

ELS ENGRANATGES DE LA INTEL·LIGÈNCIA ARTIFICIAL



Autor: Guillem Cabré Farré
Tutors: • Joan Àlamo Bretcha
• Montserrat Beltran i Berengué
Curs: 2n Batxillerat 2021-22
INS Arquitecte Manuel Raspall
Àmbit científicotecnològic

29 d'octubre de 2021

Just the nature of the AI that they're building is one that crushes all humans at all games. "Elon Musk, 29-7-2020"

La naturalesa de la intel·ligència artificial que estan construint és la que supera a tots els humans en tots els jocs.

Musk va expressar la seva màxima preocupació per la intel·ligència artificial de Google.

AGRAÏMENTS

Primer de tot, agrair al Joan Àlamo per posar al meu abast els seus coneixements sobre la intel·ligència artificial i pel seguiment del treball que em va fer. També, agrair als amics i familiars per la seva sincera opinió sobre el contingut i ajudar-me a millorar alguns aspectes del treball. Per últim, m'agradaria fer un agraïment especial a la Montse Beltran, per agafar el meu projecte a principis de juliol i per aconsellar-me durant tot l'estiu. I agrair-li també el seu entusiasme, la seva positivitat i els incomptables consells que m'ha donat.

ABSTRACT

Hoy en día, una mayoría conoce este término: inteligencia artificial, pero solamente una minoría sabe con certeza que es y cómo funciona. Se caracteriza por ser un recurso con un potencial increíble, capaz de optimizar todo tipo de mecanismos informáticos. Por esta razón, las empresas pioneras en el mundo de la informática intentan sacarle el mayor potencial.

En este proyecto de investigación, mi objetivo es introducir al lector en los fundamentos de la inteligencia artificial, haciendo énfasis en los diferentes tipos que existen y adentrarse en el tema ético que la rodea. Encontraréis una gran cantidad de ejemplos, que permiten al lector asociar este complejo concepto con nuestras vidas cotidianas y entender cómo funcionan los distintos tipos. Además, he explicado muchos procedimientos informáticos con el fin de encaminar al lector hacia un mejor entendimiento de la inteligencia artificial.

Finalmente, he creado un programa informático relacionado con el tema, que permite ver el resultado gráficamente, con el objetivo de complementar la explicación. Su funcionamiento se explica al final del proyecto.

Nowadays, a majority knows about this term: artificial intelligence, but only a minority knows for sure what it is and how it works. It is characterized by being a resource full of potential, capable of optimizing all kinds of computers mechanisms. For this reason, the pioneering companies within the scope of informatics are trying to get the most out of it.

In this research project, my goal is to introduce the reader to the basics of artificial intelligence, emphasizing the different types that exist and introducing them into the ethical topic that surrounds it. You will find a large number of examples, which allow the reader to associate this complex concept with our daily lives and to understand how the different types work. In addition, I have explained many computer procedures to guide the reader towards a better understanding of artificial intelligence.

Finally, I have created a computer program related to the topic, which allows you to see the result graphically, to complement the explanation. Its operation is explained at the end of the project.

ÍNDEX

AGRAÏMENTS	... 3
ABSTRACT	... 4
1. INTRODUCCIÓ	... 6
1.1. Motivacions	... 6
1.2. Objectius	... 6
1.3. Hipòtesi	... 7
1.4. Metodologia	... 7
2. MARC TEÒRIC	... 9
2.1. Què significa intel·ligència artificial?	... 9
2.2. D'on sorgeix?	... 10
2.3. Com funciona?	... 11
2.4. Quins tipus hi ha?	... 13
2.5. Quines aplicacions té?	... 14
2.6. Filosofia i ètica	... 18
3. MARC APLICAT	... 22
3.1. Introducció del programa	... 22
3.1.1. Limitacions	... 25
3.2. Funcionament	... 26
3.2.1. Menú	... 26
3.2.2. Intel·ligència artificial	... 28
3.3. Possibles aplicacions	... 31
4. CONCLUSIONS	... 33
4.1. Valoració personal	... 34
5. WEBGRAFIA	... 35
5.1. Altres enllaços	... 35
7. GLOSSARI	... 37
8. ABREVIATURES I SÍMBOLS	... 38

1. INTRODUCCIÓ

1.1. MOTIVACIONS

En primer lloc, vaig escollir aquest tema per fer el treball de recerca (TR)^[1] per encaminar-me cap al grau que vull estudiar, enginyeria informàtica. Cursant aquesta enginyeria s'adquireixen els coneixements per treballar envers el *software*⁽⁶⁾ i *hardware*⁽³⁾. Estava segur que volia fer un treball de recerca que girés al voltant de la programació. Principalment, vaig voler fer un programa capaç d'encriptar arxius, però en parlar-ne amb el primer tutor, el Joan Àlamo, ens vam adonar que una intel·ligència artificial (IA)^[2] podria ser molt més interessant, ja que avui en dia és un tema molt punter amb un cert misticisme al seu voltant, que porta a la gent al desconeixament, a la desconfiança i al dubte.

En segon lloc, les principals motivacions que tinc per dur a terme aquest treball de recerca són:

- Preparar-me per al grau que vull fer en un futur. Aquest estudi m'ajudarà a tenir més autonomia personal en la recerca de recursos i a complementar els meus coneixements de programació.
- Aprenentatge d'un llenguatge de programació que desconec i que avui en dia s'està utilitzant un munt per a tota mena de programes. També aprendré i utilitzaré noves funcions dels llenguatges de programació, que em permetran fer la part pràctica del meu TR.

És a dir, l'execució del treball de recerca m'aproparà cap als estudis professionals que vull cursar. Altrament, em permetrà tenir un coneixement més extens sobre les intel·ligències artificials, els llenguatges de programació i sobre els processos interns dels ordinadors.

1.2. OBJECTIUS

Els objectius principals d'aquest treball estan molt relacionats amb les motivacions. Vull que em sigui útil pel meu futur, per tant, intentaré fer-lo el millor possible, ja que és una molt bona oportunitat per aprendre, per fer recerca i per millorar la qualificació mitjana del batxillerat. Els objectius són:

- Conèixer els diferents tipus d'intel·ligències artificials.

- Comprendre el funcionament intern d'un tipus en concret d'intel·ligència artificial.
- Conèixer el perill de la intel·ligència artificial per a la humanitat.
- Aprendre un nou llenguatge de programació.
- Fer un programa que tingui una intel·ligència artificial.

1.3. HIPÒTESI

Fent recerca és possible crear una intel·ligència artificial des de zero.

1.4. METODOLOGIA

El primer que faré en el meu treball de recerca serà buscar informació sobre les intel·ligències artificials, en aquest apartat vull trobar què són, com funcionen i concretar els seus orígens. A més a més, buscaré informació sobre el funcionament d'aquestes i els diferents tipus que hi ha. Per a cada un faré una breu explicació i l'associaré amb l'aplicació que li correspongui.

Finalment parlaré del conflicte que ens genera, plantejant la pregunta: ens ha de preocupar la intel·ligència artificial?

Per començar a enfocar el treball cap a la part pràctica cercaré informació sobre el llenguatge informàtic que empraré, el *C Sharp (C#)*^[3]. Amb la recerca intentaré aprendre la sintaxi i les bases del llenguatge. Segurament hauré de buscar sobre moltíssims més aspectes del tema mentre el vagi elaborant, però les bases ja les tindrè. També m'hauré d'informar sobre els diferents tipus d'intel·ligències artificials associades al *PathFinding*^[5], per tal d'escollir un i adaptar-lo a les meves necessitats. Mitjançant esquemes, fotografies i vídeos intentaré exposar-lo de manera clara i entenedora per a un públic, avesat o no, a treballar en l'àmbit de la informàtica.

Un cop finalitzada la recerca, elaboraré un programa que tingui una intel·ligència artificial dins seu. La seva funció serà cercar la ruta més ràpida d'un punt A a un punt B. L'usuari tindrà l'opció de crear el mapa, és a dir, podrà col·locar obstacles en un pla pels quals no podrà passar la ruta i també podrà escollir un mapa predeterminat amb obstacles ja establerts. Les aplicacions del programa seran explicades al marc pràctic.

Com qualsevol programa informàtic, aquest tindrà una planificació prèvia. El primer que faré serà una anàlisi. En aquesta fase hauré de buscar quines són

les meves necessitats a satisfer. Després faré el disseny, on pensaré en quin programa escriuré el codi, com vull que l'usuari vegi el programa i per quin sistema operatiu vull que funcioni. A continuació, començaré a codificar, per tal d'aconseguir el resultat desitjat. Aquesta fase és la més llarga i important, tot i que totes són imprescindibles. El següent pas és l'explotació, on estaré provant el programa a la cerca d'errors i possibles millores. Finalment, solucionaré els problemes que hagi trobat i intentaré millorar la seva eficàcia.

El treball també comptarà amb un glossari i amb un llistat d'abreviatures i símbols. En el text, apareixerà un nombre entre parèntesis, com a subíndex davant d'un mot poc conegut, per tal d'indicar que hi ha una definició de la paraula al glossari. D'altra banda, les abreviatures i els símbols estaran acompanyats del seu significat entre parèntesis, o a l'inrevés. Això dependrà de quina de les dues formes sigui més coneguda i per tant, més utilitzada. En alguns casos no apareixerà el significat, en el text. L'objectiu d'això és facilitar la fluïdesa de la lectura, ja que són mots molt tècnics i no són essencials pel lector. Tot i això estaran explicats al llistat d'abreviatures i símbols. A més, algunes definicions estaran en cursiva i anglès, les seves traduccions estaran al llistat d'abreviatures i símbols.

2. MARC TEÒRIC

2.1. QUÈ SIGNIFICA INTEL·LIGÈNCIA ARTIFICIAL?

La intel·ligència artificial és una part de la informàtica, que es dedica al desenvolupament d'algoritmes, els quals permeten a la màquina prendre una decisió entre diferents opcions. Un algoritme és un procediment que es repeteix un nombre determinat de vegades. Un exemple d'això podria ser un programa que vol dividir X vegades, un nombre per un altre nombre. Quan a un algoritme li afegim variables, condicionals i altres funcions, podem dir a terme una tasca molt complicada en molt poc temps, comparat amb el que tardaria un humà. És important explicar que cada cicle que fa l'algoritme es diu iteració, quan s'escriu en la programació s'acostuma a expressar amb una "i".

Un ordinador no pot pensar com ho faria un humà i per aquesta simple raó és molt complicat determinar si realment es comporta de manera intel·ligent. Hi ha moltíssimes definicions del que és exactament una intel·ligència artificial, les més acceptades són les de John McCarthy, Andreas Kaplan amb Michael Haenlein i Marvin Minsky. Tots ells són matemàtics i científics de renom. A més, John McCarthy és considerat un dels pares de la intel·ligència artificial juntament amb Marvin Minsky i Claude Shannon.

McCarthy ho defineix així: és fer que una màquina es comporti d'una manera que seria considerada intel·ligent en un ésser humà. Aquesta és la definició més estesa, tot i que les altres també són importants per adquirir altres punts de vista. Andreas Kaplan i Michael Haenlein afirmen: és la capacitat que té un sistema per interpretar dades externes correctament, aprendre d'aquestes dades i fer servir els coneixements adquirits per completar tasques i assolir objectius específics mitjançant una adaptació flexible.

Finalment, Marvin Minsky ho defineix com: la realització de sistemes informàtics amb un comportament que en l'ésser humà qualificaríem d'intel·ligent. El punt de vista de Minsky és molt similar al de McCarthy. L'única diferència és que McCarthy parla de màquines en general i Minsky d'ordinadors.

2.2. D'ON SORGEIX?

Alan Turing, és conegut com l'avi de la intel·ligència artificial degut a la seva aportació durant la Segona Guerra Mundial.

Sota les ordres del servei d'intel·ligència britànic va crear una màquina capaç de desxifrar els missatges encriptats de les forces militars alemanyes. Aquest projecte va rebre el nom d'Enigma. No va ser fàcil, perquè necessitava un gran pressupost i molt de temps, dues coses de les quals no disponia. Finalment amb un equip de matemàtics i científics va aconseguir crear una màquina capaç de resoldre el problema. Era tan gran com un menjador i no tenia cap mena d'electrònica, tot estava basat en engranatges i mecanismes mecànics. Mitja dècada després va escriure un article a la revista Mind anomenat: *Computing machinery and intelligence* (maquinària computacional i intel·ligència), allà es formula la pregunta: poden pensar les màquines? A l'article es proposa un mètode per determinar si realment una màquina pensa. Aquest mètode, amb els anys, va adquirir el nom de Test de Turing. El seu funcionament és el següent: un subjecte, anomenat avaluador, fa unes preguntes, que arriben a una persona humana i a una màquina. Per determinar que la màquina és intel·ligent, l'avaluador ha de ser incapaç de distingir les dues respostes. Per tant, les respostes de la màquina seran coherents i similars a com serien les d'un humà.

Hi ha investigadors i historiadors que afirmen que el punt de partida de la IA fou l'any 1956, quan els pares de la intel·ligència artificial, John McCarthy, Marvin Minsky i Claude Shannon van encunyar formalment el terme durant la conferència de Dartmouth, com: la ciència i l'enginy de fer màquines intel·ligents, especialment programes de càlcul intel·ligent.

2.3. COM FUNCIONA?

La IA funciona a partir d'algoritmes escrits pel programador, que contenen diferents instruccions. Les més utilitzades són:

- **Variables:** emmagatzemen valors. Hi ha molts tipus de variables, les més destacades són:

- **"int"**: emmagatzema nombres sencers.
- **"bool / boolean"**: pot guardar 0 o 1, és a dir fals o cert.
- **"string"**: és el tipus de variable que pot emmagatzemar text.
- **"float"**: és molt similar a la primera.

```
int a = 123;
bool b = false;
string c = "Un, dos, tres";
float d = 123.123f;
```

Fig. 1: Sintaxi de les variables: int, bool, string i float.

L'única diferència que hi ha és que

aquesta pot guardar decimals. El nombre ha de tenir sempre una "f" al final perquè el programa el pugui reconèixer com a decimal.

- N'hi ha més com per exemple: "double", "char", "var", "long", però s'utilitzen en menor mesura.

Per escriure variables dins d'una sola variable s'utilitza la instrucció "array", aquesta és més complicada i requereix certs coneixements per entendre-la, però de manera resumida, serveix per fer diferents variables dins d'una altra.

A l'hora d'escollir-ne una el programador ha de rumiar sobre les seves necessitats, ja que per escriure un valor sencer com per exemple "23" pot fer-ho amb "int", "float" o "string". En aquest cas, la millor opció seria utilitzar l'"int", que és la que consumeix menys recursos, perquè, tot i tenir ordinadors molt potents, el programador ha de tenir molta cura de fer un programa eficient i ràpid, si no es fa així l'ordinador es podria arribar a saturar. Podem veure la seva sintaxi a la Fig. 1.

- **Loops**: són instruccions que permeten una repetició d'ordres en forma de bucle. Un ordinador és incapaç de fer una tasca infinitament, per tant, el programador ha d'escriure el nombre de vegades que vol que es repeteixi o el resultat que vol obtenir a partir dels loops. Totes les instruccions que hi van a dins s'anomenen "cos". Hi ha diferents tipus de loops:
 - **"for"**: aquesta instrucció s'utilitza quan se sap quantes vegades es vol repetir el seu cos o bé quan es vol utilitzar el valor d'una variable com a nombre de repeticions.
 - **"while"**: s'utilitza quan se sap el resultat que donarà la repetició. Per exemple, per saber quantes repeticions ha tingut el loop per dur a terme una operació.

- Hi ha d'altres, com per exemple: “foreach”, que s'empra per repetir el cos depenent del nombre de variables dins d'una “array”.
- **Condicionals:** com bé indica el seu nom, tenen la funció de crear una condició. Quan es compleix, es fa l'ordre que hi ha dins la condició. La Fig. 2 representa la condició: si 'a' (que és una variable) és menor o igual que dos, fes una acció. Al final de l'“if” el programador pot escriure un “else”. Aquest també té un cos que s'activa quan no es compleix l'“if”. En el cas de la Fig. 2, si 'a' fos 5 i hi hagués un “else”, s'activaria el cos de l'“else”. Fa la funció: si no es compleix la condició executa això.

```
if (a <= 2)
{
    //Fer una acció
}
```

Fig. 2: Representació en C# d'un condicional.

La unió d'aquests elements és la que construeix una IA, evidentment hi ha moltíssimes més funcions que varien depenent el llenguatge, però podríem dir que aquestes són les bases de la programació i que sabent això ja es podria crear algun programa amb cara i ulls.

Cada iteració dins del loop principal de la IA s'encarrega d'apropar-se al resultat desitjat. Per exemple, hi ha moltíssimes intel·ligències artificials que a partir

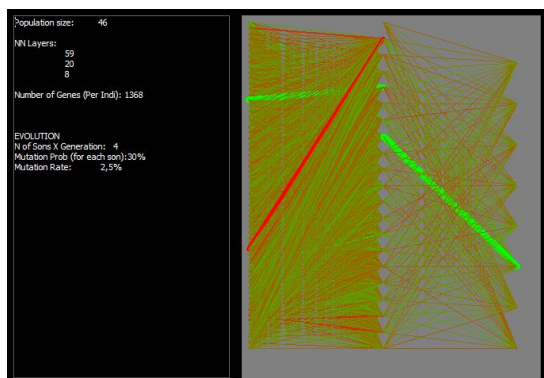


Fig. 3: Representació gràfica de la xarxa neuronal d'una IA. Generació ≈200.

d'algoritmes són capaces d'optimitzen el comportament d'un cos creat per l'usuari. Aquest cos consta de barres i músculs. La IA dona indicacions als músculs de com comprimir-se i dilatar-se, per tal de fer diferents accions, com ara: saltar, caminar, córrer...

Jo he utilitzat la IA que es diu: LEARN_2Dsticks_SimpleNN. La IA crea una xarxa neuronal que estableix com

s'ha de comportar un cos davant de diferents estímuls. Podem veure-ho representat en un mapa, el de la Fig. 3. A l'esquerra del mapa podem observar tots els estímuls que capta i a la dreta la resposta a aquests. Quan la IA fa un canvi en la xarxa neuronal es suma una generació, com si es

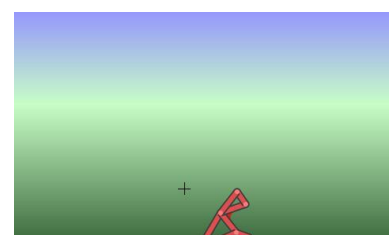


Fig. 4: Cos utilitzat per fer funcionar la IA.

tractés del món animal. La IA dona instruccions aleatòries als músculs i intenta buscar patrons. Al final de la generació es queda amb el cos que més lluny ha arribat i fa petits canvis en les instruccions que ha donat per veure si algun de la següent generació arriba més lluny. El model que vaig escollir per fer funcionar aquesta IA és un cangur. A la natura utilitzen la cua per propulsar-se i córrer més de pressa. Al cap d'unes 100, la intel·ligència artificial generacions se n'adona i ho comença a intentar. Això ho podem veure en el quart enllaç de l'apartat: altres enllaços. El cangur és el de la Fig. 4, les zones més vermelles són els músculs i les rosades les barres.

2.4. QUINS TIPUS HI HA?

Els experts classifiquen totes les intel·ligències artificials en 4 tipus. Representen el seu grau de similitud amb la ment humana. El primer és el que menys s'assimila i l'últim el que més. Aquestes quatre categories van ser creades per Arend Hintze, un professor de biologia integrada i ciències de la computació de la Universitat de Michigan. El seu objectiu és crear màquines intel·ligents capaces de prescindir d'un ensenyament humà previ. És a dir, vol crear una màquina capaç d'aprendre per ella sola. Per això, diu que aquesta màquina o robot hauria de tenir el mateix procés d'aprenentatge que el d'una persona.

- Les **màquines reactives** són els sistemes d'intel·ligència artificial que tenen un comportament reactiu. Això vol dir que a partir d'estímul ja observats pel programador, duen a terme una reacció. No tenen la capacitat suficient per crear records ni poden utilitzar experiències passades. Un exemple d'aquest tipus és el Deep Blue, un superordinador creat per IBM. El 1990 aquest ordinador va ser capaç de vèncer al vigent campió internacional d'escacs, Garry Kasparov. Deep Blue era capaç de processar fins a 200 milions de moviments cada segon.
- Un altre tipus d'IA són les anomenades de **memòria limitada**. Aquestes són capaces de cercar informació dels seus records del passat de forma limitada i temporal. Emmagatzemen la informació que han recollit durant cert temps i modifiquen els seus algoritmes per tal d'optimitzar el seu comportament. Un exemple d'aquest tipus d'IA són els programes integrats en els cotxes autònoms.

- Quan les IAs a més de comprendre el món, treuen conclusions sobre altres agents o entitats, s'anomenen de **teoria de la ment**. En psicologia, això implica la comprensió que les persones, criatures o objectes puguin tenir pensaments i emocions que afectin el seu propi comportament. És a dir, les IAs de classe "teoria de la ment" són capaces de reaccionar de diverses maneres segons el seu estat emocional. Són capaces de tenir una interacció social.
- Finalment, l'**autoconsciència** és el tipus d'intel·ligència artificial capaç de construir sistemes que poden formar representacions sobre ella mateixa. El problema d'aquest tipus d'IA és que el programador ha de tenir un coneixement complet sobre la consciència i avui en dia ningú en té. Realment estem bastant lluny d'aconseguir IAs capaces de tenir consciència sobre elles mateixes. Tot i això, enginyers i neuròlegs de tot el món s'estan esforçant per apropar-se a aquesta etapa.

2.5. QUINES APLICACIONS TÉ?

Les aplicacions de la intel·ligència artificial són infinites, avui en dia s'empren en tota mena d'àmbits. Podem veure-ho: en el buscador de centenars de milions d'usuaris que tenen Facebook, YouTube, Instagram... en coets, com els de SpaceX, que són capaços de bascular el seu pes a partir de tres motors articulats i aterrar sobre les seves potes verticalment; programes intel·ligents, que poden crear imatges a partir de paraules (es veurà posteriorment); anuncis, que apareixen depenent dels gustos o necessitats de l'usuari; automòbils, com els de la marca Tesla, que són capaços de conduir-se per si sols...

Com hem pogut veure, cada vegada apareixen en més àmbits i evidentment estan suposant un increment en la seguretat de la nostra societat. El març de 2018 va haver-hi un accident amb un pilot automàtic de Tesla, va morir una persona. Tesla va ser molt criticat, però realment, els vehicles de conducció autònoma tenen una reducció del 50% en el nombre d'accidents fatals comparats amb els vehicles convencionals.

Un exemple del seu potencial és la nova i prometedora IA anomenada VQGAN+CLIP^[4]. L'objectiu d'aquesta IA és crear imatges a partir de text. Són dues intel·ligències artificials treballant conjuntament, la primera s'encarrega de comprimir i descomprimir la imatge en patrons i la segona analitza i relaciona el text amb la imatge. Aquesta IA va ser creada per Katherine Crowson, ella sostenta un màster en enginyeria computacional. Per poder fer-la arribar a tothom van ser necessàries cinc persones més, que es van dedicar a fer des de traduccions fins a la interfície⁽⁴⁾. Com que el seu funcionament requereix molta potència gràfica, els creadors van aprofitar el servei que ofereix Google anomenat "*colab.research.google*" que ofereix als usuaris un ordinador potent treballant telemàticament. Depenent de l'ordinador que t'assignen aleatòriament, la IA funciona amb més o menys velocitat. Amb un ordinador que tingui una targeta gràfica de 15 GB triga aproximadament una hora per fer 5000 iteracions. L'enllaç a aquesta IA és el primer de l'apartat: altres enllaços. Vaig estar trastejant amb aquesta intel·ligència artificial i vaig notar que qualsevol creació pateix d'un o més col·lapses durant el seu funcionament, algunes més aviat, d'altres més tard. La intel·ligència artificial es bloqueja i la imatge en certa manera explota. Podem veure aquest fenomen en la Fig. 5; a la iteració 1700.



Fig. 5: Exemple de col·lapse de la IA.

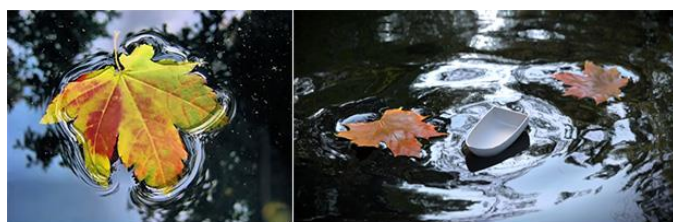
Apreciem una evolució respecte la iteració 300 i 1700, amb més detalls i efectes. De sobte, a la iteració 1750 el programa col·lapsa. Tot i això, s'agafa aquesta imatge com punt de partida i buscant i generant patrons es van fer modificacions fins que la imatge s'aproxima al text que hem escrit al principi, com podem veure a la iteració 3650. El text inicial d'aquesta creació és: amethyst, galactic, fog, Unreal Engine, hyperrealistic, que en català és: ametista, galàctic, boira, Unreal Engine i hiperrealista. Els dos últims ítems són per agregar realisme en la creació, Unreal Engine és un motor gràfic, aplicació que s'empra per crear

escenes realistes i videojocs. Les paraules del text introduït són analitzades per la IA com paraules soltes, és a dir ella les intenta lligar d'una manera o d'una altra.

Per fer la portada d'aquest treball s'ha utilitzat aquesta intel·ligència artificial. Jo vaig escriure el text per tal que la IA s'encarregués de la part artística. Vaig introduir el text: bismuth, lightning, galactic, fog, Unreal Engine, hyperrealistic, que traduït al català és: bismut (un mineral), llamp, galàctic i boira, els dos últims mots defineixen l'apartat artístic, on expresso que vull que sigui una obra realista. La imatge que vaig escollir és de la iteració 9050, és a dir que per arribar a aquest resultat el programa va crear més de nou mil fotografies. Hi ha l'opció d'adjuntar totes les imatges que ha generat per crear un vídeo, on es veu l'evolució. He penjat el vídeo al YouTube, és el segon enllaç de l'apartat: altres enllaços. Es pot veure un petit col·lapse en el minut 2:26.

Com es pot apreciar en el vídeo, les imatges creades tenen una resolució molt baixa, per això, he emprat una altra IA per a la creació de la portada, aquesta és una IA que augmenta la resolució d'una imatge a partir d'algoritmes. Evidentment és millor una imatge amb bona qualitat des d'un inici, no obstant això, aquesta IA arriba a uns resultats excepcionals. Podem utilitzar-la en format web des del tercer enllaç de l'apartat: altres enllaços.

Actualment, la IA també s'està utilitzant moltíssim en l'àmbit de la simulació. S'ha aconseguit simular de tot, des de líquids amb diferent viscositat i densitat, fins a teixits que es comporten de manera realista, o la reacció de diferents cossos davant de diversos tipus de col·lisions, com per exemple tornados o combustions. La simulació pot semblar tenir una utilitat irrellevant, però avui en dia s'està utilitzant molt en pel·lícules, tant en les convencionals, amb actors reals, com en les de dibuixos animats. Els videojocs també ho aprofiten per tenir acabats més realistes i immersius.



Real

Simulació

Fig. 6: Exemple del resultat d'una simulació feta per una IA.

Com podem veure en la Fig. 6, s'ha aconseguit simular un fenomen que passa a l'aigua. Quan un objecte es posa sobre ella es crea una membrana molecular a la superfície. Això

provoca que els objectes amb una densitat lleugerament major que la de l'aigua flotin durant un temps. Visualment s'aprecia una curvatura en l'estructura de l'aigua en el contorn de l'objecte sòlid. Aquest fenomen és anomenat tensió superficial. Podem veure com actuen les forces i velocitats en la Fig. 7. Les fletxes vermelles indiquen les velocitats de l'objecte sòlid. Les de color blau marí indiquen les velocitats que pateix, a causa de la deformació, la membrana molecular del líquid. El blau cel indica el moviment del líquid. Les vermelles

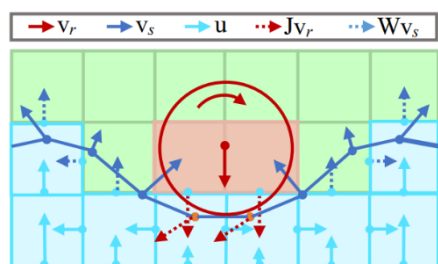


Fig. 7: Representació gràfica de les forces que ha d'interpretar la IA.

puntejades representen la velocitat de les molècules de la membrana enganxades amb el cos sòlid. Finalment, les de color blau puntejades representen el líquid que surt disparat després de la col·lisió entre el sòlid i el líquid.

Darrere d'aquesta IA hi ha un munt d'algoritmes i fórmules molt complexes, hi ha una explicació amb més detalls en el PDF (*Portable Document Format*)^[5] del setè enllaç de la webgrafia.

Durant molts anys hem vist com a les pel·lícules, la policia era capaç d'augmentar la resolució d'una imatge per tal de veure els petits detalls. Però, realment és com aprendre a fer trigonometria complexa a partir d'un llibre de matemàtiques de primària. Bàsicament, no hi ha prou informació. Això va canviar el febrer de 2019. L'empresa internacional de targetes gràfiques més important del món, Nvidia, va publicar un software capaç d'augmentar la resolució de forma realista alhora que millorava el nombre de FPS (*Frames Per Second*)^[6]. La IA va millorar l'aparença d'una imatge treballant amb molta menys potència gràfica. El projecte



Fig. 8: Comparació del rendiment i de l'apartat visual del DLSS activat i desactivat.

per exemple, veure els videojocs a 1080p (píxels)^[8] mentre que l'ordinador treballava a 720p o inferior. Oferint una millora de FPS d'entre el 20 i 50%. Podem veure la comparació del DLSS activat i desactivat a la Fig. 8. Com podem observar, quan està activat veiem una millora considerable en els FPS, que passen de 42 a 74, un 43%. Encara que sigui difícil de creure-ho, la IA és capaç de crear una imatge més nítida, gràcies a l'aprenentatge que ha tingut prèviament.

Podriem passar-nos tot el treball posant diferents exemples. Però suposo que ha quedat clar que actualment, la intel·ligència artificial conviu amb nosaltres i la seva utilització està augmentant diàriament.

2.6. FILOSOFIA I ÈTICA

Com hem pogut veure la IA està, d'una manera o altra, en les nostres vides i està ocupant llocs de treball molt importants i imprescindibles. Però ara bé, és ètic tenir ordinadors que decideixin que veiem i que deixem de veure? O encara pitjor, que triïn si una persona és útil o no. Això últim ha passat l'agost de 2021 amb Xsolla (Fig. 9), una empresa russa dedicada al món dels videojocs, que ha col·laborat amb empreses com Steam, que és la pàgina web nombre u en venda de videojocs digitals; i Twitch, branca de l'empresa Amazon enfocada a la

s'anomena Nvidia DLSS (Deep Learning Super Sample)^[7]. La IA va ser entrenada a partir d'imatges de videojocs, ella buscava algoritmes per unir-les i trobar concordances. Amb això van aconseguir que els usuaris de videojocs poguessin, per



Fig. 9: Extract de LaVanguardia sobre el cas de Xsolla.

transmissió en directe de videojocs. Xsolla ha fet una reducció del 30% en el seu nombre de treballadors seguint el consell d'una intel·ligència artificial. És realment això ètic?

Des del meu punt de vista, la decisió que va pendre Xsolla va ser molt dràstica per millorar el rendiment de l'empresa i considero que els treballadors haurien d'haver estat avisats. Tot això em recorda molt a l'edat mitjana, ja que els nobles, no tenien cap mena de pietat amb els camperols i no els importava res la seva humanitat. Per tant, crec que la utilització de la IA és correcte, ja que busca errors en el funcionament de l'empresa, però penso que s'hauria de dissenyar una altra IA que proposés llocs de treball, els quals millorarien el rendiment de l'empresa. D'aquesta manera, es podrien trobar punts dèbils i possibles millores.

Si la IA és capaç de prendre decisions sobre el destí dels humans, és capaç de dominar-nos? Això ha estat qüestionat al llarg dels últims anys, però realment ens hem de preocupar? No i sí.

Perquè per dissenyar una IA que pensi com nosaltres hauríem de conèixer el funcionament del nostre cervell al complet i actualment no en sabem gaire. Pensat d'una altra manera, la humanitat mai havia sabut com era la Terra fins a enviar una càmera fora d'aquesta. O més encara, els científics no saben realment com llueix la Via Làctia, ja que no s'ha enviat mai cap càmera més enllà d'ella. Això sí, ens podem fer una idea de com és per diferents estudis, però mai serà del tot acurat. Crec, amb molta certesa, que serà impossible aconseguir conèixer el funcionament del nostre cervell, ja que és massa complex i no podem fer un estudi com el que faríem d'una planta o d'un animal. Com que els humans no són conscients de la real diferència entre el bé i el mal, una màquina tampoc. Només pot determinar si alguna cosa s'ajusta a uns esquemes o a uns altres. Per exemple un cotxe amb un sistema de conducció autònoma no sap com reaccionar davant d'aquesta situació: mentre viatja a uns 100km/h dues persones envaeixen el carril. Com ha de reaccionar el cotxe? Frenant en sec, posant fi a la vida del conductor a causa del cop que pateix per la frenada o atropellant a les dues persones? Ni nosaltres sabem quina és l'opció més ètica. Al cotxe li és indiferent, si la reacció no ha estat programada, serà una simple elecció a l'atzar, del 50%.

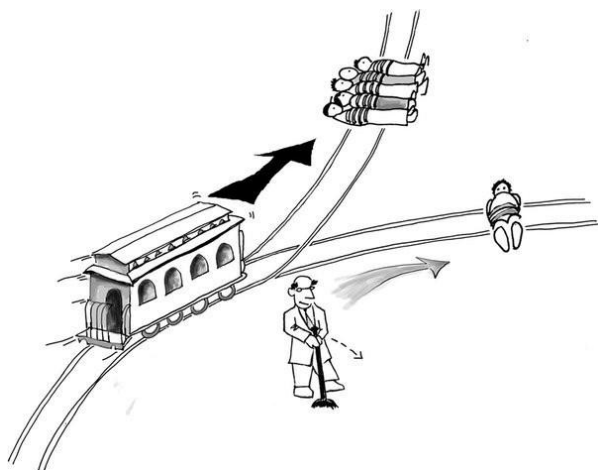


Fig. 10: Representació gràfica del problema del tramvia.

El problema plantejat és molt similar al famós dilema anomenat: el problema del tramvia. Podem veure la representació gràfica del dilema a la Fig. 10, on un tren es dirigeix cap a un grup de dues o més persones i un individu és capaç de canviar el rumb del tren per matar-ne només una, així doncs modificant el resultat. Segons una enquesta feta per Martí Montferrer, un científic amb gran

influència en xarxes socials, a Twitter, un 48% dels enquestats accionarien la palanca. Un resultat molt similar al del 50% de la reacció a l'atzar que prendria el cotxe. La principal diferència és que a un humà li resulta incòmode respondre aquesta pregunta, però a una IA li és indiferent, almenys avui en dia.

Dit això, una intel·ligència artificial pot superar la intel·ligència d'un humà? La resposta a aquesta pregunta varia depenent del que entenem com a intel·ligència. Si ho entenem com el nombre d'operacions en un període de temps, hi ha ordinadors, com el Sunway TaihuLight, que és l'ordinador més potent mai creat, que dupliquen les dels humans. Per tant, podríem dir que són més intel·ligents que els humans. Això no significa cap amenaça per a la humanitat. Però com hem vist, són capaces d'aprendre i com algunes IAs aprenen a fer caminar un cos, d'altres podrien aprendre a enganyar-nos. No sabem si algun dia podrem recrear la consciència humana en una IA, ni els físics ni els neuròlegs tenen idea de com fer-ho. Tot i que segur que es pot aconseguir. Potser, no siguem nosaltres qui ho aconseguim, però és possible que la intel·ligència artificial ho faci, que ha après a caminar, a programar i en un futur a pensar. Avui en dia podem crear una IA molt més intel·ligent que nosaltres, però sense consciència, capaç de crear altres IAs. Això, teòricament, hauria de ser inofensiu, ja que el seu objectiu és crear IAs. Però, si un humà s'interposés en el seu camí, aquesta l'apartaria. Recordem que aquesta IA seria unes quantes vegades més intel·ligent que nosaltres, que potser no tindria consciència, però sí una missió, un objectiu. Es comportaria com si fos un virus. Per complir aquest

objectiu seria capaç de superar les barreres ètiques, d'aquesta manera podria exterminar la raça humana, per tal d'aconseguir el seu objectiu.

Per concloure, actualment, les IAs són maldestres, poc exactes i amb una potència limitada, però en un futur llunyà, o potser no tant, podrien arribar a ser un problema per a la humanitat, ja que són capaces de copiar certes accions humanes amb una gran facilitat i fins i tot millorar-les. Això ho veiem en la pel·lícula de Sunspring, que va ser ideada per una IA en pocs segons, a diferència dels mesos que triga un guionista.

Tot i haver expressat la meva preocupació per les IAs, considero que són molt útils i imprescindibles pel model de vida actual. Per tant, crec que s'haurien d'utilitzar en major mesura amb l'objectiu de facilitar-nos l'existència, però amb una certa regulació, ja que ara són inofensives, però estan adquirint poder. En un futur podrien arribar a ser una arma, capaç d'exterminar la humanitat.

3. MARC APLICAT

3.1. INTRODUCCIÓ DEL PROGRAMA

Amb la creació d'aquest programa em faré una idea del funcionament d'aplicacions com Google Maps o Bing Maps a l'hora de buscar diferents rutes, totes elles amb diferents característiques. Unes són les més ràpides, d'altres les més curtes, sense peatges... A ull nu sembla bastant senzill, però és un programa molt complex, ja que cada tram ha de tenir moltíssimes anotacions, la seva velocitat límit, si hi ha algun incident, com ara un arbre envaint la via o un xoc entre cotxes... Però la principal fita és fer que trobi la ruta indicada en menys de cinc segons. Els ordinadors han evolucionat moltíssim, com és el cas del principal component d'aquests, el processador o CPU (*Central Processing Unit*)^[9] com podem veure el Fig. 11. El que s'observa en aquest gràfic és la llei de Moore, publicada per Gordon Moore el 1965, que postula: cada dos anys es duplica el

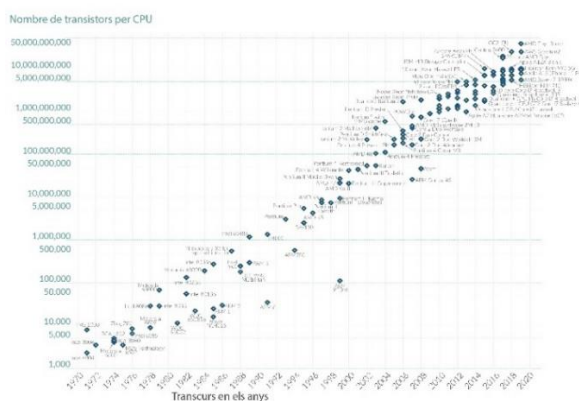


Fig. 11: Relació del nombre de transistors per xip respecte el temps.

nombre de transistors⁽⁷⁾ en les CPUs.

En conseqüència, no hauria de ser un repte crear un programa d'aquestes magnituds, per la simple raó que l'ordinador podria executar-lo sense problemes. Doncs, la veritat és que un ordinador convencional no és gaire bo provant diferents opcions consecutivament. Aquest és el punt

fort dels ordinadors quàntics, que poden provar tantes combinacions en pocs segons que són capaços de trobar la contrasenya de qualsevol usuari en molt menys temps que un ordinador convencional, que podria tardar milions d'anys provant possibles opcions. Per sort, aquests ordinadors només estan a l'abast de grans empreses com: IBM, Microsoft o la NASA. Per tant, per fer un programa que funcionés correctament en els ordinadors actuals s'hauria de descartar en gran mesura el nombre de possibles rutes.

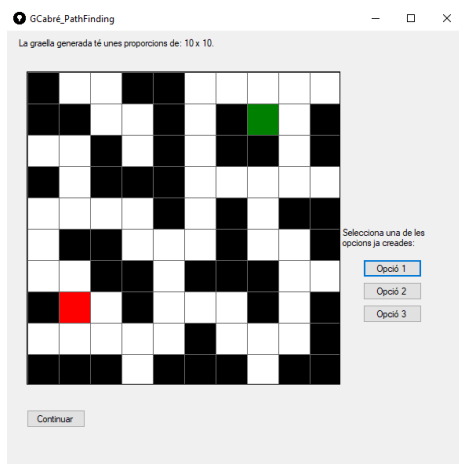


Fig. 12: Interfície del meu programa.

El que vull fer és un programa capaç de cercar la ruta més ràpida d'un punt A a un punt B. En el mapa hi haurà obstacles, tals com parets, les quals s'hauran d'evitar. A més, posaré algunes limitacions per tal de complicar el funcionament. Tot i no ser dissenyador gràfic, he creat una interfície fàcil d'utilitzar amb botons i explicacions del que hauria de fer l'usuari en tot moment. Podem veure-la a la Fig. 12. El punt vermell simbolitza l'inici de la

ruta i el verd el final. La imatge correspon a la opció: tria de diferents mapes preestablerts. A part, hi ha una opció on l'usuari pot dissenyar el mapa. Ell podrà triar la mida de la graella, per exemple: pot fer-la 13 x 7, si és el que desitja i podrà afegir-hi els obstacles en les cel·les que vulgui. També, podrà escollir la opció de selecció de mapa aleatòri, on entre 4 mapes s'escollirà un aleatòriament. Les 4 opcions són: els tres mapes o graells ja creats de forma determinada i l'últim mapa creat per l'usuari, ja que l'última creació sempre s'emmagatzema dins del programa, fins i tot si s'apaga l'ordinador.

Com s'aprecia a la Fig. 12, no és fàcil saber quina seria la ruta més curta. Aquest mapa és de 10 x 10, però hi ha l'opció de crear-ne de més grans i per tant seran més complicats d'analitzar a ull nu.

El programa que vaig utilitzar per escriure tot el codi va ser: Blend for Visual Studio 2019, un programa gratuït de la mà de Microsoft, empresa creadora de Windows.

El programa ofereix diferents opcions i formats d'execució del codi, com si de plantilles parléssim. En primera instància vaig utilitzar la plantilla anomenada: aplicació de consola, que està pensada pels sistemes operatius: Linux, MacOS i Windows i el llenguatge de programació és el C#. Al principi em va agradar bastant aquesta plantilla, ja que no era molt complicada d'utilitzar i vaig aprendre moltíssim amb ella. Però més endavant em vaig adonar que difícilment arribaria a un resultat visualment interessant. No és agradable a la vista perquè s'executa

```

A - - - X X - - X X
- - X - X - - - X
X - X - - - X - -
X X X - - - X - -
- X - - X X X - X X
- X - - - X X - - -
- X X X - - X X X -
- - - - - X - - -
X - X X X - - - B X
- - - - - X - - X

C:\Users\guill\source\repos\Prova\Prova\bin\Debug\netcoreapp3.1\Prova.exe (proceso 17912) se cerró con el código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas -> Opciones -> Depuración -> Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .
  
```

Fig. 13: Interfície del programa dissenyat per funcionar al CMD.

mitjançant la consola de Windows o CMD (*Command Prompt*)^[10]. Podem veure com llueix a la Fig. 13. Per aquesta raó, vaig decidir provar una altra plantilla, que és la que s'utilitza per a la creació de moltes aplicacions. Aquesta s'anomena: Aplicació de Windows Forms i també funciona amb C# i Windows. El canvi de plantilla em va facilitar eines tals com graells, botons, quadres de text i més, per tal de fer una interfície entenedora i ordenada. Amb la plantilla "Aplicació de Windows Forms" vaig acabar el programa, després de moltíssimes hores i més de 500 línies de codi. He publicat tot el meu programa amb una gran quantitat de petites anotacions explicant el funcionament de les parts més importants. Per descarregar-lo s'ha d'anar al sisè enllaç de l'apartat: altres enllaços. I l'explicació de com utilitzar-lo està a l'Annex 3.

Per ajudar a entendre com funcionen les connexions internes en el menú del programa he creat un mapa conceptual que es pot veure a la Fig. 14. Més endavant s'explicarà més en detall el contingut de l'ítem del mapa conceptual: Activar IA. En aquest mapa es poden distingir les 3 opcions principals per inserir un mapa o graella.

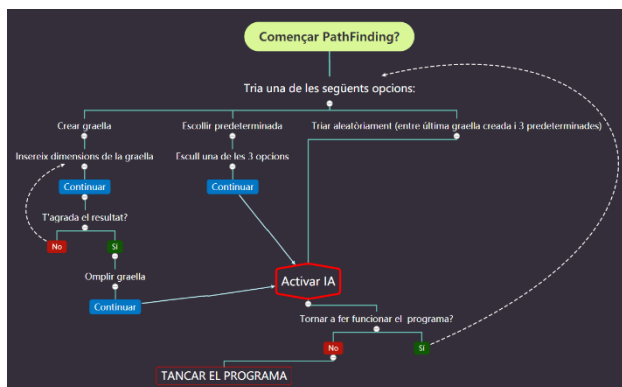


Fig. 14: Mapa conceptual del funcionament del menú.

Durant la creació del programa he tingut molts problemes, ja que tot i haver programat algunes cosetes, mai havia fet res amb aquest grau de complexitat. El programa més complex que havia fet era un cronòmetre amb un *display*⁽²⁾ i un parell de botons que va ser el meu

projecte de final de curs de programació de quart d'ESO. El vídeo del treball el podem trobar al cinquè enllaç de l'apartat: altres enllaços. Per tant, aquest projecte ha significat un gran repte per a mi. Primer he hagut de buscar tutorials, perquè no coneixia les bases sintàctiques del C#; i segon, no tenia els coneixements suficients per fer tal cosa. Hauré vist desenes de diferents vídeos, a part d'haver consultat, sense exagerar, més d'un centenar d'articles web. D'entre totes destaco la pàgina web: *Stack Overflow*. Aquí, la gent fa preguntes i l'amable comunitat que ha creat la pàgina acostuma a respondre-les.

Principalment, jo em limitava a llegir les respostes que escrivien les persones. Fins que un dia se'm va aparèixer un problema bastant gros.

Jo sabia com trobar la memòria dels mapes preestablerts a partir d'un ordinador, però quan jo posava el programa en un llapis de memòria, com que no estava en el mateix *directori*⁽¹⁾ ja establert en el codi del programa, no trobava l'arxiu on s'emmagatzemava la memòria. Per tant, el que volia era que sempre sapigués

How do I do a filePath wich depends on the location of the .exe

Asked 3 months ago · Active 3 months ago · Viewed 24 times

The screenshot shows a Stack Overflow question titled "How do I do a filePath wich depends on the location of the .exe". The question is marked as a duplicate of "get path for my .exe (duplicate)". The user describes a program that reads .txt files and wants to access them without changing the filePath manually. They provide a code snippet for filePath and ask for help. The answer, by Kieren Johnstone, explains that a relative path should be used and provides a code snippet: `string filePath = Path.GetDirectoryName(Process.GetCurrentProcess().MainModule.FileName)`.

Fig. 15: Extract de la pregunta formulada.

trobar l'arxiu sense importar la localització del programa. Després de moltes proves em vaig rendir i em va passar pel cap escriure una pregunta a *Stack Overflow*. La publicació està en el setè enllaç de l'apartat: altres enllaços i a la Fig. 15. Evidentment, per arribar a més gent vaig haver d'escriure-la en anglès. Em va sorprendre la bondat d'una persona, sobrenomenada *Hagen*, que tot i ser nou o nova en la plataforma va decidir ajudar-me. Em va proposar una línia de codi que em va solucionar el problema, a més de dirigir-me a un usuari que segons ell o ella ho explicava molt bé.

3.1.1. LIMITACIONS

El punt de vista omniscient que pot tenir qualsevol programa pot suposar un avantatge, que en certs casos no ens interessa. Per exemple, imaginem-nos un videojoc sense multijugador de trets, on els enemics són NPCs (*None Playable Character*)⁽¹¹⁾. Els enemics saben en tot moment on és el jugador i per tant, el desapareixen abans que aquest els vegi. Per aquesta simple raó, les IAs, a vegades, pateixen algunes limitacions, que generen algunes dificultats per a la IA i també pel programador, que ha de simular, d'una manera o d'una altra, el comportament humà.

Com que no volia que li resultés gaire fàcil a la IA resoldre el problema, vaig decidir que aquesta només podria analitzar les cel·les que estiguessin tocant

amb les ja explorades. Per tant, des de l'inici sabrà quines són les coordenades del punt A, però no les del punt B, que les haurà de trobar. Concretament, només disposa de la informació de les variables: AX, AY, Xtotalcages i Ytotalcages. Les dues primeres indiquen les coordenades del punt A i les altres les dimensions de la graella.

La limitació es podria treure si es desitja, però per la principal aplicació del programa considero que és millor així.

3.2. EXPLICACIÓ DEL SEU FUNCIONAMENT

3.2.1. MENÚ

Per començar, el programa té un menú, que s'actualitza cada vegada que l'usuari prem un botó. L'objectiu és proporcionar llibertat a l'hora de fer funcionar la IA, donant eines tals com: crear una graella o mapa a mesura, utilitzar-ne una de creada o fer una tria aleatòria entre les tres graelles predeterminades i l'última creada per l'usuari.

La creació té dues parts principals. En la primera, l'usuari insereix les mesures del mapa. A continuació, se li mostra com llueix i ell pot decidir si vol tornar a introduir-les o ja són del seu gust. Després, a la pantalla apareixen tres botons: paret/buit, que posen i treuen obstacles dins les cel·les, punt A i punt B, els dos últims serveixen per col·locar el punt d'inici i final del recorregut. Tot això té algunes mesures de seguretat, que com diuen els programadors, serveixen per tenir un programa més robust. I què vol dir això? Són mesures que pretenen observar totes les opcions possibles per tal que el programa no es pengi. Amb un exemple quedarà més clar: quan s'introdueixen les mides de la graella l'usuari no pot prémer el botó "continuar" fins que no ha creat una graella com a mínim de 2 cel·les. Si això no estigués, no es podrien posar els punts A i B. Una altra mesura és: el programa bloqueja l'opció d'introduir el punt A quan aquest ja s'ha introduït i el mateix passa amb el punt B. Si s'intenta, surt un missatge explicant com canviar-lo de lloc. Com aquestes mesures n'hi ha moltes més, amb l'objectiu de fer un programa robust. Quan l'usuari està satisfet amb la seva graella, es guarda dins de la memòria del programa en format .txt_[12]. Aquesta transcripció del programa al .txt la fa un tros de codi, que actua com un escàner, analitza que hi ha en cada cel·la i ho escriu. Com es veu a la Fig. 16, els buits són "-", les

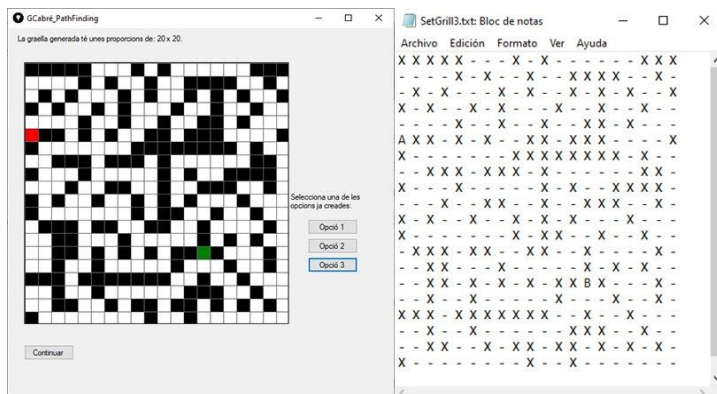


Fig. 16: Mostra de com s'emmagatzemen les graelles.

parets “X” i els punts A i B “A” “B”. Hi ha una variable anomenada *filePath* a la que se li assigna el directori de l'arxiu que conté la memòria del mapa que s'utilitzarà per a la IA. En aquest punt la localització de l'arxiu creat per l'usuari

es guarda dins de la variable *filePath*.

En segon lloc, el funcionament de l'opció d'utilitzar una graella ja creada és molt més simple que l'anterior. A la pantalla apareixen tres botons, cada un simbolitza una de les tres graelles ja creades. Al prémer qualsevol, s'executa una part del programa que té com a funció analitzar l'arxiu .txt que conté la informació de la graella assignada al botó. Aquesta part del programa l'he anomenada: *ShowSetGrill* i està més o menys a la línia 290 del programa. S'encarrega de fer un recompte de quantes files i columnes ha de tenir. Seguidament, hi ha un loop dins d'un altre loop, un és per les columnes i l'altre per les files. D'aquesta manera s'aconsegueix tenir un loop que es repeteixi tantes vegades com cel·les hi hagi, ja que s'aplica la fórmula matemàtica: l'àrea d'un rectangle és la seva base per altura. Aquesta combinació de loops analitzen cada cel·la en cerca de parets, buits i punts A i B, podem trobar-ho a la Fig. 17. El primer loop és el *foreach*, que expressa: repeteix el cos del loop tantes vegades com línies tingui l'arxiu. I el

```
foreach (String line in lines) //Loop que es repeteix per cada línia
{
    string[] cage = line.Split(' '); //Talla el string cada cop que hi ha un "espai"
    for (int XcolorCount = 0; XcolorCount < XtotalCages; XcolorCount++)
    {
        switch (cage[XcolorCount]) //TRANSCRIPTOR d'arxius .txt
        {
            case "X":
                Grill.Rows[YcolorCount].Cells[XcolorCount].Style.BackColor = Color.Black;
                break;
            case "A":
                Grill.Rows[YcolorCount].Cells[XcolorCount].Style.BackColor = Color.Red;
                AX = XcolorCount; //Defineix en AX i AY les coords del punt A
                AY = YcolorCount;
                break;
            case "B":
                Grill.Rows[YcolorCount].Cells[XcolorCount].Style.BackColor = Color.Green;
                break;
            case ".":
                Grill.Rows[YcolorCount].Cells[XcolorCount].Style.BackColor = Color.White;
                break;
        }
        YcolorCount++;
    }
}
```

Fig. 17: Fase de la IA "escàner".

segon, el *for*, diu el següent: crea una variable: *XcolorCount*, en la que per cada volta al loop se li sumi 1, repeteix *XcolorCount* mentre que *XcolorCount* sigui més petita que *XtotalCages* (variable que conté el nombre de cel·les que ha de tenir horitzontalment). La

segona instrucció, que queda entre els dos loops també és essencial, ja que talla un *string*, que recordem que és una variable que emmagatzema text, quan hi ha

un espai, seguidament emmagatzema els trossets en un *array*, variable que conté variables, anomenat *cake*. A sota del *switch* es contemplen les 4 diferents opcions: paret (X), punt A (A), punt B (B) i buit (-); quan detecta quina s'ha d'assignar, pinta la cel·la corresponent. A més, guarda les coordenades del punt A dins d'AX i AY amb la finalitat de saber on fer començar la IA.

Per acabar amb l'explicació del menú, l'opció: triar aleatòriament, és la més senzilla. Com és sabut, un ordinador no és capaç de fer tries de forma aleatòria, sinó que mitjançant algoritmes ocults aconseguix treure un nombre teòricament sense sentit. Aquests algoritmes s'han anat fent més complexos amb el pas dels anys. Un exemple és com funcionava el videojoc: *Pokémon Red* per la *Game Boy*, consola del 1989. Aquest joc aconseguia fer aparicions de criatures fantàstiques en moments "aleatoris". I com ho feia? Una variable contenia el nombre de segons que havien transcorregut des de l'inici de la partida i mitjançant una fórmula matemàtica s'alterava aquest valor, d'aquesta manera s'aconseguia un nombre suposadament aleatori. El fet de simular l'aleatorietat es coneix com a RNG (*Pseudorandom number generator*)[13]. El llenguatge de programació C# té una funció per generar nombres aleatoris i és el que jo vaig utilitzar. Com que s'ha de triar entre quatre opcions, tres d'elles són les graelles predeterminades i la restant l'última creació feta per un usuari, vaig programar-ho perquè em donés un nombre del zero al tres. En aquest punt, depenent de quin nombre hagi tocat tria una graella o una altra. Quan se sap quina s'utilitzarà, el programa executa la instrucció mencionada anteriorment: *ShowSetGrill*, la qual s'encarrega de transcriure la informació de l'arxiu .txt a la graella del programa.

3.2.2. INTEL·LIGÈNCIA ARTIFICIAL

La intel·ligència artificial que he creat es pot dividir en dues parts. A continuació les explicaré el millor possible, ometent algunes per tal de fer-ho més entenedor, ja que explicar tota la IA podria ocupar massa, perquè són unes 200 línies de codi. El codi sencer està al setè enllaç de l'apartat: altres enllaços, es pot trobar el codi, amb gran quantitat d'anotacions que expliquen moltes de les instruccions internes.

Anomeno a la primera fase "escàner". Aquesta s'encarrega de buscar el punt B, analitzant cada cel·la des del punt A. Fa la cerca mitjançant un algorisme, que

pinta les cel·les que escaneja fent-les, cada vegada, d'un color més intens, per tal d'aconseguir un degradat. Realment, el color és indiferent, per temes de contrast vaig decidir que fos de color groc, però podria haver estat rosa, taronja, blau... Com podem veure en la Fig. 18, hi ha un degradat de color groc, el qual serà essencial per a la segona part de la IA. Com es pot apreciar, en cert moment

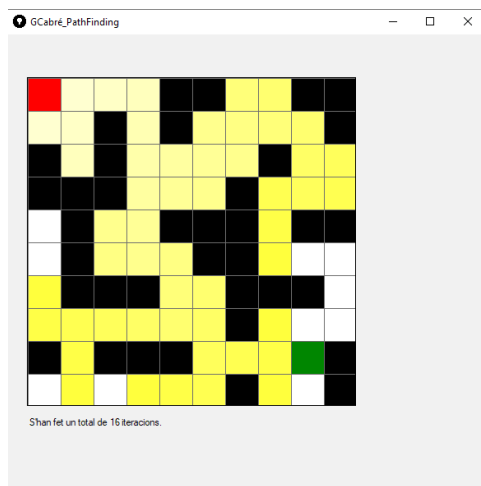


Fig. 18: Mostra del funcionament de l'“escàner”.

la “massa groga” deixa de créixer, això passa quan es troba el punt B, què és el verd. Més concretament a la iteració 16, com se'ns mostra en el quadre de text a sota de la graella. Al fer parar el procés en aquest moment permet incrementar l'eficiència del programa. Vaig aconseguir fer aquest algoritme a partir d'analitzar les quatre cel·les al voltant del punt A, el vermell. D'aquesta manera es buscava cel·les blanques per pintar-les i a la següent iteració s'utilitzaven

les cel·les al costat de les acabades de pintar, fins a trobar el quadrat verd. Però això no va ser tan fàcil com jo em pensava en un inici, ja que quan el punt vermell estava tocant un costat de la graella, com per exemple a la Fig. 18, que toca amb l'aresta superior i amb l'esquerra, el programa se saturava i deixava de funcionar. Vaig estar una bona estona intentant entendre el que passava, fins que em vaig adonar del problema. La solució va ser analitzar, per exemple, la cel·la de dalt només si la seva coordenada Y era superior a zero, aconseguint ometre la comprovació. Cada vegada que es pinta una cel·la nova es guarden les seves coordenades en una variable anomenada *MemoryXY*, que és una variable de text. En acabar la iteració, conté totes les coordenades de les cel·les a les quals s'ha avançat i les utilitza per fer la següent iteració, repetint el mateix procediment, fins a arribar al punt B. Per pintar cada iteració d'una tonalitat diferent el que es fa és que a cada iteració s'agrega un valor fix a una variable, la qual determina el color. Es passa d'un nombre a color mitjançant el famós mètode RGB (*red green blue*)^[14], que a partir d'assignar la quantitat de vermell, verd i blau s'aconsegueixen trobar 16,78 milions de colors. Cada un d'aquests tres colors llum (que no funcionen de la mateixa manera que els pigments) tenen un valor d'entre el 0 i el 255. Per fer el degradat he fet que el color per cada

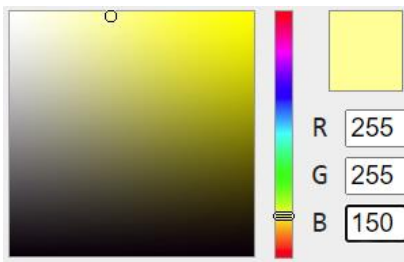


Fig.19: Exemple del funcionament del RGB. En aquest cas *ColorTimes* és 155.

iteració fos: $(255, 255, 255 - \text{ColorTimes})$ sent *ColorTimes* la variable a la qual s'agreguen 5 unitats per iteració. Els nombres dins del parèntesi indiquen la quantitat de vermell que té el color, la quantitat de verd i finalment de blau (que perd intensitat per crear el groc). Quan tots els colors tenen el valor 255, el color resultant és el blanc.

D'aquesta manera vaig aconseguir fer funcionar la primera part. He penjat una petita animació de com avança l'"escàner", aquesta està al vuitè enllaç de l'apartat: altres enllaços.

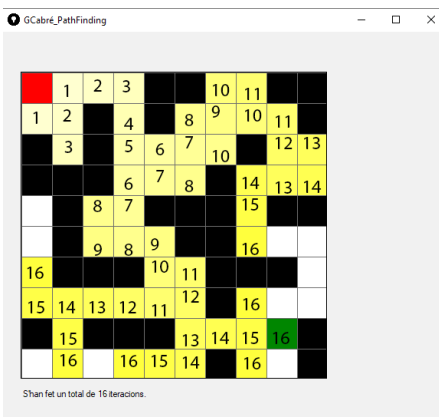


Fig. 20: Simplificació gràfica del punt de partida de la segona part de la IA.

La segona part de la IA és l'encarregada de buscar el camí més ràpid des del punt A fins al punt B. Aquesta secció utilitza el nombre de la iteració de cada cel·la de l'anterior part de la IA per tal de buscar el camí. Quan vaig haver de rumiar com guardar la iteració en la qual s'havia pintat cada cel·la per poder fer funcionar la segona part de la IA, se'm van ocórrer una gran quantitat d'opcions. Des d'emmagatzemar-lo

dins de variables, fins a crear un arxiu .txt que a on abans hi havia un "-", ara hi hagués el nombre de la iteració. Però tot això era molt complicat i reduiria el rendiment del programa. Així doncs, se'm va ocórrer utilitzar un recurs que ja estava dins del programa i així és com vaig utilitzar la mateixa graella com a forma d'emmagatzemar dades en ella. Quan ja es tenen les cel·les pintades, la IA sap el valor de l'última cel·la pintada. Amb aquesta informació busca aquesta cel·la al voltant del punt B i quan la troba, al saber que entre cel·les hi ha una variació de color de 5, és capaç de trobar la següent i la següent... fins a trobar el punt A. Aquesta part de la IA és molt més senzilla que la primera, ja que es limita a analitzar moltes menys cel·les. Per entendre-ho millor, he creat una imatge, la Fig. 20. El punt B coincideix amb la iteració nombre 16 i el camí més ràpid és anar buscant la seqüència: 16, 15, 14, 13, 12... 1. Per mostrar el camí vaig decidir pintar les cel·les per les quals passa en forma de degradat, de

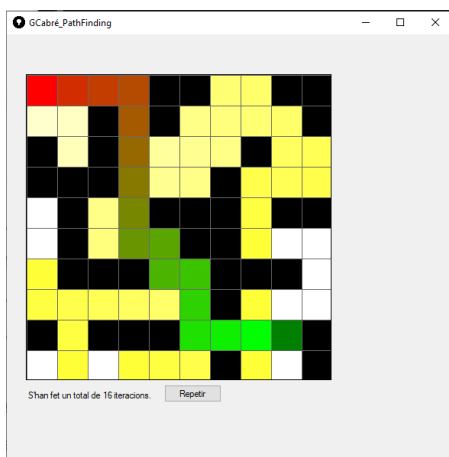


Fig. 21: Resultat de la IA.

vermell a verd. Per fer això vaig fer una fórmula molt similar a la del degradat groc. El resultat final és el de la Fig. 21.

Després de més de 500 línies de codi s'acaba el programa. Es fa aparèixer un botó amb el text "Repetir" que al ser premut, repeteix el codi del programa. D'aquesta manera, l'usuari pot tornar a utilitzar-lo, sense haver de tancar-lo i obrir-lo de nou.

3.3. POSSIBLES APLICACIONS

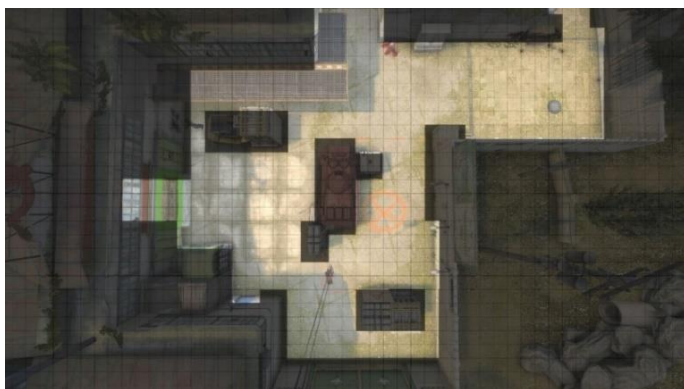


Fig. 22: Mostra gràfica de l'aplicació en videojocs.

El programa que he creates pot utilitzar-se en gran varietat d'àmbits, però el més indicat, des del meu punt de vista, és controlar el moviment dels NPCs en un videojoc.

Per fer això, s'hauria d'afegir un "tag" (etiqueta) als

obstacles del mapa del videojoc. Els obstacles podrien ser caixes, arbres, cotxes... Aquesta "tag" podria dir-se: obstacle. Amb tots els obstacles definits s'haurien d'escriure unes línies de codi per afegir-los dins de les cel·les. A la Fig. 22 es pot entendre molt millor. Un cop ja es té la graella és tan fàcil com fer funcionar el programa *PathFinding*. Els punts inicial i final, com es pot veure a la foto, són el NPC de dalt de la imatge, que està delineat de vermell i la porta, que és on es vol arribar, de color verd. Per tenir un acabat més realista s'haurien d'executar unes línies més de codi amb la finalitat de traçar el camí amb diagonals, d'aquesta manera, es reduiria la distància entre els dos punts. Això es podria fer mitjançant trigonometria.

A més a més, amb algun petit canvi també es podria fer un cercador de rutes, com el de Google Maps. Per aconseguir-ho s'hauria de fer un procediment inicial molt similar al de l'anterior aplicació. S'hauria de fer passar un escàner que

separés els carrers, carreteres, autopistes... de les zones de camps, boscos, aigua... D'aquesta manera es tindria consciència de per on es pot circular. A més també s'haurien d'afegir etiquetes a les cel·les per determinar sentits únics, velocitat, semàfors... Canviant algun aspecte dins de la IA i afegint-ne d'altres es podria aconseguir aquest programa.

Mitjançant algun canvi es podria adaptar el meu programa a una gran quantitat d'aplicacions, jo n'he anomenat un parell, però en podria haver anomenat més.

He penjat un vídeo curt al YouTube on poso a prova totes les funcions del programa, aquest està al novè enllaç de l'apartat: altres enllaços.

4. CONCLUSIONS

En aquest treball de recerca he explicat el concepte d'intel·ligència artificial, seguit de gran quantitat d'exemples. A més, he posat a prova els coneixaments que he adquirit durant el marc teòric per crear un programa informàtic i dur a terme una explicació del seu funcionament.

La complexitat del programa no m'ha permès explicar-lo de forma completa i he hagut d'obviar gran part del codi. Tot i això, he pogut explicar les parts més importants i interessants, encara que m'hauria agradat poder explicar-ne més.

Respecte a la hipòtesi: fent recerca és possible crear una intel·ligència artificial des de zero, podem conclure que la resposta a l'afirmació és positiva. Tot i que el meu programa segurament estarà molt lluny del que faria un professional, l'he completat. Com he explicat en el marc teòric, el concepte d'intel·ligència artificial no té un significat clar, però segons el corrent de pensament de McCarthy, que és el més estès, o el de Marvin Minsky, podríem dir que el meu programa sí que estaria dins de concepte. Però, si ens fixem en el pensament d'Andreas Kaplan i Michael Haenlein, no, perquè ells diuen que hauria d'analitzar els resultats per tal de millorar el seu funcionament i la meua no ho fa. Com que la de McCarthy és la més estesa i la meua IA és capaç de trobar una solució abans que un humà en terme mitjà. Podríem afirmar que qualsevol persona amb paciència i dedicació podria arribar a crear-ne una des de zero mitjançant la recerca i l'experimentació? Si jo, a partir de coneixements molt bàsics sobre la programació he arribat a un resultat amb aplicacions dins de diferents àmbits, una altra persona també podria fer-ho, però també és cert que és necessari un domini mínim de les matemàtiques, ja que el programador ha de crear fórmules matemàtiques i seqüències. Tot i això, crec que amb una bona fase d'estudi sobre les matemàtiques, la programació i les intel·ligències artificials, tothom seria capaç d'aconseguir-ho.

En quan els objectius plantejats, podem dir que s'han complert satisfactòriament mitjançant la recerca i l'exploració.

Els tres primers estan relacionats amb el marc teòric. Són: conèixer els diferents tipus d'intel·ligències artificials, comprendre el funcionament intern d'un tipus en concret d'intel·ligència artificial i conèixer el perill de la intel·ligència

artificial per a la humanitat. El primer de tots m'ha permès saber cap a on vull encaminar el programa informàtic, complementat amb el segon, que m'ha aportat els coneixements necessaris per saber com funcionarà i quins seran els seus límits. En canvi, el tercer objectiu m'ha dut cap a una reflexió, on he deixat molt clar que és una ciència de la qual no se sap cap a on avançarà i que podria suposar un problema per a la humanitat, tot i els seus infinits avantatges.

Els altres dos objectius han sigut resolts durant la fase del marc pràctic. He après a utilitzar un nou llenguatge de programació, el C#, tot i que segur que no he arribat a aprendre totes les funcions d'aquest, he adquirit un enteniment molt superior al que tenia abans d'utilitzar-lo. Actualment, em veig capacitat de fer un programa més sofisticat i això és degut, sobretot, a l'experimentació. També em considero capaç d'inicialitzar-me en un altre llenguatge de programació més fàcilment, gràcies a la recerca que he fet. L'últim objectiu l'he pogut assolir gràcies a l'anterior. Però com he explicat, el meu programa és i no és una intel·ligència artificial, depenent del corrent ideològic que agafem com a referent.

4.1. VALORACIÓ PERSONAL

Estic molt satisfet amb la recerca que he dut a terme, perquè m'ha servit per adonar-me que la programació és una activitat que realment m'apassiona i m'agradaria treballar dins d'aquest àmbit en uns anys. El marc teòric m'ha proporcionat un ventall d'informació molt ampli, el qual em servirà de cara als estudis professionals. També he après a utilitzar eines com Photoshop (editor d'imatges, emprat per fer moltes de les imatges), Premiere Pro (editor de vídeo, utilitzat per fer els vídeos), Visual Studio (processador de text, per programar) i d'altres. I per descomptat, he aconseguit dominar el llenguatge de programació C#.

5. WEBGRAFIA

1. Viquipèdia, edició pública:

https://ca.wikipedia.org/wiki/Intel%C2%B7lig%C3%A8ncia_artificial

2. BERZAL, Fernando. “Breve historia de la inteligencia artificial: el camino hacia la empresa”, Cesce (02.2017):

<https://www.cesce.es/es/-/asesores-de-pymes/breve-historia-la-inteligencia-artificial-camino-hacia-la-empresa>

3. Redacción ADP. “4 tipos de inteligencia artificial que debes conocer”, ADP (02.2021):

<https://www.apd.es/tipos-de-inteligencia-artificial/>

4. TABLADO, Fernando. “Inteligencia artificial: Definición, tipos y aplicaciones”, Grupo Atico (06.2020):

<https://protecciondatos-lopd.com/empresas/inteligencia-artificial/>

5. RUAN, Liangwang. “Solid-Fluid Interaction with Surface-Tension-Dominant Contact”, ResearchGate (05.2021):

https://www.researchgate.net/publication/351686575_Solid-Fluid_Interaction_with_Surface-Tension-Dominant_Contact

6. FARRÉS, Hector. “La empresa tecnológica que despidió al 30% de sus empleados porque la Inteligencia Artificial dijo que eran improductivos”, La Vanguardia (08.2021):

<https://www.lavanguardia.com/tecnologia/20210808/7651651/empresa-tecnologica-xsolla-despedir-empleados-improductivos-segun-inteligencia-artificial.html>

5.1. ALTRES ENLLAÇOS

1. IA capaç de generar imatges a partir de text (VQGAN+CLIP):

<https://colab.research.google.com/drive/1go6YwMFfe5MX6XM9tv-cnQiSTU50N9EeT#scrollTo=g7EDme5RYCrt>

2. Vídeo del funcionament de VQGAN+CLIP:

<https://youtu.be/EkcRvoqxuE>

3. Intel·ligència artificial que augmenta la resolució d'una imatge:

<https://bigjpg.com/>

4. Funcionament de la IA LEARN_2Dsticks_SimpleNN:
<https://youtu.be/tmDy8GkN07o>
5. Treball de final de curs de programació de 4t d'ESO:
<https://youtu.be/FWdaC0jQ1sM>
6. Carpeta on s'emagatzemen el codi i les eines per fer-lo funcionar. Per executar-lo vegeu l'últim ítem de l'annex:
<https://drive.google.com/drive/folders/1f4pCtJL7wigL-gKLfavqAaf6fVw-1Dyh?usp=sharing>
7. Enllaç de la pregunta que vaig fer a Stack Overflow:
<https://stackoverflow.com/questions/67628007/how-do-i-do-a-filepath-wich-depens-on-the-location-of-the-exe>
8. Mostra del funcionament de l'“escàner”:
<https://youtu.be/ToUHTLHYUzY>
9. Funcionament complet del meu programa:
<https://youtu.be/-vGJLrbED2Q>

7. GLOSSARI

1. **Directorí:** és un conjunt de caràcters. Té la mateixa funció que unes coordenades, trobar quelcom. Els directoris serveixen per trobar arxius ràpidament a dins de qualsevol sistema d'emmagatzematge d'un ordinador. Exemple: `F:\TR\AppPathFinding\AppPathFinding\bin\Debug\GrillSelection\SetGrill1.txt`.
2. **Display:** és un component electrònic, el qual pot ser des d'una pantalleta que mostri nombres a base de set segments il·luminats formant un 8. Fins a una pantalla d'alta resolució, com la d'una televisió.
3. **Hardware:** a diferència del software, aquest el podem veure físicament. Són tots els components d'un aparell elèctric.
4. **Interfície:** és l'apartat gràfic dels programes informàtics, en ella podem trobar botons, subfinestres, pestanyes per introduir text... Qualsevol aplicació d'avui en dia en té una.
5. **PathFinding:** és el nom que se li dona a qualsevol intel·ligència artificial que pretengui buscar un camí (*path*), que uneixi dos o més punts.
6. **Software:** és el conjunt d'instruccions que un aparell electrònic du a terme interiorment.
7. **Transistor:** és un component electrònic que quan treballa amb d'altres, connectat entre ells d'una forma específica, és capaç de fer operacions. Aquest component va reemplaçar la vàlvula de buit.

8. ABREVIATURES I SÍMBOLS

1. **TR**: treball de recerca.
2. **IA**: Intel·ligència Artificial.
3. **C#**: C sharp, és un llenguatge de programació.
4. **VQGAN+CLIP**
 1. **VQGAN**: "*Vector Quantized Generative Adversarial Network and Contrastive Language*". En català és: xarxa adversària generativa quantificada de vectors i llenguatge contrastiu.
 2. **CLIP**: "*Contrastive Language–Image Pre-training*". Que significa: entrenament contrastiu d'imatges i llenguatge.
5. **PDF**: "*Portable Document Format*", en català és: format portable de document.
6. **FPS**: "*Frames Per Second*". Vol dir: fotogrames per segon.
7. **DLSS**: "*Deep Learning Super Sample*", el seu significat és: aprenentatge per escalar una imatge.
8. **(X nombre)p**: és una forma d'expressar la resolució d'un vídeo, imatge, videojoc... avui en dia la més utilitzada és 1080p. Aquest nombre indica quants píxels d'altura té la imatge. Altres resolucions molt utilitzades són: 720p, 1440p (≈ 2K), 2160p (4K).
9. **CPU**: "*Central Processing Unit*", unitat central de processament.
10. **CMD**: "*Command Prompt*", és la consola del Windows, en ella es poden executar alguns programes.
11. **NPC**: "*None Playable Character*", personatge no jugador.
12. **.txt**: és un format que ens permet emmagatzemar text. S'acostuma a obrir amb l'aplicació bloc de notes de Windows.
13. **RNG**: "*Pseudorandom number generator*", significa: generador de nombres pseudoaleatoris.
14. **RGB**: "*red green blue*", vermell verd blau. S'utilitza per trobar un color llum.