estadística i IA per analitzar un conjunt de dades desestructurades, descobrir relacions entre ells i utilitzar-los per tal d'estimar resultats futurs.

## 2.2 Visió per computador

De les múltiples branques de la IA, la visió per computador és la que farem servir en aquest projecte, de manera que el robot obtingui a temps real l'estat de les peces del tauler.

La visió per computador requereix d'un gran volum de dades que s'analitza contínuament fins que el sistema es capaç de percebre diferències i reconèixer patrons. Per tal que reconegui un objecte concret, cal mostrar-li moltes imatges del determinat objecte i que així el pugui distingir amb facilitat. Per aconseguir-ho, es pot utilitzar Deep Learning o una xarxa neuronal convolucional (CNN), un tipus de NN [6]. Amb la primera opció el que s'aconsegueix és que l'ordinador "observi" una gran quantitat de dades i aprengui a diferenciar de forma autònoma les imatges, sense que ningú l'hagi programat perquè les reconegui com a tal. Per altra banda, una CNN desglossa les imatges en píxels i els hi assigna etiquetes, amb les quals realitzarà convolucions (amb dos funcions es genera una tercera) i predirà el que està visualitzant. S'executa de forma iterativa fins que les prediccions comencen a ser reals.

Alguns exemples d'aplicació de la visió per computador són aquests:

- Image classification

Permet predir amb força precisió a quina classe específica pertany una imatge només "observant-la" (gos, gat, poma, pera, ...).

- Object detection

Pot emprar la classificació d'imatges per identificar a quina classe pertany una imatge i, posteriorment, detectar i tabular la seva aparença en una imatge o vídeo.

- Object tracking

Aplicació de la visió per computador a utilitzar en aquest projecte, ja que permet rastrejar la posició d'un objecte a temps real, de la mateixa forma que, en aquest cas, es pretén conèixer la posició de les fitxes dins del tauler. Generalment s'utilitzen imatges en seqüència o transmissions en viu.

- Content-based image retrieval

Examina, cerca i recupera imatges de grans bancs de dades en funció del contingut de les imatges. Permet substituir l'etiquetatge manual d'imatges.

## 2.3 Eines de visió per computador

Cal considerar que hi ha múltiples eines amb les que es poden obtenir bons resultats en matèria de visió per computador. Fins fa poc, la companyia IBM disposava d'una eina per reconeixement d'imatges, anomenat IBM Watson Visual Recognition, però aquest ha estat retirat fa un parell d'anys [5]. Ara, un dels serveis d'intel·ligència artificial que està més de moda és el servei TensorFlow de Google [13], amb el que també es pot fer un reconeixement i detecció d'imatges, però no és el millor en aquest àmbit. Aquestes dues eines s'emarken dins de la categoria de Image classification i Object detection. No obstant, pels propòsits del nostre projecte necessitem una eina dins de la categoria d'Object tracking. N'existeixen diverses com poden ser [8]:

- Scikit Image

És una biblioteca de processament d'imatges open source que s'empra, fonamentalment, per aprenentatge automàtic en Python, tot utilitzant diverses eines i models estadístics. Aquesta plataforma funciona amb tots els formats d'imatges compatibles amb Python.

- SimpleCV

S'empra per crear aplicacions de visió per computador i així tenir accés a diverses llibreries potents com OpenCV de manera senzilla i eficient. Quan rep una imatge la transforma en blanc i negre i dibuixa text per, finalment, mostrar el resultat per pantalla. Gràcies al seu ús fàcil, programadors principiants poden realitzar proves bàsiques de visió per computador.

- NumPy

Aquesta eina conté estructures de dades matricials que ofereixen un millor desenvolupament a nivell computacional a múltiples llenguatges de programació. S'empra principalment per realitzar un seguit de tasques matemàtiques en matrius.

- Imutils

Desenvolupada per realitzar funcions bàsiques de processament d'imatges i també permet la visualització d'imatges de Matplotlib amb l'ajuda d'OpenCV y Python.

Imutils permet construir un esquelet topològic d'un objecte a una imatge on s'assumeix que l'objecte és de color blanc i està posicionat sobre un fons negre.

- IP SDK

Eina de processament d'imatges 2D/3D que analitza les imatges amb l'ajuda de mòduls basats en aprenentatge automàtic. Ofereix una documentació extensa de les funcions que conté pel processament d'imatges. Aquesta eina és la única de les presentades que és de pagament.

- OpenCV

OpenCV és l'eina més popular actualment en aquesta categoria [9]. Es tracta d'una llibreria de codi obert enfocada en ML i CV que incorpora més de 2500 algoritmes, els quals serveixen, entre altres coses, per reconeixement facial, identificació d'objectes i, el que interessa en aquest projecte, fer el seguiment en temps real d'objectes. OpenCV és una llibreria que funciona amb Python, per tant, aquest llenguatge serà un dels que es faran servir per al desenvolupament del projecte.

## 2.4 Node-RED

Node-RED és un entorn de programació molt potent i idoni per desenvolupaments informàtics a alt nivell, sense la dificultat de la programació a nivell de codi [4]. És una plataforma de codi obert de IBM, basada en fluxos i nodes amb una extensa llibreria que els desenvolupadors i els fabricants de dispositius comparteixen [7]. Funciona com un servidor basat en Node.js i el llenguatge de programació està basat en JavaScript. Node-RED ofereix la possibilitat de programar de forma fàcil i intuïtiva a un servidor web, al qual es pot publicar un tauler virtual o dashboard amb una representació esquemàtica de l'estat del tauler a temps real.

## 2.5 PostgreSQL

Per connectar OpenCV i Python amb Node-RED es farà servir una base de dades. Les tres més importants actualment utilitzen el llenguatge SQL (Structured Query Language) i són [10]:

- MySQL

Amb el pas del temps ha esdevingut la base de dades més popular perquè va ser la opció per defecte de molts desenvolupadors. La seva corba d'aprenentatge és petita i permet gestionar de manera senzilla bases de dades relacionals.

- MariaDB

Sorgeix d'una bifurcació amb MySQL i per això és un bon substitut de MySQL, ja que la compatibilitat es manté però és unidireccional, MariaDB inclou algunes característiques diferents i no compatibles amb MySQL, com per exemple l'afavoriment de la flexibilitat i l'adaptabilitat.

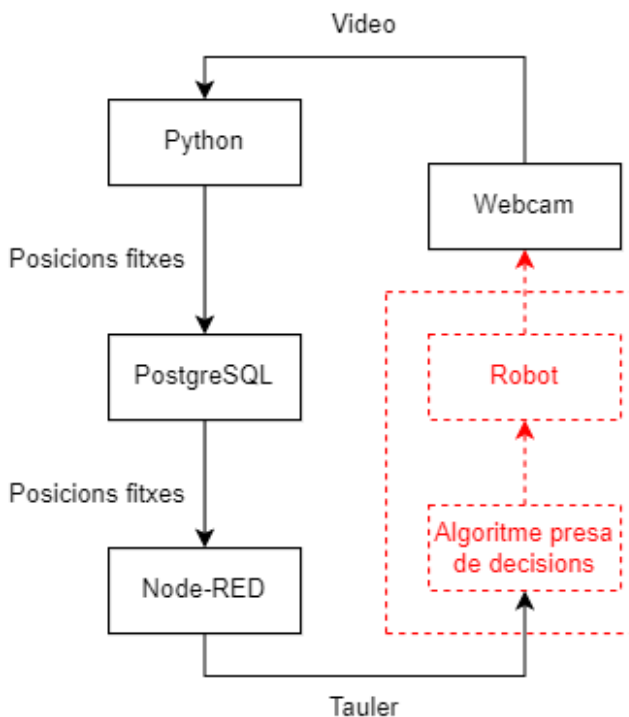
- PostgreSQL

Es tracta d'una base de dades relacional open source molt utilitzada a l'àmbit de la ciència de dades, els gràfics i la intel·ligència artificial, ja que és ideal per aplicacions de Python. Per tant, aquesta serà la base de dades a la qual s'enviarà la informació del tauler.

## 3 DESENVOLUPAMENT I PROGRAMACIÓ

### 3.1 Diagrama de treball

Per aconseguir els objectius proposats, el projecte es divideix en tres parts: la primera és processar el vídeo provinent de la webcam amb el llenguatge de programació Python i la llibreria OpenCV. Un cop la imatge sigui processada, la matriu amb la informació de l'estat del tauler es desa a una base de dades. Posteriorment, Node-RED recollirà la informació de la base de dades i mostrarà en una pàgina web un dashboard amb un esquema de la situació del tauler a temps real. El següent diagrama il·lustra els elements d'aquestes tres parts. Els blocs que es representen en color vermell no pertanyen a aquest projecte.



Imatge 3: Diagrama de treball. Font pròpia

### 3.2 Instal·lacions necessàries

Per desenvolupar aquest projecte, calen fer unes instal·lacions prèvies si no es disposa del programari necessari. El primer que s'ha de realitzar és una instal·lació del programa Python, que és el llenguatge en el que es programarà la primera part del projecte. A part de fer aquesta instal·lació, s'han d'instal·lar algunes llibreries com Numpy i OpenCV [14], aquesta darrera proporciona les eines de visió per computador. A més, s'ha d'instal·lar PostgreSQL [15], programa amb el que es crea la base de dades. Finalment, s'instal·la Node-RED. Dins de Node-RED s'han d'instal·lar diversos paquets: node-red-dashboard, node-red-contrib-ui-svg, node-red-contrib-drawsvg i node-red-contrib-postgrestor-next.

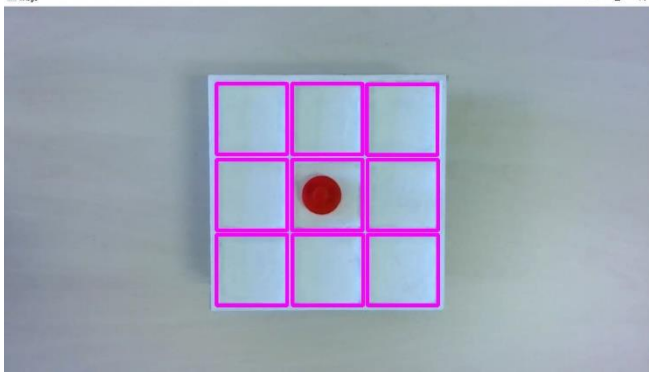
### 3.3 Divisió en imatges individuals

La primera part de la programació consisteix en dividir la imatge en subimatges per, posteriorment, analitzar el seu contingut. En aquest cas, es fa amb 9 quadrats, que posteriorment s'analitzaran de forma individual. Cada quadrat



del tauler està representat per un quadrat “virtual” que sigui exacte tant a la imatge d’on s’extreuen com als vídeos que es testejen (amb un tamany de 130x130).

Primerament, a partir d’un fitxer Python (TicTacToe.py) s’ha de crear l’arxiu on es desen les posicions del tauler, anomenat TicTacToePos. Aquest fitxer conté la posició dels quadrats que es dibuixen a la imatge del tauler. Per dibuixar un quadrat, cada vegada que es fa clic esquerre el programa dibuixa un quadrat de les dimensions esmentades prèviament i el desa al fitxer amb les posicions. Si per error un quadrat queda mal dibuixat, fent clic dret dins de l’àrea que comprén el quadrat en qüestió, l’esborra. Un cop s’han digitalitzat les nou posicions, quan s’executa de nou el fitxer TicTacToe.py, mostra la imatge inferior.



Imatge 4: Digitalització dels quadrats del tauler. Font pròpia

### 3.4 Anàlisi de les imatges individuals

En el moment que estan tots els quadrats del tauler representats de manera virtual a l’arxiu TicTacToePos, es crea i s’edita el fitxer principal per fer proves, que s’anomena main.py. Primerament es fa la crida a l’arxiu amb les posicions del tauler ja digitalitzades. Com que la imatge amb la que s’han digitalitzat els quadrats i els vídeos per realitzar les proves tenen la mateixa dimensió, els quadrats encaixen perfectament. La idea general del nostre algoritme és que a mesura que el vídeo es va reproduint, el programa retalla els nou quadrats de la imatge i els analitza un per un, convertint les imatges en píxels blancs i negres de manera que les peces corresponguin a píxels blancs i el tauler a píxels negres. L’algoritme analitza la quantitat de píxels blancs per determinar la presència de peça. Aquest valor serà diferent en funció de si no hi ha peça, hi ha un cercle o hi ha una creu. L’algoritme ha de marcar els quadrats amb un color representatiu en funció de si hi ha peça (i de quin tipus) o no. Es defineix un llindar mitjançant prova i error amb el que, depenent del número de píxels amb valor diferent a zero, es representarà el quadrat corresponent de color verd (no hi ha peça), blau (hi ha una creu) o vermell (hi ha un cercle). Com que es processen les nou imatges de forma simultània, en cas de no haver-hi peça, addicionalment incrementarà en una unitat el comptador d’espais lliures per cada quadrat sense peça. També es mostren a tots els quadrats quants píxels tenen amb un valor diferent a zero. La imatge 5 mostra el codi desenvolupat per representar els quadrats.

```
#si el número de píxels amb valor diferent a zero es menor a 900
#pinta de color verd el quadrat i incrementa el comptador
if count < 900:
    color = (0, 255, 0)
    thickness = 5
    spaceCounter += 1
#si no és menor a 900, hi ha peça
else:
    #si el número de píxels amb valor diferent a zero es menor a 1420
    #pinta de color vermell el quadrat
    if count < 1420:
        color = (0, 0, 255)
        thickness = 2
    #si el número de píxels amb valor diferent a zero es major a 1420
    #pinta de color blau el quadrat
    else:
        color = (255, 0, 0)
        thickness = 2
```

Imatge 5: Codi en Python amb els llindars definits. Font pròpia

### 3.5 Connexió amb la base de dades

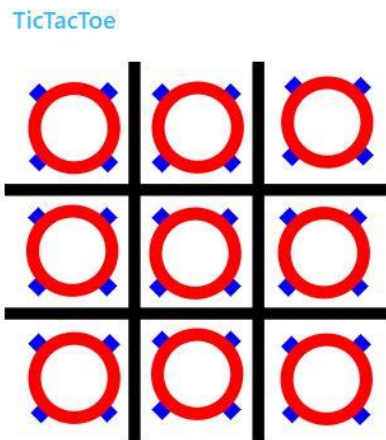
Ara el que s’ha de fer és establir connexió amb la base de dades, però abans cal crear la taula on es desen les posicions del tauler i l’estat de cada posició. Amb PostgreSQL i mitjançant sentències SQL es crea una taula amb dos columnes (una amb el número de posició i l’altre amb l’estat) i nou files (una per cada posició del tauler).

	idposition integer		status character varying
1	1	fBlava	
2	2	fBlava	
3	3	fBlava	
4	4	fBlava	
5	5	fBlava	
6	6	fBlava	
7	7	fBlava	
8	8	fBlava	
9	9	fBlava	

Imatge 6: Taula creada amb PostgreSQL. Font pròpia

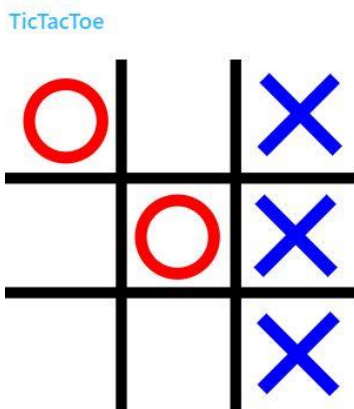
Un cop la taula està creada, es realitzen proves de connexió entre PostgreSQL i Python. Per fer-ho, es modifica el codi Python per tal que faci la connexió a la base de dades i desi la informació del tauler. El següent pas a realitzar quan aquestes proves donen un resultat positiu és combinar les dos parts (detecció de les peces en funció de l’entrada de vídeo i actualització a la base de dades) en un sol programa. Per fer-ho, es crea un nou fitxer Python (main\_withconnection.py) on s’implementarà, per una banda, el codi per connectar-se i actualitzar els valors de les posicions a la base de dades i, per altra, el que estableix l’estat de les posicions en funció del que detecta al vídeo. Es comença establint la connexió amb la base de dades, al mateix temps que es reproduïx el vídeo. El tractament del vídeo és igual al que es feia abans, cada quadrat es retalla i es processa de forma independent però al mateix temps. En aquest cas, es declara una variable de tipus llista amb nou valors (un per cada posició del tauler) inicialitzats amb el valor “blank”. La distinció de l’estat de cada posició es fa de la mateixa manera que abans (mitjançant un llindar), però en aquest cas es desa l’estat de la posició dins d’aquesta nova llista en funció del color del quadrat: blank (no hi ha peça), fBlava (hi ha creu) o fVermella (hi ha cercle). Seguidament, cada element d’aquesta llista actualitza la seva posició corresponent dins de la base de dades amb l’estat actual.

Aquest darrer pas consisteix a dissenyar un flow a Node-RED que sigui capaç de llegir la informació de la base de dades per poder mostrar-la a un dashboard. Aquest dashboard es crea mitjançant un editor de codi SVG (Scalable Vector Graphics), que és un format basat en etiquetes similar a HTML i compatible amb codi HTML. Aquest dashboard representa els nou quadrats del tauler tal com es pot veure a la següent imatge.



Imatge 7: Dashboard a Node-RED amb les peçes sobreposades.  
Font pròpia

El format SVG permet fer servir un atribut anomenat “visibility”, que pot obtenir dos valors: “visible” i “hidden”. Per tant, aquest atribut permet mostrar i amagar els dibuixos de la imatge anterior en funció de la taula de la base de dades. Segons els valors rebuts de la base de dades, la funció determina el valor de cada un dels 9 quadrats, que pot ser: blank (no hi ha fitxa), fBlava (hi ha fitxa blava) i fVermella (hi ha fitxa vermella). L’aspecte del dashboard per un cas particular és el següent.



Imatge 8: Cas particular de la partida. Font propia

Node-RED llegeix de forma periòdica el contingut de la base de dades mitjançant una sentència SQL per seleccionar-la tota. La taula és retornada com un objecte, del qual s'ha d'extreure la posició i l'estat dels nou elements. A continuació es defineix una funció que, per cada posició del tauler, mostri o amagui les fitxes depenent del valor rebut, de tal forma que les posicions al vídeo es puguin reflexar al dashboard.

Amb això conclou tot el treball desenvolupat.

## 4.1 Mètode d'avaluació

Aquest projecte es pot avaluar mitjançant el grau compliment dels objectius plantejats a l'apartat 1.1. Per tant, basant-nos en aquests objectius:

- Objectiu 1: Estudi de l'estat de l'art.

Bons resultats, tot i que s'ha anat ampliant a mesura que es desenvolupava el treball.

- Objectiu 2: Estudi d'elements hardware i software necessaris.

Igual que l'objectiu 1, quan s'ha començat a desenvolupar s'ha hagut de modificar en algunes parts.

- Objectiu 3: Desenvolupar l'escenari de proves i programar.

La programació s'ha realitzat correctament i s'ha assolit l'objectiu esperat.

- Objectiu 4: Realització de tests.

Els tests han funcionat bé, però no s'han pogut realitzar proves amb vídeos a temps real.

- Objectiu 5: Redactar la memòria i preparar la presentació.

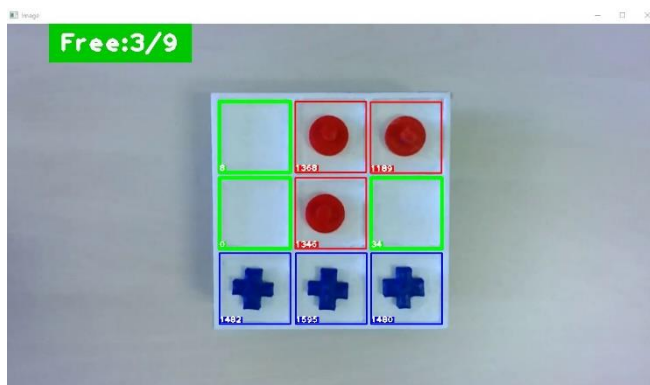
Assolit.

En relació a la planificació temporal, s'ha endarrerit lleugerament respecte al que estava previst, especialment a l'objectiu 3 que s'ha començat a realitzar al mes de maig de forma conjunta amb l'objectiu 4.

## 4.2 Resultats obtinguts

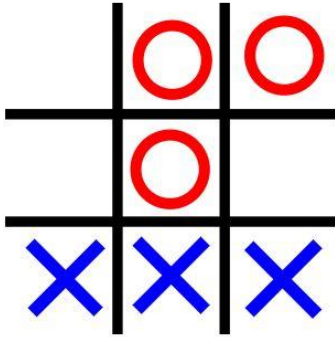
Els resultats obtinguts són tres: Un programa en Python que és capaç de detectar les peces d'un tauler de tres en ratlla i mostrar-les per pantalla, una base de dades SQL totalment funcional a la qual es pot enviar l'estat de cada cel·la a mesura que el vídeo s'està reproduint i, finalment, una interfície gràfica basada en JavaScript a la que es pot visualitzar el tauler a temps real gràcies a la informació que arriba de la base de dades. Els resultats de les proves amb tres vídeos diferents són els següents.

### 4.2.1 Vídeo 1

[illegible]

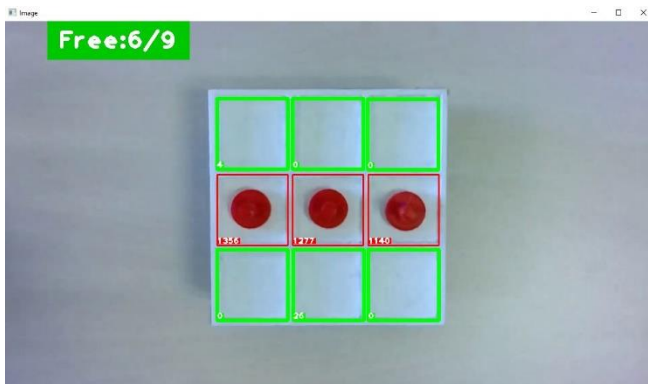


TicTacToe



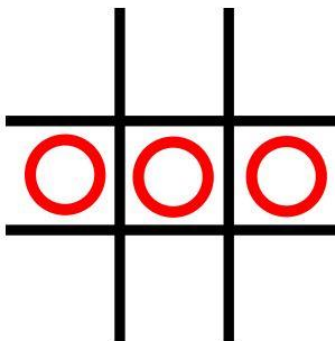
Imatges 9, 10 i 11: Resultat de la programació en Python, variable que s'envia a la base de dades i dashboard. Font pròpia

#### 4.2.2 Vídeo 2



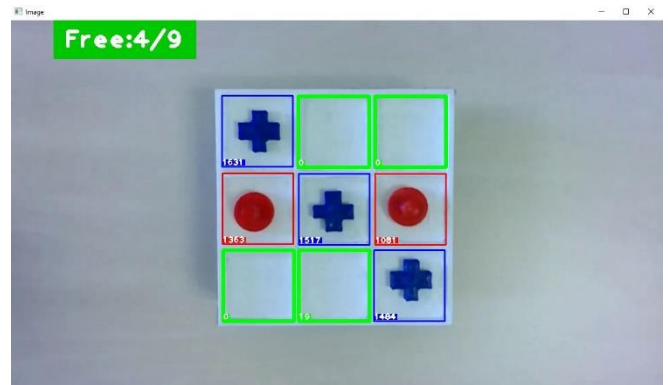
```
[ 'blank', 'blank', 'blank', 'fVermella', 'fVermella', 'fVermella', 'blank', 'blank', 'blank' ]
[ 'blank', 'blank', 'blank', 'fVermella', 'fVermella', 'fVermella', 'blank', 'blank', 'blank' ]
[ 'blank', 'blank', 'blank', 'fVermella', 'fVermella', 'fVermella', 'blank', 'blank', 'blank' ]
[ 'blank', 'blank', 'blank', 'fVermella', 'fVermella', 'fVermella', 'blank', 'blank', 'blank' ]
[ 'blank', 'blank', 'blank', 'fVermella', 'fVermella', 'fVermella', 'blank', 'blank', 'blank' ]
[ 'blank', 'blank', 'blank', 'fVermella', 'fVermella', 'fVermella', 'blank', 'blank', 'blank' ]
[ 'blank', 'blank', 'blank', 'fVermella', 'fVermella', 'fVermella', 'blank', 'blank', 'blank' ]
[ 'blank', 'blank', 'blank', 'fVermella', 'fVermella', 'fVermella', 'blank', 'blank', 'blank' ]
[ 'blank', 'blank', 'blank', 'fVermella', 'fVermella', 'fVermella', 'blank', 'blank', 'blank' ]
[ 'blank', 'blank', 'blank', 'fVermella', 'fVermella', 'fVermella', 'blank', 'blank', 'blank' ]
```

TicTacToe



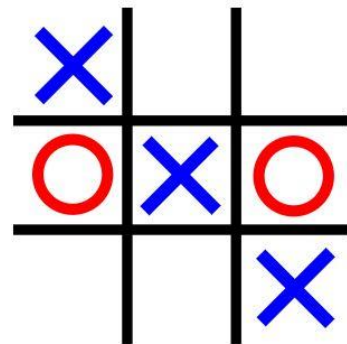
Imatges 12, 13 i 14: Resultat de la programació en Python, variable que s'envia a la base de dades i dashboard. Font pròpia

#### 4.2.3 Vídeo 3



```
[ 'fBlava', 'blank', 'blank', 'fVermella', 'fBlava', 'fVermella', 'blank', 'blank', 'fBlava' ]
[ 'fBlava', 'blank', 'blank', 'fVermella', 'fBlava', 'fVermella', 'blank', 'blank', 'fBlava' ]
[ 'fBlava', 'blank', 'blank', 'fVermella', 'fBlava', 'fVermella', 'blank', 'blank', 'fBlava' ]
[ 'fBlava', 'blank', 'blank', 'fVermella', 'fBlava', 'fVermella', 'blank', 'blank', 'fBlava' ]
[ 'fBlava', 'blank', 'blank', 'fVermella', 'fBlava', 'fVermella', 'blank', 'blank', 'fBlava' ]
[ 'fBlava', 'blank', 'blank', 'fVermella', 'fBlava', 'fVermella', 'blank', 'blank', 'fBlava' ]
[ 'fBlava', 'blank', 'blank', 'fVermella', 'fBlava', 'fVermella', 'blank', 'blank', 'fBlava' ]
[ 'fBlava', 'blank', 'blank', 'fVermella', 'fBlava', 'fVermella', 'blank', 'blank', 'fBlava' ]
[ 'fBlava', 'blank', 'blank', 'fVermella', 'fBlava', 'fVermella', 'blank', 'blank', 'fBlava' ]
[ 'fBlava', 'blank', 'blank', 'fVermella', 'fBlava', 'fVermella', 'blank', 'blank', 'fBlava' ]
```

TicTacToe



Imatges 15, 16 i 17: Resultat de la programació en Python, variable que s'envia a la base de dades i dashboard. Font pròpia

#### 4.3 Discussió

En general, el treball realitzat respon adequadament a la necessitat plantejada inicialment, que era desenvolupar un programa capaç de detectar peçes d'un tauler de tres en ratlla. Això s'ha aconseguit mitjançant una webcam, amb la que s'han gravat diferents vídeos de prova que han servit de base per poder desenvolupar el programa que pugui reconèixer la posició de les fitxes al tauler.

Cal esmentar que al començament les creus eren de color blanc, però per a l'algorisme era més complicat detectar si hi havia peça o no, ja que el tauler és de color blanc i les peçes es "camuflaven". És per això que posteriorment es va optar per utilitzar peçes de color blau, que eren clarament distingibles del tauler.

#### 5 CONCLUSIÓ I LÍNIES DE CONTINUACIÓ

La intel·ligència artificial és una eina molt potent que està permetent un desenvolupament tecnològic de dimensions considerables i de gran impacte social. Utilitzant eines de visió per computador s'ha aconseguit desenvolupar un programa que sigui capaç d'analitzar l'estat d'un tauler amb nou caselles, però es pot extrapolar a molts casos d'ús.

Personalment, aquest projecte ha sigut una gran oportunitat per poder desenvolupar competències professionals i endinsar-me més en el món de la intel·ligència artificial, en aquest cas treballant la visió per computador. Ha sigut molt interessant i motivador.

A aquest projecte s'ha aconseguit desenvolupar un programa amb el que es pugui veure el que succeeix al tauler a temps real i es mostri per un dashboard. Seguint una línia de continuïtat per un projecte el curs vinent, el robot haurà de poder calcular, mitjançant un algoritme de presa de decisions i control dels moviments, la millor jugada, de manera que aquest pugui jugar per sí mateix.

## AGRAÏMENTS

En primer lloc, agrair al tutor del treball Carles Pedret Ferré per la seva feina i disposició a l'hora de desenvolupar aquest treball. També agrair a José Antonio Corral Palmero i Yousra Benchikh pel bon treball que van desenvolupar l'any passat, ja que ha esdevingut la base d'un projecte multidisciplinar, el qual combina els fonaments teòrics i tècnics de la robòtica, l'aplicació d'intel·ligència artificial mitjançant eines de visió per computador i, finalment, llenguatges de programació d'aplicacions com JavaScript.

## REFERÈNCIES

- [1] Corral Palmero, J. A., Benchikh, Y., & Pedret Ferré, C. (Juny de 2022). *Construcción y programación de un brazo robótico*. DDD UAB. <https://ddd.uab.cat/record/264107>
- [2] Telefónica. (25 de novembre de 2022). *Qué es y cómo funciona la Inteligencia Artificial*. <https://www.telefonica.com/es/sala-comunicacion/blog/que-es-y-como-funciona-la-inteligencia-artificial/>
- [3] Ovrmind. (16 d'octubre de 2019). *Conoce las 10 principales ramas de la inteligencia artificial*. <https://ovrmind.com/ramas-inteligencia-artificial/>
- [4] Node-RED. (18 de gener de 2023). A Wikipedia. <https://es.wikipedia.org/w/index.php?title=Node-RED&direction=prev&oldid=151378567>
- [5] IBM Cloud Pak for Data. (15 de maig de 2023). *Características en desuso y cambios de plan de servicios*. <https://data-platform.cloud.ibm.com/docs/content/wsj/getting-started/serve-vice-updates.html#vr>
- [6] IBM. (s.f.). *¿Qué es la visión artificial?* <https://www.ibm.com/es-es/topics/computer-vision>
- [7] OpenJS Foundation. (s.f.). *Creating your first flow*. <https://nodered.org/docs/tutorials/first-flow>
- [8] Alternativas a... (s.f.). *Alternativas a OpenCV*. <https://alternativas-a.com/opencv/>
- [9] Gaudenz Boesch. (s.f.). *The 12 Most Popular Computer Vision Tools in 2023*. Viso.ai. <https://viso.ai/computer-vision/the-most-popular-computer-vision-tools/>
- [10] Salman Ravoof. (24 de febrer de 2023). *Lo Mejor en Software de Bases de Datos de Código Abierto: Las 10 Mejores Opciones*. Kinsta. <https://kinsta.com/es/blog/bases-de-datos-de-codigo-abierto/>
- [11] Yúbal Fernández. (13 de juny de 2023). *ChatGPT: qué es, cómo usarlo y qué puedes hacer con este chat de inteligencia artificial GPT*. Xataka. <https://www.xataka.com/basics/chatgpt-que-como-usarlo-que-puedes-hacer-este-chat-inteligencia-artificial>
- [12] David Bernal Raspall. (21 de març de 2023). *Siri, guía a fondo de todo lo que podemos hacer en 2023 con el asistente de voz de Apple*. Applesfera. <https://www.applesfera.com/nuevo/siri-guia-a-fondo-todo-que-podemos-hacer-2023-asistente-voz-apple>
- [13] Jon Larkin Alonso. (16 de juny de 2022). *¿Qué es TensorFlow y para qué sirve?* Incentro. <https://www.incentro.com/es-ES/blog/que-es-tensorflow>
- [14] OpenCV. (6 de maig de 2023). A Wikipedia. <https://es.wikipedia.org/w/index.php?title=OpenCV&oldid=150996137>
- [15] PostgreSQL. (s.f.). *About*. <https://www.postgresql.org/about/>