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

Rosa Díaz, Daniel. *Implementació d'una batalla contra un cap (boss fight) adaptatiu en un videojoc*. Treball de Final de Grau (Universitat Autònoma de Barcelona), 2026 (Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/334445>

under the terms of the  license.

Implementació d'una boss fight adaptativa en un videojoc

Daniel Rosa Díaz

Resum— Aquest Treball de Fi de Grau presenta el disseny i desenvolupament d'un sistema d'intel·ligència artificial basat en aprenentatge per reforç aplicat al comportament d'un personatge no jugador en un videojoc desenvolupat amb Unity. El projecte consisteix en la creació d'un entorn experimental de combat entre un jugador i un personatge cap, en què el jugador segueix un patró d'accions predefinit mentre el cap aprèn a adaptar la seva estratègia. L'objectiu és estudiar la capacitat d'aprenentatge i adaptació d'aquestes tècniques en un context controlat, així com analitzar-ne el rendiment i les limitacions en l'àmbit de la intel·ligència artificial aplicada als videojocs.

Paraules clau—Intel·ligència artificial, aprenentatge per reforç, Unity, videojocs, personatges no jugadors.

Abstract— This Bachelor's Thesis presents the design and development of an artificial intelligence system based on reinforcement learning for controlling a non-player character in a Unity-based video game. The project consists of creating an experimental boss fight environment in which the player follows a predefined action pattern while the boss learns to adapt its strategy. The main objective is to study the learning and adaptation capabilities of reinforcement learning techniques in a controlled scenario, as well as to analyse their performance and limitations in the field of artificial intelligence for video games.

Index Terms— Artificial intelligence, reinforcement learning, Unity, video games, non-player characters.



1 INTRODUCCIÓ

La indústria dels videojocs s'ha consolidat durant les darreres dècades com la principal forma d'entreteniment digital a escala mundial, superant altres sectors com el cinema o la música tant en volum de negoci com en nombre d'usuaris. Aquest creixement ha anat acompanyat d'un augment de la complexitat dels videojocs i de les expectatives dels jugadors, fet que ha impulsat el desenvolupament de sistemes d'intel·ligència artificial (IA) cada vegada més sofisticats. Dins d'aquest context, la IA aplicada als personatges no jugadors (NPC, *Non-Player Characters*) constitueix un element fonamental per proporcionar experiències de joc més immersives, desafiants i adaptatives.

Tradicionalment, el comportament dels NPC s'ha implementat mitjançant tècniques basades en regles, com ara màquines d'estats finits o arbres de comportament. Tot i que aquests enfocaments permeten un control precís de les accions dels personatges, presenten limitacions importants quan es tracta de generar comportaments adaptatius o d'afrontar situacions no previstes durant el disseny del joc. Per aquest motiu, les tècniques d'aprenentatge automàtic han adquirit una gran rellevància en el desenvolupament de sistemes d'intel·ligència artificial per a videojocs.

Entre aquestes tècniques destaca l'aprenentatge per reforç

(*Reinforcement Learning*, RL), una branca de l'aprenentatge automàtic en què un agent aprèn a prendre decisions mitjançant la interacció amb un entorn, sense necessitat de disposar de dades etiquetades ni d'un model previ del sistema. L'agent observa l'estat de l'entorn, executa una acció i rep una recompensa que indica la qualitat de la decisió presa. Mitjançant un procés iteratiu d'exploració i explotació, l'agent desenvolupa una política que maximitza la recompensa acumulada a llarg termini.

Un dels algorismes més representatius d'aquest paradigma és el Q-learning, un mètode d'aprenentatge temporal (*Temporal Difference Learning*) que estima el valor de les parelles estat-acció i actualitza la seva política d'actuació de manera independent de les accions realment executades, fet que el converteix en un algorisme *off-policy*. La seva simplicitat conceptual, la facilitat d'implementació i els bons resultats obtinguts en nombrosos àmbits, incloent-hi la robòtica, els sistemes de control i els videojocs, han fet del Q-learning la base de moltes tècniques modernes d'aprenentatge per reforç.

Una alternativa rellevant és l'algorisme SARSA (*State-Action-Reward-State-Action*), que comparteix els principis fonamentals del Q-learning però segueix una estratègia *on-policy*. A diferència del Q-learning, SARSA actualitza les estimacions de valor utilitzant les accions que l'agent executa realment durant l'aprenentatge. Aquesta característica acostuma a produir polítiques més conservadores en entorns amb incertesa o risc, tot i que el seu comportament final depèn en gran mesura de les característiques de l'entorn, de la funció de recompensa i dels hiperparàmetres

-
- E-mail de contacte: daniel.rosa@autonoma.cat
 - Menció realitzada: Computació
 - Treball tutoritzat per: Jordi Cortes Comellas

emprats. Les diferències entre ambdós algorismes han estat objecte d'estudi en nombrosos treballs, ja que poden donar lloc a estratègies de joc significativament diferents.

L'àmbit dels videojocs constitueix un escenari especialment adequat per a l'estudi de l'aprenentatge per reforç, atès que proporciona entorns interactius, objectius clarament definits i mecanismes naturals de recompensa. En particular, els combats entre jugador i cap final (*boss fights*) representen un problema interessant des del punt de vista de la intel·ligència artificial, ja que requereixen equilibrar comportaments ofensius i defensius, adaptar-se a les accions del jugador i mantenir un nivell de dificultat adequat.

En aquest context, el present Treball de Fi de Grau planteja el disseny, desenvolupament i avaluació d'un sistema d'intel·ligència artificial basat en aprenentatge per reforç aplicat al control del comportament d'un cap dins d'un prototip de videojoc desenvolupat amb Unity. El videojoc es concep com un entorn experimental controlat, format per una arena de combat en què el jugador segueix un patró d'accions predefinit, mentre que el cap ha d'aprendre a adaptar el seu comportament a les situacions que es produeixen durant el combat. Aquest escenari permet definir formalment l'espai d'estats, el conjunt d'accions disponibles i la funció de recompensa necessària per entrenar els diferents agents.

L'objectiu principal del treball és comparar el comportament i el rendiment dels algorismes Q-learning i SARSA en aquest entorn, analitzant les diferències derivades dels seus mecanismes d'aprenentatge i la seva capacitat d'adaptació davant d'un adversari amb un comportament determinista. Com a punt de referència, s'implementa també una estratègia *baseline* en què el cap selecciona les seves accions de manera completament aleatòria durant els combats, proporcionant un criteri de comparació senzill per avaluar els beneficis de l'aprenentatge per reforç. La comparació entre els diferents enfocaments es realitza mitjançant experiments controlats i un conjunt de mètriques objectives destinades a avaluar el rendiment i el comportament dels agents durant el procés d'entrenament i en els combats.

Organització d'aquest document

Aquest document s'estructura de la manera següent. En la Secció 1 es presenta la introducció al problema, el context dels videojocs i l'aprenentatge per reforç, així com els objectius del treball. En la secció 2 s'introdueix l'ús actual de la intel·ligència artificial als videojocs. En la Secció 3.1 es descriu el disseny i la implementació de l'entorn experimental desenvolupat amb Unity, juntament amb la integració amb Python i la definició dels agents utilitzats. En la Secció 3.2 es presenten els experiments realitzats i s'analitzen els resultats obtinguts per als algorismes Q-learning i SARSA, així com la política aleatòria de referència. Finalment es recullen les conclusions principals del treball i possibles línies de treball futur.

2 ESTAT DE L'ART

La intel·ligència artificial aplicada als videojocs ha evolucionat des de sistemes basats en regles, com les màquines d'estats finits i els arbres de comportament, fins a tècniques d'aprenentatge automàtic capaces d'adaptar el comportament dels personatges no jugadors (NPC). Tot i que els enfocaments tradicionals continuen sent àmpliament utilitzats per la seva simplicitat i eficiència, presenten limitacions quan es requereix que els agents aprenguin noves estratègies o s'adaptin al comportament del jugador.

L'aprenentatge per reforç (Reinforcement Learning, RL) s'ha consolidat com una de les principals alternatives per abordar aquest problema. En aquest paradigma, l'agent aprèn interactuant amb l'entorn i seleccionant accions amb l'objectiu de maximitzar la recompensa acumulada. Dins d'aquest àmbit, Q-learning i SARSA són dos dels algorismes tabulars més coneguts. El primer és un mètode *off-policy*, que actualitza els valors assumint sempre la millor acció futura possible, mentre que el segon és un mètode *on-policy*, que aprèn a partir de les accions que realment executa la política d'exploració. Quan els espais d'estats augmenten en complexitat, aquests algorismes han donat lloc a tècniques com Deep Q-Network (DQN), que utilitzen xarxes neuronals per aproximar la funció de valor.

En l'actualitat, gran part de la recerca en aprenentatge per reforç aplicada als videojocs se centra en tècniques de Deep Reinforcement Learning, que permeten afrontar entorns amb espais d'estats molt grans o continus mitjançant xarxes neuronals. Tot i això, els algorismes tabulars continuen sent àmpliament utilitzats en entorns controlats i de menor complexitat, ja que ofereixen una implementació més senzilla i permeten analitzar amb més facilitat el procés d'aprenentatge, la convergència de les polítiques i la influència dels diferents hiperparàmetres.

3 EXPERIMENTS

3.1 Construcció de l'entorn d'avaluació

3.1.1 Implementació a Unity

Escenari de combat

L'entorn d'avaluació s'ha desenvolupat mitjançant Unity 6 en un escenari bidimensional que representa un combat entre un jugador i un cap (boss). L'escena està formada per un únic terreny pla que elimina obstacles o elements externs que puguin influir en el comportament dels agents, de manera que les decisions depenen exclusivament de l'estat del combat.

Els dos personatges disposen de components físics basats en el sistema Rigidbody2D i Collider2D de Unity, que permeten gestionar el moviment, les col·lisions i la detecció dels atacs.

Jugador

El jugador no utilitza cap mecanisme d'aprenentatge, sinó que segueix un patró d'accions completament determinista implementat mitjançant una seqüència predefinida d'ordres. Aquest patró inclou:

- aproximació al cap,
- atacs cos a cos,
- atac màgic,
- moviments d'allunyament,
- períodes d'espera.

Cada acció s'executa de forma seqüencial i el patró es repeteix contínuament durant el combat, proporcionant un adversari amb comportament completament previsible i reproducible. El jugador decideix quan atacar a partir de la distància respecte al cap i de la detecció de col·lisions mitjançant una zona d'atac específica.

Cap (Boss)

El cap constitueix l'agent intel·ligent del sistema i és l'element sobre el qual s'apliquen els diferents algorismes d'aprenentatge.

Les accions disponibles són:

- atacs cos a cos,
- atac màgic,
- acció defensiva basada en un escut,
- maniobra de retrocés.

El sistema controla automàticament l'orientació del personatge perquè sempre miri cap al jugador. A més, l'acció defensiva activa temporalment un estat d'invulnerabilitat mitjançant un col·lisionador específic, mentre que la maniobra de retrocés genera un moviment en direcció oposada al jugador durant un interval de temps determinat.

Sistema de combat

Tant el jugador com el cap comparteixen un sistema comú de gestió de vida i dany implementat mitjançant un component específic. Ambdós personatges disposen inicialment de 100 punts de vida i poden rebre dany a través dels diferents atacs disponibles durant el combat. Quan un atac impacta sobre l'adversari, el sistema detecta la col·lisió mitjançant col·lisionadors independents i aplica una quantitat fixa de 10 punts de dany, actualitzant l'estat del personatge afectat. Per evitar múltiples impactes consecutius en un mateix instant, després de rebre un atac s'activa un breu període d'invulnerabilitat temporal. Addicionalment, el cap pot obtenir un estat d'invulnerabilitat mentre manté activa l'acció defensiva basada en l'ús de l'escut. El sistema també gestiona les animacions associades als impactes i a la derrota dels personatges. Quan els punts de vida d'un dels combatents arriben a zero, es determina automàticament el vencedor, es registra el resultat del combat i es dona per finalitzada la iteració corresponent.

Gestió de múltiples iteracions

Per facilitar l'entrenament i l'avaluació dels agents d'aprenentatge per reforç, Unity incorpora un gestor automàtic encarregat de controlar l'execució seqüencial dels combats. En finalitzar cada iteració, el sistema registra el resultat obtingut, actualitza el comptador d'iteracions executades i reinicia automàticament l'escena de combat mantenint les mateixes condicions experimentals. Aquest procés es repeteix fins a assolir el nombre d'iteracions establert per a l'experiment, permetent executar sèries completes d'entrenament, com les 1000 iteracions utilitzades en aquest treball, sense necessitat d'intervenció manual i garantint la consistència del procés d'avaluació.

3.1.2 Integració amb Python

Comunicació Unity-Python

La integració entre l'entorn de simulació desenvolupat a Unity i els algorismes d'aprenentatge implementats en Python es realitza mitjançant una comunicació bidireccional basada en sockets TCP. Unity actua com a client, establint una connexió amb un servidor Python que executa l'algorisme corresponent i manté actualitzada la política de decisió del cap durant el combat.

Informació transmesa

La informació intercanviada entre Unity i Python correspon als elements necessaris per definir l'estat del combat i actualitzar el procés d'aprenentatge. Abans de cada decisió, Unity envia la distància existent entre el jugador i el cap, així com els punts de vida actuals de tots dos combatents. Aquestes variables constitueixen la representació de l'estat utilitzada pels agents d'aprenentatge.

Durant el desenvolupament del combat també es transmeten els esdeveniments associats als impactes dels atacs, indicant quin personatge ha infligit el dany, quin l'ha rebut i la quantitat de punts de vida perduts. Addicionalment, qualsevol modificació dels punts de vida dels combatents és comunicada al sistema d'aprenentatge per mantenir una representació coherent de l'estat de l'entorn.

Quan un dels personatges es queda sense vida, Unity envia el resultat final del combat indicant el vencedor de l'episodi. Aquesta informació permet a l'algorisme calcular les recompenses finals associades a la victòria o la derrota i actualitzar els valors de la seva política abans d'iniciar una nova iteració.

Execució automàtica de les iteracions

La comunicació entre Unity i Python es complementa amb un sistema d'execució automàtica dels combats que facilita el procés d'entrenament dels agents. Una vegada finalitza un episodi, Unity registra el resultat del combat i actualitza el nombre d'iteracions executades. Si encara no s'ha assolit el nombre total d'iteracions configurat, l'escena de combat es reinicia automàticament mantenint la connexió amb el

procés de Python i iniciant un nou episodi sota les mateixes condicions experimentals.

Aquest mecanisme permet executar seqüències completes d'entrenament, com les 1000 iteracions utilitzades en els experiments, sense intervenció manual i garantint que tots els agents siguin avaluats en un entorn consistent i completament reproducible.

3.2 Comparació entre Q-learning i SARSA

El primer experiment té com a objectiu comparar el comportament dels algorismes Q-learning i SARSA en el problema de presa de decisions del cap (boss), així com avaluar el seu rendiment respecte a una política aleatòria utilitzada com a referència (baseline). L'objectiu principal és analitzar les diferències en el procés d'aprenentatge i en la qualitat de les polítiques obtingudes quan tots els agents s'enfronten a les mateixes condicions experimentals.

Els tres agents comparteixen el mateix espai d'estats i el mateix conjunt d'accions definits en l'entorn d'avaluació. L'estat del sistema es representa mitjançant la discretització de la distància entre els combatents i dels punts de vida del jugador i del cap, generant un espai de 27 estats possibles.

Distància entre agents	Proper, mitjà, lluny
Vida jugador	Baixa, mitjana, alta
Vida boss	Baixa, mitjana, alta

Per garantir una comparació equitativa, els algorismes Q-learning i SARSA utilitzen els mateixos hiperparàmetres d'entrenament. El factor d'aprenentatge es fixa en $\alpha = 0,3$, proporcionant un equilibri entre l'adaptació a noves experiències i la conservació del coneixement adquirit. El factor de descompte s'estableix en $\gamma = 0,95$, prioritzant les recompenses futures i afavorint l'aprenentatge d'estratègies a llarg termini. L'exploració es duu a terme mitjançant una política ϵ -greedy amb un valor inicial de $\epsilon = 1$, que decreix progressivament segons un factor de 0,99 després de cada iteració fins a un valor mínim de $\epsilon = 0,05$. Els valors dels hiperparàmetres s'han seleccionat després de diverses proves preliminars, buscant un equilibri entre velocitat d'aprenentatge i estabilitat. Un valor d' $\alpha=0.3$ permet incorporar nova informació sense oblidar ràpidament l'experiència prèvia, mentre que $\gamma=0.95$ prioritza les recompenses futures. El valor inicial $\epsilon=1$ afavoreix una exploració completa durant les primeres iteracions.

Els dos algorismes comparteixen igualment la mateixa funció de recompensa. Aquesta ha estat dissenyada manualment amb l'objectiu d'afavorir un comportament ofensiu però coherent amb el context del combat. A més del resultat final de l'episodi, es recompensa la selecció d'accions apropiades segons la distància entre els combatents i es penalitzen tant les accions repetitives com el dany rebut pel Boss.

1. Penalització temporal per pas

A cada iteració del combat s'aplica una penalització constant per afavorir la rapidesa de resolució:

- $R_{temps} = -0.25$

2. Recompensa per repetició d'accions

Per evitar comportaments repetitius, s'aplica una penalització creixent quan es repeteix la mateixa acció:

- Si es repeteix la mateixa acció consecutivament:
 $R_{rep} = -0.5 \times (n - 1)$
on n és el nombre de repeticions consecutives.
- Si l'acció és nova (primera execució):
 $R_{novetat} = +0.3$

3. Recompensa per acció segons distància

La recompensa depèn de l'acció seleccionada i la distància amb l'enemic.

Encantament (SPELL)

- Distància curta: -1.0
- Distància mitjana: -0.2
- Distància llarga: $+0.5$

Cos a cos (MELEE)

- Distància curta: $+0.8$
- Distància mitjana: $+0.1$
- Distància llarga: -1.5

Escut Independentment de la distància: -2.0

Retrocés

- Distància curta: $+0.8$
- Distància mitjana: $+0.2$
- Distància llarga: -1.5

4. Recompensa per dany infligit al jugador

El dany causat pel Boss al jugador es pondera segons l'acció prèvia i la distància:

Encantament (SPELL)

- Curta: $-0.5 \times dany$
- Mitjana: $+1.5 \times dany$
- Llarga: $+3.0 \times dany$

MELEE

- Curta: $+3.0 \times dany$
- Mitjana: $+1.5 \times dany$
- Llarga: $-1.0 \times dany$

ESCUT

- $+0.5 \times dany$

RETROCÉS

- Curta: $+2.0 \times dany$
- Mitjana o llarga: $+1.0 \times dany$

5. Penalització per dany rebut pel Boss

Quan el Boss rep dany, s'aplica una penalització proporcional:

- Si l'última acció és ESCUT:
 $R_{dany} = -2.0 \times dany \text{ rebut}$
- En cas contrari:
 $R_{dany} = -4.0 \times dany \text{ rebut}$

6. Recompensa final del combat

El resultat global de l'episodi s'incorpora amb una recompensa de gran magnitud:

- Victòria del Boss: $+150$
- Derrota del Boss: -150

Com a punt de referència, s'implementa una política aleatòria en què el cap selecciona cadascuna de les seves accions amb probabilitat uniforme i sense cap mecanisme d'aprenentatge, mantenint inalterades la resta de condicions experimentals. Aquesta política permet establir una línia base sobre la qual comparar el rendiment dels algorismes d'aprenentatge per reforç.

Cada agent s'avalua durant un conjunt de 1000 iteracions independents sota les mateixes condicions experimentals. Durant aquest procés es registren diferents mètriques per analitzar el comportament dels agents, incloent l'evolució del paràmetre d'exploració ϵ , el valor Q mitjà, la recompensa acumulada i la taxa de victòries del cap. Aquestes mètriques permeten estudiar tant la velocitat de convergència dels algorismes com la qualitat de les polítiques apreses i comparar-les amb el comportament de la política aleatòria.

3.2.1 Resultats experimentals

En aquest apartat es presenten els resultats obtinguts en la comparació entre els tres agents avaluats: Q-learning, SARSA i la política aleatòria (baseline). L'anàlisi se centra en l'evolució de les mètriques definides durant l'entrenament i permet contrastar el comportament dels diferents enfocaments sota les mateixes condicions experimentals.

Evolució del paràmetre d'exploració epsilon (ϵ)

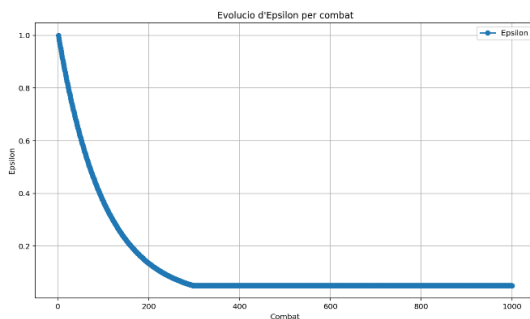


Figura 1: Evolució paràmetre ϵ amb Q-learning

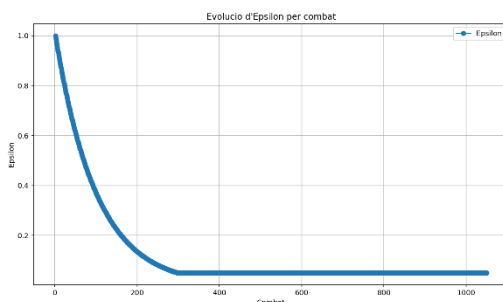


Figura 2: Evolució paràmetre ϵ amb sarsa

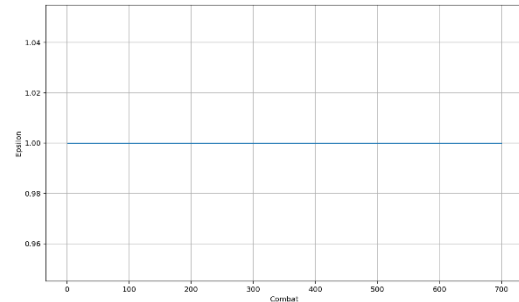


Figura 3: Evolució paràmetre ϵ amb accions aleatòries

La política aleatòria no utilitza un mecanisme d'exploració basat en ϵ , ja que les accions es seleccionen de manera completament estocàstica i uniforme. Per tant, en aquesta mètrica només són comparables Q-learning i SARSA, els quals mostren una evolució idèntica en termes de reducció d'exploració, garantint una transició controlada cap a l'exploració de la política apresada.

Recompensa acumulada per episodi

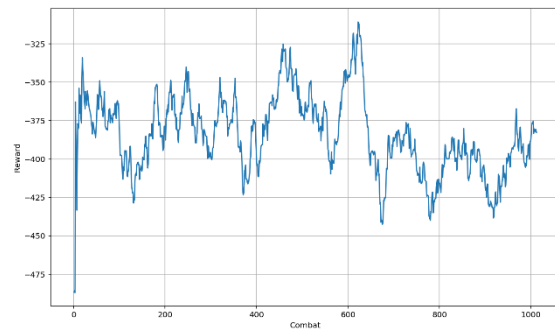


Figura 4: Taxa recompenses per episodi Aleatori

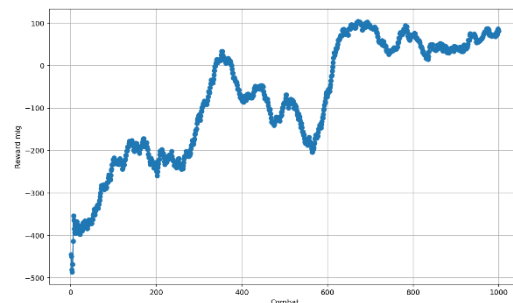


Figura 5: Taxa recompenses per episodi Q-learning

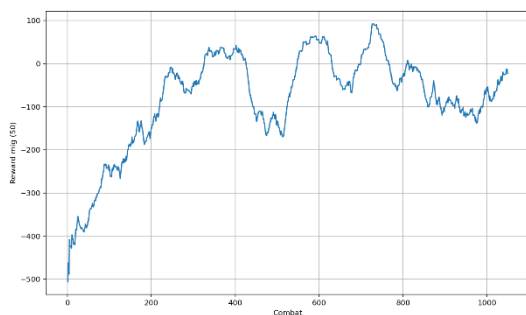


Figura 6: Taxa recompenses per episodi Sarsa

Taxa de victòries del cap

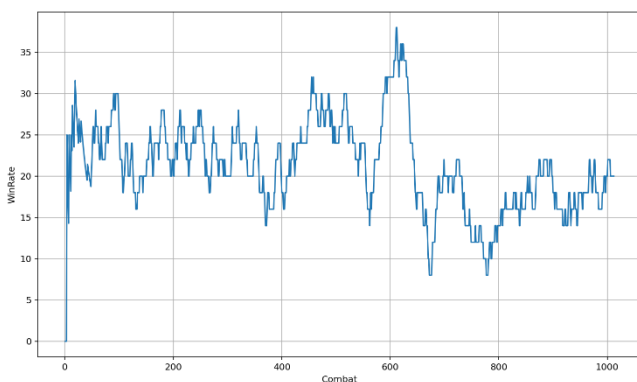


Figura 7: Taxa de victòries per episodi aleatori.

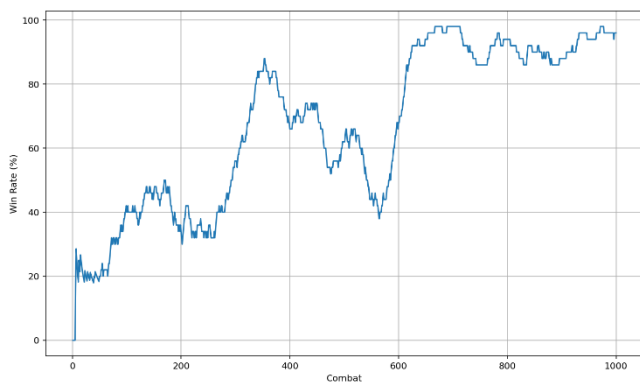


Figura 8: Taxa de victòries per episodi Q-learning.

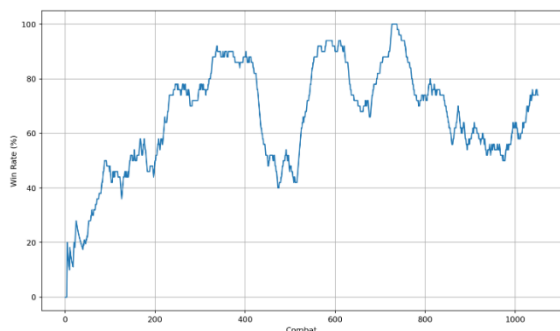


Figura 9: Taxa de victòries per episodi Sarsa.

3.2.2 Comparació global dels agents i síntesi dels resultats

En aquest apartat es resumeixen i integren els principals resultats obtinguts a partir de l'anàlisi de les diferents mètriques utilitzades durant l'experiment, amb l'objectiu d'avaluar el comportament global dels agents i la seva capacitat d'aprenentatge en l'entorn de combat.

En primer lloc, es confirma una diferència clara i consistent entre els agents basats en aprenentatge per reforç (Q-learning i SARSA) i la política aleatòria (baseline). Aquesta última no incorpora cap mecanisme d'optimització ni d'adaptació al llarg del temps, fet que es reflecteix en una absència total de millora tant en la recompensa acumulada com en la taxa de victòries. El seu comportament es manté estocàstic i invariant durant totes les iteracions, servint únicament com a punt de referència per a la comparació del rendiment.

En canvi, tant Q-learning com SARSA mostren una evolució positiva i progressiva en les mètriques analitzades, indicant que ambdós algorismes són capaços d'aprendre una política efectiva a partir de la interacció amb l'entorn. Aquesta millora està directament relacionada amb el procés de reducció progressiva del paràmetre d'exploració ϵ , que permet passar d'una fase inicial dominada per l'exploració de l'espai d'accions a una fase final on predomina l'explotació de les estratègies apreses. Aquest equilibri entre exploració i explotació resulta clau per a la convergència dels algorismes.

Pel que fa a la taxa de victòries, s'observen diferències importants en el comportament dels dos agents. Q-learning presenta una convergència més clara i definida, assolint una estabilització aproximada a partir dels 600 episodis, moment a partir del qual la taxa de victòries es manté de forma majoritària en un rang entre el 80% i el 100%. Tot i això, es detecta una caiguda puntual al voltant dels 900 episodis, amb valors propers al 70%, seguida d'una recuperació ràpida, fet que indica un comportament generalment estable però amb certa sensibilitat a situacions específiques del combat.

Per contra, SARSA mostra un comportament menys estable i més variable, amb una fase inicial de creixement fins aproximadament els 700 episodis, on arriba a assolir valors propers al 100%. No obstant això, a partir d'aquest punt s'observa una desestabilització significativa, amb fluctuacions constants que oscil·len aproximadament entre el 50% i el 100%, sense arribar a una convergència clara. Aquest comportament suggereix una major dependència de les trajectòries d'exploració i una menor estabilitat en la política final.

En l'entorn desenvolupat i sota les condicions experimentals utilitzades, els resultats suggereixen que Q-learning aconsegueix una política més robusta, estable i consistent en fases avançades de l'entrenament, mentre que SARSA, tot i obtenir bons resultats en determinats moments,

presenta una major variabilitat i una menor estabilitat global en el seu comportament.

3.3 Impacte en la exploració del valor ϵ

3.3.1 Resultats experimentals

En aquest apartat es mostren els resultats d'executar 1000 iteracions amb Q-learning i Sarsa modificant epsilon per que inicialment sigui 0.5 i que decreixi progressivament amb un factor de 0.995.

Recompensa acumulada per episodi

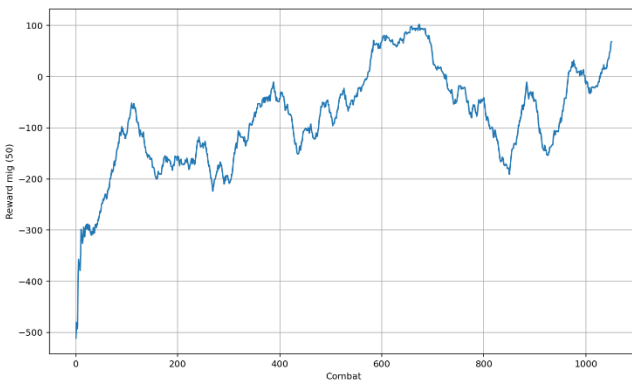


Figura 10: Taxa recompensa per episodi Sarsa ϵ inicial = 0.5

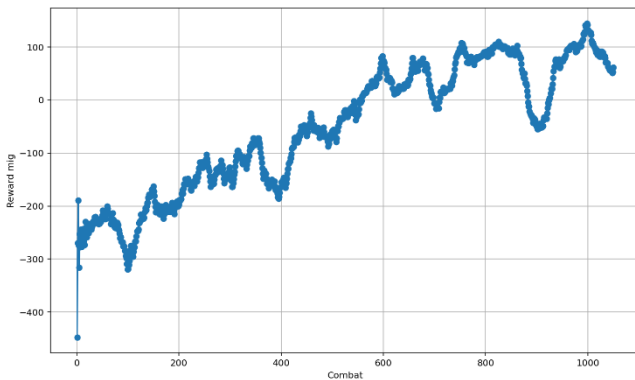


Figura 11: Taxa recompensa per episodi Q-learning ϵ inicial = 0.5

Taxa de victòries del cap

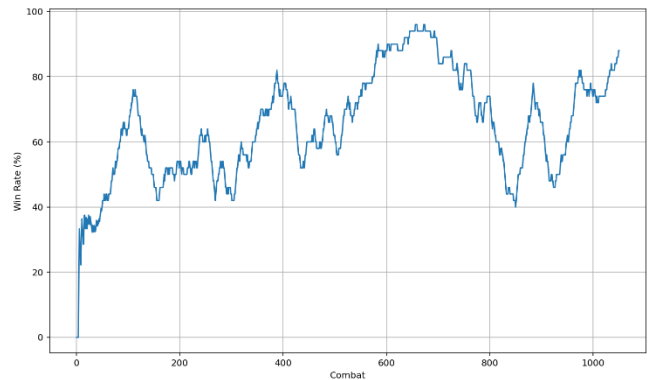


Figura 12: Taxa victories per episodi Sarsa amb ϵ inicial = 0.5

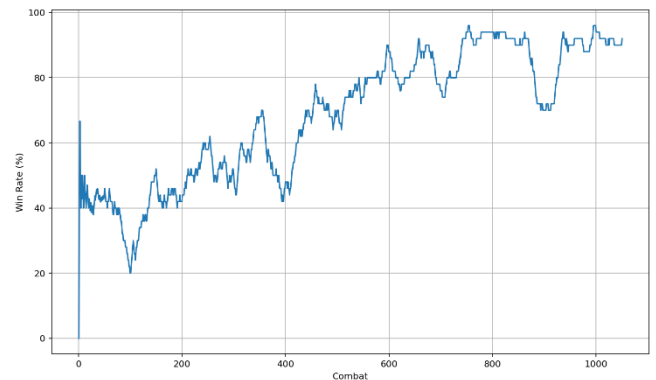


Figura 13: Taxa victories per episodi Q-learning amb ϵ inicial = 0.5

3.3.2 Comparació global dels agents i síntesi dels resultats

L'anàlisi dels resultats obtinguts permet avaluar l'impacte que té la reducció del valor inicial del paràmetre d'exploració ϵ sobre el procés d'aprenentatge dels algorismes Q-learning i SARSA. En aquest experiment es manté la mateixa configuració d'hiperparàmetres que en l'apartat anterior, modificant únicament el valor inicial d' ϵ , que passa d'1 a 0,5, amb l'objectiu d'analitzar com afecta el balanç entre exploració i explotació.

En el cas de Q-learning, amb $\epsilon = 1$ l'agent presenta una fase inicial d'aprenentatge relativament irregular, conseqüència de l'elevat nivell d'exploració. Tot i aquesta variabilitat, a partir d'aproximadament els 600 combats la política convergeix i la taxa de victòries es manté majoritàriament entre el 80 % i el 100 %, amb un comportament força estable fins al final de l'entrenament. Quan el valor inicial es redueix a $\epsilon = 0,5$, l'aprenentatge és més progressiu i constant durant les primeres iteracions, ja que l'agent aprofita abans el coneixement adquirit. No obstant això, apareix una caiguda puntual al voltant dels 900 combats, on la taxa de

victòries disminueix aproximadament fins al 70 %. Aquesta davallada és temporal i l'agent recupera ràpidament el seu rendiment, finalitzant novament amb valors propers al 90 %. Això indica que la reducció d' ϵ millora la regularitat de l'aprenentatge inicial, encara que no elimina completament les fluctuacions durant les fases avançades.

Pel que fa a SARSA, les diferències entre ambdues configuracions són més significatives. Amb $\epsilon = 1$, l'agent mostra un creixement sostingut de la taxa de victòries durant aproximadament els 400 primers combats. A partir d'aquest punt, però, el comportament esdevé molt més inestable, amb oscil·lacions contínues que es mantenen fins al final de l'entrenament i impedeixen una convergència clara de la política. En canvi, amb $\epsilon = 0,5$, l'evolució és considerablement més estable. Tant la recompensa acumulada com la taxa de victòries augmenten de forma progressiva durant la major part de l'entrenament. En canvi a partir dels 700 combats es torna molt més inestable amb una caiguda fins al 40% de victòries, però a partir dels 900 combats torna a pujar fins al 70-80%.

En conjunt, els resultats indiquen que la reducció del valor inicial d' ϵ afavoreix un procés d'aprenentatge més estable, especialment en el cas de SARSA, que redueix notablement la variabilitat observada amb una exploració inicial completa, encara que es torna més inestable al final. En Q-learning, la millora és menys pronunciada, ja que l'algorisme aconsegueix convergir correctament en ambdues configuracions; tanmateix, amb $\epsilon = 0,5$ l'aprenentatge inicial és més consistent, mentre que amb $\epsilon = 1$ la convergència és més lenta però acaba estabilitzant-se de manera satisfactòria. Aquests resultats posen de manifest que l'elecció del valor inicial d' ϵ influeix directament en la velocitat d'aprenentatge i en l'estabilitat de la política obtinguda, essent especialment rellevant en algorismes on-policy com SARSA, que són més sensibles a les decisions preses durant la fase d'exploració.

4 CONCLUSIÓ

En aquest treball s'ha desenvolupat i avaluat un sistema d'intel·ligència artificial basat en aprenentatge per reforç per al control del comportament d'un cap (boss) en un entorn de videojoc creat amb Unity. L'objectiu principal ha estat comparar el rendiment dels algorismes Q-learning i SARSA, utilitzant una política aleatòria com a referència.

La construcció de l'entorn ha permès definir un escenari controlat i reproduïble, amb una comunicació eficient entre Unity i Python mitjançant sockets, facilitant l'entrenament automàtic dels agents.

Els resultats mostren clarament que els agents d'aprenentatge per reforç superen la política aleatòria en totes les mètriques, evidenciant la seva capacitat d'adaptació. La

baseline, en canvi, manté un comportament constant sense millora.

Entre els dos algorismes, Q-learning presenta un comportament més estable i un millor rendiment final, amb una taxa de victòries que s'estabilitza aproximadament a partir dels 600 episodis entre el 80% i el 100%, tot i algunes fluctuacions puntuals. SARSA, en canvi, mostra una major variabilitat i una manca d'estabilització clara en fases avançades.

En conjunt, a partir dels experiments realitzats es pot observar que Q-learning ofereix un millor equilibri entre rendiment i estabilitat en aquest entorn de combat, mentre que SARSA presenta un comportament més irregular. Finalment, el treball confirma la viabilitat de l'aprenentatge per reforç per al desenvolupament d'IA en videojocs i obre la porta a futures millores en la representació de l'estat i les funcions de recompensa.

Una limitació d'aquest estudi és que cada configuració s'ha avaluat mitjançant una única execució d'entrenament. En treballs futurs seria convenient repetir els experiments amb diferents llavors aleatòries i analitzar la mitjana i la desviació estàndard dels resultats per obtenir conclusions estadísticament més robustes.

En conjunt, els resultats obtinguts indiquen que els algorismes tabulars continuen essent una alternativa viable en entorns reduïts i amb espais d'estats limitats. Tot i que tècniques modernes basades en Deep Reinforcement Learning permeten afrontar problemes molt més complexos, Q-learning i SARSA ofereixen una implementació senzilla, interpretable i amb uns costos computacionals reduïts, fet que els converteix en una opció adequada per a prototips i entorns experimentals com el desenvolupat en aquest treball.

ÚS DE LA IA GENERATIVA

En el desenvolupament inicial del present Treball de Fi de Grau s'ha utilitzat una eina d'IA generativa com a suport en les fases preliminars de planificació i estructuració del projecte.

Concretament, la IA generativa s'ha emprat per:

- Clarificar conceptes teòrics relacionats amb l'aprenentatge per reforç
- Ajudar a estructurar la planificació temporal del projecte.
- Refinar la definició dels objectius generals i específics.
- Revisar l'organització del backlog i la descomposició del treball en fases.
- Millorar la redacció i formalització d'alguns apartats preliminars de l'informe.

La IA generativa no s'utilitza per generar el codi final del projecte ni per automatitzar la implementació dels algorismes d'aprenentatge.

L'ús de la IA generativa s'ha concebut com una eina de suport conceptual i organitzatiu, similar a l'ús d'altres recursos de consulta (manuals, documentació tècnica o fòrums

especialitzats), contrastant sempre la informació amb fonts acadèmiques verificables quan ha estat necessari. Així mateix, qualsevol aportació conceptual rellevant ha estat revisada críticament abans d'incorporar-la al disseny del projecte, garantint l'autoria pròpia del treball i el compliment dels principis d'integritat acadèmica.

AGRAÏMENTS

Vull expressar el meu agraïment al meu tutor del Treball de Fi de Grau, Jordi Cortes Comellas, per la seva orientació, suport i seguiment durant tot el desenvolupament del projecte. Els seus consells i observacions han estat fonamentals per a la correcta evolució del treball i per a la millora dels resultats obtinguts.

BIBLIOGRAFIA

Llibres

- [1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA: MIT Press, 2018.
- [2] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Harlow: Pearson, 2021.
- [3] I. Millington and J. Funge, *Artificial Intelligence for Games*, 3rd ed. Boca Raton, FL: CRC Press, 2016.
- [4] G. N. Yannakakis and J. Togelius, *Artificial Intelligence and Games*. Cham: Springer, 2018.

Articles de revista

- [5] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, pp. 279–292, 1992.
- [6] C. B. Browne et al., "A survey of Monte Carlo Tree Search methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1–43, Mar. 2012.

Informes tècnics

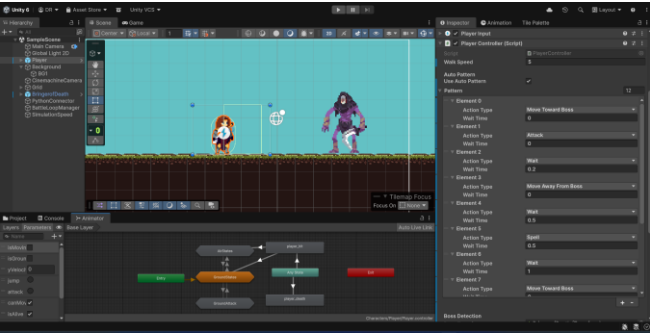
- [7] G. A. Rummery and M. Niranjan, "On-line Q-learning using connectionist systems," University of Cambridge, Cambridge, UK, Tech. Rep., 1994.

Recursos d'Internet

- [8] Artificial Intelligence. (s. f.). *Reinforcement Learning Tutorial Series* [Online]. Available: https://www.youtube.com/watch?v=QHJIXSkwmjo&list=PLyH-qXFkNSxmDU8ddesIEAtmXIDRLPd_V – Última consulta: 26/03/2026.
- [9] Unity Technologies. (s. f.). *Unity Documentation* [Online]. Available: <https://docs.unity3d.com> – Última consulta: 26/03/2026.
- [10] A. ResearchGate. (s. f.). *Q-Learning Algorithms: A Comprehensive Classification and Applications* [Online]. Available: https://www.researchgate.net/publication/335805245_Q-Learning_Algorithms_A_Comprehensive_Classification_and_Applications – Última consulta: 14/06/2026.

APÈNDIX

A1. ENTORN UNITY



Imatge 1: Entorn Unity.